Magnitude-Contrastive Network for Unsupervised Graph Anomaly Detection

Fei Ge[1], Ji Zhang[2(✉)], Zhen Wang[3], Yuqian Zhou[1], and Zhao Li[4]

[1] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
{gefei12,zhouyuqian}@nuaa.edu.cn
[2] The University of Southern Queensland, Darling Heights, Australia
Ji.Zhang@usq.edu.au
[3] Zhejiang Lab, Hangzhou, China
wangzhen@zhejianglab.com
[4] Hangzhou Yugu Technology, Hangzhou, China

Abstract. Effectively identifying anomalous nodes within networks is crucial for various applications, such as fraud detection, network intrusion prevention, and social network activity monitoring. Existing graph anomaly detection methods based on contrastive learning have shown promising results but often suffer from limitations due to their local focus, such as community blindness (overlooking the inherent community structure of networks), limited anomaly scope (focusing solely on the local perspective), and high computational cost. To address these challenges, we propose a novel and efficient graph anomaly detection method called MCEN-GAD, which leverages a multi-level contrastive learning approach to identify anomalies across multiple network dimensions. Specifically, MCEN-GAD incorporates three contrastive networks: a patch-level contrastive network for local anomaly detection, a community-level contrastive network for identifying anomalies within specific communities, and a global-level anomaly detection network for exploring more global anomalous information. MCEN-GAD integrates the anomaly scores from these three levels using a weighted sum approach, achieving a comprehensive understanding of anomalous activity within the network. This multi-level integration allows MCEN-GAD to effectively capture anomalies across different network dimensions and provide a more robust anomaly detection framework. The experimental results clearly demonstrate the remarkable effectiveness and efficiency of our method compared with the state-of-the-art approaches on six benchmark datasets.

Keywords: Anomaly Detection · Contrastive Learning · Unsupervised Learning

W. Zhang et al. (Eds.): APWeb-WAIM 2024, LNCS 14964, pp. 481–493, 2024.
https://doi.org/10.1007/978-981-97-7241-4_30

1 Introduction

Anomaly detection is a data mining process aimed at identifying uncommon patterns within datasets [16]. While traditional methods use feature vectors to detect outliers [18,20], they often overlook the relationships between objects, which can limit their effectiveness. Recently, graph anomaly detection has become prominent [10] as it represents real-world objects and their interconnections as nodes and edges, capturing essential structural information. This approach allows graph-based anomaly detection methods to identify anomalies that might not be obvious when only considering individual objects or their attributes. Currently, these methods have been effectively applied in various domains, including finance [3,19], network security [1], and social media [11,21], demonstrating their potential in preventing various harmful events.

Supervised learning can effectively differentiate between normal and anomalous patterns using labeled data, achieving high accuracy in anomaly detection. However, the lack of prior knowledge about anomalies and the extensive effort needed to label data make these techniques less suitable for graph anomaly detection. In contrast, unsupervised methods emerge as more appropriate for this task, as they do not rely on labeled data and are capable of uncovering patterns and identifying anomalies based solely on the inherent information of the graph itself. Recent advancements have seen the rise of many unsupervised approaches [4,5,7,12,15], including self-supervised methods like contrastive learning, which excels by modeling the relationships between samples to create more informative representations. This allows contrastive learning to more effectively distinguish between normal and anomalous patterns in graphs, enhancing anomaly detection performance.

Although these methods have shown promising results, current contrastive learning-based approaches [6,8,14,17,22,24] typically assess anomalies from a local perspective, presenting several issues that need to be addressed: (1) These methods overlook the significance of community structures, thus posing the risk of incorrectly labeling two nodes from the same community as negative pairs. However, community structures are quite common across many real-world networks. For example, in social networks, users sharing similar interests typically have close connections, whereas those with differing interests tend to maintain a certain distance from each other. Therefore, users within the same interest community possess similar characteristics, and identifying them as negative pairs would significantly weaken the model's ability to recognize normal patterns. (2) Due to complex interactions between structures and attributes, anomalous nodes display diversity and manifest across different scales. For instance, a user in a social network who posts gambling content in a community focused on healthy living would be deemed a community anomaly due to the deviation from the community's theme and norms. Globally, a node with extensive connections to unrelated communities could indicate an abnormal role, such as serving as a 'middleman' in a botnet. Relying solely on local data for anomaly detection often leads to suboptimal results. Thus, it is crucial to develop a comprehensive approach that assesses graph anomalies from multiple dimensions. (3) Moreover,

these models often require numerous iterations of batch sampling during both training and testing phases, as well as multiple rounds of anomaly score calculation. Consequently, the runtime rises rapidly with the size of the dataset and the number of training and evaluation rounds, leading to substantial time costs.

