Improving Robustness of GNN-based Anomaly Detection by Graph Adversarial Training

Xiangping Zheng[1,3,5], Bo Wu[2]*, Alex X. Zhang[4], Wei Li[1,3,5]
1.College of Computer Science and Technology, Harbin Engineering University
2.R&D and External Relations Department, Xiangjiang Laboratory
3.Modeling and Emulation in E-Government National Engineering Laboratory
4.Harvest Fund Management Co.,Ltd.
5.Qingdao Innovation and Development Base of Harbin Engineering University
{xpzheng,wei.li}@hrbeu.edu.cn, bowuchn@163.com, zhangxuan01@jsfund.cn

Abstract

Graph neural networks (GNNs) play a fundamental role in anomaly detection, excelling at the identification of node anomalies by aggregating information from neighboring nodes. Nonetheless, they exhibit vulnerability to attacks, with even minor alterations in the graph structure or node attributes resulting in substantial performance degradation. To address this critical challenge, we introduce an innovative mechanism for graph adversarial training, meticulously designed to bolster GNN-based anomaly detection systems against potential poisoning attacks. This novel approach follows a two-step framework. (1) In the initial phase, we employ a Multiple-Objective Generative Adversarial Attack (MO-GAA), which focuses on generating feature modifications and inducing structural disruptions within the graph. Its primary objective is to mimic the adversarial behavior of potential attackers on the anomaly detection graph, with the explicit intention of confounding the anomaly detector. (2) In the subsequent stage, we introduce Purification-Based Adversarial Attack Defense (PB-AAD), a method specifically designed to rectify any contamination and restore the integrity of the graph. The central aim of PB-AAD is to counteract the destructive actions carried out by potential attackers. Our empirical findings, derived from extensive experiments conducted on four real-world anomaly detection datasets, serve to demonstrate how MO-GAA systematically disrupts the graph, compromising the effectiveness of GNN-based detectors, while PB-AAD effectively mitigates these adversarial actions, thereby enhancing the overall robustness of GNN-based anomaly detectors.

Keywords: Graph neural networks, Adversarial Attack, Robustness Detection

1. Introduction

Anomaly detection, a critical task focused on identifying instances that deviate from expected patterns within a dataset, has garnered significant attention and assumes a vital role in various domains, including credit card fraud detection, spam filtering, and hacker intrusion detection (Akoglu et al., 2015; Ma et al., 2023; Zheng et al., 2023a). Nevertheless, the challenge lies in the intricate nature of anomaly detection, owing to issues such as data sparsity and the implicit features that characterize anomalies (Pang et al., 2021; Zheng et al., 2023c). Take e-commerce websites as an example. A multitude of legitimate users engage in purchasing products and assessing merchants based on service quality. However, a small minority of malicious users may intentionally manipulate ratings for illicit gains. These malicious users skillfully blend in with the majority, making it arduous for classifiers to distinguish them from ordinary users, primarily due to the subtlety of their distinguishing features.

Recently, spurred by notable strides in the field of graph neural networks (GNNs), a new wave of GNN-based anomaly detection methodologies has emerged. These techniques are specifically engineered to excel in the identification of anomalies. At their core, GNN-based anomaly detectors function by constructing a graph that establishes connections among diverse objects. They harness the intrinsic capabilities of GNNs to discern anomalies from normal instances. Notably, these detectors offer the advantage of end-to-end and semi-supervised training, effectively mitigating the need for extensive feature engineering and costly data annotation efforts (Dou et al., 2020; Tang et al., 2022; Yu et al., 2021). Diverging from conventional anomaly detection models, GNN-based detectors excel in generating high-caliber node embeddings through the iterative aggregation of information from neighboring nodes. These embeddings effectively encapsulate the fundamental characteristics of anomalous data.

In spite of their commendable achievements, extant GNN-based anomaly detectors exhibit a high susceptibility to significant disruptions in the face of potential attacks on the underlying graph structure. This vulnerability detrimentally affects the performance of GNN-based fraud detection systems. Illustrated in **Figure 1**, we provide a scenario of an attacked financial transaction network, where

*Corresponding author

LREC-COLING 2024, pages 8902–8912
20-25 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

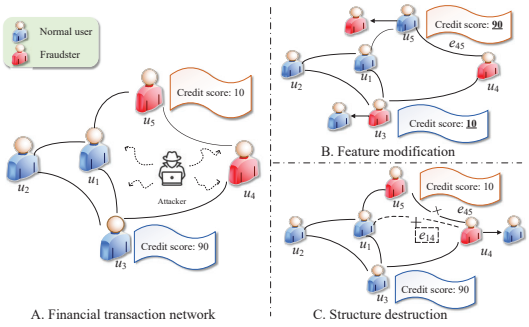


Figure 1: An illustration of an attacked financial transaction network. The blue one and red one denote the normal user and fraudster, respectively. There are two types of attack modes against the anomaly detection graph. (1) Attribute modification: the attacker modifies the attribute of normal user u_5 and fraudster u_3 to make it delusive for detectors (bold underline indicating the modified credit scores). (2) Structure destroy: the attacker destroys the graph structure by deleting edge $e_{4,5}$ (cross) and adding edge e_{14} (dotted line) to attenuate the fraudster's suspiciousness (e.g., fraudster u_4 camouflages himself via connecting many normal users.)

(u_1, u_2, u_3) represent legitimate users (depicted in blue), and (u_4, u_5) represent fraudulent entities (depicted in red). Two distinct modes of attack become evident: **(1) Feature Manipulation**: In this primary mode of attack, the assailant endeavors to manipulate user attributes, aiming to confound anomaly detectors. As observed in Figure 1 (B), the attacker has altered the credit score attribute of the normal user u_2, reducing it from 90 to 60. This diminished credit score diminishes trust in user u_2, often resulting in transaction failures as cautious traders shy away. Conversely, the attacker artificially inflates the credit score of fraudster u_5, raising it from 10 to 90, thus attracting a greater number of users to engage in transactions with them. The attacker's tactics involving attribute manipulation serve as an effective means to evade feature-based detectors. **(2) Structural Disruption**: Graph structure also remains susceptible to manipulation by attackers, as depicted in Figure 1 (C). In this scenario, normal users u_1 and u_3 engage in transactions, thereby creating a transaction edge e_{13} between them. The attacker intervenes by tampering with the transaction records of u_3, specifically deleting edge e_{13}. This action leads to a reduction in the count of successful transactions for u_3, suggesting that u_3 may no longer be considered a normal active user. Consequently, there is an increased likelihood of u_3 being misclassified as a fraudster by the detection system. Similarly, the attacker manipulates the transaction records of fraudster u_4, adding edge e_{14} while deleting edge e_{45}. This strategic maneuver boosts the count of effective, high-quality transactions for fraudster u_4, enabling them to evade detection. Furthermore, the transitive nature of graph connections, in which edges disseminate information across different nodes, amplifies the risk of widespread misclassifications in GNN-based models. These models heavily rely on the recognition of user relationships, thereby exacerbating the severity of misclassifications.

