Transferable Structure-based Adversarial Attack of Heterogeneous Graph Neural Network

Yu Shang
Department of Electronic Engineering, Tsinghua University
Beijing, China
shangy21@mails.tsinghua.edu.cn

Yudong Zhang
Department of Electronic Engineering, Tsinghua University
Beijing, China
zhangyd16@mails.tsinghua.edu.cn

Jiansheng Chen*
University of Science and Technology Beijing
Beijing, China
jschen@ustb.edu.cn

Depeng Jin
Department of Electronic Engineering, Tsinghua University
Beijing, China
jindp@tsinghua.edu.cn

Yong Li
Department of Electronic Engineering, Tsinghua University
Beijing, China
liyong07@tsinghua.edu.cn

ABSTRACT

Heterogeneous graph neural networks (HGNNs) have achieved remarkable development recently and exhibited superior performance in various tasks. However, recently HGNNs have been shown to have robustness weakness towards adversarial perturbations, which brings critical pitfalls for real applications, *e.g.* node classification and recommender systems. In particular, the transfer-based black-box attack is the most practical method to attack unknown models and poses a great threat to the reliability of HGNNs. In this work, we take the first step to explore the transferability of adversarial examples of HGNNs. Due to the overfitting of the source model, the adversarial perturbations generated by traditional methods usually exhibit unpromising transferability. To address this problem and boost adversarial transferability, we expect to seek common vulnerable directions of different models to attack. Inspired by the observation of the notable commonality of edge attention distribution between different HGNNs, we propose to guide the perturbation generation toward disrupting edge attention distribution. This edge attention-guided attack prioritizes the perturbation on edges that are more likely to be given common attention by different models, which benefits the transferability of adversarial perturbations. Finally, we develop two edge attention-guided attack methods towards heterogeneous relations tailored for HGNNs, called EA-FGSM and EA-PGD. Extensive experiments on six representative models and two datasets verify the effectiveness of our methods and form an unprecedented transfer robustness benchmark for HGNNs.

CCS CONCEPTS

• **Computing methodologies** → **Adversarial learning**.

KEYWORDS

Adversarial attack; Structure-based attack; Transferability; Heterogeneous graph neural network

ACM Reference Format:
Yu Shang, Yudong Zhang, Jiansheng Chen, Depeng Jin, and Yong Li. 2023. Transferable Structure-based Adversarial Attack of Heterogeneous Graph Neural Network. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583780.3615095

1 INTRODUCTION

Heterogeneous graph neural networks (HGNNs) have attracted increasing attention in recent years due to their wide application in the real world, *e.g.*, recommender system [3, 14, 18, 22] and security [5, 10, 11, 17]. Compared with homogeneous graph neural networks, HGNNs are more appropriate to model the complex interactions in these scenarios by extracting rich semantic information from the graph with multiple relations. Up to now, there have been massive works using HGNN-based models to achieve superior performance of a series of tasks, *e.g.,* node classification [12, 28, 32, 37], linking prediction [19, 24] and recommendation [4, 6, 7, 31]. Although the notable progress of HGNNs has been witnessed, recently it has been found that they are vulnerable to adversarial perturbations [38]. They set a gray-box setting in which they attack a surrogate GCN model to generate perturbations for HGNNs. While in practice attackers usually have rare information about target models, which drives the exploration of black-box adversarial attacks. In general, black-box attacks can be classified into two types according to the mechanism attackers utilize: query-based attack and transfer-based attack [26]. Query-based attacks estimate the gradient of unknown models through queried information. However, the costly query in reality makes this kind of attack hard to implement. By comparison, the transfer-based attack is more practical, which depends on the transferability of adversarial perturbations to achieve an efficient attack. However, currently there's little study on achieving such transfer-based attacks on HGNNs, which threaten the practical utilization of HGNN-based applications such as e-commerce[18, 23] and cyber security[11, 40].

*The corresponding authors.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3615095

To fill this blank, we propose the first work about the transferable adversarial attack of HGNNs, focusing on the representative node classification task (the method can be easily expanded to other tasks, *e.g.,* linking prediction and recommendation). In this work, we concentrate on the structure-based attack, which only allows adding or deleting edges. Aiming to boost the transferability of obtained perturbation, we try to explore some common characteristics shared by different HGNNs. We acquire inspiration from the intuition that different HGNNs take the same graph as input and may rely on some common key message-passing pathways to make the prediction. For example, in the movie classification task, an edge with a famous comedy actor usually forms an important pathway because it contains clear semantics for the movie type, and such edges are expected to be commonly useful for different HGNNs. Based on this we make a reasonable hypothesis that different HGNNs might share similar attention on a subset of edges. We further verify this assumption by analyzing edge attention distribution similarity between different HGNNs, confirming the existence of considerable overlap as shown in Figure 2.

The similarity between different HGNNs on edge attention inspires us to use the model's edge attention to guide the generation of adversarial perturbations. Besides, noting that there's similarity in all types of edges, we propose to conduct the attack on heterogeneous relations to search for the commonly relied edges instead of attacking the single edge type in [38]. Taking ohgbn-imdb dataset as an example, different HGNNs show similarity both in relation movie-actor and movie-director. In this case, perturbing heterogeneous relations is expected to achieve better transferability. Furthermore, to filter the model-specific noise and get reliable gradients, we utilize the integrated gradient of multiple sampled graphs for the final perturbation generation. Finally, we integrate these designs and develop two efficient transferable attack methods for HGNNs, called EA-FGSM and EA-PGD, respectively. Extensive experiments demonstrate the effectiveness of the two proposed methods in boosting adversarial transferability between HGNNs.

In summary, our main contributions are as follows:

- We are the first to systematically evaluate the transferability of adversarial examples for HGNNs. We further conduct an extensive study of the transferability of different perturbation generation methods applied to representative models, and form a transfer robustness benchmark for HGNNs.
- We propose a novel strategy to improve the transferability of adversarial perturbations for HGNNs. We discover the commonality of edge attention distribution between different HGNNs and introduce this characteristic to guide the perturbation generation, which helps mitigate overfitting to the source model.
- We develop two efficient transferable attack generation methods called EA-FGSM and EA-PGD special for HGNNs. Extensive experiments demonstrate the improvement of transferability achieved by our methods.

2 RELATED WORK

Heterogeneous graph neural network. Different from normal GNNs, HGNNs are designed to deal with the heterogeneity of graph data and extract abundant semantics for representation learning. According to the treatment of the graph heterogeneity, HGNNs can be roughly divided into two categories: HGNNs based on one-hop neighbor aggregation which are similar to traditional GNNs and HGNNs based on meta-path neighbor aggregation [39]. HGNNs based on one-hop neighbor aggregation introduce type-specific convolution in the message-passing procedure and the aggregation only considers one-hop neighbors. For example, RGCN [24] deals with graph heterogeneity by using relation-specific weight matrices and aggregating one-hop messages. Another type of HGNNs is based on hand-crafted meta-paths, which describe certain composite relations between nodes. In this kind of HGNNs, the aggregation is implemented in neighbors linked by meta-path. For example, HAN [32] utilizes node-level attention and semantic-level attention to fuse information from different meta-paths.

Adversarial attack on graph neural network. Recently, numerous studies focusing on adversarial attacks [20, 27, 29, 34, 36, 42] and defense [15, 30, 41] on homogeneous graph neural networks were proposed. According to attackers' knowledge, they can be roughly classified into two categories: white-box attacks and black-box attacks. As for white-box attack, Xu *et al.* [36] reformed the Projected Gradient Descent (PGD) algorithm to make it applicable to discrete graph data. In terms of black-box attack, Ma *et al.* [20] proposed a reinforcement learning-based method that achieved attacking only by rewiring edges instead of adding or deleting edges.