To address these challenges, we propose a **Ma**ni**G**nitude-**C**ontrastiv**E** framework for Unsupervised **G**raph **A**nomaly **D**etection, named MGCE-GAD. This model is capable of effectively capturing anomalous nodes at different levels while maintaining efficient runtime performance. Specifically, we initially construct multi-level contextual information, enabling MGCE-GAD to detect anomalies from multiple perspectives by considering local neighborhood patterns, community structures, and global attribute distributions. Following this, MCEN-GAD learns normal patterns through an elaborately designed multi-level contrastive learning networks for patch-level and community-level anomaly detection and incorporates a novel unsupervised anomaly detection approach for detecting global anomalies. Finally, we evaluate the anomaly scores of each node by jointly considering the patch-level, community-level, and global-level anomaly scores, comprehensively capturing anomalies. Experimental results on six datasets demonstrate the effectiveness of our method, while also exhibiting competitive efficiency. In summary, our main contributions are as follows:

- We propose MGCE-GAD, an effective and efficient magnitude-contrastive learning framework that comprehensively captures anomalous nodes using a multi-level contrastive approach and a novel unsupervised anomaly detection method.
- We develop a novel unsupervised approach for detecting global anomalies. Once trained, we can quantify the anomaly scores of each node by simply comparing the nodes' attributes and structural properties with the overall patterns observed in the network.
- We compare MGCE-GAD with SOTA benchmarks on six datasets. The results show that MGCE-GAD significantly improves accuracy while maintaining competitive runtime efficiency, making it a valuable tool for anomaly detection in large-scale graphs.

The rest of this paper is laid out as follows: Sect. 2 reviews the related work. The research problem of this study is defined in Sect. 3.1. Our proposed approach is detailed in Sect. 3.2. Section 4 discusses the experimental results, and the paper is concluded in Sect. 5.

2 Related Work

Unsupervised deep learning methods can learn the intrinsic structures and patterns from the graph itself, making them highly suitable for scenarios where labels are scarce. These methods can be categorized into generative methods and contrastive methods [13]. Generative methods primarily use graph autoencoders and identify anomalies by comparing the reconstruction errors between original and reconstructed networks. Models like DOMINANT [5] encode and

decode both structural and attribute matrix, calculating anomaly scores based on these errors. AnomalyDAE [7] further considers interactions between structure and attributes using dual autoencoders. SpaceAE [12] combines spectral autoencoders with density estimation for anomaly detection, while ComGA [15] enhances detection by integrating community representations into the GCN layer. AS-GAE [23] designs a location-aware graph autoencoder and uses a supermodular function to assess anomalies. Although these methods can capture biased patterns through reconstruction errors, their effectiveness largely depends on the representations learned by the graph autoencoders, resulting in inadequate support for processing complex real-world data and potential anomaly patterns.

Unlike generative anomaly detection methods, current contrastive methods measure anomalies by evaluating the similarity between target nodes and their neighbors. CoLA [14] is the first to apply contrastive learning to the task of graph anomaly detection by introducing "node-subgraph" contrast. It pairs the target node with its partial neighboring substructure as positive instances, and contrasts it with other substructures as negative instances. ANEMONE [8] introduces "node v.s. node" contrastive strategy to capture more anomaly information. GRADATE [6] adds subgraph-subgraph contrast with augmented view. SL-GAD [24] and Sub-CR [22] jointly detect anomalies by combining node-subgraph contrast and node attribute reconstruction. The above methods employ a strategy of batch training of nodes and multiple rounds of evaluation, which significantly extend the duration needed for both training and testing, consequently imposing a notable cost in terms of time efficiency. Subsequently, PREM [17] simplifies the detection process by calculating neighbor features through a preprocessing module and using a linear discriminator. These methods focus on detecting anomalous nodes with local information, neglecting both the presence of anomalies in diverse amplitude spaces and the significant roles of community structures and global information in graph anomaly detection.

3 Methodology

This section gives the overall introduction including the definition of the problem and a detailed description of our models.

