ECAI 2023
K. Gal et al. (Eds.)

© 2023 The Authors.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).
doi:10.3233/FAIA230440

Revisiting Graph Contrastive Learning for Anomaly Detection

Zhiyuan Liu[a, b], **Chunjie Cao**[a, b; *], **Fangjian Tao**[a, b] **and Jingzhang Sun**[a, b; *]

[a]School of Cyberspace Security, Hainan University
[b]Key Laboratory of Information Retrieval of Hainan Province
ORCiD ID: Zhiyuan Liu https://orcid.org/0000-0002-9862-393X,
Chunjie Cao https://orcid.org/0000-0001-9439-8256, Fangjian Tao https://orcid.org/0000-0001-8637-7383,
Jingzhang Sun https://orcid.org/0000-0002-6961-6677

Abstract. Combining Graph neural networks (GNNs) with contrastive learning for anomaly detection has drawn rising attention recently. Existing graph contrastive anomaly detection (GCAD) methods have primarily focused on improving detection capability through graph augmentation and multi-scale contrast modules. However, the underlying mechanisms of how these modules work have not been fully explored. We dive into the multi-scale and graph augmentation mechanism and observed that multi-scale contrast modules do not enhance the expression, while the multi-GNN modules are the hidden contributors. Previous studies have tended to attribute the benefits brought by multi-GNN to the multi-scale modules. In the paper, we delve into the misconception and propose Multi-GNN and Augmented Graph contrastive framework MAG, which unified the existing GCAD methods in the contrastive self-supervised perspective. We extracted two variants from the MAG framework, L-MAG and M-MAG. The L-MAG is the lightweight instance of the MAG, which outperform the state-of-the-art on Cora and Pubmed with the low computational cost. The variant M-MAG equipped with multi-GNN modules further improve the detection performance. Our study sheds light on the drawback of the existing GCAD methods and demonstrates the potential of multi-GNN and graph augmentation modules. Our code is available at https://anonymous.4open.science/r/MAG-Framework-74D0.

1 Introduction

Anomaly detection has garnered significant attention in industry, such as network intrusions [12, 31], money laundering [16, 25] and financial fraud detection [20], since it plays a critical role in identifying anomalous patterns and mitigating potential risks. Previously, shallow learning methods like ANOMOLOUS [19] and Radar [11] were benefited from its residual analysis technique for anomaly detection. However, they are hard to handle the non-linear high-dimensional data and complex interaction patterns. In response, graph neural network (GNN) methods have emerged as powerful network skeletons for anomaly detection due to the capability to model complex patterns.

Still, detecting anomalies is challenging, since abnormal instances are often scarce and difficult to label [1]. To address this issue, contrastive learning, benefited from its self-supervised property, has been combined with GNN models for anomaly detection. Existing graph contrastive anomaly detection (GCAD) methods, such as ANEMONE [7], SL-GAD [33], and GRADATE [4], have utilized graph augmentation or multi-scale contrast modules to upgrade their models. However, these incremental works enhance the expression of the model by adding different multi-scale contrasts or graph augmentation strategies intuitively without any empirical design guidance. The impact of multi-scale contrast and graph augmentation on GCAD has not been extensively studied.

Revisiting the ANEMONE [7], we found that the ANEMONE method actually benefited from the multi-GNN modules, not the additional node-node contrast loss. For graph augmentation, the combination of masked feature and removed edge show a significant competitiveness.

In this paper, we proposed Multi-GNN and Augmented Graph contrastive framework MAG, which unified the existing GCAD methods in the contrastive self-supervised perspective. By adjusting the hyper-parameters of the MAG framework, we could degrade MAG to the classical GCAD methods, such as CoLA [14], ANEMONE [7], SL-GAD [33], or GRADATE [33] methods. We traversed thoroughly the single contrast instances of the MAG framework and observed that the normal node-subgraph contrast had better detection performance than the node-node, sugraph-subgraph, and masked node-subgraph contrasts. Unlike the GRADATE [4] model used a variety of multi-scale contrast combinations, our lightweight L-MAG surpasses the state-of-the-art on Cora and Pubmed with the low computational cost. The variant M-MAG model equipped with multi-GNN modules further improve the detection performance. Our contributions can be summarized as follows:

- To the best of our knowledge, we are the first group to unify GCAD models in the contrastive self-supervised perspective.
- We suggested that the multi-scale contrast modules are the *"puppets"*, the backstage *"pusher"* are the multi-GNN modules in GCAD.
- We provided empirical design guidance for different scale contrasts and graph augmentation strategies in GCAD.
- The lightweight L-MAG outperforms the state-of-the-art with the low computational cost, the M-MAG improve detection performance further.

* Corresponding Authors. Email: caochunjie@hainanu.edu.cn; jingzhangsun@hainanu.edu.cn.

2 Background on Graph Anomaly Detection

For simplicity, we use capital letters, bold lowercase letters, and lowercase letters to denote matrices, vectors, and constants respectively, e.g. $X, \boldsymbol{x}, x$. Giving graph $\mathcal{G}(\mathcal{V}, X, A)$, $\mathcal{V}$ is composed of a series of nodes $\{\boldsymbol{v_1}, \boldsymbol{v_i}, ..., \boldsymbol{v_n}\}$, $X \in \mathbf{R}^{n\times d}$ consists of a set of vectors $\{\boldsymbol{x_1}, \boldsymbol{x_i}, ..., \boldsymbol{x_n}\}$, $\boldsymbol{x_i} \in \mathbf{R}^d$. $A \in \mathbf{R}^{n\times n}$ is the adjacency matrix of $\mathcal{G}$, where the entry $A_{i,j}$ equals to 1 if there is an edge between the $\boldsymbol{v_i}$ and $\boldsymbol{v_j}$, otherwise 0. For semi-supervised setting, we denote $\boldsymbol{y_L} = \{y_{l1}, y_{l2}, .., y_{lp}\}$ as the known labels, and $\boldsymbol{y_U} = \{y_{u1}, y_{u2}, .., y_{uq}\}$ represents the unknown labels that we have to deduce. In this section, we will brief typical graph anomaly detection techniques and formulate them below.

