Graph Anomaly Detection via Multi-Scale Contrastive Learning Networks with Augmented View

Jingcan Duan[1], Siwei Wang[1], Pei Zhang[1], En Zhu[1]*, Jingtao Hu[1], Hu Jin[1], Yue Liu[1], Zhibin Dong[1]

[1]College of Computer, National University of Defense Technology, Changsha, China
{jingcan_duan, yueliu19990731}@163.com, {wangsiwei13, zhangpei, enzhu, hujingtao17, jinhu, dzb20}@nudt.edu.cn

Abstract

Graph anomaly detection (GAD) is a vital task in graph-based machine learning and has been widely applied in many real-world applications. The primary goal of GAD is to capture anomalous nodes from graph datasets, which evidently deviate from the majority of nodes. Recent methods have paid attention to various scales of contrastive strategies for GAD, i.e., node-subgraph and node-node contrasts. However, they neglect the subgraph-subgraph comparison information which the normal and abnormal subgraph pairs behave differently in terms of embeddings and structures in GAD, resulting in sub-optimal task performance. In this paper, we fulfill the above idea in the proposed multi-view multi-scale contrastive learning framework with subgraph-subgraph contrast for the first practice. To be specific, we regard the original input graph as the first view and generate the second view by graph augmentation with edge modifications. With the guidance of maximizing the similarity of the subgraph pairs, the proposed subgraph-subgraph contrast contributes to more robust subgraph embeddings despite of the structure variation. Moreover, the introduced subgraph-subgraph contrast cooperates well with the widely-adopted node-subgraph and node-node contrastive counterparts for mutual GAD performance promotions. Besides, we also conduct sufficient experiments to investigate the impact of different graph augmentation approaches on detection performance. The comprehensive experimental results well demonstrate the superiority of our method compared with the state-of-the-art approaches and the effectiveness of the multi-view subgraph pair contrastive strategy for the GAD task. The source code is released at https://github.com/FelixDJC/GRADATE.

Introduction

Over the past few years, graph-based machine learning has attracted great attention (Wu et al. 2020; Liu et al. 2022c). As a representative task in graph learning, graph anomaly detection, which intends to discern the anomalies from the majority of nodes, is becoming an increasingly glamorous application for researchers (Ma et al. 2021). Due to its vital value in the prevention of harmful events, GAD has already been widely used in many fields, e.g., misinformation detection (Wu et al. 2019), financial fraud detection (Huang et al. 2018), network intrusion detection (Garcia-Teodoro et al. 2009), etc. Unlike the data from other anomaly detection fields (Cheng et al. 2021a,b; Hu et al. 2022), graph data includes node features and graph structure. The mismatch between these two types of information produces two typical anomalous nodes, i. e. feature anomalies and structure anomalies (Liu et al. 2021). The former refers to nodes that differ from their neighbors in terms of features, while the latter refers to a group of nodes that are dissimilar but closely connected.

To detect these two categories of anomalies, many previous methods have made great efforts and achieved impressive results. LOF (Breunig et al. 2000) obtains anomalous information about a node by comparing it with its contextual nodes in terms of features. SCAN (Xu et al. 2007) accomplishes the GAD task from network structure. By utilizing such two types of information, ANOMALOUS (Peng et al. 2018) detects anomalies based on CUR decomposition and residual analysis. (Müller et al. 2013; Perozzi et al. 2014) perform feature subspace selection and find anomalous nodes in the subspace. The methods above rely on specific domain knowledge and cannot mine deep non-linear information in graph datasets. These make it difficult for them to improve the detection performance further.

Thanks to the powerful graph information acquisition capability, graph convolutional networks (GCN) (Kipf and Welling 2017) have recently achieved excellent performance in many graph-data tasks. It is naturally applied to detect anomalies in graph. The pioneering work DOMINANT (Ding et al. 2019) introduces GCN to accomplish the task for the first practice. Specifically, DOMINANT compares the restructured feature and adjacency matrices with the original input matrices. The nodes with more considerable variations have a higher probability of being anomalies. Although it performs well and is simple to implement, some anomalous information will be ignored. GCN generates node representations by aggregating the information from neighborhoods, which will make anomalies more indistinguishable (Tang et al. 2022). Based on the contrastive learning paradigm, CoLA (Liu et al. 2021) detects anomalies by calculating the relationship between nodes and their neighborhoods. This approach digs the local feature and structure information around the nodes. Meanwhile, it masks the features of the target node during training, which allevi-

*Corresponding author
Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

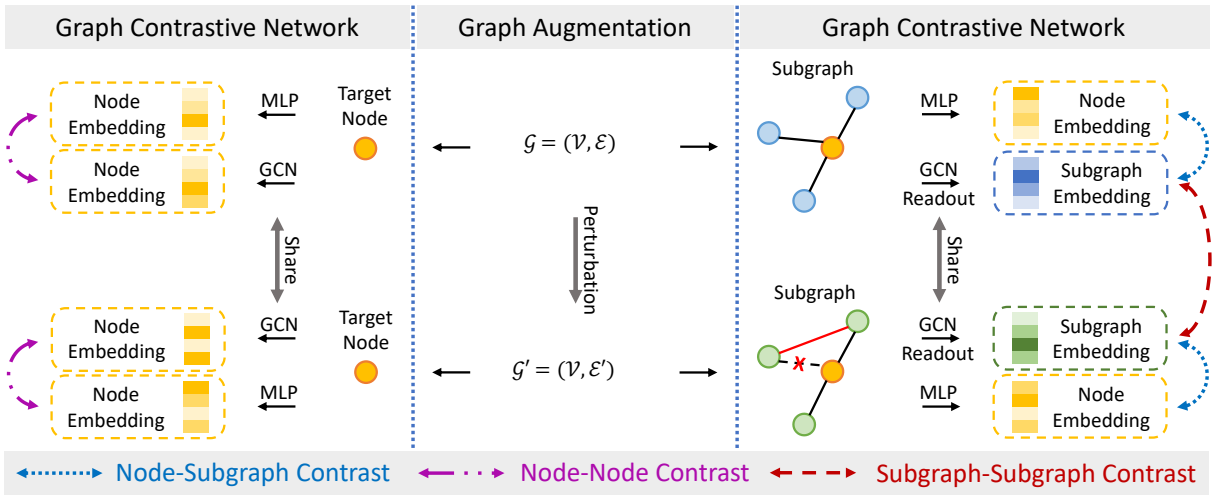


Figure 1: Overview of the GRADATE model. It is composed of two main modules: (1) Graph augmentation. We take the original graph as the first view and the edge-modified graph as the second view. Subgraphs used in the later module are generated by random walk with restart. (2) Graph contrastive network. The network captures various anomalous information from multiple scales under two views by building node-subgraph, node-node, and subgraph-subgraph contrasts. Then we comprehensively calculate the anomaly score for each node.

ates representation averaging. Differently, ANEMONE (Jin et al. 2021a) adds the node-node contrast to contrastive networks and focuses on the node-level anomalous information.

