Data Science and Engineering
https://doi.org/10.1007/s41019-024-00249-8

RESEARCH PAPER

Decoupling Anomaly Discrimination and Representation Learning: Self-supervised Learning for Anomaly Detection on Attributed Graph

YanMing Hu[1] · Chuan Chen[1] · BoWen Deng[3] · YuJing Lai[1] · Hao Lin[4] · ZiBin Zheng[2] · Jing Bian[1]

Received: 26 September 2023 / Revised: 14 February 2024 / Accepted: 28 March 2024

Abstract
Anomaly detection on attributed graphs is a crucial topic for practical applications. Existing methods suffer from semantic mixture and imbalance issue because they commonly optimize the model based on the loss function for anomaly discrimination, mainly focusing on anomaly discrimination and ignoring representation learning. Graph Neural networks based techniques usually tend to map adjacent nodes into close semantic space. However, anomalous nodes commonly connect with numerous normal nodes directly, conflicting with the assortativity assumption. Additionally, there are far fewer anomalous nodes than normal nodes, leading to the imbalance problem. To address these challenges, a unique algorithm, decoupled self-supervised learning for anomaly detection (DSLAD), is proposed in this paper. DSLAD is a self-supervised method with anomaly discrimination and representation learning decoupled for anomaly detection. DSLAD employs bilinear pooling and masked autoencoder as the anomaly discriminators. By decoupling anomaly discrimination and representation learning, a balanced feature space is constructed, in which nodes are more semantically discriminative, as well as imbalance issue can be resolved. Experiments conducted on various six benchmark datasets reveal the effectiveness of DSLAD.

Keywords Anomaly detection · Graph neural networks · Graph mining · Self-supervised learning

✉ Chuan Chen
chenchuan@mail.sysu.edu.cn

YanMing Hu
huym27@mail2.sysu.edu.cn

BoWen Deng
dengbw3@mail2.sysu.edu.cn

YuJing Lai
laiyj23@mail2.sysu.edu.cn

Hao Lin
linhao@mucfc.com

ZiBin Zheng
zhzibin@mail.sysu.edu.cn

Jing Bian
mcsbj@mail.sysu.edu.cn

[1] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China

[2] School of Software Engineering, Sun Yat-sen University, Zhuhai, China

[3] School of System Science and Engineering, Sun Yat-sen University, Guangzhou, China

[4] Merchants Union Consumer Finance Co., Ltd, Shenzhen, China

1 Introduction

To display the intricate and interconnected data, attributed graphs are frequently employed. Recently, anomaly detection on attributed graphs has attracted lots of interest, which seeks to identify some minority patterns (such as nodes, and edges.) that deviate from the majority tremendously on the graph [1]. Anomaly detection on the attributed graph can be deployed in many real-world scenarios, such as spotting fraud in transaction networks, spotting incorrect citation relations among academic papers, and spotting users who deliver spam in postal transportation networks.

However, anomaly detection on attributed graphs is quite a challenging task that primarily faces three challenges. First, it is a heavy cost to obtain enough labels for anomalous nodes. Therefore, supervised models are not applicable for anomaly detection, as evidenced by the fact that ground-truth labels and the class of anomalies are always unknown [2]. Second, anomalous nodes' neighbors are commonly normal nodes. GNN-based algorithms largely rely on aggregating messages from neighbors [3–6]. The messages of abnormal nodes are mostly in the high-frequency part, which contradicts the low-pass filtering of graph neural

Published online: 04 May 2024

networks [7, 8]. As a consequence, anomalous nodes are buried by messages from normal nodes, leading anomalous nodes to be represented similarly to normal nodes, which is referred as semantic mixture. The workflow of GNN-based anomaly detection algorithms is shown in Fig. 1. A visualization of anomaly detection conducted by graph convolution network (GCN) on Cora is displayed in Fig. 2. It is easy to see that quite a few abnormal nodes are mixed with normal nodes. The representations with the semantic mixture would interfere with anomaly detection, especially for the algorithms that mainly focus on anomaly discrimination. However, algorithms that only upgrade the parameters of the model through anomaly detection errors neglect the significant improvement that representation learning brings. Third, the number of anomalous nodes is far less than that of normal nodes. Traditional deep learning algorithms suffer from imbalance issue [9–13] that the majority dominates the embedding training and the minority is often mistakenly identified as the majority. It is observed in Fig. 2 that the representations of abnormal nodes are scattered among normal nodes. Therefore, it is urgent to propose an effective self-supervised algorithm to address the above three challenges for anomaly detection.

Several methods for anomaly detection on attributed graphs have been proposed. These methods have achieved great success in anomaly discrimination, but they still have some drawbacks. The shallow methods, such as AMEN [14], are limited by the capacity for expressiveness. The node-classification-targeted methods, such as DOMINANT [2], simply training model by graph reconstruction and taking reconstruction errors as anomaly scores, are not directly designed for anomaly detection. The anomaly-detection-targeted methods, such as CoLA [15] optimize the model directly for anomaly detection, but they mainly revolve around anomaly discrimination, paying insufficient attention to representation learning, which leads to semantic mixture and imbalance issue.

To overcome the aforementioned challenges, in this paper, we propose a novel method DSLAD for anomaly detection. In DSLAD, both contrastive learning and generative learning are adopted to discriminate anomaly. Especially, DSLAD contrasts node-subgraph pairs and measures reconstruction errors to calculate anomaly scores. The anomaly score is further categorized into context anomaly score and reconstruction anomaly score, deployed with bilinear pooling and masked autoencoder respectively as anomaly discriminator. Considering semantic mixture and imbalance issue, we introduce contrastive representation learning and decouple it with anomaly discrimination. The contrastive representation learning and anomaly discrimination work as two branches. The contrastive representation learning module does not involve the output of anomaly discrimination, providing extra optimization for the graph neural network. Through decoupling anomaly discrimination and contrastive representation learning, DSLAD maps nodes into a balanced semantic space with a little semantic mixture.

The contributions of this work can be summarized as follows:

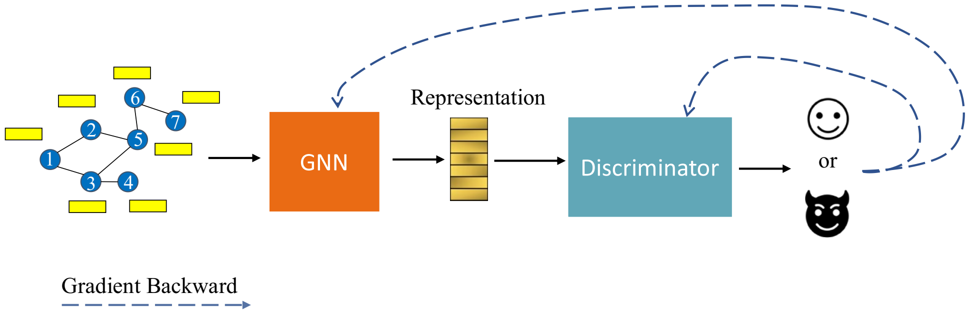
- We introduce contrastive representation learning into the anomaly detection task, which makes nodes more semantically distinguishable and vastly benefits anomaly discrimination.
- We decouple contrastive representation learning and anomaly discrimination, adopting a dynamic weight allocation strategy for these two task branches, ulteriorly resolving semantic mixture and imbalance issue in anomaly detection.
- We conduct a series of experiments on six datasets and the results demonstrate the superiority of DSLAD over the existing models.

