

Reinforcement Neighborhood Selection for Unsupervised Graph Anomaly Detection

Yuanchen Bei¹, Sheng Zhou^{1†}, Qiaoyu Tan², Hao Xu³,
Hao Chen⁴, Zhao Li¹, Jiajun Bu¹

¹ Zhejiang University, Hangzhou, China

² New York University Shanghai, Shanghai, China

³ Unaffiliated, Beijing, China

⁴ The Hong Kong Polytechnic University, Hong Kong SAR, China

yuanchenbei@zju.edu.cn, zhousheng_zju@zju.edu.cn, qiaoyu.tan@nyu.edu, kingsleyhsu1@gmail.com,
sundaychenhao@gmail.com, lzjoey@gmail.com, bjj@zju.edu.cn

Abstract—Unsupervised graph anomaly detection is crucial for various practical applications as it aims to identify anomalies in a graph that exhibit rare patterns deviating significantly from the majority of nodes. Recent advancements have utilized Graph Neural Networks (GNNs) to learn high-quality node representations for anomaly detection by aggregating information from neighborhoods. However, the presence of anomalies may render the observed neighborhood unreliable and result in misleading information aggregation for node representation learning. Selecting the proper neighborhood is critical for graph anomaly detection but also challenging due to the absence of anomaly-oriented guidance and the interdependence with representation learning. To address these issues, we utilize the advantages of reinforcement learning in adaptively learning in complex environments and propose a novel method that incorporates Reinforcement neighborhood selection for unsupervised graph ANomaly Detection (RAND). RAND begins by enriching the candidate neighbor pool of the given central node with multiple types of indirect neighbors. Next, RAND designs a tailored reinforcement anomaly evaluation module to assess the reliability and reward of considering the given neighbor. Finally, RAND selects the most reliable subset of neighbors based on these rewards and introduces an anomaly-aware aggregator to amplify messages from reliable neighbors while diminishing messages from unreliable ones. Extensive experiments on both three synthetic and two real-world datasets demonstrate that RAND outperforms the state-of-the-art methods.

Index Terms—graph anomaly detection, unsupervised learning, neighborhood selection, message passing

I. INTRODUCTION

Unsupervised graph anomaly detection aims to discover the graph anomalies that exhibit rare patterns from the majority in an unsupervised manner. It has gained increasing attention from both academia and industry in recent years for the sparsity of anomaly labels and the significant real-world applications [1], [2], such as financial fraud detection [3], social rumor detection [4], and computer network intrusion detection [5]. Benefiting from the success of Graph Neural Networks (GNNs) in learning effective node representations [6], [7], recent advances have widely adopted GNNs to learn high-quality node embeddings for unsupervised graph anomaly detection [8], [9].

† Corresponding author.

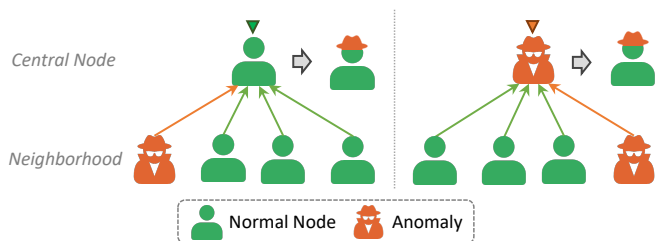


Fig. 1. A toy example of the negative impact on central normal nodes (left) and anomaly nodes (right) by unreliable neighborhood in GNN-based representation learning and anomaly detection.

The existing GNN-based methods for graph anomaly detection have mostly learned representation by aggregating information from neighborhoods. This is motivated by the *homophily assumption* that nodes close in the graph tend to exhibit similar patterns as the central nodes [10], which is widely observed in real-world graphs. However, in the graph anomaly detection scenario, the presence of anomalies can disrupt this homophily assumption and render the **neighborhood unreliable**. More specifically, the misleading information aggregated from the unreliable neighborhood will result in suboptimal representation learning and anomaly detection performance. For the normal nodes, as illustrated in the left part of Figure 1, they suffer from aggregating noisy information from the anomaly nodes included in their neighborhood, resulting in inaccurate node modeling. For the anomaly nodes, as illustrated in the right part of Figure 1, they can camouflage themselves as normal ones by establishing connections with numerous regular nodes [11] and aggregating conventional messages from their normal neighborhood. Consequently, the observed neighborhood becomes less reliable due to the presence of anomalies, making it crucial to carefully select appropriate neighborhoods for accurate representation learning and effective anomaly detection.

Although important, selecting appropriate neighborhood in unsupervised graph anomaly detection poses the following significant challenges: First, the unsupervised setting prevents us from accessing the nodes’ ground-truth labels, which makes it lacks **anomaly-oriented guidance** for neighborhood selec-

tion [1], [12]. Second, the appropriate selection of neighborhoods and the high-quality node representation learning are **mutually dependent**. However, in the early stage of model training, both the reliability of neighborhoods and the effectiveness of representation learning are inadequate. Therefore, it is necessary to dynamically adjust the neighborhood selection strategy during the training process to enhance the quality of representation learning for anomaly distinguishing. Lastly, anomalies in the real world exhibit a wide range of diversity, with significant variations in abnormal patterns across different scenarios [12]. For instance, anomalies in social networks differ greatly from those in power networks. Consequently, the neighborhood selection process should **autonomously adapt** to accommodate different graphs.

To tackle the above challenges, in the paper, we propose a novel method that incorporates Reinforcement neighborhood selection for unsupervised graph ANomaly Detection (**RAND**). Reinforcement learning (RL) has demonstrated superior capability in adaptive learning in complex environments, which conveniently fulfills the demands of neighbor selection mentioned above. Specifically, RAND first extends the observed neighborhood to several noteworthy groups that may potentially benefit the anomaly detection, including 1-hop, 2-hop, high-order, and attribute-based neighborhoods. Then, RAND follows a reinforcement learning paradigm and treats the dynamic probability-based neighborhood selection as the *action*. The quality of selection is evaluated by the consistency of anomaly scores between central nodes and selected neighborhoods, which serves as the *reward* of the action. Subsequently, we further design an anomaly-aware message aggregator to fully leverage the information contained in the selected neighborhoods. Finally, RAND adopts reconstruction on the graph properties for model training and anomaly scoring in an unsupervised way. Extensive experiments conducted on both synthetic and real-world datasets demonstrate that RAND outperforms state-of-the-art models. The main contributions of this paper are organized as follows:

- We highlight the negative impact of anomaly nodes on neighborhood reliability, which is crucial for existing GNN-based graph anomaly detection methods.
- We propose RAND, a novel unsupervised graph anomaly detection method with reinforcement neighborhood selection and anomaly-aware message aggregation.
- We conduct extensive experiments on both widely-used synthetic and real-world datasets, which show that RAND outperforms the state-of-the-art methods.

II. RELATED WORKS

A. Unsupervised Graph Anomaly Detection

Traditional methods for unsupervised graph anomaly detection mainly focus on feature engineering or directly utilize instance attributes with shallow neural networks, e.g., SCAN [13] and MLPAE [14], regardless of the instance relationship modeling. Due to the success of GNNs, recent works propose to consider both the attribute and topology abnormal

patterns for anomaly mining and achieve state-of-the-art unsupervised graph anomaly detection performance. Among them, GAAN [15], ALARM [16], and AAGNN [17] try to improve the basic GNNs with enhanced unsupervised modeling training framework or representation learning procedure to strengthen the model sensitivity to anomalies. Another type of model adopts the deep graph autoencoder, e.g., GCNAE [18], Dominant [1], AnomalyDAE [19], and ComGA [20], based on the imbalanced number of nodes, the model learns the patterns of most normal nodes in the graph, thus the anomalies cannot be well reconstructed. The reconstruction error is used to evaluate whether a node is an anomaly. Recently, with the wide application of self-supervised learning in graphs, the methods based on graph contrastive learning become another category of models, i.e. CoLA [21], ANEMONE [22], SL-GAD [23], and Sub-CR [24], in which the magnitude of difference between positive and negative sample pairs designed for node identification is utilized to evaluate a node’s abnormality.

Most of the methods adopt vanilla GNN message passing to unsupervised learn node representations for anomaly distinguishing. Nevertheless, the neighborhood unreliable caused by anomaly nodes will bring noise information during the message passing and is yet to be addressed well.

