information

MDPI

Article

DGAM: Dual-Guided Anomaly Mining for Semi-Supervised Graph Anomaly Detection

Xingxuan Li [1], Ting Guo [1,*] and Zhen Tian [2]

[1] School of Computer Science and Technology, North University of China, Taiyuan 030051, China; 2307054216@st.nuc.edu.cn
[2] James Watt School of Engineering, University of Glasgow, Glasgow G12 8QQ, UK
* Correspondence: gting@nuc.edu.cn

Abstract

For the challenging scenario in which only normal node labels are available in semi-supervised graph anomaly detection, existing generative methods usually synthesize abnormal nodes through random perturbation or feature interpolation. However, these methods fail to consider node abnormality comprehensively from both structural and attribute perspectives, resulting in generated pseudo-anomalies of limited quality and insufficient reliability. In order to address this problem, we propose DGAM (dual-guided anomaly mining), a framework for selecting pseudo-anomaly nodes based on the dual-index measurement of topological anomaly and feature consistency. The core of the framework is the joint anomaly evaluation module, which quantifies node anomaly through two computable metrics. The topological boundary score (TBS) measures the boundary of a node's topological position based on the proportion of connections between a node and labeled normal nodes in its K-hop neighborhood. The feature deviation score (FDS) evaluates the consistency of a node's local features by calculating the average cosine similarity between its features and those of its K-hop neighbors. The module selects a fixed set of nodes with higher comprehensive anomaly scores from the labeled normal nodes as pseudo-anomalies, so as to construct a training set containing explicit supervision signals. The model adopts a shared encoder architecture and jointly optimizes the classification loss based on pseudo-labels and the embedding regularization loss of the graph nodes to learn a more discriminative node representation. Experimental results on multiple real-world graph datasets show that DGAM can stably improve anomaly detection performance, effectively verifying the effectiveness of the proposed screening mechanism and joint training strategy.

Keywords: graph neural network; graph anomaly detection; semi-supervised graph anomaly detection

Check for updates

Academic Editor: Luigi Laura

Received: 13 April 2026
Revised: 16 May 2026
Accepted: 20 May 2026
Published: 23 May 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Graph anomaly detection (GAD) has attracted increasing research attention owing to its critical role in real-world applications such as fraud detection, network intrusion detection, and spam filtering [1–3]. However, GAD remains challenging as anomalies are inherently rare, making high-quality anomaly labels difficult and prohibitively expensive to obtain. While unsupervised GAD methods avoid the need for labels entirely, they fail to leverage the readily available normal node labels. Existing semi-supervised methods attempt to incorporate label information, but still require both labeled normal and abnormal

nodes, making them impractical due to the high cost of anomaly annotation. Therefore, semi-supervised graph anomaly detection, which only uses a small number of normal node labels for training, has become a more practical research paradigm [4].

To tackle GAD with only normal supervision, a natural strategy is to augment the training set by synthesizing pseudo-anomaly nodes, thereby providing negative class supervision signals. These methods fall into two categories: interpolation-based methods generate pseudo-anomalies via interpolation between normal nodes [5,6], while perturbation-based methods simulate anomalies by injecting random noise into node features or the adjacency matrix [4,7,8]. Although these methods improve detection performance to some extent, they generally suffer from a critical drawback: the lack of a measurable evaluation of node anomaly degree. As the synthesis is random and undirected, the generated pseudo-anomalies may deviate from the true anomaly distribution, limiting the model's generalization ability.

To address these challenges, this paper proposes a framework for selecting pseudo-anomalies based on the dual-index measurement of topological anomaly and feature consistency. The core idea of the method is to identify nodes from the labeled normal set $\mathcal{V}_l$ that, through the calculation of an abnormality index, exhibit characteristics most similar to those of true anomalies. These nodes are then selected as pseudo-anomalies to participate in training. Specifically, we design two complementary measurable indicators: topological boundary score (TBS) and feature deviation score (FDS). TBS measures the connectivity compactness of a node with labeled normal nodes within its K-hop neighborhood. Since anomalous nodes are usually located at the periphery of normal communities and exhibit sparse connections with labeled normal nodes, a higher TBS indicates that a node resides on the topological boundary and is a potential anomaly. FDS calculates the average cosine similarity between a node and its K-hop neighbors to reflect the feature consistency within a local neighborhood. The features of anomalous nodes often deviate significantly from their neighbors, resulting in high FDS (which signifies low similarity), thus capturing abnormal patterns at the feature level. By using weighted fusion of TBS and FDS, we calculate a comprehensive abnormality score for each labeled normal node and select those with the highest scores as pseudo-anomalies. This allows us to construct a training set containing clear positive and negative supervision signals. The model adopts a shared encoder architecture and jointly optimizes the classification loss based on pseudo-labels and the embedding regularization loss to learn a more discriminative node representation. The main contributions of this paper are summarized as follows:

- We propose DGAM, a semi-supervised graph anomaly detection framework based on topological isolation and feature deviation. It uses TBS and FDS metrics to select pseudo-anomalies from labeled normal nodes as hard negative proxies, providing an effective basis for semi-supervised anomaly detection with only normal labels.
- A dual-index screening strategy is proposed by combining TBS and FDS. TBS identifies structurally isolated nodes, while FDS captures nodes with deviating local features. Their weighted fusion enables more effective pseudo-anomaly selection, contributing to improved anomaly detection performance in experiments.
- A joint training objective is designed to optimize pseudo-label classification while introducing embedding norm regularization to enhance the discriminative power and generalization of node representations.
- Experimental results on multiple real-world graph datasets show that DGAM outperforms existing state-of-the-art baselines, verifying the effectiveness of the proposed method.

2. Related Work

2.1. Graph Anomaly Detection

Graph anomaly detection aims to identify nodes that deviate significantly from the majority in structure or attributes. According to label availability, existing methods fall into two categories: unsupervised and semi-supervised. Unsupervised methods do not rely on labels and typically identify anomalies based on reconstruction error or contrastive learning, including DOMINANT [9], AnomalyDAE [10], OCGNN [11], CoLA [12], ANEMONE [13], AD-GCL [14], CVGAD [15], and AC2L-GAD [16]. However, they completely rely on the data distribution and struggle with complex anomaly patterns. Semi-supervised methods use limited labels to improve detection. Some assume both normal and abnormal labels are available, such as BWGNN [17], while a more challenging setting assumes only normal labels, where GGAD [4] generates pseudo-anomaly nodes to provide negative supervision.

2.2. Generative Graph Anomaly Detection

To address the scarcity of abnormal samples, generative methods augment the training set by synthesizing pseudo-anomaly nodes. Existing synthesis strategies can be mainly divided into two categories: feature interpolation and noise perturbation. Feature interpolation methods generate new samples by linear or nonlinear interpolation between normal node features, such as AuGAN [18]; however, the generated samples are often too smooth to simulate the irregular characteristics of real anomalies. Noise perturbation methods generate pseudo-anomalies by adding random noise to normal nodes, including DAGAD [8] and GGAD [4]; however, the perturbation process is still random, and the generated pseudo-anomalies may deviate from the real anomaly distribution.

3. Preliminaries

3.1. Problem Definition

This work addresses semi-supervised anomaly detection on attributed graphs. We consider an attributed graph $\mathcal{G} = (\mathcal{V}, \mathbf{A}, \mathbf{X})$ with $N = |\mathcal{V}|$ nodes. $\mathbf{A} \in \{0,1\}^{N \times N}$ is the binary adjacency matrix, where $\mathbf{A}_{ij} = 1$ indicates an edge between nodes v_i and v_j. $\mathbf{X} \in \mathbb{R}^{N \times d}$ denotes the node attribute matrix, with $\mathbf{x}_i$ representing the d-dimensional attribute vector of the i-th node.

In our semi-supervised setting, only a small number of nodes are labeled, all of which are known to be normal. Let $\mathcal{V}_l$ denote the set of labeled normal nodes, and $\mathcal{V}_u = \mathcal{V} \setminus \mathcal{V}_l$ denote the unlabeled set. Our goal is to learn a scoring function $g : \mathcal{V} \to \mathbb{R}$ that assigns lower scores to normal nodes than anomalies, thus achieving reliable separation between the two classes.