2 Related Work

2.1 Graph Neural Networks

Recently, graph representation learning has achieved considerable success with GNNs. The core idea of GNNs is aggregating messages from neighbors to update node representations, which is based on the assortativity assumption. GNNs can be divided into two categories: spectral-based methods and spatial-based methods. The former category includes GCN [4], passing message by first-order approximation of Chebyshev filter. The latter category includes GAT [6] and GraphSAGE [16]. GAT utilizing the attention mechanism,

Fig. 1 Workflow of GNN-based anomaly detection algorithms



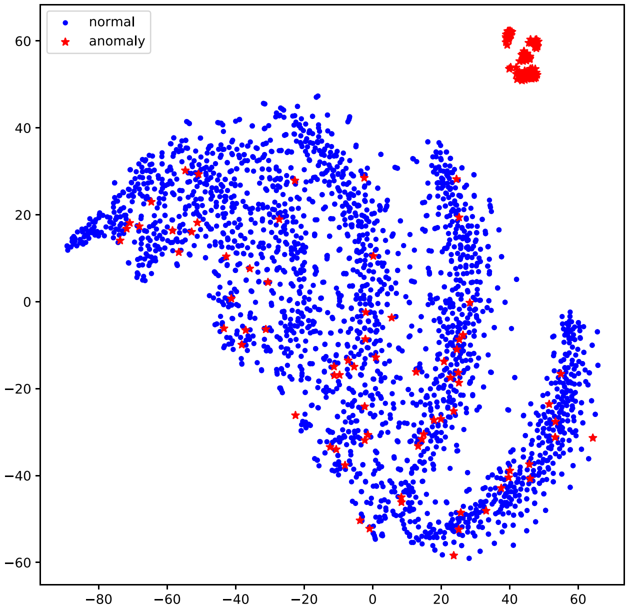


Fig. 2 A visualization of the semantic mixture on Cora. Quite a portion of anomalies are mixed with normal nodes and the anomalies can not be detected well

assigns weight to each edge when aggregating messages. GraphSAGE proposes an inductive representation learning manner to cope with tasks on large-scale graphs.

Apart from the above fundamental GNN models, many advanced GNN models are also proposed to learn the graph representations better. To avoid the sparsity issue and filter the noise information, [17] proposes a framework preserving low-order proximities, mesoscopic community structure information and attribute information for network embedding. MTSN [18], a dynamic graph neural network, captures local high-order graph signals, learns attribute information based on motifs, and preserves timing information by temporal shifting. To alleviate the over-smoothing issue, NAIE [19] adopts an adaptive strategy to smooth attribute information and topology information, and develop an autoencoder to enhance the embedding capacity.

2.2 Graph Self-supervised Learning

Self-supervised graph learning, a new learning paradigm that trains models without labels, has been widely used in computer vision [20] and natural language processing [21]. Self-supervised learning in graph can be categorized into: graph contrastive learning (e.g. SimGRACE [22]), graph generative learning (e.g. Graph Completion [23]) and graph predictive learning (e.g. CDRS [24]). Without augmentation, SimGRACE uses a formal encoder and a perturbation encoder to embed the graph, and then pulls close the same semantics while pushing away the different semantics between the two hidden spaces. Graph Completion removes features of the target node, and then reconstructs it from the unmasked neighboring nodes. CDRS makes a pseudo node classification task collaborated with the clustering task to improve representation learning.

2.3 Anomaly Detection on Attributed Graph

Anomaly detection on attributed graph works for identifying patterns that notably diverge from the majority. Many methods have been proposed for anomaly detection, including the shallow methods and the deep methods. The shallow methods include [14], Radar [25], and ANOMALOUS [26]. AMEN measures the correlation of features between the target node and its ego-networks to detect anomalies. Radar analyzes the residuals of attribute information and its coherence with graph information to detect anomalies. ANOMALOUS integrates CUR decomposition and residual analysis to detect anomalies. The shallow methods are limited by their expressiveness ability in graph embedding. The deep methods can be further divided into two classes. The first class deep methods include DOMINANT [2] and DGIAD [15, 27]. DOMINANT reconstructs the adjacency matrix and the attribute matrix, then distinguish anomaly through reconstruction error. DGI contrasts node and graph for embedding. Deployed with a trained discriminator, DGI can be used for anomaly detection and we rename this method as DGIAD in this paper. The first class deep methods, node classification models equipped only with an anomaly discriminator, are not developed for anomaly detection and still don’t show satisfactory performance. The second class deep methods include HCM [28], CoLA [15], SL-GAD [29], ANEMONE [30],CONAD [31], Sub-cr [32] and GRADATE [33]. HCM trains GNNs with self-supervised learning and Bayesian learning to distinguish anomalies. CoLA, SL-GAD and ANEMONE contrast nodes and subgraphs to discriminate anomaly. CONAD augments graphs with prior knowledge for anomaly detection training. Sub-cr enhances anomaly detection by multi-view contrastive learning and graph reconstruction. GRADATE extends contrastive learning to node-node level and subgraph-subgraph level. The second class deep methods optimize model for anomaly detection, but neglect representation learning to overcome semantic mixture and imbalance issue.

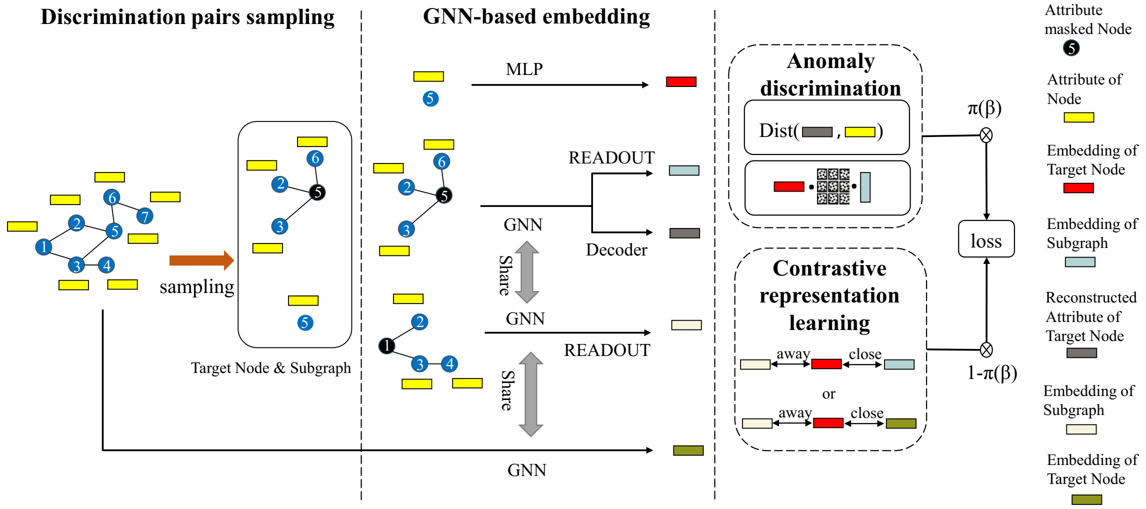


Fig. 3 Framework of DSLAD. There are four components in DSLAD: discrimination pairs sampling, GNN-based embedding, anomaly discrimination and contrastive representation learning. DSLAD firstly selects a set of target nodes and samples neiboring subgraphs of them. Next, the nodes and the sampled subgraphs are embedded into low dimension vectors by GNN for anomaly detection and contrastive representation learning. Finally, the discriminators measures the distance between node-subgraph pairs to discriminate anomaly while target nodes are pulled close to positive samples and pushed away from negative samples (when training only). Especially, anomaly discrimination and representation learning are decoupled

3 Problem Definition

In this section, the problem definition of anomaly detection on attributed graph will be introduced. Given an attributed graph $\mathcal{G} = (\mathcal{V}, \mathbf{X}, \mathbf{A})$, the target of anomaly detection is to learn a mapping mechanism $\mathcal{F}(\cdot)$ to calculate the anomaly score $\mathbf{s}_i, i \in \mathcal{V}$ for nodes in $\mathcal{G}$. The anomaly score $\mathbf{s}_i$ describes the abnormal degree of the node i. It is easy to detect an anomaly, if the mapping mechanism $\mathcal{F}(\cdot)$ is well designed and outputs accurate anomaly scores. For the convenience of reading this paper, all important notations are explained in Table 1.