However, existing works neglect further exploitation of subgraph information and do not directly optimize their embeddings for graph anomaly detection. (Jiao et al. 2020; Hafidi et al. 2022; Han et al. 2022) have demonstrated that subgraph representation learning benefits graph-based machine learning tasks. It will significantly facilitate mining local features and structure information for respective subgraphs. For GAD, more representative and intrinsic subgraph embeddings can help to compute a more reliable relationship between nodes and their neighborhoods, which is the crucial step in the contrastive strategy.

To tackle the issue, we propose a new **GR**aph **A**nomaly **D**etection framework via multi-scale contrastive learning networks with newly added subgraph-subgraph contrast and **A**ugmen**TE**d view (termed **GRADATE**). To be specific, we regard the original input graph as the first view and adopt edge modification as graph augmentation technology to generate the second view. In each view, subgraphs are sampled by random walk. Then we build a multi-view contrastive network with node-subgraph, node-node, and subgraph-subgraph contrasts. The first two contrasts can capture subgraph-level and node-level anomalous information from each view. Subgraph-subgraph contrast is defined between two views and digs more local anomalous information for detection. In this way, node-subgraph contrast will be evidently enhanced. After that, we combine various anomalous information and calculate the anomaly score for each node. Finally, we explore and analyze the effect of different graph augmentation on subgraph representation learning in GAD. Our main contributions are listed as follows:

- We introduce subgraph-subgraph contrast to GAD for the first practice and propose a multi-scale contrastive learning networks framework with an augmented view.
- We investigate the effects of different graph augmentation on subgraph representation learning for the task.
- Extensive experiments on six benchmark datasets prove the effectiveness of the edge modification-based subgraph-subgraph contrast strategy for graph anomaly detection and the superiority of GRADATE compared to the state-of-the-art methods.

Related Work

Graph Anomaly Detection

The early works (Li et al. 2017; Perozzi and Akoglu 2016; Peng et al. 2018) usually adopt the non-deep paradigm, which detect the anomalous information from either node features and network structure. However, they cannot continuously improve their performance without digging deeper information. In recent years, the rise of neural networks (Tu et al. 2021, 2022; Liang et al. 2022; Liu et al. 2022c) has enhanced the capability of models to mine deep nonlinear information. The reconstruction-based approach DOMINANT (Ding et al. 2019) obtains node anomaly scores by calculating variations of the feature and structure matrices after GCN. AAGNN (Zhou et al. 2021) applies the one-class SVM in graph anomaly detection. HCM (Huang et al. 2021) regards the hop-count estimate of the node and its one-order neighbors as its anomaly score. CoLA (Liu et al. 2021) firstly introduces the contrastive learning paradigm (Yang et al. 2022b,a) to detect node anomalies in graph. Later methods (Jin et al. 2021a; Zheng et al. 2021; Zhang, Wang, and Chen 2022; Duan et al. 2022) make further improvements based on CoLA.

Graph Constrastive Learning

Contrastive learning is one of the most crucial paradigms in unsupervised learning. Graph contrastive learning exca-

Notations	Definitions
$\mathcal{G}$	An undirected attributed graph
v_i	The i-th node of $\mathcal{G}$
$\mathbf{A} \in \mathbb{R}^{n \times n}$	The adjacency matrix of $\mathcal{G}$
$\mathbf{D} \in \mathbb{R}^{n \times n}$	The degree matrix of $\mathbf{A}$
$\mathbf{X} \in \mathbb{R}^{n \times d}$	The feature matrix of $\mathcal{G}$
$\mathbf{H}^{(\ell)} \in \mathbb{R}^{n \times d'}$	The ℓ-th layer hidden representation matrix
$\boldsymbol{h}_i^{(\ell)} \in \mathbb{R}^{1 \times d'}$	The ℓ-th layer hidden representation of v_i
$\mathbf{W}^{(\ell)} \in \mathbb{R}^{d' \times d'}$	The ℓ-th layer network parameters
R	The number of anomaly detection rounds
S_i	The final anomaly score of v_i

Table 1: Notation summary

vates supervised information for downstream tasks without expensive labels and has made great achievements (Liu et al. 2022a). Based on the negative sample usage strategy, the existing works can be divided into negative-based and negative-free Subcategories. For the first type, DGI (Velickovic et al. 2019) maximizes the mutual information between nodes and graphs to obtain useful supervised information. SUBG-CON (Jiao et al. 2020) and GraphCL (Hafidi et al. 2022) form node-subgraph contrasts to learn better node representations. For the second type, BGRL (Thakoor et al. 2021) applies a siamese network to draw affluent information from two views. (Liu et al. 2022b,d,e) takes the advantage of Barlow Twins (Zbontar et al. 2021), which designs a special loss function to avoid representation collapse.

Graph Augmentation

Graph augmentation produces plausible variations of graph datasets (Ding et al. 2022). It expands the datasets and improves the generalization capability of the model without expensive labels (Zhao et al. 2022). Most methods focus on the operations for nodes or edges in graph. (Wang et al. 2021; Feng et al. 2020; You et al. 2020) pay attention to modify the node features. RoSA (Zhu et al. 2022) uses random walk with restart as the graph augmentation to learn the robust representation of nodes. (Klicpera, Weißenberger, and Günnemann 2019; Zhao et al. 2021) adjust the adjacency matrix by adding or removing edges.

Problem Definition

In the following section, we formalize the graph anomaly detection task. For the given undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ and $\mathcal{E}$ indicate the node set with N nodes and the edge set with M edges, separately. In addition, the node feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ and the adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ contain the node feature information and the graph structure information. Here, $\mathbf{A}_{ij} = 1$ means there is an edge between node v_i and v_j, otherwise $\mathbf{A}_{ij} = 0$. The graph anomaly detection model works for learning an anomaly scoring function f, which estimates the anomaly score S_i for each node. The larger S_i represents that the node is more likely to be anomalous. Table 1 demonstrates the main used notations in this paper.