2.1 GNN-based

GNN-based methods treat anomaly detection as an unbalance binary classification task. Like GNN classifications [15, 24, 32], we obtain node representations Z via GNN mapping function $\mathcal{F}$, where $\mathcal{F}$ can be the skeleton of GCN [9], GAT [27] et al. The probability score $\hat{\boldsymbol{y}} \in R^n$ can be obtained by transforming Z to $\hat{\boldsymbol{y}}$ with the multilayer perceptron. The weighted binary cross entropy between the real labeled $\boldsymbol{y_L}$ and the probability score $\hat{\boldsymbol{y}}$ will be optimized for model training.

$$\begin{aligned} \hat{\boldsymbol{y}} &= MLP(\mathcal{F}(X, A)) \\ \mathcal{L} &= \frac{1}{p} \cdot \sum_{i}^{p} (\alpha \cdot y_{li} \log \hat{y_{li}} + (1 - y_{li}) \log(1 - \hat{y_{li}})) \end{aligned} \tag{1}$$

where p is the number of the known labels $\boldsymbol{y_L}$, α is the balance factor to regulate the imbalance between the normal and abnormal nodes. In the stage of inference, node $\boldsymbol{v_i}$ can be classified by the corresponding probability score $\hat{\boldsymbol{y}}_i$. Larger $\hat{\boldsymbol{y}}_i$, more abnormal. Based on the above procedures, Tang et.al [24] analyzed from graph spectral perspective and designed spectral and spatially localized bandpass filters to better fit the anomaly detection task. And establish a solid graph anomaly detection benchmark [23] to offer pivotal insights into the current advancements of graph anomaly detection. Zhang et al. [32] concated intermediate representations, introduced fraud-aware and imbalance-oriented classification modules to overcome graph inconsistency and imbalance drawbacks in fraud detection.

2.2 Reconstruction-based

Reconstruction-based methods reconstruct the original graph $\mathcal{G}(\mathcal{V}, X, A)$ via the graph autoencoder architecture [10]. It has been observed that normal nodes tend to have richer consistency with neighbouring nodes [22]. Thus, normal nodes are more easily in recovering than abnormal nodes. Exploiting the property, DOMINANT [3] as one of the classical reconstruction algorithms was presented. Differently, Fan et.al [5] suggested that existing methods neglected the complex cross-modality interactions between network structure and node attribute. To this end, AnomalyDAE incorporates the attention mechanism to assess the significance of neighboring nodes, while also utilizing a dual autoencoder to enhance cross-modality representation capabilities.

2.3 Contrastive-based

One of crucial modules for contrastive learning is to construct instance pairs. In GCAD, for a given node $\boldsymbol{v_i}$, we sample its subgraph

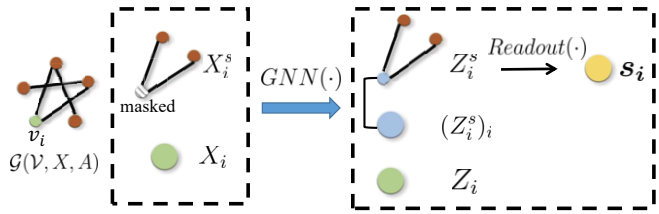

Figure 1. The three types representation of the leftmost light green node v_i, node feature Z_i (the rightmost green), masked node feature $(Z_i^s)_i$ (the rightmost blue) and subgraph feature $\boldsymbol{s_i}$ (the rightmost yellow).

$\mathcal{G}_i$ (X_i in $\mathcal{G}_i$ masked with 0) using random walk restart (RWR [26]) method and find a distinct node $\boldsymbol{v_j}, (i \neq j)$ as the negative pair of $\boldsymbol{v_i}$, where X_i is the attribute features of node $\boldsymbol{v_i}$. We put the node feature $X_i \in \mathcal{R}^d$ and its sampled subgraph $\mathcal{G}_i(X_i^s, A_i^s)$ to the GNN mapping $\mathcal{F}$ to get the represented node feature Z_i and its subgraph representation $Z_i^s \in \mathcal{R}^{m\times d}$, where m is the number of nodes in sampled subgraph $\mathcal{G}_i$. We apply readout function to flatten Z_i^s to $\boldsymbol{s_i} \in \mathcal{R}^d$. Due to $\boldsymbol{s_i}$ derived from node $\boldsymbol{v_i}$, the logical distance between Z_i and $\boldsymbol{s_i}$ shall be close. Similarly, $\boldsymbol{s_j}$ derived from $\boldsymbol{v_j}$, which shall be far away from Z_i. We can formulate as follows:

$$\begin{aligned} \boldsymbol{s_i} &= Readout(\mathcal{F}(\mathcal{G}_i(X_i^s, A_i^s))), \quad Z_i = \mathcal{F}(X_i) \\ y_i &= Bilinear(\boldsymbol{s_i}, Z_i), \quad \hat{y_i} = Bilinear(\boldsymbol{s_j}, Z_i) \\ \mathcal{L}_1 &= \frac{1}{n} \cdot \sum_{i}^{n} \log y_i + \log(1 - \hat{y_i}) \end{aligned} \tag{2}$$

where $Bilinear(\cdot)$ is the bilinear function to obtain consistency score between two vectors, n denotes the number of nodes. One of the classical GCAD models CoLA [14] achieve single scale node-subgraph contrast, which operates similar with the above formula. However, Jin et.al. [7] illustrated that existing efforts only model the instance pairs in a single scale aspect, thus limiting in capturing complex anomalous patterns. To this end, ANEMONE equipped with the additional node-node contrast was proposed. Following the above expression, ANEMONE can be formulated as below.

$$\begin{aligned} Z_i^s &= \mathcal{F}(\mathcal{G}_i(X_i^s, A_i^s)) \\ y_i^{(1)} &= Bilinear((Z_i^s)_i, Z_i), \quad \hat{y_i}^{(1)} = Bilinear((Z_j^s)_j, Z_i) \\ \mathcal{L}_2 &= \frac{1}{n} \cdot \sum_{i}^{n} \log y_i^{(1)} + \log(1 - \hat{y_i}^{(1)}) \\ \mathcal{L} &= \mathcal{L}_1 + \mathcal{L}_2 \end{aligned} \tag{3}$$

where $(Z_i^s)_i$ is $\boldsymbol{v_i}$ corresponding node feature in Z_i^s as shown in Fig. 1. Instead of contrasting with the subgraph features $\boldsymbol{s_i}$, we use $(Z_i^s)_i$ to construct positive pairs $((Z_i^s)_i, Z_i)$ and negative pairs $((Z_j^s)_j, Z_i)$ in node-node scale. Due to subgraph $\mathcal{G}_i$ masked $\boldsymbol{v_i}$ feature with 0, $(Z_i^s)_i$ can be treated as the masked node feature of $\boldsymbol{v_i}$, which have high consistency with Z_i. We supposed that each graph $\mathcal{G}$ can generate three type views of node $\boldsymbol{v_i}$, **subgraph features $\boldsymbol{s_i}$, node features Z_i, and masked node features** $(Z_i^s)_i$, as shown in Fig. 1. It's natural to consider that subgraph-subgraph contrast shall be a promising idea for modeling more complex interaction patterns. GRADATE [4] constructed multi-scale contrasts, including node-node, subgraph-node, subgraph-subgraph. To capture more comprehensive graph level representations, Luo et.al [17] designed a new graph level evaluation metrics and built an end-to-end anomaly detection framework based on contrastive learning. GCCAD [2] de-

signed a graph corrupting strategy and removed suspicious links during message passing to increase the negative instances.