B. Heterophily-based GNNs

For the powerful graph modeling ability, many GNNs have been proposed in recent years, such as GCN [6], GraphSage [7], and GAT [25]. They implicitly assume that connected nodes have similar behaviors (i.e. attributes or labels), which is typically called the *homophily assumption* [10], [26].

Recently, another type of graph modeling method called *heterophily-based GNN* is proposed for representation learning under heterophilic connections. These methods can be mainly divided into two kinds: (1) The first kind of approach is to mix multiple types of information to minimize the passing and proportion of heterophilic information, such as MixHop [27] and H2GCN [28]. (2) The second approach is to model homophilic information and heterophilic information separately by label information, and then using a fusion module to obtain the final node representation to alleviate information conflicts, such as LINKX [29] and GloGNN [30]. Recently, H2-FDetector [31] identifies the heterophilic connections with the supervision of labeled nodes for fraud detection. However, in unsupervised graph anomaly detection, it lacks effective label signals and particular anomaly modeling for them to work.

C. Deep Reinforcement Learning on Graphs

With the success of deep reinforcement learning (RL) in various research fields, such as robotics [32] and games [33]. Recently, some works have started to explore the application of RL for graph data mining [34].

The fundamental framework involves introducing deep RL to guide the learning process of GNNs, obtaining better node representations. Among them, Policy-GNN [35] chooses the suitable number of GNN aggregation layers for different nodes with RL. SUGAR [36] develops a reinforcement pooling

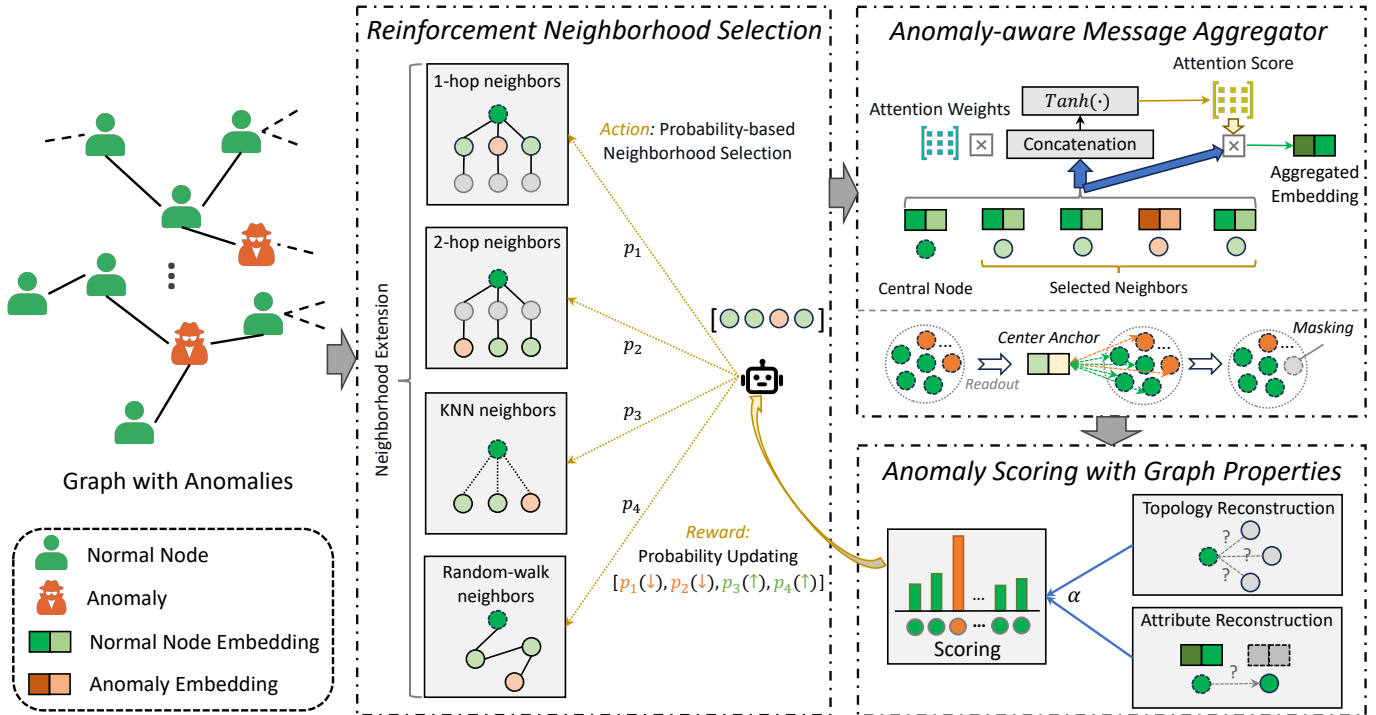


Fig. 2. Network architecture of the proposed RAND model. RAND is composed of three main parts: (i) the Reinforcement Neighborhood Selection which extends the neighborhood into multiple perspectives and adaptively selects neighboring nodes by multi-arm bandit reinforcement learning. (ii) the Anomaly-aware Message Aggregator which modifies the traditional message aggregator to make the learned representations more distinguishable for anomalies. (iii) the Anomalies Scoring with Graph Properties evaluates nodes' abnormality through their ability to reconstruct the graph properties, i.e. topology and attribute. The anomaly scoring consistency between central nodes and their selected neighborhoods will be regarded as the reward of the neighborhood selecting action.

to select striking subgraphs for better GNN interpretability. Then, ANS-GT [37] introduces multi-arm bandits based on the attention matrix for informative node sampling in graph transformer. CARE-GNN [11] proposes RL-improved GNN based on label-aware similarity measurement for dissimilar neighbor filtering in supervised fraud detection. However, it is difficult for them to effectively confront the challenges associated with unsupervised graph anomaly detection.

III. PROBLEM STATEMENT

Notations. Let $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ be an attributed graph with a node set $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ and the edge set \mathcal{E} , where $|\mathcal{V}| = n$. $\mathbf{A} \in \mathbb{R}^{n \times n}$ denotes the graph adjacency matrix, $A_{i,j} = 1$ indicates that there is an edge between node v_i and node v_j , and otherwise $A_{i,j} = 0$. $\mathbf{X} \in \mathbb{R}^{n \times d}$ denotes the node attribute matrix, the i -th row $\mathbf{x}_i = \mathbf{X}[i, :] \in \mathbb{R}^d$ indicates the attribute vector of v_i with d dimensional representation. \mathcal{N}_i is the neighbor set of a central node i in the graph \mathcal{G} .

Definition 1. Unsupervised Graph Anomaly Detection: Given an abnormal attributed graph $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ containing n node instances, and b of them are anomalies ($b \ll n$), whose attributes, connections or behaviors are different from most other normal nodes. The target of graph anomaly detection is to learn a model $\mathcal{F}(\cdot) : \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^n$ in the unsupervised manner that outputs anomaly score vector \mathbf{S} to measure the degree of abnormality of nodes, where a larger score means a higher abnormality.

IV. METHODOLOGY

A. Overall Framework of RAND

To select appropriate neighborhoods and fully leverage the information within the selected neighborhoods to learn anomaly-distinguishing representations, RAND consists of three main modules: Reinforcement Neighborhood Selection, Anomaly-aware Message Aggregator, and Anomaly Scoring with Graph Properties. First, *Reinforcement Neighborhood Selection* extends the concept of the directly connected neighborhood from various perspectives and adaptively selects suitable neighboring nodes with dynamic selection probabilities by multi-arm bandit reinforcement learning. Then, *Anomaly-aware Message Aggregator* modifies the traditional message aggregator with more distinguishing operators to make the aggregation better utilize the information of the selected neighboring nodes. Finally, *Anomaly Scoring with Graph Properties* evaluates the degree of abnormality of each node with the graph fundamental properties (i.e. topology and attribute). According to the scoring consistency between central nodes and their selected neighborhoods, RAND provides reward feedback to dynamically update the selection probabilities. The overall framework of RAND is illustrated in Figure 2 and the details of each part are introduced as follows.

B. Reinforcement Neighborhood Selection

Due to the messages passed into the center node are of significance for the quality of the learned representation [38],

the insight of reinforcement neighborhood selection is to mine appropriate neighboring nodes for central nodes rather than limit to the directly connected neighbors with potential noisy and polluted information. On account of the unsupervised setting, we cannot directly select neighboring nodes using label information. Coincidentally, inspired by [37] for neighborhood sampling, we found that this scenario satisfies the adversarial conditions to apply the multi-armed bandit RL for adaptively selecting suitable neighborhoods for each central node.