4 Method

In this section, a thorough introduction to DSLAD will be given. As shown in Fig. 3, DSLAD consists of four modules, discrimination pair sampling, GNN-based embedding, anomaly discrimination and contrastive representation learning. On attributed graph, contrastive learning at the node-subgraph level is powerful for graph representation learning [34, 35]. It has been discovered that detecting anomalies at the node-subgraph level is effective [15]. To detect the anomaly, we sample discrimination pairs at node-subgraph level. The target nodes and their sampled subgraphs are then embedded into low-dimension vectors via GNN. Next, the embedding vectors of target nodes and their sampled subgraphs are fed into anomaly discrimination and contrastive representation learning. By contrastive representation learning, the semantic mixture and imbalance issue can be lightened, and decoupling anomaly discrimination and contrastive representation learning can further alleviate it.

4.1 Discrimination Pair Sampling

The key to anomaly detection is finding the patterns significantly different from the majority. Therefore, discrimination pairs are crucial to this task. Graph objects can be categorized into edge, node, subgraphs, and graph. Any two of them, excluding edge, can be selected to constitute discrimination pairs. We sample discrimination pairs at node-subgraph level. The procedure is as follows:

- *Target node selection* A set of nodes are randomly selected from the input graph every epoch without replacement so that each node has the same chance of being chosen.
- *Subgraph sampling* For every selected target node, a neighboring subgraph is sampled via random walks with restart (RWR) [36] as augmentation, avoiding introducing extra anomalies. Other sampling methods also can be considered. The size of the neighboring subgraph is fixed to K, which determines the scope of the target node for matching.
- *Attribute mask* The attributes of the target node are masked with zero vectors in the sampled subgraph, making it more difficult to identify the information of the tar-

Table 1 Statements of important notations

Notations	Statements
$\mathcal{G} = (\mathcal{V}, \mathbf{X}, \mathbf{A})$	An attributed graph
v_i	The i-th node in $\mathcal{G}$
$\mathcal{G}_i$	The subgraph originated from v_i
K	The number of nodes in $\mathcal{G}_i$
$\mathbf{X} \in \mathbb{R}^{N \times d(0)}$	The attribute matrix of $\mathcal{G}$
$\mathbf{A} \in \mathbb{R}^{N \times N}$	The adjacency matrix of $\mathcal{G}$
$\mathbf{X}_{\{i\}} \in \mathbb{R}^{K \times d(0)}$	The attribute matrix of $\mathcal{G}_i$
$\mathbf{A}_{\{i\}} \in \mathbb{R}^{K \times K}$	The adjacency matrix of $\mathcal{G}_i$
$\mathbf{h}^{(l)}_{\{i\}} \in \mathbb{R}^{1 \times d(l)}$	The embedding of v_i in the l−th layer
$\mathbf{W}^{(l))} \in \mathbb{R}^{d(l-1) \times d(l)}$	The weight matrix in the l-th layer
$\mathbf{H}^{(l)}_{\{i\}} \in \mathbb{R}^{N \times d(l)}$	The hidden matrix in the l-th layer of $\mathcal{G}_i$
$\mathbf{Z}^i \in \mathbb{R}^{K \times d}$	The context representation matrix of $\mathcal{G}_i$
$\mathbf{U}_{\{i\}} \in \mathbb{R}^{N \times d(0)}$	The reconstructed attribute matrix of $\mathcal{G}_i$
$\mathbf{g}_i \in \mathbb{R}^{1 \times d}$	The subgraph-level representation of $\mathcal{G}_i$
$\mathbf{e}_i \in \mathbb{R}^{1 \times d}$	The node-level representation of v_i
$x_i \in \mathbb{R}^{1 \times d(0)}$	The attribute of v_i
$\mathbf{W}_d \in \mathbb{R}^{d \times d}$	The weight matrix of bilinear pooling
$\mathbf{s}_i^{con(-)}$	The negative context anomaly score of v_i
$\mathbf{s}_i^{con(+)}$	The positive context anomaly score of v_i
$\mathbf{s}_i^{rec}$	The reconstruction anomaly score of v_i
$\mathbf{s}_i$	The anomaly score of v_i

get node in the subgraph. This mechanism will improve the ability of anomaly detection [15, 29].

Target nodes and neighboring subgraphs are combined as discrimination pairs for anomaly discrimination. A positive pair includes a node and a subgraph sampled from it, while a negative pair includes a node and a subgraph sampled from other nodes. A toy sample of positive pair and negative pair is displayed in Fig. 4 for a better understanding.

4.2 GNN-Based Embedding

For anomaly discrimination and contrastive representation learning, obtained target nodes and their neighboring subgraphs are mapped into low-dimensional embedding space by GNNs.

We apply a GCN encoder and a GCN autoencoder to embed the graph and reconstruct the attribute matrix, respectively.

Target node v_i is embedded as a graph with only one node. The GNN propagation formula can thus be simplified to MLP:

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\mathbf{h}_i^{(l)}\mathbf{W}^{(l)}\right). \tag{1}$$

And $\mathbf{e}_i \in \mathbb{R}^d$ is used to denote the output of GCN encoder, which is the node level representation vector of v_i.

On the K-nodes subgraph $\mathcal{G}_i$ sampled from node v_i, the adjacency matrix is denoted by $\mathbf{A}_{\{i\}} \in \mathbb{R}^{K \times K}$ and the attribute matrix is denoted by $\mathbf{X}_{\{i\}} \in \mathbb{R}^{K \times d(0)}$. Then, the GNN operator is applied to:

$$\mathbf{H}_{\{i\}}^{(l+1)} = \sigma\left(\widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}_{\{i\}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}_{\{i\}}^{(l)}\mathbf{W}^{(l)}\right), \tag{2}$$

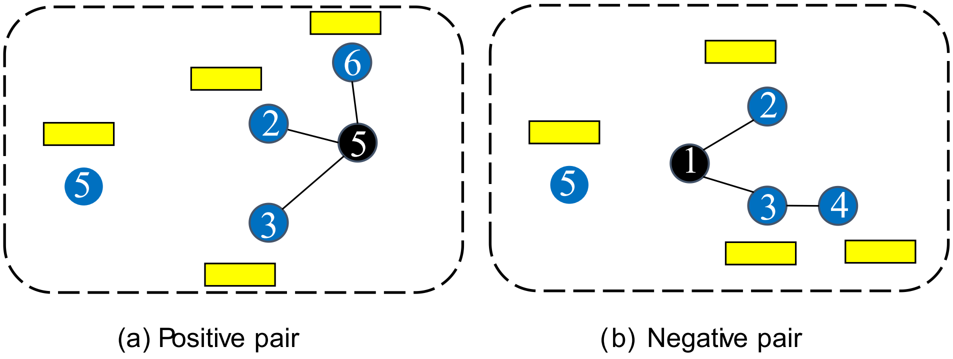
where $\widetilde{\mathbf{A}}_{\{i\}} = \mathbf{A}_{\{i\}} + \mathbf{I}_K$, and $\mathbf{H}_{\{i\}}^{(0)} = \mathbf{X}_{\{i\}}$.