Despite the considerable progress made in adversarial attacks on homogeneous graph neural networks (*e.g.* GCN), the adversarial robustness of HGNNs still remains unclear and less explored. Recently, Zhang *et al.* [38] paid attention to the robustness of HGNNs and summarized two causes leading to the weak robustness of HGNNs: perturbation enlargement effect and soft attention mechanism. However, the adversarial perturbation was generated by attacking a surrogate homogeneous GCN model, which overlooked the heterogeneity of the graph data and restricted the generalization of obtained perturbation. Besides, the number of perturbed edges was too large to serve as a practical perturbation in real applications. Aiming to craft more practical adversarial perturbations of HGNNs, in this work we set a more challenging black-box attack scenario and take the first step toward exploring the transfer-based adversarial attack on HGNNs.

3 PRELIMINARIES

3.1 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 $\phi : \mathcal{V} \rightarrow \mathcal{A}$ and an edge type mapping function $\psi : \mathcal{E} \rightarrow \mathcal{R}$. $\mathcal{A}$ and $\mathcal{R}$ denote the predefined sets of node types and edge types, where $|\mathcal{A}| + |\mathcal{R}| > 2$. For each type $r \in \mathcal{R}$, $\boldsymbol{A}_r$ represents the corresponding binary adjacency matrix.

3.2 Heterogeneous Graph Neural Network

HGNNs are proposed to handle the complex structure and rich semantic information in the heterogeneous graph. In terms of the model design, it is flexible to conduct the message passing and aggregation, leading to a large design space [39]. Despite the diversity of model design, all existing HGNNs take the same heterogeneous graphs and features as input and can be summarized and formulated

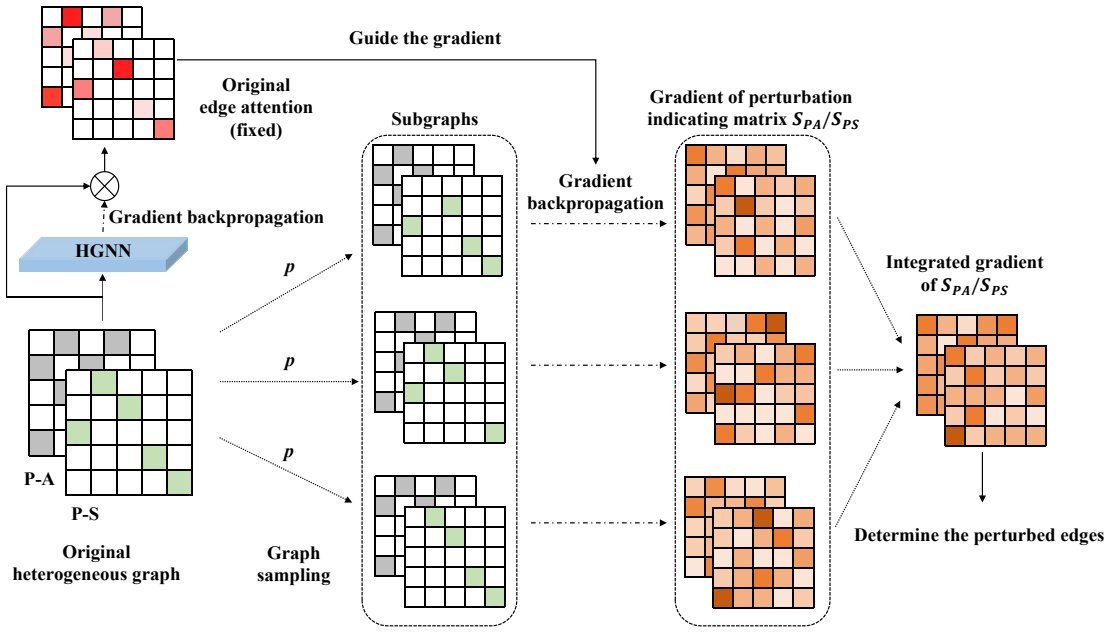

Figure 1: Illustration of the whole attack process on the local source model. The attack is driven by disrupting both the model prediction and the original edge attention of heterogeneous relations. In order to get reliable gradients and weaken model-specific noise, the final gradient for determination is obtained by integrating the gradient of multiple sampled graphs.

as follows:

$$f_{HGNN}(\boldsymbol{A}_{R_1}, \boldsymbol{A}_{R_2}, ..., \boldsymbol{A}_{R_l}; \boldsymbol{X}), \tag{1}$$

where $R_1, R_2, ..., R_l \in \mathcal{R}$ denote l different types of edges in the heterogeneous graph and $\boldsymbol{A}_{R_1}, \boldsymbol{A}_{R_2}, ..., \boldsymbol{A}_{R_l}$ are corresponding adjacency matrix, $\boldsymbol{X}$ represents original node features. Taking ACM citation graph as an example, there are three types of nodes (Author (A), Paper (P), Subject (S)), and two types of edges (P-A and P-S). The input adjacency matrices of HGNNs are $\boldsymbol{A}_{R_1} \in \mathbb{R}^{N_P \times N_A}$ and $\boldsymbol{A}_{R_2} \in \mathbb{R}^{N_P \times N_S}$, where N_P, N_A, N_S denotes the number of papers, authors and subjects, respectively.

3.3 Structure-based adversarial attack on GNNs

Structure-based adversarial attack only allows to add or delete edges from the original graph to mislead the model. Here we introduce a Boolean perturbation indicating matrix $\boldsymbol{S} \in \{0, 1\}^{m \times n}$ with the same size as the original adjacency matrix. The element in $\boldsymbol{S}$ represents whether the corresponding edge is modified(added or removed): $s_{ij} = 1$ indicates the edge (i, j) is modified and $s_{ij} = 0$ means no modification. Given the adjacency matrix $\boldsymbol{A}$, its supplement is calculated by $\overline{\boldsymbol{A}} = \boldsymbol{1} - \boldsymbol{A}$, where $\boldsymbol{1} \in \mathbb{R}^{m \times n}$ represents the matrix whose elements are all one. Then a perturbed graph topology $\boldsymbol{A}'$ against $\boldsymbol{A}$ is given by:

$$\boldsymbol{A}' = \boldsymbol{A} + \boldsymbol{C} \odot \boldsymbol{S}, \boldsymbol{C} = \overline{\boldsymbol{A}} - \boldsymbol{A}, \tag{2}$$

where $\odot$ denotes the element-wise product operation. The positive entry of $\boldsymbol{C}$ denotes the edge that can be added to the graph $\boldsymbol{A}$, and the negative entry denotes the edge that can be removed from $\boldsymbol{A}$. Now we formalize the concept of structure-based attack on GNNs: finding $\boldsymbol{S}$ satisfying pre-defined perturbation constraints (*e.g.* $\leq 5\%$ of the total number of edges) in Eq. (2) to mislead GNNs.

4 TRANSFERABLE STRUCTURE-BASED ATTACK ON HGNNS

Under the setup of transfer attacks, attackers can only access the information of a source model to generate adversarial perturbation. Similar to existing findings in transfer attacks on images [16, 35] which shows limited transferability due to overfitting to the source model, transfer attacks on HGNNs also suffer from this issue.

To tackle this problem, we assume that the key to boosting transferability is to search for the common characteristic of different HGNNs. Following this motivation, we discover that there exists a non-negligible similarity of edge attention distribution between different HGNNs through a comprehensive analysis (Section 4.1). Then we exploit this common characteristic of diverse HGNNs and propose to introduce the model's attention on edges to guide the search of adversarial perturbations on heterogeneous relations (Section 4.2). In order to further enhance the transferability, we conduct random sampling and use the integrated gradient to update the perturbation, which could further suppress the model-specific noise. Based on these designs, we develop two efficient attack methods called EA-FGSM (short for **E**dge **A**ttention-guided **FGSM**) and EA-PGD (short for **E**dge **A**ttention-guided **PGD**), which effectively improve the transferability of the adversarial perturbation (Section 4.3). Figure 1 shows the illustration of the whole attack process.