In summary, attackers aim to subvert GNN-based models through a process known as a "poisoning attack," involving the modification of node features and the disruption of graph structure before dataset training. Such attacks pose a significant threat to financial systems, internet security, and various other critical infrastructure. Given this pressing concern, safeguarding GNN-based anomaly detection against poisoning attacks becomes imperative. To address these challenges, we introduce a graph adversarial training mechanism designed to bolster the robustness of GNN-based anomaly detectors. Specifically, we present a Multiple-Objective Generative Adversarial Attack (MO-GAA) mechanism, which employs a generator to replicate attacker behaviors, introducing disturbances to feature and adjacency matrices within defined constraints. MO-GAA, grounded in real-world attack scenarios, maximizes node misclassification and edge prediction errors to befuddle detectors and generate modified feature and adjacency matrices. Following this, we implement a Purification-Based Adversarial Attack Defense (PB-AAD) mechanism, utilizing contrastive learning techniques (Oord et al., 2018). PB-AAD, guided by principles of denoising, exploits low-rank and sparsity attributes to cleanse and fortify the graph structure.

This work yields the following insights and contributions:

(1) We critically analyze the robustness of GNN-based anomaly detection systems. This analysis is crucial for understanding the inherent vulnerabilities and limitations of these detectors, offering valuable perspectives on their operational challenges.

(2) We develop MO-GAA, a model designed to emulate real-life attacker behaviors, alongside PB-AAD, a defense mechanism against graph poisoning attacks. The synergistic application of MO-GAA and PB-AAD unveils the tactics used by attackers to compromise graphs and introduces strategies to address the challenges of potential graph poisoning attacks. To our knowledge, this is the pioneering use of graph adversarial training to counteract poisoning attacks in GNN-based anomaly detection.

(3) We carry out comprehensive experiments across four real-world anomaly detection datasets. The results demonstrate that MO-GAA significantly disrupts the graph structure beyond the effects of

random attacks, while PB-AAD substantially improves the robustness of GNN-based anomaly detection systems.

2. Related Work

Graph Neural Netwoks: Graph neural networks (GNNs) have marked a revolutionary stride in the field of graph representation learning. A variety of GNN frameworks have been instrumental in propelling this success, each contributing unique perspectives and techniques(Zheng et al., 2022a). The Graph Convolutional Network (GCN) (Kipf and Welling, 2017; Zheng et al., 2023b,d) revolutionizes the way information is processed on graphs. It employs a symmetric regularized adjacency matrix alongside a propagation mechanism that enables the extensive dissemination of information across the graph. This approach allows GCNs to effectively capture and integrate node-level information, facilitating a comprehensive understanding of the graph's structure and dynamics. In parallel, the Graph Attention Networks (GAT) (Velickovic et al., 2018) take inspiration from the attention mechanism, allowing for a more nuanced and selective aggregation of neighbor nodes' features. This is achieved through masked self-attentional layers, which assign varying attention weights to different neighboring nodes. Such a mechanism ensures that crucial features are highlighted and integrated, leading to more informed and context-aware node representations. GraphSAGE (Hamilton et al., 2017) extends the versatility of GNNs by introducing a neighbor node sampling strategy combined with aggregation functions such as mean, max, or LSTM. This approach allows for the efficient handling of large-scale graphs by capturing and preserving local structural patterns within the node embeddings, thereby enhancing the model's ability to generalize and adapt to unseen data. Moreover, the Graph Isomorphism Network (GIN) (Xu et al., 2019b; Zheng et al., 2022b) breaks new ground by leveraging a multi-layer perceptron (MLP) to aggregate node features. This methodology aligns closely with the discriminating power of the Weisfeiler-Lehman graph isomorphism test (Weisfeiler and Leman, 1968), establishing GIN as a formidable architecture capable of distinguishing between different graph structures effectively.

GNN-based Anomaly Detection: Recent research has witnessed a notable shift in the field of complex anomaly detection. Researchers have begun to represent intricate anomaly detection data as graph structures and employ GNNs for the classification of abnormal and normal nodes (Dou et al., 2020). CARE-GNN (Dou et al., 2020) constructs a graph based on multiple relations, aggregating node neighborhood information for the purpose of detecting review fraud. GEM (Liu et al., 2018) focuses on learning weighting parameters for different graph structures to effectively detect malicious accounts. Player2Vec (Zhang et al., 2019) encodes both intra- and inter-relation neighbor information, resulting in discriminative node embeddings specifically designed for cybercrime identification. GeniePath (Liu et al., 2019)adapts the receptive fields of GNNs to better capture a wide range of financial fraud-related information. These research endeavors have significantly advanced the field by harnessing the capabilities of GNNs for anomaly detection within graph-structured data.

Adversarial Graph Robustness Analysis: Our research is situated within the realm of adversarial graph robustness analysis, encompassing both adversarial graph attacks and graph adversarial defense mechanisms. Adversarial graph attacks, as explored in previous works (Zügner et al., 2018; Zügner and Günnemann, 2019; Xu et al., 2019a), seek to identify perturbations that disrupt the performance of GNN models in downstream tasks, such as node classification, link prediction, and graph classification. These perturbations typically involve the modification of node features or the addition/deletion of edges within the graph structure. In contrast, the primary objective of graph adversarial defense strategies (Zhu et al., 2019; Entezari et al., 2020; Jin et al., 2020) is to protect the graph from adversarial attacks. These defense mechanisms aim to restore the graph's original integrity by addressing issues such as node noise removal (Zhu et al., 2019) and graph structure purification (Entezari et al., 2020; Jin et al., 2020), thereby enhancing the overall robustness of GNN models. Our work differs from the above methods in two aspects. First, we employ MO-GAA to simulate the attacker's behaviors and propose node and edge misclassification errors to confuse the GNN-based anomaly detectors, whether they are feature-based or relationship-based. Second, we integrate the modified feature and adjacency matrices to PB-AAD and utilize contrastive learning (Oord et al., 2018) to make the discriminator adaptively keep away from noise nodes. Besides, we leverage the low rank and graph sparsity properties to purify the graph structure. To the best of our knowledge, we are the first to introduce graph adversarial training to improve the robustness of GNN-based anomaly detection.