The output of the GCN encoder is denoted by $\mathbf{Z}^i \in \mathbb{R}^{K \times d}$, which is the context representation matrix of subgraph $\mathcal{G}_i$. And the output of the GCN autoencoder on $\mathcal{G}_i$ is denoted by $\mathbf{U}_{\{i\}} \in \mathbb{R}^{K \times d(0)}$, which is the reconstructed attribute matrix of $\mathcal{G}_i$.

The readout module summarizes $\mathbf{Z}^i$ into its subgraph-level representation $\mathbf{g}_i \in \mathbb{R}^d$. We take average pooling as the readout module. The subgraph-level representation can then be formulated as:

$$\mathbf{g}_i = readout(\mathbf{Z}^i) = \frac{1}{K-1}\sum_{j=0, j \neq c_i}^{K} \mathbf{Z}^i\,[j, :], \tag{3}$$

where c_i is the index of v_i in neighboring subgraph $\mathcal{G}_i$.



(a) Positive pair

(b) Negative pair

Fig. 4 A toy sample of positive pair and negative pair. The subgraph is sampled by RWR that originates from the black node

4.3 Anomaly Discrimination

In this subsection, We will describe how DSLAD discriminates anomaly. Context anomaly and reconstruction anomaly are two subtypes of anomaly discrimination. Here is more information on them in depth:

4.3.1 Context Anomaly

Anomalies differ from the other majority significantly. The anomalous nodes are supposed to be far away from normal nodes in the embedding space. To assess how a discrimination pair matches, we take bilinear pooling as the discriminator.

Given a node v_i and the relevant neighboring subgraph $\mathcal{G}_i$, the context anomaly score of this discrimination pair can be calculated as:

$$\mathbf{s}_i^{con} = disc(\mathbf{g}_i, \mathbf{e}_i) = \phi(\mathbf{g}_i \mathbf{W}_d {\mathbf{e}_i}^T), \tag{4}$$

where $\mathbf{W}_d \in \mathbb{R}^{d \times d}$ is a learnable weight matrix, and $\phi(\cdot)$ is non-linear and non-negative activation function. Here we use Sigmoid as the activation function.

We use graph contrastive learning at the node-subgraph level, taking both positive and negative discrimination pairs into account. For target node v_i, we take $\mathbf{P}$ positive discrimination pairs and $\mathbf{Q}$ negative discrimination pairs to compute context anomaly score. The positive score $\mathbf{s}_i^{con(-)}$ and the negative score $\mathbf{s}_i^{con(+)}$ are formulated as:

$$\mathbf{s}_i^{con(-)} = \frac{1}{\mathbf{Q}} \phi\left(\sum_{\mathbf{g}_j \in \{\mathbf{g}^{(-)}\}} \mathbf{g}_j \mathbf{W}_d \mathbf{e}_i^T \right), \tag{5}$$

$$\mathbf{s}_i^{con(+)} = \frac{1}{\mathbf{P}} \phi\left(\sum_{\mathbf{g}_j \in \{\mathbf{g}^{(+)}\}} \mathbf{g}_j \mathbf{W}_d \mathbf{e}_i^T \right), \tag{6}$$

where $\{\mathbf{g}^{(-)}\} \in \mathbb{R}^d$ and $\{\mathbf{g}^{(+)}\} \in \mathbb{R}^d$ denote the positive neighboring subgraph set and the negative neighboring subgraph set for the target node v_i, respectively. For simplicity, we set $\mathbf{P} = \mathbf{Q} = 1$.

In this part, our optimization goal is maximizing the agreement with the context anomaly score and the ground-truth label (label 1 for positive pairs and 0 for negative pairs). The loss function of context anomaly score can be formulated as:

$$L_{con} = -\frac{1}{2|\mathcal{V}|} \sum_{i \in \mathcal{V}} \left(log\left(s_i^{con(+)}\right) + log\left(1 - s_i^{con(-)}\right) \right) \tag{7}$$

4.3.2 Reconstruction Anomaly

Inspired by [23, 29], we introduce the reconstruction error as a supplementary mechanism to anomaly discrimination. For the target node v_i, we mask its attributes on the neighboring subgraph $\mathcal{G}_i$. DSLAD tries to reconstruct the attributes of the target node v_i, from the other nodes on $\mathcal{G}_i$. l_2-norm is adopted to quantitatively measure the distance between the original information and the reconstructed information.

The index of v_i in the neighboring subgraph $\mathcal{G}_i$ is c_i. The reconstructed attribute vector of v_i in neighboring subgraph $\mathcal{G}_i$ is denoted by $\mathbf{U}_{\{i\}}[c_i, :]$.

In order to train the masked autoencoder for reconstruction anomaly, we adopt MSE as the loss function for this portion. It can be written as:

$$L_{rec} = -\frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} ||\mathbf{U}_{\{i\}}[c_i, :] - x_i||^2, \tag{8}$$

where $x_i \in \mathbb{R}^{d(0)}$ is the original attribute vector of node v_i.

4.4 Contrastive Representation Learning

In the context anomaly module, the loss function in Eq. (7) mainly focuses on anomaly discrimination and pays less attention to representation learning. Moreover, all nodes are assumed to contribute equally. After message passing, semantic mixture and imbalance issue inevitably occur. To impede the normal nodes from dominating the representation learning, we implement the contrastive representation learning module and set the number of positive samples and negative samples equal.

For the target node v_i, we choose the neighboring subgraph $\mathcal{G}_j$ $(j \neq i)$ sampled from node v_j as the negative sample, in consistency with the context anomaly module. We propose two optional augmentation strategies to generate the positive sample. One augmentation strategy, denoted by *local_aug*, selects the neighboring subgraph $\mathcal{G}_i$ as the positive sample. Another augmentation strategy, denoted by *global_aug*, embeds the entire graph without the mask and then takes the embedding of v_i as the positive sample. The *local_aug* aggregates messages from a part of neighbors and removes the influence of attributes of the target node, while the *global_aug* aggregates messages from all neighbors within a specific hop and retains the influence of attributes of the target node. The key difference relies on the influence of attributes of the target node and neighbor selection. $\mathbf{e}_i^-$ denotes representation vector of negative sample, and $\mathbf{e}_i^- = \mathbf{g}_j, j \neq i$. $\mathbf{e}_i^+$ denotes representation vector of positive sample, and

$$\mathbf{e}_i^+ = \begin{cases} \mathbf{g}_i & local\_aug \\ f(\mathbf{X}, \mathbf{A})[i, :] & global\_aug \end{cases}, \tag{9}$$

where $\mathbf{g}_i \in \mathbb{R}^d$ is computed by Eq. (3) and f denotes GCN encoders in the context anomaly module. For a better understanding of the augmentation views, the procedure for generating positive and negative samples is displayed in Fig. 5.

