symmetry

MDPI

Article

Global-Local Complementary Fusion: Unsupervised Graph Anomaly Detection via Diffusion Reconstruction and Contrastive Learning

Ruibin Hu [1], Qian Chen [2,3], Huiying Xu [2,3,*], Ruidong Wang [4], Huazhen Jin [5,*], Xiao Huang [4] and Xinzhong Zhu [2,3]

[1] Science and Technology Department, Zhejiang Normal University, Jinhua 321004, China; huruibin@zjnu.cn
[2] Zhejiang Key Laboratory of Intelligent Education Technology and Application, Zhejiang Normal University, Jinhua 321004, China; chenqian2002@zjnu.edu.cn (Q.C.); zxz@zjnu.edu.cn (X.Z.)
[3] College of Computer Science and Technology, Zhejiang Normal University, Jinhua 321004, China
[4] University of Kansas Joint College of Education, Zhejiang Normal University, Jinhua 321004, China; iswangrd@zjnu.edu.cn (R.W.); huangxiao@zjnu.edu.cn (X.H.)
[5] Jinhua Audit Bureau, Jinhua 321000, China
* Correspondence: xhy@zjnu.edu.cn (H.X.); jinhuazhen@163.com (H.J.)

Abstract

Anomaly detection on attributed graphs is essential for scientific integrity, cybersecurity, and financial oversight, where abnormal patterns often manifest as breaks in structure or attributes. However, existing unsupervised methods are difficult to combine both global and local perspectives to detect anomalies. To address this issue, we propose DCGAD, a unified unsupervised framework that captures anomalies by fusing global reconstruction error and local view inconsistency. Our model leverages diffusion reconstruction to strengthen global semantic information, employing two parallel autoencoders to reconstruct the graph structure based on the original features and diffusion-enhanced features, respectively, to capture global structural differences. Complementarily, the model samples two local subgraph views per target node and uses multi-view contrastive learning to evaluate local contextual inconsistencies. By jointly optimizing these two complementary objectives, our proposed model achieves collaborative use of local and global information. Extensive experiments on six real-world graph datasets show that DCGAD outperforms other state-of-the-art approaches, achieving excellent scores on citation networks and significant gains on social and collaborative platforms.

Keywords: anomaly detection; graph neural networks; contrastive learning; graph diffusion; attributed graph

Academic Editor: Zhixun Su

Received: 28 April 2026
Revised: 24 May 2026
Accepted: 1 June 2026
Published: 3 June 2026

1. Introduction

Graph-structured data pervade virtually all real-world domains, including financial transaction systems, citation networks, social media platforms, and recommendation systems. Extracting meaningful patterns from these interconnected data sources has garnered extensive research attention, particularly for the task of graph anomaly detection, pinpointing entities (nodes, edges, or subgraphs) that depart markedly from the majority of normative patterns. Notable examples include detecting fraudulent transactions on financial networks [1] and identifying malicious accounts spreading misinformation on social networks [2], both of which are essential to protecting system integrity and security.

Nevertheless, graph anomaly detection presents distinct and non-trivial challenges. Real-world graphs inherently encode both intricate structural dependencies and rich node attribute information, such that anomalies can manifest in the structural space, the attribute space, or their intersection. Furthermore, ground-truth anomaly labels are typically scarce or absent in real-world scenarios, making fully supervised methods impractical [2]. These limitations have spurred a surge of research into unsupervised graph anomaly detection, from traditional shallow techniques to modern deep learning-based frameworks, with various methods continuously emerging.

Early shallow approaches predominantly depend on hand-engineered anomaly scoring metrics. AMEN [3] assesses node neighborhoods by integrating attribute and structural information to derive normality scores. Radar [4] learns a network-regularized regression function and uses regression residuals as anomaly indicators. ANOMALOUS [5] performs anomaly detection via residual analysis. While computationally efficient, these shallow methods lack the expressive power to model the complex non-linear interdependencies present in graph-structured data.

Deep learning-based approaches have achieved remarkable advances in graph anomaly detection. Autoencoder-based frameworks, exemplified by DOMINANT [6], use graph convolutional networks to encode joint structural and attributive information into latent embeddings, then reconstruct the input to generate reconstruction errors as anomaly scores. SpecAE [7] integrates spectral autoencoders with density estimation, while AEGIS [8] extends autoencoder-based detection to inductive settings. Nevertheless, most of these methods center on node-level reconstruction objectives and fail to fully leverage the rich contextual information from neighboring nodes and local subgraphs. As anomaly detection inherently seeks to identify patterns that deviate from normative behavior [9], contextual information is of paramount importance, yet it remains largely underutilized in existing autoencoder-based methods.

To address these limitations, we propose DCGAD, a novel unsupervised graph anomaly detection framework that jointly optimizes two complementary learning objectives. Our core idea is to design a collaborative dual-branch anomaly detection framework that can integrate local and global graph information and overcome the limitations of existing methods. Specifically, we first sample two localized subgraph views for each target node using random walks with restart. These views are then processed by a shared GNN encoder to obtain node and subgraph embeddings. The framework then jointly optimizes: (1) a graph diffusion reconstruction module that applies heat kernel diffusion to enhance large-scale structural patterns and reconstructs the adjacency matrix from both diffusion-augmented and original features, capturing global structural anomalies; and (2) a multi-view contrastive learning module that directly compares each target node against its surrounding subgraphs in the embedding space, identifying local contextual inconsistencies. Finally, anomaly scores are computed by combining the outputs from both modules, providing a comprehensive assessment of each node's abnormality. Figure 1 illustrates the overall workflow of DCGAD.

Our main contributions are summarized as follows:

- We develop a diffusion-enhanced reconstruction module that applies heat kernel diffusion to amplify global structural patterns, coupled with a parallel reconstruction branch that preserves original feature information. This design effectively captures anomalies that manifest as global structural inconsistencies.
- We design a multi-view contrastive learning module that directly compares each target node against its surrounding subgraph contexts, enabling effective identification of local contextual anomalies that complement the global perspective.

- We propose a novel unsupervised framework for graph anomaly detection that integrates graph diffusion reconstruction with multi-view contrastive learning, jointly optimizing these two complementary objectives for unsupervised anomaly detection on attributed graphs.
- Extensive experiments on six real-world benchmark datasets demonstrate that DCGAD consistently outperforms advanced methods by substantial margins, achieving near-perfect AUC scores on citation networks and significant improvements on social networks and collaborative platforms.

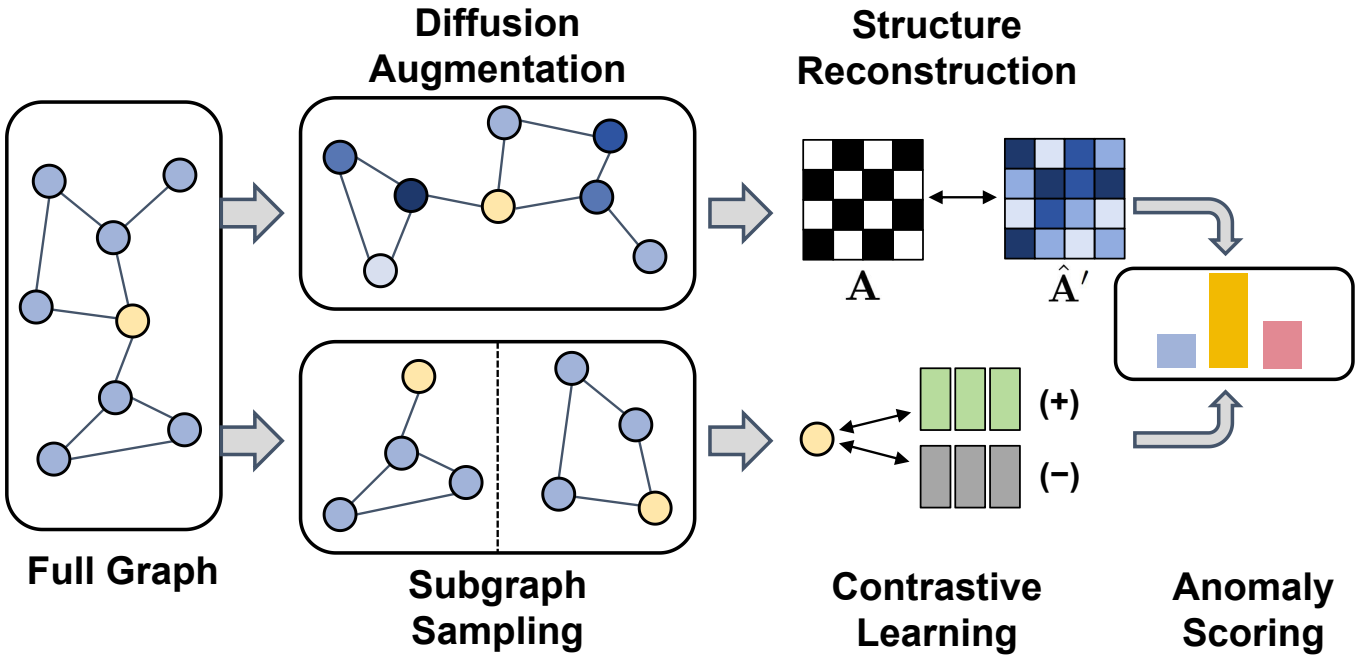


Figure 1. Overall workflow of the proposed DCGAD framework for graph anomaly detection. The abnormality of each node is evaluated from two complementary perspectives: diffusion-based reconstruction and multi-view contrastive learning.

The remainder of this paper is organized as follows. Section 2 reviews related work on graph anomaly detection and graph contrastive learning. Section 3 formulates the problem and defines key notations. Section 4 presents the proposed DCGAD methodology in detail. Section 5 reports and discusses experimental results. Section 6 concludes the paper and outlines future research directions.

2. Related Work

2.1. Graph Anomaly Detection

Anomaly detection in attributed graphs is critical for real-world applications such as financial fraud detection [10,11], cybersecurity monitoring, and social network analysis [12,13]. Early efforts relied on hand-crafted features and shallow models [5,14], but failed to capture complex non-linear relationships in high-dimensional graph data, limiting detection performance.