3. Preliminaries

In this section, we first formalize the problem of the GNN framework and anomaly detection on graphs.

Graph. Graphs are ubiquitous data structures that model the pairwise interactions between entities (Pareja et al., 2020), formulated by a quadruple

$\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{A}\}$, where $\mathcal{V}$ and $\mathcal{E}$ represent the set of nodes and edges, respectively. $\mathbf{X} \in \mathbb{R}^{n \times d_0}$ is the initial feature matrix with n and d_0 denoting the number of nodes and the d_0-dimensional initial features, respectively. The adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is constructed by the connection of edges between nodes. where the element $a_{ij} \in \{0, 1\}$ of the adjacency matrix represents whether the node v_i links to the node v_j. Concretely, taking the Figure 1 (A) as an example, we regard the users $(u_1, u_2, u_3, u_4, u_5)$ as nodes while the transaction records $(e_{12}, e_{14}, e_{15}, e_{23}, e_{34}, e_{45})$ can be instantiated edges connected to different users.

In our work, we adopt GIN (Xu et al., 2019b), a state-of-the-art graph neural network, as our encoder, which has been proved to be able to encapsulate the neighborhood homophily as well as the structural homophily (Wang et al., 2021). GIN utilizes the sum-like aggregator and MLP for neighborhood node information transformation, i.e.,

$$\mathbf{h}_i^{(l)} = \mathrm{MLP}^{(l)}\left((1+\epsilon)\cdot \mathbf{h}_i^{l-1} + \sum_{j \in \mathcal{N}(i)} \mathrm{ReLU}\left(\mathbf{h}_j^l\right)\right) \tag{1}$$

where $\epsilon^{(l)}$ is either a learnable parameter or a fixed scalar. We stack L layers to obtain the final node representation $\mathbf{h}_i^L$. Since our approach has a generator G and discriminator D, we denote the final node embedding of generator G as $\mathbf{h}_i^G$, the final node embedding of discriminator D as $\mathbf{h}_i^D$.

Anomaly Detection on Graphs. The purpose of anomaly detection is to distinguish abnormal items and normal items in the datasets, which can be defined as a binary classification problem. However, it is arduously expensive and sometimes infeasible to access sufficient labeled data. Thus we formulate the anomaly detection on a partially labeled graph $\mathcal{G}$ as follows: For a given graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{A}, Y^L\}$, where Y^L is the set of the partial labels on nodes and each $y_i \in Y^L$ is a binary value which takes value 1 if the corresponding node v_i is abnormal and 0 otherwise. The objective of the anomaly detection model is to learn a predictive function in the following:

$$f_\theta := \mathbb{E}_{\mathbf{x}_i \sim \mathcal{G}} \log p\left(y_i \mid \mathbf{x}_i, y_i \in Y^L\right), y_i \in \{0, 1\}. \tag{2}$$

We obtain the final node representation by GIN encoder f_θ and employ a classifier (i.e., MLP) to distinguish the normal and abnormal nodes, where θ is the parameters to be learned.

4. Methodology

Our methodology comprises two distinct components. The first component, Generative Adversarial Attack, is specialized in the art of graph manipulation to deliberately undermine the performance of the anomaly detector. The second component, Adversarial Attack Defense, is dedicated to the task of purging noise from the perturbed graph, thereby fortifying the robustness of anomaly detection. We have conveniently summarized all essential notations which are employed in this paper within **Table 1** for quick reference.

Symbol	Definition
$\mathcal{G}$; $\mathcal{V}$; $\mathcal{E}$; Y^L	Graph; Node set; Edge set; Node label set
$\mathbf{X}$; $\mathbf{A}$	Feature matrix; Adjacent matrix
$\mathbf{X}'$; $\mathbf{A}'$	Modified feature matrix; Modified adjacent matrix
G;D	Generator; Discriminator
$\mathcal{N}_i$	The neighbors of node v_i
$\mathcal{S}_i$	The negative sample set for node v_i
l; L	GNN layer number; Total number of layers
$\mathbf{h}_i^G$; $\mathbf{h}_i^D$	The final node embedding of G and D
n;d_0	The number of nodes; The initial dimension of features

Table 1: Glossary of Notations

4.1. Multiple-objective Generative Adversarial Attack

In this section, we embark on a systematic exploration of techniques for undermining the performance of anomaly detection through graph manipulation. As illustrated in **Figure 1**, it's important to note that malicious attackers may not only tamper with user node features (graph attributes) but also disrupt the connections between users (graph structure), thereby muddling the detector's ability to identify outliers. Our objective is for the generator G to faithfully emulate real-world attack scenarios by attempting to modify both the features and adjacency matrix, much like genuine attackers. To achieve this, we construct two transformation matrices implemented using Multilayer Perceptrons (MLP), i.e.,

$$\mathbf{X}' = \mathbf{X} \cdot \mathbf{M}, \mathbf{A}' = \mathbf{A} \cdot \mathbf{S} \tag{3}$$

where $\mathbf{M} \in \mathbb{R}^{n \times d_0}$ and $\mathbf{S} \in \mathbb{R}^{n \times n}$ denotes feature transformation matrix and adjacency transformation matrix.

Indeed, the most straightforward method of attack would be to erase all node features and delete every edge, resulting in the complete obliteration of the entire graph. Yet, such a scenario is almost implausible in real-world situations. Therefore, the underlying principle of the generative adversarial attack hinges on the generator G being capable of automatically generating subtle perturbations that minimize their detectability by the GNN anomaly detector. To maintain some level of realism and ensure that the generator doesn't radically alter the graph, we introduce a constraint by limiting the number of permissible alterations using a predefined budget Δ as inspired by (Zügner et al., 2018).

$$\|\mathbf{A}' - \mathbf{A}\|_0 + \|\mathbf{X}' - \mathbf{X}\|_0 \leq \Delta \tag{4}$$

where $\mathbf{A}'$ and $\mathbf{X}'$ are modified adjacency matrix and feature matrix, respectively.