3.1 Problem Definition

Consider an attribute network $G = (V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ is a set of nodes, with $n = |V|$ representing the number of nodes, and E is a set representing the edges. Let $\mathbf{X} \in \mathbb{R}^{n \times p}$ denote the node attribute matrix, where the $i-th$ row vector $x_i \in \mathbb{R}^p$ is the attribute information for the node v_i. In addition, we define $\mathbf{A} \in \mathbb{R}^{n \times n}$ to represent the adjacency matrix. Specifically, $\mathbf{A}_{ij} = 1$ indicates the presence of an edge connecting nodes v_i and v_j, otherwise $\mathbf{A}_{ij} = 0$.

Given an attributed graph G, the purpose of this paper is to detect nodes that exhibit patterns or features significantly different from the majority of nodes under strictly unsupervised settings.

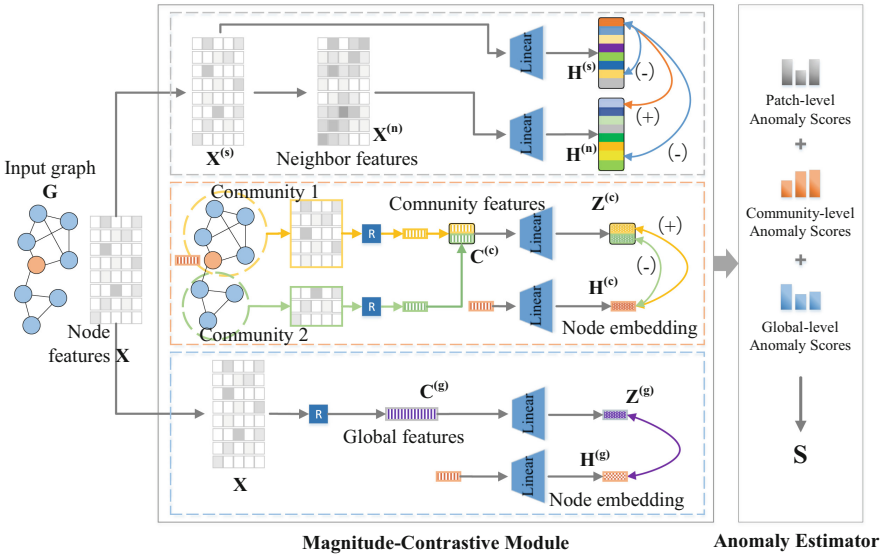

Fig. 1. The framework of MGCE-GAD consists of two main components: the Magnitude-Contrastive module and the Anomaly Estimator. Our objective is to predict the anomaly score of a target node (highlighted in red). Within the Magnitude-Contrastive module, the Patch-level Contrastive Network (depicted with gray dashed lines), the Community-level Contrastive Network (depicted with orange dashed lines), and the Global-level Anomaly Detection Network (depicted with blue dashed lines) respectively learn the consistency between nodes and their neighbor-net, community-net, and global-net. Finally, in the rightmost section, we calculate the anomaly score for each node from three sources. (Color figure online)

3.2 Description of MGCE-GAD

In this section, we introduce our proposed MGCE-GAD. The overall architecture is illustrated in Fig. 1. MGCE-GAD determines the anomaly score by leveraging two primary components: magnitude-contrastive module and Anomaly Estimator. Within the magnitude-contrastive module, the networks learn alignment between nodes and different levels of context, ensuring a comprehensive capture of prevalent patterns. Subsequently, we determine the anomaly score s via a scoring mechanism that consolidates three scores, wherein higher scores indicate the presence of anomalies. In the following subsections, we introduce MGCE-GAD in detail.

Patch-Level Contrast Network. In this module, we establish the local normal patterns of nodes by analyzing the similarity between the nodes and their patch-net. To this end, our initial step involves the formulation of both raw features and neighbor features. Specifically, we leverage the attribute information $\mathbf{X}$ of the nodes as the raw features $\mathbf{X}^{(s)}$, thus preserving the integrity and

originality of node. Furthermore, to acquire the neighbor features, we adopt an anonymous message-passing scheme, which can be defined as follows:

$$\mathbf{X}^{(n)} = \mathbf{M}(\tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2})^k\mathbf{X}, \tag{1}$$

where $\mathbf{M}$ is a self-anonymous mask matrix, with all diagonal elements being 0 and the remaining elements being 1. $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with self-loops. $\tilde{\mathbf{D}}$ is the degree matrix of $\tilde{\mathbf{A}}$. k is the number of message passing steps.