The rise of Graph Neural Networks (GNNs) has enabled end-to-end learning from structural and attributive information. CARE-GNN [15] uses adaptive neighbor filtering for fraud detection. AnomalyDAE [16] adopts a dual-autoencoder to reconstruct attributes and topology, using errors as anomaly signals. CONAD [17] integrates prior knowledge with partially labeled data via semi-supervised contrastive learning. SL-GAD [18] contrasts nodes with subgraphs to highlight deviations, while NLGAD [19] adds node–node com-

parisons. GADAM [20] decouples local inconsistency mining from message propagation for global-view detection.

Despite advances, most existing methods face two key challenges: insufficient utilization of global structural information and reliance on a single contrastive strategy that fails to detect anomalies across different views. Our DCGAD addresses these with a diffusion reconstruction module which amplifies global structural patterns via heat kernel diffusion and a multi-view contrastive module which captures local inconsistencies, enabling robust detection from complementary perspectives.

2.2. Graph Contrastive Learning

Graph contrastive learning (GCL) is a self-supervised paradigm that learns discriminative representations by contrasting positive/negative pairs, generating supervision from graph data without any labels. DGI [21] maximizes mutual information between node embeddings and a global graph summary. GCC [22] treats sampled subgraphs as contrastive instances, and iGCL [23] enforces view invariance without negative samples. LLNCL [24] dynamically updates structure and features via multi-view contrastive learning. These works show that GCL effectively captures local/global patterns without labels.

Recently, GCL has been adapted to anomaly detection. CoLA [25] identifies anomalies via node–subgraph consistency checks. ANEMONE [26] uses multi-scale node–node and node–subgraph comparisons with InfoNCE loss [27]. Sub-CR [28] combines multi-view contrastive learning with attribute reconstruction. GRADATE [29] introduces the subgraph-subgraph comparison and builds a multi-view, multi-scale network.

Our DCGAD differs from existing work in contrastive learning. The proposed multi-view contrastive module emphasizes semi-global contextual information by directly comparing the target node with its two sampled subgraph views, while the diffusion module operates on the full graph to capture global structural patterns. This synergistic design allows DCGAD to outperform methods that use only a single learning strategy.

3. Problem Definition

We formulate a formal definition of the unsupervised graph anomaly detection task. Consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ denotes the set of N nodes, and $\mathcal{E}$ represents the set of M edges. The node attributes and graph topology are encoded in the feature matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$ and the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, respectively. Specifically, $\mathbf{A}_{ij} = 1$ indicates an edge connecting v_i and v_j, while $\mathbf{A}_{ij} = 0$ otherwise. For unsupervised graph anomaly detection, the goal is to obtain a scoring function f without manual labels, which calculates an anomaly score for each node. The higher the anomaly score, the more likely the node is to be anomalous. A summary of all key notation and their explanation is provided in Table 1.

Table 1. Summary of the main notations in the paper and their corresponding descriptions.

Notations	Description
$\mathcal{G} = (\mathbf{X}, \mathbf{A})$	Attributed input graph
$\mathcal{V}, \mathcal{E}$	Node set and edge set of $\mathcal{G}$
$\mathbf{A} \in \mathbb{R}^{N \times N}$	Adjacency matrix of $\mathcal{G}$
$\mathbf{X} \in \mathbb{R}^{N \times D}$	Node feature matrix of $\mathcal{G}$
$\mathbf{x}_i \in \mathbb{R}^D$	Feature vector of node v_i (row of $\mathbf{X}$)
$\mathcal{N}(v_i)$	Neighbor set of node $v_i \in \mathcal{V}$

Table 1. *Cont.*

Notations	Description
$v_t \in \mathcal{V}$	Selected target node
$\mathcal{G}_t^{\phi_1}, \mathcal{G}_t^{\phi_2}$	Two sampled subgraph views centered at v_t
$\mathbf{X}_t^{\phi_i} \in \mathbb{R}^{K \times D}$	Node feature matrix of $\mathcal{G}_t^{\phi_i}$
$\mathbf{A}_t^{\phi_i} \in \mathbb{R}^{K \times K}$	Adjacency matrix of $\mathcal{G}_t^{\phi_i}$
$f(v_t)$	Final anomaly score of node v_t
$\mathbf{h}_t \in \mathbb{R}^{D'}$	Embedding vector of target node v_t
$\mathbf{s}_{\phi_i} \in \mathbb{R}^{D'}$	Subgraph-level embedding of view $\mathcal{G}_t^{\phi_i}$
$\mathbf{Z}_{\phi_i} \in \mathbb{R}^{K \times D'}$	Node embedding matrix of $\mathcal{G}_t^{\phi_i}$
$\mathbf{Z}_{\phi_i}[j,:] \in \mathbb{R}^{D'}$	Embedding of the j-th node in $\mathbf{Z}_{\phi_i}$
$\mathbf{W}_e \in \mathbb{R}^{D \times D'}$	Trainable weight matrix of GNN encoder
$\mathbf{W}_c \in \mathbb{R}^{D' \times D'}$	Trainable scoring matrix of contrastive discriminator
$\widetilde{\mathbf{X}} \in \mathbb{R}^{N \times D}$	Diffusion augmented node feature matrix
$\mathbf{H}_{rec} \in \mathbb{R}^{N \times D'}$	Latent attribute embedding from diffusion reconstruction
$\hat{\mathbf{A}}$, $\hat{\mathbf{A}}'$	Reconstructed adjacency matrices from two parallel branches
$\mathcal{L}_{dr}$	Diffusion reconstruction loss
$\mathcal{L}_{con}$	Multi-view contrastive loss
D	Dimension of raw node features
D'	Dimension of latent embeddings
α, β	Balancing coefficients for the two loss terms
p	Trade-off parameter for dual reconstruction views
K	Number of nodes in sampled subgraph view
R	Number of sampling rounds

4. Methodology

In this section, we describe the architecture of our proposed DCGAD, an unsupervised framework designed to detect anomalous nodes in attributed graphs. Figure 2 illustrates its four main modules: Graph View Construction, Graph Diffusion Reconstruction, Multi-View Contrastive Learning, and Anomaly Score Calculation. The process begins by selecting a target node and is followed by generating two local subgraph views centered around it. To fully capture both global structural inconsistencies and local contextual mismatches, we design two complementary tasks: a graph diffusion reconstruction module and a multi-view contrastive learning module. The former reconstructs node attributes and structural linkages via diffusion-enhanced autoencoders, where anomalies tend to exhibit larger reconstruction errors due to their deviation from global patterns. The latter directly contrasts the target node against its local subgraphs in the latent embedding space, emphasizing semi-local topological discrepancies. By jointly optimizing these two objectives, our model learns to identify anomalies from both global and local perspectives. During inference, two scoring functions derived from these modules are combined to assign higher anomaly scores to nodes that exhibit attributive or structural anomalies.

The remainder of this section is organized as follows. Sections 4.1–4.4 detail the four core components of DCGAD. Section 4.5 presents the overall algorithm and discusses its computational complexity.

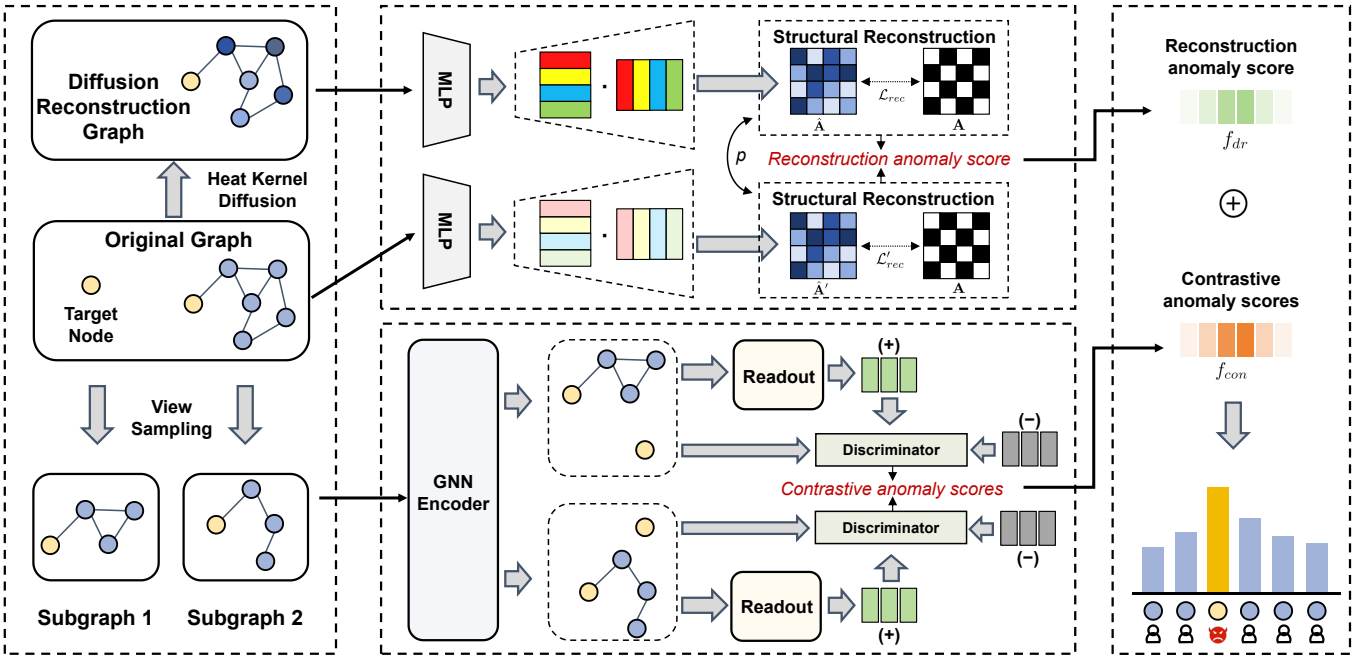


Figure 2. Overall architecture of the proposed DCGAD framework, comprising four main components: Graph view construction, graph diffusion reconstruction, multi-view contrastive learning, and anomaly score calculation. Given an input attributed graph, we first sample a target node and generate two localized subgraph views centered around it. The framework then jointly optimizes two unsupervised objectives: (1) a graph diffusion reconstruction module that reconstructs both node attributes and structural linkages via heat kernel diffusion and an MLP, and (2) a multi-view contrastive learning module that uses a bilinear discriminator to discriminate the target node against its surrounding subgraphs. Finally, anomaly scores are derived from both modules and aggregated to produce the final detection result, where abnormal nodes have higher anomaly scores.