4.1 Edge Attention Distribution Similarity Analysis

We assume that the key to boost adversarial transferability is to guide the search of perturbations toward the common vulnerable directions of both the source and target models. Considering that

all HGNNs take the same graph as input and might consistently rely on some critical message-passing pathways to make true predictions. Thus it is reasonable to hypothesize that different HGNNs share similar attention on a subset of edges. In order to verify this assumption, we conduct an analysis of edge attention similarity between different HGNNs in this section.

Inspired by the Grad-CAM [25] widely used in assessing the model's attention on different image regions by gradients, we introduce a gradient-based method to measure the model's attention on different edges for HGNNs. Intuitively, gradients could characterize how edge changes will affect the model prediction and serve as the proxy of attention. Formally, for a given HGNN model f_{HGNN}, the attention on edges with relation r is formulated as:

$$\boldsymbol{Attn}(f_{HGNN}; \boldsymbol{A}_r) = g(\frac{\partial \mathcal{L}_{CE}(f_{HGNN}(\boldsymbol{A}_{R_1}, ..., \boldsymbol{A}_{R_l}; \boldsymbol{X}), c)}{\partial \boldsymbol{A}_r}) \odot \boldsymbol{A}_r, \tag{3}$$

where $\odot$ represents the element-wise product, $\mathcal{L}_{CE}$ is the cross-entropy loss used for node classification, c is the true label, and g is the function defined as:

$$g(x) = \begin{cases} 0 & if\ x > 0, \\ x & if\ x \leq 0. \end{cases} \tag{4}$$

In the attention matrix calculated by Eq. (3), the negative value means deleting this edge will lead to the increase of loss value and worsen the model performance. The minimum negative value corresponds to the most influential edge for the overall performance. Here we only reserve the negative value so that we can focus on attacking those useful edges. Based on this measurement, we analyze the cosine edge attention similarity (1 means identical edge attention and 0 means totally different edge attention) between six typical HGNNs (HAN, RGCN, GTN, SimpleHGN, HGT and MHNF) for different relations of two datasets. The result is shown in Figure 2 and we summarize the findings as follows:

- **Considerable similarity of edge attention distribution between diverse HGNNs.** From the visualization result it can be found that HGNNs indeed share some commonality on edge attention distribution, for example, in ohgbn-acm dataset, HGNNs show notable similarity in terms of paper-author relation and the highest similarity reaches 0.67. Besides, we find that meta-path-based models (HAN, GTN, MHNF) show consistently high similarity between each other, probably due to the similar semantic information encoded by the used meta-paths.
- **The edges with similar attention are heterogeneous.** The results also indicate that in each relation there are edges with common attention. This finding also implies that only perturbing edges with a fixed relation is inadequate for transfer attacks. It is expected to boost the adversarial transferability by introducing heterogeneous perturbations.

4.2 Edge Attention-guided Attack Loss

Given that different HGNNs share similar edge attention, perturbations on the edge attention distribution may effectively transfer to other models. Therefore, together with dropping the overall performance, we expect the perturbation could also disrupt the edge attention distribution. We achieve this goal by adding a loss term L_{attn} to guide the perturbation generation, for the relation $r \in \mathcal{R}$, the loss term is formulated as follows:

$$\mathcal{L}^r_{attn} = \|\boldsymbol{Attn}(f_{HGNN}; \boldsymbol{A}_r) - \boldsymbol{Attn}(f_{HGNN}; \boldsymbol{A}'_r)\|_2, \tag{5}$$

where $\boldsymbol{A}_r$ is the original input adjacency matrix, and $\boldsymbol{A}'_r$ is perturbed matrix. The final loss $\mathcal{L}$ combines the attention disruption loss L_{attn} and a CW-type loss similar to Carlini-Wagner (CW) attacks for misleading image classifiers [2]:

$$\mathcal{L}_{CW} = \sum_{i \in \mathcal{V}} \max\{Z_{i,c} - \max_{y_i \neq c} Z_{i,y_i}, -\kappa\}, \tag{6}$$

$$\mathcal{L} = \mathcal{L}_{CW} - \lambda \sum_{r \in \mathcal{R}} \mathcal{L}^r_{attn}, \tag{7}$$

where $Z_{i,c}$ denotes the probability of assigning node i to class c, κ (set as 0 here) is a confidential level of making wrong predictions, λ controls the ratio of the two loss terms. The objective of the attack is to minimize the attack loss $\mathcal{L}$, where the first term will mislead the final decision of the model and the second term will destroy the attention distribution on critical edges.

4.3 Edge Attention-guided Perturbation Generation

Similar to what's mentioned in Section 3.3, here we introduce the perturbation indicating matrix of different relations $\boldsymbol{S}_{R_1}, \boldsymbol{S}_{R_2}, ..., \boldsymbol{S}_{R_l}$ into the model input, and we reformulate HGNNs as follows during the attack process:

$$f_{HGNN}(\boldsymbol{A}_{R_1}, \boldsymbol{A}_{R_2}, ..., \boldsymbol{A}_{R_l}; \boldsymbol{S}_{R_1}, \boldsymbol{S}_{R_2}, ..., \boldsymbol{S}_{R_l}; \boldsymbol{X}). \tag{8}$$

Given the proposed attack loss in Eq. (7), we now formulate the perturbation generation as the following optimization problem:

$$\underset{\boldsymbol{S}_{R_1}, \boldsymbol{S}_{R_2}, ..., \boldsymbol{S}_{R_l}}{minimize}\ \mathcal{L}(\boldsymbol{A}_{R_1}, \boldsymbol{A}_{R_2}, ..., \boldsymbol{A}_{R_l}; \boldsymbol{S}_{R_1}, \boldsymbol{S}_{R_2}, ..., \boldsymbol{S}_{R_l}; \boldsymbol{X}), \tag{9}$$

where $\|\boldsymbol{S}_{R_1}\|_1 + \|\boldsymbol{S}_{R_2}\|_1 + ... + \|\boldsymbol{S}_{R_l}\|_1 \leq \Delta$.

The above optimization is actually a combinatorial optimization problem because $\boldsymbol{S}$ is a Boolean matrix where the elements are restricted in $\{0, 1\}$. In order to achieve the objective in Eq. (9), here we develop two perturbation generation methods specially for attacking HGNNs, called EA-FGSM and EA-PGD, respectively.

4.3.1 Perturbation Generation through EA-FGSM. Fast Gradient Sign Method (FGSM) [8] attacks the model by conducting gradient update along the direction of the sign of gradients of loss function w.r.t pixels of images, which takes a single step to determine the perturbation direction. The similar idea has been borrowed in attacking GNNs. For example, in order to attack HGNNs, Zhang *et al.* [38] attacked a homogeneous surrogated model (GCN) by changing the edge with the largest gradient of the loss function in each iteration. However, they conduct the attack for every node and cause a massive magnitude of perturbations, which does not meet the unnoticeable requirement of adversarial examples. What's more, their attack limits the attack on single relation of the graph resulting in poor transferability. We correct the attack setup and design an improved edge attention-guided attack method towards heterogeneous relations, which is called EA-FGSM.