2.4 Ensemble Model

An intuitive idea comes that since anomaly detection benefits from both reconstruction and contrastive methods, taking advantage of the both shall yield a better result. SL-GAD [33] was composed of generative attribute regression and multi-view contrastive learning modules to capture the anomalies. Differently, Mul-GAD [15] utilized redundancy reduction techniques to eliminate the harms of similar information generated by multi-view modeling, which achieve satisfactory performance in the semi-supervised setting.

3 Methodology

In this section, we would detail the used GNN backbone, graph augmentation, different contrast patterns, the MAG framework, and ending with a time complexity analysis for our model. We first construct a GNN backbone, where the graph data $\mathcal{G}(V, X, A)$ passes through the specified GNN backbone to obtain three types of node views. To achieve contrastive learning, we obtain more node views using graph augmentation. Then, construct different contrast loss functions for different pairs of node views and weight them together as a final loss function. The whole process can be regraded as an instance of our MAG framework.

3.1 Preliminary

We formulate the specified GNN backbone used in our MAG framework, which is well-known as graph convolutional network (GCN [9]). The message propagation of its l-th layers can be formulated as follows:

$$x_i^{(l)} = f_{relu}\left(\sum_{v_j \in \{v_i\} \cup \mathcal{N}(v_i)} a_{i,j} W^{(l)} x_j^{(l-1)}\right) \tag{4}$$

where $x_i^{(l)}$ is the l-th layer representation of node v_i and the $\mathcal{N}(v_i)$ denotes the collection of the v_i neighbors. The $a_{i,i}$ is the entry (i, j) of the $\hat{A}$, $\hat{A} = D^{-\frac{1}{2}}\overline{A}D^{-\frac{1}{2}}, \overline{A} = A + I_n, D_{i,i} = \sum_j \overline{A}_{i,j}$. $f_{relu}(x) = max(0, x)$ is the non-linear activation function to empower the model with non-linear modeling capability.

3.2 Graph Augmentation

3.2.1 Feature Augmentation

Supposing p is the probability of the node attribute being masked, $\boldsymbol{m} \in \{0, 1\}^d$ adhered to the Bernoulli distribution $\boldsymbol{m} \sim \mathcal{B}(d, 1-p)$. A augmented feature $\hat{X}$ can be computed as follows.

$$\begin{aligned} \hat{X}_i &= X_i \odot \boldsymbol{m}, i = 1, 2...n \\ \hat{X} &= concat(\hat{X}_1, ..., \hat{X}_n) \end{aligned} \tag{5}$$

where $\odot$ denotes the element-wise product between two vectors.

3.2.2 Structure Augmentation

The random edge perturbation [29, 30] is one of the typically structure augmentation methods. Assuming p is the ratio of perturbed edges. We specify the $\hat{A}$ as:

$$\hat{A} = A \odot (1 - L) + (1 - A) \odot L \tag{6}$$

where $\odot$ is element-wise multiplication and $L \in \mathcal{R}^{n \times n}$ denotes a pertubation location matrix where $L_{i,j} = L_{j,i} = 1$ if node v_i and v_j would be perturbed. In a undirected graph, the number of perturbed edges equals to the half of $\sum_{i,j}^{n} A_{i,j}$. The p can be calculated as $\sum_{i,j}^{n} L_{i,j} / \sum_{i,j}^{n} A_{i,j}$. Besides edge perturbation, edge diffusion [6, 8] updates the structure via generating a different topological view. We applied two frequently used edge diffusion methods in this paper, which is Personalized PageRank (PPR) and Heat Kernel (HK). PPR is an extension of the classic PageRank [18] algorithm, originally developed by Google for ranking web pages. PPR assigns a probability distribution to each node, indicating the likelihood of a random walk starting from a specific node and landing on target node in the graph. By controlling the teleportation probability, PPR can personalize the ranking and measure the influence of nodes. Heat Kernel diffusion is based on the heat equation, which models the diffusion of heat over time. In the graph, the heat kernel measures the likelihood of a random walk starting from a node and reaching another node after a specific time. By varying the diffusion time, we can capture different levels of local and global influence within the graph. Their closed-form solutions of PPR and HK can be formulated as:

$$\begin{aligned} \hat{A}^{(PPR)} &= \alpha\left(I - (1-\alpha)D^{-1/2}AD^{-1/2}\right)^{-1} \\ \hat{A}^{(HK)} &= exp\left(tAD^{-1} - t\right) \end{aligned} \tag{7}$$

where α denotes teleport probability in a random walk and t is the diffusion time. D is the degree matrix of adjacency matrix A.

3.3 Multi-scale Contrast

Multi-scale contrast in GCAD can be abstracted as node-node, subgraph-subgraph, and node-subgraph contrasts, which focus on different interaction patterns. By summarising the previous GCAD methods [4, 7, 14, 33], we noticed that graph $\mathcal{G}$ can generate three type views of node $\boldsymbol{v_i}$, **subgraph features** $\boldsymbol{s_i}$, **node features** Z_i, **and masked node features** $(Z_i^s)_i$ as shown in Fig. 1. These basic elements are the foundations to construct different contrast combinations. Given the graph $\mathcal{G}$, we obtain them as follows:

$$\begin{aligned} Z_i^s = \mathcal{F}(\mathcal{G}_i(X_i^s, A_i^s)), \quad Z_i = \mathcal{F}(X_i) \\ \boldsymbol{s_i} = Readout(Z_i^s) \end{aligned} \tag{8}$$

where $\mathcal{F}(\cdot)$ is GNN backbone, such as GCN [9], GAT [27] et al. X_i^s is the neighbours of the node v_i, where $(X_i^s)_i$ is masked with 0. Thus, Z_i is derived from node v_i via GNN mapping, while $\boldsymbol{s_i}$ and $(Z_i^s)_i$ derived from the neighbors of node v_i.