4.1. Graph View Construction

Existing graph self-supervised learning research shows well-designed discrimination pairs are critical for encoders to capture structural and attributive information [21,30–32]. Like visual representation learning, these methods fall into two families: generative paradigms predict same-scale auxiliary properties (e.g., node attribute regression, graph structure prediction) [33]; contrastive methods distinguish instances within/across scales (e.g., "node vs. graph" comparisons) [30,34]. However, not all strategies directly apply to anomaly detection, as this task differs fundamentally from standard representation learning. Inspired by [25], we posit that anomalies often manifest as node–context mismatches, which form the basis of our graph view design.

To capture anomalous patterns from a complementary perspective, we design two learning tasks of different granularity. The first task performs global structure reconstruction, where we use heat kernel graph diffusion to enhance the full graph structure and measure the difference between the reconstructed representation matrix and the original matrix. The second task introduces a cross-view contrastive mechanism that pairs the target node with its locally sampled subgraphs, reasoning over both latent representations and topological structures [35]. As illustrated on the left side of Figure 2, the process begins with selecting a target node, followed by generating two subgraph views around it. While more than two views are feasible, they may introduce redundancy and degrade performance depending on the augmentations [30,36].

Our contrastive objective forms two pairs between the target node and the two subgraph views. Specifically, for node-level anomaly detection, we first sample a target node

from the input graph via uniform sampling without replacement. Existing augmentations like feature masking and edge perturbation [37] may introduce artificial anomalies by altering original linkages or attributes, making them unsuitable. Instead, we adopt random walks with restart (RWR) [38] for augmentation, which preserves underlying graph semantics. This method extracts fixed-size K subgraphs centered at the target node, controlling the receptive field of contextual information. We note that graph diffusion [39] is another promising augmentation direction for global information, which we leave for future work. Within each sampled subgraph, we anonymize the target node by zeroing out its features. This increases the difficulty of the two self-supervised pretext tasks and promotes robust training [33,40], preventing information leakage and encouraging the model to rely on contextual cues for anomaly identification.

4.2. Graph Diffusion Reconstruction

According to [6], the abnormality of an instance often manifests as the discrepancy between its original and reconstructed representations. This discrepancy can be measured via the ℓ_2-norm distance, where larger values indicate higher reconstruction errors and thus greater likelihood of being anomalous. Among various reconstruction representation methods, the autoencoder (AE) [41] is a classic and effective technique. It is a neural architecture initially designed for unsupervised representation learning, comprising two main components: an encoder and a decoder. For an input feature vector $\mathbf{x}$, a standard AE is formulated as:

$$\mathbf{x}' = \mathrm{AE}(\mathbf{x}) = \mathcal{D}(\mathcal{E}(\mathbf{x})), \tag{1}$$

where $\mathbf{x}' = \mathcal{D}(\mathbf{z})$ denotes the reconstructed feature vector of $\mathbf{x}$, $\mathbf{z} = \mathcal{E}(\mathbf{x})$ is the latent representation, and $\mathcal{E}(\cdot)$, $\mathcal{D}(\cdot)$ represent the encoder and decoder networks, respectively. The training objective of an AE is to minimize the discrepancy between $\mathbf{x}'$ and $\mathbf{x}$, typically achieved by reducing their ℓ_2-norm distance $\|\mathbf{x}' - \mathbf{x}\|_2^2$, which encourages the model to capture latent invariant patterns from the input data.

Nevertheless, when applied to graph representation learning, a conventional AE only reconstructs node attributes while overlooking structural information, rendering it suboptimal for this scenario. To address this deficiency, we design a graph structure diffusion reconstruction module.

Graph diffusion functions act as a low-pass filter, which enhances large-scale structural patterns while attenuating noise, thus exhibiting superior performance in structure reconstruction tasks. Among the diverse existing diffusion methods, heat kernel diffusion is extensively utilized, as it characterizes the thermodynamic information propagation process between nodes. When the time step t is sufficiently small, its staged diffusion procedure can be formulated as the forward Euler discretization of the continuous heat equation $\partial \mathbf{H}/\partial t = -\mathbf{L}\mathbf{H}$:

$$\widetilde{\mathbf{H}}_{t+1} = (\mathbf{I} - \lambda \mathbf{L})\widetilde{\mathbf{H}}_t, \tag{2}$$

where $\widetilde{\mathbf{H}}_0 = \mathbf{X}$, and λ controls the diffusion intensity. We iterate this process T times to approximate continuous-time diffusion, yielding $\widetilde{\mathbf{X}} = \widetilde{\mathbf{H}}_T$.

Within the structure autoencoder, a single-layer MLP projects the diffusion-enhanced attribute data into a latent attribute embedding $\mathbf{H}_{rec}$, defined as:

$$\mathbf{H}_{rec} = \sigma(\mathbf{W}^{\mathcal{S}}\widetilde{\mathbf{X}} + b^{\mathcal{S}}), \tag{3}$$

The reconstruction function $d_p(\cdot)$ outputs the reconstructed adjacency matrix $\hat{\mathbf{A}}$ from $\mathbf{H}_{rec}$, with the corresponding reconstruction loss $\mathcal{L}_{rec}$:

$$\hat{\mathbf{A}} = d_p(\mathbf{H}_{rec}) = \mathbf{H}_{rec}\mathbf{H}_{rec}^T, \tag{4}$$

$$\mathcal{L}_{rec} = \|\mathbf{A} - \hat{\mathbf{A}}\|_2, \tag{5}$$

where $d_p(\cdot)$ employs an inner product between $\mathbf{H}_{rec}$ and its transpose $\mathbf{H}_{rec}^T$ to ensure computational efficiency of reconstruction.

We concurrently train an autoencoder with the same architecture to reconstruct the original feature matrix $\mathbf{X}$. From this we obtain $\hat{\mathbf{A}}'$ and $\mathcal{L}'_{rec}$, defined respectively as:

$$\hat{\mathbf{A}}' = d_p(\mathbf{X}') = \mathbf{X}'\mathbf{X}'^T, \tag{6}$$

$$\mathcal{L}'_{rec} = \|\mathbf{A} - \hat{\mathbf{A}}'\|_2, \tag{7}$$

where $\mathbf{X}'$ denotes the latent attribute embedding obtained by reconstructing the original feature matrix $\mathbf{X}$ through an autoencoder. This parallel branch complements the reconstruction from a multi-view perspective.

Finally, the overall loss for graph diffusion reconstruction is formulated as:

$$\mathcal{L}_{dr} = p\mathcal{L}_{rec} + (1-p)\mathcal{L}'_{rec}. \tag{8}$$

where $p \in (0,1)$ serves as a trade-off hyperparameter balancing the two reconstruction views.

4.3. Multi-View Contrastive Learning

As stated previously, anomalous nodes often exhibit inconsistencies with their local neighborhoods. Although our diffusion reconstruction module captures global structural anomalies, it lacks sensitivity to fine-grained local context. To improve this, we design a multi-view contrastive module that contrasts each target node with its surrounding subgraphs, as shown in Figure 2.

Given two subgraph views $\mathcal{G}_t^{\phi_1}$ and $\mathcal{G}_t^{\phi_2}$ rooted at node v_t, we first compute their node representations using a one-layer graph convolutional network (GCN) [42]:

$$\mathbf{Z}_{\phi_i} = \sigma\big(\tilde{\mathbf{A}}_t^{\phi_i}\mathbf{X}_t^{\phi_i}\mathbf{W}_e\big), \quad i \in \{1,2\}, \tag{9}$$

where $\tilde{\mathbf{A}}_t^{\phi_i} = (\mathbf{D}_t^{\phi_i})^{-\frac{1}{2}}(\mathbf{A}_t^{\phi_i} + \mathbf{I})(\mathbf{D}_t^{\phi_i})^{-\frac{1}{2}}$ is the normalized adjacency with self-loops, $\mathbf{D}_t^{\phi_i}$ is the degree matrix, $\mathbf{W}_e$ is a trainable weight matrix, and σ denotes a ReLU activation. The target node v_t itself is encoded separately via a linear projection:

$$\mathbf{h}_t = \sigma(\mathbf{x}_t\mathbf{W}_e). \tag{10}$$

To obtain a subgraph-level summary for each view, we apply average pooling [30,43] over the node embeddings:

$$\mathbf{s}_{\phi_i} = \frac{1}{K}\sum_{j=1}^{K}\mathbf{Z}_{\phi_i}[j,:]. \tag{11}$$

We then construct positive pairs $(\mathbf{h}_t, \mathbf{s}_{\phi_i})$ and negative pairs $(\mathbf{h}_t, \mathbf{s}_{\phi_i}^{\text{neg}})$, where $\mathbf{s}_{\phi_i}^{\text{neg}}$ is the embedding of a randomly sampled subgraph different from $\mathcal{G}_t^{\phi_i}$. A bilinear discriminator [25] assigns a matching score to each pair:

$$c_t^{\phi_i} = \sigma(\mathbf{h}_t^\top\mathbf{W}_c\mathbf{s}_{\phi_i}), \qquad \tilde{c}_t^{\phi_i} = \sigma(\mathbf{h}_t^\top\mathbf{W}_c\mathbf{s}_{\phi_i}^{\text{neg}}), \tag{12}$$

with $\mathbf{W}_c$ a learnable matrix and σ the sigmoid function. The contrastive loss follows the Jensen–Shannon divergence formulation [21]:

$$\ell_{\text{con}}^{(j)} = -\frac{1}{2N}\sum_{i=1}^{N}\Big(\log c_i^{\phi_j} + \log(1 - \tilde{c}_i^{\phi_j})\Big), \quad j = 1,2, \tag{13}$$

and the overall contrastive objective combines the two views:

$$\mathcal{L}_{\text{con}} = \frac{1}{2}(\ell_{\text{con}}^{(1)} + \ell_{\text{con}}^{(2)}). \tag{14}$$

4.4. Anomaly Score Calculation

After training the two modules separately to avoid unbalanced optimization, we compute anomaly scores from each module and then fuse them. The diffusion reconstruction score for a node v_i is defined as the weighted sum of reconstruction errors from the two parallel branches:

$$f_{\text{dr}}(v_i) = p\|\mathbf{A}_i - \hat{\mathbf{A}}_i\|_1 + (1-p)\|\mathbf{A}_i - \hat{\mathbf{A}}_i'\|_1, \tag{15}$$

where p is the trade-off parameter. This score lies in $[0, 1]$; higher values indicate stronger global structural anomalies.