3.2. Graph Neural Networks

Graph neural networks (GNNs) learn node representations by aggregating neighborhood information and are core tools for processing graph-structured data. Given a graph $\mathcal{G}$ and input features $\mathbf{H}^{(0)} = \mathbf{X}$, the propagation rule of the ℓ-th layer can be generalized as:

$$\mathbf{H}^{(\ell)} = \text{AGGREGATE}\left(\mathbf{A}, \mathbf{H}^{(\ell-1)}, \mathbf{W}^{(\ell)}\right), \tag{1}$$

where $\mathbf{W}^{(\ell)}$ denotes the trainable weight matrix of the ℓ-th layer, and $\mathbf{H}^{(\ell)}$ denotes the output node representations of the ℓ-th layer. After L layers of propagation, we obtain the final node representation $\mathbf{Z} = \mathbf{H}^{(L)} \in \mathbb{R}^{N \times h}$, which is used for subsequent anomaly scoring tasks.

In this paper, we adopt a graph convolutional network (GCN) as the basic encoder. The inter-layer propagation rule of GCN is as follows:

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

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ denotes the adjacency matrix with self-loops, $\tilde{\mathbf{D}}$ is the degree matrix of $\tilde{\mathbf{A}}$, and $\sigma(\cdot)$ is the activation function. GCN aggregates neighborhood information via symmetric normalization, which can effectively capture the structural and attribute features of the graph.

4. Our Method

This section describes the proposed semi-supervised graph anomaly detection method in detail. As shown in Figure 1, the overall framework consists of two core components: the topological and feature anomaly measurement module and the training stage. First, the anomaly measurement module evaluates two complementary metrics, the topological boundary score (TBS) and the feature deviation score (FDS), to select pseudo-anomalies from the labeled normal node set $\mathcal{V}_l$. Then, the training stage jointly optimizes the node classification loss and the embedding regularization constraint on the original graph, enabling the encoder to learn more discriminative node representations.

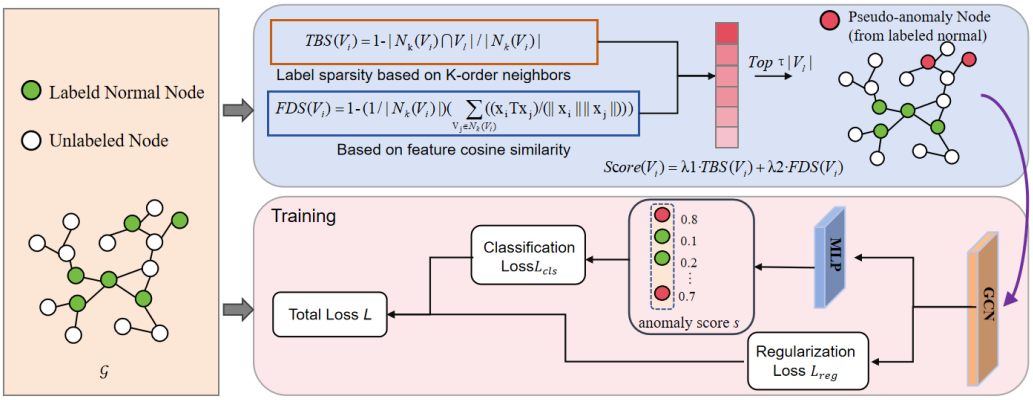


Figure 1. The overall framework of DGAM consists of two core modules: (1) the topological and feature anomaly measurement module, which screens pseudo-anomalies from the labeled normal node set $\mathcal{V}_l$ using TBS and FDS; (2) the training stage, which jointly optimizes classification loss and embedding regularization to learn discriminative node representations.

4.1. Topological and Feature Anomaly Measurement Module

In the semi-supervised scenario where only normal node labels are available, ground-truth abnormal samples cannot be directly obtained. Our motivation stems from the observation that anomalous nodes are typically located at the periphery of normal communities and exhibit sparse topological connections with labeled normal nodes. Furthermore, the features of anomalous nodes often deviate significantly from the local patterns of their neighborhood. Based on these insights, we design two metrics to identify pseudo-anomalies from the labeled normal set $\mathcal{V}_l$.

4.1.1. Topological Boundary Score (TBS)

For any node v_i, define the set of its K-hop neighbors as $\mathcal{N}_K(v_i)$, that is, the set of nodes reachable by a path of length at most K, excluding v_i itself. Self-loops, if present in the raw graph, are removed during neighbor computation. For isolated nodes or nodes with $|\mathcal{N}_K(v_i)| = 0$, we set $\mathrm{TBS}(v_i) = 0$ and $\mathrm{FDS}(v_i) = 0.5$ to avoid undefined operations and to assign neutral anomaly scores to nodes lacking local context. Given that the set of

labeled normal nodes $\mathcal{V}_l$ is known, the topological boundary score of node v_i is defined as follows:

$$\mathrm{TBS}(v_i) = 1 - \frac{|\mathcal{N}_K(v_i) \cap \mathcal{V}_l|}{|\mathcal{N}_K(v_i)|}, \tag{3}$$

where $|\cdot|$ denotes the cardinality of a set. This metric measures the degree to which a node is topologically isolated from labeled normal nodes within its local neighborhood. Anomalous nodes tend to reside at the periphery of or outside normal communities, resulting in a low proportion of labeled normal nodes among their neighbors and thus a high TBS value. Conversely, normal nodes typically cluster within normal communities, surrounded by a dense concentration of labeled normal nodes, yielding a low TBS value. Under uniform random labeling with a moderate labeling rate, TBS primarily reflects topological boundary position rather than label sparsity, because the denominator includes all K-hop neighbors in the training set regardless of their label status.

4.1.2. Feature Deviation Score (FDS)

In addition to topological structure, the deviation of node features from those of neighboring nodes provides another important basis for anomaly identification. Anomalous nodes tend to exhibit feature distributions that diverge significantly from the normal patterns of their local neighborhood, resulting in low feature similarity with their neighbors. To capture this phenomenon, we define the feature deviation score (FDS) by computing the average cosine similarity between a node and its K-hop neighbors, and converting it into a deviation score in $[0, 1]$:

$$\mathrm{FDS}(v_i) = 1 - \frac{\bar{c}(v_i) + 1}{2}, \quad \bar{c}(v_i) = \frac{1}{|\mathcal{N}_K(v_i)|} \sum_{v_j \in \mathcal{N}_K(v_i)} \frac{\mathbf{x}_i^\top \mathbf{x}_j}{\|\mathbf{x}_i\| \, \|\mathbf{x}_j\|}, \tag{4}$$

where $\mathbf{x}_i \in \mathbb{R}^d$ denotes the feature vector of node v_i and $\|\cdot\|$ denotes the L_2 norm. Here, $\bar{c}(v_i) \in [-1, 1]$ is the average cosine similarity between v_i and its K-hop neighbors. The linear mapping $\mathrm{FDS}(v_i) = 1 - (\bar{c}(v_i) + 1)/2$ transforms this similarity into a deviation score in $[0, 1]$: a mean similarity of 1 (identical features) yields FDS $= 0$ (no deviation), while -1 (opposite features) yields FDS $= 1$ (maximal deviation). Prior to computing FDS, all node features are row-normalized (each feature vector is scaled to unit L_2 norm). This preprocessing is applied consistently across all methods. If a target node v_i has $\|\mathbf{x}_i\| = 0$, we set all its cosine similarities to 0, yielding $\mathrm{FDS}(v_i) = 0.5$ (neutral score). If a neighbor v_j has $\|\mathbf{x}_j\| = 0$, we replace its norm with $\epsilon = 10^{-8}$ to avoid division by zero; its dot product is naturally 0, so its contribution to the average similarity is effectively zero. These treatments ensure numerical stability and assign neutral anomaly scores when feature information is absent.