3.3.1 Node-node Contrast

Node features Z_i and masked node features $(Z_i^s)_i$ are utilized in this part.

$$\begin{aligned} y_i = Bilinear((Z_i^s)_i, Z_i), \quad \hat{y_i} = Bilinear((Z_j^s)_j, Z_i) \\ \mathcal{L}_{nn} = \frac{1}{n} \cdot \sum_{i}^{n} \log y_i + \log(1 - \hat{y_i}) \end{aligned} \tag{9}$$

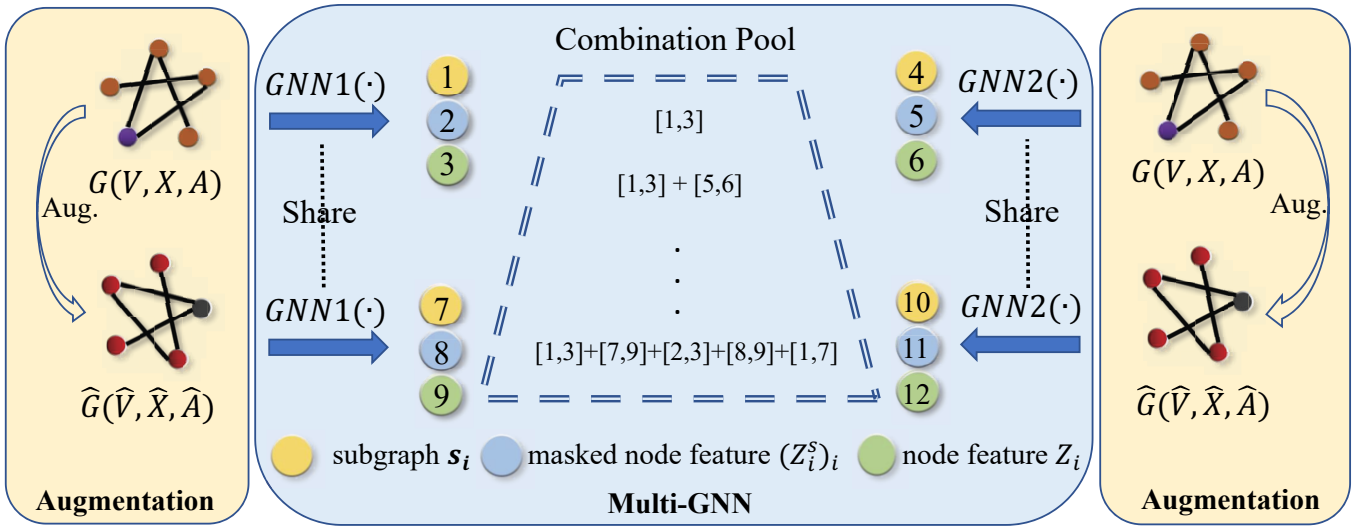

Figure 2. The overview framework of our MAG, which unified the CoLA [14], ANEMONE [7] and GRADATE [4] via contrast combinations from top to bottom in the combination pool. The MAG framework consists of two modules: graph augmentation and multi-GNN modules. The normal node-subgraph, masked node-subgraph, node-node, and subgraph-subgraph contrast pairs correspond to the green-yellow, blue-yellow, green-blue, yellow-yellow pairs, respectively. For example, the [1,3]+[5,6] in the combination pool denote the used of normal node-subgraph pair and node-node pair.

where $(Z_i^s)_i$ is the node v_i corresponding representation in Z_i^s. $Bilinear(\cdot)$ is the bilinear function to obtain the similarity score of the two inputs. Due to $(Z_i^s)_i$ and Z_i derived from the same node, their consistency score y_i is high. Conversely, the consistency between $(Z_j^s)_j$ and Z_i is low. Based on the intuition, we construct loss function $\mathcal{L}_{nn}$ and optimize it.

3.3.2 Subgraph-subgraph Contrast

We increase the subgraph views of node v_i by adding a new GNN mapping $\hat{\mathcal{F}}(\cdot)$. $(\boldsymbol{s_i}, \hat{\boldsymbol{s}}_{\boldsymbol{i}})$ and $(\boldsymbol{s_j}, \hat{\boldsymbol{s}}_{\boldsymbol{i}})$ are employed to build the positive and negative instance pairs.

$$\begin{aligned} \hat{\boldsymbol{s}}_{\boldsymbol{i}} &= Readout(\hat{\mathcal{F}}(\mathcal{G}_i(X_i^s, A_i^s))) \\ y_i = Bilinear(\boldsymbol{s_i}, \hat{\boldsymbol{s}}_{\boldsymbol{i}}), &\quad \hat{y}_i = Bilinear(\boldsymbol{s_j}, \hat{\boldsymbol{s}}_{\boldsymbol{i}}) \\ \mathcal{L}_{ss} &= \frac{1}{n} \cdot \sum_i^n \log y_i + \log(1 - \hat{y}_i) \end{aligned} \tag{10}$$

where $\boldsymbol{s_i}$, $\hat{\boldsymbol{s}}_{\boldsymbol{i}}$ form the position instance pairs, while $\boldsymbol{s_j}$, $\hat{\boldsymbol{s}}_{\boldsymbol{i}}$ are regarded as negative instance pairs.

3.3.3 Node-subgraph Contrast

There are two expressions for node-subgraph contrasts. For identification, we call *normal node-subgraph contrast* if used Z_i, *masked node-subgraph contrast* if used $(Z_i^s)_i$.

$$\begin{aligned} y_i = Bilinear(\boldsymbol{s_i}, Z_i), &\quad \hat{y}_i = Bilinear(\boldsymbol{s_j}, Z_i) \\ \mathcal{L}_{ns}^n &= \frac{1}{n} \cdot \sum_i^n \log y_i + \log(1 - \hat{y}_i) \end{aligned} \tag{11}$$

where Z_i denotes the normal node features, while $(Z_i^s)_i$ below refers to the masked feature derived only from node v_i neighbours.

$$\begin{aligned} y_i = Bilinear(\boldsymbol{s_i}, (Z_i^s)_i), &\quad \hat{y}_i = Bilinear(\boldsymbol{s_j}, (Z_i^s)_i) \\ \mathcal{L}_{ns}^m &= \frac{1}{n} \cdot \sum_i^n \log y_i + \log(1 - \hat{y}_i) \end{aligned} \tag{12}$$

