Scilight

Transactions on Graph Intelligence and Network Applications
https://www.sciltp.com/journals/tgina

Article

From Unsupervised to Few-Shot Graph Anomaly Detection: A Multi-Scale Contrastive Learning Approach

Yu Zheng [1], Junjun Pan [1], Yue Tan [1], Ming Jin [1], Yixin Liu [1,*], Lianhua Chi [2,*], Khoa Phan [2] and Yi-Ping Phoebe Chen [2]

[1] School of Information and Communication Technology (ICT), Griffith University, Gold Coast, QLD 4215, Australia
[2] Department of Computer Science and Information Technology, La Trobe University, Melbourne, VIC 3086, Australia
* Correspondence: yixin.liu@griffith.edu.au (Y.L.); l.chi@latrobe.edu.au (L.C.)

How To Cite: Zheng, Y.; Pan, J.; Tan, Y.; et al. From Unsupervised to Few-shot Graph Anomaly Detection: A Multi-Scale Contrastive Learning Approach. *Transactions on Graph Intelligence and Network Applications* **2026**.

Received: 26 May 2026
Revised: 30 June 2026
Accepted: 14 July 2026
Published: 30 July 2026

Abstract: Graph anomaly detection has rapidly advanced in safeguarding graph applications, such as social networks, finance, and e-commerce. However, existing efforts in graph anomaly detection typically only consider the information in a single scale (view), thus inevitably limiting their capability in capturing anomalous patterns in complex graph data. To address this limitation, in this paper, we propose a novel framework, graph ANomaly dEtection framework with Multi-scale cONtrastive lEarning (ANEMONE in short). By using a graph neural network as a backbone to encode the information from multiple graph scales (i.e., graph views), we learn a better representation for nodes in a graph. In maximizing the agreements between instances at both the patch and context levels concurrently, we estimate the anomaly score of each node with a statistical anomaly estimator according to the degree of agreement from multiple perspectives. To further exploit a handful of few-shot ground-truth anomalies that may be collected in real-world applications, we further propose an extended algorithm, ANEMONE-FS, to integrate valuable information in our method. Extensive experiments under purely unsupervised and few-shot settings demonstrate that our proposed method ANEMONE and its variant ANEMONE-FS consistently outperform state-of-the-art algorithms on six benchmark datasets.

Keywords: anomaly detection; self-supervised learning; graph neural networks; few-shot learning

1. Introduction

Graph-structured data has been widely used in many domains due to its strong ability to capture relationships, making it extremely useful for modeling social networks, biology, physics, and traffic, etc [1]. To safeguard critical graph applications in e-commence [2], cyber-security [3], and finance [4], graph anomaly detection (GAD) has drawn increasing attention in the research community. For instance, we can spot fraudulent sellers by jointly considering their properties and behaviors with GAD [5]. Similarly, GAD can help detect abnormal accounts (social bots) that spread misinformation in social networks [6].

Despite its importance, GAD remains a challenging task due to the non-Euclidean nature of graph data [1]. Unlike anomaly detection on tabular data, GAD requires not only understanding the attributes associated with each node, but also incorporating the underlying graph structure. This complexity has imposed significant challenges on GAD. Existing research to address the challenges can be roughly divided into shallow and deep methods. The early shallow methods typically exploit mechanisms such as ego-network analysis [7], residual analysis [8], or CUR decomposition [9]. These methods are conceptually simple, but they may not be able to learn nonlinear representations from complex graph data, leading to a sub-optimal anomaly detection performance. To address this challenge, recent studies have begun to utilize the strong representational capacity of deep learning-based models. For example, DOMINANT and SPECAE [10,11] employ a graph autoencoder (GAE) to learn the non-linear hidden representation, and estimate the anomaly score for each node based on the reconstruction error. While methods achieve

considerable improvements over shallow methods, they fail to effectively capture the contextual information (e.g., subgraph around a node) for anomaly detection, thereby resulting in unsatisfactory performance.

To address the issue, CoLA [12] introduces a contrastive learning paradigm for GAD and demonstrates by training a discriminator to distinguish positive ego-subgraph pairs from negative ones, where negative pairs are constructed using subgraphs sampled from randomly selected nodes. The score predicted by the discriminator is then used as the anomaly score. Although CoLA achieves promising performance, its contrastive learning strategy mainly focuses on local neighborhood structures around ego nodes, limiting its ability to capture higher-level semantic patterns and long-range dependencies across the entire graph. Due to the complexity of graph data, anomalies are often present at multiple scales. For example, in e-commerce applications, fraudulent sellers may interact with only a small number of users or items (i.e., local anomalies), whereas other fraudsters may be embedded within larger communities (i.e., global anomalies). Such multi-scale characteristics call for more fine-grained and intelligent anomaly detection systems.

Moreover, existing GAD methods are designed in a purely unsupervised manner but cannot utilize annotations when available. Despite the cost of expert knowledge, it is often still feasible in real-world scenarios to collect and annotate a small number of anomaly samples. These few-shot samples provide valuable prior information for characterizing rare anomalies and can substantially benefit anomaly detection systems [13]. Unfortunately, most existing GAD methods [10–12] cannot incorporate any supervision, thereby limiting their practicality in real-world applications.

To overcome these limitations, in this paper, we propose a graph ANomaly dEtection framework with Multi-scale cONtrastive lEarning (`ANEMONE` in short) to detect anomalous nodes in graphs. Our key idea is to construct multi-scale views from the original graph and employ contrastive learning at both patch and context levels simultaneously to capture anomalous patterns hidden in complex graphs. Specifically, we first employ two graph neural networks (GNNs) as encoders to learn representations for each node and a subgraph around the target node, respectively. We then construct positive and negative pairs at both the patch and context levels, and formulate a contrastive learning objective that maximizes the similarity between positive pairs while minimizing it between negative pairs. The anomaly score of each node is then estimated using a novel anomaly estimator that leverages the statistics of multi-round contrastive scores. Moreover, our framework is general and flexible, capable of naturally incorporating ground-truth anomalies in few-shot settings. In particular, labeled anomalies are seamlessly integrated into the contrastive learning framework as additional negative pairs at both the patch and context levels, leading to a new few-shot GAD algorithm, `ANEMONE-FS`. Extensive experiments on six benchmark datasets validate the effectiveness of `ANEMONE` for unsupervised GAD, as well as the effectiveness of `ANEMONE-FS` in few-shot settings. The main contributions of this work are summarized as follows:

- We propose a general framework, `ANEMONE`, based on contrastive learning for GAD. Our method exploits multi-scale information at both the patch- and context-levels to capture anomalous patterns hidden in complex graphs.
- We present a simple approach based on the multi-scale framework, `ANEMONE-FS`, to exploit the valuable few-shot anomalies at hand. Our method essentially enhances the flexibility of contrastive learning for anomaly detection and facilitates broader applications.
- We conduct extensive experiments on six benchmark datasets to demonstrate the superiority of our `ANEMONE` and `ANEMONE-FS` for both unsupervised and few-shot GAD.

The remainder of this paper is organized as follows. Section 2 reviews related work, and Section 3.1 presents the problem formulation. Section 3.2 introduces the proposed methods `ANEMONE` and `ANEMONE-FS`. Section 4 reports the experimental results, Section 5 discusses the results, and Section 6 concludes the paper.

2. Related Works

In this section, we survey the representative works in three related topics, including graph neural networks, graph anomaly detection, and contrastive learning.

2.1. Graph Neural Networks

In recent years, graph neural networks (GNNs) have achieved significant success in dealing with graph-related machine learning problems and applications [1,14–16]. Considering both attributive and structural information, GNNs can learn a low-dimensional representation for each node in a graph. Current GNNs can be categorized into two types: spectral and spatial methods. The former type of methods was initially developed on the basis of spectral theory [14,17,18]. Bruna et al. [17] first extend the convolution operation to the graph domain using spectral graph filters. Afterward, ChebNet [18] simplifies spectral GNNs by introducing Chebyshev polynomials as the convolution filter. GCN [14] further utilizes the first-order approximation of the Chebyshev filter to learn node representations more efficiently.

The second type of methods adopts the message-passing mechanism in the spatial domain, which propagates and aggregates local information along edges, to perform a convolution operation [15, 19, 20]. GraphSAGE [19] learns node representations by sampling and aggregating neighborhoods. GAT [15] leverages the self-attention mechanism to assign a weight for each edge when performing aggregation. GIN [20] introduces a summation-based aggregation function to ensure that GNN is as powerful as the Weisfeiler-Lehman graph isomorphism test. For a comprehensive review, we refer readers to the recent survey [1].

2.2. Graph Anomaly Detection

Anomaly detection is a conventional data mining problem aiming to identify anomalous data samples that deviate significantly from others [21]. Compared to detecting anomalies from text/image data [22], graph anomaly detection (GAD) is often more challenging since the correlations among nodes should also be considered when measuring the abnormality of samples. To tackle the challenge, some traditional solutions use shallow mechanisms like ego-network analysis (e.g., AMEN [7]), residual analysis (e.g., Radar [8]), and CUR decomposition (e.g., ANOMALOUS [9]) to model the anomalous patterns in graph data. Recently, deep learning has become increasingly popular for GAD [21]. As an example of unsupervised methods, DOMINANT [10] employs a graph autoencoder model to reconstruct attribute and structural information of graphs, and the reconstruction errors are leveraged to measure node-level abnormality. ADA-GAD [23] extends the idea of reconstruction by incorporating edge pruning to deal with noisy training data. CoLA [12] considers a contrastive learning model that models abnormal patterns via learning node-subgraph agreements. Recently, affinity-based methods [4, 24, 25] directly utilize affinity between node attributes as a cue to spot malicious nodes. Among semi-supervised methods, SemiGNN [26] is a representative method that leverages hierarchical attention to learn from multi-view graphs for fraud detection. GDN [27] adopts a deviation loss to train a GNN for few-shot node anomaly detection. TUNE [28] overcomes the distribution shift between train and test domains brought by the unseen normal category. Apart from the aforementioned methods for attributed graphs, some recent works also target identifying anomalies from dynamic graphs [29, 30] and text-attribute graphs [31–33]. From the graph-level perspective, there has been research on detecting anomalies with the out-of-distribution techniques [34–36]. Recently, generalist GAD has emerged as a popular research topic due to its strong capability to reduce annotation and training costs through a one-for-all paradigm [37–39].

2.3. Graph Contrastive Learning

Originating from visual representation learning [40–42], contrastive learning has become increasingly popular in addressing self-supervised representation learning problems in various areas. In graph deep learning, recent works based on contrastive learning show competitive performance on graph representation learning scenarios [43, 44]. DGI [45] learns by maximizing the mutual information between node representations and a graph-level global representation, which makes the first attempt to adapt contrastive learning in GNNs. GMI [46] jointly utilizes edge-level contrast and node-level contrast to discover high-quality node representations. GCC [47] constructs a subgraph-level contrastive learning model to learn structural representations. GCA [48] introduces an adaptive augmentation strategy to generate different views for graph contrastive learning. MERIT [49] leverages a bootstrapping mechanism and multi-scale contrastiveness to learn informative node embeddings for network data. Apart from learning effective representations, graph contrastive learning is also applied to various applications, such as drug-drug interaction prediction [50] and social recommendation [51].