This metric characterizes local anomalies at the feature level: normal nodes tend to exhibit feature consistency with their neighbors, resulting in low FDS values, whereas anomalous nodes deviate substantially from the local feature distribution, yielding high FDS values. This design captures the widely observed local feature inconsistency of graph anomalies. FDS thus complements TBS by capturing anomalous patterns in the feature dimension rather than the topological dimension.

4.1.3. Pseudo-Anomaly Screening

Based on the two metrics defined above, we calculate a comprehensive abnormality score for each labeled normal node $v_i \in \mathcal{V}_l$:

$$\mathrm{Score}(v_i) = \lambda_1 \cdot \mathrm{TBS}(v_i) + \lambda_2 \cdot \mathrm{FDS}(v_i), \tag{5}$$

where λ_1 and λ_2 are hyperparameters that regulate the relative importance of the topological and feature-level metrics. Both TBS and FDS lie in $[0, 1]$, ensuring they contribute on comparable scales in the weighted sum. All labeled normal nodes in $\mathcal{V}_l$ are sorted in descending order according to their comprehensive scores. The top $\tau \cdot |\mathcal{V}_l|$ nodes with the highest scores are selected as pseudo-anomaly nodes, denoted as $\mathcal{V}_{\text{pseudo}}$. The remaining nodes in $\mathcal{V}_l$ are retained as high-confidence normal samples for the subsequent training phase. This screening process is performed once prior to training to establish a fixed set of pseudo-labels, ensuring stable supervision signals for the model.

These selected nodes are not treated as genuine anomalies; they serve as boundary normal proxies that temporarily act as positive samples in the subsequent training. We select pseudo-anomaly candidates exclusively from $\mathcal{V}_l$ rather than from $\mathcal{V}_u$. This is because $\mathcal{V}_l$ is the only set guaranteed to be free of real anomalies, ensuring that no true anomaly is inadvertently assigned a synthetic label during training. Selecting from $\mathcal{V}_u$, by contrast, would risk supervision leakage, as unlabeled anomalies could be misused as training positives. Our strategy thus provides controlled and contamination-free proxy supervision.

4.2. Training

After obtaining the pseudo-anomalies, we jointly utilize them alongside the remaining normal nodes for model training. This section details the adopted graph neural network architecture and the multi-task training objective.

4.2.1. Model Architecture

We adopt the graph convolutional network (GCN) as the base encoder and obtain the node embedding $\mathbf{Z}$ by aggregating neighborhood information through two GCN layers. Subsequently, $\mathbf{Z}$ is fed into a multilayer perceptron (MLP), which serves as the anomaly scoring network, outputting the anomaly score s_i for each node v_i. The MLP outputs a scalar logit $s_i \in \mathbb{R}$. The final anomaly score is defined as $g(v_i) = s_i$. Since AUROC and AUPRC are rank-based metrics, using raw logits is equivalent to using sigmoid probabilities $\sigma(s_i)$ for evaluation.

4.2.2. Training Objective

To simultaneously use the supervision signals of pseudo-anomaly nodes and the structural information of full graph nodes, we design a joint loss function:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{\text{cls}} + \beta \cdot \mathcal{L}_{\text{reg}}, \tag{6}$$

where α and β are trade-off hyperparameters that balance the two objectives.

Classification loss $\mathcal{L}_{\text{cls}}$: A training set is constructed based on the pseudo-anomaly node set $\mathcal{V}_{\text{pseudo}}$ and the remaining labeled normal node set $\mathcal{V}_l \setminus \mathcal{V}_{\text{pseudo}}$, where the labels of normal nodes are 0 and those of pseudo-anomaly nodes are 1. Note that the positive labels assigned to $\mathcal{V}_{\text{pseudo}}$ are synthetic pseudo-labels, not real anomaly labels from the ground truth. These labels are used exclusively in $\mathcal{L}_{\text{cls}}$ and do not override the original normal labels in the ground truth. Binary cross-entropy loss is used as follows:

$$\mathcal{L}_{\text{cls}} = -\frac{1}{|\mathcal{V}_{\text{train}}|} \sum_{v_i \in \mathcal{V}_{\text{train}}} [y_i \log(\sigma(s_i)) + (1 - y_i) \log(1 - \sigma(s_i))], \tag{7}$$

the training set $\mathcal{V}_{\text{train}}$ is composed of all nodes in the labeled set $\mathcal{V}_l$, y_i is the node label, $\sigma(\cdot)$ is the Sigmoid function, and s_i is the anomaly logit output by the model. The binary cross-entropy loss is applied without class weighting or resampling; the positive/negative

balance is controlled solely by the pseudo-anomaly proportion τ, which is tuned on the validation set.

Regularization loss $\mathcal{L}_{\text{reg}}$: To prevent overfitting and enhance the generalization ability of the embeddings, we impose an L_2 norm constraint on the embeddings of all nodes:

$$\mathcal{L}_{\text{reg}} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{z}_i\|_2, \tag{8}$$

where $\mathbf{z}_i$ is the embedding vector of node v_i. This penalty encourages compact embeddings and prevents the model from relying on spuriously large norms to separate pseudo-anomalies from normal nodes.

By jointly optimizing the classification loss and the regularization loss, the model maintains the compactness of the normal node embedding while learning to discriminate abnormal nodes, which improves the overall generalization performance. The complete algorithm is presented in Algorithm 1.

Algorithm 1 DGAM: dual-guided anomaly mining for semi-supervised GAD

Require: Graph $\mathcal{G} = (\mathcal{V}, \mathbf{A}, \mathbf{X})$; Labeled normal node set $\mathcal{V}_l$; Hyperparameters: K, τ, λ_1, λ_2, α, β
Ensure: Anomaly scores $\{s_i\}_{v_i \in \mathcal{V}}$

// Phase 1: Pseudo-Anomaly Screening (pre-training)
for each $v_i \in \mathcal{V}_l$ **do**
  Compute K-hop neighbors $\mathcal{N}_K(v_i)$ (excluding v_i itself)
  if $|\mathcal{N}_K(v_i)| = 0$ **then**
    $\text{TBS}(v_i) \leftarrow 0$, $\text{FDS}(v_i) \leftarrow 0.5$
  else
    $\text{TBS}(v_i) \leftarrow 1 - \frac{|\mathcal{N}_K(v_i) \cap \mathcal{V}_l|}{|\mathcal{N}_K(v_i)|}$
    $\bar{c}(v_i) \leftarrow \frac{1}{|\mathcal{N}_K(v_i)|}\sum_{v_j \in \mathcal{N}_K(v_i)} \frac{\mathbf{x}_i^\top \mathbf{x}_j}{\|\mathbf{x}_i\|\|\mathbf{x}_j\|}$
    $\text{FDS}(v_i) \leftarrow 1 - \frac{\bar{c}(v_i)+1}{2}$
  end if
  $\text{Score}(v_i) \leftarrow \lambda_1 \cdot \text{TBS}(v_i) + \lambda_2 \cdot \text{FDS}(v_i)$
end for
Sort $\mathcal{V}_l$ by Score in descending order
$\mathcal{V}_{\text{pseudo}} \leftarrow$ top $\tau \cdot |\mathcal{V}_l|$ nodes
$\mathcal{V}_{\text{normal}} \leftarrow \mathcal{V}_l \setminus \mathcal{V}_{\text{pseudo}}$
Assign synthetic labels: $y_i = 1$ for $v_i \in \mathcal{V}_{\text{pseudo}}$, $y_i = 0$ for $v_i \in \mathcal{V}_{\text{normal}}$
// Note: positive labels are synthetic pseudo-labels, not ground-truth anomalies
// Phase 2: Model Training
Initialize GCN encoder f_{GCN} and MLP scorer f_{MLP}
for epoch $= 1$ to MaxEpoch **do**
  $\mathbf{Z} \leftarrow f_{\text{GCN}}(\mathbf{A}, \mathbf{X})$ {Node embeddings}
  $s_i \leftarrow f_{\text{MLP}}(\mathbf{z}_i)$ for all $v_i \in \mathcal{V}$
  $\mathcal{L}_{\text{cls}} \leftarrow -\frac{1}{|\mathcal{V}_{\text{train}}|}\sum_{v_i \in \mathcal{V}_{\text{train}}} [y_i \log(\sigma(s_i)) + (1 - y_i)\log(1 - \sigma(s_i))]$
  $\mathcal{L}_{\text{reg}} \leftarrow \frac{1}{|\mathcal{V}|}\sum_{v_i \in \mathcal{V}} \|\mathbf{z}_i\|_2$
  $\mathcal{L} \leftarrow \alpha \cdot \mathcal{L}_{\text{cls}} + \beta \cdot \mathcal{L}_{\text{reg}}$
  Update f_{GCN}, f_{MLP} via Adam optimizer on $\mathcal{L}$
  Evaluate on validation set; save best model if improved
end for
// Phase 3: Inference
Load best model checkpoint
$\mathbf{Z} \leftarrow f_{\text{GCN}}(\mathbf{A}, \mathbf{X})$
$s_i \leftarrow f_{\text{MLP}}(\mathbf{z}_i)$ for all $v_i \in \mathcal{V}$
return $\{s_i\}_{v_i \in \mathcal{V}}$