Different from previous methods that require multiple rounds of sampling and aggregating neighbor information through trained graph neural networks, our approach operates on all nodes in one pass and captures the neighbor features through a certain number of message passing steps (k steps), without the need for training, making it more convenient and efficient.

After obtaining the raw features and neighbor features, for the target node v_i, we utilize a linear layer to map its feature vectors $x_i^{(s)}$ and $x_i^{(n)}$ to representations $\mathbf{h}_i^{(s)}$ and $\mathbf{h}_i^{(n)}$, which can be described as follows:

$$\begin{aligned} \mathbf{h}_i^{(s)} &= \mathbf{x}_i^{(s)}\mathbf{W}_1 + \mathbf{b}_1, \\ \mathbf{h}_i^{(n)} &= \mathbf{x}_i^{(n)}\mathbf{W}_2 + \mathbf{b}_2, \end{aligned} \tag{2}$$

where $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{b}_1$, and $\mathbf{b}_2$ are learnable parameters.

To train the discriminative contrastive network, we define raw feature embedding $\mathbf{h}_i^{(s)}$ and neighbor feature embedding $\mathbf{h}_i^{(n)}$ of node v_i as positive pairs, and their similarity $l_i^{(pos)}$ represents the positive score:

$$l_i^{(pos)} = cos(\mathbf{h}_i^{(s)}, \mathbf{h}_i^{(n)}), \tag{3}$$

where cos denotes the cosine similarity function. Besides, $\mathbf{h}_j^{(s)}$ of randomly sampled node v_j and $\mathbf{h}_k^{(n)}$ of randomly sampled node v_k are designated as negative samples, so the negative term in contrastive learning can be expressed as:

$$\begin{aligned} l_i^{(neg_s)} &= cos(\mathbf{h}_i^{(s)}, \mathbf{h}_j^{(s)}), \\ l_i^{(neg_n)} &= cos(\mathbf{h}_i^{(s)}, \mathbf{h}_k^{(n)}). \end{aligned} \tag{4}$$

Finally, we design the following loss function to ensure that normal nodes resemble their corresponding neighbors while distinctly differing from negative samples. Specifically, we first use a linear mapping to project the positive and negative items $l \in [-1, 1]$ to $\hat{l} \in [0, 1]$ and then compute the loss:

$$\mathcal{L}^{(p)} = -\sum_{i=1}^{n}(\log(\hat{l}_i^{(pos)}) + \alpha_1 \log(1 - \hat{l}_i^{(neg_s)}) + \alpha_2 \log(1 - \hat{l}_i^{(neg_n)})), \tag{5}$$

where α_1 and α_2 are two balancing factors used to weigh the importance of two negative samples ($\mathbf{h}_j^{(s)}$ and $\mathbf{h}_k^{(n)}$).

Community-Level Contrast Network. Nodes within the same community usually have similar features, but previous studies [6,8,14,17,22,24] have largely ignored this, focusing instead on local information. This oversight can lead to the misclassification of nodes from the same community as negative pairs, significantly undermining the model's ability to recognize normal patterns. Additionally, the degree of anomaly is closely correlated with the consistency between a target node and other nodes within the same community. Therefore, it's crucial to integrate community information to accurately identify node pairs within the same community and understand community-specific patterns. To achieve this, we use the METIS algorithm [9] with a readout function to extract community information and a contrastive network to learn the community-level coherence.

Specifically, by using the METIS algorithm, we partition the graph G into c communities based on the connections between nodes. Correspondingly, we use the community information matrix $\mathbf{C}^{(c)} \in \mathbb{R}^{c \times p}$ to record the features of each community. $\mathbf{C}^{(c)}[h, :]$ represents the features of the h-th community, which is obtained via a readout function:

$$\mathbf{C}^{(c)}[h, :] = \frac{1}{|V_h|} \sum_{v_i \in V_h} \mathbf{x}_i, \tag{6}$$

where V_h represents the set of nodes in the h-th community, and x_i represents the attribute information of node i.