3. Materials and Methods

3.1. Problem Definition

In this section, we introduce and define the problem of unsupervised and few-shot graph anomaly detection (GAD). Throughout the paper, we use bold uppercase (e.g., $\mathbf{X}$), calligraphic (e.g., $\boldsymbol{\mathcal{V}}$), and lowercase letters (e.g., $\mathbf{x}^{(i)}$) to denote matrices, sets, and vectors, respectively. We also summarize all important notations in Table 1. In this work, we mainly focus on the anomaly detection tasks on attributed graphs that are widely used in real-world applications. Formally speaking, we define attributed graphs and graph neural networks (GNNs) as follows:

Definition 1 (Attributed Graphs)**.** Given an attributed graph $\mathcal{G} = (\mathbf{X}, \mathbf{A})$, we denote its node attribute (i.e., feature) and adjacency matrices as $\mathbf{X} \in \mathbb{R}^{N \times D}$ and $\mathbf{A} \in \mathbb{R}^{N \times N}$, where N and D are the number of nodes and feature dimensions. An attribute graph can also be defined as $\mathcal{G} = (\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{E}}, \mathbf{X})$, where $\boldsymbol{\mathcal{V}} = \{v_1, v_2, \cdots, v_N\}$ and $\boldsymbol{\mathcal{E}} = \{e_1, e_2, \cdots, e_M\}$ are node and edge sets. Thus, we have $N = |\boldsymbol{\mathcal{V}}|$, the number of edges $M = |\boldsymbol{\mathcal{E}}|$, and define $\mathbf{x}_i \in \mathbb{R}^D$ as the attributes of node v_i. To represent the underlying node connectivity, we let $\mathbf{A}_{ij} = 1$ if there exists

Table 1. Summary of important notations.

Symbols	Description
$\mathcal{G} = (\mathbf{X}, \mathbf{A})$	An attributed graph
$\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{E}}$	The node and edge set of $\mathcal{G}$
$\boldsymbol{\mathcal{V}}^{\mathcal{L}}$	The labeled node set where $\boldsymbol{\mathcal{V}}^{\mathcal{L}} \in \boldsymbol{\mathcal{V}}$
$\boldsymbol{\mathcal{V}}^{\mathcal{U}}$	The unlabeled node set where $\boldsymbol{\mathcal{V}}^{\mathcal{U}} \in \boldsymbol{\mathcal{V}}$
$\mathbf{A} \in \mathbb{R}^{N \times N}$	The adjacency matrix of $\mathcal{G}$
$\mathbf{X} \in \mathbb{R}^{N \times D}$	The node feature matrix of $\mathcal{G}$
$\mathbf{x}^{(i)} \in \mathbb{R}^{D}$	The feature vector of v_i where $\mathbf{x}^{(i)} \in \mathbf{X}$
$\boldsymbol{\mathcal{N}}(v_i)$	The neighborhood set of node $v_i \in \mathcal{V}$
$\mathcal{G}_p^{(i)}, \mathcal{G}_c^{(i)}$	Two generated subgraphs of v_i
$\mathbf{A}_{view}^{(i)} \in \mathbb{R}^{K \times K}$	The adjacency matrix of $\mathcal{G}_{view}^{(i)}$ where $view \in \{p, c\}$
$\mathbf{X}_{view}^{(i)} \in \mathbb{R}^{K \times D}$	The node feature matrix of $\mathcal{G}_{view}^{(i)}$ where $view \in \{p, c\}$
$y^{(i)}$	The anomaly score of v_i
$\mathbf{H}_p^{(i)} \in \mathbb{R}^{K \times D'}$	The node embedding matrix of $\mathcal{G}_p^{(i)}$
$\mathbf{H}_c^{(i)} \in \mathbb{R}^{K \times D'}$	The node embedding matrix of $\mathcal{G}_c^{(i)}$
$\mathbf{h}_p^{(i)} \in \mathbb{R}^{1 \times D'}$	The node embeddings of masked node v_i in $\mathbf{H}_p^{(i)}$
$\mathbf{h}_c^{(i)} \in \mathbb{R}^{1 \times D'}$	The contextual embeddings of $\mathcal{G}_c^{(i)}$
$\mathbf{z}_p^{(i)}, \mathbf{z}_c^{(i)} \in \mathbb{R}^{1 \times D'}$	The node embeddings of v_i in patch-level and context-level networks
$s_p^{(i)}, \tilde{s}_p^{(i)} \in \mathbb{R}$	The positive and negative patch-level contrastive scores of v_i
$s_c^{(i)}, \tilde{s}_c^{(i)} \in \mathbb{R}$	The positive and negative context-level contrastive scores of v_i
$\boldsymbol{\Theta}, \boldsymbol{\Phi} \in \mathbb{R}^{D \times D'}$	The trainable parameter matrices of two graph encoders
$\mathbf{W}_p, \mathbf{W}_c \in \mathbb{R}^{D' \times D'}$	The trainable parameter matrices of two bilinear mappings
N^L, N^U, N	The number of labeled, unlabeled, and all nodes in $\mathcal{G}$
K	The number of nodes in subgraphs
D	The dimension of node attributes in $\mathcal{G}$
D'	The dimension of node embeddings
R	The number of evaluation rounds in anomaly scoring

an edge between v_i and v_j in $\boldsymbol{\mathcal{E}}$, otherwise $\mathbf{A}_{ij} = 0$. In particular, given a node v_i, we define its neighborhood set as $\boldsymbol{\mathcal{N}}(v_i) = \{v_j \in \boldsymbol{\mathcal{V}} | \mathbf{A}_{ij} \neq 0\}$.

Definition 2 (Graph Neural Networks)**.** Given an attributed graph $\mathcal{G} = (\mathbf{X}, \mathbf{A})$, a parameterized graph neural network $GNN(\cdot)$ aims to learn the low-dimensional embeddings of $\mathbf{X}$ by considering the topological information $\mathbf{A}$, denoted as $\mathbf{H} \in \mathbb{R}^{N \times D'}$, where D' is the embedding dimensions and we have $D' \ll D$. For a specific node $v_i \in \boldsymbol{\mathcal{V}}$, we denote its embedding as $\mathbf{h}^{(i)} \in \mathbb{R}^{D'}$ where $\mathbf{h}^{(i)} \in \mathbf{H}$.

In this paper, we focus on two different GAD tasks, namely unsupervised and few-shot GAD. Firstly, we define the problem of unsupervised GAD as follows:

Definition 3 (Unsupervised Graph Anomaly Detection)**.** Given an unlabeled attribute graph $\mathcal{G} = (\mathbf{X}, \mathbf{A})$, we intend to train and evaluate a GAD model $\mathcal{F}(\cdot) : \mathbb{R}^{N \times D} \rightarrow \mathbb{R}^{N \times 1}$ across all nodes in $\boldsymbol{\mathcal{V}}$, where we use $\mathbf{y}$ to denote the output node anomaly scores, and $y^{(i)}$ is the anomaly score of node v_i.

For graphs with limited prior knowledge (i.e., labeling information) on the underlying anomalies, we define the problem of few-shot GAD as follows:

Definition 4 (Few-Shot Graph Anomaly Detection)**.** For an attribute graph $\mathcal{G} = (\mathbf{X}, \mathbf{A})$, we have a small set of labeled anomalies $\boldsymbol{\mathcal{V}}^L$ and the rest set of unlabeled nodes $\boldsymbol{\mathcal{V}}^U$, where $|\boldsymbol{\mathcal{V}}^L| \ll |\boldsymbol{\mathcal{V}}^U|$ since it is relatively expensive to label anomalies in the real world so that only a very few labeled anomalies are typically available. Thus, our goal is to learn a model $\mathcal{F}(\cdot) : \mathbb{R}^{N \times D} \rightarrow \mathbb{R}^{N \times 1}$ on $\boldsymbol{\mathcal{V}}^L \cup \boldsymbol{\mathcal{V}}^U$, which measures node abnormalities by calculating their anomaly scores $\mathbf{y}$. It is worth noting that during the evaluation, the well-trained model $\mathcal{F}^*(\cdot)$ is only tested on $\boldsymbol{\mathcal{V}}^U$ to prevent the potential information leakage.

3.2. Methodology

In this section, we introduce the proposed `ANEMONE` and `ANEMONE-FS` algorithms for node-level GAD under unsupervised and few-shot supervised settings. The overall frameworks are illustrated in Figures 1 and 2, and consist of four main components: augmented subgraphs generation, patch-level contrastive network, context-level

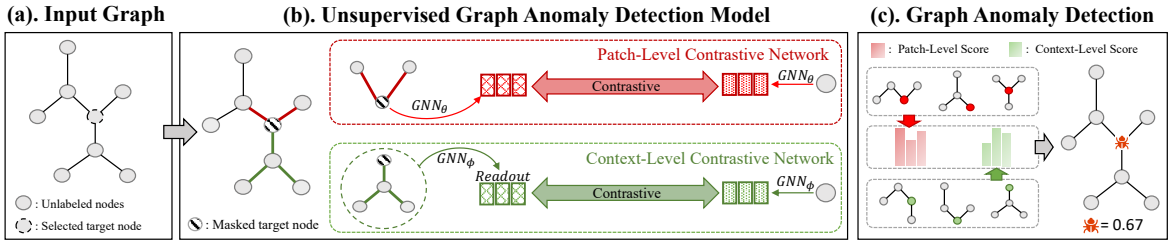


Figure 1. The conceptual framework of ANEMONE. Given an attributed graph $\mathcal{G}$, we first sample a batch of target nodes and construct their corresponding anonymized subgraphs, which are then fed into two complementary contrastive networks. Then, we design a multi-scale (i.e., patch-level and context-level) contrastive network to learn agreements between node and contextual embeddings from different perspectives. During inference, the two contrastive scores are statistically annealed to produce the final anomaly score for each node in $\mathcal{G}$.