The contrastive anomaly score is derived from the discriminator outputs. For a normal node, the positive score $c_i^{\phi_j}$ should be near 1 while the negative score $\tilde{c}_i^{\phi_j}$ near 0. For an anomalous node, both scores tend to approach 0.5 due to contextual mismatch. Thus we define:

$$f_{\text{con}}(v_i) = \frac{1}{2}\sum_{j=1}^{2}\gamma(\tilde{c}_i^{\phi_j} - c_i^{\phi_j}), \tag{16}$$

where γ is a scaling factor that maps the value range $[-1, 0]$ to $[0, 1]$.

The final anomaly score for node v_i is obtained by averaging over R independent sampling rounds to reduce variance:

$$f(v_i) = \frac{1}{R}\sum_{r=1}^{R}\Big(\alpha f_{\text{con}}(v_i) + \beta f_{\text{dr}}(v_i)\Big), \tag{17}$$

with α, β balancing the two components. The same coefficients are used in the joint training loss:

$$\mathcal{L} = \alpha\mathcal{L}_{\text{con}} + \beta\mathcal{L}_{\text{dr}}. \tag{18}$$

4.5. Algorithm and Complexity Analysis

We hereby evaluate the computational overhead of DCGAD. The most time-consuming operations come from subgraph extraction and subsequent self-supervised learning procedures. For each target node v_t, employing RWR to obtain a local subgraph of size K incurs a cost of $\mathcal{O}(\eta K)$, where η denotes the average degree of the input graph. Once the subgraph views are constructed, the contrastive learning component processes each view with a complexity of $\mathcal{O}(K^2)$, primarily due to inner product operations in the discriminator. Meanwhile, the diffusion reconstruction module involves T iterative updates, each costing $\mathcal{O}(KD)$, where D is the feature dimension, and the subsequent adjacency matrix reconstruction requires $\mathcal{O}(K^2)$ for the inner product computation. Given that D is typically much lower than K, the overall per-node training cost is dominated by $\mathcal{O}(K^2 + TKD)$. Taking into account all N nodes, the total training complexity amounts to $\mathcal{O}(NK(K + \eta + TD))$. During inference, scoring each node involves constant-time operations per sampling round. Considering R independent sampling rounds for statistical robustness, the combined training and inference complexity of DCGAD is $\mathcal{O}(RNK(K + \eta + TD))$. We summarize the overall algorithm flow of DCGAD in Algorithm 1.

Algorithm 1 DCGAD: Overall Algorithm Procedure

Input: Attributed graph $\mathcal{G}$; Epoch limit E; Batch size B; Sampling rounds R. **Output:** Final anomaly scoring function $f(\cdot)$.

Initialize parameters $\mathbf{W}_e$ and $\mathbf{W}_c$ randomly;
// Training phase
for $epoch = 1$ to E **do**
  Partition $\mathcal{V}$ into mini-batches $\mathcal{B}$ each of size B;
  for each batch $b = (v_1, \cdots, v_B) \in \mathcal{B}$ **do**
    Generate two localized subgraphs for every node in b, denoted as $\{\mathcal{G}_i^{\phi_1}\}_{i=1}^{B}$ and $\{\mathcal{G}_i^{\phi_2}\}_{i=1}^{B}$;
    Obtain embeddings for target nodes and their associated subgraph views using Equations (9) and (10);
    Reconstruct node attributes via diffusion according to Equation (3);
    Compute total loss via $\mathcal{L}$ Equation (18) by combining Equations (8) and (14);
    Perform back-propagation to update $\mathbf{W}_e$ and $\mathbf{W}_c$;
  end for
end for
// Inference phase
for each $v_i \in \mathcal{V}$ **do**
  for $r = 1$ to R **do**
    Compute anomaly score $f(v_i)$ via Equations (15)–(17);
  end for
  Take the average of $f(v_i)$ over R rounds and record as the final score for v_i;
end for

5. Experimental Study

This section evaluates the proposed DCGAD model on six real-world benchmark datasets. We compare against state-of-the-art anomaly detection and contrastive learning approaches, adopting their original configurations to ensure fair comparisons. Additional ablation and parameter sensitivity studies are conducted to further analyze the characteristics of DCGAD.

5.1. Datasets

Table 2 summarizes the six real-world graph datasets used to evaluate DCGAD performance and its baselines. Detailed descriptions are provided below.

Table 2. The statistics of the datasets. There are the number of nodes, edges, features, and anomaly ratios in the table.

Dataset	Nodes	Edges	Features	Anomalies
Books [44]	1418	3695	21	2.0%
Reddit [44]	10,984	168,016	64	3.3%
ACM [12]	16,484	71,980	8337	3.6%
Cora [45]	2708	5429	1433	5.5%
CiteSeer [45]	3327	4732	3703	4.5%
Tolokers [45]	11,758	519,000	10	21.82%

- **Cora, Citeseer, and ACM**. These three datasets are widely adopted benchmarks in citation network analysis. The papers are treated as nodes, and the citations between them form the edges. Each paper is encoded as a sparse bag-of-words vector representing its textual content.
- **Books**. Built from Amazon's co-purchase records, this dataset captures book nodes with features including price, rating, and review count. The amazonfail tag information provides the source for ground-truth labeling.

- **Reddit**. It is a social network dataset that comprises 1000 active subreddits and 10,000 active users as nodes. Each user receives a binary label for whether they have been banned (anomaly). The text of the posts is converted into features; the feature vector for a user or subreddit is the sum of features from all their associated posts.
- **Tolokers**. This dataset is extracted from a collaborative work environment. The feature vector for each node comprises user profile information and task performance metrics.

Among the six datasets, Books, Reddit, and Tolokers provide genuine anomaly labels, enabling evaluation of DCGAD's ability to generalize to naturally occurring anomalies. The remaining three datasets, ACM, Cora and Citeseer, do not contain real anomaly labels, so we artificially inject synthetic anomalies following [25,33]. To maintain experimental integrity, all data are sourced from the original versions rather than the preprocessed ones commonly used in existing studies.

5.2. Experimental Setup

The experimental setups are described below, including the baseline methods, evaluation metrics, and parameter settings.

5.2.1. Baselines

We compare DCGAD with representative baselines under standard metrics. The baselines comprise two categories: (1) three autoencoder-based models, there are DOMINANT [6], AnomalyDAE [16], and AdONE [46]; and (2) six contrastive learning methods, there are CoLA [25], CONAD [17], Sub-CR [28], GRADATE [29], NLGAD [19], and GAD-NR [47]. Brief descriptions of the six state-of-the-art contrastive methods are provided below:

- **CoLA** [25]: Employs unsupervised contrastive learning to capture local relationships between nodes and their neighboring subgraphs, enabling effective detection of both structural and attribute anomalies.
- **CONAD** [17]: Eransforms abstract abnormal patterns into a computable format and incorporates prior knowledge via semi-supervised contrastive learning to enhance detection performance.
- **Sub-CR** [28]: Integrates multi-view contrastive learning with attribute reconstruction, enabling collaborative anomaly detection across topology and attributes.
- **GRADATE** [29]: Pioneers subgraph-subgraph comparison and combines it with node–subgraph and node–node comparisons to construct a multi-view, multi-scale detection framework.
- **NLGAD** [19]: Devises a hybrid strategy for selecting normal nodes from unlabeled data and adopts a two-stage training mechanism to strengthen the model's discrimination of anomalies.
- **GAD-NR** [47]: Introduces a novel graph autoencoder variant that takes neighborhood reconstruction as the primary indicator for graph anomaly detection.

5.2.2. Evaluation Metrics

We adopt AUC-ROC, a standard measure for anomaly detection tasks, to assess the performance of DCGAD and all baseline methods. The ROC curve plots the true positive rate (correctly flagged anomalies) against the false positive rate (normal nodes misidentified as anomalies). AUC quantifies the area beneath this curve, ranging within $[0, 1]$, where higher AUC values reflect better detection capability.

5.2.3. Parameter and Configuration Settings

For efficiency, the sampled subgraph size K is set to 4. The GNN encoder employs a one-layer GCN with hidden dimension $d = 128$. The coefficient α remains 1 in all datasets, while β is tuned from $\{0.2, 0.4, 0.6, 0.8, 1\}$. The discrimination modules are optimized via Adam. Learning rates are configured as follows: 0.001 for Cora, Citeseer and Reddit; 0.005 for ACM; 0.01 for Books and Tolokers. Training epochs are 80 for Citeseer, Books, Tolokers and ACM, and 100 for Cora and Reddit. The number of sampling rounds R is 256. Regarding the experimental environment, our implementation runs on Python 3.8 with PyTorch 1.13.1, CUDA 11.7, and DGL 1.1.2 + cu116 for graph data preprocessing. All experiments are conducted on an NVIDIA RTX 3090 GPU equipped with 24 GB memory.