Similar to the Patch-Level contrastive network, we pass the community features $\mathbf{C}^{(c)}$ and node features $\mathbf{X}$ through different linear layer to obtain low-dimensional embeddings $\mathbf{Z}^{(c)}$ and $\mathbf{H}^{(c)}$, which can be represented as:

$$\begin{aligned} \mathbf{Z}^{(c)} &= \mathbf{C}^{(c)}\mathbf{W}_3 + \mathbf{b}_3, \\ \mathbf{H}^{(c)} &= \mathbf{X}\mathbf{W}_4 + \mathbf{b}_4. \end{aligned} \tag{7}$$

Community anomalies refer to nodes that possess different attribute values compared to other nodes within the same community [16]. Thus, our approach to defining contrastive instance pairs in this segment concentrates on the relationship between nodes and their community structure. Specifically, we introduce a "node versus community-net" instance pairing strategy: a target node paired with its own community network constitutes a positive instance, while pairing it with networks from other communities forms negative pairs.

In order to find inherent normal patterns from a community perspective, we train the community-level contrastive network by maximizing the consistency between the target node and its community-net while minimizing its consistency with others. The learning objective can be described as:

$$\mathcal{L}^{(c)} = -\sum_{i=1}^{n} \log \frac{e^{cos(\mathbf{h}_i^{(c)}, \mathbf{z}_j^{(c)})/\tau}}{\sum_{\mathbf{z}_k^{(c)} \in \mathbf{Z}^{(c)} \setminus \mathbf{z}_j^{(c)}} e^{cos(\mathbf{h}_i^{(c)}, \mathbf{z}_k^{(c)})/\tau}}, \tag{8}$$

where $\mathbf{h}_i^{(c)}$ represents the embedding of the target node v_i, while $\mathbf{z}_j^{(c)}$ is the feature embedding of the community to which v_i belongs, τ is a hyper-parameter.

Global-Level Anomaly Detection Network. Inspired by [2], we delve into the capability of identifying anomalous by leveraging global features in an unsupervised manner. The fundamental assumption is that features of normal nodes tend to exhibit homogeneity, exerting minimal influence on the global features. In contrast, features of anomalous nodes diverge markedly from the norm, thereby having a significant impact on global features. This distinction enables the module to more effectively discern the unique patterns inherent to anomalous nodes. Based on the above assumption, we are empowered to leverage the global features as a reliable tool to differentiate between normal and anomalous nodes. The details of the implementation are outlined below.

Specifically, we first obtain the global features $\mathbf{C}^{(g)}$ through a readout function, which can be represented as:

$$\mathbf{C}^{(g)} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i, \tag{9}$$

where n denotes the number of nodes in the graph.

After that, we employ different linear layer to project the node features and global features into the low-dimensional space, resulting in the acquisition of $\mathbf{H}^{(g)}$ and $\mathbf{Z}^{(g)}$:

$$\begin{aligned}\mathbf{H}^{(g)} &= \mathbf{X}\mathbf{W}_5 + \mathbf{b}_5,\\ \mathbf{Z}^{(g)} &= \mathbf{C}^{(g)}\mathbf{W}_6 + \mathbf{b}_6,\end{aligned} \tag{10}$$

where $\mathbf{H}^{(g)} \in \mathbb{R}^{n\times p}$ and $\mathbf{Z}^{(g)} \in \mathbb{R}^{1\times p}$.

To measure the similarity between nodes and their global-net, we first expand $\mathbf{Z}^{(g)} \in \mathbb{R}^{1\times p}$ to $\mathbf{Z}^{(g)'} \in \mathbb{R}^{n\times p}$. Then, we construct the following loss function to maximize the consistency between nodes and global-net:

$$\mathcal{L}^{(g)} = -cos(\mathbf{H}^{(g)}, \mathbf{Z}^{(g)'}). \tag{11}$$

Loss Function. To simultaneously learn shared patterns at different levels, we optimize the joint loss function:

$$\mathcal{L} = \mathcal{L}^{(p)} + \beta\mathcal{L}^{(c)} + \gamma\mathcal{L}^{(g)}, \tag{12}$$

where β and γ are hyper-parameters that balance components' importance.

Graph Anomaly Scoring. By optimizing the loss function, the target node has a higher matching degree with its positive pairs and a lower one with its negative pairs. At the patch-level, the compatibility between the target node and its positive pair describes the similarity between the node and its neighbors. Based on the homogeneity assumption, we directly use it as a measure of normality in the local perspective, where the anomaly score $s^{(p)}$ for node v_i equals to $-l_i^{(pos)}$. Similarly, at the community-level, the anomaly score $s^{(c)}$ for node v_i is obtained by $-cos(\mathbf{h}_i^{(c)}, \mathbf{z}_j^{(c)})$, where $\mathbf{z}_j^{(c)}$ is the feature embedding of the corresponding community. At the global-level, as analyzed in Sect. 3.2, after training,

a node that is more similar to global features is considered more anomalous. So we obtain the global-level anomaly score $s^{(g)}$ by $cos(\mathbf{H}^{(g)}, \mathbf{Z}^{(g)'})$.