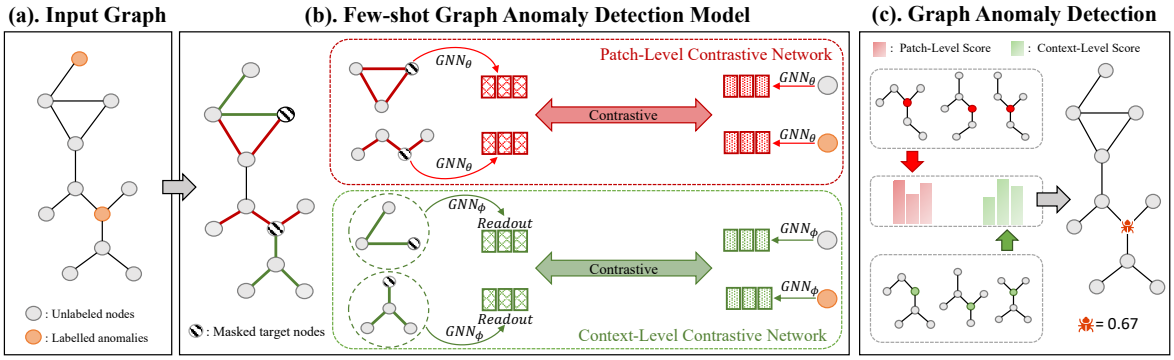


Figure 2. The conceptual framework of ANEMONE−FS follows a pipeline similar to ANEMONE. Given an attributed graph $\mathcal{G}$, we first sample a batch of target nodes. It is then fed into a multi-scale contrastive network equipped with two contrastive routes, where the agreements between node and contextual embeddings are maximized for unlabeled nodes while minimized for labeled anomalies in a mini-batch. During inference, we employ the same graph anomaly scorer as in ANEMONE to estimate the anomaly score of each node in $\mathcal{G}$.

contrastive network, and statistical graph anomaly scorer. Given a target node in an input graph, we first exploit its local context by generating two correlated subgraphs centered at that node. We then propose two general yet effective contrastive learning mechanisms tailored for GAD. Specifically, for an attributed graph without prior knowledge of anomalies, ANEMONE learns patch-level and context-level consistencies by maximizing (1) the mutual information between node embeddings within the patch-level contrastive network and (2) the mutual information between node embeddings and their contextual representations in the context-level contrastive network. Since anomalies are rare, our well-trained model can identify salient structural and attribute mismatches by producing significantly higher contrastive scores for anomalous nodes relative to their surrounding context. Moreover, when a small number of labeled anomalies are available, the proposed ANEMONE−FS variant further leverages this supervision signal to enrich the training objective. By introducing an alternative contrastive pathway, we can minimize the patch-level and context-level similarity for labeled anomalous nodes while still maximizing that for unlabeled nodes, thereby enabling unsupervised and few-shot learning for ANEMONE. Finally, we design a unified graph anomaly scorer that aggregates patch-level and context-level contrastive signals through statistical annealing at inference time. This scorer applies to both unsupervised and few-shot supervised settings, providing a consistent mechanism for anomaly quantification across both regimes.

In the rest of this section, we introduce the four key components of ANEMONE in Sections 3.2.1–3.2.3 and 3.2.5. In particular, Section 3.2.4 details how ANEMONE can be extended to the more powerful ANEMONE−FS by incorporating supervision from a small number of labeled anomalies. Finally, Section 3.2.6 presents the training objectives of both ANEMONE and ANEMONE−FS, along with their optimization procedures, algorithmic implementations, and computational complexity analysis.

3.2.1. Augmented Subgraphs Generation

Graph contrastive learning relies on constructing positive and negative pairs to derive supervision signals from rich attribute and structural information in graphs [43]. Recently, graph augmentations, such as attribute masking, edge modification, subgraph sampling, and graph diffusion, are widely used to enhance the representation learning capability of contrastive models [44,49]. However, not all of them are directly applicable to anomaly detection tasks.

For example, edge modification and graph diffusion can distort the original topological information, thus hindering the model from effectively distinguishing an abnormal node from its surrounding context. To avoid this issue, we adopt an anonymized subgraph sampling mechanism to generate graph views for our contrastive networks. This design is based on two key motivations. First, the agreement between a node and its surrounding context (i.e., its induced subgraph) is often sufficient to characterize its abnormality [52,53]. Second, subgraph sampling provides diverse contextual views for each node, which improves the robustness of both model training and statistical anomaly scoring. Based on these motivations, we introduce the proposed augmentation strategy below.

1. Target node sampling. As this work mainly focuses on node-level anomaly detection, we first sample a batch of target nodes from a given attributed graph. It is worth noting that for a specific target node, its annotation may or may not be available, resulting in different contrastive routes as shown in the middle red and green dashed boxes in Figures 1 and 2. We discuss this in detail in the following subsections.
2. Surrounding context sampling. Although several widely adopted graph augmentations are available [43], most of them are designed to slightly attack the original graph attributes or topological information to learn robust and expressive node-level or graph-level representations, which violate our motivations mentioned above and introduce extra anomalies. Thus, in this work, we employ the subgraph sampling as the primary augmentation strategy to generate augmented graph views for each target node based on the random walk with restart (RWR) algorithm [54]. Taking a target node v_i for example, we generate its surrounding contexts by sampling subgraphs centered at it with a fixed size K, denoted as $\mathcal{G}_p^{(i)} = (\mathbf{A}_p^{(i)}, \mathbf{X}_p^{(i)})$ and $\mathcal{G}_c^{(i)} = (\mathbf{A}_c^{(i)}, \mathbf{X}_c^{(i)})$ for patch-level and context-level contrastive networks. In particular, we let the first node in $\mathcal{G}_p^{(i)}$ and $\mathcal{G}_c^{(i)}$ as the starting (i.e., target) node. In case the target node is isolated, we pad the sequence with the target node itself.
3. Target node anonymization. Although the generated graph views can be directly fed into the contrastive networks, a key limitation arises: the attributes of the target node are involved in computing both its patch-level and context-level embeddings, which may lead to representation shortcuts during multi-level contrastive learning. To address this issue and construct more challenging pretext tasks that improve representation learning [43], we anonymize the target node in each graph view by fully masking its attributes, i.e., $\mathbf{X}_p^{(i)}[1,:] \rightarrow \overrightarrow{0}$ and $\mathbf{X}_c^{(i)}[1,:] \rightarrow \overrightarrow{0}$.

3.2.2. Patch-Level Contrastive Network

The objective of patch-level contrastiveness is to learn the local agreement between the representation of the masked target node v_i and its surrounding context $\mathcal{G}_p^{(i)}$. The underlying intuition is that discrepancies between a node and its directly connected neighbors provide an effective signal for detecting local anomalies, i.e., nodes that deviate from their immediate structural and attribute neighborhood. For instance, in e-commerce fraud detection, fraudulent users often interact with structurally or semantically unrelated neighbors, which can be effectively captured by patch-level contrastive learning in attributed graphs. As shown in Figure 1, our patch-level contrastive network consists of two main components: the graph encoder and contrastive module.

Graph Encoder

The patch-level graph encoder takes a target node and one of its subgraphs (i.e., surrounding context $\mathcal{G}_p^{(i)}$) as the input. Specifically, the node embeddings of $\mathcal{G}_p^{(i)}$ are calculated in below:

$$\begin{aligned}\mathbf{H}_p^{(i)} &= GNN_\theta\left(\mathcal{G}_p^{(i)}\right) = GCN\left(\mathbf{A}_p^{(i)}, \mathbf{X}_p^{(i)}; \mathbf{\Theta}\right) \\ &= \sigma\left(\widetilde{\mathbf{D}_p^{(i)}}^{-\frac{1}{2}} \widetilde{\mathbf{A}_p^{(i)}} \widetilde{\mathbf{D}_p^{(i)}}^{-\frac{1}{2}} \mathbf{X}_p^{(i)} \mathbf{\Theta}\right),\end{aligned} \tag{1}$$

where $\mathbf{\Theta} \in \mathbb{R}^{D \times D'}$ denotes the set of trainable parameters of the patch-level graph neural network $GNN_\theta(\cdot)$. For simplicity and follow [52], we adopt a single layer graph convolution network (GCN) [14] as the backbone encoder, where $\sigma(\cdot)$ denotes the ReLU activation in a typical GCN layer, $\widetilde{\mathbf{A}_p^{(i)}} = \mathbf{A}_p^{(i)} + \mathbf{I}$, and $\widetilde{\mathbf{D}_p^{(i)}}$ is the calculated degree matrix of $\mathcal{G}_p^{(i)}$ by row-wise summing $\widetilde{\mathbf{A}_p^{(i)}}$. Alternatively, one may also replace GCN with other off-the-shelf graph neural networks to aggregate messages from nodes' neighbors to calculate $\mathbf{H}_p^{(i)}$.

In patch-level contrastiveness, our discrimination pairs are the masked and original target node embeddings (e.g., $\mathbf{h}_p^{(i)}$ and $\mathbf{z}_p^{(i)}$ for a target node v_i), where the former one can be easily obtained via $\mathbf{h}_p^{(i)} = \mathbf{H}_p^{(i)}[1,:]$. To calculate the embeddings of original target nodes, e.g., $\mathbf{z}_p^{(i)}$, we only have to fed $\mathbf{x}^{(i)} = \mathbf{X}[i,:]$ into $GNN_\theta(\cdot)$ without the underlying graph structure since there is only a single node v_i. In such a way, $GNN_\theta(\cdot)$ degrades to an MLP that is parameterized with $\mathbf{\Theta}$. We illustrate the calculation of $\mathbf{z}_p^{(i)}$ as follows:

$$\mathbf{z}_p^{(i)} = GNN_\theta\left(\mathbf{x}^{(i)}\right) = \sigma\left(\mathbf{x}^{(i)}\mathbf{\Theta}\right), \tag{2}$$

where adopting $\mathbf{\Theta}$ ensures $\mathbf{z}_p^{(i)}$ and $\mathbf{h}_p^{(i)}$ are mapped into the same latent space to assist the following contrastive learning procedure.

Patch-Level Contrasting

To measure the agreement between $\mathbf{h}_p^{(i)}$ and $\mathbf{z}_p^{(i)}$, we adopt a bilinear mapping to compute the similarity between them (i.e., the positive score in ANEMONE), denoted as $\mathbf{s}_p^{(i)}$:

$$\mathbf{s}_p^{(i)} = Bilinear\left(\mathbf{h}_p^{(i)}, \mathbf{z}_p^{(i)}\right) = \sigma\left(\mathbf{h}_p^{(i)}\mathbf{W}_p{\mathbf{z}_p^{(i)}}^\top\right), \tag{3}$$

where $W_p \in \mathbb{R}^{D' \times D'}$ is a set of trainable weighting parameters, and $\sigma(\cdot)$ denotes the Sigmoid activation in this equation.