Method

In this section, we introduce the proposed framework, GRADATE. It consists of two main modules. In the graph augmentation module, we treat the original graph as the first view and produce the second view by edge modification. In the graph contrastive network module, we first obtain anomalous information from the feature comparison of nodes and subgraphs, regarded as node-subgraph contrast. For each view, subgraphs are sampled by random walk and form pairs with the target node. Then we build node-node contrast to capture node-level anomalies. The newly added subgraph-subgraph contrast directly optimizes the subgraph embeddings for GAD between two views. In this process, the performance of node-subgraph contrast will be boosted. After that, we adopt an integrated loss function to train these three contrasts. Finally, we synthesize various anomalous information and compute the anomaly score of each node.

Graph Augmentation

Graph augmentation is crucial for the self-supervised learning paradigm. It can help the model to dig deeper semantic information of the graph. In this paper, we utilize edge modification to create the second view. Then we sample subgraphs by random walk. The nodes and subgraphs will form the input to the graph contrastive network.

Edge Modification. Edge Modification (EM) builds the second view by perturbing the graph edges. Inspired by (Jin et al. 2021b), we not only delete edges in the adjacency matrix but also add the same number of edges simultaneously. In practice, we first set a fixed proportion P to uniformly and randomly drop $\frac{PM}{2}$ edges from the adjacency matrix. Then $\frac{PM}{2}$ edges are added into the matrix uniformly and randomly. In this way, we attempt to learn robust representations of the subgraphs without destroying the properties of the graph. Ablation Study section will further discuss the graph augmentation method for generating the second view.

Random Walk. For a target node, an effective anomaly detection method is measuring the feature distance between it and its neighborhoods (Liu et al. 2021). Thus, we employ random walk with restart (RWR) (Qiu et al. 2020) to sample the subgraphs around the nodes. A lower feature similarity indicates a higher anomalous degree of the target node.

Graph Contrastive Network

The contrastive learning paradigm has been proved effective for GAD (Liu et al. 2021). We construct a multi-view graph contrastive network, which includes three parts, i.e., node-subgraph, node-node, and subgraph-subgraph contrasts. The first two contrasts are defined in each view and will be strengthened by the information fusion of the two views. Node-subgraph contrast is mainly used to capture anomalous information from the neighborhood of nodes. The second contrast allows better detection of node-level anoma-

lies. In the meantime, subgraph-subgraph contrast works between two views. It will straightforwardly optimize the subgraph embeddings for GAD, which significantly benefits node-subgraph contrast.

Node-Subgraph Contrast. A target node v_i forms a positive pair with its located subgraph and forms a negative pair with a random subgraph where another node v_j is located. We first adopt a GCN layer that maps the features of nodes in the subgraph to the embedding space. It is worth noting that the features of the target node in the subgraph are masked, i.e., set to 0. The subgraph hidden-layer representation can be defined as:

$$\mathbf{H}_i^{(\ell+1)} = \sigma\left(\widetilde{\mathbf{D}}_i^{-\frac{1}{2}}\widetilde{\mathbf{A}}_i\widetilde{\mathbf{D}}_i^{-\frac{1}{2}}\mathbf{H}_i^{(\ell)}\mathbf{W}^{(\ell)}\right), \tag{1}$$

where $\mathbf{H}_i^{(\ell+1)}$ and $\mathbf{H}_i^{(\ell)}$ indicate the $(\ell+1)$-th and ℓ-th layer hidden representation, $\widetilde{\mathbf{D}}_i^{-\frac{1}{2}}\widetilde{\mathbf{A}}_i\widetilde{\mathbf{D}}_i^{-\frac{1}{2}}$ is the normalization of the adjacency matrix, $\mathbf{W}^{(\ell)}$ denotes the network parameters.

Then, the subgraph final representation $\boldsymbol{z}_i$ is calculated by a *Readout* function. Specifically, we utilize the average function to achieve *Readout*:

$$\boldsymbol{z}_i = Readout\left(\mathbf{Z}_i\right) = \sum_{j=1}^{n_i}\frac{\left(\mathbf{Z}_i\right)_j}{n_i}. \tag{2}$$

Correspondingly, we leverage MLP to transform the target node features to the same embedding space as the subgraph. The node hidden-layer representation is:

$$\boldsymbol{h}_i^{(\ell+1)} = \sigma\left(\boldsymbol{h}_i^{(\ell)}\mathbf{W}^{(\ell)}\right), \tag{3}$$

where $\mathbf{W}^{(\ell)}$ is shared with the above GCN layer. $\boldsymbol{e}_i$ is the target node final embedding.

In each view, the anomalous degree of the target node is related to the similarity s_i^1 of the subgraph and the node embeddings. We adopt a *Bilinear* model to measure the relationship:

$$s_i^1 = Bilinear\left(\boldsymbol{z}_i, \boldsymbol{e}_i\right) = sigmoid\left(\boldsymbol{z}_i\mathbf{W}\boldsymbol{e}_i^\top\right). \tag{4}$$

Generally, the target node and subgraph representations tend to be similar in positive pairs, i.e., $s_i^1 = 1$. On the contrary, they may be dissimilar in negative pairs, i.e., $s_i^1 = 0$. Hence, we employ the binary cross-entropy (BCE) loss (Velickovic et al. 2019) to train the contrast:

$$\mathcal{L}_{NS}^1 = -\sum_{i=1}^{N}\left(y_i\log\left(s_i^1\right) + \left(1-y_i\right)\log\left(1-s_i^1\right)\right), \tag{5}$$

where y_i is equal to 1 in positive pairs, and is equal to 0 in negative pairs.

We can obtain the similarity degree s_i^2 and BCE loss $\mathcal{L}_{NS}^2$ in another view as well. It is worth mentioning that the two networks under two views use the same architecture and share parameters. Therefore, the final node-subgraph contrast loss is:

$$\mathcal{L}_{NS} = \alpha\mathcal{L}_{NS}^1 + \left(1-\alpha\right)\mathcal{L}_{NS}^2, \tag{6}$$

where $\alpha \in (0,1)$ is a trade-off parameter to balance the importance between two views.