The final anomaly score is the weighted sum of the scores of three levels:

$$s = s^{(p)} + \beta s^{(c)} + \gamma s^{(g)}, \tag{13}$$

where β and γ are shared with Eq. (12).

4 Experiments

4.1 Experimental Setup

Datasets. To evaluate the performance of the proposed method and compare it in various scenarios, we tested it on six different datasets: Citation, CiteSeer, Cora, EAT, Flickr, and WebKB. Since these datasets lack natural anomalies, we injected structural anomalies and attribute anomalies into each dataset, following the methods described in [5,14].

Baselines. In this section, we compare the proposed model with current SOTA benchmarks, including four contrastive learning methods: CoLA [14], ANEMONE [8], GRADATE [6] and PREM [17], as well as SL-GAD [24] and SubCR [22], two hybrid methods that combine generative learning and contrastive learning.

Evaluation Metric. We evaluate our method with the common graph anomaly detection metric, AUC, which ranges from 0 to 1, with higher values denoting better performance. In addition, we also recorded the time needed for training and evaluating different GAD methods.

Implementation Details. We set the number of propagation steps k to 2 for all datasets. On the Citation, CiteSeer, Cora, EAT, Flickr, and WebKB datasets, we set the epochs to 400, 100, 100, 200, 1500 and 200, and set the learning rates to 2e-4, 3e-4, 3e-4, 3e-4, 5e-4, and 1e-4, respectively. The patch-level hyper-parameters α_1 and α_2 are selected from {0.1, 0.2,...,1} and {0.1, 0.2, 0.3, 0.4, 0.5}. In addition, we choose the balance factor β in {1, 10, 20, 50, 100} and γ in {1, 10, 20, 50, 100}. The details of the selection are shown in section ??. For the baseline, we follow the settings in their paper to obtain the reported results.

4.2 Result and Analysis

Accuracy. We evaluated the anomaly detection performance by comparing MGCE-GAD with six baselines. The results recorded in Table 1 demonstrate the superiority of our proposed approach, consistently achieving the best results across all datasets. The primary reason is that, unlike baseline methods that solely focus on the local scale, MGCE-GAD can analyze across three different levels, thereby possessing stronger modeling capability for normal data and achieving superior anomaly detection performance.

Table 1. AUC values (%) on six datasets.

Method	Citation	Citeseer	Cora	EAT	Flickr	WebKB
CoLA	0.7519	0.8944	0.9051	0.6853	0.7466	0.9167
ANEMONE	0.7816	0.9266	0.9206	0.7038	0.7563	0.9238
GRADATE	0.7603	0.8921	0.8973	0.7258	0.7444	0.9326
PREM	0.8655	0.9782	0.9541	0.8207	0.8636	0.9574
SL-GAD	0.7834	0.9235	0.9148	0.6903	0.7888	0.9244
Sub-CR	0.7674	0.9257	0.9068	0.6735	0.7909	0.9029
MGCE-GAD	**0.9213**	**0.9883**	**0.9580**	**0.8743**	**0.8730**	**0.9686**

Efficiency. We evaluated the efficiency of MGCE-GAD by directly comparing its total runtime (training plus testing) with all baselines. Table 2 records the runtime of each model. It is worth noting that the runtime of MGCE-GAD has remained competitive: (1) Compared to its closest competitor PREM, the runtime of MGCE-GAD is comparable to PREM and exhibits significant advantages in the Citation, Flickr and EAT datasets, indicating that our model is more suitable for dense graphs where nodes are closely connected to each other. (2) MGCE-GAD runs in much less time than other methods (CoLA, ANEMONE, GRADATE, SL-GAD and Sub-CR). This is mainly because other methods require multiple rounds of sampling, with the rounds increasing significantly with dataset size, training, and testing iterations. In contrast, our model avoids sampling operations, thus reducing time costs.

Table 2. The total time (s) for training and testing with different GAD models.