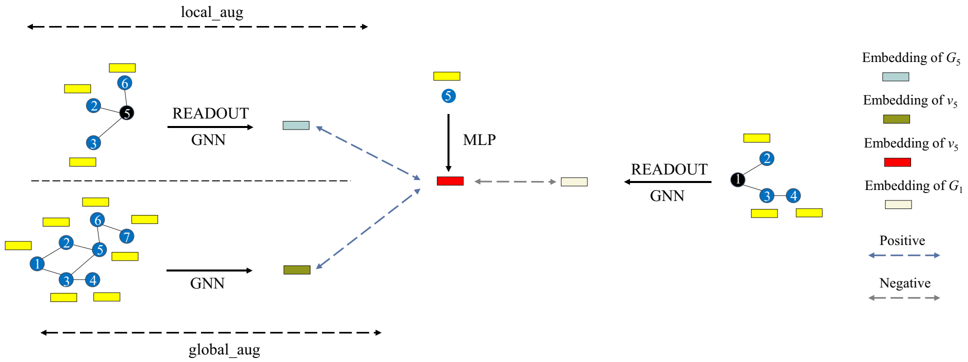


Fig. 5 Procedure of generating positive samples and negative samples

Both the number of positive samples and negative samples are set to 1 for simplicity and fairness. We adopt infoNCE [37] as the loss function of contrastive representation learning:

$$L_{CL} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}}^{|\mathcal{V}|} -log \frac{exp^{(\mathbf{e}_i \cdot \mathbf{e}_i^+ / \tau)}}{exp^{(\mathbf{e}_i \cdot \mathbf{e}_i^+ / \tau)} + exp^{(\mathbf{e}_i \cdot \mathbf{e}_i^- / \tau)}}, \tag{10}$$

where τ is a temperature parameter greater than 0.

4.5 Decoupling

In this subsection, we explain why and how we decouple anomaly discrimination and contrastive representation learning. When behaviors and label semantics are excessively inconsistent in anomaly detection tasks, [38] has shown that training graph representation learning and anomaly discrimination jointly may lead to performance degradation. Moreover, the problem of class imbalance can also be resolved significantly by decoupling representation learning and anomaly discrimination. At the beginning of training, discriminators are prone to predict arbitrarily, producing erroneous results while contrastive representation learning forms a balanced semantic space [39, 40]. During training, discriminators make increasingly accurate predictions while performance gained by contrastive representation learning decays [41, 42]. Gradually shifting to anomaly discrimination from contrastive learning enhances the effectiveness. Based on the above analysis, instead of jointly training by classification loss, we decouple anomaly discrimination and contrastive representation learning and give them dynamic weights. The workflow of our decoupled anomaly detection algorithm is shown in Fig. 6.

Let β denotes the ratio of current epoch to the number of training epochs, whose value indicates the training process. $\pi(\beta)$ is the factor balancing the anomaly discrimination loss and the contrastive representation learning loss, where $\pi(\cdot)$ is a mapping function. The final loss function can be written as:

$$L = \pi(\beta)(\alpha L_{con} + (1 - \alpha)L_{rec}) + \lambda(1 - \pi(\beta))L_{CL}, \tag{11}$$

where α and λ are the hyperparameters that control the role of different anomaly scores, and scale contrastive representation learning, respectively. And $\pi(\beta)$ increases with β.

4.6 Anomaly Score Calculation

We could calculate the final anomaly score for each node after training.

For node v_i, context anomaly score $\mathbf{s}_i^{con}$ and reconstruction anomaly score $\mathbf{s}_i^{rec}$ can be inferenced as follows:

$$\mathbf{s}_i^{con} = \mathbf{s}_i^{con(-)} - \mathbf{s}_i^{con(+)}, \tag{12}$$

where $\mathbf{s}_i^{con(-)}$ and $\mathbf{s}_i^{con(+)}$ are calculated by Eqs. (5) and (6) respectively. The score function can simultaneously consider the impact of both positive and negative pairs. For normal nodes, the similarity of positive pair matching is high, while the similarity of negative pair matching is low. For anomalous nodes, the similarity of positive and negative pair matching are both low.

$$\mathbf{s}_i^{rec} = ||\mathbf{U}_{\{i\}}[c_i, :] - x_i||_2^2, \tag{13}$$

where the index of v_i in the neighboring subgraph $\mathcal{G}_i$ is c_i, the reconstructed attribute vector of v_i in neighboring subgraph $\mathcal{G}_i$ is $\mathbf{U}_{\{i\}}[c_i, :]$, and x_i is the original attribute vector of node v_i.

By MinMaxScalar, we transform the context anomaly score $\mathbf{S}^{con}$ to [0,1] for standardization:

$$\mathbf{s}_i^{con} = \frac{\mathbf{s}_i^{con} - \mathbf{s}_{min}^{con}}{\mathbf{s}_{max}^{con} - \mathbf{s}_{min}^{con}}, \tag{14}$$

where $\mathbf{s}_{min}^{con}$ and $\mathbf{s}_{max}^{con}$ are the min and the max of context anomaly scores, respectively. Similarly, reconstruction anomaly score $\mathbf{S}^{rec}$ is also transformed to [0,1] by MinMaxScalar:

$$\mathbf{s}_i^{rec} = \frac{\mathbf{s}_i^{rec} - \mathbf{s}_{min}^{rec}}{\mathbf{s}_{max}^{rec} - \mathbf{s}_{min}^{rec}}, \tag{15}$$

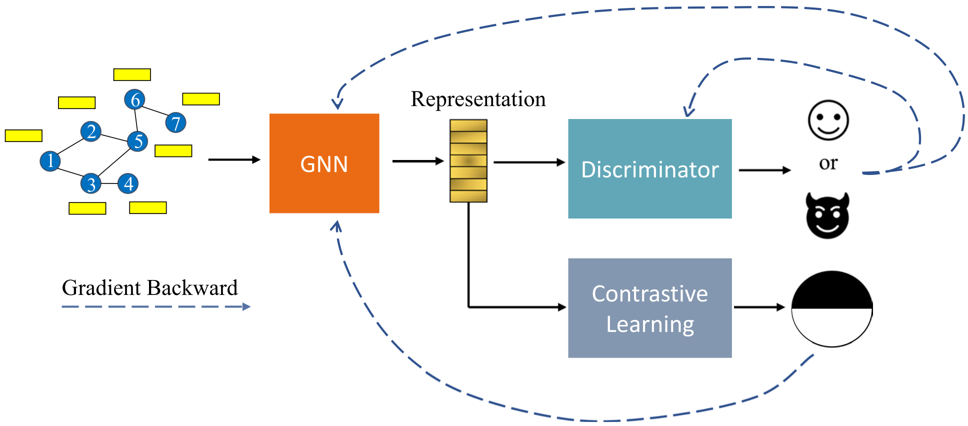


Fig. 6 Workflow of our decoupled anomaly detection algorithm

where $\mathbf{s}^{rec}_{min}$ and $\mathbf{s}^{rec}_{max}$ are the min and the max of reconstruction anomaly scores, respectively.

Combining transformed context anomaly score and reconstruction anomaly score, we can get the final anomaly score $\mathbf{s}_i$ of node v_i:

$$\mathbf{s}_i = \alpha \mathbf{s}^{con}_i + (1 - \alpha)\mathbf{s}^{rec}_i. \tag{16}$$

Neighboring subgraph is sampled stochastically. To reduce the sampling variance, we take the averaging anomaly score over R times as the final anomaly score.