5.3. Comparison Experiments

We evaluate DCGAD against six baselines in this subsection. The resulting AUC scores are summarized in Table 3, and Figure 3 illustrates the ROC curve of our method. From these results, we draw the following key observations:

Table 3. Performance comparison (AUC-ROC) between DCGAD and baseline methods on six benchmark datasets. We evaluate the model performance using the mean ± standard deviation of 10 independent experimental runs and **bold** the best result on each dataset.

Model	Tolokers	ACM	Cora	Citeseer	Books	Reddit
DOMINANT	0.5130 ± 0.0027	0.7466 ± 0.0029	0.8247 ± 0.0032	0.8383 ± 0.0165	0.5071 ± 0.0204	0.5622 ± 0.0058
AnomalyDAE	0.5942 ± 0.0115	0.8471 ± 0.0015	0.8980 ± 0.0141	0.8475 ± 0.0358	0.5514 ± 0.0597	0.5574 ± 0.0412
AdONE	0.5260 ± 0.0330	0.7881 ± 0.0046	0.8473 ± 0.0068	0.8236 ± 0.0097	0.5366 ± 0.0035	0.5075 ± 0.0029
CoLA	0.4531 ± 0.0152	0.8217 ± 0.0161	0.8740 ± 0.0252	0.8954 ± 0.0243	0.4187 ± 0.0708	0.5573 ± 0.0114
CONAD	0.4753 ± 0.0249	0.8495 ± 0.0008	0.8716 ± 0.0015	0.8501 ± 0.0002	0.5261 ± 0.0309	0.5642 ± 0.0003
Sub-CR	0.4939 ± 0.0038	0.7831 ± 0.0078	0.9092 ± 0.0037	0.9260 ± 0.0031	0.5729 ± 0.0086	0.5582 ± 0.0111
GRADATE	0.5223 ± 0.0222	0.8598 ± 0.0145	0.9220 ± 0.0130	0.8820 ± 0.0049	0.5486 ± 0.0093	0.5679 ± 0.0008
NLGAD	0.5591 ± 0.0118	0.8622 ± 0.0042	0.9184 ± 0.0035	0.9442 ± 0.0027	0.5354 ± 0.0101	0.5902 ± 0.0087
GAD-NR	0.4814 ± 0.0143	0.8298 ± 0.0459	0.8849 ± 0.0247	0.9060 ± 0.0139	0.6532 ± 0.0271	0.5799 ± 0.0092
DCGAD	**0.6231 ± 0.0153**	**0.9690 ± 0.0004**	**0.9720 ± 0.0011**	**0.9754 ± 0.0021**	**0.6897 ± 0.0019**	**0.6389 ± 0.0036**

- DCGAD achieves the best AUC-ROC scores across all six benchmark datasets, outperforming every baseline method by a substantial margin. On the ACM, Cora, and Citeseer datasets, our method attains near-perfect scores, representing improvements of over 10% compared to the second-best baselines. These results demonstrate that jointly optimizing diffusion reconstruction and multi-view contrastive learning enables DCGAD to effectively capture both global structural anomalies and local contextual inconsistencies.
- The visualization results in Figure 4 further corroborate the effectiveness of DCGAD. For each dataset, the anomaly scores assigned to abnormal nodes are consistently higher than those assigned to normal nodes, with clear separability between the two distributions. Notably, on the Tolokers dataset, which has a relatively high anomaly ratio, DCGAD still maintains strong discriminative power.
- Compared to autoencoder-based methods (DOMINANT, AnomalyDAE, AdONE), DCGAD demonstrates superior detection performance, particularly on high-dimensional feature datasets such as ACM and Cora. This advantage stems from our diffusion reconstruction module, which leverages heat kernel diffusion to enhance structural patterns while the parallel reconstruction branch complements the learning from a multi-view perspective, overcoming the limitations of conventional autoencoders that solely focus on attribute reconstruction.

- Among contrastive learning baselines, CoLA and GAD-NR achieve relatively competitive results on certain datasets. However, these methods rely on a single discrimination strategy (node–subgraph contrast or neighborhood reconstruction), which limits their ability to detect anomalies that manifest differently across structural and attribute spaces. DCGAD addresses this limitation by integrating complementary self-supervised objectives, enabling more robust anomaly identification.
- DCGAD achieves its most significant performance gains on citation networks (ACM, Cora, Citeseer), with AUC improvements of 10.68%, 5.00%, and 3.12% over the second-best methods, respectively. The ROC curves in Figure 3 confirm this trend, showing that DCGAD maintains high true positive rates even at low false positive rates on these datasets. On social network datasets (Reddit, Books) and the collaborative platform dataset (Tolokers), our method also achieves consistent improvements, demonstrating strong generalization across diverse graph types.
- A clear performance gap exists between datasets with injected anomalies (ACM, Cora, Citeseer) and those with real anomalies (Reddit, Tolokers, Books). DCGAD attains AUC scores greater than 0.96 in the former but less than 0.7 in the latter. This discrepancy is mainly attributed to the nature of the injected anomalies, which are typically generated by systematic structural perturbations or attribute flips, producing strong and consistent deviation patterns that align well with the dual detection mechanisms of DCGAD. In contrast, real-world anomalies such as banned users or unreliable contributors are often subtle and exhibit high heterophily, not always manifesting as obvious structural or contextual mismatches. This observation suggests that while DCGAD excels in standard benchmarks, its practical deployment in real-world graphs may benefit from domain-specific feature engineering or weak supervision to robustly capture diverse anomaly patterns.

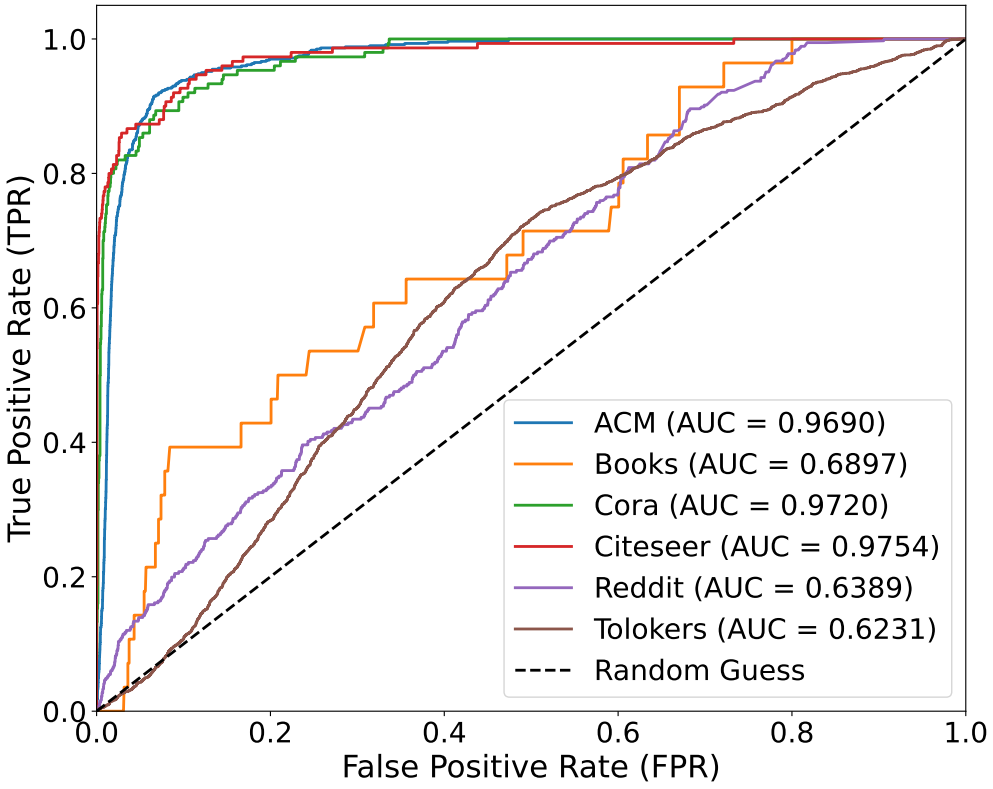


Figure 3. The AUC-ROC curves of the DCGAD model on six benchmark datasets. The diagonal dashed line represents random guessing (AUC = 0.5).