Method	Citation			Citeseer			Cora		
	train	test	**total**	train	test	**total**	train	test	**total**
CoLA	3374.4873	2441.0346	5815.5219	379.8038	763.3216	1143.1254	201.8765	554.7965	756.6730
ANEMONE	6244.4578	2295.5294	8539.9872	462.8005	844.5614	1307.3619	245.7012	545.8040	791.5052
GRADATE	1670.3446	1476.9883	3147.3329	1376.8534	1215.7249	2592.5783	1682.0033	1018.9400	2700.9433
PREM	23.0920	0.0040	23.0960	1.0184	0.0010	**1.0194**	1.0648	0.0010	1.0658
SL-GAD	3212.3100	1970.6843	5182.9943	307.6172	1312.9743	1620.5915	159.6807	406.1497	565.8304
Sub-CR	2110.2807	1887.7300	3998.0107	256.4191	1880.2020	2136.6211	147.4676	926.8223	1074.2899
MGCE-GAD	20.1780	0.0141	**20.1921**	1.4315	0.0060	1.4375	1.0437	0.0050	**1.0487**
Method	EAT			Flickr			WebKB		
	train	test	**total**	train	test	**total**	train	test	**total**
CoLA	40.8901	69.7986	110.6887	454.6967	269.1792	723.8759	256.3872	225.5874	481.9746
ANEMONE	8.8649	86.0592	94.9241	500.5351	295.0420	795.5771	275.2657	193.1283	468.3940
GRADATE	56.3361	29.6570	85.9931	772.5452	403.7988	1176.3440	504.5110	331.5318	836.0428
PREM	1.6627	0	1.6627	95.7373	0.0184	95.7557	1.1306	0	**1.1306**
SL-GAD	68.4590	48.0875	116.5465	1339.1638	759.0519	2098.2157	333.5034	142.0280	475.5314
Sub-CR	89.9699	168.5871	258.5570	815.9356	620.7275	1436.6631	211.9687	387.6281	599.5968
MGCE-GAD	1.3040	0.0040	**1.3080**	88.5366	0.0200	**88.5566**	1.3372	0.0040	1.3412

4.3 Ablation Study

In order to verify the effectiveness of the proposed community-level contrast network and global-level anomaly detection network, we conducted ablation experiments. The experimental results are shown in Table 3. NP denotes that only patch-level contrast network are included, i.e. model PREM [17], NP+NC and NP+NG refer to adding community-level and global-level networks to the patch-level network, respectively. NP+NC+NG denotes MGCE-GAD, which consists of three networks.

Observing the experimental results, we can draw the following conclusions. First, our full model excelled on four datasets and performed well on others, confirming its effectiveness. Second, data from the first three rows highlight the significant impact of both community and global-level modules, validating the effectiveness of each component. Third, integrating these modules typically improves performance compared to using them individually, as they help detect anomalies at their respective scales for more comprehensive detection.

Table 3. Results of ablation study.

	Citation	Citeseer	Cora	EAT	Flickr	WebKB
NP	0.8655	0.9782	0.9541	0.8207	0.8636	0.9574
NP+NC	0.8670	0.9834	0.9574	0.8615	0.8664	0.9554
NP+NG	**0.9220**	**0.9885**	0.9571	0.8484	0.8693	0.9657
NP+NC+NG	0.9213	0.9883	**0.9580**	**0.8743**	**0.8730**	**0.9686**

5 Conclusion

This paper focuses on the critical problem of detecting anomalous nodes under unsupervised settings. We propose an effective and efficient novel graph anomaly detection method named MGCE-GAD, which learns normal patterns through a elaborate designed multi-level contrastive learning framework for patch-level and community-level anomaly detection and incorporates a novel unsupervised anomaly detection approach for detecting global anomalies. By jointly considering information from three sources, we provide reasonable anomaly scores for nodes. Extensive experiments on six benchmark datasets demonstrate the superiority of our method over baseline approaches, while also exhibiting competitive runtime efficiency. Future work will focus on extending MGCE-GAD for heterogeneous graphs, adjusting the learning framework to handle varied node and edge types and their distinctive features.

Acknowledgments. This research is supported in part by Natural Science Foundation of China (No. 62172372) and Zhejiang Provincial Natural Science Foundation (No. LTGG24F030003).