5. Experiments

5.1. Experimental Setup

We used four public real-world graph anomaly detection datasets in our experiments, as summarized in Table 1. Detailed descriptions of each dataset are provided below:

- **Amazon** [19]: User-product review network (Musical Instruments). Anomaly: users with <20% helpful votes. Undirected, unweighted. 25 handcrafted features (user activity, review helpfulness).
- **Reddit** [20]: User-subreddit interaction graph. Anomaly: banned users (spam/harassment). Undirected, unweighted. 64 features (text embeddings from user posts).
- **Elliptic** [21]: Bitcoin transaction network. Anomaly: illicit transactions. Directed weighted, converted to undirected unweighted. 93 transaction metadata features.
- **Photo** [22]: Amazon co-purchase network (Photo category). Anomaly: products with abnormal purchasing patterns. Undirected, unweighted. 745 bag-of-words features from product reviews.

Table 1. Key statistics of the datasets used.

Dataset	Type	Nodes	Edges	Attributes	Anomalies (%)
Amazon	Co-review	11,944	4,398,392	25	6.9
Reddit	Social Media	10,984	168,016	64	3.3
Elliptic	Bitcoin Transaction	46,564	73,248	93	9.8
Photo	Co-purchase	7535	119,043	745	9.2

All datasets are used in their standard public benchmark form without additional conversion. The only preprocessing step is row-normalization of node features before FDS computation and model input.

In the experiments, we randomly sample 50% of the normal nodes from the training set as labeled normal nodes and use the remaining nodes as unlabeled data. The training, validation, and test sets are partitioned at a ratio of 0.3/0.1/0.6 (uniform random sampling without stratification).

We adopt a transductive semi-supervised setting. The full graph (including all training, validation, and test nodes) is used during message passing, with features and edges of all nodes available during training. Only the labeled normal nodes in the training set are used to construct the training objective. Pseudo-anomalies are selected exclusively from these labeled normal nodes without accessing any anomaly labels. Validation set anomaly labels are used only for model selection (early stopping and hyperparameter tuning), following the standard protocol adopted by prior work and baselines. Test set labels are used only for final evaluation.

The following representative graph anomaly detection methods are selected as baselines, each with its method type and relevance briefly described:

- DOMINANT [9]: Autoencoder-based method reconstructing both node features and graph structure; anomaly score combines reconstruction errors.
- AnomalyDAE [10]: Dual autoencoder with attention mechanism, jointly learning structural and attribute embeddings.
- OCGNN [11]: One-class graph neural network integrating SVM objective with GNN representation learning.
- AEGIS [23]: Generative adversarial network that synthesizes pseudo-anomalies to train a discriminative detector.
- GAAN [24]: GAN-based method generating fake graph nodes; discriminator distinguishes real vs. fake node pairs.

- TAM [25]: Local affinity maximization with heterophily edge pruning; anomaly score based on affinity deficiency.
- CHRN [26]: Heterophily-aware method bridging spatial and spectral domains; uses label-aware edge indicator to prune heterophilous edges.
- GGAD [4]: Generative semi-supervised method, closest to our setting, generating pseudo-anomaly nodes for one-class classification.
- TAQ-GAD [27]: Generative semi-supervised method that selects pseudo-anomalies via topological anomaly quantification and augments training with virtual anomaly centers.

To adapt unsupervised baselines to our normal-label semi-supervised setting, we apply the following standard modifications: reconstruction-based methods (DOMINANT, AnomalyDAE) are trained exclusively on labeled normal nodes; OCGNN optimizes its one-class center only on normal instances; TAM maximizes affinity only on normal samples; GAN-based methods (AEGIS, GAAN) use labeled normal nodes as real samples and generate pseudo-anomalies as negatives; CHRN removes anomaly labels and uses only post-aggregation scores.

Model performance was evaluated using two widely used metrics: AUROC measures the overall ability of the model to distinguish between normal and abnormal nodes on a scale of [0, 1], where higher is better. AUPRC summarizes the precision-recall trade-off for the anomalous (positive) class, and is especially informative under class imbalance. Its value range is [0, 1], where higher is better.

5.2. Implementation Details

DGAM is implemented with PyTorch 1.10.0 and Python 3.8. The encoder is a two-layer GCN with hidden dimension 300 and PReLU activation. Dropout (rate 0.3) is applied after each GCN layer. The anomaly scorer is a two-layer MLP ($300 \rightarrow 128 \rightarrow 1$) with PReLU in the hidden layer. Full-graph training is performed with the Adam optimizer. The learning rate and number of epochs are dataset-specific: Amazon (lr = 0.0001, epochs = 550), Reddit (lr = 0.001, epochs = 100), Elliptic (lr = 0.0001, epochs = 200), and Photo (lr = 0.01, epochs = 300). No weight decay is applied.

In the topological anomaly measurement module, the node neighborhood is defined as the K-hop neighbor, and $K = 2$ is set uniformly for all datasets. The model adopts a consistent embedding dimension of 300. We sort all the labeled normal nodes by their comprehensive anomaly scores, and select the top τ proportion as the pseudo-anomaly nodes. The fusion weights λ_1 and λ_2 of TBS and FDS are both set to 1. In the training loss, the classification loss weight $\alpha = 0.5$ and the regularization loss weight $\beta = 0.5$.

5.3. Main Experiment Results

We compared DGAM with the baselines, and the results are shown in Table 2. Below, we discuss dataset-specific performance patterns.

Amazon. On Amazon, TAQ-GAD achieves the best AUROC (0.9474) and AUPRC (0.7973), followed by GGAD (0.9443 and 0.7922). DGAM obtains 0.9215 ± 0.0512 and 0.7762 ± 0.0702, respectively. DGAM is slightly lower than the top baselines on this dataset. Amazon has a very high edge density (over four million edges) and a moderate anomaly ratio (6.9%); generative methods like GGAD and TAQ-GAD may better cover the anomaly distribution in this setting by synthesizing diverse pseudo-anomalies. DGAM selects pseudo-anomalies exclusively from labeled normal nodes, which may be less diverse when normal nodes do not fully capture all anomaly patterns.

Elliptic. On Elliptic, DGAM achieves the best AUPRC of 0.4124 ± 0.0061, a substantial improvement over the best baseline (GGAD, 0.2425). The AUROC (0.7463 ± 0.0087) is competitive, ranking among the top methods alongside TAQ-GAD (0.7453) and CHRN

(0.7315). This indicates that DGAM is particularly effective at ranking anomalies higher on this dataset, where illicit transactions tend to be structurally peripheral and feature-distinctive, making them well-suited for detection by the dual TBS and FDS metrics. The notably small standard deviation (0.0061 for AUPRC) further demonstrates DGAM's stable performance on this dataset.