The main idea of EA-FGSM is to change the edge with the most significant effect on the loss function for all relations in each iteration. Different from existing FGSM-based attack [38], we take an integrated gradient strategy to enhance the attack. Specifically, in

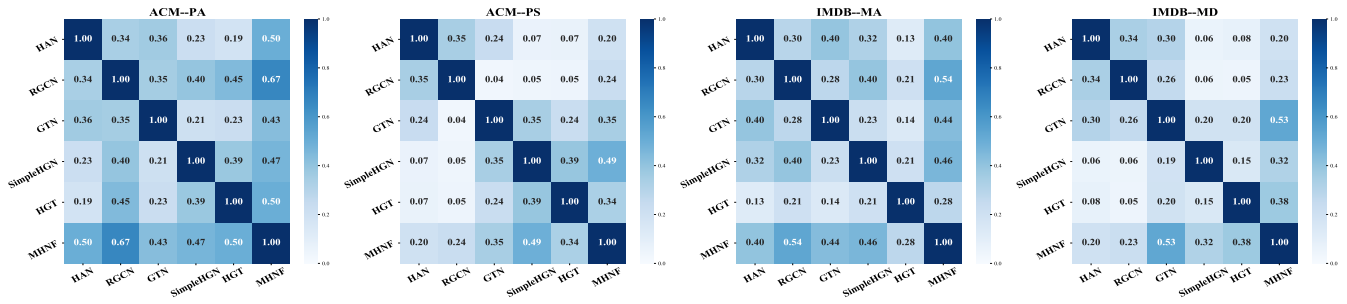

Figure 2: Similarity of edge attention patterns for different types of relations between different HGNNs on ohgbn-acm(relation: paper-author(PA), paper-subject(PS)) and ohgbn-imdb(relation: movie-actor(MA), movie-director(MD)).

each iteration, we first conduct a graph sampling for each relation r, which is formulated as:

$$\widetilde{A_r} = A_r \odot M_r^k, M_r^k \sim Bernoulli(p), \tag{10}$$

where M_r^k is the k_{th} binary mask matrix with the same size as A_r, p controls the sampling rate. Then we calculate the gradient of the loss function in Eq. (7) w.r.t. the perturbation indicating matrix $\widetilde{S_r}$:

$$G^{S_r} = \frac{\partial \mathcal{L}(\widetilde{A_{R_1}}, \widetilde{A_{R_2}}, ..., \widetilde{A_{R_l}}; S_{R_1}, S_{R_2}, ..., S_{R_l}; X)}{\partial S_r}. \tag{11}$$

Considering that different HGNNs will pay model-specific attention on some edges to better fit themselves to the data domain. In order to get reliable gradients and weaken the aforementioned model-specific noise, we aggregate the gradients of K sampled graphs and perturb the edge $e^*_{r_m}$ with the minimum gradient (the negative gradient with maximum absolute value) in all relations, where r_m is the edge type of $e^*_{r_m}$. After repeating the above process for Δ times, we can obtain the perturbed adjacency matrix of each relation. The pseudo-code of EA-FGSM is shown in Algorithm 1.

4.3.2 Perturbation Generation through EA-PGD. Different from FGSM method, projected gradient descent (PGD) [21] has been proven to be a more powerful attack. As for the attack on GNNs, Xu *et al.* [36] proposed a PGD-based attack achieving state-of-the-art attack performance on homogeneous GNNs. To make it fit for attacking HGNNs and boosting adversarial transferability, we refine it by introducing heterogeneous relation-oriented and edge attention-guided attacks. Different from EA-FGSM, EA-PGD generates the perturbation indicating matrix S by iterative updating instead of picking the perturbed edge one by one discretely.

In order to solve the optimization problem in Eq. (9), here we relax the elements in S from discrete set $\{0, 1\}$ to continuous range $[0, 1]$. However, when implementing the method we find there's a great gap between the attack performance of continuous S and discrete form. To alleviate this problem we add another loss term to guide the value in S to move towards 0 or 1. Intuitively, the gap will be significant if the obtained S has many elements near 0.5 because it is not satisfying whether it is given 0 or 1. In this situation, making the value far away from 0.5 could help narrow the gap, and the attack loss finally used for EA-PGD is:

$$\mathcal{L} = L_{CW} - \lambda L_{attn} - \|S - 0.5 * \mathbf{1}\|_2, \tag{12}$$

where $\mathbf{1}$ is the matrix whose elements are all one. After obtaining the integrated gradient G_r of the above loss function, the continuous perturbation indicating matrix can be obtained by iteratively

Algorithm 1 Perturbation generation using EA-FGSM

Input: Trained model $f_{HGNN}(A_{R_1}, A_{R_2}, ..., A_{R_l}; S_{R_1}, S_{R_2}, ..., S_{R_l}; X)$, original adjacency matrix of different relations $A_{R_1}, A_{R_2}, ..., A_{R_l}$, the budge of perturbations Δ, perturbation indicating matrix $S_{R_1}, S_{R_2}, ..., S_{R_l}$, sampling number K
Output: Modified adjacency matrix of different relations $A'_{R_1}, A'_{R_2}, ..., A'_{R_l}$
$N_{change} = 0$
while $N_{change} \leq \Delta$ **do**
  Reset $S_{R_1}, S_{R_2}, ..., S_{R_l}$ all to zero-matrix
  Initial: $G_{R_1}, G_{R_2}, ..., G_{R_l} = 0$
  for k=1 to K **do**
    for each relation r in $\mathcal{R}$ **do**
      $\widetilde{A_r} = A_r \odot M_r^k$ // Graph sampling for each relation
    end for
    for each relation r in $\mathcal{R}$ **do**
      $G_r = G_r + G^{S_r}$ // Integrate gradient
    end for
  end for
  Select the edge $e^*_{r_m}$ of type r_m with minimum value in $\{G_{R_1}, G_{R_2}, ..., G_{R_l}\}$
  $A_{r_m} = A_{r_m} + e^*_{r_m}$ // Perturb the edge e^*_{rm}
  $N_{change} = N_{change} + 1$
end while
return $A_{R_1}, A_{R_2}, ..., A_{R_l}$

projected gradient descent:

$$S_r^{(t)} = \Pi[S_r^{(t-1)} - \eta_t G_r], \tag{13}$$

where η_t is learning rate, Π denotes the projection operation forcing the perturbed edge number no more than the predefined budge $\Delta_{R_1}, \Delta_{R_2}, ..., \Delta_{R_l}$, the closed-form solution has been given in [36]:

$$\Pi[S] = \begin{cases} P_{[0,1]}[S - \mu\mathbf{1}] & if \mu > 0 \ and \ \|P_{[0,1]}[S - \mu\mathbf{1}]\|_1 = \epsilon, \\ P_{[0,1]}[S] & if \ \|P_{[0,1]}[S]\|_1 \leq \epsilon. \end{cases} \tag{14}$$

where $P_{[0,1]}[x]$ denotes the operation clamping the value in x to $[0, 1]$, μ could be solved by the bisection method [1]. After getting the continuous form of S, the next goal is to recover a binary solution from it. Since the element in S can be interpreted as a probability, the binary element s^*_{ij} can be determined by a Bernoulli