Applicability analysis: In multi-armed bandit RL, it highlights two conditions [37], [39]: 1) The impact of the action can be varied over time. 2) The rewards for the action are not independent random variables throughout training. Our scenario is consistent with its assumptions for: 1) It is intuitive that the influence of selected neighboring nodes on the anomaly detection performance can shift over time [37]. 2) The rewards based on the anomaly scores are linked to the model training process. These two properties satisfy the adversarial premise of utilizing multi-armed bandit RL.

Action: Considering different selection strategies can obtain neighborhoods with different preferences, we utilize multi-armed bandit to determine the preferences for the selection strategies. Let $\mathcal{W}^t = (w_1^t, \dots, w_K^t)$ be the adaptive weight vector in training iteration t , where w_k^t is the weight corresponding to the k -th selection strategy in iteration t , and K is the number of selection heuristics. Then, the weight vector \mathcal{W}^t will be mapped to a strategy-grained probability vector $\mathcal{P}^t = (p_1^t, \dots, p_K^t)$, where $p_k^t \in [p_{min}, 1]$, and $p_{min} \geq 0$ is a controllable constant that constrains the lower bound of probability, which is set to a default value of 0.05. RAND adaptively selects candidate nodes for aggregating based on the probability vector as the reinforcement action (bandit) in the environment of an abnormal graph. For each center node, we consider the fine-grained selection probability matrix $\mathcal{Q}^t \in \mathbb{R}^{K \times n}$. Specifically, $Q_{k,j}^t$ denotes the k -th selection strategy's preference on selecting node j in iteration t and $Q_{k,j}^t$ is normalized where $\sum_{j=1}^n Q_{k,j}^t = 1$. Note that RAND is a general framework and is not restricted to a certain set of selection heuristics. Here we adopt four representative selection heuristics [37]:

- **1-hop neighbors:** The directly connected neighbors sampled from the normalized adjacency matrix.
- **2-hop neighbors:** The indirectly connected nodes sampled from the power of normalized adjacency matrix.
- **KNN neighbors:** The nodes whose relationship is based on the cosine similarity of node attributes.
- **Random-walk neighbors:** The nodes with informative high-order patterns. The Personalized PageRank [40] is utilized for generating the random-walk weight.

Given strategy-grained probability p^t and fine-grained probability \mathcal{Q}^t , the final selecting probability for node i at training epoch t is:

$$\Phi_i^t = \sum_{k=1}^K p_k^t \cdot Q_{k,i}^t \quad (1)$$

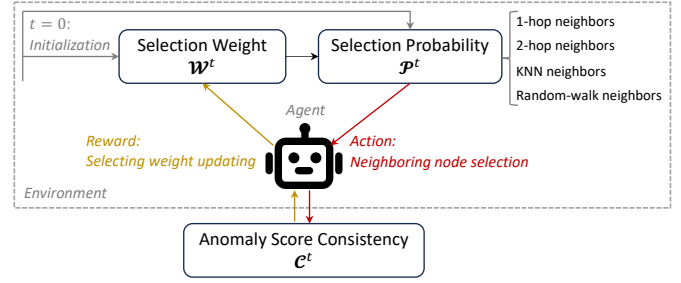


Fig. 3. The overall pipeline of the reinforcement neighborhood selection.

Reward: Nevertheless, the follow-up question is how to update the selecting probability. For a central node, a good selection should bring more consistent and homophilic information to it. According to this philosophy, we design the reward of the selection action based on the anomaly score consistency \mathcal{C}^t between the central node and its selected neighboring nodes, of which the anomaly scoring details will be introduced in the following subsection D.

Given the anomaly score y_i for the center node i , we first obtain the score vector $\mathbf{Y}_{\mathcal{N}_i}$ of its last selected neighboring nodes and calculate the score similarity distribution between i and \mathcal{N}_i as follows:

$$\mathcal{C}_i^t = \text{Softmax}\left(\frac{1}{\text{abs}(y_i - y_j) + \epsilon}\right), j \in \mathcal{N}_i. \quad (2)$$

Then, we give the reward based on the distribution similarity between \mathcal{C}_i^t and Φ_i^t , in which a higher similarity indicates the selecting way is more meaningful and thus higher reward will be given. Formally, the reward is calculated by a dot product scheme as follows:

$$r_k^t = \frac{1}{n} \sum_{i=1}^n \mathcal{C}_i^t \cdot \frac{p_k^t \cdot \mathcal{Q}_{k,\mathcal{N}_i}^t}{\Phi_{\mathcal{N}_i}^t} \quad (3)$$

where r_k^t is the reward vector of different selection strategies. Then the selection strategy weight can be updated as follows:

$$w_k^{t+1} = w_k^t e^{\left(\frac{p_{min}}{2}\right)\left(r_k + \frac{1}{p_k}\right)\delta_1 \sqrt{\frac{\ln(n/\delta_2)}{KT}}} \quad (4)$$

where T is the update interval, δ_1, δ_2 are controllable parameters. Note that we calculate the reward U after warm-up epochs U (we set $U = 3$ in our experiments) to ensure the model has the basic ability to distinguish anomalies.

Finally, the strategy-grained selection probability matrix for k -th strategy can be updated as follow:

$$p_k^{t+1} = (1 - K p_{min}) \cdot \frac{w_k^{t+1}}{SW^{t+1}} + p_{min}, \quad (5)$$

$$SW^{t+1} = \sum_{k=1}^K w_k^{t+1}. \quad (6)$$

We iterate the RAND training with the updated strategy-grained selection probability matrix p_k^{t+1} . Figure 3 illustrates the general process of reinforcement neighborhood selection.

C. Anomaly-aware Message Aggregator

After we obtain the selected neighborhood, our target is to learn the anomaly-distinguishing representation with the homophilic message under unsupervised settings. The insight of the designed Anomaly-aware Message Aggregator $f_{ama}(\cdot)$ is to: (i) Mask potential anomaly nodes, disallowing message passing for this subset of nodes to expand their representation dissimilarity with the majority of normal nodes; (ii) Modify the message aggregation function in previous works by using operations that provide more message differentiation capability to aggregate more homophilic patterns for the central node. With the above insights, $f_{ama}(\cdot)$ contains the two main stages: *center node masking* and *distinguishing message passing*.

Center Node Masking. We first project nodes' sparse features \mathbf{X} into the latent space embeddings \mathbf{E} with a two-layer MLP. Then, we mask potential anomaly nodes to prevent them from aggregating messages from a large number of normal nodes around them, as this could even help them to disguise themselves. We define a *center anchor* of the overall graph, those nodes that are farther from the center anchor are considered as the potential anomalies, which will be banned from message passing. Specifically, the representation of the center anchor \mathbf{E}_{ca} is obtained by a readout function:

$$\mathbf{E}_{ca} = \text{Readout}(\mathbf{E}) \quad (7)$$

where $\text{Readout}(\cdot)$ can be a kind of pooling operation (such as min, max, mean, and weighted pooling [21]), here we use the mean pooling for simplicity. Then, with a given mask rate $mr \geq 0$, we mask those nodes that are farthest from the center anchor \mathbf{E}_{ca} , of which the mask nodes can be calculated as:

$$M_{id} = \text{Top}_{(mr \cdot |\mathcal{V}|)}[\text{argsort}_{max}(\|\mathbf{E}_i - \mathbf{E}_{ca}\|_2^2, i \in \mathcal{V})] \quad (8)$$

where $\|\cdot\|_2^2$ is the Euclidean distance and M_{id} is the set of masked center nodes that will be skipped from the following message passing. For these masked nodes, their encodings will be identified with their original representation:

$$\mathbf{H}_i = \mathbf{E}_i, i \in M_{id} \quad (9)$$

where \mathbf{H}_i is the graph encoding of center node i .