Equation 2 illustrates how we can derive the classification probability of node v_i by employing the GIN encoder and maximize the probability that node v_i pertains to the correct label y_i. Conversely, if our objective is to undermine the node classification performance of the detector by attacking the graph, we should ensure that the classification probability of node v_i belonging to the original category y_i is no longer the highest. Consequently, we formulate a loss function for all nodes as follows:

$$\mathcal{L}_n = \sum_{i}^{n} (\max_{c \neq y_i} \ln \sigma(\mathbf{h}_{i,c}^{G}) - \ln \sigma(\mathbf{h}_{i,y_i}^{G})) \quad (5)$$

where σ represents softmax function and denotes the probability of node assigning node v_i to its original class y_i or another class $c\,(c \neq y_i)$.

Equation 5 outlines how to directly employ node classification errors to guide the generator in modifying the original adjacency matrix $\mathbf{A}$ and feature matrix $\mathbf{X}$ into $\mathbf{A}'$ and $\mathbf{X}'$ under the limited condition Δ. However, the loss function $\mathcal{L}_n$ solely concerns the individual node, leading the generator to focus on modifying node features without much sensitivity to the graph's structural changes. As depicted in **Figure 2**, the trained discriminator D adeptly discriminates between abnormal nodes (indicated by blue) and normal nodes (depicted in red), with the dotted line signifying the classification boundary. When the graph is subjected to an attack, and features are altered, the abnormal node $v0$ shifts below the boundary, while the normal node v_1 moves above it. As a result, $v0$ may appear camouflaged, resembling normal nodes. Nevertheless, we can still identify $v0$ by considering its proximity to v_2 and v_3 (i.e., other abnormal nodes) since they maintain close relationships. In a similar manner, we can determine that v_1 is a normal node based on its connections to neighbors v_4 and v_5. From this perspective, maximizing node classification errors predominantly affects node features and may not be as effective for relational GNN-based anomaly detectors. Therefore, we propose utilizing edge prediction errors to gauge the extent of damage to the graph's structure. Specifically, for a given node v_i, we can assess edge predictions between node v_i and v_j as a binary classification problem:

$$\sigma(\mathbf{h}_i^{G(\top)} \cdot \mathbf{h}_j^{G}) = 1, v_j \in \mathcal{N}_i \quad (6)$$

where $\top$ denotes transpose of vector $\mathbf{h}_i^G$, and node v_j is the neighborhood node of v_i. Then we compute edge prediction error on the whole graph $\mathcal{G}$ and employ negative sampling (Mikolov et al., 2013)

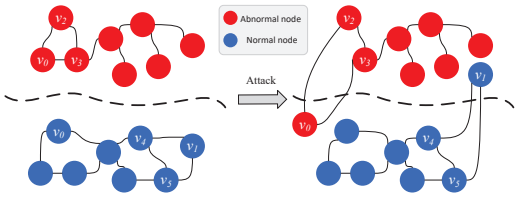


Figure 2: An example of misclassification. The trained detector can distinguish abnormal nodes from normal nodes, and the dotted line represents the classification boundary. After attacking, the feature-based detector will misclassify v_0 and v_1, but the relationship-based detector can still identify normal and abnormal nodes from the connections of neighbor nodes.

to obtain loss function:

$$\mathcal{L}_e = \sum_{i}^{n} \sum_{j}^{\mathcal{N}_i} (\log \sigma(\mathbf{h}_i^{G(\top)} \cdot \mathbf{h}_j^{G}) - \log \sigma(\mathbf{h}_i^{G(\top)} \cdot \mathbf{h}_k^{G})) \quad (7)$$

where $\mathbf{h}_k^L$ denotes the node vector, of which means v_k is not neighborhood nodes of v_i, i.e., $v_k \notin \mathcal{N}_i$ We define $\mathcal{S}_i$ as negative sample set for node v_i and use a negative sampling ratio of 1 (one negative pair per one positive pair, i.e., $|\mathcal{S}_i| = |\mathcal{N}_i|$). Then, We combine the node classification and edge prediction errors to obtain the final loss function, i.e.,

$$\mathcal{L}_G = \alpha \mathcal{L}_n + (1 - \alpha)\mathcal{L}_e \quad (8)$$

where α denotes the weight.

4.2. Purification-based Adversarial Attack Defence

The basic idea of graph purification-based adversarial attack defense (PB-AAD) is how to utilize the modified feature and adjacency matrices to avoid the interference of node noise and purify the graph structure so that the trained GNN-based detector can correctly classify the future adversarial samples. To tackle the above challenges, we leverage contrastive learning to denoise the node and a graph structure purification mechanism, which provides great help for GNN-based detector defense against the potential attack.

Concretely, for the contrastive learning, we use a discriminator to encode $(\mathbf{X}, \mathbf{A})$ and $(\mathbf{X}', \mathbf{A}')$, respectively. The node vector $\mathbf{h}_i^D$ obtained from $(\mathbf{X}, \mathbf{A})$ represents positive sample while the node vector $(\mathbf{h}_j^G)'$ obtained from $(\mathbf{X}', \mathbf{A}')$ represents negative sample. We use the InfoNCE loss (Oord et al., 2018) to increase the difference between positive and negative sample pairs so that the clean nodes are far away from the noisy nodes. Positive sample pairs can be represented by node v_i and its

surrounding neighbors. We use a first-order r-ego network, which can be defined as:

Definition 1. r-ego network. Ego network originated from social networks (Li et al., 2014). It consists of a focal node (ego) and the nodes to which the ego is directly connected, plus the edges. Given a vertex v, the r-ego network can be defined as a subgraph $S_v = \{(u, e) \mid d(v, u) \leq r\}$. The $d(v, u)$ represents the shortest path distance between node v and u in graph $\mathcal{G}$. In our work, we adopt the first-order ego network (i.e., 1-ego) to represent each node's local structure.