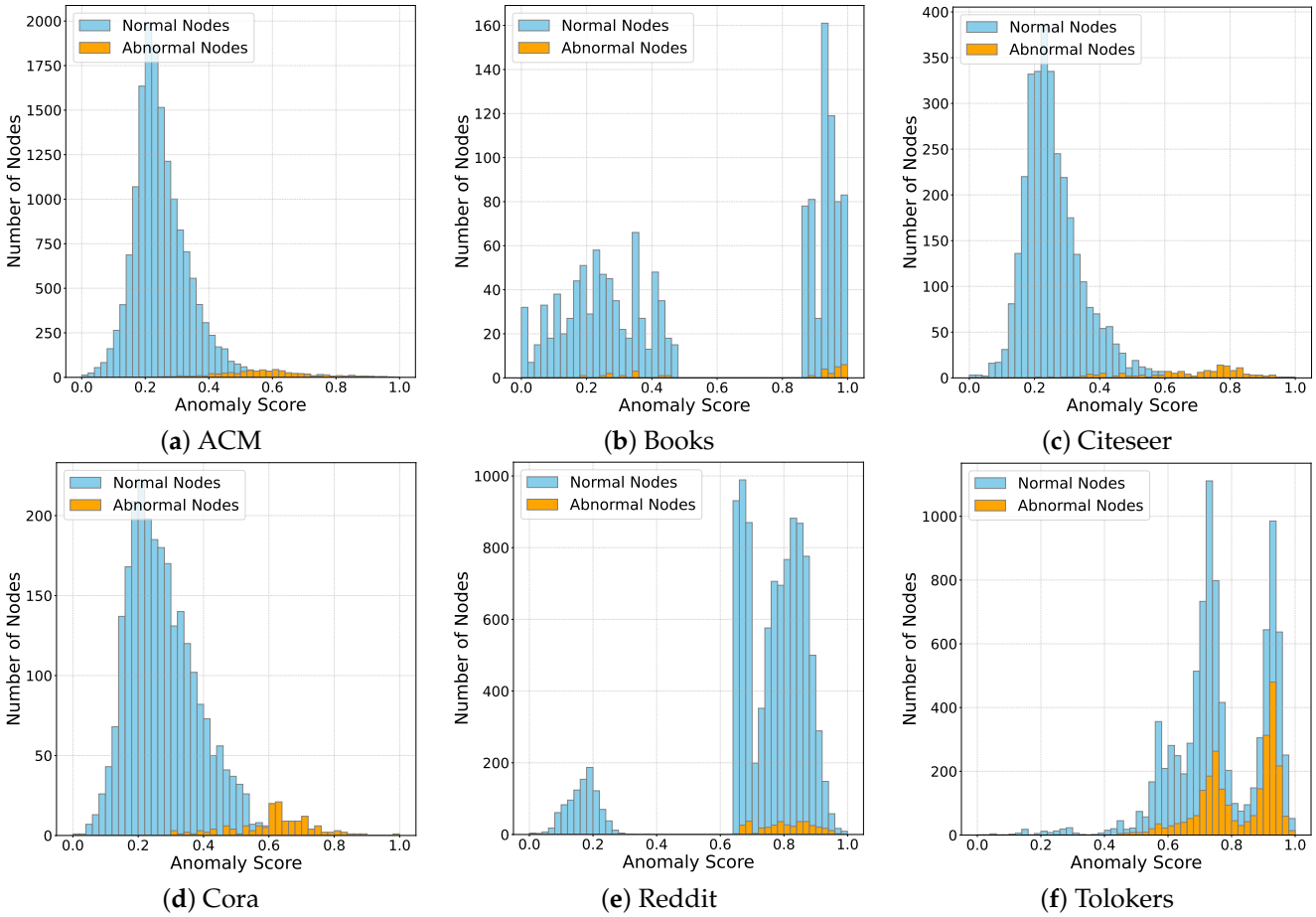


Figure 4. Visualization results of anomaly detection by DCGAD on six benchmark datasets, where blue represents the score distribution of normal nodes, and yellow represents the score distribution of abnormal nodes.

5.4. Ablation Study

We conduct ablation experiments to assess the contribution of each core component in DCGAD. Six simplified variants are constructed: (1) w/o GDR, which removes the graph diffusion reconstruction module and relies solely on multi-view contrastive learning; (2) w/o GCL, which discards the contrastive learning module and only retains diffusion reconstruction; (3) w/PPR, which uses Personalized PageRank to replace the heat kernel diffusion operator; (4) w/Max-P, which replaces average pooling with maximum pooling; (5) w/Sum-P, which replaces average pooling with sum pooling; (6) w/Att-P, which replaces average pooling with graph attention-based pooling. The experimental results are reported in Table 4, leading to the following observations:

- The complete DCGAD consistently achieves the highest AUC across all six datasets, confirming that both modules contribute synergistically to anomaly detection. Comparing the two variants, w/o GCL exhibits significantly lower performance than w/o GDR on every dataset, suggesting that contrastive learning contributes more to anomaly detection than diffusion reconstruction in our framework. Nevertheless, removing either component results in suboptimal performance, validating that the two modules address complementary aspects of graph anomalies.
- Removing the contrastive learning module (w/o GCL) causes a dramatic performance degradation across all datasets, with average AUC reduction of approximately 13.4%. This indicates that the multi-view contrastive learning component, which captures local contextual inconsistencies by comparing target nodes with their surrounding subgraphs, plays an indispensable role in DCGAD's detection capability.

- The diffusion reconstruction module also contributes substantially to overall performance. When this module is removed (w/o GDR), the AUC drops by 3.2% on average. This demonstrates that leveraging heat kernel diffusion to capture global structural anomalies effectively complements the contrastive learning objective, particularly on datasets where structural information is more discriminative.
- Replacing heat kernel diffusion with Personalized PageRank (w/PPR) causes a consistent and significant performance drop across all datasets, with an average AUC decline of approximately 10.3%. The degradation is severe on Cora and ACM, where the structural anomalies introduced by edge rewiring rely on a globally coherent diffusion signal to be detected. PPR incorporates a restart mechanism that biases the propagated information toward the local neighborhood and high-degree nodes, which weakens the module's ability to establish a faithful global structural baseline.
- Compared with the three alternative pooling strategies, average pooling achieves the best result across all six datasets. Sum pooling (w/Sum-P) yields competitive scores on citation networks, but degrades moderately on real anomaly datasets. Max pooling (w/Max-P) performs consistently worse, indicating that discarding non-maximum features discards valuable contextual signals. Notably, the attention-based pooling variant (w/Att-P) underperforms average pooling in all datasets, with particularly severe drops on ACM, Cora and Reddit. This degradation is likely due to overfitting of the additional attention parameters, especially in subgraphs with high structural heterophily where attention weights may amplify noise. These results confirm that the simple average pooling readout is not a performance bottleneck for DCGAD; rather, it provides a robust and stable representation of local subgraph contexts.

Table 4. The results of the ablation experiment to investigate the effectiveness of each component on DCGAD. All results are the average of ten independent experimental runs and we **bold** the best result on each dataset.

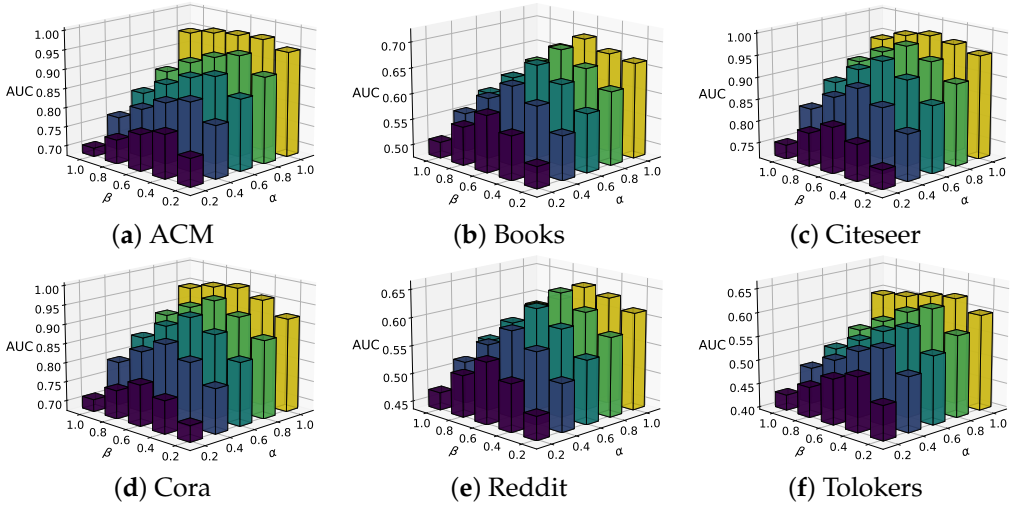
Variants	Tolokers	ACM	Cora	Citeseer	Books	Reddit
w/o GDR	0.5900	0.9366	0.9347	0.9323	0.6126	0.6121
w/o GCL	0.5351	0.8271	0.8782	0.8475	0.5644	0.5585
w/PPR	0.4832	0.8350	0.7879	0.8805	0.5265	0.5427
w/Max-P	0.6069	0.7782	0.7071	0.6495	0.6424	0.4532
w/Sum-P	0.6049	0.9510	0.9698	0.9749	0.6416	0.6366
w/Att-P	0.5846	0.4538	0.3817	0.7319	0.5826	0.5675
DCGAD	**0.6231**	**0.9690**	**0.9720**	**0.9754**	**0.6897**	**0.6389**

5.5. Parameters Sensitivity

We perform sensitivity analyses to evaluate how key hyperparameters affect the detection performance of DCGAD, including balance factors α and β, number of sampling rounds R, subgraph size K, hidden dimension d, diffusion reconstruction trade-off parameter p, and GCN layers L.

5.5.1. Balance Factors

We investigate the joint effect of α and β in Equations (17) and (18) on all six datasets. Figure 5 presents representative results on ACM, Books, Citeseer, Cora, Reddit, and Tolokers. Several observations can be made. First, the AUC values exhibit a clear upward trend as α increases when β is fixed, indicating that the contrastive learning module contributes positively to anomaly detection across all datasets. Second, the optimal β varies by dataset. Based on these observations, we fix $\alpha = 1$ and tune β from $\{0.2, 0.4, 0.6, 0.8, 1.0\}$ per dataset.