3.3.4 Inference Phase

In the training, the whole networks are updated via optimizing the contrastive loss function. In the inference stage, we obtain the consistency scores of positive and negative pairs of node v_i, y_i and $\hat{y}_i$. For the normal nodes, the predicted score of positive instance pairs y_i tended to 1, while the negative pairs $\hat{y}_i$ were closed to 0. For the anomalous node, both of the y_i and $\hat{y}_i$ are closed to 0.5, which means that its positive and negative pairs would be less discriminative. Thus, the anomaly score can be computed as $(\hat{y}_i - y_i)$. Following the [7, 14, 33], we sampled R rounds to obtain the mean and standard derivation for stability. The procedure can be formulated as follows:

$$\begin{aligned} f_1(v_i) &= \frac{\sum_{r=1}^R (\hat{y}_i^{(r)} - y_i^{(r)})}{R} = \overline{x} \\ f_2(v_i) &= \sqrt{\sum_{r=1}^R ((\hat{y}_i^{(r)} - y_i^{(r)}) - \overline{x})^2 / R} = s \\ f(v_i) &= \overline{x} + s \end{aligned} \tag{13}$$

where $f(v_i)$ is the final anomaly score for node v_i, which denotes the sum of the mean and standard derivation. Each node's subgraph is obtained through random walks. We employ multiple sampling iterations to mitigate the potential bias introduced by a single sampling instance on the model training. R is the number of times we sampled. We use R to avoid the impact of randomness. As elucidated in the aforementioned study [7, 14], the detection performance become more stable with the rising of R. However, larger R also leads to longer computational time. One potential trade-off is to set R to 256, which could obtain stable result and avoid large computational costs, following [7, 14].

3.4 MAG Framework

As shown in Fig. 2, each graph generates three views for node v_i, which is subgraph feature $\boldsymbol{s_i}$ (yellow), masked node feature $(Z_i^s)_i$

(blue), and node feature Z_i (green). We increase the graph views via the graph augmentation and multi-GNN modules. The augmented graph $\hat{\mathcal{G}}$ share the training parameters with the original graph $\mathcal{G}$. The graph convolutional network (GCN [9]) is used as GNN backbone in our framework. These views can be combined as the positive or negative instance pairs and further establish the contrastive loss function. In the combination pool, [1,3] form normal node-subgraph contrast pairs. [1,3]+[5,6] added the additional node-node contrast pairs to model complex interactive pattern. In our unified framework, different combinations are implemented by adjusting hyper-parameters, which is simple and flexible. Following the formula in section 3.3, the [1,3]+[5,6] is implemented as follows:

$$\begin{aligned} \mathcal{L} &= \alpha\mathcal{L}_{ns}^{n} + \beta\mathcal{L}_{nn} \\ f_{all}(v_i) &= \alpha f_{L_{ns}^{n}}(v_i) + \beta f_{L_{nn}}(v_i) \end{aligned} \tag{14}$$

where the α and β are the balance factors to weigh different contrastive loss. In the inference stage, we obtain $f_{all}(v_i)$ as our final anomaly score. $f_{L_{ns}^{n}}(v_i)$ and $f_{L_{nn}}(v_i)$ can be obtained according to the formula 13. The same process applied to the three or more combinations. As shown in Fig. 2, the three combinations in the combination pool from top to bottom is the prototype of CoLA [14], ANEMONE [7], and GRADATE [4] methods, respectively. We compared the result of our combination with their real algorithm as shown in Table. 2, which show a small margin. Our MAG framework unified the classical GCAD algorithms within limited fluctuation. We further proposed the two variants of MAG, L-MAG and M-MAG. The L-MAG is the prototype of the single combination [4,9], which outperform the existing state-of-the-art on Cora and Pubmed with the low computational cost. For the multiply contrast combinations, the combination of [1,3]+[4,6] (M-MAG model) show better detection performance.

3.5 Complexity Analysis

We compute the time complexity by considering the three main components, which are the graph augmentation, subgraph sampling and the GCN modules. The time complexity of contrast loss function is far less than the other three, so we ignore them. For the L-MAG model, the time complexity of the graph augmentation is $O(|V|+d)\cdot p$, where p is the ratio of perturbed node and edges, $|V|$ is the number of edges. The time complexity of each RWR subgraph sampling is $O(c\sigma)$, where c is the number of nodes in subgraph and σ is the mean degree of graph. We sample R rounds for each node, thus the total time complexity is $O(cn\sigma R)$. For the GCN model, the time complexity is $O(2Lnd^2)$, where L is the layer of GCN model and multiplying by two means that two GNNs are used. The overall time complexity of L-MAG is $O(cn\sigma R+(|V|+d)\cdot p+2Lnd^2)$. Likewise, the time complexity for the M-MAG is $O(2cn\sigma R+2Lnd^2)$ without graph augmentation, but with one additional subgraph sampling. In practice, the subgraph sampling module is more time consuming than GNN module, as the GNN module could be computed by GPU acceleration. For the term of graph augmentation, its computational cost is small and only starts to be perceived as time consuming when the edges or feature dimensions are counted in millions. In our GeForce RTX 3080 case, L-MAG is about twice as fast as M-MAG.

4 Experiments and Results

In this section, we dived into the MAG framework and provided the empirical evidence to demonstrate that our MAG model does unify the classical GCAD algorithm. To gain a deeper understanding, we propose four valuable research questions.

- **RQ1**: Can the MAG framework unify the classical GCAD algorithm?
- **RQ2**: Does the graph augmentation and multi-GNN modules actually work?
- **RQ3**: Is the multi-scale contrast module effective?
- **RQ4**: How is the potential of the MAG framework in single combination condition? Can the final proposed M-MAG surpass the existing methods?

Table 1. Statistics of the datasets. A half-and-half split between structure and contextual anomalies.