Node-Node Contrast. Node-node contrast can effectively discover the node-level anomalies. Similarly, the target node features will be masked. And its representation is aggregated from another nodes in the subgraph. In each view, it forms a positive pair with the same node after MLP, and forms a negative pair with another node after MLP. We leverage a new GCN to obtain the representation of the subgraph:

$$\mathbf{H}_i'^{(\ell+1)} = \sigma\left(\widetilde{\mathbf{D}'}_i^{-\frac{1}{2}}\widetilde{\mathbf{A}'}_i\widetilde{\mathbf{D}'}_i^{-\frac{1}{2}}\mathbf{H}_i'^{(\ell)}\mathbf{W}'^{(\ell)}\right), \tag{7}$$

where $\mathbf{W}'^{(\ell)}$ is different from parameter matrix used in node-subgraph contrast. $\boldsymbol{h}_i'^{(\ell+1)} = \mathbf{H}_i'^{(\ell+1)}[1,:]$ is the $(\ell+1)$-th layer hidden representation of v_i. And $\boldsymbol{u}_i$ is the target node final embedding.

In the meantime, we use a MLP to map the node features into the same hidden space:

$$\hat{\boldsymbol{h}}_i^{(\ell+1)} = \sigma\left(\hat{\boldsymbol{h}}_i^{(\ell)}\mathbf{W}'^{(\ell)}\right), \tag{8}$$

where $\mathbf{W}'^{(\ell)}$ is shared with Eq. (7). After MLP, $\hat{\boldsymbol{e}}_i$ is the target node final embedding.

Similar to the node-subgraph contrast, we adopt a *Bilinear* model to evaluate the relationship $\hat{s}_i^1$ between $\boldsymbol{u}_i$ and $\hat{\boldsymbol{e}}_i$. Then the node-node contrast loss function can be defined as:

$$\mathcal{L}_{NN}^1 = -\sum_{i=1}^{N}\left(\hat{y}_i\log\left(\hat{s}_i^1\right) + \left(1-\hat{y}_i\right)\log\left(1-\hat{s}_i^1\right)\right). \tag{9}$$

Likewise, similarity degree $\hat{s}_i^2$ and loss $\mathcal{L}_{NN}^2$ can also be computed for another view. So the final node-node contrast loss function is:

$$\mathcal{L}_{NN} = \alpha\mathcal{L}_{NN}^1 + \left(1-\alpha\right)\mathcal{L}_{NN}^2, \tag{10}$$

where the view-balance parameter α is shared with the node-subgraph contrast loss in Eq. (6).

Subgraph-Subgraph Contrast. Subgraph-subgraph contrast is defined between two views. It aims to learn more representative and intrinsic subgraph embeddings for GAD, which will help node-subgraph contrast discriminate the relationship of nodes and their neighborhoods. In practice, we directly optimize the subgraph representations under the joint loss with node-subgraph contrast.

A subgraph forms a positive pair with the perturbed subgraph where the same target node v_i locates in another view. Different from common graph contrastive methods (You et al. 2020), it forms negative pairs with two subgraphs where another node v_j locates in two views. The node v_j is the same one which subgraph forms a negative pair with v_i in node-subgraph contrast. Inspired by (Oord, Li, and Vinyals 2018), we employ a loss function to optimize the contrast:

$$\mathcal{L}_{SS} = -\sum_{i=1}^{n}\log\frac{\exp\left(\boldsymbol{z}_i^1\cdot\boldsymbol{z}_i^2\right)}{\exp\left(\boldsymbol{z}_i^1\cdot\boldsymbol{z}_j^1\right)+\exp\left(\boldsymbol{z}_i^1\cdot\boldsymbol{z}_j^2\right)}, \tag{11}$$

where $\boldsymbol{z}_i^1$ and $\boldsymbol{z}_i^2$ represent the embeddings of the subgraphs that the target node v_i belongs to in two views. And $\boldsymbol{z}_j^1$ and $\boldsymbol{z}_j^2$ are the embeddings of the subgraphs where another node v_j locates in two views, respectively.

Algorithm 1: Proposed model GRADATE.

Input: An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$; Number of training epochs E; Batch size B.

Output: Anomaly score function f.

for $e = 1$ to E **do**
  Form two views from the original graph and the graph augmented by edge modification. And sample subgraphs by random walk in each view.
  $\mathcal{V}$ is divided into batches with size B by random.
  for $v_i \in B$ **do**
    In node-subgraph contrast, estimate the representation similarity of the target node and two subgraphs in positive and negative pairs via Eq. (4).
    In node-node contrast, calculate the representation similarity of the target node and two nodes in positive and negative pairs.
    Calculate the node-subgraph, node-node, and subgraph-subgraph loss via Eq. (6), (10) and (11). Then obtain the final joint loss via Eq. (12).
    Back propagation and update trainable parameters.
  end for
end for
By multi-round detections, calculate the final anomaly score for each node via Eq. (13), (14) and (15).

Loss Function. To integrate the advantages of three contrasts, we optimize the joint loss function:

$$\mathcal{L} = \beta \mathcal{L}_{NS} + (1 - \beta)\, \mathcal{L}_{NN} + \gamma \mathcal{L}_{SS}, \tag{12}$$

where $\beta \in (0, 1)$ is a balance parameter of subgraph-level and node-level anomalous information. $\gamma \in (0, 1)$ is a trade-off parameter of $\mathcal{L}_{SS}$.

Anomaly Score Calculation. In node-subgraph and node-node contrasts, a normal node is similar to the subgraph or node in its positive pair and dissimilar to the subgraph or node in its negative pair. On the contrary, an anomalous node is dissimilar to the node or subgraph in both positive and negative pairs. Naturally, we define the anomaly score of the target node as follows:

$$s_i = s_i^n - s_i^p, \tag{13}$$

where s_i^p and s_i^n represent the similarity of positive and negative pairs.

Then we comprehensively fuse the anomalous information from two views and three contrast. The anomaly score can further be represented as:

$$\begin{aligned} s_i &= \alpha s_i^1 + (1 - \alpha)\, s_i^2, \\ \hat{s}_i &= \alpha \hat{s}_i^1 + (1 - \alpha)\, \hat{s}_i^2, \\ S_i &= \beta s_i + (1 - \beta)\, \hat{s}_i, \end{aligned} \tag{14}$$

where α and β are shared with Eq. (6), (10) and (12).