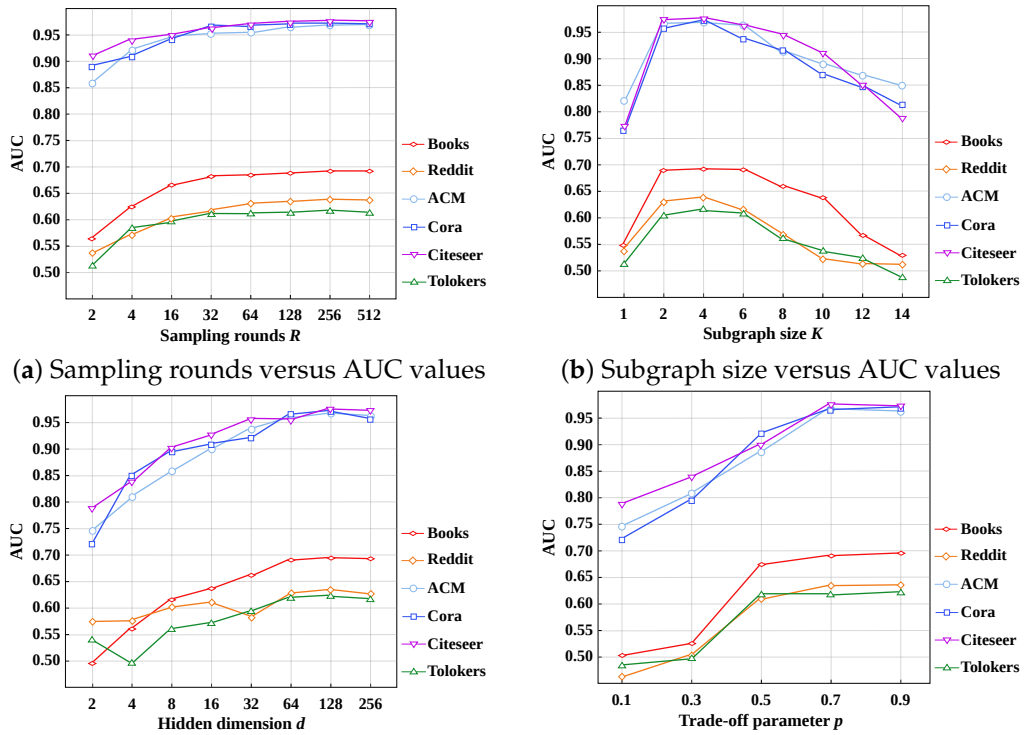
(**a**) ACM (**b**) Books (**c**) Citeseer

(**d**) Cora (**e**) Reddit (**f**) Tolokers

Figure 5. AUC values of DCGAD on six datasets under varying α and β weights. Warmer colors correspond to higher AUC values.

5.5.2. Sampling Rounds

Figure 6a illustrates how the number of sampling rounds R influences anomaly detection performance, with R varied from 2 to 512. The results demonstrate that performance improves substantially as R increases from small values, stabilizing around $R = 128$ for most datasets. Beyond $R = 256$, performance gains become marginal. This behavior confirms that multiple sampling rounds are necessary to obtain statistically stable anomaly scores by reducing the bias introduced by random subgraph sampling. Considering both effectiveness and computational efficiency, we set $R = 256$ for all experiments.



(**a**) Sampling rounds versus AUC values (**b**) Subgraph size versus AUC values

(**c**) Hidden dimension versus AUC values (**d**) Trade-off parameter versus AUC values

Figure 6. Parameter sensitivity analysis of DCGAD on six benchmark datasets is presented. Subfigure (**a**) illustrates the performance variation with different sampling rounds, while keeping other hyperparameters at their default values as specified in Section 5.2. Likewise, subfigures (**b**–**d**) examine the effects of subgraph size, hidden dimension, and the trade-off parameter, respectively.

5.5.3. Subgraph Size

We examine the sensitivity of subgraph size K ranging from 1 to 14, with results reported in Figure 6b. The optimal K varies across datasets but generally falls within $\{2, 4\}$. When $K = 1$, the subgraph contains only the target node itself, providing no contextual information and resulting in significantly degraded performance (e.g., AUC drops to 0.74 on ACM and 0.54 on Reddit). As K increases beyond the optimal value, performance gradually declines due to the inclusion of noisy or irrelevant nodes from distant neighborhoods. These results suggest that a modest subgraph size ($K = 2$ or 4) strikes the best balance between capturing sufficient contextual information and avoiding extraneous noise. We adopt $K = 4$ as the default setting.

5.5.4. Hidden Dimension

Figure 6c shows the effect of hidden dimension d (ranging from 2 to 256) on detection performance. For all datasets, AUC improves steadily as d increases from 2 to 32, indicating that higher-dimensional embeddings enable better representation of node attributes and structural patterns. Optimal performance is achieved at $d = 128$ for most datasets. When d exceeds 128, performance either plateaus or declines slightly, likely due to overfitting caused by excessive parameterization. Given that $d = 128$ consistently yields near-optimal results across all datasets, we set the hidden dimension to 128 for all experiments.

5.5.5. Trade-Off Parameter

We analyze the sensitivity of the diffusion reconstruction trade-off parameter p in Equation (8), which balances the two parallel reconstruction branches. As shown in Figure 6d, p is varied from 0.1 to 0.9. The results reveal a consistent trend: performance improves substantially as p increases from 0.1 to 0.7, after which it plateaus or increases marginally. At $p = 0.1$ (heavily favoring the original feature reconstruction branch), AUC values are notably low across all datasets, e.g., 0.74 on ACM and 0.48 on Citeseer. At $p = 0.5$, performance improves but remains suboptimal. Peak performance is generally achieved around $p = 0.7$ for most datasets, indicating that the diffusion-augmented reconstruction branch provides stronger structural anomaly signals than the original feature reconstruction branch. Based on these observations, we set $p = 0.7$ as the default configuration.

5.5.6. GCN Layers

To investigate whether deeper GCN encoders can bring additional gains, we evaluate DCGAD with two-layer and three-layer GCN configurations on all six datasets. The results are reported in Table 5. It can be observed that the single-layer GCN consistently outperforms both deeper variants across every dataset. Performance degradation is particularly pronounced on three real anomaly datasets with lower AUC scores.

We attribute this behavior to two main reasons. First, anomaly detection on attributed graphs often relies on fine-grained local discrepancies between a node and its immediate neighbors. Deeper GCN layers aggregate information from multi-hop neighborhoods, which can over-smooth the node representations and dilute the local irregularity that is essential for distinguishing anomalous nodes from normal ones. Second, in datasets with high heterophily, such as Tolokers and Reddit, nodes further away in the graph may carry irrelevant or noisy features. Incorporating these distant signals through additional GCN layers introduces confounding information that reduces the quality of both contrastive and reconstruction objectives. Therefore, a single-layer GCN strikes the best balance by capturing sufficient local context while avoiding excessive smoothing and noise propagation.

Table 5. AUC values of DCGAD on six datasets under varying GCN Layers. All results are the average of ten independent experimental runs and we **bold** the best result on each dataset.

GCN Layers	Tolokers	ACM	Cora	Citeseer	Books	Reddit
Two-layer	0.4044	0.9311	0.8565	0.4868	0.6542	0.5501
Three-layer	0.4646	0.8603	0.7405	0.4018	0.6322	0.5188
One-layer (ours)	**0.6231**	**0.9690**	**0.9720**	**0.9754**	**0.6897**	**0.6389**

6. Conclusions

In this paper, we investigate the problem of unsupervised graph anomaly detection. We observed that existing unsupervised methods either rely solely on attribute reconstruction or employ single-scale contrastive learning, failing to jointly capture global structural inconsistencies and local contextual mismatches. To address this limitation, we proposed DCGAD, a novel unsupervised framework that integrates graph diffusion reconstruction with multi-view contrastive learning. We first generate two localized subgraph views centered on each target node. Within the framework, the diffusion reconstruction module utilizes heat kernel diffusion to refine the global graph structure and reconstructs the adjacency matrix using diffusion-augmented features and original features, thereby precisely capturing global structural anomalies. Complementarily, the multi-view contrastive learning module directly contrasts the target node with its contextual neighborhood subgraph in the latent embedding space, enabling reliable detection of local structural inconsistencies. Extensive experiments conducted on six widely adopted real-world graph datasets verify that the proposed DCGAD method consistently outperforms existing state-of-the-art approaches. In particular, our approach yields nearly perfect AUC results on citation network datasets and achieves remarkable performance improvements on social networks and collaborative interaction platforms.

The current framework is primarily designed for static attributed graphs. In future work, we plan to extend DCGAD to dynamic graph settings [48], where the temporal evolution of anomalies can provide richer signals for detection. Additionally, we aim to develop a unified framework that simultaneously detects anomalies at multiple levels [49] (graph-level, node-level, and edge-level), enabling more comprehensive anomaly identification in different granularities of graph data.

Author Contributions: Conceptualization, X.H.; methodology, Q.C. and H.X.; software, Q.C.; validation, R.H. and Q.C.; formal analysis, R.H. and Q.C.; investigation, R.H. and H.J.; resources, X.Z. and H.J.; data curation, R.H. and H.X.; writing—original draft preparation, Q.C.; writing—review and editing, X.Z. and R.W.; visualization, Q.C.; supervision, X.Z. and H.X.; project administration, X.H. and H.J.; funding acquisition, X.Z., X.H. and H.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (62376252); Key Project of Natural Science Foundation of Zhejiang Province (LZ22F030003); Zhejiang Province Leading Geese Plan (2025C02025, 2025C01056); Zhejiang Province Province-Land Synergy Program (2025SDXT004-3) and National Social Science Fund of China (BHA210121).

Data Availability Statement: The original data presented in the study are openly available at https://github.com/mala-lab/Awesome-Deep-Graph-Anomaly-Detection (accessed on 28 April 2026).