Reddit and Photo. On Reddit, TAQ-GAD achieves the best AUROC (0.6682) and AUPRC (0.0780), while DGAM obtains 0.6599 $\pm$ 0.0396 and 0.0590 $\pm$ 0.0123, respectively, remaining competitive with the top methods. On Photo, DGAM gives the best AUROC (0.7175 $\pm$ 0.0772) and a competitive AUPRC (0.2440 $\pm$ 0.1197). These two datasets are sparser and exhibit more complex anomaly patterns. In such settings, DGAM's dual metrics (TBS and FDS) provide effective signals for selecting pseudo-anomalies, contributing to its relative advantage. The moderate standard deviations on Photo reflect sensitivity to the random sampling of labeled normal nodes on this smaller dataset.

Overall, DGAM shows competitive performance across the four datasets. The dataset-specific results suggest that DGAM works particularly well on sparse graphs or when anomaly patterns are diverse (e.g., Elliptic, Photo), while generative baselines may remain competitive on denser graphs (e.g., Amazon). These results confirm that the pseudo-anomaly nodes jointly screened by TBS and FDS provide effective supervision signals for training.

Table 2. Comparison of AUROC and AUPRC for each method on the datasets. The best results are highlighted in **bold**. DGAM results are averaged over 3 runs (mean $\pm$ std).

Setting	Method	Dataset							
		AUROC				AUPRC			
		Amazon	Reddit	Elliptic	Photo	Amazon	Reddit	Elliptic	Photo
Unsupervised	DOMINANT	0.7025	0.5105	0.2960	0.5136	0.1315	0.0380	0.0454	0.1039
	AnomalyDAE	0.7783	0.5091	0.4963	0.5069	0.1429	0.0319	0.0872	0.0987
	OCGNN	0.7165	0.5246	0.2581	0.5307	0.1352	0.0375	0.0616	0.0965
	AEGIS	0.6059	0.5349	0.4553	0.5516	0.1200	0.0413	0.0827	0.0972
	GAAN	0.6513	0.5216	0.2590	0.4296	0.0852	0.0348	0.0436	0.0768
	TAM	0.8303	0.6062	0.4039	0.5675	0.4024	0.0437	0.0502	0.1013
Semi-supervised	DOMINANT	0.8867	0.5194	0.3256	0.5314	0.7289	0.0414	0.0652	0.1283
	AnomalyDAE	0.9171	0.5280	0.5409	0.5272	0.7748	0.0362	0.0949	0.1177
	OCGNN	0.8810	0.5622	0.2881	0.6461	0.7538	0.0400	0.0640	0.1501
	AEGIS	0.7593	0.5605	0.5132	0.5936	0.2616	0.0441	0.0912	0.1110
	GAAN	0.6531	0.5349	0.2724	0.4355	0.0856	0.0362	0.0611	0.0768
	TAM	0.8405	0.5829	0.4150	0.6013	0.5183	0.0446	0.0552	0.1087
	CHRN	0.9346	0.5731	0.7315	0.6223	0.7865	0.0500	0.2101	0.1420
	GGAD	0.9443	0.6354	0.7290	0.6476	0.7922	0.0610	0.2425	0.1420
	TAQ-GAD	**0.9474**	**0.6682**	0.7453	0.7107	**0.7973**	**0.0780**	0.3573	0.2073
	DGAM	0.9215 $\pm$ 0.0512	0.6599 $\pm$ 0.0396	**0.7463** $\pm$ **0.0087**	**0.7175** $\pm$ **0.0772**	0.7762 $\pm$ 0.0702	0.0590 $\pm$ 0.0123	**0.4124** $\pm$ **0.0061**	**0.2440** $\pm$ **0.1197**

5.4. Ablation Study

To verify the effectiveness of each module in DGAM, we design multiple variants for the ablation study. Ablation experiments were conducted with a fixed random seed.

First, we examine the impact of the topological and feature anomaly measurement module. The baseline only uses the original labeled normal nodes for embedding regularization training, without introducing classification signals. After adding TBS or FDS screening alone, the model obtains pseudo-anomaly supervision, and the performance is improved on most datasets. As shown in Table 3, on Reddit, +TBS achieves a higher AUROC than +FDS, but +FDS yields a slightly higher AUPRC. This reflects that TBS and FDS capture complementary information from topological boundary and feature deviation perspectives.

When the two metrics are combined (+TBS+FDS), the performance is generally robust. On Reddit and Photo, the combination achieves the best AUROC and AUPRC. On Elliptic, while +FDS alone yields a marginally higher AUPRC (0.4132) than +TBS+FDS (0.4056), the combined variant remains highly competitive on both metrics (AUROC 0.7469, AUPRC 0.4056). This suggests that topological and feature signals may contribute to different aspects of detection, and the fusion provides a balanced solution rather than over-optimizing for a single metric. Across all datasets, the combination consistently achieves strong performance, confirming the effectiveness of the dual-index screening.

Second, we investigate the role of the regularization loss $\mathcal{L}_{\text{reg}}$. Table 4 compares $\mathcal{L}_{\text{cls}}$ with the full model $\mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{reg}}$. Adding $\mathcal{L}_{\text{reg}}$ further improves AUPRC on Reddit and Elliptic, with Elliptic showing an increase from 0.3591 to 0.4056. On Photo, the full model achieves comparable AUROC and better AUPRC. This demonstrates that the regularization loss helps the model learn a more compact embedding and improves anomaly detection performance.

Overall, the ablation experiments indicate that the joint screening of TBS and FDS, together with the regularization loss, generally contributes to the model's performance. The results verify the effectiveness of the proposed method and the rationality of the design.

Table 3. Ablation study of score components (TBS and FDS). The best results for each metric and dataset are highlighted in **bold**.

Variant	AUROC			AUPRC		
	Reddit	**Elliptic**	**Photo**	**Reddit**	**Elliptic**	**Photo**
Baseline	0.5913	0.3879	0.6427	0.0478	0.0731	0.1360
+TBS	0.6617	**0.7475**	0.7226	0.0576	0.4017	0.3169
+FDS	0.5865	0.7467	0.6504	0.0580	**0.4132**	0.1397
+TBS+FDS	**0.6814**	0.7469	**0.7718**	**0.0713**	0.4056	**0.3647**

Table 4. Ablation study of the regularization loss $\mathcal{L}_{\text{reg}}$. The best results for each metric and dataset are highlighted in **bold**.

Variant	AUROC			AUPRC		
	Reddit	**Elliptic**	**Photo**	**Reddit**	**Elliptic**	**Photo**
$\mathcal{L}_{\text{cls}}$	0.6804	0.7337	**0.7805**	0.0661	0.3591	0.3335
$\mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{reg}}$	**0.6814**	**0.7469**	0.7718	**0.0713**	**0.4056**	**0.3647**

5.5. Computational Complexity and Runtime

The time complexity of the TBS and FDS modules is as follows. TBS requires $O(|V_l| \cdot |N_k(v_i)|)$ operations, where $|V_l|$ is the number of labeled normal nodes (typically much smaller than the total number of nodes, e.g., 15% in our experiments) and $|N_k(v_i)|$ is the size of the k-hop neighborhood (with $k = 2$). FDS additionally computes cosine similarity between node features and each neighbor, leading to $O(|V_l| \cdot |N_k(v_i)| \cdot F)$, where F is the feature dimension. In practice, $|N_k(v_i)|$ remains bounded for sparse graphs, making the preprocessing efficient.

The subsequent GCN training has a complexity similar to that of baseline GNNs, i.e., $O(L \cdot (|E| \cdot H + n \cdot H^2))$, where L is the number of layers, $|E|$ is the number of edges, H is the hidden dimension, and n is the number of nodes. Table 5 reports actual CPU training times. DGAM achieves the lowest runtime on Amazon and Reddit, competitive performance on Photo, and on Elliptic, it is faster than most baselines (except DOMINANT and GAAN). Overall, DGAM offers a good balance between accuracy and efficiency.

Table 5. Runtimes (in seconds) of different methods on four datasets (CPU). The fastest runtime per dataset is highlighted in **bold**.