Once detection with only once random walk cannot capture sufficient semantic information. Multi-round detections are essential to compute anomaly score for each node. Inspired by (Jin et al. 2021a), we calculate the final anomaly score by the mean and the standard deviation from the results of multi-round detections:

$$\begin{aligned} \bar{S}_i &= \frac{1}{R} \sum_{r=1}^{R} S_i^{(r)}, \\ S_i &= \bar{S}_i + \sqrt{\frac{1}{R} \sum_{r=1}^{R} \left(S_i^{(r)} - \bar{S}_i \right)^2}, \end{aligned} \tag{15}$$

where R is the number of anomaly detection rounds.

In general, the overall procedures of GRADATE are shown in Algorithm 1.

Experiments

We conduct extensive experiments on six graph benchmark datasets to verify the excellent performance of GRADARE. The results also confirm the effectiveness of subgraph-subgraph contrast and edge modification for GAD.

Experiment Settings

Details of the experiment settings are shown as follows: **(1)Datasets.** The proposed method is evaluated on six benchmark datasets which details are shown in Table 2. The datasets include Citation (Yuan et al. 2021), Cora (Sen et al. 2008), WebKB (Craven et al. 1998), UAI2010 (Wang et al. 2018), UAT and EAT (Mrabah et al. 2022). **(2)Anomaly Injection.** Following DOMINANT, we inject the same number of feature and structure anomalies into the original datasets which do not have anomalous nodes before. The total number of anomalies for each dataset is shown in the last column of Table 2. **(3)Baselines.** For the GAD task, we compare with eight well-known baseline methods. They are summarized in the first column of Table 3. The first two models are non-deep algorithms, and the rest are based on graph neural networks. Following CoLA, the node features of all datasets are reduced to 30 by PCA before running ANOMALOUS. **(4)Metric.** We adopt a widely-used anomaly detection metric AUC to evaluate the above methods.

Datasets	Nodes	Edges	Attributes	Anomalies
EAT	399	5993	203	30
WebKB	919	1662	1703	60
UAT	1190	13599	239	60
Cora	2708	5429	1433	150
UAI2010	3067	28311	4973	150
Citation	8935	15098	6775	450

Table 2: The statistics of datasets.

Model Parameters

In node-subgraph and subgraph-subgraph contrasts, both GCN models have one layer and use ReLU as the activation function. The size of subgraphs in the network is set to 4. Both node and subgraph features are mapped to 64 dimensions in hidden space. Besides, we implement 400 epochs of model training and 256 rounds of anomaly score calculation.

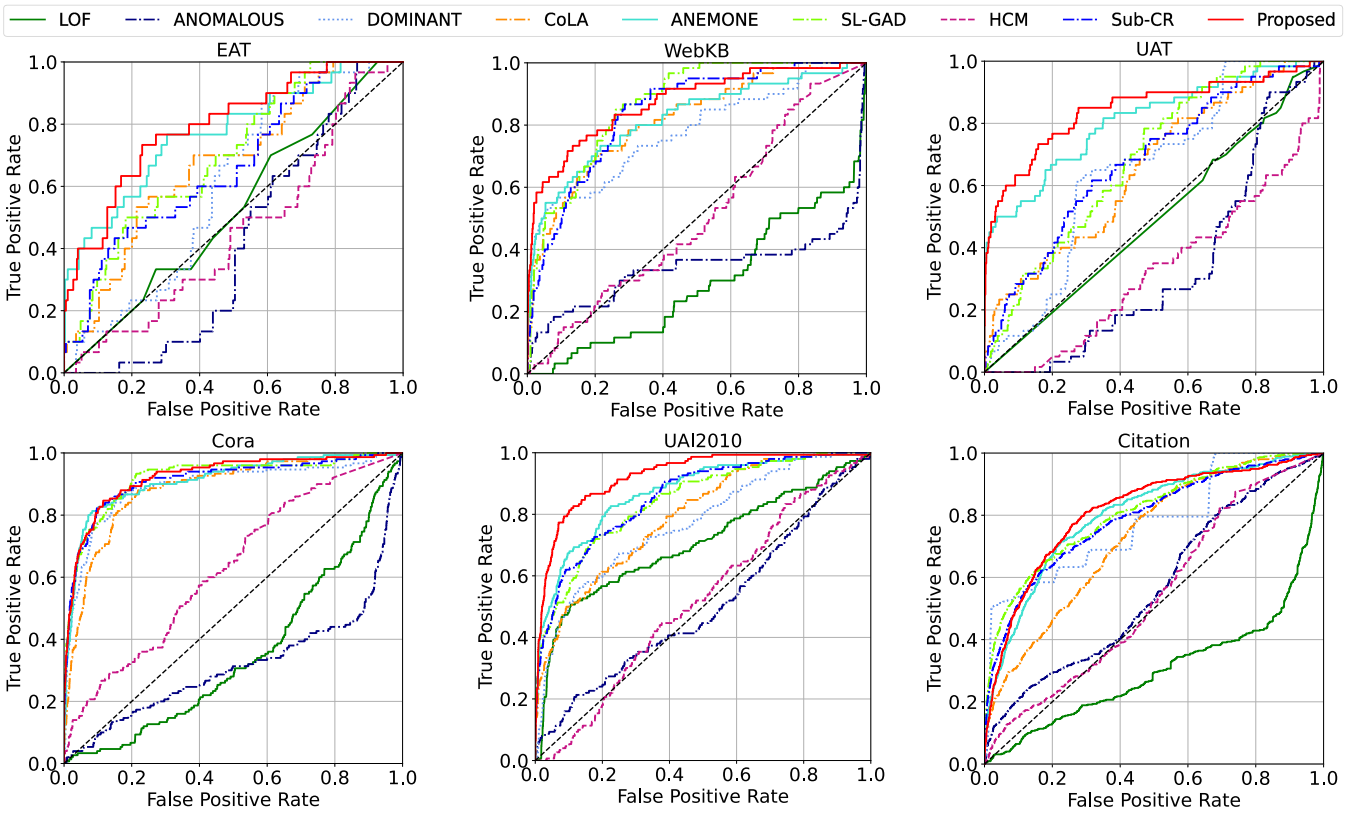


Figure 2: ROC curves on six benchmark datasets. The area under the curve is larger, the anomaly detection performance is better. The black dotted lines are the "random line", indicating the performance under random guessing.