Then, we denote the positive pair and negative pair as $(\mathbf{h}_i^D, \mathbf{h}_i^{ego})$ and $(\mathbf{h}_i, (\mathbf{h}_j^G)')$, respectively. InfoNCE loss can be described as a process that only uses its own information to distinguish positive and negative samples. The formula is as follows:

$$\mathcal{L}^{node} = -\log \frac{\exp\left(\mathbf{h}_i^{D(\top)} \mathbf{h}_i^{D(ego)} / \tau\right)}{\sum_{j=0}^{n} \exp\left(\mathbf{h}_i^{D(\top)} (\mathbf{h}_j^G)' / \tau\right)} \tag{9}$$

where τ is the temperature hyper-parameter.

For the purification of the graph structure, we delve into the low-rank and sparsity properties of the graph, which are conducive to the refinement and restoration of the graph structure, as highlighted in prior work (Jin et al., 2020). Anomaly detection graphs inherently exhibit low-rank and sparsity characteristics, as both anomalous and normal nodes often tend to cluster within communities, with connections limited to a relatively small number of neighbors. In such a context, we employ the discriminator to derive the reconstructed adjacency matrix using node embeddings and the Hadamard product, aimed at restoring the original graph structure from the noisy and perturbed graph. This process can be framed as a structural learning problem as follows:

$$\begin{aligned} \mathcal{L}^{graph} = -||\sigma(\mathbf{H}^G \odot \mathbf{H}^{G(\top)}) - \mathbf{A}'||_2^2, \\ \mathbf{H}^G \odot \mathbf{H}^{G(\top)} \in \mathbb{R}^{n \times n} \end{aligned} \tag{10}$$

where σ represents the sigmoid activate function and $\mathbf{H}^D$ is node embeddings of the whole graph. $||\cdot||_2^2$ denotes the ℓ_2 norm, which enforces the reconstruct adjacency to retain the properties of low rank and sparsity. We combine the loss $\mathcal{L}^{node}$, $\mathcal{L}^{graph}$ and cross-entropy loss of node classification with a total loss as follows:

$$\mathcal{L}_D = \mathcal{L}^{node} + \beta \mathcal{L}^{graph} + \gamma \mathcal{L}_{CE} \tag{11}$$

where β and γ is a balancing coefficient. The cross-entropy loss $\mathcal{L}_{CE}$ can be defined as:

$$\mathcal{L}_{CE} = -\frac{1}{|Y^L|} \sum_{i=1}^{|Y^L|} \left(y_i \cdot \log p_i + (1 - y_i) \cdot \log\left(1 - p_i\right)\right). \tag{12}$$

4.3. Model Discussion

In our research, we employ the Multiple-Objective Generative Adversarial Attack (MO-GAA) to meticulously emulate the objectives of potential attackers. This is achieved by generating modified feature and adjacency matrices, which are meticulously tailored based on the errors associated with node classification and edge prediction. These adjustments are strategically designed to mimic the sophisticated strategies that attackers might employ.

Following the generation of these perturbed matrices, they serve as the foundation for our adversarial training process. This is a critical step in our methodology, as it allows our model to learn and adapt from the intricacies and nuances introduced by the MO-GAA, enhancing its defensive capabilities against potential adversarial threats.

To further bolster the robustness of our approach, we incorporate contrastive learning mechanisms aimed at effectively identifying and eliminating noise from the nodes. This step is crucial for ensuring the accuracy and reliability of our node representations, significantly reducing the likelihood of false positives and negatives that can arise from noisy data.

Additionally, we apply ℓ_2 normalization to the graph's structure, an essential process that aims to enhance the clarity and purity of the graph. This normalization process helps in maintaining the integrity of the graph structure, ensuring that the relationships and interactions within the graph are accurately represented, free from distortions that could impact the model's performance.

The computational complexity of our model is delineated as $\mathcal{O}(n + n * |\mathcal{E}| + n^2)$, where n represents the number of nodes and $\mathcal{E}$ denotes the number of edges. This complexity reflects the comprehensive nature of our approach, accounting for individual nodes, their interactions through edges, and the overall network structure, ensuring a holistic and thorough analysis and fortification of the graph against adversarial activities. This multi-faceted approach underlines the depth and breadth of our study, highlighting its significance and potential impact in the realm of anomaly detection.

5. Experiments

5.1. Experimental Settings

Dataset	#Users(% normal, abnormal)	#Objects	#Edges
Reddit	10,000 (96.34%, 3.66%)	984	78,516
Wiki	8,227 (97.36%, 2.64%)	1,000	18,257
Alpha	3,286 (61.21%, 38.79%)	3,754	24,186
Amazon	27,197 (91.73%, 8.27%)	5,830	52,156

Table 2: Dataset and graph statistics

Dataset: We have selected four real anomaly detection datasets to evaluate the effectiveness of our approach. The datasets are as follows: **(1) Reddit** (Kumar et al., 2019): This dataset represents a user-subreddit graph, and it contains ground truth tags for forbidden users (abnormal users). **(2) Wiki** (Kumar et al., 2019): The Wiki dataset is an editor-page graph, capturing one month of edits on Wikipedia pages. It includes public ground-truth labels for banned users (abnormal users). **(3) Alpha** (Kumar et al., 2018): Derived from the Bitcoin trading website Alpha, the Alpha dataset is a user-user trust graph that describes user ratings during transactions. Users with excessively low credit scores are labeled as fraudsters (abnormal users). **(4) Amazon** (Kumar et al., 2018): This dataset is a user-product graph. Users who engage in malicious activities (Fayazi et al., 2015) are considered fraudsters (abnormal users). For a summary of the dataset statistics, please refer to **Table 2**.

Baselines: We select various GNN-based anomaly detectors to verify the attack ability of MO-GAA and compare the defensive capability of PB-AAD to these baselines. We choose the GCN (Kipf and Welling, 2017)„ GAT (Velickovic et al., 2018), GraphSAGE (Hamilton et al., 2017) and GIN (Xu et al., 2019b) to represent general GNN models. In addition, we select GeniePath (Liu et al., 2019), CARE-GNN (Dou et al., 2020) and CONAD (Jin et al., 2021) as three state-of-the-art GNN-based fraud detectors. Since CONAD (Jin et al., 2021) learning is performed in an unsupervised way. We use SVM to do supervised node classification after obtaining the node embeddings by CONAD.