Method	Amazon	Reddit	Elliptic	Photo
DOMINANT	1592	125	**1119**	437
AnomalyDAE	1656	161	8296	445
OCGNN	765	162	3517	125
AEGIS	1121	166	5638	417
GAAN	1678	94	1866	307
TAM	4516	432	13,200	165
GGAD	658	368	5146	**106**
TAQ-GAD	534	134	2400	480
DGAM (Ours)	**523**	**76**	2160	122

5.6. Parameter Analysis

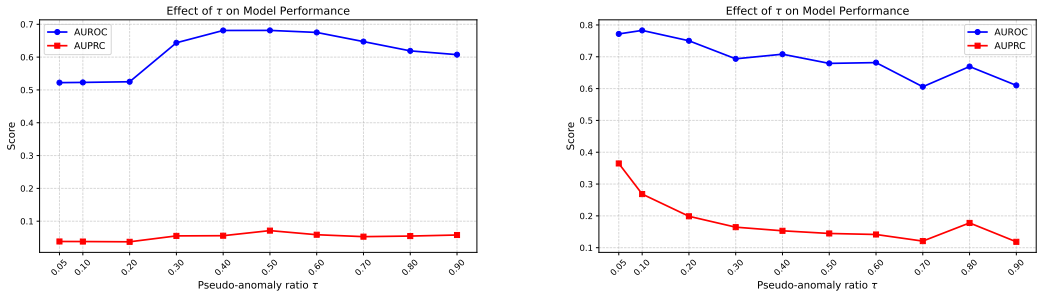
All parameter analysis experiments were performed using a single representative run with a fixed random seed to investigate sensitivity trends.

5.6.1. Effect of the Proportion of Pseudo-Anomalies τ

τ controls the proportion of pseudo-anomalies screened from labeled normal nodes, and is a key hyperparameter to balance the quantity and quality of pseudo-anomalies. Figure 2 shows the AUROC and AUPRC curves on the Photo and Reddit datasets when τ varies from 0.05 to 0.9. The exact numerical values for each τ are reported in Table 6.

Table 6. AUROC and AUPRC values at each evaluated τ on Reddit and Photo. The best value for each metric and dataset is highlighted in **bold**.

τ	Reddit		Photo	
	AUROC	AUPRC	AUROC	AUPRC
0.05	0.5222	0.0382	0.7718	**0.3647**
0.10	0.5229	0.0380	**0.7829**	0.2685
0.20	0.5249	0.0372	0.7503	0.1987
0.30	0.6437	0.0552	0.6936	0.1645
0.40	0.6812	0.0558	0.7081	0.1531
0.50	**0.6814**	**0.0713**	0.6792	0.1447
0.60	0.6752	0.0588	0.6817	0.1415
0.70	0.6473	0.0529	0.6058	0.1207
0.80	0.6190	0.0547	0.6693	0.1778
0.90	0.6075	0.0579	0.6101	0.1185



(**a**) Reddit (**b**) Photo

Figure 2. Performance of DGAM on Reddit and Photo datasets under different proportions τ of pseudo-anomalies selected from labeled normal nodes. τ ranges from 0.05 to 0.9. The left and right subfigures show results on Reddit and Photo, respectively. Higher AUROC and AUPRC values indicate better detection performance.

On the Photo dataset, the AUPRC reaches its maximum at $\tau = 0.05$ (0.3647), while the AUROC achieves its highest value at $\tau = 0.10$ (0.7829), with a comparable value of 0.7718 at $\tau = 0.05$. Both metrics show a general downward trend as τ increases beyond 0.10, with a sharper drop at $\tau = 0.9$ (AUROC 0.6101, AUPRC 0.1185), indicating that the Photo dataset is sensitive to noise, and a high proportion of pseudo-anomalies introduces low-quality samples.

On the Reddit dataset, the performance steadily improves as τ increases, reaches the optimal value at $\tau = 0.5$ (AUROC 0.6814, AUPRC 0.0713), and then slightly decreases, suggesting that this dataset requires a relatively broad coverage of pseudo-anomalies to fully capture diverse anomaly patterns, while too few pseudo-anomalies fail to provide sufficient supervision signals.

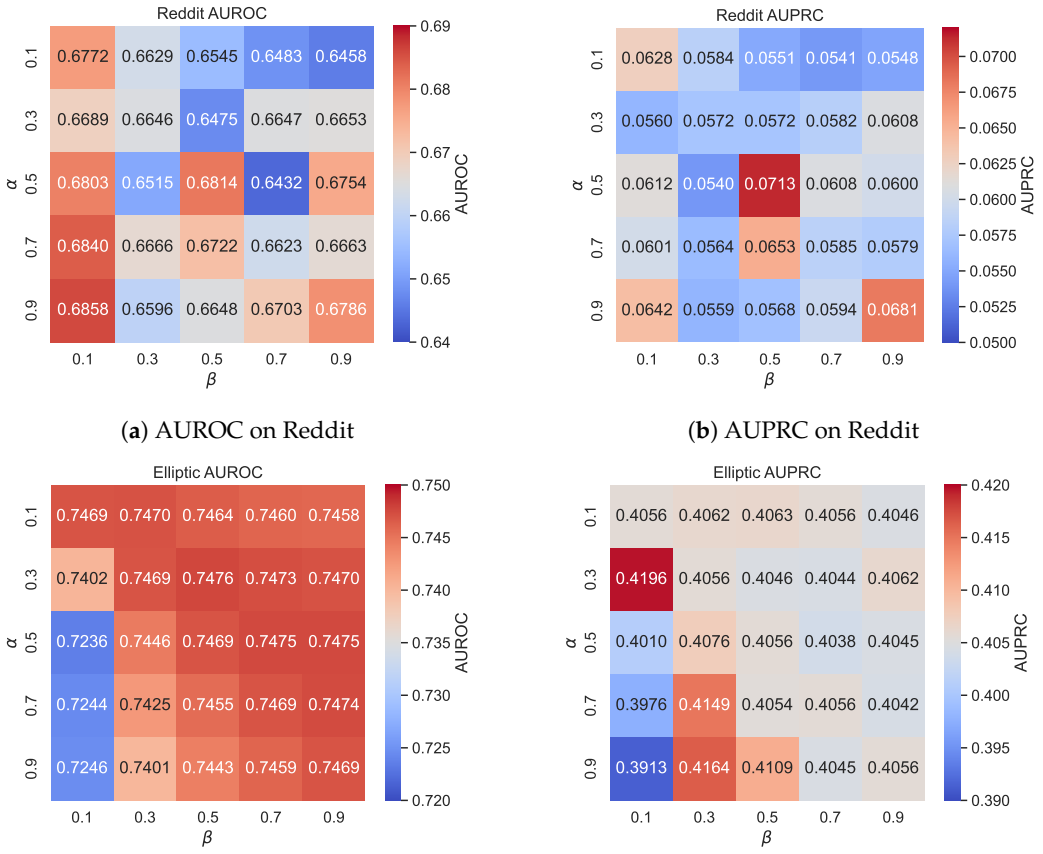
The above differences show that the optimal τ is closely related to the dataset characteristics. Datasets sensitive to noise (such as Photo, Elliptic, and Amazon) should use a small τ (0.05) to ensure pseudo-anomaly quality, while datasets with diverse anomaly patterns (such as Reddit) need a larger τ (0.5) to cover more potential anomalies. The optimal τ is selected based on the validation set AUROC/AUPRC.

5.6.2. Effect of Loss Weights α and β

α and β balance the contribution of classification loss $\mathcal{L}_{\text{cls}}$ and regularization loss $\mathcal{L}_{\text{reg}}$, respectively. To evaluate the sensitivity of the model to the weight selection, we search on the grid of $\alpha, \beta \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, and the results on two representative datasets, Reddit and Elliptic, are shown in Figure 3. On the Reddit dataset, the AUROC is stable between 0.645 and 0.686, and the AUPRC fluctuates between 0.054 and 0.072; even under extreme weight combinations, the performance does not show a significant decrease, indicating that the model is highly robust to the choice of loss weights on this dataset. On the Elliptic dataset, the AUROC is maintained in the range of 0.724–0.748 with a small fluctuation range, while the AUPRC varies between 0.391 and 0.420, showing slightly higher sensitivity to weight combinations than on Reddit; nevertheless, most weight combinations achieve satisfactory performance, and the fluctuations remain within an acceptable range. Based on these observations, the proposed method is insensitive to the choice of α and β, maintaining stable anomaly detection performance over a wide range of weight values.