We further introduce a patch-level negative sampling mechanism to facilitate model training and prevent it from being dominated by positive pairs. Specifically, we first construct a subgraph centered at an irrelevant node v_j and obtain its patch-level representation $\mathbf{h}^{(j)}p$. We then compute its similarity with $\mathbf{z}^{(i)}p$ (serving as a negative score) using the same bilinear mapping function:

$$\tilde{\mathbf{s}}_p^{(i)} = Bilinear\left(\mathbf{h}_p^{(j)}, \mathbf{z}_p^{(i)}\right) = \sigma\left(\mathbf{h}_p^{(j)}\mathbf{W}_p{\mathbf{z}_p^{(i)}}^\top\right). \tag{4}$$

In practice, we train the model in a mini-batch manner as mentioned in Section 3.2.1. Thus, $\mathbf{h}_p^{(j)}$ can be easily acquired by using other masked target node embeddings in the same mini-batch with size B. Finally, the patch-level contrastive objective of ANEMONE (under the context of unsupervised GAD) can be formalized with the Jensen-Shannon divergence [45]:

$$\mathcal{L}_p = -\frac{1}{2n}\sum_{i=1}^{B}\left(log\left(\mathbf{s}_p^{(i)}\right) + log\left(1 - \tilde{\mathbf{s}}_p^{(i)}\right)\right). \tag{5}$$

3.2.3. Context-Level Contrastive Network

Different from the patch-level contrastiveness, the objective of context-level contrasting is to capture global agreement between the contextual embedding of a masked target node v_i in $\mathcal{G}_c^{(i)}$ and the embedding of itself by mapping its attributes to the latent space. The underlying intuition is to identify the global anomalies that are difficult to detect by directly comparing a node with its local context. For example, fraudsters may also camouflage themselves within large communities, making anomaly detection more challenging at the local level. To enable the model to detect these anomalies, we propose the context-level contrastiveness with a multi-scale (i.e., node versus graph) contrastive learning schema. In the middle part of Figure 1, we illustrate the conceptual design of our context-level contrastive network, which has three main components: graph encoder, readout module, and contrastive module.

Graph Encoder and Readout Module

The context-level graph encoder shares the identical neural architecture of the patch-level graph encoder, but it has a different set of trainable parameters $\mathbf{\Phi}$. Specifically, given a subgraph $\mathcal{G}_c^{(i)}$ centered at the target node v_i, we calculate the node embeddings of $\mathcal{G}_c^{(i)}$ in a similar way:

$$\mathbf{H}_c^{(i)} = GNN_\phi\left(\mathcal{G}_c^{(i)}\right) = \sigma\left(\widetilde{\mathbf{D}_c^{(i)}}^{-\frac{1}{2}}\widetilde{\mathbf{A}_c^{(i)}}\widetilde{\mathbf{D}_c^{(i)}}^{-\frac{1}{2}}\mathbf{X}_c^{(i)}\mathbf{\Phi}\right). \tag{6}$$

The main difference between the context-level and patch-level contrastiveness is that the former aims to contrast target node embeddings with subgraph embeddings (i.e., node versus subgraph), while the aforementioned patch-level contrasting learns the agreements between the masked and original target node embeddings (i.e., node versus node). To obtain the contextual embedding of target node v_i (i.e., $\mathbf{h}_c^{(i)}$), we aggregate all node embeddings in $\mathbf{H}_c^{(i)}$ with an average readout function:

$$\mathbf{h}_c^{(i)} = readout\left(\mathbf{H}_c^{(i)}\right) = \frac{1}{K}\sum_{j=1}^{K}\mathbf{H}_c^{(i)}[j,:], \tag{7}$$

where K denotes the number of nodes in a contextual subgraph.

Similarly, we can also obtain the embedding of v_i via a non-linear mapping:

$$\mathbf{z}_c^{(i)} = GNN_\phi\left(\mathbf{x}^{(i)}\right) = \sigma\left(\mathbf{x}^{(i)}\mathbf{\Phi}\right), \tag{8}$$

where $\mathbf{z}_c^{(i)}$ and $\mathbf{h}_c^{(i)}$ are projected to the same latent space with a shared set of parameters $\mathbf{\Phi}$.

Context-Level Contrasting

We measure the similarity between $\mathbf{h}_c^{(i)}$ and $\mathbf{z}_c^{(i)}$ (i.e., the positive score in ANEMONE) with a different parameterized bilinear function:

$$\mathbf{s}_c^{(i)} = Bilinear\left(\mathbf{h}_c^{(i)}, \mathbf{z}_c^{(i)}\right) = \sigma\left(\mathbf{h}_c^{(i)}\mathbf{W}_c{\mathbf{z}_c^{(i)}}^\top\right), \tag{9}$$

where $W_c \in \mathbb{R}^{D' \times D'}$ and $\sigma(\cdot)$ is the Sigmoid activation. Similarly, there is a context-level negative sampling mechanism to avoid model collapse, where negatives $\mathbf{h}_c^{(j)}$ are obtained from other irrelevant surrounding contexts $\mathcal{G}_c^{(j)}$ where $j \neq i$. Thus, the negative score can be obtained via:

$$\tilde{\mathbf{s}}_c^{(i)} = Bilinear\left(\mathbf{h}_c^{(j)}, \mathbf{z}_c^{(i)}\right) = \sigma\left(\mathbf{h}_c^{(j)}\mathbf{W}_c{\mathbf{z}_c^{(i)}}^\top\right). \tag{10}$$

Finally, the context-level contrastiveness is ensured by optimizing the following objective:

$$\mathcal{L}_c = -\frac{1}{2n}\sum_{i=1}^{B}\left(log\left(\mathbf{s}_c^{(i)}\right) + log\left(1 - \tilde{\mathbf{s}}_c^{(i)}\right)\right). \tag{11}$$

3.2.4. Few-Shot Multi-Scale Contrastive Network

The patch-level and context-level contrastive objectives are designed to detect attributive and structural graph anomalies in an unsupervised manner, as implemented in ANEMONE (see Algorithm 1). However, how to effectively utilize few-shot supervision remains challenging. Therefore, we propose an extension of ANEMONE, termed ANEMONE-FS, to perform few-shot GAD without substantially modifying the overall framework (i.e., Figure 2) and training objective (i.e., Equations (5), (11) and (17)). This design further boosts the performance of ANEMONE significantly with only a few labeled anomalies, which can be easily acquired in many real-world applications. Specifically, given a mini-batch of target nodes $\mathcal{V}_B = \{\mathcal{V}_B^L, \mathcal{V}_B^U\}$ consisting of labeled anomalies and unlabeled nodes, we introduce an additional contrastive route into both patch-level and context-level contrastiveness, as the bottom dashed arrows, i.e., the so-called negative pairs, in the red and green dashed boxes shown in the middle part of Figure 2.

Few-Shot Patch-Level Contrasting

For nodes in $\mathcal{V}_B^U$, we follow Equations (3) and (4) to compute positive and negative scores as in ANEMONE. However, for a target node v_k in $\mathcal{V}_B^L$, we minimize the mutual information between its masked node embedding $\mathbf{h}_p^{(k)}$ and original node embedding $\mathbf{z}_p^{(k)}$, which equivalents to enrich the patch-level negative set with an additional negative pair:

$$\tilde{\mathbf{s}}_p^{(k)} = Bilinear\left(\mathbf{h}_p^{(k)}, \mathbf{z}_p^{(k)}\right) = \sigma\left(\mathbf{h}_p^{(k)}\mathbf{W}_p{\mathbf{z}_p^{(k)}}^\top\right). \tag{12}$$

The underlying intuition is twofold. First, for most unlabeled nodes, we assume that they are normal rather than anomalous. Therefore, the mutual information between the masked and original target node embeddings should be maximized, enabling the model to distinguish normal nodes from a few anomalies in an attributed graph. Second, for a labeled target node (i.e., an anomaly), this mutual information should instead be minimized, encouraging the model to learn how anomalies differ from their surrounding context. As a result, there are two types of patch-level negative pairs in ANEMONE-FS: (1) $\mathbf{h}_p^{(j)}$ and $\mathbf{z}_p^{(i)}$ where $v_i \in \mathcal{V}_B^U$ and $i \neq j$; (2) $\mathbf{h}_p^{(k)}$ and $\mathbf{z}_p^{(k)}$ where $v_k \in \mathcal{V}_B^L$.

Few-Shot Context-Level Contrasting

Similarly, for a node v_k in $\mathcal{V}_B^L$, we minimize the mutual information between its contextual embedding $\mathbf{h}_c^{(k)}$ and the embedding of itself $\mathbf{h}_c^{(k)}$ by treating them as a negative pair:

Algorithm 1 The Proposed ANEMONE Algorithm

Input: Attributed graph $\mathcal{G}$ with a set of unlabeled nodes $\boldsymbol{\mathcal{V}}$; Maximum training epochs E; Batch size B; Number of evaluation rounds R.
Output: Well-trained GAD model $\mathcal{F}^*(\cdot)$.

Randomly initialize the trainable parameters $\boldsymbol{\Theta}$, $\boldsymbol{\Phi}$, $\mathbf{W}_p$, and $\mathbf{W}_c$;
/ Model training */*
for $e \in 1, 2, \cdots, E$ **do**
  $\boldsymbol{\mathcal{B}} \leftarrow$ Randomly split $\boldsymbol{\mathcal{V}}$ into batches with size B;
  for batch $\widetilde{\boldsymbol{\mathcal{B}}} = (v_1, \cdots, v_B) \in \boldsymbol{\mathcal{B}}$ **do**
    Sample two anonymized subgraphs for each node in $\widetilde{\boldsymbol{\mathcal{B}}}$, i.e., $\{\mathcal{G}_p^{(1)}, \cdots, \mathcal{G}_p^{(B)}\}$ and $\{\mathcal{G}_c^{(1)}, \cdots, \mathcal{G}_c^{(B)}\}$;
    Calculate the masked and original node embeddings via Equations (1) and (2);
    Calculate the masked node and its contextual embeddings via Equations (6)–(8);
    Calculate the patch-level positive and negative scores for each node in $\widetilde{\boldsymbol{\mathcal{B}}}$ via Equations (3) and (4);
    Calculate the context-level positive and negative scores for each node in $\widetilde{\boldsymbol{\mathcal{B}}}$ via Equations (9) and (10);
    Calculate the loss $\mathcal{L}$ via Equations (5), (11) and (17);
    Back propagate to update trainable parameters $\boldsymbol{\Theta}$, $\boldsymbol{\Phi}$, $\mathbf{W}_p$, and $\mathbf{W}_c$;
  end for
end for
/ Model inference */*
for $v_i \in \mathcal{V}$ **do**
  for evaluation round $r \in 1, 2, \cdots, R$ **do**
    Calculate $b_p^{(i)}$ and $b_c^{(i)}$ via Equation (14);
  end for
  Calculate the final patch-level and context-level anomaly scores $y_p^{(i)}$ and $y_c^{(i)}$ over R evaluation rounds via Equation (15);
  Calculate the final anomaly score $y^{(i)}$ of v_i via Equation (16);
end for

$$\tilde{\mathbf{s}}_c^{(k)} = Bilinear\left(\mathbf{h}_c^{(k)}, \mathbf{z}_c^{(k)}\right) = \sigma\left(\mathbf{h}_c^{(k)} \mathbf{W}_c {\mathbf{z}_c^{(k)}}^\top\right). \tag{13}$$

The underlying intuition is similar to that of the few-shot patch-level contrastive learning module. Accordingly, we also introduce two types of negative pairs in this module to provide richer supervision signals from different perspectives.