Distinguishing Message Passing. The traditional message passing functions are insufficient to distinguish neighborhood messages [7], [25] since they consider all information from the neighborhood as homophilic messages and assimilate them. To address this, we modify the message passing function with fine-grained and more distinguishing operators. The distinguishing message passing contains the following main operations: (i) We calculate the attention score for the central node and its entire selected neighborhood to evaluate one neighbor's consistency not only with the central node but also with the other neighbors of the central node, rather than compute between the central node and each of its neighbors separately in previous research. (2) We extend the attention vector to an attention matrix to obtain the fine-grained dimensional attention score and utilize $\text{Tanh}(\cdot)$ as the activation function rather than $\text{Softmax}(\cdot)$ to further distinguishing the

negative messages for the center node. Formally, the process of message passing can be described as follows:

$$\mathbf{H}_i = (\mathbf{att}_{i,i} \odot \mathbf{E}_i + \sum_{j \in \mathcal{N}_i} \mathbf{att}_{i,j} \odot \mathbf{E}_j) \mathbf{W}_{mp} \quad (10)$$

$$\mathbf{att}_i = \text{Tanh}(\text{Concat}[\mathbf{E}_i, \{\mathbf{E}_j, j \in \mathcal{N}_i\}]) \mathbf{W}_{att} \quad (11)$$

where \mathbf{W}_{mp} and \mathbf{W}_{att} are trainable parameters, $\mathbf{att}_i \in \mathbb{R}^{|\mathcal{N}| \times d}$ is the *attention score matrix* of the center node i , and $\mathbf{att}_{i,j} \in \mathbb{R}^d$ is the score vector of node j to node i .

D. Anomaly Scoring with Graph Properties

The anomalies exhibit inconsistent patterns that deviate from the majority of the graph [1], [41]. Therefore, with the well-trained node representations, we conduct anomaly scoring to mining anomalies based on the reconstruction ability of nodes on the main graph properties, i.e. topology and attribute.

Topology Reconstruction. The topology is an important property of nodes in a graph, thus the reconstruction capability is a factor for potential anomalies mining. For the learned representation \mathbf{H} , the topology reconstruction function $g_t(\cdot)$ reconstructs the graph adjacency matrix as follows:

$$\hat{\mathbf{A}} = g_t(\mathbf{H}) = \mathbf{H} \cdot \mathbf{H}^T \quad (12)$$

$$\mathbf{D}_{topo} = \|\mathbf{A} - \hat{\mathbf{A}}\|_2^2 \quad (13)$$

where $g_t(\cdot)$ is a simple inner product operation between \mathbf{H} and its self-transposition \mathbf{H}^T to ensure the reconstruction efficiency, and $\mathbf{D}_{topo} \in \mathbb{R}^n$ is the topology reconstruction error vector of the nodes, of which each element $D_{topo,i}$ is viewed as a factor for topological anomaly scoring of node i .

Attribute Reconstruction. The attribute is also the key property of nodes that reflects the characteristics of the node itself. The attribute reconstruction capability is another evaluation factor to mine anomalies due to the inconsistent attribute is difficult to be reconstructed. Given the representation \mathbf{H} , an attribute reconstruction function $g_a(\cdot)$ is implemented to reconstruct the nodes' original attribute as follows:

$$\hat{\mathbf{X}} = g_a(\mathbf{H}, \mathbf{A}) \quad (14)$$

$$\mathbf{D}_{attr} = \|\mathbf{X} - \hat{\mathbf{X}}\|_2^2 \quad (15)$$

where $g_a(\cdot)$ can be a kind of graph encoder, such as MLP and GCN [6], and $\mathbf{D}_{attr} \in \mathbb{R}^n$ is the attribute reconstruction error vector of the nodes, of which each element $D_{attr,i}$ can be utilized for attributed anomaly scoring of node i .

Therefore, we utilize the above two aspects of graph property reconstruction errors for the final anomaly scoring. Here the scoring process of node i is calculated as:

$$\begin{aligned} \text{score}(i) &= (1 - \alpha) \cdot D_{topo,i} + \alpha \cdot D_{attr,i} \\ &= (1 - \alpha) \cdot \|\mathbf{a}_i - \hat{\mathbf{a}}_i\|_2^2 + \alpha \cdot \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 \end{aligned} \quad (16)$$

where α is a balanced parameter between the topology reconstruction and attribute reconstruction. Nodes with larger scores are more likely to be considered as anomalies.

E. Objective Function

To jointly learn the topology and attribute reconstruction errors for model training, the objective function of RAND can be formulated as:

$$\mathcal{L}_{topo} = \frac{1}{|\mathcal{V}|} \sum \|\mathbf{A} - \hat{\mathbf{A}}\|_2^2 \quad (17)$$

$$\mathcal{L}_{attr} = \frac{1}{|\mathcal{V}|} \sum \|\mathbf{X} - \hat{\mathbf{X}}\|_2^2 \quad (18)$$

$$\mathcal{L} = (1 - \alpha) \cdot \mathcal{L}_{topo} + \alpha \cdot \mathcal{L}_{attr} + \lambda \cdot \|\theta\|_2^2 \quad (19)$$

where θ is the parameter set of RAND, α is the balanced hyper-parameter the same as mentioned Eq.(16), and λ is the regularizer parameter.

F. Model Analysis

Convergence Analysis. Let r^* be the reward value of the optimal solution, r_k be the reward value of the k -th selection strategy, p_k^* be the selection probability that achieves the optimal value r^* for r_k .

Lemma 1. For any given selection strategy k , the expected value of w_k^t is w_k^* .

Prove: Consider Eq.(4), taking the expected value, we have:

$$\begin{aligned} \mathbb{E}[w_k^{t+1}] &= \mathbb{E}\left[w_k^t e^{\left(\frac{p_{min}}{2}\right)\left(r_k + \frac{1}{p_k}\right)\delta_1 \sqrt{\frac{\ln(n/\delta_2)}{KT}}}\right] \\ &= \mathbb{E}[w_k^t] \mathbb{E}\left[e^{\left(\frac{p_{min}}{2}\right)\left(r_k + \frac{1}{p_k}\right)\delta_1 \sqrt{\frac{\ln(n/\delta_2)}{KT}}}\right]. \end{aligned} \quad (20)$$

According to the Jensen's inequality for exponential functions and its convexity property, we have:

$$\begin{aligned} \mathbb{E}\left[e^{\left(\frac{p_{min}}{2}\right)\left(r_k + \frac{1}{p_k}\right)\delta_1 \sqrt{\frac{\ln(n/\delta_2)}{KT}}}\right] &\geq e^{\mathbb{E}\left[\left(\frac{p_{min}}{2}\right)\left(r_k + \frac{1}{p_k}\right)\delta_1 \sqrt{\frac{\ln(n/\delta_2)}{KT}}\right]} \\ &\geq e^{\left(\frac{p_{min}}{2}\right)\left(\mathbb{E}[r_k] + \frac{1}{\mathbb{E}[p_k]}\right)\delta_1 \sqrt{\frac{\ln(n/\delta_2)}{KT}}}. \end{aligned} \quad (21)$$

Substituting Eq.(21) into Eq.(20) along with the definition of p_k^* , we get:

$$\begin{aligned} \mathbb{E}[w_k^{t+1}] &\geq \mathbb{E}[w_k^t] e^{\left(\frac{p_{min}}{2}\right)\left(\mathbb{E}[r_k] + \frac{1}{\mathbb{E}[p_k]}\right)\delta_1 \sqrt{\frac{\ln(n/\delta_2)}{KT}}} \\ &= \mathbb{E}[w_k^t] \frac{w_k^*}{\mathbb{E}[w_k^t]} \end{aligned} \quad (22)$$

Therefore, we have $\mathbb{E}[w_k^{t+1}] \geq w_k^*$ that the expected value of w_k^t is w_k^* , which proves Lemma 1.

Lemma 2. For any given selection strategy k , the expected value of p_k^t approaches p_k^* .

Prove: Taking the expected value of the updating of p_k^t in Eq.(5) and Eq.(6), we have:

$$\begin{aligned} \mathbb{E}[p_k^{t+1}] &= (1 - Kp_{min}) \cdot \mathbb{E}\left[\frac{w_k^{t+1}}{SW^{t+1}}\right] + p_{min} \\ &= (1 - Kp_{min}) \cdot \frac{\mathbb{E}[w_k^{t+1}]}{\mathbb{E}[SW^{t+1}]} + p_{min} \end{aligned} \quad (23)$$

Then, since the expected value of w_k^t approaches w_k^* , we have:

$$\lim_{t \rightarrow \infty} \mathbb{E}[w_k^t] = w_k^* \quad (24)$$

$$\lim_{t \rightarrow \infty} \mathbb{E}[SW^t] = Kw_k^* \quad (25)$$

$$\begin{aligned} \lim_{t \rightarrow \infty} \mathbb{E}[p_k^{t+1}] &= (1 - Kp_{min}) \cdot \frac{\lim_{t \rightarrow \infty} \mathbb{E}[w_k^{t+1}]}{\lim_{t \rightarrow \infty} \mathbb{E}[SW^{t+1}]} + p_{min} \\ &= (1 - Kp_{min}) \cdot \frac{w_k^*}{Kw_k^*} + p_{min} \\ &= p_k^* \end{aligned} \quad (26)$$

Here, Lemma 2 is proved.