Parameter Settings: To ensure a fair comparison, we maintain consistent variables and follow optimal configurations for all baseline models. Specifically, we set the input feature dimension to 64, node representation dimension to 128, the number of GNN layers to 2, learning rate to 0.01, and employ the Adam optimizer (Kingma and Ba, 2014) across all models. For CARE-GNN, we adhere to its default setting, which fixes the number of GNN layers at 1 as specified in its source code. Default values of 0.1 are used for parameters Δ, α, β and γ. An early stopping strategy is applied during the training of CARE-GNN due to observed performance degradation with excessive training epochs. For GCN, GIN, GAT, GraphSage, CARE-GNN, CONAD, GeniePath, we use the source code provided by their authors. We modified these codes to make them adapt to our tasks. We record the best testing results after 300 epochs in each fold and report the average best AUC score across different folds. In real-world anomaly detection scenarios, data points are often imbalanced, with normal users dominating the dataset, and abnormal users, such as fraudsters, representing a minority. Consequently, we evaluate the performance using the widely recognized ROC-AUC (AUC) metric. We conduct a 10-fold evaluation on all datasets except for Amazon, where limited labeled abnormal nodes require a 5-fold evaluation.

Hardware Environment: Our experiments were conducted on the following hardware setup: CPU: AMD 2700X CPU @ 3.70GHz, GPU: NVIDIA GTX 1080Ti@11GB, Memory: 64GB.

5.2. Experimental Results

In this section, we evaluate the performance of our model to defend against the attack in anomaly detection on real-world datasets. Particularly, we mainly answer the following research questions:

- **RQ1**: How does our multiple-objective generative adversarial attack (MO-GAA) perform compared to a random attack (i.e., drop the edges randomly)?
- **RQ2**: How does purification-based adversarial attack defense (PB-AAD) perform under the MO-GAA and simple attack compared to other state-of-the-art GNN-based anomaly detection modes?
- **RQ3**: How does every single module in MO-GAA and PB-AAD work?
- **RQ4**: How do hyper-parameters affect the MO-GAA and PB-AAD?

5.3. Overall Comparison

- **RQ1**. Overall, as shown in **Table 3**, our proposed model, MO-GAA, effectively simulates the attacker's behavior by incorporating node and edge classification errors, resulting in performance degradation across all GNN-based anomaly detectors. In comparison to random attacks, MO-GAA inflicts more significant damage on the anomaly detection graph under the same constraints, highlighting the efficiency of our proposed model MO-GAA.
- **RQ2.** First, when the anomalydetection graph is clean, PB-AAD consistently shows strong performance across all datasets, which ascertains our proposed method's effectiveness. We note that existing baselines have already obtained high enough performance, while our approach still pushes that boundary forward. Second, in the case of random perturbation on the graph, the performance of PB-AAD is only slightly degraded compared to other GNN-based anomaly detectors. Finally, under the fierce attack of MO-GAA, our proposed method PB-AAD can still maintain good performance, which shows good anti-interference and robustness.

Models	Wiki			Reddit			Alpha			Amazon		
Attack mode	Clean	Random	MO-GAA	Clean	Random	MO-GAA	Clean	Random	MO-GAA	Clean	Random	MO-GAA
GCN	0.7271	0.6350	0.5801	0.7210	0.6580	0.6154	0.8358	0.7568	0.6450	0.8375	0.7116	0.6130
GIN	0.7290	0.6512	0.6120	0.7029	0.6415	0.5913	0.8465	0.7613	0.6382	0.7927	0.7216	0.6145
GAT	0.7237	0.6671	0.5515	0.7165	0.6423	0.5864	0.8314	0.7413	0.6478	0.8257	0.7161	0.6416
GraphSAGE	0.7241	0.6315	0.6130	0.7200	0.6123	0.5913	0.8343	0.7513	0.6519	0.7738	0.7194	0.6276
CARE-GNN	0.7320	0.6846	0.6400	0.7200	0.6689	0.6116	0.8420	0.7656	0.6916	0.8290	0.7260	0.6513
GeniePath	0.7390	0.6870	0.6541	0.7254	0.6661	0.6193	0.8490	0.7843	0.6984	0.8380	0.7312	0.6530
CONAD	0.7023	0.6581	0.6223	0.7034	0.6523	0.6141	0.8445	0.7546	0.6813	0.8021	0.7198	0.6313
PB-AAD	**0.7412**	**0.7113**	**0.6913**	**0.7310**	**0.6813**	**0.6345**	**0.8513**	**0.8219**	**0.7214**	**0.8513**	**0.7646**	**0.7013**

Table 3: Results of ROC-AUC. The bold denotes the best results. We denote the anomaly detection graph without attack as clean. The random means random permutation on the anomaly detection graph.

Model	Clean	MO-GAA-n	MO-GAA-e	MO-GAA
GCN	0.8358	0.7012	0.6913	0.6450
GIN	0.8465	0.7015	0.6816	0.6382
GAT	0.8314	0.7067	0.6791	0.6478
GraphSAGE	0.8343	0.7131	0.7016	0.6519
CARE-GNN	0.8420	0.7101	0.7264	0.6916
GeniePath	0.8490	0.7113	0.7264	0.6984
CONAD	0.8445	0.7046	0.7045	0.6813
PB-AAD-d	0.8510	0.7416	0.7348	0.7048
PB-AAD-s	0.8499	0.7365	0.7365	0.7164
PB-AAD	**0.8513**	**0.7553**	**0.7439**	**0.7214**

Table 4: Ablation study for MO-GAA and PB-AAD.

5.4. Model Analysis

Next, we investigate the underlying mechanism of MO-GAA and PB-AAD. Since similar trends are observed for different datasets, here we only report the results w.r.t. the Alpha dataset.