Methods	EAT	WebKB	UAT	Cora	UAI2010	Citation
LOF (Breunig et al. 2000)	0.5255	0.2903	0.4906	0.3538	0.7052	0.3059
ANOMALOUS (Peng et al. 2018)	0.4109	0.3417	0.3356	0.3198	0.5026	0.5656
DOMINANT (Ding et al. 2019)	0.6023	0.7787	0.6503	0.8929	0.7698	0.7748
CoLA (Liu et al. 2021)	0.6762	0.8175	0.6538	0.8847	0.7949	0.7296
ANEMONE (Jin et al. 2021a)	<u>0.7726</u>	0.8208	<u>0.8087</u>	0.9122	<u>0.8731</u>	0.8028
SL-GAD (Zheng et al. 2021)	0.6974	<u>0.8678</u>	0.6851	<u>0.9192</u>	0.8454	<u>0.8095</u>
HCM (Huang et al. 2021)	0.4536	0.5064	0.3262	0.6276	0.5210	0.5414
Sub-CR (Zhang, Wang, and Chen 2022)	0.6672	0.8423	0.6788	0.9133	0.8571	0.7903
GRADATE	**0.7980**	**0.8740**	**0.8451**	**0.9237**	**0.9262**	**0.8138**

Table 3: Performance comparison for AUC. The bold and underlined values indicate the best and runner-up results, respectively.

Result and Analysis

In this subsection, we evaluate the anomaly detection performance of GRADATE by comparing it with eight baseline methods. Figure 2 demonstrates the ROC curves for nine models. Meanwhile, Table 3 shows the comparison results of AUC values corresponding to Figure 2. For the results, we have the following conclusions:

- We can intuitively find that GRADATE outperforms its competitors on these six datasets. To be specific, GRADATE achieves notable AUC gains of **2.54%**, **0.62%**, **3.64%**, **0.45%**, **5.31%**, and **0.43%** on EAT, WebKB, UAT, Cora, UAI2010, and Citation, respectively. And as shown in Figure 2, the under-curve areas of GRADATE are significantly larger than competitors.
- We observe that most neural network-based methods outperform the shallow methods, LOF and ANOMALOUS. Shallow methods have inherent limitations on dealing with the high-dimension features of graph data.
- Among the deep methods, contrastive learning-based methods, CoLA, ANEMONE, SL-GAD, Sub-CR, and GRADATE work better. It indicates that the contrastive learning-based pattern can effectively detect anomalies by mining the feature and structure information from graph. With the newly added subgraph-subgraph contrast and multi-view learning strategies, GRADATE achieves the best performance.

Ablation Study

Various Scale Contrastive Strategy. To verify the effectiveness of the proposed subgraph-subgraph contrast, we perform ablation study experiments. For convenience, **NS**, **NS+SS**, **NS+NN**, and **NS+NN+SS** indicate only using node-subgraph contrast (CoLA), using node-subgraph and subgraph-subgraph contrasts, using node-subgraph and node-node contrasts (ANEMONE), and using above three contrasts (GRADATE), separately. As shown in Table 4, adding subgraph-subgraph contrast can enhance the detection performance by boosting node-subgraph contrast. Using all three contrasts will lead to the best performance.

	EAT	WebKB	UAT	Cora	UAI2010	Citation
NS	0.6762	0.7949	0.6538	0.8847	0.8175	0.7296
NS+SS	0.6800	0.8310	0.6603	0.8956	0.9055	0.6978
NS+NN	0.7726	0.8208	0.8087	0.9122	0.8731	0.8028
NS+NN+SS	**0.7980**	**0.8740**	**0.8451**	**0.9237**	**0.9262**	**0.8138**

Table 4: Ablation study for contrast scale w.r.t. AUC.

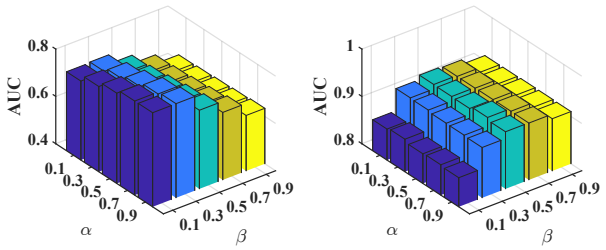
Graph Augmentation Strategy. In the meantime, we adopt four different graph augmentation to form the second view and explore their effects on performance. **Gaussian Noised Feature (GNF)** means node features are randomly perturbed with Gaussian noise. **Feature Mask (FM)** indicates random parts of node features are masked. **Graph Diffusion (GD)** utilizes the graph diffusion matrix which is generated by the diffusion model (Hassani and Khasahmadi 2020; Klicpera, Weißenberger, and Günnemann 2019). GNF and FM are perturbations of node features. GD and **Edge Modification (EM)** are widely-used graph augmentation methods on graph edges. As shown in Table 5, EM achieves the best performance on all datasets. On further analysis, the perturbation of node features may disrupt the features of normal nodes. It will harm the comparison between the nodes and their neighborhoods, which is the basis of contrastive learning for GAD. This can cause some normal nodes to be misclassified as anomalies and lead to performance degradation. Moreover, GD is a structure-based graph augmentation method. However, its primary purpose is to capture global information. Hence, EM is more compatible with subgraph-subgraph contrast than GD, which can improve node-subgraph contrast to dig the local neighborhood information of nodes.

	EAT	WebKB	UAT	Cora	UAI2010	Citation
GNF	0.7548	0.8183	0.8327	0.9031	0.9193	0.7902
FM	0.7782	0.8148	0.8256	0.8924	0.9171	0.8034
GD	0.7618	0.8062	0.8143	0.9026	0.9161	0.8030
EM	**0.7980**	**0.8740**	**0.8451**	**0.9237**	**0.9262**	**0.8138**

Table 5: Ablation study for graph augmentation w.r.t. AUC.

Sensitivity Analysis

Balance Parameter α, β and γ. We discuss the three important balance parameters in the loss function. As shown in Figure 3, the hyper-parameter α and β are effective in improving the detection performance on EAT and UAI2010. Similar phenomena can be observed on the other datasets. In practice, we set α to 0.9, 0.1, 0.7, 0.9, 0.7, and 0.5 on EAT, WebKB, UAT, Cora, UAI2010 and Citation. Meanwhile, we make β to 0.3, 0.7, 0.1, 0.3, 0.5, and 0.5.



(a) EAT (b) UAI2010

Figure 3: Sensitivity analysis for the balance parameters α and β w.r.t. AUC on EAT and UAI2010.

Figure 4 illustrates the performance variation of GRADATE when γ varies from 0.1 to 0.9. From the figure, we observe that GRADATE tends to perform well by setting γ to 0.1 across all benchmarks.