Overall, this plug-and-play extension enhances the self-supervised contrastive anomaly detection framework in ANEMONE by introducing additional negative pairs derived from a small set of labeled anomalies, enabling improved performance in more realistic application scenarios.

3.2.5. Statistical Graph Anomaly Scorer

So far, we have introduced two contrastive mechanisms in ANEMONE and ANEMONE-FS for different GAD tasks. After training, we propose a universal statistical graph anomaly scorer for both ANEMONE and ANEMONE-FS to calculate the anomaly score of each node in $\mathcal{G}$ during inference. Specifically, we first generate R subgraphs centered at a target node v_i for both contrastive networks. Then, we calculate patch-level and context-level contrastive scores accordingly, i.e., $[\mathbf{s}_{p,1}^{(i)}, \cdots, \mathbf{s}_{p,R}^{(i)}, \mathbf{s}_{c,1}^{(i)}, \cdots, \mathbf{s}_{c,R}^{(i)}, \tilde{\mathbf{s}}_{p,1}^{(i)}, \cdots, \tilde{\mathbf{s}}_{p,R}^{(i)}, \tilde{\mathbf{s}}_{c,1}^{(i)}, \cdots, \tilde{\mathbf{s}}_{c,R}^{(i)}]$. After this, we define the base patch-level and context-level anomaly scores of v_i as follows:

$$b_{view,j}^{(i)} = \tilde{\mathbf{s}}_{view,j}^{(i)} - \mathbf{s}_{view,j}^{(i)}, \tag{14}$$

where $j \in \{1, \cdots, R\}$, and "view" corresponds to p or c to denote the patch-level or context-level base score, respectively. If v_i is a normal node, then $\mathbf{s}_{view,j}^{(i)}$ and $\tilde{\mathbf{s}}_{view,j}^{(i)}$ are expected to be close to 1 and 0, leading $b_{view,j}^{(i)}$ close to -1. Otherwise, if v_i is an anomaly, then $\mathbf{s}_{view,j}^{(i)}$ and $\tilde{\mathbf{s}}_{view,j}^{(i)}$ are close to 0.5 due to the mismatch between v_i and its surrounding contexts, resulting in $b_{view,j}^{(i)} \to 0$. Therefore, we have $b_{view,j}^{(i)}$ in the range of $[-1, 0]$.

Although we can directly use the base anomaly score $b_{view,j}^{(i)}$ to indicate whether v_i is an anomaly, we design a more sophisticated statistical abnormality scorer to calculate the final patch-level and context-level anomaly scores $y_{view}^{(i)}$ of v_i based on $b_{view,j}^{(i)}$:

$$\begin{aligned} \bar{b}_{view}^{(i)} &= \sum_{j=1}^{R} b_{view,j}^{(i)}/R, \\ y_{view}^{(i)} &= \bar{b}_{view}^{(i)} + \sqrt{\sum_{j=1}^{R} \left(b_{view,j}^{(i)} - \bar{b}_{view}^{(i)}\right)^2/R}. \end{aligned} \tag{15}$$

The underlying intuitions behind the above equation are: (1) An abnormal node usually has a larger base anomaly score; (2) The base scores of an abnormal node are typically unstable (i.e., with a larger standard deviation) under R evaluation rounds. Finally, we anneal $y_p^{(i)}$ and $y_c^{(i)}$ to obtain the final anomaly score of v_i with a tunable hyper-parameter $\alpha \in [0, 1]$ to balance the importance of two anomaly scores at different scales:

$$y^{(i)} = \alpha y_c^{(i)} + (1 - \alpha) y_p^{(i)}. \tag{16}$$

3.2.6. Model Training and Algorithms

Model Training

By combining the patch-level and context-level contrastive losses defined in Equations (5) and (11), we have the overall training objective by minimizing the following loss:

$$\mathcal{L} = \alpha \mathcal{L}_c + (1 - \alpha)\mathcal{L}_p, \tag{17}$$

where α is same as in Equation (16) to balance the importance of two contrastive modules.

The overall procedures of ANEMONE and ANEMONE-FS are in Algorithms 1 and 2. Specifically, in ANEMONE, we first sample a batch of nodes from the input attributed graph (line 5). Then, we calculate the positive and negative contrastive scores for each node (lines 6–10) to obtain the multi-scale contrastive losses, which are adopted to calculate the overall training loss (line 11) to update all trainable parameters (line 12). For ANEMONE-FS, the differences are twofold. Firstly, it takes an attributed graph with a few available labeled anomalies as the input. Secondly, for labeled anomalies and unlabeled nodes in a batch, it has different contrastive routes in patch-level and context-level contrastive networks (lines 9–12). During the model inference, ANEMONE and ANEMONE-FS share the same anomaly scoring mechanism, where the statistical anomaly score for each node in $\boldsymbol{\mathcal{V}}$ or $\boldsymbol{\mathcal{V}}^U$ is calculated (lines 16–22 in Algorithm 1 and lines 18–24 in Algorithm 2).

Complexity Analysis

We analyze the time complexity of ANEMONE and ANEMONE-FS algorithms in this subsection. For the shared anonymized subgraph sampling module, the time complexity of using the RWR algorithm to sample a subgraph centered at v_i is $\mathcal{O}(Kd)$, where K and d are the number of nodes in a subgraph and the average node degree in $\mathcal{G}$. For the two proposed contrastive modules, the computational complexity is dominated by the underlying graph encoders, resulting in a complexity of $\mathcal{O}(K^2)$. Thus, given N nodes in $\boldsymbol{\mathcal{V}}$, the time complexity of model training is $\mathcal{O}\big(NK(d+K)\big)$ in both ANEMONE and ANEMONE-FS. During inference, the time complexity of ANEMONE is $\mathcal{O}\big(RNK(d+K)\big)$, where R denotes the total evaluation rounds. For ANEMONE-FS, its inference time complexity is $\mathcal{O}\big(RN^UK(d+K)\big)$, where $N^U = |\boldsymbol{\mathcal{V}}^{\boldsymbol{\mathcal{U}}}|$ denotes the number of unlabeled nodes in $\mathcal{G}$. Since the computational complexity is mainly dominated by subgraph sampling and the graph encoder, ANEMONE achieves comparable efficiency to CoLA [12]. Moreover, because ANEMONE does not require adjacency matrix reconstruction, it is more efficient than DOMINANT [10], whose time complexity is $\mathcal{O}(Nd + N^2)$.

4. Results

In this section, we conduct a series of experiments to evaluate the anomaly detection performance of the proposed ANEMONE and ANEMONE-FS under both unsupervised and few-shot learning scenarios. Specifically, we address the following research questions through empirical analysis:

- RQ1: How do the proposed ANEMONE and ANEMONE-FS perform in comparison to state-of-the-art GAD methods?
- RQ2: How does the performance of ANEMONE-FS change by providing different numbers of labeled anomalies?
- RQ3: How do the contrastiveness in patch-level and context-level influence the performance of ANEMONE and ANEMONE-FS?
- RQ4: How do the key hyper-parameters impact the performance of ANEMONE?

Algorithm 2 The Proposed ANEMONE-FS Algorithm

Input: Attributed graph $\mathcal{G}$ with a set of labeled and unlabeled nodes $\mathcal{V} = \{\mathcal{V}^L, \mathcal{V}^U\}$ where $|\mathcal{V}^L| \ll |\mathcal{V}^U|$; Maximum training epochs E; Batch size B; Number of evaluation rounds R.
Output: Well-trained GAD model $\mathcal{F}^*(\cdot)$.

Randomly initialize the trainable parameters $\mathbf{\Theta}$, $\mathbf{\Phi}$, $\mathbf{W}_p$, and $\mathbf{W}_c$;
/* *Model training* */
for $e \in 1, 2, \cdots, E$ **do**
  $\mathcal{B} \leftarrow$ Randomly split $\mathcal{V}$ into batches with size B;
  for batch $\widetilde{\mathcal{B}} = \{\mathcal{V}_B^L, \mathcal{V}_B^U\} = (v_1, \cdots, v_B) \in \mathcal{B}$ **do**
    Sample two anonymized subgraphs for each node in $\widetilde{\mathcal{B}}$, i.e., $\{\mathcal{G}_p^{(1)}, \cdots, \mathcal{G}_p^{(B)}\}$ and $\{\mathcal{G}_c^{(1)}, \cdots, \mathcal{G}_c^{(B)}\}$;
    Calculate the masked and original node embeddings via Equations (1) and (2);
    Calculate the masked node and its contextual embeddings via Equations (6)–(8);
    Calculate the patch-level positive and negative scores for each node in $\mathcal{V}_B^U$ via Equations (3) and (4);
    Calculate the patch-level extra negative scores for each node in $\mathcal{V}_B^L$ via Equation (12);
    Calculate the context-level positive and negative scores for each node in $\mathcal{V}_B^U$ via Equations (9) and (10);
    Calculate the context-level extra negative scores for each node in $\mathcal{V}_B^L$ via Equation (13);
    Calculate the loss $\mathcal{L}$ via Equations (5), (11) and (17);
    Back propagate to update trainable parameters $\mathbf{\Theta}$, $\mathbf{\Phi}$, $\mathbf{W}_p$, and $\mathbf{W}_c$;
  end for
end for
/* *Model inference* */
for $v_i \in \mathcal{V}^U$ **do**
  for evaluation round $r \in 1, 2, \cdots, R$ **do**
    Calculate $b_p^{(i)}$ and $b_c^{(i)}$ via Equation (14);
  end for
  Calculate the final patch-level and context-level anomaly scores $y_p^{(i)}$ and $y_c^{(i)}$ over R evaluation rounds via Equation (15);
  Calculate the final anomaly score $y^{(i)}$ of v_i via Equation (16);
end for

4.1. Datasets

We conduct experiments on six commonly used datasets for GAD, including four citation network datasets [55,56] (i.e., Cora, CiteSeer, PubMed, and ACM) and two social network datasets [57] (i.e., BlogCatalog and Flickr). Dataset statistics are summarized in Table 2.

Since ground-truth anomalies are inaccessible for these datasets, we follow previous works [10,12] to inject two types of synthetic anomalies (i.e., structural anomalies and contextual anomalies) into the original graphs. For structural anomaly injection, we use the injection strategy proposed by [58]: several groups of nodes are randomly selected from the graph, and then we make the nodes within one group fully linked to each other. In this way, such nodes can be regarded as structural anomalies. To generate contextual anomalies, following [59], a target node along with 50 auxiliary nodes is randomly sampled from the graph. Then, we replace the features of the target node with the features of the farthest auxiliary node (i.e., the auxiliary node with the largest features' Euclidean distance to the target node). By this, we denote the target node as a contextual anomaly. We inject two types of anomalies with the same quantity, and the total number is provided in the last column of Table 2. The injected anomalies provide a reasonable simulation of real-world fraudulent behaviors. Specifically, clique-based structural anomalies mimic collusive behaviors among spammers, while attribute anomalies are constructed to represent abnormal users with unusual characteristics.