5.4.1. Ablation Study.

To answer the **RQ3**, We conducted an ablation study to examine the various components of MO-GAA and PB-AAD. We designed two ablated variants for MO-GAA, specifically MO-GAA-n (with only node classification error) and MO-GAA-e (with only edge classification error). We assessed both node and edge classification errors. Furthermore, we created two ablated variants for PB-AAD, known as PB-AAD-n (with only node classification error) and PB-AAD-e (with only edge classification error). We compared these four ablated variants against each other, as well as the baseline methods. As shown in **Table 4**, in the case of MO-GAA, both MO-GAA-n and MO-GAA-e resulted in performance degradation across all baseline models. Notably, MO-GAA-e appeared to be more detrimental than MO-GAA-n, suggesting that edge classification errors have a more pronounced impact on the internal mechanisms of GNN-based detectors. On the other hand, PB-AAD-d and PB-AAD-s effectively defended against MO-GAA itself and its variants, highlighting the ability of the node denoising module and structure purification to eliminate noisy nodes and interfering edges. Additionally, PB-AAD-d outperformed PB-AAD-s, indicating that PB-AAD-n could enhance the robustness of GNN-based detectors in most scenarios.

model	0.1	0.15	0.2	0.25	0.3
GCN	0.6450	0.6268	0.5912	0.5671	0.5412
GIN	0.6382	0.6343	0.5752	0.5712	0.5612
GAT	0.6478	0.6216	0.5452	0.5413	0.5411
GraphSAGE	0.6519	0.6244	0.5466	0.5431	0.5314
CARE-GNN	0.6916	0.6714	0.6121	0.5812	0.5612
GeniePath	0.6984	0.6731	0.6041	0.5712	0.5711
CONAD	0.6813	0.6524	0.5912	0.5721	0.5643
PB-AAD	**0.7214**	**0.7012**	**0.6542**	**0.6014**	**0.5864**

Table 5: Parameter analysis for budget Δ

5.4.2. Parameter Analysis.

To answer **RQ4**, we executed a comprehensive parameter analysis to examine the sensitivity of our model to various hyper-parameters.

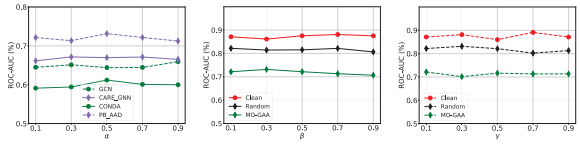
Initially, we adjusted the threshold parameter, denoted as Δ, across a set of values: $\{0.1, 0.15, 0.2, 0.25, 0.3\}$. This variation aimed to assess the influence of different operational budgets allocated for the MO-GAA process. As illustrated in **Tabel 5** , a trend becomes apparent: increasing the Δ parameter, which corresponds to loosening the constraints, results in a more pronounced decline in the performance of GNN-based detectors. This observation is critical as it underscores the impact of resource allocation on the efficacy of adversarial attacks within our model framework.

Subsequently, we delved into the effects of modifying the α parameter, testing values within the range $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. The graphical data presented in Figure 3a indicates that the impact of MO-GAA remains consistent across these variations, suggesting that our model's response to MO-GAA is robust against changes in the α parameter.

Similarly, we explored adjustments to the β and γ parameters within the same value spectrum $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. The performance metrics,

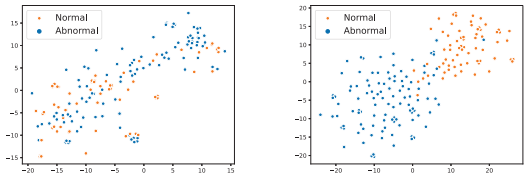
depicted in **Figure 3(b)** and **3(c)**, reveal that PB-AAD maintains effective operation despite fluctuations in β and γ values. This indicates a significant level of stability and resilience in our defense mechanism, showcasing its capability to safeguard against adversarial threats under varied settings.

Overall, the analysis provides essential insights into the operational characteristics of our proposed MO-GAA and PB-AAD approaches. It establishes that while certain parameters critically influence the model's susceptibility to adversarial tactics, our defense mechanisms exhibit commendable stability and adaptability across a range of parameter settings. This reinforces the practical applicability and robustness of our solutions in the dynamic landscape of anomaly detection.



(a) Different α. (b) Different β. (c) Different γ.

Figure 3: Parameter sensitivity study on the Alpha dataset in experiments.



(a) PB-AAD (MO-GAA). (b) PB-AAD (random).

Figure 4: Graph visualizations. Blue indicates abnormal nodes and red as normal nodes.

5.4.3. Graph visualization.

Graph visualization stands out as a powerful technique for the qualitative assessment of node embeddings generated by various methodologies. We apply the t-SNE technique (van der Maaten and Hinton, 2008), a well-regarded dimensionality reduction method, to translate the embeddings of abnormal and normal nodes from the Amazon dataset into a two-dimensional framework. In this visualization, abnormal nodes are depicted with blue circles, while normal nodes are represented by red circles, facilitating an intuitive understanding of the spatial distribution and separation of these entities.

As indicated in **Figure 4(a)**, PB-AAD exhibits remarkable capability in segregating abnormal from normal nodes into distinct clusters, even when subjected to random permutations. This clear delineation underlines PB-AAD's robustness and its ability to effectively counteract the noise and interference typically associated with random attacks. The ability to maintain this separation is crucial for the practical utility of anomaly detection systems, particularly in environments characterized by unpredictable disruptions.

Furthermore, our examination extends to scenarios involving more sophisticated attacks, specifically those orchestrated using the MO-GAA technique. As demonstrated in **Figure 4(b)**, despite the enhanced complexity and targeted nature of MO-GAA attacks, PB-AAD manages to maintain a discernible separation between abnormal and normal nodes. Although there is a slight intermingling of nodes, indicating a challenging adversarial environment, a visible boundary remains between the two categories. This suggests that, while not completely impervious, PB-AAD significantly enhances the resilience of graph-based anomaly detection systems against sophisticated adversarial manipulations, affirming its value in enhancing security measures in data-sensitive domains.

6. Conclusion

In this study, we explore the effects of poison attacks on anomaly detection graphs, which lead to a significant degradation in GNN-based detectors' performance. To tackle this issue, we adopt a dual-step strategy. Initially, we replicate attacker behaviors using the Multiple-Objective Generative Adversarial Attack (MO-GAA), exploiting node and edge classification inaccuracies to compromise the anomaly detection graph. Next, we enhance defenses against the disruptive influences of MO-GAA with the Purification-Based Adversarial Attack Defense (PB-AAD). This countermeasure incorporates a node denoising component through contrastive learning and a structural purification strategy to remove misleading edges. Our comprehensive experiments demonstrate that MO-GAA generates more detrimental effects compared to arbitrary attacks. Nonetheless, PB-AAD proves effective in defending against potential poison attacks on the anomaly detection graph, thereby significantly improving the resilience of GNN-based anomaly detectors.