Graph	Nodes	Edges	Features	Anomalies
Cora	2,708	5,429	1,433	150
Citeseer	3,327	4,732	3,703	150
Pubmed	19,717	88,648	500	600

4.1 Experimental Setting

4.1.1 Datasets

Following the [6, 14, 28], we use the three popular citation networks, Cora, Citeseer, and Pubmed [21]. The anomalous nodes were generated by perturbing the graph structure and modifying the node features. Thus, the graph networks are composed of structure and contextual abnormal nodes. The injection algorithm follow as [7, 14] and the statistic detail was listed in Table. 1

4.1.2 Baseline

We compare with the classical shallow learning methods, Radar, and ANOMALOUS. DOMINANT and AnomalyDAE are the reconstructed-based methods. The final categories are contrastive-based methods, CoLA, ANEMONE, SL-GAD and GRADATE. For convenience, we achieve the Radar, ANOMALOUS, DOMINANT and AnomalyDAE with a python library for graph outlier detection (PyGOD [13]). The other algorithm will be reproduced using the open source code. It is worth noting that we would set the same hyper-parameters for a fair comparison.

4.1.3 Evaluation

The range of the anomaly score in this paper is not a probability value between [0,1]. Thus, it's not suitable to define a passing line to identify normal or anomaly nodes. The common used accuracy, precision, and recall are not taken into consider. Conversely, the Area Under Curve (AUC) is proper in this case, which will be our evaluation metrics in subsequent experiment.

4.1.4 Parameter Setting

For our MAG, the training epochs and learning rate were set to 100 and 1e-3 for all datasets. The hidden dimension and batch size were set to 64, 300. We sampled 256 rounds in the inference and set the size of sampled subgraph to 4 following [7, 14]. The balance factor was set to (0.3,0.7) for M-MAG model.

Table 2. The average result (AUC/%) of CoLA, ANEMONE, and GRADATE in Cora over three seeds, compared with our corresponding MAG combination in the same hyper-parameters setting.

	CoLA	ANEMONE	GRADATE
Origin	89.5	90.6	90.5
Our MAG	90.3	91.1	89.8
Difference	+ 0.8	+ 0.5	- 0.7

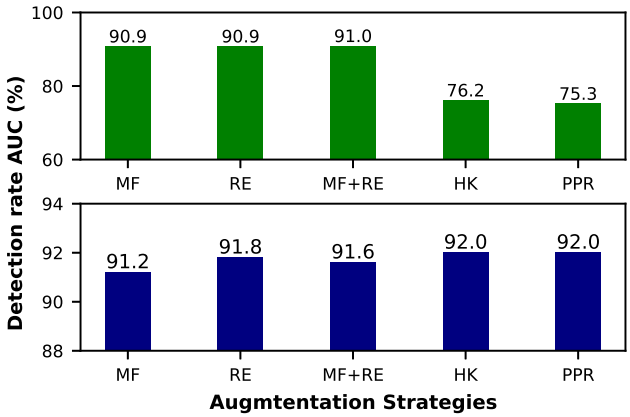

Figure 3. The MF, RE denote the masked feature, removed edge, respectively. The HK and PPR are the typical graph diffusion methods, which are heat kernel and personalized pagerank. The top one use the combination [1,3]+[7,9] of MAG framework, the bottom use [1,3]+[10,12]. Experiments are conducted on the Cora.

4.2 *Experimental Evidence for Unified*

To answer the RQ1, we compared the experimental result with CoLA, ANEMONE, and GRADATE, which are reproduced using the open source code. Their MAG combination correspond to [1,3], [1,3]+[5,6], [1,3]+[7,9]+[2,3]+[8,9]+[1,7]. Our combinations construct the similar contrast loss, while the details of the implementation are not totally same. For a fair comparison, we keep the same hyper-parameters, such as epoch, learning rate, and balance factors. For the graph augmentation module, we use the combination of masked feature and removed edge for our MAG framework, since it show a stable enhancement performance in most cases as shown in Fig. 3. As shown in Table. 2, our MAG combination models have a little margin with the corresponding algorithms, which may be caused by the different graph augmentation strategies and the randomness of the seeds. In fact, we unify GCAD model in the multi-scale contrast module, which is the most important part in GCAD. However, these relatively small margin still indicate the reasonableness of the unified model to a certain extent. Specifically, our model is highly flexible and achieve the combination of varying contrast losses by only altering the hyper-parameters.

4.3 *Benefits of Graph Augmentation and Multiply GNN*

To answer the RQ2, we have compared masked feature, removed edge, masked feature + removed edge, PPR diffusion, HK diffusion for graph augmentation. The ratio of masked and removed is set to 0.2 to increase the modeling difficulties. As shown in Fig. 3, masked feature + removed edge have the most stable performance. We attribute the HK, PPR failures on the top bar to the limitations of the single GNN model. To examine the difference between the single and multiple GNNs, we compare the single and double GNN models. As shown in Table. 3, the origin and M-G denote single and double GNNs respectively. The double one shows a higher detection AUC. We attribute the result to the fewer training parameters and the statistically unstable properties of the single GNN model.

Table 3. The origin, M-S, M-SG and M-G correspond to the combination [1,3], [1,3]+[2,3], [1,3]+[5,6], [1,3]+[4,6] of the MAG framework and denote single-GNN, multi-scale, the combination of multi-scale and multi-GNN, and multi-GNN, respectively. The best detection AUC (%) is in bold and the runner-up is in underline, over three seeds.

	Origin	M-S	M-SG	M-G
Cora	90.3	90.0	<u>90.6</u>	**92.0**
Citeseer	91.2	90.0	<u>92.1</u>	**92.5**

Table 4. The average results (AUC/%) of normal node-subgraph, node-node, subgraph-subgraph, masked node-subgraph contrast in Cora using corresponding single combinations of our MAG framework over three seeds, respectively.

	N-NS	NN	SS	M-NS
Average	90.96	86.03	73.81	69.66

4.4 *Scam of Multi-scale Contrast*

We found that multi-scale modules do not improve the model performance, which multi-GNN modules do. To answer the RQ3, we constructed the origin, M-S, M-G, and M-SG as shown in Table. 3, which denoted single-GNN, multi-scale, multi-GNN, and the combination of multi-scale and multi-GNN. Compared origin with M-S, the additional node-node contrast [2,3] in M-S has no benefit and even causes a corrupt performance. However, the additional node-node contrast [5,6] for M-SG get a better result. In fact, we found that the gains for M-SG derived from multi-GNN modules, not the additional node-node contrast. The result that M-G outperform the M-SG further confirm the statement. Actually, the prototype of the M-SG is the ANEMONE [7] algorithm, which claimed that their improvement is benefited from the additional node-node contrast. They illustrated that the extra node-node contrast was able to model complex interaction patterns, which resulted the better performance. It's a scam of multi-scale modules, the multi-GNN is the hidden pushers. We supposed that the multi-GNN module increased the parameters to be trained, leading a higher degree of freedom for model. The high degree of freedom enable the model to have the potential to capture more sophisticated interaction among nodes. However, we do not negate multi-scale contrasts module due to the good performance of multi-GNN. We believe that the poor performance of multi-scale could be a result of not using a proper integration strategy, i.e. the loss function of one scale should not simply be added or weighed with the loss function of other scale. More complex loss functions need to be designed to fuse the loss function of different scales into one.