Convergence Guarantee. From the proven lemmas, we have: Firstly, considering the selection probability p_k^t of each selection strategy k , it can approach p_k^* . Therefore, in finite steps, each selection strategy will be sufficiently learned. Secondly, since the weight w_k^t of each selection strategy approaches the optimal value w_k^* , each selection strategy will be selected enough times in finite steps to obtain a reward value r_k that is close enough to the optimal value r^* . Thus, RAND can converge to the optimal solution within finite steps.

Complexity Analysis. The main complexity of RAND is from three parts: neighborhood selection, message aggregator, and anomaly scoring. Specifically, 1) the time complexity of the neighborhood selection is $\mathcal{O}(K \cdot n)$, where $K \ll n$. 2) the complexity of $f_{ama}(\cdot)$ is $\mathcal{O}(M \cdot n)$, where $M \ll n$ represents the sampled size of the central node which is set to 20 as default. 3) the anomaly scoring decoders $g_t(\cdot)$ and $g_a(\cdot)$ has the complexity of $\mathcal{O}(n^2)$ and $\mathcal{O}(n \cdot d)$, respectively. Therefore, the overall complexity of RAND is $\mathcal{O}((K + M) \cdot n + n^2)$.

V. EXPERIMENTS

A. Experimental Settings

Datasets: We adopt five widely-used datasets to verify the effectiveness of RAND, including three synthetic datasets: Cora, Citeseer, and Flickr, and two real-world datasets: Weibo, and Reddit. The dataset details are introduced as follows and the statistics of the datasets are shown in Table I.

Synthetic datasets: (1) **Cora**¹ [42] is a classical citation network consisting of 2,708 scientific publications (contains 150 injected anomalies) along with 5,429 links between them. (2) **Citeseer**¹ [42] is also a citation network consisting of 3,327 scientific publications (contains 150 injected anomalies) with 4,732 links. (3) **Flickr**² [43] is a social network dataset acquired from the image hosting and sharing website Flickr. In this dataset, 7,575 nodes denote the users (contains 450 injected anomalies), and 239,738 edges represent the following relationships between users.

Real-world datasets: (1) **Weibo**³ [44] is a user-posts-hashtag graph from Tencent-Weibo platform, which collects information from 8,405 users (contains 868 suspicious users).

¹<https://linqs.soe.ucsc.edu/datac>

²<http://socialcomputing.asu.edu/pages/datasets>

³https://github.com/zhao-tong/Graph-Anomaly-Loss/tree/master/data/weibo_s

The provided user-user graph is used, which connects users who used the same hashtag. (2) **Reddit**⁴ [45] is a user-subreddit graph from a social media platform, Reddit, which consists of one month of user posts on subreddits. The 1,000 most active subreddits and the 10,000 most active users (containing 366 banned users) are extracted. We convert it to a user-user graph for experiments, which connects users who have edited the same subreddits.

TABLE I
STATISTICS OF THE EXPERIMENTAL DATASETS.

Dataset	# nodes	# edges	# attributes	# anomalies
Cora	2,708	5,429	1,433	150
Citeseer	3,327	4,732	3,703	150
Flickr	7,575	239,738	12,407	450
Weibo	8,405	407,963	400	868
Reddit	10,000	20,744,044	64	366

Note that the anomaly generation in the synthetic datasets is following the anomaly injection method that has been widely used in previous research [21]–[24], which is to generate a combined set of anomalies for each dataset by perturbing topological structure and nodal attributes, respectively. The detailed description of the anomaly injection is as follows.

Injection of topological anomalies. To obtain topological anomalies, the topological structure of networks is perturbed by generating small cliques composed of nodes that were originally not related. The insight is that in a small clique, a small group of nodes are significantly more interconnected with each other than the average, which can be considered a typical situation of topological anomalies in real-world graphs [21]. When generating a clique with the clique size p and the number of cliques q , we randomly select p nodes from the set of nodes \mathcal{V} and connect them fully. This implies that all the selected p nodes are considered topological anomalies. To generate q cliques, we repeat this process q times. This results in a total of $p \times q$ topological anomalies. Following previous works, the value of p is fixed as 15 and the value of q is set to 5, 5, 15 for Cora, Citeseer, and Flickr, respectively.

Injection of attributed anomalies. We inject attributed anomalies by disturbing the attribute of nodes, which is introduced in [46]. To generate an attributed anomaly, a node v_i is randomly selected as the target, and then another k nodes (v_1^c, \dots, v_k^c) are sampled as a candidate set \mathcal{V}^c . Next, we compute the Euclidean distance between the attribute vector \mathbf{x}_c of each $v^c \in \mathcal{V}^c$ and the attribute vector \mathbf{x}_i of v_i . We then select the node $v_j^c \in \mathcal{V}^c$ that has the largest Euclidean distance to v_i and change \mathbf{x}_i to \mathbf{x}_j^c . Following the previous works, the value of k is set to 50.

Baselines: We compare the proposed RAND with seventeen representative state-of-the-art unsupervised graph anomaly detection models, including five main groups:

i) **Shallow Detection Models:** (1) SCAN [13] is a clustering algorithm, which clusters vertices based on structural similar-

ity to detect anomalies. (2) MLPAE [14] utilizes autoencoders onto both anomalous and benign data with a nonlinear MLP.

ii) **Improved GNN-based Models:** (1) GAAN [15] is a generative adversarial training framework with a GNN encoder to obtain real and fake node representations and a discriminator to recognize whether two connected nodes are from the real or fake graph. (2) ALARM [16] is a multi-view representation learning framework with multiple GNN encoders and a well-designed fusion operator between them. (3) AAGNN [17] is an enhanced GNN, which utilizes subtractive aggregation to represent each node as the deviation from its neighbors.

iii) **Graph AutoEncoder-based Models:** (1) GCNAE [18] is the GCN-based variational graph autoencoder and utilizes the reconstruction loss for anomaly detection. (2) Dominant [1] is a deep graph autoencoder-based method with a shared encoder. It detects the anomalies by computing the weighted sum of reconstruction error terms. (3) AnomalyDAE [19] is a dual graph autoencoder method with asymmetrical cross-modality interactions between structure and attribute. (4) ComGA [20] is a community-aware graph anomaly detection framework with a designed tailored deep GCN.

iv) **Graph Contrastive Learning Models:** (1) CoLA [21] is a graph contrastive learning method that detects anomalies by evaluating the agreement between each node and its sampled subgraph. (2) ANEMONE [22] is a multi-scale contrastive learning method, which captures the anomaly pattern by learning the agreements between nodes at both patch and context levels. (3) SL-GAD [23] is self-supervised trained with generative and multi-view contrastive perspectives concurrently. (4) Sub-CR [24] employs the graph diffusion-based multi-view contrastive learning along with attribute reconstruction.

v) **Heterophily GNN-based Models:** (1) MixHop [27] is a graph convolutional network with the mixed aggregation of multi-hop neighbors during one message passing. (2) H2GCN [28] is a heterophily graph model by the separate encoding of ego&neighbor-embedding with higher-order neighbors. (3) LINKX [29] is an MLP-based model with separate modeling for topology and features. (4) GloGNN [30] is a method that considers both homophily and heterophily with the combination of low-pass and high-pass filters. We compare their autoencoder architectures for unsupervised learning.

Evaluation Metrics: We evaluate the models with AUC and AP, the widely-adopted metrics in previous anomaly detection works [1], [47], to evaluate the detection performance. The higher AUC and AP values indicate better anomaly detection performance. Note that we run all the experiments *five* times with different random seeds and report the average results with standard deviation to prevent extreme cases.

Hyper-parameter Settings: The embedding size is fixed to 64 and the embedding parameters are initialized with the Xavier method. The loss function is optimized with Adam optimizer. The learning rate of RAND is searched from $\{5 \times 10^{-2}, 1 \times 10^{-2}, 5 \times 10^{-3}, 1 \times 10^{-3}\}$. For all baselines, we retain the parameter settings in their corresponding papers to keep the comparison fair. All experiments are conducted on the Centos system equipped with NVIDIA RTX-3090 GPUs.