Algorithm 2 Perturbation generation using EA-PGD

Input: Trained model $f_{HGNN}(\boldsymbol{A}_{R_1}, \boldsymbol{A}_{R_2}, ..., \boldsymbol{A}_{R_l}; \boldsymbol{S}_{R_1}, \boldsymbol{S}_{R_2}, ..., \boldsymbol{S}_{R_l}; \boldsymbol{X})$, original adjacency matrix of different types $\boldsymbol{A}_{R_1}, \boldsymbol{A}_{R_2}, ..., \boldsymbol{A}_{R_l}$, the maximum number of perturbations $\Delta_{R_1}, \Delta_{R_2}, ..., \Delta_{R_l}$, sampling number K, perturb indicating matrix $\boldsymbol{S}^{(0)}_{R_1}, \boldsymbol{S}^{(0)}_{R_2}, ..., \boldsymbol{S}^{(0)}_{R_l}$, learning rate η_t and iterations T
Output: Modified adjacency matrix of different types $\boldsymbol{A}'_{R_1}, \boldsymbol{A}'_{R_2}, ..., \boldsymbol{A}'_{R_l}$
for $t = 1, 2, ..., T$ **do**
  Initial: $\boldsymbol{G}_{R_1}, \boldsymbol{G}_{R_2}, ..., \boldsymbol{G}_{R_l} = 0$
  for k=1 to K **do**
    for each relation r in $\mathcal{R}$ **do**
      $\widetilde{\boldsymbol{A}_r} = \boldsymbol{A}_r \odot \boldsymbol{M}_r^k$ // Graph sampling for each relation
    end for
    for each relation r in $\mathcal{R}$ **do**
      $\boldsymbol{G}_r = \boldsymbol{G}_r + \boldsymbol{G}^{S_r}$ // Integrate gradient
    end for
  end for
  for each relation r in $\mathcal{R}$ **do**
    $\boldsymbol{G}_r = \frac{\boldsymbol{G}_r}{\|\boldsymbol{G}_r\|}$
    $\boldsymbol{S}_r^{(t)} = \Pi[\boldsymbol{S}_r^{(t-1)} - \eta_t \boldsymbol{G}_r]$
  end for
end for
Conduct Bernoulli sampling on $\boldsymbol{S}^{(t)}_{R_1}, \boldsymbol{S}^{(t)}_{R_2}, ..., \boldsymbol{S}^{(t)}_{R_l}$ to get binary matrix $\boldsymbol{S}^*_{R_1}, \boldsymbol{S}^*_{R_2}, ..., \boldsymbol{S}^*_{R_l}$
Calculate $\boldsymbol{A}'_{R_1}, \boldsymbol{A}'_{R_2}, ..., \boldsymbol{A}'_{R_l}$ according to Eq. (2).
return $\boldsymbol{A}'_{R_1}, \boldsymbol{A}'_{R_2}, ..., \boldsymbol{A}'_{R_l}$

sampling:

$$s^*_{ij} = \begin{cases} 1 & with\ probability\ s_{ij}, \\ 0 & with\ probability\ 1 - s_{ij}. \end{cases} \tag{15}$$

Finally, the perturbed adjacency matrix of each relation could be calculated according to Eq. (2). The pseudo-code of EA-PGD is shown in Algorithm 2.

4.4 Time Complexity Analysis

In this section, we analyze the time complexity of our proposed EA-FGSM and EA-PGD. Our method conducts an extra calculation of the attention matrix in the loss function, while the introduced computational time can be overlooked because it just needs to take the backpropagation once to get the attention matrix in each step. EA-FGSM needs K steps to get integrated gradients for l relations repeating Δ times and the complexity is $O(lK\Delta)$. In the experiment we control Δ as only 5% of the total number of edges and K as 5 to reduce the required computation. By comparison, EA-PGD only iterates T (set as 200) steps to generate all perturbed edges, which is more efficient with complexity $O(lKT)$.

5 EXPERIMENTS

5.1 Experimental Settings

Datasets. We evaluate the transfer attack performance on two benchmark datasets of heterogeneous graph for node classification task [9]: (1) **ohgbn-acm** for paper classification including 3 node types: paper (P), author (A) and subject (S). (2)**ohgbn-imdb** for movie classification including 3 node types: movie (M), actor (A) and director (D). The detailed information is shown in Table 1.

Table 1: Statistics of the dataset.

Dataset	Relations(A-B)	# of A	# of B	# of A-B	Train	Val	Test
ohgbn-acm	Paper-Author	3025	5912	9936	600	300	2125
	Paper-Subject	3025	57	3025			
ohgbn-imdb	Movie-Actor	4661	5841	13983	300	300	2339
	Movie-Director	4661	2270	4661			

HGNNs. We evaluate the transfer attack performance on six widely-used models: HAN [32], RGCN [24], GTN [37], SimpleHGN [19], HGT [12] and MHNF [28]. We concentrate on evasion attack which conducts attack after model training and carefully tune the hyper-parameters and choose the best model as the attack target.

Attack methods. Since there's no direct attack method specially for HGNNs, here we form two kinds of baselines for comparison:

- **Attack on homogeneous relation.** We implement three methods of attacking the single type of edges of HGNNs. For ohgbn-acm we choose to attack the relation paper-author and attack movie-actor relation for ohgbn-imdb. DICE [33] is implemented by adding or deleting edges randomly. FGSM is the attack method used in [38] for HGNNs, which attacks one specified edge type by picking the edge with maximum absolute gradient one by one. For a fair comparison, we refine the method to directly attack HGNNs instead of the surrogate GCN model. PGD [36] uses the projected gradient descent method to learn the perturbation indicating matrix and samples to get the discrete perturbed graph.
- **Attack on heterogeneous relations.** For a fair comparison, we have refined FGSM and PGD to attack multiple types of edges. Specifically, the difference between EA-FGSM and EA-PGD is there's no edge attention guidance and integrated gradient.

Parameter settings. In the experiments, our attack goal is to decrease the overall performance of all test nodes, the budge of the perturbation Δ is set as 5% of the total number of all types of edges, which is widely used in the attack for GNNs [13, 36]. The sampling number K for integrating gradient is 5 and the sampling rate p is 0.9 to avoid the dramatic change of the original graph structure. The coefficient λ in the loss function is 10. For EA-PGD, we have tried a lot of settings for perturbation budge of different relations $\Delta_{R_1}, \Delta_{R_2}, ..., \Delta_{R_l}$ and choose $\Delta_{P-A}/\Delta_{P-S} = 4/1$ for ohgbn-acm and $\Delta_{M-A}/\Delta_{M-D} = 1/1$ for ohgbn-imdb with the best attack performance. The learning rate $\eta_t = 1/\sqrt{t}$ and iterations $T = 200$.

5.2 Comparison of Transferability

The performance of transfer attack on the two datasets is shown in Table 2 and Table 3. We conduct white-box attack on six models in turn and take the obtained perturbed graph as input of the other five models to test the adversarial transferability. Here we report Micro-F1 as the metric of HGNN performance for the node classification task. From the result we have the following observations:

- **Our proposed methods consistently achieve better adversarial transferability in all experiments.** Statistically, our method (the better one in EA-FGSM and EA-PGD) gets 70.87% improvement on average compared with the best homogeneous

Table 2: Results (Micro-F1) of transfer attacks on ohgbn-acm dataset between six models. The first column shows source models and the first row lists target models. The attack methods are classified according to perturbing homogeneous or heterogeneous edges. The best results are highlighted in bold. () indicates white-box attack where the target model is the source model, not belonging to transfer attack.