4.2. Baselines

We compare our proposed ANEMONE and ANEMONE-FS with three types of baseline methods, including (1) shallow learning-based unsupervised methods (i.e., AMEN [7], Radar [8]), (2) deep learning-based unsupervised methods (i.e., ANOMALOUS [9], DOMINANT [10], and CoLA [12]), and (3) semi-supervised methods (i.e., DeepSAD [22], SemiGNN [26], and GDN [27]). Details of these methods are introduced as follows:

- AMEN [7] is an unsupervised GAD method which detects anomalies by analyzing the attribute correlation of the ego-network of nodes.
- Radar [8] identifies anomalies in graphs by residual and attribute-structure coherence analysis.

Table 2. The statistics of the datasets. The upper two datasets are social networks, and the remainder are citation networks.

Dataset	Nodes	Edges	Features	Anomalies
Cora [55]	2708	5429	1433	150
CiteSeer [55]	3327	4732	3703	150
PubMed [55]	19,717	44,338	500	600
ACM [56]	16,484	71,980	8337	600
BlogCatalog [57]	5196	171,743	8189	300
Flickr [57]	7575	239,738	12,407	450

- ANOMALOUS [9] is an unsupervised method for attributed graphs, which performs anomaly detection via CUR decomposition and residual analysis.
- DGI [45] is an unsupervised contrastive learning method for representation learning. In DGI, we use the score computation module in [12] to estimate nodes' abnormality.
- DOMINANT [10] is a deep graph autoencoder-based unsupervised method that detects anomalies by evaluating the reconstruction errors of each node.
- CoLA [12] is a contrastive learning-based anomaly detection method that captures anomalies with a GNN-based contrastive framework.
- DeepSAD [22] is a deep learning-based anomaly detection method for non-structured data. We take node attributes as the input of DeepSAD.
- SemiGNN [26] is a semi-supervised fraud detection method that uses an attention mechanism to model the correlation between different neighbors/views.
- GDN [27] is a GNN-based model that detects anomalies in few-shot learning scenarios. It leverages a deviation loss to train the detection model in an end-to-end manner.

4.3. Experimental Setting

4.3.1. Evaluation Metric

We employ a widely used metric, AUC-ROC [10, 12], to evaluate the performance of different anomaly detection methods. The ROC curve indicates the plot of true positive rate against false positive rate, and the AUC value is the area under the ROC curve. The value of AUC is within the range $[0, 1]$, and a larger value represents a stronger detection performance. To reduce the bias caused by randomness and compare fairly [12], for all datasets, we conduct a 5-run experiment and report the average performance.

4.3.2. Dataset Partition

In an unsupervised learning scenario, we use graph data $\mathcal{G} = (\mathbf{X}, \mathbf{A})$ to train the models, and evaluate the anomaly detection performance on the full node set $\boldsymbol{\mathcal{V}}$ (including all normal and abnormal nodes). Under few-shot settings, we train the models with $\mathcal{G}$ and k anomalous labels (where k is the size of the labeled node set $|\boldsymbol{\mathcal{V}}^L|$), and use the rest set of nodes $\boldsymbol{\mathcal{V}}^U$ to measure the models' performance. Specifically, in each run, we randomly sample k anomalous nodes for supervision and evaluate the model on the remaining anomalous nodes. To ensure a fair comparison, all semi-supervised baselines are evaluated using the same set of supervised nodes.

4.3.3. Parameter Settings

In our implementation, the size K of the subgraph and the dimension of embeddings are fixed to 4 and 64, respectively. The trade-off parameter α is searched in $\{0.2, 0.4, 0.6, 0.8, 1\}$. The number of testing rounds of the anomaly estimator is set to 256. We train the model with the Adam optimizer with a learning rate 0.001. To be consistent with CoLA [12], for Cora, Citeseer, and Pubmed datasets, we train the model for 100 epochs; for ACM, BlogCatalog, and Flickr datasets, the number of epochs is 2000, 1000, and 500, respectively.

4.4. Performance Comparison (RQ1)

We evaluate `ANEMONE` in an unsupervised learning scenario where labeled anomaly is unavailable, and evaluate `ANEMONE-FS` in a few-shot learning scenario ($k = 10$) where 10 annotated anomalies are known during model training.

4.4.1. Performance in Unsupervised Learning Scenario

In an unsupervised scenario, we compared ANEMONE with 6 unsupervised baselines. The ROC curves on 4 representative datasets are illustrated in Figure 3, and the comparison of AUC value on all 6 datasets is provided in Table 3. From these results, we have the following observations.

- ANEMONE consistently outperforms all baselines on six benchmark datasets. The performance gain is due to (1) the two-level contrastiveness successfully capturing anomalous patterns in different scales and (2) the well-designed anomaly estimator effectively measuring the abnormality of each node.
- The deep learning-based methods significantly outperform the shallow learning-based methods, which illustrates the capability of GNNs in modeling data with complex network structures and high-dimensional features.
- The ROC curves by ANEMONE are very close to the points in the upper left corner, indicating our method can precisely discriminate abnormal samples from a large number of normal samples.

4.4.2. Performance in Few-Shot Learning Scenario

In a few-shot learning scenario, we consider both unsupervised and semi-supervised baseline methods for comparison with our methods. The results are demonstrated in Table 4. As we can observe, ANEMONE and ANEMONE-FS achieve consistently better performance than all baselines, which validates that our methods can handle a few-shot learning setting as well. Also, we find that ANEMONE-FS has better performance than ANEMONE, meaning that our proposed solution for few-shot learning can further leverage the knowledge from a few labeled anomalies. In comparison, the semi-supervised learning methods (i.e., DeepSAD, SemiGNN, and GDN) do not show a competitive performance, indicating their limited capability in exploiting the label information.

4.5. Few-Shot Performance Analysis (RQ2)

In order to verify the effectiveness of ANEMONE-FS in different few-shot anomaly detection settings, we change the number k of anomalous samples for model training to form k-shot learning settings for evaluation. We perform experiments on four datasets (i.e., Cora, CiteSeer, ACM, and BlogCatalog) and select k from $\{1, 3, 5, 10, 15, 20\}$. The experimental results are demonstrated in Table 5, where the performance in an unsupervised setting (denoted as "Unsup.") is also reported as a baseline.

The results show that our ANEMONE-FS can achieve good performance even when only one anomaly node is provided (i.e., 1-shot setting). A representative example is the results of ACM dataset, where a 1.16% performance gain is brought by one labeled anomaly. Such an observation indicates that ANEMONE-FS can effectively leverage the knowledge from scarce labeled samples to better model the anomalous patterns. Another finding is that the anomaly detection performance generally increases following the growth of k, especially when $k \leq 15$. This finding demonstrates that ANEMONE-FS can further optimize the anomaly detection model when more labeled anomalies are given.

4.6. Ablation Study (RQ3)

In this experiment, we investigate the contribution of patch- and context-level contrastiveness to the anomaly detection performance of ANEMONE and ANEMONE-FS. In concrete, we adjust the value of the trade-off parameter α and the results are illustrated in Figure 4. Note that $\alpha = 0$ and $\alpha = 1$ mean that the model only considers patch- and context-level contrastive learning, respectively.

As we can observe in Figure 4, ANEMONE and ANEMONE-FS can achieve the highest AUC values when α is between 0.2 and 0.8, and the best selections of α for each dataset are quite different. Accordingly, we summarize that jointly considering the contrastiveness in both levels always brings the best detection performance. We also observe that, on citation networks such as Cora, CiteSeer, and PubMed, the context-level contrastive network outperforms the patch-level one, whereas on social networks, patch-level contrastiveness yields better results. This reflects that citation networks benefit more from global semantic information, while social networks rely more on local relational patterns. These findings demonstrate that the two types of contrastiveness capture complementary anomaly signals, enabling our method to generalize across diverse application scenarios with different relational properties. Moreover, this observation provides a useful heuristic for selecting an appropriate α: a larger α is preferred for graphs where global semantics are more important, whereas a smaller α is more suitable for graphs with limited global consistency but rich neighborhood information.

Table 3. The comparison of anomaly detection performance (i.e., AUC) in an unsupervised learning scenario. The best performance is highlighted in bold.

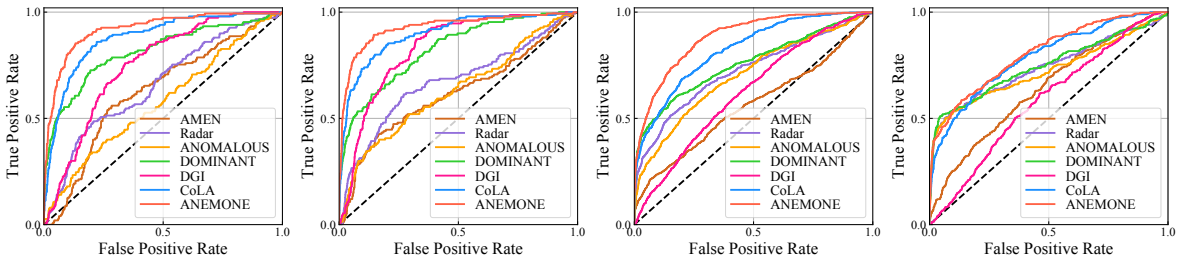
Method	Cora	CiteSeer	PubMed	ACM	BlogCatalog	Flickr
AMEN [7]	0.6266	0.6154	0.7713	0.5626	0.6392	0.6573
Radar [8]	0.6587	0.6709	0.6233	0.7247	0.7401	0.7399
ANOMALOUS [9]	0.5770	0.6307	0.7316	0.7038	0.7237	0.7434
DGI [45]	0.7511	0.8293	0.6962	0.6240	0.5827	0.6237
DOMINANT [10]	0.8155	0.8251	0.8081	0.7601	0.7468	0.7442
CoLA [12]	0.8779	0.8968	0.9512	0.8237	0.7854	0.7513
ANEMONE	**0.9057**	**0.9189**	**0.9548**	**0.8709**	**0.8067**	**0.7637**

Table 4. The comparison of anomaly detection performance (i.e., AUC) in a few-shot learning scenario. The best performance is highlighted in bold.

Method	Cora	CiteSeer	PubMed	ACM	BlogCatalog	Flickr
AMEN [7]	0.6257	0.6103	0.7725	0.5632	0.6358	0.6615
Radar [8]	0.6589	0.6634	0.6226	0.7253	0.7461	0.7357
ANOMALOUS [9]	0.5698	0.6323	0.7283	0.6923	0.7293	0.7504
DGI [45]	0.7398	0.8347	0.7041	0.6389	0.5936	0.6295
DOMINANT [10]	0.8202	0.8213	0.8126	0.7558	0.7391	0.7526
CoLA [12]	0.8810	0.8878	0.9517	0.8272	0.7816	0.7581
DeepSAD [22]	0.4909	0.5269	0.5606	0.4545	0.6277	0.5799
SemiGNN [26]	0.6657	0.7297	OOM	OOM	0.5289	0.5426
GDN [27]	0.7577	0.7889	0.7166	0.6915	0.5424	0.5240
ANEMONE	0.8997	0.9191	0.9536	0.8742	0.8025	0.7671
ANEMONE-FS	**0.9155**	**0.9318**	**0.9561**	**0.8955**	**0.8124**	**0.7781**

Table 5. Few-shot performance analysis of ANEMONE-FS.