⁴<http://files.pushshift.io/reddit>

TABLE II

UNSUPERVISED GRAPH ANOMALY DETECTION COMPARISON RESULTS ON AUC AND AP METRICS (MEAN \pm STANDARD DEVIATION IN PERCENTAGE OVER *five* TRIAL RUNS). THE BEST AND SECOND-BEST RESULTS IN EACH COLUMN ARE HIGHLIGHTED IN **BOLD FONT** AND UNDERLINED.

Model	Synthetic Datasets						Real-world Datasets			
	Cora		Citeseer		Flickr		Weibo		Reddit	
	AUC	AP	AUC	AP	AUC	AP	AUC	AP	AUC	AP
SCAN	0.6604 \pm 0.0163	0.0859 \pm 0.0044	0.6689 \pm 0.0136	0.0731 \pm 0.0032	0.6503 \pm 0.0120	0.3035 \pm 0.0227	0.7011 \pm 0.0000	0.1855 \pm 0.0000	0.4978 \pm 0.0000	0.0364 \pm 0.0000
MLPAE	0.7565 \pm 0.0108	0.3528 \pm 0.0188	0.7396 \pm 0.0112	0.3124 \pm 0.0169	0.7466 \pm 0.0041	0.3484 \pm 0.0087	0.8946 \pm 0.0028	0.6696 \pm 0.0102	0.5108 \pm 0.0310	0.0359 \pm 0.0029
GAAN	0.7917 \pm 0.0118	0.3271 \pm 0.0124	0.8066 \pm 0.0036	0.3495 \pm 0.0101	0.7463 \pm 0.0041	0.3552 \pm 0.0119	0.9249 \pm 0.0000	0.8104 \pm 0.0000	0.5683 \pm 0.0001	0.0493 \pm 0.0001
ALARM	0.8271 \pm 0.0223	0.2503 \pm 0.0379	0.8325 \pm 0.0121	0.3027 \pm 0.0559	0.6085 \pm 0.0034	0.0726 \pm 0.0013	0.9226 \pm 0.0000	0.8071 \pm 0.0000	0.5644 \pm 0.0003	0.0466 \pm 0.0001
AAGNN	0.7590 \pm 0.0056	0.3744 \pm 0.0171	0.7202 \pm 0.0140	0.2345 \pm 0.0303	0.7454 \pm 0.0033	0.3506 \pm 0.0087	0.8066 \pm 0.0027	0.6679 \pm 0.0009	0.5442 \pm 0.0299	0.0400 \pm 0.0030
GCNAE	0.7959 \pm 0.0104	0.3544 \pm 0.0350	0.7678 \pm 0.0114	0.3321 \pm 0.0243	0.7471 \pm 0.0056	0.3359 \pm 0.0125	0.8449 \pm 0.0032	0.5650 \pm 0.0030	0.5037 \pm 0.0015	0.0346 \pm 0.0001
Dominant	0.8773 \pm 0.0134	0.3090 \pm 0.0438	0.8523 \pm 0.0051	0.3999 \pm 0.0092	0.6129 \pm 0.0035	0.0734 \pm 0.0013	0.8423 \pm 0.0117	0.6163 \pm 0.0237	0.5752 \pm 0.0056	0.0570 \pm 0.0019
AnomalyDAE	0.8594 \pm 0.0068	0.3586 \pm 0.0148	0.8092 \pm 0.0059	0.3487 \pm 0.0103	0.7418 \pm 0.0051	0.3635 \pm 0.0107	0.8881 \pm 0.0165	0.6681 \pm 0.0874	0.4315 \pm 0.0001	0.0319 \pm 0.0000
ComGA	0.7382 \pm 0.0162	0.1539 \pm 0.0242	0.7004 \pm 0.0150	0.0921 \pm 0.0085	0.6658 \pm 0.0033	0.1896 \pm 0.0120	0.9248 \pm 0.0006	0.8097 \pm 0.0009	0.4317 \pm 0.0001	0.0320 \pm 0.0001
MixHop	0.7796 \pm 0.0107	0.2412 \pm 0.0300	0.7401 \pm 0.0122	0.3146 \pm 0.0195	0.7447 \pm 0.0060	0.3517 \pm 0.0229	0.8612 \pm 0.0018	0.6278 \pm 0.0061	0.5400 \pm 0.0175	0.0398 \pm 0.0028
H2GCN	0.7827 \pm 0.0104	0.3479 \pm 0.0285	0.7361 \pm 0.0169	0.3064 \pm 0.0139	0.7463 \pm 0.0042	0.3526 \pm 0.0135	0.8546 \pm 0.0020	0.5604 \pm 0.0088	0.5476 \pm 0.0113	0.0401 \pm 0.0010
LINKX	0.7601 \pm 0.0098	0.3605 \pm 0.0242	0.7416 \pm 0.0117	0.3187 \pm 0.0180	0.7466 \pm 0.0042	0.3636 \pm 0.0065	0.8018 \pm 0.0094	0.2822 \pm 0.0088	0.5576 \pm 0.0162	0.0421 \pm 0.0020
GloGNN	0.7563 \pm 0.0083	0.3470 \pm 0.0269	0.7419 \pm 0.0124	0.3214 \pm 0.0220	0.7430 \pm 0.0013	0.3583 \pm 0.0054	0.9128 \pm 0.0251	0.7465 \pm 0.1280	0.5436 \pm 0.0364	0.0468 \pm 0.0085
CoLA	0.8887 \pm 0.0147	0.4804 \pm 0.0494	0.8211 \pm 0.0118	0.2302 \pm 0.0265	0.5612 \pm 0.0224	0.0694 \pm 0.0047	0.4842 \pm 0.0238	0.1006 \pm 0.0155	0.5149 \pm 0.0233	0.0393 \pm 0.0020
ANEMONE	0.8966 \pm 0.0119	0.5425 \pm 0.0489	0.8513 \pm 0.0159	0.3229 \pm 0.0212	0.5579 \pm 0.0271	0.0712 \pm 0.0062	0.3607 \pm 0.0120	0.0863 \pm 0.0147	0.4952 \pm 0.0188	0.0387 \pm 0.0026
SL-GAD	0.8159 \pm 0.0238	0.3361 \pm 0.0337	0.7287 \pm 0.0201	0.2122 \pm 0.0273	0.7240 \pm 0.0087	0.3146 \pm 0.0123	0.4298 \pm 0.0073	0.0899 \pm 0.0036	0.5488 \pm 0.0142	0.0424 \pm 0.0032
Sub-CR	0.8968 \pm 0.0118	0.4771 \pm 0.0102	0.9060 \pm 0.0095	0.4751 \pm 0.0254	0.7423 \pm 0.0038	0.3611 \pm 0.0145	0.6404 \pm 0.0070	0.4900 \pm 0.0103	0.5327 \pm 0.0156	0.0379 \pm 0.0013
RAND (ours)	0.9689 \pm 0.0013	<u>0.4981</u> \pm 0.0196	0.9695 \pm 0.0047	0.5451 \pm 0.0110	0.7625 \pm 0.0054	0.3673 \pm 0.0096	0.9805 \pm 0.0011	0.8639 \pm 0.0128	0.6022 \pm 0.0079	0.0629 \pm 0.0027
Improv. (%)	+8.04%	—	+7.01%	+14.73%	+2.06%	+1.02%	+6.01%	+6.60%	+4.69%	+10.35%

B. Main Results

In this subsection, we compare RAND with seventeen state-of-the-art baselines. The comparison results are reported in Table II. From these results, we have the following observation:

- **RAND can achieve significant improvement over state-of-the-art models on both synthetic and real-world datasets.** From the table, we observe that RAND achieves the best AUC metric in all datasets and the optimal AP metric in 4 out of the 5 datasets. Specifically, for the AUC metric, RAND outperforms the best baseline by 8.04%, 7.01%, 2.06%, 6.01%, and 4.69% on Cora, Citeseer, Flickr, Weibo, and Reddit, respectively, which brings the gains of **5.56%** on average. These results verify the effectiveness of RAND in distinguishing representation learning and better anomaly detection performance.
- **The performance difference of baseline models on real-world datasets is more pronounced compared to that on synthetic datasets,** which exhibits the same observation with [12] and shows that detecting real-world anomalies is more challenging. This may be because anomalies in the real world are more diverse, while the patterns of injected anomalies in the synthetic datasets are relatively uniform due to the fixed injection method. Therefore, in the face of diverse real-world anomaly patterns, the design of adaptive solutions is more appropriate.
- **Graph contrastive learning methods perform well on synthetic datasets but relatively poorly on real-world datasets.** This suggests that using sole contrastive schemes with simple vanilla GNN encoders cannot fully explore the diverse abnormal patterns in the real world. The heterophilic connections brought by real-world anomalies are more diverse, and encoding nodes indiscriminately through the vanilla GNN would further weaken the sample quality of contrastive learning.