5.6.3. Effect of Fusion Weights λ_1 and λ_2

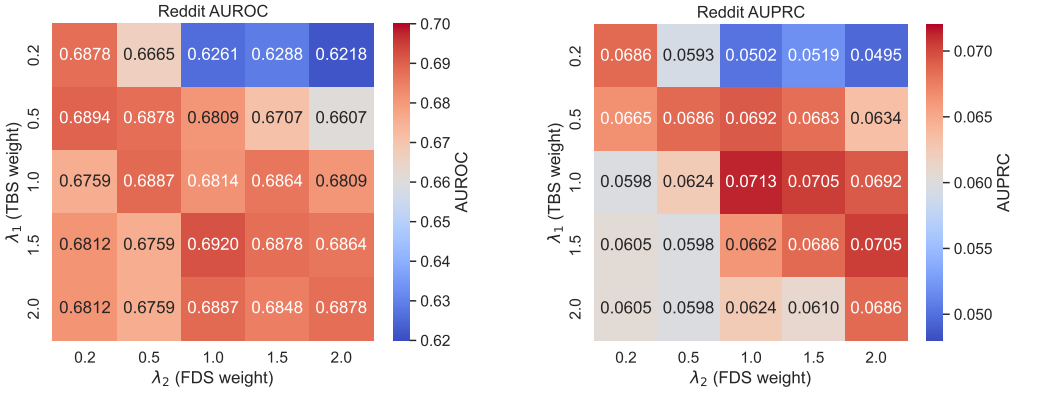
λ_1 and λ_2 balance the contribution of TBS and FDS, respectively. Figure 4 presents the AUROC and AUPRC on the Reddit and Photo datasets for $\lambda_1, \lambda_2 \in \{0.2, 0.5, 1.0, 1.5, 2.0\}$. On Reddit, for AUROC, the highest value (0.6920) is observed at $(\lambda_1 = 1.5, \lambda_2 = 1.0)$. Performance remains relatively stable when $\lambda_1 \geq 0.5$ and $\lambda_2 \leq 1.0$, whereas a clear drop occurs when λ_2 becomes too large (e.g., $\lambda_2 = 2.0$). For AUPRC, the maximum (0.0713) is achieved at the equal-weight setting $(\lambda_1 = 1.0, \lambda_2 = 1.0)$; a comparable value (0.0692) is also obtained at $(\lambda_1 = 0.5, \lambda_2 = 1.0)$. On Photo, the AUROC reaches its peak at $(\lambda_1 = 1.5, \lambda_2 = 2.0)$ with a value of 0.7751, and the equal-weight setting gives 0.7718. The AUPRC on Photo attains its best at $(\lambda_1 = 0.5, \lambda_2 = 0.2)$ with 0.3936, while the default $(\lambda_1 = 1.0, \lambda_2 = 1.0)$ yields 0.3647. Performance on Photo varies more noticeably with different weight combinations, indicating higher sensitivity to the choice of λ_1 and λ_2 compared to Reddit. Overall, the model's performance does not fluctuate drastically within a moderate range of λ_1 and λ_2 on both datasets. The default configuration $(\lambda_1 = \lambda_2 = 1.0)$ provides a balanced trade-off between AUROC and AUPRC, and is therefore adopted for all experiments.



(**a**) AUROC on Reddit (**b**) AUPRC on Reddit

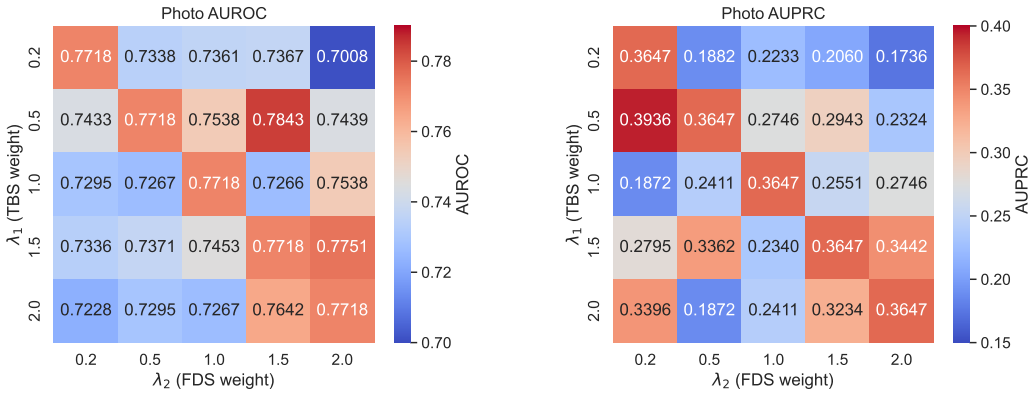
(**c**) AUROC on Elliptic (**d**) AUPRC on Elliptic

Figure 3. Parameter sensitivity analysis of loss weights α and β on the Elliptic and Reddit datasets. Each heatmap cell shows the average AUROC or AUPRC value for a given (α, β) pair, with $\alpha, \beta \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. Higher values indicate better performance.



(**a**) AUROC on Reddit. (**b**) AUPRC on Reddit.

Figure 4. *Cont*.



(**c**) AUROC on Photo. (**d**) AUPRC on Photo.

Figure 4. Parameter sensitivity of the TBS weight λ_1 and FDS weight λ_2 on the Photo and Reddit datasets. $\lambda_1, \lambda_2 \in \{0.2, 0.5, 1.0, 1.5, 2.0\}$. Each cell reports the AUROC or AUPRC value. Higher values indicate better performance.

6. Limitations

Despite its competitive performance, DGAM has several limitations that warrant discussion.

First, the topological assumption underlying TBS—that anomalous nodes reside at the periphery of normal communities—may not hold in all scenarios. When anomalies are deeply embedded within normal communities or when normal bridge nodes lie on community boundaries, TBS may produce misleadingly high or low scores. In such cases, the quality of selected pseudo-anomalies may degrade, and the downstream detection performance may be affected.

Second, the feature deviation score FDS relies on the premise that anomalous nodes exhibit feature patterns distinct from their neighbors. When anomalies share similar features with their local context—e.g., in sophisticated camouflage attacks—FDS may fail to capture the deviation, reducing the effectiveness of the dual-index screening.

Third, the current method requires validation set anomaly labels for selecting the pseudo-anomaly proportion τ and other hyperparameters. While this protocol is standard in the field, developing a fully label-free validation criterion remains an open challenge for practical deployment in scenarios where no anomaly labels are available.

Fourth, DGAM selects pseudo-anomalies exclusively from the labeled normal set V_l, which limits the diversity of the selected candidates. On datasets with dense graph structures and diverse anomaly patterns (e.g., Amazon), generative baselines that synthesize new samples may have an advantage in covering the anomaly distribution more broadly.

Finally, the current study is limited to static attributed graphs. Extending the framework to dynamic or temporal graphs, where anomaly patterns evolve over time, is an important direction for future work.

These limitations point to several directions for future improvement, including community-aware normalization of TBS, adaptive fusion of topological and feature signals, and label-free hyperparameter selection strategies.