Setting	Cora	CiteSeer	ACM	BlogCatalog
Unsup.	0.8997	0.9191	0.8742	0.8025
1-shot	0.9058	0.9184	0.8858	0.8076
3-shot	0.9070	0.9199	0.8867	0.8125
5-shot	0.9096	0.9252	0.8906	0.8123
10-shot	0.9155	0.9318	0.8955	0.8124
15-shot	0.9226	0.9363	0.8953	0.8214
20-shot	0.9214	0.9256	0.8965	0.8228

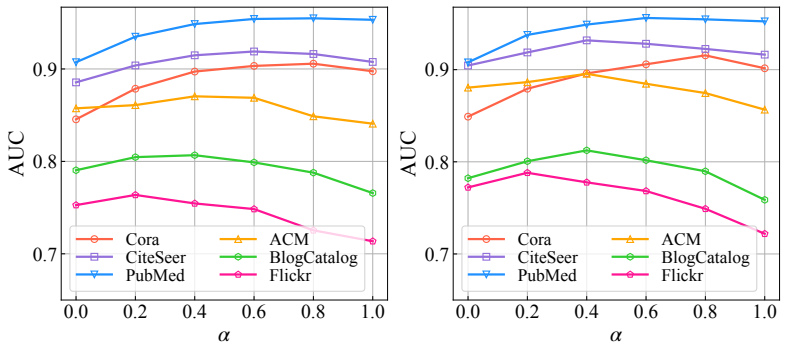


(a) ROC curve of Cora. (b) ROC curve of CiteSeer. (c) ROC curve of ACM. (d) ROC curve of BlogCatalog.

Figure 3. The comparison of ROC curves on four datasets in an unsupervised learning scenario.

4.7. Parameter Sensitivity (RQ4)

To study how our method is impacted by the key hyper-parameters, we conduct experiments for ANEMONE with different selections of evaluation rounds R, subgraph size K, and hidden dimension D'.



(a) AUC w.r.t. α in ANEMONE. (b) AUC w.r.t. α in ANEMONE−FS.

Figure 4. Anomaly detection performance with different selection of trade-off parameter α in unsupervised and few-shot learning scenarios.

4.7.1. Evaluation Rounds

To explore the sensitivity of ANEMONE to evaluation rounds R, we tune the R from 1 to 512 on six datasets, and the results are demonstrated in Figure 5a. As we can find in the figure, the detection performance is relatively poor when $R < 4$, indicating that too few evaluation rounds are insufficient to represent the abnormality of each node. When R is between 4 and 256, we can witness a significant growing trade of AUC following the increase of R, which demonstrates that adding evaluation rounds within certain ranges can significantly enhance the performance of ANEMONE. An over-large R ($R = 512$) does not boost the performance but brings higher computational cost. Hence, we fix $R = 256$ in our experiments to balance performance and efficiency.

4.7.2. Subgraph Size

In order to investigate the impact of subgraph size, we search for the node number of contextual subgraph K in the range of $\{2, 3, \cdots, 10\}$. We plot the results in Figure 5b. We find that ANEMONE is not sensitive to the choice of K on datasets except Flickr, which verifies the robustness of our method. For citation networks (i.e., Cora, CiteSeer, PubMed, and ACM), a suitable subgraph size between 3 and 5 results in the best performance. Differently, BlogCatalog requires a larger subgraph to consider more contextual information, while Flickr needs a smaller augmented subgraph for contrastive learning.

4.7.3. Hidden Dimension

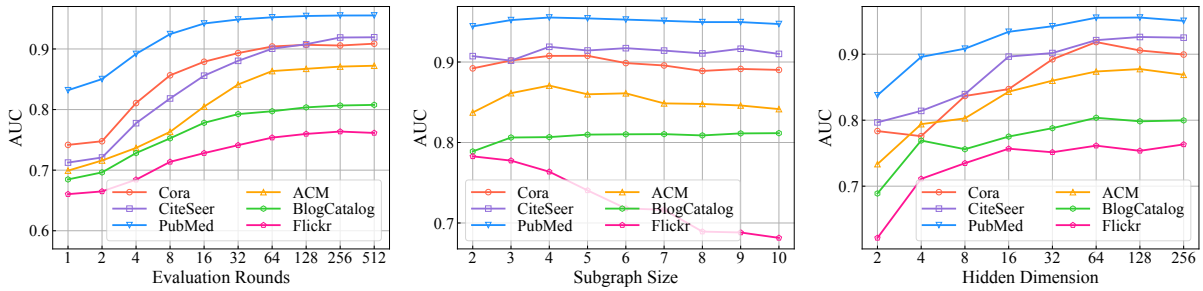
In this experiment, we study the selection of hidden dimension D' in ANEMONE. We alter the value of D' from 2 to 256 and the effect of D' on AUC is illustrated in Figure 5c. As shown in the figure, when D' is within $[2, 64]$, there is a significant boost in anomaly detection performance with the growth of D'. This observation indicates that node embeddings with a larger length can help ANEMONE capture more complex information. We also find that the performance gain becomes light when D' is further enlarged. Consequently, we finally set $D' = 64$ in our main experiments.

5. Discussion

The experimental results demonstrate that ANEMONE consistently achieves superior performance over a wide range of unsupervised baselines across all benchmark datasets, indicating strong generalization in graph anomaly detection. The corresponding ROC curves further confirm that the model effectively separates anomalous and normal nodes even under extreme class imbalance. These improvements are primarily attributed to the proposed multi-scale contrastive learning framework, which jointly captures local structural perturbations at the patch level and global consistency at the context level. Compared with prior approaches that rely on a single view of representation learning [10, 12], this dual-level design enables more comprehensive modeling of heterogeneous anomaly patterns.

In the few-shot setting, ANEMONE−FS consistently outperforms both unsupervised and semi-supervised baselines, including methods specifically designed for label utilization [22, 26, 27]. This suggests that effective few-shot anomaly detection requires not only incorporating supervision, but also aligning it with a representation space that is already sensitive to anomalous deviations. The consistent gains of ANEMONE−FS over ANEMONE further demonstrate that a small number of labeled anomalies can meaningfully refine the learned decision boundary without overfitting. The k-shot analysis further shows that performance improves rapidly in the low-shot regime and gradually saturates, highlighting strong label efficiency.

The ablation study validates the complementarity of patch-level and context-level contrastive objectives.



(a) AUC w.r.t. number of evaluation rounds. (b) AUC w.r.t. size of subgraph. (c) AUC w.r.t. hidden dimension.

Figure 5. Parameter sensitivities of ANEMONE w.r.t. three hyper-parameters on six benchmark datasets.

Optimal performance is achieved when both are jointly optimized, while performance degradation occurs when either component is removed. The dataset-dependent dominance of each level further suggests that anomalies manifest at different structural granularities across graphs, underscoring the necessity of multi-scale modeling. Parameter sensitivity analysis further demonstrates the robustness of the model to key hyperparameters. Specifically, the performance remains stable across a wide range of subgraph sizes and embedding dimensions, while benefiting from a sufficient number of evaluation rounds without requiring excessively large values.

Despite these advantages, ANEMONE and ANEMONE-FS are designed under a transductive setting, where the entire graph is available during training. In real-world applications, however, graphs are often dynamic, with unseen nodes continuously arriving and inducing distribution shifts at test time, which remains an open challenge. In addition, while attributed graphs provide a compact representation of node features, real-world scenarios such as social networks often contain rich raw information (e.g., textual content) that may not be fully captured by pre-defined attributes, making text-attributed graph anomaly detection an important and promising direction for future work.

Overall, these results position ANEMONE as a strong multi-scale contrastive learning framework for graph anomaly detection, while ANEMONE-FS demonstrates that even minimal supervision can be effectively leveraged when integrated into a well-structured representation space. These findings also suggest that future work could benefit from incorporating foundation models or in-context learning to further enhance graph anomaly detection in complex graph-structured data.

6. Conclusions

In this paper, we investigate the problem of graph anomaly detection. By jointly capturing anomalous patterns from multiple scales with both patch level and context level contrastive learning, we propose a novel algorithm, ANEMONE, to learn the representation of nodes in a graph. With a statistical anomaly estimator to capture the agreement from multiple perspectives, we predict an anomaly score for each node so that anomaly detection can be conducted subsequently. As a handful of ground-truth anomalies may be available in real applications, we further extend our method as ANEMONE-FS, a powerful method to utilize labeled anomalies to handle the settings of few-shot graph anomaly detection. Experiments on six benchmark datasets validate the performance of the proposed ANEMONE and ANEMONE-FS. In the future, leveraging the in-context learning capacity, which only requires a few samples, we will look into foundation models or large language models [38,60], for effective graph anomaly detection.

Author Contributions

Y.Z.: methodology, software, investigation; J.P.: software, investigation, writing—review and editing; Y.T.: writing—review and editing, formal analysis; M.J.: data curation, conceptualization, project administration; Y.L.: writing—original draft preparation, conceptualization, visualization, project administration; L.C.: supervision, funding acquisition, resources; K.P.: supervision, funding acquisition; Y.-P.P.C.: supervision, funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding

The work of Y. Liu was partially supported by the ARC under Grant No. DE260101172.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The datasets used in this study can be found at https://github.com/TrustAGI-Lab/CoLA

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

During the preparation of this work, the authors used LLM models to help to polish the text. After using it, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