4.5 *Single Combination*

Although it is hard to traverse the search space in the multi-combination case, the single combination is feasible. To answer

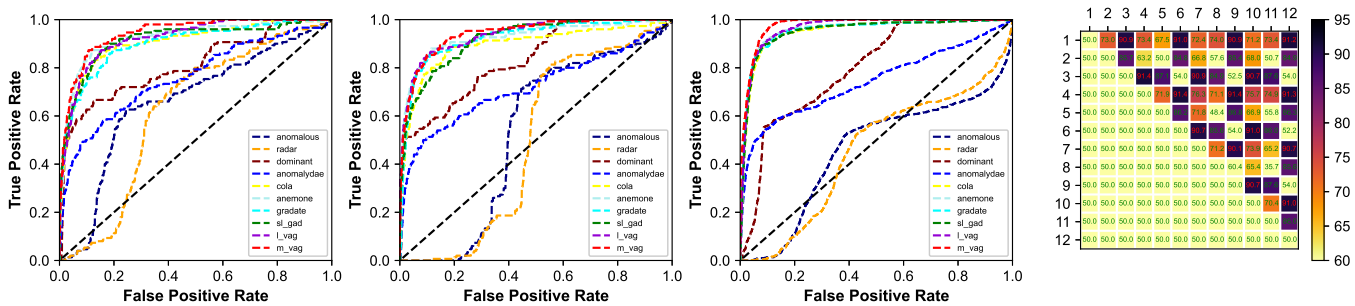

Figure 4. The three roc curves are conducted on Cora, Citeseer, and Pubmed datasets from left to right, respectively. The values in the heat plot denote the detection AUC of different contrast combination. For example, the biggest AUC value 91.4 shows repeatedly in combination [3,4], [4,6], [4,9].

Table 5. Comparison with the existing state-of-the-art. The detecting results AUC(%) over three seeds, on Cora, Citeseer, Pubmed datasets. The best performance method in each experiment is in bold and the runner-up is in underline.

Alg. \ Data.	Cora	Citeseer	Pubmed
Radar [11]	64.8	62.2	54.5
ANOMALOUS [19]	67.8	66.4	54.1
DOMINANT [3]	81.0	83.1	80.5
AnomalyDAE [5]	76.2	72.1	78.8
CoLA [14]	89.8	90.9	95.1
ANEMONE [7]	90.8	<u>91.8</u>	95.4
SL-GAD [33]	91.3	91.7	95.6
GRADATE [4]	90.9	91.6	94.8
L-MAG (Ours)	<u>91.4</u>	<u>91.8</u>	<u>95.7</u>
M-MAG (Ours)	**92.2**	**93.0**	**96.2**

RQ4, we search all the single combination and plot the heat map in AUC detection rate for a clarify observation. As shown the heap map in Fig. 4, the node-subgraph contrast show a excellent performance, which have a average of 90.9%. We have summarised the other contrast patterns in Table. 4. The results illustrate that the ranking of gain in detection AUC by different contrast patterns are normal node-subgraph, node-node, subgraph-subgraph, masked node-subgraph, respectively. It's worth noting that one of the combinations [4, 9] even outperform the existing state-of-the-art in some situations without complex contrast combination, which would be the the lightweight instance of our MAG framework called L-MAG in subsequent experiment.

4.6 Comparison with Existing Methods

To answer the RQ4, we propose two variant models in GCAD field, L-MAG and M-MAG. L-MAG is the combination of [4,9] and M-MAG is the combination of [1,3]+[4,6] As shown in Table. 5 and Fig. 4, our L-MAG outperform the existing model on Cora and Pubmed with a low computational cost, while the M-MAG model further improves detection AUC benefited from the multi-GNN modules.

5 Conclusion

In this paper, we proposed the multi-GNN and augmented GCAD framework MAG. Our MAG framework is able to unify the classical GCAD methods by combining different contrast patterns. The proposed lightweight variant L-MAG outperform the state-of-the-art on Cora and Pubmed with the low computational cost. The variant M-MAG equipped with multi-GNN modules further improve the detection performance. Revisiting the multi-scale contrast and multi-GNN modules, we observed that the ANENONE method benefited from the multi-GNN modules, not the additional node-node contrast. We suggested that multi-scale contrast modules were the surfaced *"puppet"*, while the multi-GNN modules were the real *"pushers"* for complex interaction modeling. For augmentation, the masked feature and removed edge are relatively better options. In the single combination of the MAG, the normal node-subgraph express higher detection AUC than node-node, subgraph-subgraph, and masked node-subgraph contrast. The MAG framework has a vast amount of combinations, which are challenging to traverse thoroughly. Therefore, analysing the deeper mechanisms of multi-scale contrasts and finding a better contrast combinations is a worthwhile subject. Transferring MAG framework to a more realistic scene (e.g. heterogeneous or dynamic graph) also deserves more attention.

Acknowledgements

This work was supported by the National Natural Science Foundation of China Enterprise Innovation and Development Joint Fund (No. U19B2044) and the National Key Research and Development Program of China (No. 2021YFB2700600).

References

[1] Varun Chandola, Arindam Banerjee, and Vipin Kumar, 'Anomaly detection: A survey', *ACM Computing Surveys (CSUR)*, **41**(3), 1–58, (2009).

[2] Bo Chen, Jing Zhang, Xiaokang Zhang, Yuxiao Dong, Jian Song, Peng Zhang, Kaibo Xu, Evgeny Kharlamov, and Jie Tang, 'Gccad: Graph contrastive learning for anomaly detection', *IEEE Transactions on Knowledge and Data Engineering*, (2022).

[3] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu, 'Deep anomaly detection on attributed networks', in *Proceedings of the 2019 SIAM International Conference on Data Mining*, pp. 594–602. SIAM, (2019).

[4] Jingcan Duan, Siwei Wang, Pei Zhang, En Zhu, Jingtao Hu, Hu Jin, Yue Liu, and Zhibin Dong, 'Graph anomaly detection via multi-scale contrastive learning networks with augmented view', *arXiv preprint arXiv:2212.00535*, (2022).