		Attack	HAN	RGCN	GTN	SimpleHGN	HGT	MHNF
	Attack type	No perturbation	0.9035	0.9219	0.9092	0.9144	0.9021	0.9144
HAN	Homogeneous	DICE	(0.8416)	0.8826	0.8831	0.8912	0.8947	0.8925
		PGD	(0.8326)	0.8769	0.8823	0.8920	0.8914	0.8895
		FGSM	(0.8574)	0.8794	0.8859	0.8953	0.8913	0.8944
	Heterogeneous	PGD	**(0.7860)**	0.8682	0.8826	0.8901	0.8905	0.8813
		FGSM	(0.7980)	0.8691	0.8771	0.8884	0.8866	0.8767
		EA-PGD	(0.7871)	0.8575	0.8753	0.8874	0.8873	0.8749
		EA-FGSM	(0.7949)	**0.8517**	**0.8721**	**0.8737**	**0.8724**	**0.8701**
RGCN	Homogeneous	DICE	0.8546	(0.8435)	0.8937	0.8348	0.8711	0.8326
		PGD	0.8521	(0.8448)	0.8952	0.8367	0.8684	0.8287
		FGSM	0.8467	(0.8455)	0.8942	0.8376	0.8744	0.8412
	Heterogeneous	PGD	0.8245	**(0.7748)**	0.8932	0.8267	0.8684	0.8287
		FGSM	0.8273	(0.7921)	0.8889	0.8123	0.8663	0.8264
		EA-PGD	0.8028	(0.7813)	0.8826	0.8032	0.8536	0.8169
		EA-FGSM	**0.7929**	(0.7975)	**0.8776**	**0.7741**	**0.8451**	**0.8112**
GTN	Homogeneous	DICE	0.8433	0.8768	(0.8019)	0.8820	0.8927	0.8814
		PGD	0.8425	0.8749	(0.7837)	0.8802	0.8935	0.8802
		FGSM	0.8301	0.8810	(0.8015)	0.8838	0.8952	0.8779
	Heterogeneous	PGD	0.8168	0.8765	**(0.7601)**	0.8795	0.8910	0.8651
		FGSM	0.8082	0.8701	(0.7675)	0.8739	0.8908	0.8668
		EA-PGD	0.8012	0.8679	(0.7687)	0.8703	0.8843	0.8543
		EA-FGSM	**0.7816**	**0.8664**	(0.7618)	**0.8607**	**0.8823**	**0.8498**
SimpleHGN	Homogeneous	DICE	0.8469	0.8530	0.8556	(0.8329)	0.8502	0.8450
		PGD	0.8479	0.8528	0.8579	(0.8273)	0.8480	0.8437
		FGSM	0.8487	0.8403	0.8504	(0.8394)	0.8417	0.8435
	Heterogeneous	PGD	0.8547	0.8389	0.8478	**(0.8017)**	0.8406	0.8380
		FGSM	0.8346	0.8359	0.8464	(0.8145)	0.8415	0.8362
		EA-PGD	0.8480	0.8293	0.8402	(0.8076)	0.8314	0.8329
		EA-FGSM	**0.8296**	**0.8169**	**0.8324**	(0.8201)	**0.8202**	**0.8202**
HGT	Homogeneous	DICE	0.8815	0.8769	0.8926	0.8937	(0.8440)	0.8867
		PGD	0.8810	0.8798	0.8925	0.8945	(0.8438)	0.8846
		FGSM	0.8839	0.8766	0.8939	0.8940	(0.8479)	0.8858
	Heterogeneous	PGD	0.8801	0.8715	0.8826	0.8803	(0.8287)	0.8775
		FGSM	0.8786	0.8659	0.8838	0.8809	(0.8273)	0.8729
		EA-PGD	0.8684	0.8651	0.8791	0.8669	(0.8252)	0.8658
		EA-FGSM	**0.8677**	**0.8587**	**0.8771**	**0.8640**	**(0.8245)**	**0.8607**
MHNF	Homogeneous	DICE	0.8498	0.8945	0.8627	0.8820	0.8969	(0.8572)
		PGD	0.8456	0.8935	0.8614	0.8748	0.8974	(0.8438)
		FGSM	0.8408	0.8874	0.8654	0.8775	0.8953	(0.8591)
	Heterogeneous	PGD	0.8321	0.8746	0.8422	0.8560	0.8912	(0.8277)
		FGSM	0.8277	0.8757	0.8508	0.8682	0.8889	**(0.8224)**
		EA-PGD	0.8258	0.8729	0.8318	0.8457	0.8869	(0.8322)
		EA-FGSM	**0.8225**	**0.8700**	**0.8277**	**0.8390**	**0.8831**	(0.8525)

attack methods on ohgbn-acm and 50.85% on ohgbn-imdb in terms of drop of Micro-F1. Compared with the best heterogeneous attack methods, we get 32.21% improvement on average for ohgbn-acm and 14.86% for ohgbn-imdb. The overall result demonstrates the effectiveness of our proposed methods.

- **The performance of transfer attack shows notable correlation with edge attention similarity.** Intuitively, the perturbation is more likely to transfer when the source model and target model share many similarities. Here the edge attention similarities between HGNNs is basically in line with this rule. For example, the perturbation obtained by attacking RGCN and SimpleHGN shows better transferability on attacking MHNF for ohgbn-acm dataset, which matches the similarity analysis shown in Figure 2.
- **EA-FGSM exhibits better performance than EA-PGD in general.** From the results of transfer attack, EA-FGSM outperforms EA-PGD in most cases. We assume that the crux might still be the gap between continuous and discrete forms of perturbation in EA-PGD. According to our observation, the continuous perturbation indicating matrix S during attack generation shows

Table 3: Results (Micro-F1) of transfer attacks between six models on ohgbn-imdb dataset. The first column shows source models, and the first row lists target models. The attack methods are classified into two types according to perturbing homogeneous or heterogeneous edges. The best results are highlighted in bold. () indicates white-box attack where the target model is the source model, not belonging to transfer attack.

		Attack	HAN	RGCN	GTN	SimpleHGN	HGT	MHNF
	Attack type	No perturbation	0.6084	0.6032	0.5950	0.6109	0.5767	0.6191
HAN	Homogeneous	DICE	(0.5114)	0.5546	0.5515	0.5984	0.5560	0.5894
		PGD	(0.5079)	0.5504	0.5498	0.5956	0.5512	0.5879
		FGSM	(0.5109)	0.5489	0.5455	0.5904	0.5485	0.5840
	Heterogeneous	PGD	(0.5018)	0.5480	0.5220	0.5848	0.5476	0.5583
		FGSM	(0.4699)	0.5412	0.5277	0.5852	0.5476	0.5601
		EA-PGD	(0.4938)	*0.5429*	0.5198	0.5823	0.5459	0.5528
		EA-FGSM	**(0.4666)**	**0.5352**	**0.5118**	**0.5809**	**0.5428**	**0.5515**
RGCN	Homogeneous	DICE	0.5695	(0.5045)	0.5560	0.5512	0.5421	0.5522
		PGD	0.5678	(0.5027)	0.5539	0.5478	0.5396	0.5475
		FGSM	0.5669	(0.5070)	0.5459	0.5426	0.5314	0.5476
	Heterogeneous	PGD	0.5581	(0.4859)	0.5361	0.5570	0.5228	0.5412
		FGSM	0.5369	**(0.4528)**	0.5288	0.5407	0.5221	0.5337
		EA-PGD	0.5503	(0.5002)	0.5302	0.5438	0.5204	0.5366
		EA-FGSM	**0.5316**	(0.4776)	**0.5268**	**0.5201**	**0.5133**	**0.5231**
GTN	Homogeneous	DICE	0.5628	0.5729	(0.4314)	0.5856	0.5513	0.5176
		PGD	0.5616	0.5702	(0.4208)	0.5870	0.5516	0.5137
		FGSM	0.5579	0.5673	(0.4288)	0.5828	0.5485	0.5042
	Heterogeneous	PGD	0.5398	0.5653	(0.4204)	0.5910	0.5408	0.5054
		FGSM	0.5352	0.5622	(0.4215)	0.5891	0.5416	0.5027
		EA-PGD	0.5315	0.5631	**(0.4189)**	0.5868	0.5389	0.4976
		EA-FGSM	**0.5254**	**0.5526**	(0.4190)	**0.5735**	**0.5352**	**0.4925**
SimpleHGN	Homogeneous	DICE	0.5602	0.5746	0.5581	(0.5217)	0.5446	0.5725
		PGD	0.5576	0.5721	0.5517	(0.5037)	0.5410	0.5701
		FGSM	0.5498	0.5687	0.5455	(0.5019)	0.5348	0.5639
	Heterogeneous	PGD	0.5508	0.5689	0.5401	(0.4921)	0.5198	0.5486
		FGSM	0.5468	**0.5562**	0.5382	**(0.4382)**	0.5185	0.5445
		EA-PGD	0.5465	0.5587	0.5368	(0.4844)	0.5156	0.5455
		EA-FGSM	**0.5425**	0.5584	**0.5332**	(0.4784)	**0.5109**	**0.5408**
HGT	Homogeneous	DICE	0.5585	0.5894	0.5406	0.5619	(0.4712)	0.5539
		PGD	0.5574	0.5856	0.5394	0.5568	(0.4875)	0.5502
		FGSM	0.5528	0.5867	0.5370	0.5434	(0.5122)	0.5472
	Heterogeneous	PGD	0.5440	0.5876	0.5387	0.5612	(0.3963)	0.5368
		FGSM	0.5496	0.5844	0.5301	0.5630	**(0.3386)**	0.5349
		EA-PGD	0.5405	0.5703	0.5276	0.5445	(0.3882)	0.5277
		EA-FGSM	**0.5396**	**0.5681**	**0.5228**	**0.5382**	(0.4104)	**0.5224**
MHNF	Homogeneous	DICE	0.5821	0.5795	0.5123	0.5932	0.5598	(0.4972)
		PGD	0.5812	0.5800	0.5079	0.5879	0.5576	(0.4827)
		FGSM	0.5759	0.5720	0.5051	0.6000	0.5528	(0.4629)
	Heterogeneous	PGD	0.5687	0.5465	0.5139	0.5519	0.5456	(0.4587)
		FGSM	0.5626	0.5386	0.5002	0.5553	0.5369	(0.4291)
		EA-PGD	0.5644	0.5384	0.5036	**0.5467**	0.5388	**(0.4012)**
		EA-FGSM	**0.5546**	**0.5295**	**0.4947**	0.5476	**0.5291**	(0.4340)