References

1. Caville, E., Lo, W.W., Layeghy, S., Portmann, M.: Anomal-e: a self-supervised network intrusion detection system based on graph neural networks. Knowl.-Based Syst. **258**, 110030 (2022)
2. Chen, B., et al.: Gccad: graph contrastive learning for anomaly detection. TKDE (2022)
3. Chen, T., Tsourakakis, C.: Antibenford subgraphs: unsupervised anomaly detection in financial networks. In: KDD, pp. 2762–2770 (2022)
4. Ding, K., Li, J., Agarwal, N., Liu, H.: Inductive anomaly detection on attributed networks. In: IJCAI, pp. 1288–1294 (2021)
5. Ding, K., Li, J., Bhanushali, R., Liu, H.: Deep anomaly detection on attributed networks. In: SDM, pp. 594–602 (2019)
6. Duan, J., Wang, S., Zhang, P., Zhu, E., Hu, J., Jin, H., Liu, Y., Dong, Z.: Graph anomaly detection via multi-scale contrastive learning networks with augmented view. In: AAAI, vol. 37, pp. 7459–7467 (2023)
7. Fan, H., Zhang, F., Li, Z.: Anomalydae: dual autoencoder for anomaly detection on attributed networks. In: ICASSP, pp. 5685–5689 (2020)
8. Jin, M., Liu, Y., Zheng, Y., Chi, L., Li, Y.F., Pan, S.: Anemone: graph anomaly detection with multi-scale contrastive learning. In: CIKM, pp. 3122–3126 (2021)
9. Karypis, G., Kumar, V.: Multilevel graph partitioning schemes. In: ICPP, pp. 113–122 (1995)
10. Kim, H., Lee, B.S., Shin, W.Y., Lim, S.: Graph anomaly detection with graph neural networks: current status and challenges. IEEE Access **10**, 111820–111829 (2022)
11. Li, D., Guo, H., Wang, Z., Zheng, Z.: Unsupervised fake news detection based on autoencoder. IEEE Access **9**, 29356–29365 (2021)
12. Li, Y., Huang, X., Li, J., Du, M., Zou, N.: Specae: spectral autoencoder for anomaly detection in attributed networks. In: CIKM, pp. 2233–2236 (2019)

13. Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., Tang, J.: Self-supervised learning: Generative or contrastive. TKDE **35**(1), 857–876 (2021)
14. Liu, Y., Li, Z., Pan, S., Gong, C., Zhou, C., Karypis, G.: Anomaly detection on attributed networks via contrastive self-supervised learning. TNNLS **33**(6), 2378–2392 (2021)
15. Luo, X., Wu, J., Beheshti, A., Yang, J., Zhang, X., Wang, Y., Xue, S.: Comga: community-aware attributed graph anomaly detection. In: WSDM, pp. 657–665 (2022)
16. Ma, X., Wu, J., Xue, S., Yang, J., Zhou, C., Sheng, Q.Z., Xiong, H., Akoglu, L.: A comprehensive survey on graph anomaly detection with deep learning. TKDE **35**(12), 12012–12038 (2021)
17. Pan, J., Liu, Y., Zheng, Y., Pan, S.: Prem: a simple yet effective approach for node-level graph anomaly detection. In: ICDM, pp. 1253–1258 (2023)
18. Pang, G., Cao, L., Chen, L., Liu, H.: Learning representations of ultrahigh-dimensional data for random distance-based outlier detection. In: KDD, pp. 2041–2050 (2018)
19. Pei, Y., Lyu, F., Van Ipenburg, W., Pechenizkiy, M.: Subgraph anomaly detection in financial transaction networks. In: ICAIF, pp. 1–8 (2020)
20. Wang, Z., Lan, C.: Towards a hierarchical bayesian model of multi-view anomaly detection. In: IJCAI (2020)
21. Yu, S., Ren, J., Li, S., Naseriparsa, M., Xia, F.: Graph learning for fake review detection. FAIA **5**, 922589 (2022)
22. Zhang, J., Wang, S., Chen, S.: Reconstruction enhanced multi-view contrastive learning for anomaly detection on attributed networks. In: IJCAI (2022)
23. Zhang, Z., Zhao, L.: Unsupervised deep subgraph anomaly detection. In: ICDM, pp. 753–762 (2022)
24. Zheng, Y., Jin, M., Liu, Y., Chi, L., Phan, K.T., Chen, Y.P.P.: Generative and contrastive self-supervised learning for graph anomaly detection. TKDE **35**(12), 12220–12233 (2021)