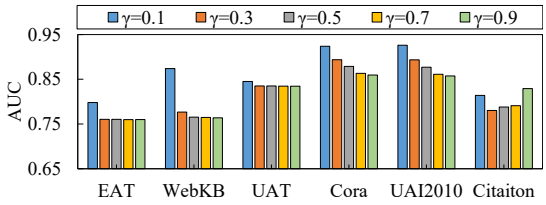


Figure 4: Trade-off parameter γ w.r.t. AUC.

Edge Modification Proportion P. We also investigate the influence of different parameterized edge modification. Figure 5 shows that detection performance receives a relatively small fluctuation by modification proportion P on UAT, UAI2010, and Citation. Comprehensively, we fixedly set $P = 0.2$ on all datasets.

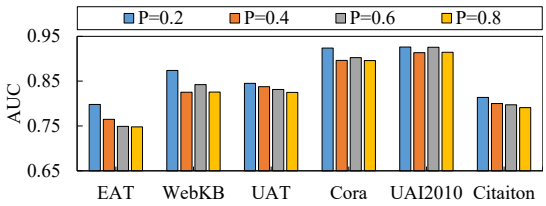


Figure 5: Perturbation proportion P w.r.t. AUC.

Conclusion

In this paper, we propose a new graph anomaly detection framework via multi-scale contrastive learning networks with an augmented view. We introduce subgraph-subgraph contrast to GAD and investigate the impact of different graph augmentation technologies on the task. Extensive experiments on six benchmark datasets prove GRADATE outperforms the competitors. In the future, we will continue to explore the contrastive learning pattern for the task.

Acknowledgments

This work was supported by the National Key R&D Program of China (project no. 2022ZD0209103) and the National Natural Science Foundation of China (project no. 61922088, 61976196, 61872377 and 61872371).

References

Breunig, M. M.; Kriegel, H.-P.; Ng, R. T.; and Sander, J. 2000. LOF: identifying density-based local outliers. In *Proc. of SIGMOD*.

Cheng, Z.; Wang, S.; Zhang, P.; Wang, S.; Liu, X.; and Zhu, E. 2021a. Improved autoencoder for unsupervised anomaly detection. *International Journal of Intelligent Systems*.

Cheng, Z.; Zhu, E.; Wang, S.; Zhang, P.; and Li, W. 2021b. Unsupervised outlier detection via transformation invariant autoencoder. *IEEE Access*.

Craven, M.; DiPasquo, D.; Freitag, D.; McCallum, A.; Mitchell, T. M.; Nigam, K.; and Slattery, S. 1998. Learning to Extract Symbolic Knowledge from the World Wide Web. In *Proc. of AAAI*.

Ding, K.; Li, J.; Bhanushali, R.; and Liu, H. 2019. Deep anomaly detection on attributed networks. In *Proc. of SDM*.

Ding, K.; Xu, Z.; Tong, H.; and Liu, H. 2022. Data augmentation for deep graph learning: A survey. *arXiv preprint arXiv:2202.08235*.

Duan, J.; Wang, S.; Liu, X.; Zhou, H.; Hu, J.; and Jin, H. 2022. GADMSL: Graph Anomaly Detection on Attributed Networks via Multi-scale Substructure Learning. *arXiv preprint arXiv:2211.15255*.

Feng, W.; Zhang, J.; Dong, Y.; Han, Y.; Luan, H.; Xu, Q.; Yang, Q.; Kharlamov, E.; and Tang, J. 2020. Graph random neural networks for semi-supervised learning on graphs. *Proc. of NeurIPS*.

Garcia-Teodoro, P.; Diaz-Verdejo, J.; Maciá-Fernández, G.; and Vázquez, E. 2009. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*.

Hafidi, H.; Ghogho, M.; Ciblat, P.; and Swami, A. 2022. Negative sampling strategies for contrastive self-supervised learning of graph representations. *Signal Processing*.

Han, Y.; Hui, L.; Jiang, H.; Qian, J.; and Xie, J. 2022. Generative Subgraph Contrast for Self-Supervised Graph Representation Learning. In *Proc. of ECCV*.

Hassani, K.; and Khasahmadi, A. H. 2020. Contrastive multi-view representation learning on graphs. In *Proc. of ICML*.

Hu, J.; Yu, G.; Wang, S.; Zhu, E.; Cai, Z.; and Zhu, X. 2022. Detecting Anomalous Events from Unlabeled Videos via Temporal Masked Auto-Encoding. In *Proc. of ICME*.

Huang, D.; Mu, D.; Yang, L.; and Cai, X. 2018. CoDetect: Financial fraud detection with anomaly feature detection. *IEEE Access*.

Huang, T.; Pei, Y.; Menkovski, V.; and Pechenizkiy, M. 2021. Hop-count based self-supervised anomaly detection on attributed networks. *arXiv preprint arXiv:2104.07917*.

Jiao, Y.; Xiong, Y.; Zhang, J.; Zhang, Y.; Zhang, T.; and Zhu, Y. 2020. Sub-graph contrast for scalable self-supervised graph representation learning. In *Proc. of ICDM*.

Jin, M.; Liu, Y.; Zheng, Y.; Chi, L.; Li, Y.-F.; and Pan, S. 2021a. ANEMONE: Graph Anomaly Detection with Multi-Scale Contrastive Learning. In *Proc. of CIKM*.

Jin, M.; Zheng, Y.; Li, Y.-F.; Gong, C.; Zhou, C.; and Pan, S. 2021b. Multi-scale contrastive siamese networks for self-supervised graph representation learning. In *Proc. of IJCAI*.

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

Klicpera, J.; Weißenberger, S.; and Günnemann, S. 2019. Diffusion improves graph learning. In *Proc. of ICONIP*.

Li, J.; Dani, H.; Hu, X.; and Liu, H. 2017. Radar: Residual Analysis for Anomaly Detection in Attributed Networks. In *Proc. of IJCAI*.

Liang, K.; Liu, Y.; Zhou, S.; Liu, X.; and Tu, W. 2022. Relational Symmetry based Knowledge Graph Contrastive Learning.

Liu, Y.; Jin, M.; Pan, S.; Zhou, C.; Zheng, Y.; Xia, F.; and Yu, P. 2022a. Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering*.

Liu, Y.; Li, Z.; Pan, S.; Gong, C.; Zhou, C.; and Karypis, G. 2021. Anomaly detection on attributed networks via contrastive self-supervised learning. *IEEE transactions on neural networks and learning systems*.