powerful attack ability. However, the final perturbation is the discrete form of it through probabilistic sampling, whose attack performance drops dramatically. We mitigate this problem by adding an external term in the loss function while more treatment might be needed to further improve EA-PGD.

5.3 Ablation Study

In our method, there are three key designs to promote adversarial transferability: heterogeneous-relation attack, edge attention guidance and integrated gradient. We assess their contribution respectively through ablation studies focusing on EA-FGSM which outperforms other methods. We compare the complete method of EA-FGSM with the versions without the aforementioned components and show the results in Table 4. Here we report the results of transferring the perturbation by attacking RGCN to other models as examples. From the result it can be seen that the full method achieves the best performance, indicating that the three key designs of our method collaborate well to boost the transferability.

Table 4: Ablation study on transfer attack with RGCN as the source model on two datasets.

Dataset	Attack method	HAN	GTN	SimpleHGN	HGT	MHNF
ohgbn-acm	EA-FGSM	**0.7929**	**0.8776**	**0.7741**	**0.8451**	**0.8112**
	w/o het-relation attack	0.8368	0.8936	0.8295	0.8715	0.8375
	w/o edge attention guidance	0.8120	0.8841	0.8006	0.8565	0.8216
	w/o integrated gradient	0.8037	0.8802	0.7908	0.8577	0.8204
ohgbn-imdb	EA-FGSM	**0.5316**	**0.5268**	**0.5201**	**0.5133**	**0.5231**
	w/o het-relation attack	0.5568	0.5404	0.5389	0.5278	0.5456
	w/o edge attention guidance	0.5359	0.5286	0.5363	0.5216	0.5318
	w/o integrated gradient	0.5378	0.5358	0.5335	0.5212	0.5357

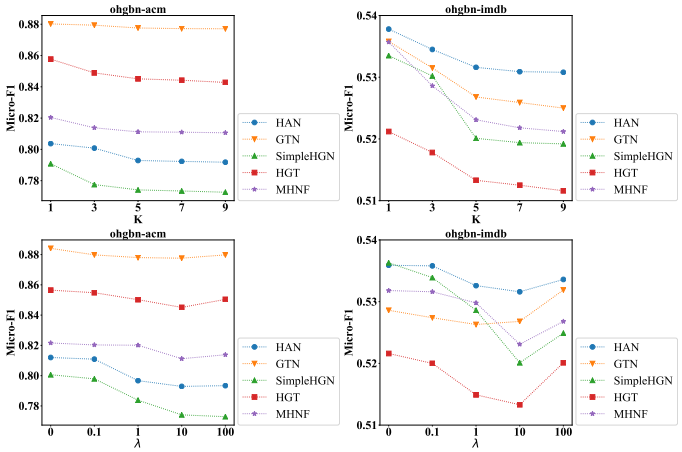

Figure 3: Study of the effect of K and λ on transfer attack performance with RGCN as the source model on two datasets.

5.4 Hyper-parameter Study

Here we explore the effect of two important hyper-parameters in our method: the number of integrated gradients K and the coefficient λ controlling the ratio of loss terms. We focus on EA-FGSM and use RGCN as the source model. The results are shown in Figure 3. We test the attack performance with $K \in [1, 3, 5, 7, 9]$. It can be observed that overall the transferability increases with more integrated gradients due to the elimination of model-specific noise and focus more on perturbing the edges with common attention. Noting that there's little attack performance gain after $K = 5$, we choose $K = 5$ as the final setting. As for λ, we test the attack performance with $\lambda \in [0, 0.1, 1, 10, 100]$. Overall the transfer attack performance reaches the peak when λ is 10.

5.5 Case Study

For a more intuitive understanding, in this section we visualize a real case from ohgbn-imdb dataset as shown in Figure 4 to explain why our method achieves superior adversarial transferability. Here we show the sub-graph around the movie-type node M_{11} and its neighbor nodes including three actor-type nodes $A_{11}, A_{1037}, A_{1947}$ and one director-type node D_{11}. The number on edges is the attention value calculated as Eq. (3). The negative value with a larger absolute value means more significant influence on the model prediction. The case shows the perturbation generated by attacking HGT and the transfer result on HAN, RGCN and SimpleHGN through vanilla FGSM and EA-FGSM. It can be seen that vanilla FGSM only concentrates on disrupting the edge with the largest influence on

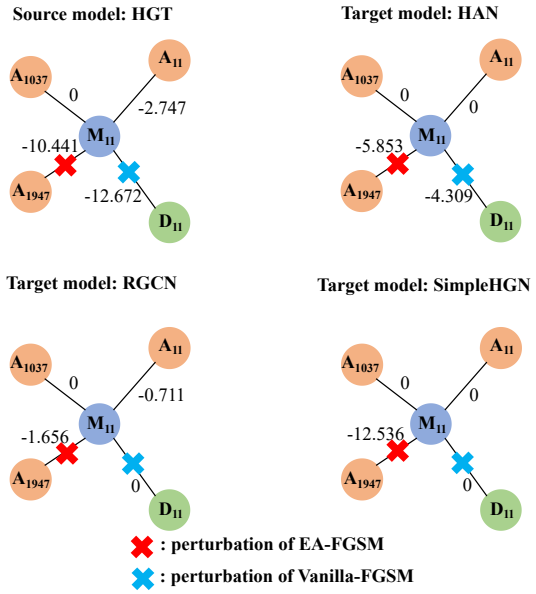

Figure 4: A real example from ohgbn-imdb dataset to illustrate the effectiveness of our method to boost adversarial transferability.

the performance of source model so it determines to delete the edge linking to D_{11}. However, this leads to overfitting of the source model and fail to attack the target models. By comparison, EA-FGSM searches the edge which additionally considers the attack on the edge attention distribution. As a result, it chooses to perturb the edge linking to A_{1947}, which is validated to be more commonly vulnerable. The case study verifies that our method indeed prioritizes perturbing the common vulnerable edges, verifying the effectiveness to introduce edge attention to guide the perturbation search.

6 CONCLUSION

In this work we first shed light on the transferability of adversarial perturbation on HGNNs, which is challenging due to the overfitting of the source model. To address this problem, we propose to implement the attack with the guidance of edge attention, driving the resultant perturbations toward common vulnerable directions. Following this we finally form two efficient structure-based attack methods: EA-FGSM and EA-PGD. Numerous experiments demonstrate the superiority of our methods on adversarial transferability. This work provides a novel benchmark for transfer robustness of HGNNs. In the future, we will consider attacks on models with defense and other tasks, *e.g.*, HGNN-based recommendation.