7. Acknowledgements

This research was sponsored by the Stable Supporting Fund of National Key Laboratory of Underwater Acoustic Technology, JCKYS 2023604SSJS013, and Natural Science Foundation of Heilongjiang Province, LH2023F020, National Natural Science Foundation of China, 62272126.

8. References

Leman Akoglu, Hanghang Tong, and Danai Koutra. 2015. Graph based anomaly detection and description: a survey. *Data Min. Knowl. Discov.*, 29(3):626–688.

Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S. Yu. 2020. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *CIKM*, pages 315–324.

Negin Entezari, Saba A. Al-Sayouri, Amirali Darvishzadeh, and Evangelos E. Papalexakis. 2020. All you need is low (rank): Defending against adversarial attacks on graphs. In *WSDM*, pages 169–177.

Amir Fayazi, Kyumin Lee, James Caverlee, and Anna Cinzia Squicciarini. 2015. Uncovering crowdsourced manipulation of online reviews. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, pages 233–242. ACM.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*, pages 1024–1034.

Ming Jin, Yixin Liu, Yu Zheng, Lianhua Chi, Yuan-Fang Li, and Shirui Pan. 2021. ANEMONE: graph anomaly detection with multi-scale contrastive learning. In *CIKM*, pages 3122–3126.

Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. 2020. Graph structure learning for robust graph neural networks. In *KDD*, pages 66–74.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Srijan Kumar, Bryan Hooi, Disha Makhija, Mohit Kumar, Christos Faloutsos, and V. S. Subrahmanian. 2018. REV2: fraudulent user prediction in rating platforms. In *WSDM*, pages 333–341.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. 2019. Predicting dynamic embedding trajectory in temporal interaction networks. In *SIGKDD*, pages 1269–1278.

Rui Li, Chi Wang, and Kevin Chen-Chuan Chang. 2014. User profiling in an ego network: co-profiling attributes and relationships. In *WWW*, pages 819–830.

Ziqi Liu, Chaochao Chen, Longfei Li, Jun Zhou, Xiaolong Li, Le Song, and Yuan Qi. 2019. Geniepath: Graph neural networks with adaptive receptive paths. In *AAAI*, pages 4424–4431.

Ziqi Liu, Chaochao Chen, Xinxing Yang, Jun Zhou, Xiaolong Li, and Le Song. 2018. Heterogeneous graph neural networks for malicious account detection. In *CIKM*, pages 2077–2085.

Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z. Sheng, Hui Xiong, and Leman Akoglu. 2023. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Trans. Knowl. Data Eng.*, 35(12):12012–12038.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, pages 3111–3119.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel. 2021. Deep learning for anomaly detection: A review. *ACM Comput. Surv.*, 54(2):38:1–38:38.

Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao B. Schardl, and Charles E. Leiserson. 2020. Evolvegcn: evolving graph convolutional networks for dynamic graphs. In *AAAI*, pages 5363–5370.

Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. 2022. Rethinking graph neural networks for anomaly detection. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 21076–21089. PMLR.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.

Yanling Wang, Jing Zhang, Shasha Guo, Hongzhi Yin, Cuiping Li, and Hong Chen. 2021. Decoupling representation learning and classification for gnn-based anomaly detection. In *SIGIR*, pages 1239–1248.

B. Y. Weisfeiler and A. A. Leman. 1968. A reduction of a graph to a canonical form and an algebra arising during this reduction (in russian).

Kaidi Xu, Hongge Chen, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Mingyi Hong, and Xue Lin. 2019a. Topology attack and defense for graph neural networks: An optimization perspective. In *IJCAI*, pages 3961–3967.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019b. How powerful are graph neural networks? In *ICLR*.

Zhiyong Yu, Xiangping Zheng, Fangwan Huang, Wenzhong Guo, Lin Sun, and Zhiwen Yu. 2021. A framework based on sparse representation model for time series prediction in smart city. *Frontiers Comput. Sci.*, 15(1):151305.

Yiming Zhang, Yujie Fan, Yanfang Ye, Liang Zhao, and Chuan Shi. 2019. Key player identification in underground forums over attributed heterogeneous information network embedding framework. In *CIKM*, pages 549–558.

Xiangping Zheng, Xun Liang, and Bo Wu. 2023a. Select the best: Enhancing graph representation with adaptive negative sample selection. In *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023*, pages 1–5. IEEE.

Xiangping Zheng, Xun Liang, Bo Wu, Junlan Feng, Yuhui Guo, and Sensen Zhang. 2023b. Intent does matter! propagating high-order relations for exploring interest preferences. In *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023*, pages 1–5. IEEE.

Xiangping Zheng, Xun Liang, Bo Wu, Yuhui Guo, and Hui Tang. 2022a. Adaptive attention graph capsule network. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022*, pages 3588–3592. IEEE.

Xiangping Zheng, Xun Liang, Bo Wu, Yuhui Guo, and Xuan Zhang. 2022b. Graph capsule network with a dual adaptive mechanism. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 1859–1864. ACM.

Xiangping Zheng, Xun Liang, Bo Wu, Jun Wang, Yuhui Guo, Sensen Zhang, and Yuefeng Ma. 2023c. Modeling high-order relation to explore user intent with parallel collaboration views. In *Database Systems for Advanced Applications - 28th International Conference, DASFAA 2023, Tianjin, China, April 17-20, 2023, Proceedings, Part II*, volume 13944 of *Lecture Notes in Computer Science*, pages 489–504. Springer.

Xiangping Zheng, Xun Liang, Bo Wu, Jun Wang, Yuhui Guo, Xuan Zhang, and Yuefeng Ma. 2023d. A multi-scale interaction motion network for action recognition based on capsule network. In *Proceedings of the 2023 SIAM International Conference on Data Mining, SDM 2023, Minneapolis-St. Paul Twin Cities, MN, USA, April 27-29, 2023*, pages 505–513. SIAM.

Dingyuan Zhu, Ziwei Zhang, Peng Cui, and Wenwu Zhu. 2019. Robust graph convolutional networks against adversarial attacks. In *SIGKDD*, pages 1399–1407.

Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. 2018. Adversarial attacks on neural networks for graph data. In *SIGKDD*, pages 2847–2856.

Daniel Zügner and Stephan Günnemann. 2019. Adversarial attacks on graph neural networks via meta learning. In *ICLR*.