5 Experiments

In this part, a succession of experiments are carried out on six real-world datasets to examine the effectiveness of our model.

5.1 Datasets

Six frequently used real-world datasets for anomaly identification, including four citation network datasets and two social network datasets, are applied to evaluate our model.

The following is a brief overview of the six datasets:

- *Citation network datasets* Cora, Citeseer, Pubmed [43] and ACM [44] are four public citation network datasets, composed of scientific publications. In the four citation networks, the published papers are transformed into nodes while edges represent the citation relationships between papers. And the description text of papers can be transformed into nodes features.
- *Social network datasets* BlogCatalog and Flickr [45] are acquired from the websites for sharing blogs and images, respectively. In the two datasets, each user is represented by a node, and links among nodes illustrate the relationships between corresponding users. Users often describe themselves with personalized information, such as posting blogs and public photos. Features can be extracted from such information.

Considering that there are no ground-truth anomaly labels in above six real-world datasets, injecting synthetic anomaly nodes into datasets to simulate real anomalies is used widely. We follow the perturbation processing in [2, 15] to inject anomalies with both attribute anomalies and structure anomalies into the six datasets. For the attribute anomaly injection, we select M_a nodes and replace their features with stochastically selected remote nodes. For the structure anomaly injection, we pick up M_a nodes and divide them into M_c clusters averagely. Nodes within the same cluster are connected with each other. The statistics of these contaminated datasets are depicted in Table 2.

5.2 Baselines

We choose some of the state-of-the-art methods as baselines to compare with our proposed DSLAD on the above six real-world datasets. These methods are divided into three categories:

(1) The shallow method: The shallow method detects anomalies without deep learning. We pick up the following three models for comparison:

Table 2 The statistics of datasets

Dataset	Anomalies	nodes	Features	Edges
Cora	150	2,708	1,433	5,429
Citeseer	150	3,327	3,703	4,732
Pubmed	600	19,717	500	44,338
ACM	600	16,484	8,337	71,980
BlogCatalog	300	5,196	8,189	171,743
Flickr	450	7,575	12,407	239,739

- *AMEN* [14] compares the correlation of features of the target nodes and their ego-networks to identify the nodes with low scores as anomalies.
- *Radar* [25] analyzes the residuals of attribute information and its coherence with graph information to detect the abnormal nodes as anomalies.
- *ANOMALOUS* [26] utilizes CUR decomposition and residual analysis to distinguish the irregular nodes as anomalies.

(2) The node-classification-targeted method: The node-classification-targeted methods simply expand the node classification model with an anomaly detection module to detect anomalies. We choose the following two models for comparison:

- *DOMINANT* [2] learns node embeddings by autoencoders and take the reconstruction errors as the anomaly scores.
- *DGIAD* [15, 27] uses DGI to learn node embeddings and takes the bilinear pooling to compute the anomaly scores.

(3) The anomaly-detection-targeted method: The anomaly-detection-targeted method is designed to detect anomalies directly, even without considering node classification. We select the following three models for comparison:

- *CoLA* [15] learns node embedding by GCN and contrasts nodes and subgraphs to discriminate anomaly.
- *ANEMONE* [30] expands *CoLA* with the patch-level contrast.
- *SL-GAD* [29] expands *CoLA* with the reconstruction error.
- *Sub-cr* [32] adopts graph diffusion for contrastive learning and enhances the anomaly detection by graph reconstruction.
- *GRADATE* [33] expands *CoLA* to node-node level contrast and subgraph-subgraph level contrast.

5.3 Evaluation Metrics

We utilize ROC-AUC, a widely used metric for anomaly detection, to quantify the performance of DSLAD and the baselines. The ROC curve is depicted by the true positive rate (y-variable) and the false positive rate (x-variable). AUC is the area enclosed by the ROC curve the x-axis. AUC always falls between 0 and 1. The better performance is indicated by the higher AUC.

5.4 Experiments Setting

We set neighboring subgraph size $k \in Z^+$ in [2,10]. Layers of GNN encoders and GNN decoders are set as 3 on Flickr, and 1 on the other five datasets. The hidden dimension d is set as 64 while test rounds $R = 256$. The batch size is 300. We choose $\pi(\beta) = \beta$ and select λ from $\{0.5, 1, 1.5, 2, 2.5, 3\}$. Our proposed model is implemented in server with Ubuntu 20.04.1 LTS, Pytorch 1.10, dgl0.4.3post2, Intel(R) Xeon(R) Gold 6132 CPU @ 2.60GHz, and GeForce RTX 2080 Ti. We execute DSLAD over 8 times to measure the effectiveness statistically.

5.5 Comparison Results

To verify the effectiveness of our model in anomaly detection task, we conducted comparison experiments for all baselines and DSLAD with AUC metric on six benchmark datasets and results are shown in Table 3. Based on the results, we can make the following observations:

- Compared with the most advanced baselines, our method outperforms baselines on all benchmark datasets with a large margin. It reveals the effectiveness of our method.
- The shallow methods AMEN, Radar, and ANOMALOUS perform worse than other baselines because of the limitation of expressiveness capacity.
- The node-classification-targeted methods DOMINANT and DGIAD perform better than shallow methods. DOMINANT reconstructs attribute matrix and adjacency matrix, not directly targeting to detect anomalies. DGIAD contrasts the nodes and the whole graph, utilizing very little local information.
- The anomaly-detection-targeted methods CoLA, ANEMONE, SL-GAD, Sub-cr and GRADATE make a step further. However, they mainly concentrate on training anomaly discriminator, still constrained by semantic mixture and imbalance issue.

5.6 Differences in the Distribution of Anomaly Scores

The anomaly score measures the agreement among target nodes and positive (negative) samples. The semantic mixture can be revealed by the differences in the distribution of anomaly scores. To explore the semantic mixture between normal and abnormal nodes, we use the Wasserstein distance to describe the difference in the distribution of anomaly scores. The Wasserstein distance between the scores of normal and abnormal nodes on six datasets is demonstrated in Table 5. It can be observed that the result of Wasserstein

Table 3 Comparison experiment results of anomaly detection by AUC metric on six benchmark datasets

Methods	Cora	Citeseer	Pubmed	BlogCatalog	Flickr	ACM
AMEN [2016]	0.6266	0.6154	0.7713	0.6392	0.6573	0.5626
Radar [2017]	0.6587	0.6709	0.6233	0.7401	0.7399	0.7247
ANOMALOUS [2018]	0.5770	0.6307	0.7316	0.7237	0.7434	0.7038
DOMINANT [2018]	0.8155	0.8251	0.8081	0.7468	0.7442	0.7601
DGIAD [2019]	0.7511	0.8293	0.6962	0.5827	0.6237	0.6240
CoLA [2021]	0.8799	0.8968	0.9512	0.7854	0.7513	0.8237
SL-GAD [2021]	<u>0.9130</u>	0.9136	0.9672	<u>0.8184</u>	<u>0.7966</u>	0.8538
ANEMONE [2021]	0.9057	0.9189	0.9548	0.8067	0.7637	0.8709
Sub-cr [2022]	0.9080	<u>0.9331</u>	<u>0.9677</u>	0.8139	0.7872	0.8047
GRADATE [2023]	0.9062	0.9233	0.9578	0.7517	0.7348	<u>0.8722</u>
Ours	**0.9196**	**0.9481**	**0.9772**	**0.8275**	**0.8631**	**0.8809**

The best performance and the second-best performance methods are marked by bold and underlined fonts respectively. P-value=0.0331 (popmean=mean+std), and the std can be seen in Table 4

Table 4 Standard deviation and P-value on six datasets

	Mean	Std	P-value(mean+std)
Cora	0.9196	0.0028	0.0331
Citeseer	0.9481	0.0027	
Pubmed	0.9772	0.0004	
ACM	0.8809	0.0022	
BlogCatalog	0.8275	0.0041	
Flickr	0.8631	0.0017	

distance meets with the result in Table 3. In terms of anomaly scores, there is a significant difference between normal nodes and abnormal nodes. The semantic mixture can be alleviated by our method.