Conflicts of Interest: Author Huazhen Jin was employed by the company "Jinhua Audit Bureau". The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Pourhabibi, T.; Ong, K.-L.; Kam, B.H.; Boo, Y.L. Fraud Detection: A Systematic Literature Review of Graph-Based Anomaly Detection Approaches. *Decis. Support Syst.* **2020**, *133*, 113303. [CrossRef]
2. Latah, M. Detection of Malicious Social Bots: A Survey and a Refined Taxonomy. *Expert Syst. Appl.* **2020**, *151*, 113383. [CrossRef]
3. Perozzi, B.; Akoglu, L. Scalable Anomaly Ranking of Attributed Neighborhoods. In *Proceedings of the 2016 SIAM International Conference on Data Mining*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2016; pp. 207–215.
4. Li, J.; Dani, H.; Hu, X.; Liu, H. Radar: Residual Analysis for Anomaly Detection in Attributed Networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*; International Joint Conferences on Artificial Intelligence: Pasadena, CA, USA, 2017; pp. 2152–2158.
5. Peng, Z.; Luo, M.; Li, J.; Liu, H.; Zheng, Q. ANOMALOUS: A Joint Modeling Approach for Anomaly Detection on Attributed Networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*; International Joint Conferences on Artificial Intelligence: Pasadena, CA, USA, 2018; pp. 3513–3519.
6. Ding, K.; Li, J.; Bhanushali, R.; Liu, H. Deep Anomaly Detection on Attributed Networks. In *Proceedings of the 2019 SIAM International Conference on Data Mining*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2019; pp. 594–602.
7. Li, Y.; Huang, X.; Li, J.; Du, M.; Zou, N. SpecAE: Spectral AutoEncoder for Anomaly Detection in Attributed Networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 2233–2236.
8. Ding, K.; Li, J.; Agarwal, N.; Liu, H. Inductive Anomaly Detection on Attributed Networks. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*; International Joint Conferences on Artificial Intelligence: Pasadena, CA, USA, 2020; pp. 1288–1294.
9. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly Detection: A Survey. *ACM Comput. Surv.* **2009**, *41*, 1–58. [CrossRef]
10. Wang, D.; Lin, J.; Cui, P.; Jia, Q.; Wang, Z.; Fang, Y.; Yu, Q.; Zhou, J.; Yang, S.; Qi, Y. A Semi-Supervised Graph Attentive Network for Financial Fraud Detection. In *Proceedings of the 2019 IEEE International Conference on Data Mining (ICDM)*; IEEE: Piscataway, NJ, USA, 2019; pp. 598–607.
11. Liu, Y.; Ao, X.; Qin, Z.; Chi, J.; Feng, J.; Yang, H.; He, Q. Pick and Choose: A GNN-Based Imbalanced Learning Approach for Fraud Detection. In *Proceedings of the Web Conference 2021*; Association for Computing Machinery: New York, NY, USA, 2021; pp. 3168–3177.
12. Tang, J.; Zhang, J.; Yao, L.; Li, J.; Zhang, L.; Su, Z. Arnetminer: Extraction and Mining of Academic Social Networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; Association for Computing Machinery: New York, NY, USA, 2008; pp. 990–998.
13. Wang, R.; Zhang, F.; Huang, X.; Tian, C.; Xi, L.; Fan, H. CaCo: Attributed Network Anomaly Detection via Canonical Correlation Analysis. *IEEE Trans. Ind. Inform.* **2024**, *20*, 461–470. [CrossRef]
14. Müller, E.; Sánchez, P.I.; Mülle, Y.; Böhm, K. Ranking Outlier Nodes in Subspaces of Attributed Graphs. In *Proceedings of the 2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW)*; IEEE: Piscataway, NJ, USA, 2013; pp. 216–222.
15. Dou, Y.; Liu, Z.; Sun, L.; Deng, Y.; Peng, H.; Yu, P.S. Enhancing Graph Neural Network-Based Fraud Detectors against Camouflaged Fraudsters. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*; Association for Computing Machinery: New York, NY, USA, 2020; pp. 315–324.
16. Fan, H.; Zhang, F.; Li, Z. Anomalydae: Dual Autoencoder for Anomaly Detection on Attributed Networks. In *Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: Piscataway, NJ, USA, 2020; pp. 5685–5689.
17. Xu, Z.; Huang, X.; Zhao, Y.; Dong, Y.; Li, J. Contrastive Attributed Network Anomaly Detection with Data Augmentation. In *Advances in Knowledge Discovery and Data Mining*; Springer International Publishing: Cham, Switzerland, 2022; Volume 13281, pp. 444–457.
18. 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. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 12220–12233. [CrossRef]
19. Duan, J.; Zhang, P.; Wang, S.; Hu, J.; Jin, H.; Zhang, J.; Zhou, H.; Liu, X. Normality Learning-Based Graph Anomaly Detection via Multi-Scale Contrastive Learning. In *Proceedings of the 31st ACM International Conference on Multimedia*; Association for Computing Machinery: New York, NY, USA, 2023; pp. 7502–7511.
20. Chen, J.; Zhu, G.; Yuan, C.; Huang, Y. Boosting Graph Anomaly Detection with Adaptive Message Passing. In Proceedings of the Twelfth International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024.
21. Veličković, P.; Fedus, W.; Hamilton, W.L.; Liò, P.; Bengio, Y.; Hjelm, R.D. Deep Graph Infomax. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
22. Qiu, J.; Chen, Q.; Dong, Y.; Zhang, J.; Yang, H.; Ding, M.; Wang, K.; Tang, J. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*; Association for Computing Machinery: New York, NY, USA, 2020; pp. 1150–1160.

23. Li, H.; Cao, J.; Zhu, J.; Luo, Q.; He, S.; Wang, X. Augmentation-Free Graph Contrastive Learning of Invariant-Discriminative Representations. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 11157–11167. [CrossRef]
24. He, L.; Cheng, D.; Zhang, G.; Zhang, S. Leveraging Long-Range Nodes in Multi-View Graph Contrastive Learning. *Inf. Fusion* **2025**, *122*, 103186. [CrossRef]
25. Liu, Y.; Li, Z.; Pan, S.; Gong, C.; Zhou, C.; Karypis, G. Anomaly Detection on Attributed Networks via Contrastive Self-Supervised Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 2378–2392. [CrossRef] [PubMed]
26. Jin, M.; Liu, Y.; Zheng, Y.; Chi, L.; Li, Y.-F.; Pan, S. ANEMONE: Graph Anomaly Detection with Multi-Scale Contrastive Learning. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*; Association for Computing Machinery: New York, NY, USA, 2021; pp. 3122–3126.
27. van den Oord, A.; Li, Y.; Vinyals, O. Representation Learning with Contrastive Predictive Coding. *arXiv* **2018**, arXiv:1807.03748.
28. Zhang, J.; Wang, S.; Chen, S. Reconstruction Enhanced Multi-View Contrastive Learning for Anomaly Detection on Attributed Networks. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*; International Joint Conferences on Artificial Intelligence: Pasadena, CA, USA, 2022; pp. 2376–2382.
29. 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 *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI Press: Washington, DC, USA, 2023; Volume 37, pp. 7459–7467.
30. Hassani, K.; Khasahmadi, A.H. Contrastive Multi-View Representation Learning on Graphs. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2020; pp. 4116–4126.
31. Hu, W.; Liu, B.; Gomes, J.; Zitnik, M.; Liang, P.; Pande, V.; Leskovec, J. Strategies for Pre-Training Graph Neural Networks. In Proceedings of the International Conference on Learning Representations, Virtual, 26 April–1 May 2020.
32. You, Y.; Chen, T.; Wang, Z.; Shen, Y. When Does Self-Supervision Help Graph Convolutional Networks? In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2020; pp. 10871–10880.
33. Liu, Y.; Pan, S.; Jin, M.; Zhou, C.; Xia, F.; Yu, P.S. Graph Self-Supervised Learning: A Survey. *arXiv* **2021**, arXiv:2103.00111. [CrossRef]
34. Jiao, Y.; Xiong, Y.; Zhang, J.; Zhang, Y.; Zhang, T.; Zhu, Y. Sub-Graph Contrast for Scalable Self-Supervised Graph Representation Learning. In *Proceedings of the 2020 IEEE International Conference on Data Mining (ICDM)*; IEEE: Piscataway, NJ, USA, 2020; pp. 222–231.
35. Tian, Y.; Krishnan, D.; Isola, P. Contrastive Multiview Coding. In *Proceedings of the Computer Vision—ECCV 2020*; Springer International Publishing: Cham, Switzerland, 2020; Volume 12351, pp. 776–794.
36. Tosh, C.; Krishnamurthy, A.; Hsu, D. Contrastive Learning, Multi-View Redundancy, and Linear Models. In *Proceedings of the Algorithmic Learning Theory*; PMLR: Cambridge, MA, USA, 2021; pp. 1179–1206.
37. Jin, M.; Zheng, Y.; Li, Y.-F.; Gong, C.; Zhou, C.; Pan, S. Multi-Scale Contrastive Siamese Networks for Self-Supervised Graph Representation Learning. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*; International Joint Conferences on Artificial Intelligence: Pasadena, CA, USA, 2021; pp. 2577–2583.
38. 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*; IEEE: Piscataway, NJ, USA, 2006; pp. 613–622.
39. Klicpera, J.; Weißenberger, S.; Günnemann, S. Diffusion Improves Graph Learning. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32, pp. 13354–13366.
40. You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; Shen, Y. Graph Contrastive Learning with Augmentations. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 5812–5823.
41. Zhou, C.; Paffenroth, R.C. Anomaly Detection with Robust Deep Autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; Association for Computing Machinery: New York, NY, USA, 2017; pp. 665–674.
42. 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.
43. Ying, Z.; You, J.; Morris, C.; Ren, X.; Hamilton, W.; Leskovec, J. Hierarchical Graph Representation Learning with Differentiable Pooling. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2018; Volume 31, pp. 4800–4810.
44. Tang, L.; Liu, H. Relational Learning via Latent Social Dimensions. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; Association for Computing Machinery: New York, NY, USA, 2009; pp. 817–826.
45. Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; Eliassi-Rad, T. Collective Classification in Network Data. *AI Mag.* **2008**, *29*, 93–106. [CrossRef]
46. Bandyopadhyay, S.; N, L.; Vivek, S.V.; Murty, M.N. Outlier Resistant Unsupervised Deep Architectures for Attributed Network Embedding. In *Proceedings of the 13th International Conference on Web Search and Data Mining*; Association for Computing Machinery: New York, NY, USA, 2020; pp. 25–33.

47. Roy, A.; Shu, J.; Li, J.; Yang, C.; Elshocht, O.; Smeets, J.; Li, P. GAD-NR: Graph Anomaly Detection via Neighborhood Reconstruction. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 576–585.
48. Liu, Y.; Pan, S.; Wang, Y.G.; Xiong, F.; Wang, L.; Chen, Q.; Lee, V.C.S. Anomaly Detection in Dynamic Graphs via Transformer. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 12081–12094. [CrossRef]
49. Lin, Y.; Tang, J.; Zi, C.; Zhao, H.V.; Yao, Y.; Li, J. UniGAD: Unifying Multi-Level Graph Anomaly Detection. *arXiv* **2024**, arXiv:2411.06427.