TABLE III

ABLATION STUDY RESULTS COMPARED WITH TWO RAND VARIANTS.

Variant	Flick	Weibo	Reddit
RAND	0.7625 \pm 0.0054	0.9805 \pm 0.0011	0.6022 \pm 0.0079
RAND- <i>w/o</i> TR	0.7475 \pm 0.0041	0.8698 \pm 0.0093	0.5801 \pm 0.0096
RAND- <i>w/o</i> AR	0.5937 \pm 0.0252	0.9725 \pm 0.0015	0.6019 \pm 0.0052

C. Ablation Study

To verify the effectiveness of the design of RAND, we conduct various ablation studies on RAND. First, we conduct the component ablation study on RAND with its two component variants: (1) RAND-*w/o* TR removes the topology reconstruction module for both training and anomaly scoring. (2) RAND-*w/o* AR removes the attribute reconstruction for both training and anomaly scoring. The study results can be found in Table III, from which we have the following observations: First, compared to *w/o* TR and *w/o* AR variants, RAND gains significant improvements which proves the necessity and effectiveness of these modules. Second, the impact of *w/o* TR and *w/o* AR varies from the datasets, which is related to the diversity of anomalies on different datasets.

Furthermore, we conduct the ablation study on the message aggregator of RAND by comparing it with GraphSAGE [7] and GAT [25]. Figure 4-(a) illustrates the study results. From the results, we can find that the designed message aggregator achieves the best detection performance. Furthermore, GAT performs relatively better results than GraphSAGE, the possible reason is that the attention mechanism in GAT helps it discriminate the anomalies. In Figure 5 we further conduct the node representation visualization through the widely-used T-SNE [48] method, and from it we can find that the aggregator of AMPN has a significantly greater discriminatory ability for anomalies. Then, in Figure 4-(b), we study the anomaly mask

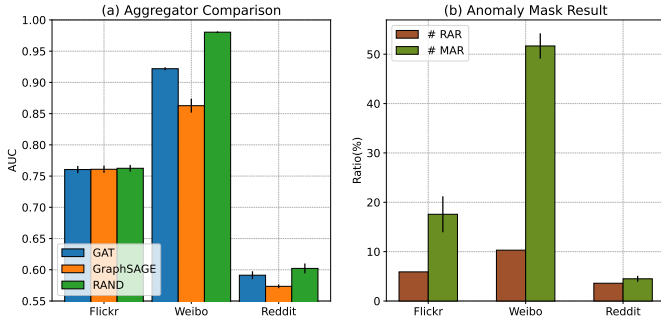


Fig. 4. (a): Ablation study to analyze the effect of designed message aggregator of RAND. (b): Study of the anomaly mask result in Eq.(8), where **#RAR** denotes the **raw anomaly ratio** of each dataset and **#MAR** denotes the **masked anomaly ratio** by RAND with the mask rate of 3%.

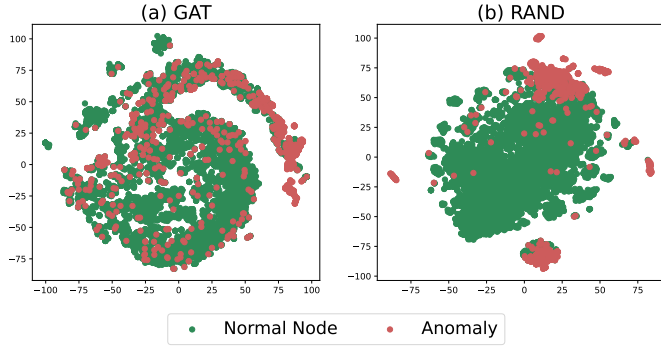


Fig. 5. The visualization of node representations from the trained models. The message passing of our proposed RAND can learn more distinguishable embeddings between normal nodes and anomalies in the latent space.

results of RAND under the set mask rate of 3%. We can observe that by using only a 3% masking rate, it is possible to mask the ratio of anomalies that are much higher than the raw anomaly rate in the dataset (in Weibo, the masked anomaly rate even exceeds 50%), which helps RAND to widen the gap between representations of normal nodes and anomalies.

D. Case Study

We further conduct the case study of the dynamic changing of the selection probabilities of different strategies during the training process, of which the results are shown in Figure 6.

From the table, we can find that RAND has different preferences for different types of neighbors on different datasets, which is due to the diverse characteristics of the graphs and their anomalies. For example, on Flickr, RAND prefers to aggregate KNN neighbors, which indicates the structurally defined neighbors are not so effective for it. This conclusion is consistent with the result that *w/o AR* has a greater impact than *w/o TR* in the ablation study on Flickr, indicating that the adaptive neighbor selection we designed is meaningful.

E. Parameter Study

Effectiveness of the balanced parameter: We investigate the effectiveness of the changing of the balanced parameter α on RAND from 0.0 to 1.0 with a step of 0.1. The study

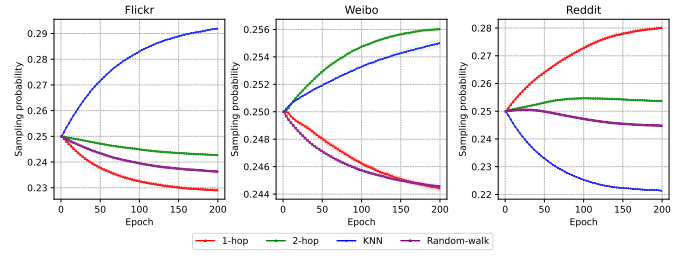


Fig. 6. Case study of the dynamic selection probability changing during the training iterations.

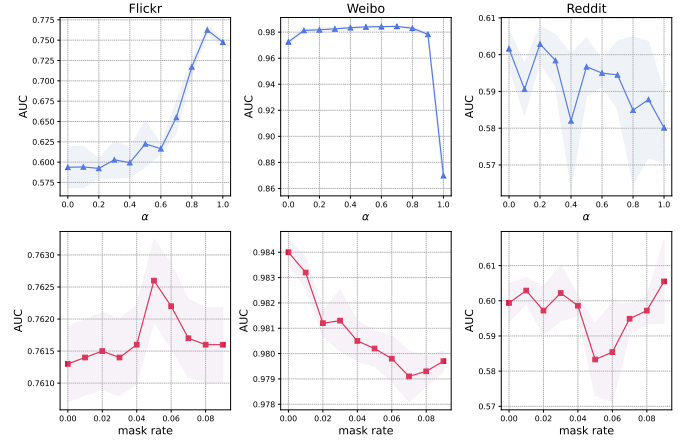


Fig. 7. Parameter study results on the balanced hyper-parameter α and mask rate of Flickr, Weibo, and Reddit datasets on the AUC metric.

results are shown in the first row of Figure 7. From the results, we observe that a suitable value of α is important for the model performance, which varies from different datasets, e.g. a related large value of α is better for Flickr while a small value is better for Reddit.

Effectiveness of the mask rate: We also study the effect of the mask rate by varying it from the range of [0, 0.09]. The study results are shown in the second row of Figure 7. From the results, we can find that the suitable mask rate varies from different datasets, and adding a few node masking can generally bring performance gains to the model but without positive gains on the Weibo dataset.

VI. CONCLUSION

In this paper, we propose RAND, a novel method for unsupervised graph anomaly detection with reinforcement neighborhood selection. RAND first extends the candidate neighborhoods and then adaptively selects proper neighbors by reinforcement learning. Furthermore, RAND introduces a more anomaly-distinguishing message aggregator for passing more consistent messages for central nodes in an unsupervised manner. Extensive experiments on both synthetic and real-world datasets illustrate that our proposed RAND achieves state-of-the-art graph anomaly detection performance. In future works, we will explore more efficient neighborhood selection schemes to make the model more suitable for anomaly detection on large-scale graphs.