[5] Haoyi Fan, Fengbin Zhang, and Zuoyong Li, 'Anomalydae: Dual autoencoder for anomaly detection on attributed networks', in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5685–5689. IEEE, (2020).

[6] Kaveh Hassani and Amir Hosein Khasahmadi, 'Contrastive multi-view representation learning on graphs', in *International Conference on Machine Learning*, pp. 4116–4126. PMLR, (2020).

[7] Ming Jin, Yixin Liu, Yu Zheng, Lianhua Chi, Yuan-Fang Li, and Shirui Pan, 'Anemone: Graph anomaly detection with multi-scale contrastive learning', in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 3122–3126, (2021).

[8] Zekarias T Kefato and Sarunas Girdzijauskas, 'Self-supervised graph neural networks without explicit negative sampling', *arXiv preprint arXiv:2103.14958*, (2021).

[9] Thomas N Kipf and Max Welling, 'Semi-supervised classification with graph convolutional networks', *arXiv preprint arXiv:1609.02907*, (2016).

[10] Thomas N Kipf and Max Welling, 'Variational graph auto-encoders', *arXiv preprint arXiv:1611.07308*, (2016).

[11] Jundong Li, Harsh Dani, Xia Hu, and Huan Liu, 'Radar: Residual analysis for anomaly detection in attributed networks.', in *International Joint Conference on Artificial Intelligence*, volume 17, pp. 2152–2158, (2017).

[12] Zhida Li, Ana Laura Gonzalez Rios, and Ljiljana Trajković, 'Machine learning for detecting anomalies and intrusions in communication networks', *IEEE Journal on Selected Areas in Communications*, **39**(7), 2254–2264, (2021).

[13] Kay Liu, Yingtong Dou, Yue Zhao, Xueying Ding, Xiyang Hu, Ruitong Zhang, Kaize Ding, Canyu Chen, Hao Peng, Kai Shu, George H. Chen, Zhihao Jia, and Philip S. Yu, 'Pygod: A python library for graph outlier detection', *arXiv preprint arXiv:2204.12095*, (2022).

[14] Yixin Liu, Zhao Li, Shirui Pan, Chen Gong, Chuan Zhou, and George Karypis, 'Anomaly detection on attributed networks via contrastive self-supervised learning', *IEEE Transactions on Neural Networks and Learning Systems*, **33**(6), 2378–2392, (2021).

[15] Zhiyuan Liu, Chunjie Cao, and Jingzhang Sun, 'Mul-gad: a semi-supervised graph anomaly detection framework via aggregating multi-view information', *arXiv preprint arXiv:2212.05478*, (2022).

[16] Wai Weng Lo, Siamak Layeghy, and Marius Portmann, 'Inspection-l: Practical gnn-based money laundering detection system for bitcoin', *arXiv preprint arXiv:2203.10465*, (2022).

[17] Xuexiong Luo, Jia Wu, Jian Yang, Shan Xue, Hao Peng, Chuan Zhou, Hongyang Chen, Zhao Li, and Quan Z Sheng, 'Deep graph level anomaly detection with contrastive learning', *Scientific Reports*, **12**(1), 19867, (2022).

[18] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd, 'The pagerank citation ranking: Bringing order to the web.', Technical report, Stanford infolab, (1999).

[19] Zhen Peng, Minnan Luo, Jundong Li, Huan Liu, Qinghua Zheng, et al., 'Anomalous: A joint modeling approach for anomaly detection on attributed networks.', in *International Joint Conference on Artificial Intelligence*, pp. 3513–3519, (2018).

[20] Tahereh Pourhabibi, Kok-Leong Ong, Booi H Kam, and Yee Ling Boo, 'Fraud detection: A systematic literature review of graph-based anomaly detection approaches', *Decision Support Systems*, **133**, 113303, (2020).

[21] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad, 'Collective classification in network data', *AI Magazine*, **29**(3), 93–93, (2008).

[22] Chaoming Song, Shlomo Havlin, and Hernan A Makse, 'Self-similarity of complex networks', *Nature*, **433**(7024), 392–395, (2005).

[23] Jianheng Tang, Fengrui Hua, Ziqi Gao, Peilin Zhao, and Jia Li, 'Gadbench: Revisiting and benchmarking supervised graph anomaly detection', *arXiv preprint arXiv:2306.12251*, (2023).

[24] Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li, 'Rethinking graph neural networks for anomaly detection', in *International Conference on Machine Learning*, pp. 21076–21089. PMLR, (2022).

[25] Milind Tiwari, Adrian Gepp, and Kuldeep Kumar, 'A review of money laundering literature: the state of research in key areas', *Pacific Accounting Review*, (2020).

[26] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan, 'Fast random walk with restart and its applications', in *Sixth International Conference on Data Mining (ICDM'06)*, pp. 613–622. IEEE, (2006).

[27] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio, 'Graph attention networks', *arXiv preprint arXiv:1710.10903*, (2017).

[28] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm, 'Deep graph infomax.', *International Conference on Learning Representations (Poster)*, **2**(3), 4, (2019).

[29] Lirong Wu, Haitao Lin, Cheng Tan, Zhangyang Gao, and Stan Z Li, 'Self-supervised learning on graphs: Contrastive, generative, or predictive', *IEEE Transactions on Knowledge and Data Engineering*, (2021).

[30] Yaochen Xie, Zhao Xu, Jingtun Zhang, Zhengyang Wang, and Shuiwang Ji, 'Self-supervised learning of graph neural networks: A unified review', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2022).

[31] Vinod Yegneswaran, Paul Barford, and Johannes Ullrich, 'Internet intrusions: Global characteristics and prevalence', *ACM SIGMETRICS Performance Evaluation Review*, **31**(1), 138–147, (2003).

[32] Ge Zhang, Jia Wu, Jian Yang, Amin Beheshti, Shan Xue, Chuan Zhou, and Quan Z Sheng, 'Fraudre: Fraud detection dual-resistant to graph inconsistency and imbalance', in *2021 IEEE International Conference on Data Mining (ICDM)*, pp. 867–876. IEEE, (2021).

[33] Yu Zheng, Ming Jin, Yixin Liu, Lianhua Chi, Khoa T Phan, and Yi-Ping Phoebe Chen, 'Generative and contrastive self-supervised learning for graph anomaly detection', *IEEE Transactions on Knowledge and Data Engineering*, (2021).