Liu, Y.; Tu, W.; Zhou, S.; Liu, X.; Song, L.; Yang, X.; and Zhu, E. 2022b. Deep Graph Clustering via Dual Correlation Reduction. In *Proc. of AAAI*.

Liu, Y.; Xia, J.; Zhou, S.; Wang, S.; Guo, X.; Yang, X.; Liang, K.; Tu, W.; Li, Z. S.; and Liu, X. 2022c. A Survey of Deep Graph Clustering: Taxonomy, Challenge, and Application. *arXiv preprint arXiv:2211.12875*.

Liu, Y.; Yang, X.; Zhou, S.; and Liu, X. 2022d. Simple Contrastive Graph Clustering. *arXiv preprint arXiv:2205.07865*.

Liu, Y.; Zhou, S.; Liu, X.; Tu, W.; and Yang, X. 2022e. Improved Dual Correlation Reduction Network. *arXiv preprint arXiv:2202.12533*.

Ma, X.; Wu, J.; Xue, S.; Yang, J.; Zhou, C.; Sheng, Q. Z.; Xiong, H.; and Akoglu, L. 2021. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering*.

Mrabah, N.; Bouguessa, M.; Touati, M. F.; and Ksantini, R. 2022. Rethinking graph auto-encoder models for attributed graph clustering. *IEEE Transactions on Knowledge and Data Engineering*.

Müller, E.; Sánchez, P. I.; Mülle, Y.; and Böhm, K. 2013. Ranking outlier nodes in subspaces of attributed graphs. In *2013 IEEE 29th international conference on data engineering workshops (ICDEW)*.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Peng, Z.; Luo, M.; Li, J.; Liu, H.; and Zheng, Q. 2018. ANOMALOUS: A Joint Modeling Approach for Anomaly Detection on Attributed Networks. In *Proc. of IJCAI*.

Perozzi, B.; and Akoglu, L. 2016. Scalable anomaly ranking of attributed neighborhoods. In *Proc. of SDM*.

Perozzi, B.; Akoglu, L.; Iglesias Sánchez, P.; and Müller, E. 2014. Focused clustering and outlier detection in large attributed graphs. In *Proc. of KDD*.

Qiu, J.; Chen, Q.; Dong, Y.; Zhang, J.; Yang, H.; Ding, M.; Wang, K.; and Tang, J. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proc. of KDD*.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*.

Tang, J.; Li, J.; Gao, Z.; and Li, J. 2022. Rethinking Graph Neural Networks for Anomaly Detection. *arXiv preprint arXiv:2205.15508*.

Thakoor, S.; Tallec, C.; Azar, M. G.; Munos, R.; Veličković, P.; and Valko, M. 2021. Bootstrapped Representation Learning on Graphs. In *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*.

Tu, W.; Zhou, S.; Liu, X.; Guo, X.; Cai, Z.; zhu, E.; and Cheng, J. 2021. Deep Fusion Clustering Network. In *Proc. of AAAI*.

Tu, W.; Zhou, S.; Liu, X.; Yue, L.; Cai, Z.; zhu, E.; Changwang, Z.; and Cheng, J. 2022. Initializing Then Refining: A Simple Graph Attribute Imputation Network. In *Proc. of IJCAI*.

Velickovic, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2019. Deep Graph Infomax. *ICLR (Poster)*.

Wang, W.; Liu, X.; Jiao, P.; Chen, X.; and Jin, D. 2018. A unified weakly supervised framework for community detection and semantic matching. In *Proc. of KDD*.

Wang, Y.; Wang, W.; Liang, Y.; Cai, Y.; and Hooi, B. 2021. Mixup for node and graph classification. In *Proc. of WWW*.

Wu, L.; Morstatter, F.; Carley, K. M.; and Liu, H. 2019. Misinformation in social media: definition, manipulation, and detection. *ACM SIGKDD Explorations Newsletter*.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Philip, S. Y. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*.

Xu, X.; Yuruk, N.; Feng, Z.; and Schweiger, T. A. 2007. Scan: a structural clustering algorithm for networks. In *Proc. of KDD*.

Yang, X.; Hu, X.; Zhou, S.; Liu, X.; and Zhu, E. 2022a. Interpolation-Based Contrastive Learning for Few-Label Semi-Supervised Learning. *IEEE Transactions on Neural Networks and Learning Systems*.

Yang, X.; Liu, Y.; Zhou, S.; Liu, X.; and Zhu, E. 2022b. Mixed Graph Contrastive Network for Semi-Supervised Node Classification. *arXiv preprint arXiv:2206.02796*.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph contrastive learning with augmentations. *Proc. of NeurIPS*.

Yuan, X.; Zhou, N.; Yu, S.; Huang, H.; Chen, Z.; and Xia, F. 2021. Higher-order Structure Based Anomaly Detection on Attributed Networks. In *2021 IEEE International Conference on Big Data (Big Data)*.

Zbontar, J.; Jing, L.; Misra, I.; LeCun, Y.; and Deny, S. 2021. Barlow twins: Self-supervised learning via redundancy reduction. In *Proc. of ICML*.

Zhang, J.; Wang, S.; and Chen, S. 2022. Reconstruction Enhanced Multi-View Contrastive Learning for Anomaly Detection on Attributed Networks. *arXiv preprint arXiv:2205.04816*.

Zhao, T.; Liu, G.; Günnemann, S.; and Jiang, M. 2022. Graph Data Augmentation for Graph Machine Learning: A Survey. *arXiv preprint arXiv:2202.08871*.

Zhao, T.; Liu, Y.; Neves, L.; Woodford, O.; Jiang, M.; and Shah, N. 2021. Data augmentation for graph neural networks. In *Proc. of AAAI*.

Zheng, Y.; Jin, M.; Liu, Y.; Chi, L.; Phan, K. T.; and Chen, Y.-P. P. 2021. Generative and Contrastive Self-Supervised Learning for Graph Anomaly Detection. *IEEE Transactions on Knowledge and Data Engineering*.

Zhou, S.; Tan, Q.; Xu, Z.; Huang, X.; and Chung, F.-l. 2021. Subtractive Aggregation for Attributed Network Anomaly Detection. In *Proc. of CIKM*.

Zhu, Y.; Guo, J.; Wu, F.; and Tang, S. 2022. RoSA: A Robust Self-Aligned Framework for Node-Node Graph Contrastive Learning. *arXiv preprint arXiv:2204.13846*.