7. Conclusions

In this paper, we propose DGAM, a framework for selecting pseudo-anomaly nodes for semi-supervised graph anomaly detection, which identifies pseudo-abnormal nodes from labeled normal nodes by using both topological and feature metrics. Compared with generative approaches that synthesize anomalies via random perturbation or interpolation, the pseudo-anomalies selected by DGAM are guided by topological and feature signals,

yielding candidates that more closely resemble real anomalies and enabling more effective anomaly detection. Specifically, DGAM designs the topological boundary score (TBS) and feature deviation score (FDS) to characterize node anomaly from the two dimensions of topological boundary and local feature consistency, respectively, and selects the nodes with the highest comprehensive anomaly scores as pseudo-anomaly nodes through weighted fusion. On this basis, a joint training strategy is used to jointly optimize the node classification loss and embedding regularization loss to further enhance the discrimination and generalization ability of the representation.

Experimental results on four real-world graph datasets demonstrate that DGAM achieves competitive performance across all datasets, with the best overall results on Photo and Elliptic, competitive performance on Reddit, and results slightly below the top baselines on Amazon. Ablation studies verify the complementarity of TBS and FDS, as well as the effectiveness of the regularization loss. Parameter analysis indicates that DGAM shows promising sensitivity behavior across the tested hyperparameter configurations. Future work will explore an adaptive strategy for τ selection without validation labels—e.g., by detecting the elbow point in the sorted composite anomaly score curve—and extend the proposed framework to dynamic graph scenarios.

Author Contributions: Conceptualization, X.L.; methodology, X.L. and Z.T.; formal analysis, X.L.; data curation, X.L.; writing—original draft preparation, X.L.; writing—review and editing, T.G. and Z.T.; supervision, T.G.; project administration, T.G.; funding acquisition, T.G. All authors have read and agreed to the published version of the manuscript.

Funding: The Natural Science Foundation of Shanxi Province, China (Grant No. 202403021222153); the Shanxi Provincial Key Laboratory (CICIP2024002); the Shanxi Higher Education Institutions Science and Technology Innovation Project (2024L164).

Data Availability Statement: Publicly available datasets were analyzed in this study. The datasets (Amazon, Reddit, Elliptic, and Photo) are openly available in public repositories. The Amazon and Photo datasets can be found at https://github.com/shchur/gnn-benchmark (accessed on 19 May 2026). The Reddit dataset can be found at http://snap.stanford.edu/graphsage/ (accessed on 19 May 2026). The Elliptic dataset can be found at https://www.kaggle.com/datasets/ellipticco/elliptic-data-set (accessed on 19 May 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ma, X.; Wu, J.; Xue, S.; Yang, J.; Zhou, C.; Sheng, Q.Z.; Xiong, H.; Akoglu, L. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 12012–12038. [CrossRef]
2. Tang, J.; Hua, F.; Gao, Z.; Zhao, P.; Li, J. Gadbench: Revisiting and benchmarking supervised graph anomaly detection. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 29628–29653.
3. Chunawala, H.; Kumbhar, S.; Pandey, A.; Rajput, B.J.; Sahu, G.; Guru, A. A Survey of Anomaly Detection in Graphs: Algorithms and Applications. In *Graph Mining: Practical Uses and Instruments for Exploring Complex Networks*; Springer Nature: Cham, Switzerland, 2025; pp. 21–31.
4. Qiao, H.; Wen, Q.; Li, X.; Lim, E.; Pang, G. Generative semi-supervised graph anomaly detection. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 4660–4688.
5. Fernández, A.; Garcia, S.; Herrera, F.; Chawla, N.V. SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *J. Artif. Intell. Res.* **2018**, *61*, 863–905. [CrossRef]
6. Zhou, Q.; Chen, Y.; Xu, Z.; Wu, Y.; Pan, M.; Das, M.; Yang, H.; Tong, H. Graph anomaly detection with adaptive node mixup. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*; Association for Computing Machinery: New York, NY, USA , 2024; pp. 3494–3504.
7. Chen, N.; Liu, Z.; Hooi, B.; He, B.; Fathony, R.; Hu, J.; Chen, J. Consistency training with learnable data augmentation for graph anomaly detection with limited supervision. In Proceedings of the Twelfth International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024.

8. Liu, F.; Ma, X.; Wu, J.; Yang, J.; Xue, S.; Beheshti, A.; Zhou, C.; Peng, H.; Sheng, Q.Z.; Aggarwal, C.C. Dagad: Data augmentation for graph anomaly detection. In *2022 IEEE International Conference on Data Mining (ICDM)*; IEEE: Piscataway, NJ, USA, 2022; pp. 259–268.
9. 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.
10. Fan, H.; Zhang, F.; Li, Z. Anomalydae: Dual autoencoder for anomaly detection on attributed networks. In *ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: Piscataway, NJ, USA, 2020; pp. 5685–5689.
11. Wang, X.; Jin, B.; Du, Y.; Cui, P.; Tan, Y.; Yang, Y. One-class graph neural networks for anomaly detection in attributed networks. *Neural Comput. Appl.* **2021**, *33*, 12073–12085. [CrossRef]
12. 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.* **2021**, *33*, 2378–2392. [CrossRef] [PubMed]
13. Jin, M.; Liu, Y.; Zheng, Y.; Chi, L.; Li, Y.; 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.
14. Xu, Y.; Peng, Z.; Shi, B.; Hua, X.; Dong, B.; Wang, S.; Chen, C. Revisiting graph contrastive learning on anomaly detection: A structural imbalance perspective. *Proc. Aaai Conf. Artif. Intell.* **2025**, *39*, 12972–12980. [CrossRef]
15. Jin, D.; Cao, J.; Wang, X.; Feng, B.; He, D.; Wang, L.; Dang, J. Rethinking Contrastive Learning in Graph Anomaly Detection: A Clean-View Perspective. *arXiv* **2025**, arXiv:2505.18002. [CrossRef]
16. Berahmand, K.; Forouzandeh, S.; Mohammadi, M.; Moradi, P.; Jalili, M. AC2L-GAD: Active Counterfactual Contrastive Learning for Graph Anomaly Detection. *arXiv* **2026**, arXiv:2601.21171.
17. Tang, J.; Li, J.; Gao, Z.; Li, J. Rethinking graph neural networks for anomaly detection. In Proceedings of the International Conference on Machine Learning, PMLR, Baltimore, MD, USA, 17–23 July 2022; pp. 21076–21089.
18. Zhou, S.; Huang, X.; Liu, N.; Zhou, H.; Chung, F.; Huang, L. Improving generalizability of graph anomaly detection models via data augmentation. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 12721–12735. [CrossRef]
19. 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.
20. Kumar, S.; Zhang, X.; Leskovec, J. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 1269–1278.
21. Weber, M.; Domeniconi, G.; Chen, J.; Weidele, D.K.I.; Bellei, C.; Robinson, T.; Leiserson, C. Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks for Financial Forensics. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Anchorage, Alaska, 4–8 August 2019.
22. McAuley, J.; Targett, C.; Shi, Q.; Hengel, A.V.D. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*; Association for Computing Machinery: New York, NY, USA, 2015; pp. 43–52.
23. Ding, K.; Li, J.; Agarwal, N.; Liu, H. Inductive anomaly detection on attributed networks. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, Yokohama Yokohama, Japan, 7–15 January 2021; pp. 1288–1294.
24. Chen, Z.; Liu, B.; Wang, M.; Dai, P.; Lv, J.; Bo, L. Generative adversarial attributed network anomaly detection. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*; Association for Computing Machinery: New York, NY, USA, 2020; pp. 1989–1992.
25. Qiao, H.; Pang, G. Truncated affinity maximization: One-class homophily modeling for graph anomaly detection. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 49490–49512.
26. Gao, Y.; Wang, X.; He, X.; Liu, Z.; Feng, H.; Zhang, Y. Addressing heterophily in graph anomaly detection: A perspective of graph spectrum. In *Proceedings of the ACM Web Conference*; Association for Computing Machinery: New York, NY, USA, 2023; pp. 1528–1538.
27. Guo, T.; Fan, Y.; Cui, C.; Liang, J.; Zhao, J.; Wang, D. Topological Anomaly Quantification for Semi-supervised Graph Anomaly Detection. In Proceedings of the Fourteenth International Conference on Learning Representations (ICLR), Rio de Janeiro, Brazil, 23–27 April 2026.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.