ACKNOWLEDGMENTS

This work is supported in part by the National Key Research and Development Program of China (No. 2020YFA0711403), and the National Natural Science Foundation of China (No. U22B2057, No. U21B2036, No. U20B2060, No. U20B2062).

REFERENCES

[1] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

[2] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*. Ieee, 39–57.

[3] Shaohua Fan, Junxiong Zhu, Xiaotian Han, Chuan Shi, Linmei Hu, Biyu Ma, and Yongliang Li. 2019. Metapath-guided heterogeneous graph neural network for intent recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2478–2486.

[4] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *The world wide web conference*. 417–426.

[5] Yujie Fan, Mingxuan Ju, Shifu Hou, Yanfang Ye, Wenqiang Wan, Kui Wang, Yinming Mei, and Qi Xiong. 2021. Heterogeneous temporal graph transformer: An intelligent system for evolving android malware detection. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2831–2839.

[6] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. 2023. A survey of graph neural networks for recommender systems: challenges, methods, and directions. *ACM Transactions on Recommender Systems* 1, 1 (2023), 1–51.

[7] Chen Gao, Yu Zheng, Wenjie Wang, Fuli Feng, Xiangnan He, and Yong Li. 2022. Causal Inference in Recommender Systems: A Survey and Future Directions. *arXiv preprint arXiv:2208.12397* (2022).

[8] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[9] Hui Han, Tianyu Zhao, Cheng Yang, Hongyi Zhang, Yaoqi Liu, Xiao Wang, and Chuan Shi. 2022. OpenHGNN: An Open Source Toolkit for Heterogeneous Graph Neural Network. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3993–3997.

[10] Shifu Hou, Yanfang Ye, Yangqiu Song, and Melih Abdulhayoglu. 2017. Hindroid: An intelligent android malware detection system based on structured heterogeneous information network. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 1507–1515.

[11] Binbin Hu, Zhiqiang Zhang, Chuan Shi, Jun Zhou, Xiaolong Li, and Yuan Qi. 2019. Cash-out user detection based on attributed heterogeneous information network with a hierarchical attention mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 946–953.

[12] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020*. 2704–2710.

[13] Hussain Hussain, Tomislav Duricic, Elisabeth Lex, Denis Helic, Markus Strohmaier, and Roman Kern. 2021. Structack: Structure-based adversarial attacks on graph neural networks. *arXiv preprint arXiv:2107.11327* (2021).

[14] Jiarui Jin, Jiarui Qin, Yuchen Fang, Kounianhua Du, Weinan Zhang, Yong Yu, Zheng Zhang, and Alexander J Smola. 2020. An efficient neighborhood-based interaction model for recommendation on heterogeneous graph. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 75–84.

[15] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. 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.

[16] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2016. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770* (2016).

[17] Ziqi Liu, Chaochao Chen, Xinxing Yang, Jun Zhou, Xiaolong Li, and Le Song. 2018. Heterogeneous graph neural networks for malicious account detection. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 2077–2085.

[18] Yuanfu Lu, Yuan Fang, and Chuan Shi. 2020. Meta-learning on heterogeneous information networks for cold-start recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1563–1573.

[19] Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. 2021. Are we really making much progress? Revisiting, benchmarking and refining heterogeneous graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1150–1160.

[20] Yao Ma, Suhang Wang, Tyler Derr, Lingfei Wu, and Jiliang Tang. 2019. Attacking graph convolutional networks via rewiring. *arXiv preprint arXiv:1906.03750* (2019).

[21] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).

[22] Yitong Pang, Lingfei Wu, Qi Shen, Yiming Zhang, Zhihua Wei, Fangli Xu, Ethan Chang, Bo Long, and Jian Pei. 2022. Heterogeneous global graph neural networks for personalized session-based recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*. 775–783.

[23] Xiaoru Qu, Zhao Li, Jialin Wang, Zhipeng Zhang, Pengcheng Zou, Junxiao Jiang, Jiaming Huang, Rong Xiao, Ji Zhang, and Jun Gao. 2020. Category-aware graph neural networks for improving e-commerce review helpfulness prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2693–2700.

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

[25] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*. 618–626.

[26] Lichao Sun, Yingtong Dou, Carl Yang, Kai Zhang, Ji Wang, S Yu Philip, Lifang He, and Bo Li. 2022. Adversarial attack and defense on graph data: A survey. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[27] Yiwei Sun, Suhang Wang, Xianfeng Tang, Tsung-Yu Hsieh, and Vasant Honavar. 2020. Adversarial attacks on graph neural networks via node injections: A hierarchical reinforcement learning approach. In *Proceedings of the Web Conference 2020*. 673–683.

[28] Yundong Sun, Dongjie Zhu, Haiwen Du, and Zhaoshuo Tian. 2022. MHNF: Multi-hop Heterogeneous Neighborhood information Fusion graph representation learning. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[29] Xianfeng Tang, Yandong Li, Yiwei Sun, Huaxiu Yao, Prasenjit Mitra, and Suhang Wang. 2020. Transferring robustness for graph neural network against poisoning attacks. In *Proceedings of the 13th international conference on web search and data mining*. 600–608.

[30] Binghui Wang, Jinyuan Jia, Xiaoyu Cao, and Neil Zhenqiang Gong. 2021. Certified robustness of graph neural networks against adversarial structural perturbation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1645–1653.

[31] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 950–958.

[32] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The world wide web conference*. 2022–2032.

[33] Marcin Waniek, Tomasz P Michalak, Michael J Wooldridge, and Talal Rahwan. 2018. Hiding individuals and communities in a social network. *Nature Human Behaviour* 2, 2 (2018), 139–147.

[34] Huijun Wu, Chen Wang, Yuriy Tyshetskiy, Andrew Docherty, Kai Lu, and Liming Zhu. 2019. Adversarial examples on graph data: Deep insights into attack and defense. *arXiv preprint arXiv:1903.01610* (2019).

[35] Weibin Wu, Yuxin Su, Xixian Chen, Shenglin Zhao, Irwin King, Michael R Lyu, and Yu-Wing Tai. 2020. Boosting the transferability of adversarial samples via attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1161–1170.

[36] Kaidi Xu, Hongge Chen, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Mingyi Hong, and Xue Lin. 2019. Topology attack and defense for graph neural networks: An optimization perspective. *arXiv preprint arXiv:1906.04214* (2019).

[37] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. 2019. Graph transformer networks. *Advances in neural information processing systems* 32 (2019).

[38] Mengmei Zhang, Xiao Wang, Meiqi Zhu, Chuan Shi, Zhiqiang Zhang, and Jun Zhou. 2022. Robust heterogeneous graph neural networks against adversarial attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 4363–4370.

[39] Tianyu Zhao, Cheng Yang, Yibo Li, Quan Gan, Zhenyi Wang, Fengqi Liang, Huan Zhao, Yingxia Shao, Xiao Wang, and Chuan Shi. 2022. Space4HGNN: A Novel, Modularized and Reproducible Platform to Evaluate Heterogeneous Graph Neural Network. *arXiv preprint arXiv:2202.09177* (2022).

[40] Qiwei Zhong, Yang Liu, Xiang Ao, Binbin Hu, Jinghua Feng, Jiayu Tang, and Qing He. 2020. Financial defaulter detection on online credit payment via multi-view attributed heterogeneous information network. In *Proceedings of The Web Conference 2020*. 785–795.

[41] Dingyuan Zhu, Ziwei Zhang, Peng Cui, and Wenwu Zhu. 2019. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 1399–1407.

[42] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. 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.