5.7 Augmentation Strategy

In this subsection, we assess the effectiveness of the augmentation strategy to our method. As illustrated in Fig. 7a, our model is not sensitive to the augmentation strategy on Cora, and performs better with *local_aug* than with *global_aug* on the other datasets except for Flickr.

The main reason may be that Flickr has the most complex attribute information, which is more important than structure information. And on the other five datasets, structure information has a greater impact on the other five datasets than attribute information.

Based on the above observations, we choose *global_aug* as augmentation strategy on Flickr, and *local_aug* as augmentation strategy on the rest five datasets.

5.8 $\pi(\beta)$ Strategy

In this subsection, we investigate how different mapping functions $\pi(\beta)$ would effect our model. Constant e.g. 0.5, linear growth e.g. β, and activation function e.g. $1 - exp(-\beta)$, $sigmoid(\beta)$,and $tanh(\beta)$ are taken into consideration. As demonstrated in Fig. 7 (b), when setting $\pi(\beta) = \beta$, the best results are acquired, although our model is not sensitive to $\pi(\beta)$ on Pubmed and Flickr. Linear $\pi(\beta)$ also has the best generalization.

5.9 Ablation Studies

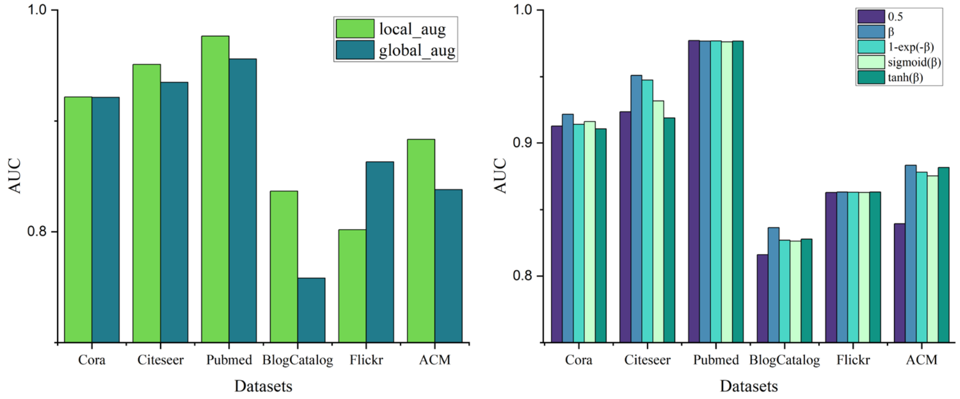
In this subsection, we conduct ablation studies for better understanding the effectiveness of each components in our method. Here, three variants are defined as:

- *DSLAD w/o cl*: remove contrastive representation learning and set $\pi(\beta) = 1$.
- *DSLAD w/o con*: remove context score.
- *DSLAD w/o rec*: remove reconstruction score.

DSLAD w/o cl can be regarded as a variants with only anomaly detection. *DSLAD w/o con* can be regarded as a variants without context score. *DSLAD w/o rec* can be regarded as a variants without reconstruction score.

Table 5 Wasserstein distance ($\times 10^{-2}$) between the scores of normal and abnormal nodes on six datasets

	Cora	Citeseer	Pubmed	BlogCatalog	Flickr	ACM
Ours	19.0854	19.6866	26.8413	14.7390	15.2717	15.8089



(a) Positive augmentation

(b) Mapping function $\pi(\cdot)$

Fig. 7 Performance comparison between different positive augmentation strategies and mapping function $\pi(\cdot)$

Table 6 Ablation studies on six benchmark datasets. How each module would effect the whole model is explored

Variants	Cora	Citeseer	Pubmed	BlogCatalog	Flickr	ACM
DSLAD	**0.9196**	**0.9481**	**0.9772**	**0.8275**	**0.8631**	**0.8809**
DSLAD w/o cl	0.8948	0.9357	0.9726	0.8163	0.7883	0.8232
DSLAD w/o con	0.8275	0.8036	0.8050	0.7474	0.744	0.7463
DSLAD w/o rec	0.9060	0.9023	0.9545	0.7982	0.6954	0.8565

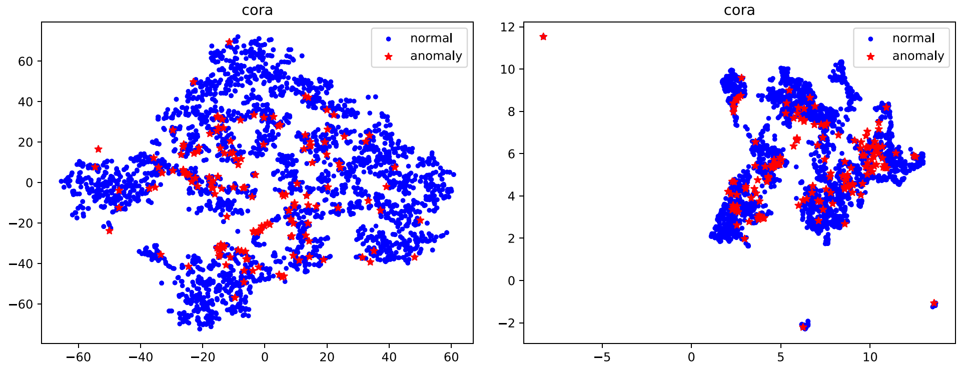
Bold values indicates the best performance

Variants DSLAD w/o cl, DSLAD w/o con and DSLAD w/o rec are generated by removing contrastive representation learning and set $\pi(\cdot)$=1, removing context score and removing reconstruction score, respectively

As shown in Table 6, DSLAD outperforms other variants, indicating that all components play an important role in our method and they could make mutual promotion. Removing contrastive representation learning and setting $\pi(\beta) = 1$ causes remarkable performance degradation. Obviously, decoupling anomaly detection and representation learning tremendously promotes anomaly discrimination by reducing class imbalance and lightening semantic mixture. Additionally, removing either context or reconstruction scores performance would result in performance degradation with the former being more noticeable.This demonstrates that context score and reconstruction score complement each other, while context scores are more effective for anomaly detection.