1. Wu, Z.; Pan, S.; Chen, F.; et al. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 4–24.
2. Yu, J.; Wang, H.; Wang, X.; et al. Group-based fraud detection network on e-commerce platforms. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Long Beach, CA, USA, 6–10 August 2023.
3. Wang, C.; Zhu, H. Wrongdoing monitor: A graph-based behavioral anomaly detection in cyber security. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 2703–2718.
4. Pan, J.; Liu, Y.; Zheng, X.; et al. A label-free heterophily-guided approach for unsupervised graph fraud detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Philadelphia, PA, USA, 25 February–4 March 2025; Vol. 39, pp. 12443–12451.
5. Pourhabibi, T.; Ong, K.L.; Kam, B.H.; et al. Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decis. Support Syst.* **2020**, *133*, 113303.
6. Latah, M. Detection of malicious social bots: A survey and a refined taxonomy. *Expert Syst. Appl.* **2020**, *151*, 113383.
7. Perozzi, B.; Akoglu, L. Scalable anomaly ranking of attributed neighborhoods. In Proceedings of the SIAM International Conference on Data Mining, Miami, FL, USA, 5–7 May 2016; pp. 207–215.
8. Li, J.; Dani, H.; Hu, X.; et al. Radar: Residual Analysis for Anomaly Detection in Attributed Networks. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, VIC, Australia, 19–25 August 2017; pp. 2152–2158.
9. Peng, Z.; Luo, M.; Li, J.; et al. ANOMALOUS: A Joint Modeling Approach for Anomaly Detection on Attributed Networks. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 3513–3519.
10. Ding, K.; Li, J.; Bhanushali, R.; et al. Deep anomaly detection on attributed networks. In Proceedings of the SIAM International Conference on Data Mining, Calgary, AB, Canada, 2–4 May 2019; pp. 594–602.
11. Li, Y.; Huang, X.; Li, J.; et al. Specae: Spectral autoencoder for anomaly detection in attributed networks. In Proceedings of the ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 2233–2236.
12. Liu, Y.; Li, Z.; Pan, S.; et al. Anomaly Detection on Attributed Networks via Contrastive Self-Supervised Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 2378–2392.
13. Pang, G.; Shen, C.; van den Hengel, A. Deep anomaly detection with deviation networks. In Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 353–362.
14. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
15. Veličković, P.; Cucurull, G.; Casanova, A.; et al. Graph Attention Networks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
16. Wu, Z.; Pan, S.; Long, G.; et al. Beyond low-pass filtering: Graph convolutional networks with automatic filtering. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 6687–6697.
17. Bruna, J.; Zaremba, W.; Szlam, A.; et al. Spectral networks and locally connected networks on graphs. In Proceedings of the International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014.
18. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In Proceedings of the Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; Vol. 29, pp. 3844–3852.
19. Hamilton, W.L.; Ying, R.; Leskovec, J. Inductive representation learning on large graphs. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 1025–1035.

20. Xu, K.; Hu, W.; Leskovec, J.; et al. How Powerful are Graph Neural Networks? In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
21. Ma, X.; Wu, J.; Xue, S.; et al. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 12012–12038.
22. Ruff, L.; Vandermeulen, R.A.; Görnitz, N.; et al. Deep Semi-Supervised Anomaly Detection. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
23. He, J.; Xu, Q.; Jiang, Y.; et al. Ada-gad: Anomaly-denoised autoencoders for graph anomaly detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–27 February 2024; Vol. 38, pp. 8481–8489.
24. Pan, J.; Liu, Y.; Zheng, Y.; et al. Prem: A simple yet effective approach for node-level graph anomaly detection. In Proceedings of the IEEE International Conference on Data Mining, Shanghai, China, 1–4 December 2023; pp. 1253–1258.
25. Qiao, H.; Pang, G. Truncated affinity maximization: One-class homophily modeling for graph anomaly detection. In Proceedings of the Conference on Neural Information Processing Systems, New Orleans, LA, USA, 10–16 December 2023; Vol. 36, pp. 49490–49512.
26. Wang, D.; Lin, J.; Cui, P.; et al. A semi-supervised graph attentive network for financial fraud detection. In Proceedings of the IEEE International Conference on Data Mining, Beijing, China, 8–11 November 2019; pp. 598–607.
27. Ding, K.; Zhou, Q.; Tong, H.; et al. Few-shot Network Anomaly Detection via Cross-network Meta-learning. In Proceedings of The Web Conference, Ljubljana, Slovenia, 19–23 April 2021; pp. 2448–2456.
28. Pan, J.; Liu, Y.; Zhou, C.; et al. Correcting False Alarms from Unseen: Adapting Graph Anomaly Detectors at Test Time. In Proceedings of the AAAI Conference on Artificial Intelligence, Singapore, 20–27 January 2026.
29. Wang, H.; Wu, J.; Hu, W.; et al. Detecting and assessing anomalous evolutionary behaviors of nodes in evolving social networks. *ACM Trans. Knowl. Discov. Data* **2019**, *13*, 12.
30. Liu, Y.; Pan, S.; Wang, Y.G.; et al. Anomaly Detection in Dynamic Graphs via Transformer. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 12081–12094.
31. Ma, X.; Li, D.; Shao, M.; et al. LLM-enhanced energy contrastive learning for out-of-distribution detection in text-attributed graphs. In Proceedings of the AAAI Conference on Artificial Intelligence, Singapore, 20–27 January 2026; Vol. 40, pp. 24308–24316.
32. Xu, Y.; Chen, J.; Peng, Z.; et al. Court of LLMs: Evidence-Augmented Generation via Multi-LLM Collaboration for Text-Attributed Graph Anomaly Detection. In Proceedings of the 33rd ACM International Conference on Multimedia, Dublin, Ireland, 27–31 October 2025; pp. 2437–2446.
33. Pan, J.; Liu, Y.; Zheng, Y.; et al. CAMERA: Adapting to Semantic Camouflage in Unsupervised Text-Attributed Graph Fraud Detection. In Proceedings of the International Joint Conference on Artificial Intelligence, Bremen, Germany, 15–21 August 2026.
34. Liu, Y.; Ding, K.; Lu, Q.; et al. Towards self-interpretable graph-level anomaly detection. In Proceedings of the Advances in Neural Information Processing Systems 36 (NeurIPS 2023), New Orleans, LA, USA, 10–16 December 2023.
35. Shen, X.; Wang, Y.; Zhou, K.; et al. Optimizing ood detection in molecular graphs: A novel approach with diffusion models. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Barcelona, Spain, 25–29 August 2024; pp. 2640–2650.
36. Wang, Y.; Liu, Y.; Shen, X.; et al. Unifying unsupervised graph-level anomaly detection and out-of-distribution detection: A benchmark. In Proceedings of the International Conference on Learning Representations, Singapore, 24–28 April 2025.
37. Pan, J.; Zheng, Y.; Tan, Y.; et al. A Survey of Generalization of Graph Anomaly Detection: From Transfer Learning to Foundation Models. In Proceedings of the IEEE International Conference on Knowledge Graph, Limassol, Cyprus, 13–14 November 2025; pp. 316–323.
38. Liu, Y.; Li, S.; Zheng, Y.; et al. Arc: A generalist graph anomaly detector with in-context learning. In Proceedings of the Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 10–15 December 2024; Vol. 37, pp. 50772–50804.
39. Liu, Y.; Li, S.; Zheng, Y.; et al. From Few-Shot to Zero-Shot: Towards Generalist Graph Anomaly Detection. *IEEE Trans. Knowl. Data Eng.* **2026**, *38*, 4357–4372.
40. He, K.; Fan, H.; Wu, Y.; et al. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 9729–9738.
41. Chen, T.; Kornblith, S.; Norouzi, M.; et al. A simple framework for contrastive learning of visual representations. In Proceedings of the 37th International Conference on Machine Learning, Virtual Event, 13–18 July 2020; pp. 1597–1607.
42. Grill, J.B.; Strub, F.; Altché, F.; et al. Bootstrap Your Own Latent—A New Approach to Self-Supervised Learning. In Proceedings of the Advances in Neural Information Processing Systems 33 (NeurIPS 2020), Virtual Event, 6–12 December 2020; Vol. 33, pp. 21271–21284.
43. Liu, Y.; Jin, M.; Pan, S.; et al. Graph self-supervised learning: A survey. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 5879–5900.
44. Zheng, Y.; Jin, M.; Pan, S.; et al. Toward graph self-supervised learning with contrastive adjusted zooming. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *35*, 8882–8896.
45. Veličković, P.; Fedus, W.; Hamilton, W.L.; et al. Deep Graph Infomax. In Proceedings of the International Conference on

Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
46. Peng, Z.; Huang, W.; Luo, M.; et al. Graph representation learning via graphical mutual information maximization. In Proceedings of The Web Conference, Taipei, Taiwan, 20–24 April 2020; pp. 259–270.
47. Qiu, J.; Chen, Q.; Dong, Y.; et al. Gcc: Graph contrastive coding for graph neural network pre-training. In Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, 23–27 August 2020; pp. 1150–1160.
48. Zhu, Y.; Xu, Y.; Yu, F.; et al. Graph contrastive learning with adaptive augmentation. In Proceedings of The Web Conference, Ljubljana, Slovenia, 19–23 April 2021; pp. 2069–2080.
49. Jin, M.; Zheng, Y.; Li, Y.F.; et al. Multi-Scale Contrastive Siamese Networks for Self-Supervised Graph Representation Learning. In Proceedings of the International Joint Conference on Artificial Intelligence, Montreal, QC, Canada, 19–26 August 2021.
50. Wang, Y.; Min, Y.; Chen, X.; et al. Multi-view Graph Contrastive Representation Learning for Drug-Drug Interaction Prediction. In Proceedings of The Web Conference, Ljubljana, Slovenia, 19–23 April 2021; pp. 2921–2933.
51. Yu, J.; Yin, H.; Li, J.; et al. Self-Supervised Multi-Channel Hypergraph Convolutional Network for Social Recommendation. In Proceedings of The Web Conference, Ljubljana, Slovenia, 19–23 April 2021; pp. 413–424.
52. Jin, M.; Liu, Y.; Zheng, Y.; et al. ANEMONE: Graph Anomaly Detection with Multi-Scale Contrastive Learning. In Proceedings of the 30th ACM International Conference on Information and Knowledge Management, Gold Coast, QD, Australia, 1–5 November 2021; pp. 3122–3126.
53. Zheng, Y.; Jin, M.; Liu, Y.; et al. Generative and Contrastive Self-Supervised Learning for Graph Anomaly Detection. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 12220–12233.
54. Tong, H.; Faloutsos, C.; Pan, J.Y. Fast random walk with restart and its applications. In Proceedings of the Sixth International Conference on Data Mining (ICDM'06), Hong Kong, China, 18–22 December 2006; pp. 613–622.
55. Sen, P.; Namata, G.; Bilgic, M.; et al. Collective classification in network data. *AI Mag.* **2008**, *29*, 93–93.
56. Tang, J.; Zhang, J.; Yao, L.; et al. Arnetminer: extraction and mining of academic social networks. In Proceedings of the 14th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 990–998.
57. Tang, L.; Liu, H. Relational learning via latent social dimensions. In Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Paris, France, 28 June–1 July, 2009; pp. 817–826.
58. Ding, K.; Li, J.; Liu, H. Interactive anomaly detection on attributed networks. In Proceedings of the ACM International Conference on Web Search and Data Mining, Melbourne, VIC, Australia, 11–15 February 2019; pp. 357–365.
59. Song, X.; Wu, M.; Jermaine, C.; et al. Conditional anomaly detection. *IEEE Trans. Knowl. Data Eng.* **2007**, *19*, 631–645.
60. Pan, S.; Zheng, Y.; Liu, Y. Integrating graphs with large language models: Methods and prospects. *IEEE Intell. Syst.* **2024**, *39*, 64–68.