REFERENCES

- [1] K. Ding, J. Li, R. Bhanushali, and H. Liu, "Deep anomaly detection on attributed networks," in *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 2019, pp. 594–602.
- [2] F. Liu, X. Ma, J. Wu, J. Yang, S. Xue, A. Beheshti, C. Zhou, H. Peng, Q. Z. Sheng, and C. C. Aggarwal, "Dagad: Data augmentation for graph anomaly detection," in *2022 IEEE International Conference on Data Mining (ICDM)*, 2022, pp. 259–268.
- [3] D. Wang, J. Lin, P. Cui, Q. Jia, Z. Wang, Y. Fang, Q. Yu, J. Zhou, S. Yang, and Y. Qi, "A semi-supervised graph attentive network for financial fraud detection," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 598–607.
- [4] T. Sun, Z. Qian, S. Dong, P. Li, and Q. Zhu, "Rumor detection on social media with graph adversarial contrastive learning," in *Proceedings of the ACM Web Conference*, 2022, pp. 2789–2797.
- [5] X. Zhou, W. Liang, W. Li, K. Yan, S. Shimizu, I. Kevin, and K. Wang, "Hierarchical adversarial attacks against graph-neural-network-based iot network intrusion detection system," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9310–9319, 2021.
- [6] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.
- [7] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.
- [8] X. Ma, J. Wu, S. Xue, J. Yang, C. Zhou, Q. Z. Sheng, H. Xiong, and L. Akoglu, "A comprehensive survey on graph anomaly detection with deep learning," *IEEE Transactions on Knowledge and Data Engineering*, 2021.
- [9] H. Kim, B. S. Lee, W.-Y. Shin, and S. Lim, "Graph anomaly detection with graph neural networks: Current status and challenges," *IEEE Access*, 2022.
- [10] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual review of sociology*, pp. 415–444, 2001.
- [11] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, "Enhancing graph neural network-based fraud detectors against camouflaged fraudsters," in *CIKM*, 2020, pp. 315–324.
- [12] K. Liu, Y. Dou, Y. Zhao, X. Ding, X. Hu, R. Zhang, K. Ding, C. Chen, H. Peng, K. Shu *et al.*, "Bond: Benchmarking unsupervised outlier node detection on static attributed graphs," in *NIPS Datasets and Benchmarks Track*, 2022.
- [13] X. Xu, N. Yuruk, Z. Feng, and T. A. Schweiger, "Scan: a structural clustering algorithm for networks," in *KDD*, 2007, pp. 824–833.
- [14] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *MLSDA 2nd workshop on machine learning for sensory data analysis*, 2014, pp. 4–11.
- [15] Z. Chen, B. Liu, M. Wang, P. Dai, J. Lv, and L. Bo, "Generative adversarial attributed network anomaly detection," in *CIKM*, 2020, pp. 1989–1992.
- [16] Z. Peng, M. Luo, J. Li, L. Xue, and Q. Zheng, "A deep multi-view framework for anomaly detection on attributed networks," *IEEE Transactions on Knowledge and Data Engineering*, 2020.
- [17] S. Zhou, Q. Tan, Z. Xu, X. Huang, and F.-I. Chung, "Subtractive aggregation for attributed network anomaly detection," in *CIKM*, 2021, pp. 3672–3676.
- [18] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.
- [19] H. Fan, F. Zhang, and Z. Li, "Anomalydae: Dual autoencoder for anomaly detection on attributed networks," in *ICASSP*. IEEE, 2020, pp. 5685–5689.
- [20] X. Luo, J. Wu, A. Beheshti, J. Yang, X. Zhang, Y. Wang, and S. Xue, "Comga: Community-aware attributed graph anomaly detection," in *WSDM*, 2022, pp. 657–665.
- [21] Y. Liu, Z. Li, S. Pan, C. Gong, C. Zhou, and G. Karypis, "Anomaly detection on attributed networks via contrastive self-supervised learning," *IEEE transactions on neural networks and learning systems*, 2021.
- [22] M. Jin, Y. Liu, Y. Zheng, L. Chi, Y.-F. Li, and S. Pan, "Anemone: Graph anomaly detection with multi-scale contrastive learning," in *CIKM*, 2021, pp. 3122–3126.
- [23] Y. Zheng, M. Jin, Y. Liu, L. Chi, K. T. Phan, and Y.-P. P. Chen, "Generative and contrastive self-supervised learning for graph anomaly detection," *IEEE Transactions on Knowledge and Data Engineering*, 2021.
- [24] J. Zhang, S. Wang, and S. Chen, "Reconstruction enhanced multi-view contrastive learning for anomaly detection on attributed networks," in *IJCAI*, 2022, pp. 2376–2382.
- [25] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.
- [26] T. Wang, D. Jin, R. Wang, D. He, and Y. Huang, "Powerful graph convolutional networks with adaptive propagation mechanism for homophily and heterophily," in *AAAI*, vol. 36, no. 4, 2022, pp. 4210–4218.
- [27] S. Abu-El-Haija, B. Perozzi, A. Kapoor, N. Alipourfard, K. Lerman, H. Harutyunyan, G. Ver Steeg, and A. Galstyan, "Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing," in *ICML*. PMLR, 2019, pp. 21–29.
- [28] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, and D. Koutra, "Beyond homophily in graph neural networks: Current limitations and effective designs," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7793–7804, 2020.
- [29] D. Lim, F. Hohne, X. Li, S. L. Huang, V. Gupta, O. Bhalerao, and S. N. Lim, "Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods," *Advances in Neural Information Processing Systems*, vol. 34, pp. 20887–20902, 2021.
- [30] X. Li, R. Zhu, Y. Cheng, C. Shan, S. Luo, D. Li, and W. Qian, "Finding global homophily in graph neural networks when meeting heterophily," in *ICML*. PMLR, 2022, pp. 13242–13256.
- [31] F. Shi, Y. Cao, Y. Shang, Y. Zhou, C. Zhou, and J. Wu, "H2-fdetector: a gnn-based fraud detector with homophilic and heterophilic connections," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 1486–1494.
- [32] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.
- [33] M. Lanctot, V. Zambaldi, A. Gruslys, A. Lazaridou, K. Tuyls, J. Pérolat, D. Silver, and T. Graepel, "A unified game-theoretic approach to multi-agent reinforcement learning," *Advances in neural information processing systems*, vol. 30, 2017.
- [34] M. Nie, D. Chen, and D. Wang, "Reinforcement learning on graphs: A survey," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2023.
- [35] K.-H. Lai, D. Zha, K. Zhou, and X. Hu, "Policy-gnn: Aggregation optimization for graph neural networks," in *KDD*, 2020, pp. 461–471.
- [36] Q. Sun, J. Li, H. Peng, J. Wu, Y. Ning, P. S. Yu, and L. He, "Sugar: Subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism," in *Proceedings of the Web Conference 2021*, 2021, pp. 2081–2091.
- [37] Z. ZHANG, Q. Liu, Q. Hu, and C.-K. Lee, "Hierarchical graph transformer with adaptive node sampling," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35, 2022, pp. 21171–21183.
- [38] E. Dai, W. Jin, H. Liu, and S. Wang, "Towards robust graph neural networks for noisy graphs with sparse labels," in *WSDM*, 2022, pp. 181–191.
- [39] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The non-stochastic multiarmed bandit problem," *SIAM journal on computing*, vol. 32, no. 1, pp. 48–77, 2002.
- [40] H. Wang, Z. Wei, J. Gan, S. Wang, and Z. Huang, "Personalized pagerank to a target node, revisited," in *KDD*, 2020, pp. 657–667.
- [41] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data mining and knowledge discovery*, vol. 29, no. 3, pp. 626–688, 2015.
- [42] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, pp. 93–93, 2008.
- [43] L. Tang and H. Liu, "Relational learning via latent social dimensions," in *KDD*, 2009, pp. 817–826.
- [44] T. Zhao, C. Deng, K. Yu, T. Jiang, D. Wang, and M. Jiang, "Error-bounded graph anomaly loss for gnns," in *CIKM*, 2020, pp. 1873–1882.
- [45] S. Kumar, X. Zhang, and J. Leskovec, "Predicting dynamic embedding trajectory in temporal interaction networks," in *KDD*, 2019, pp. 1269–1278.
- [46] X. Song, M. Wu, C. Jermaine, and S. Ranka, "Conditional anomaly detection," *IEEE Transactions on knowledge and Data Engineering*, vol. 19, no. 5, pp. 631–645, 2007.
- [47] J. Tang, J. Li, Z. Gao, and J. Li, "Rethinking graph neural networks for anomaly detection," in *ICML*, 2022.
- [48] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.