5.10 Visualization of Representations

In order to verify that the semantic mixture and imbalance issue are resolved by decoupling anomaly detection and representation learning, we demonstrate the visualization of embeddings on Cora in Fig. 8. In Fig. 8a, CoLA can't form sufficiently distinguishable clusters of the anomalous nodes. There are still quite a few anomalous nodes sporadically mixed with the normal nodes. The anomaly detection



(a) CoLA

(b) DSLAD

Fig. 8 The visualization of embeddings: **a** embeddings of CoLA on Cora, **b** context embeddings of DSLAD on Cora

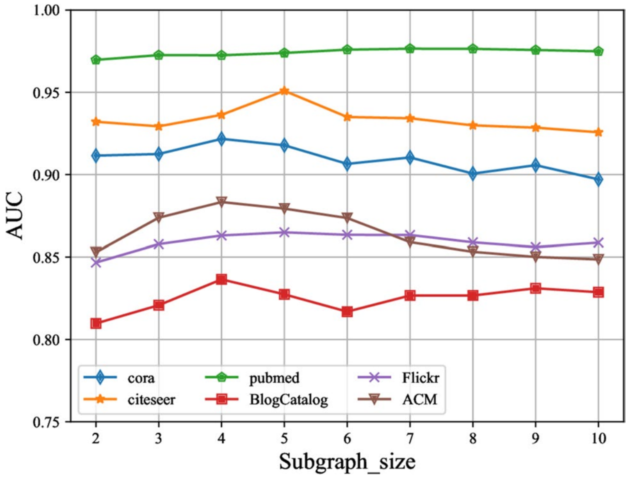


Fig. 9 Performance comparison using different subgraph size on the six benchmark datasets

algorithms, which mainly optimize the model by anomaly discrimination, are often limited by insufficient expressiveness. In Fig. 8b, it can be observed that both the normal nodes and the anomalous nodes are denser and can be better discriminated by DSLAD than by CoLA. Decoupling anomaly detection and representation learning significantly solves the problems of semantic mixture and imbalance.

5.11 Parameter Sensitivity

In this subsection, a series of experiments are conducted to study the effect of hyperparameters.

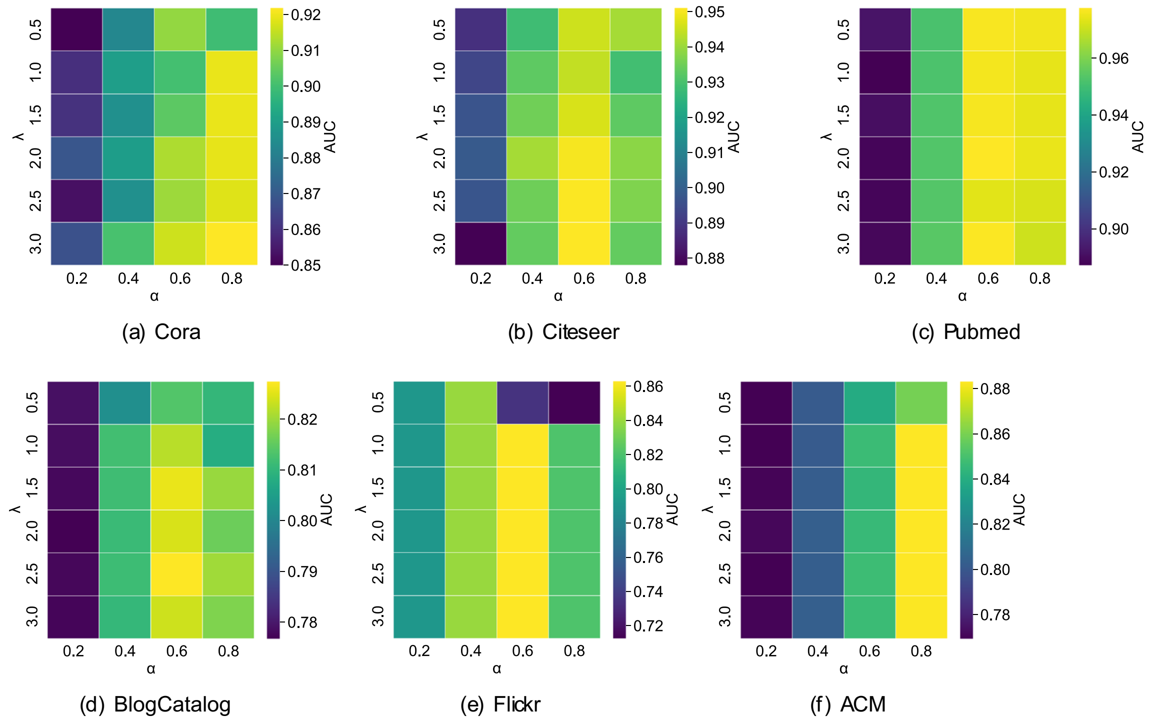
5.11.1 Subgraph Size K

DSLAD is executed on six benchmark datasets with subgraph size K within [2,10]. It is seen from Fig. 9 that AUC grows until the peak and then drops with subgraph size K increasing. DSLAD achieves the AUC peak at $K = 5$ on Citeseer and Flickr, at $K = 7$ on Pubmed, and $K = 4$ on the rest datasets. These results show that too small subgraph contains insufficient information, restricting anomaly detection; Too large subgraph contains tedious information, which would hurt our model; Applicable subgraph size guarantee DSLAD in best performance.

5.11.2 Effect of Hyperparameter α and λ

Besides hyperparameter subgraph size K, we also discuss α and λ.

To explore the effect of hyperparameter α, we select its value from {0.2,0.4,0.6,0.8}. As illustrated in Fig. 10, when $\alpha = 0.6$, DSLAD has the best performance on Citeseer, BlogCatalog, and Flickr. When $\alpha = 0.8$, DSLAD has the



(a) Cora

(b) Citeseer

(c) Pubmed

(d) BlogCatalog

(e) Flickr

(f) ACM

Fig. 10 Parameter sensitivity studies for hyperparameter α and λ. The values of AUC are correlated with colors by viridis

best performance on Cora, Pubmed, and ACM. This shows that no matter any datasets, the context anomaly score has a greater impact than the reconstruction anomaly score, supporting the assertion that the context score affects more.

To explore the effect of hyperparameter λ, we select its value from $\{0.5,1.0,1.5,2.0,2.5,3.0\}$. As illustrated in Fig. 10, DSLAD has the best performance when $\lambda = 3.0$, $\lambda = 3.0$, $\lambda = 2.0$, $\lambda = 1.0$, $\lambda = 2.5$, $\lambda = 1.5$ on Cora, Citeseer, Pubmed, BlogCatalog, Flickr and ACM respectively. A suitable setting of the hyperparameter λ could prompt hidden space mapping and provide significant benefits for anomaly detection.

6 Conclusion

In this paper, a novel framework called DSLAD is proposed for graph anomaly detection. DSLAD is composed of four modules: discrimination pair sampling, GNN-based embedding, anomaly discrimination and contrastive representation learning. Both contrastive learning and generative learning are employed to discriminate anomaly. They complement one another and improve the effectiveness. The contrastive representation learning, greatly alleviating the semantic mixture and imbalance problem, generates a more balanced semantic space and facilitates node embedding. By decoupling anomaly discrimination and contrastive representation learning, the performance of DSLAD is undoubtedly improved. In the future, we will explore a unified representation learning framework for anomaly detection and node classification.

Acknowledgements The research is supported by the National Key Research and Development Program of China (2023YFB2703700), the National Natural Science Foundation of China (62176269), the Guangzhou Science and Technology Program (2023A04J0314).

Availability of Data and Materials All datasets are publicly available.

Declarations

Conflict of interest The authors declare that they have no conflicts of interest.

Ethics Approval Not applicable.

References

1. Ma X, Wu J, Xue S, Yang J, Zhou C, Sheng QZ, Xiong H, Akoglu L (2021) A comprehensive survey on graph anomaly detection with deep learning. IEEE Trans Knowl Data Eng
2. Ding K, Li J, Bhanushali R, Liu H (2019) Deep anomaly detection on attributed networks. In: Proceedings of the 2019 SIAM international conference on data mining, pp 594–602. SIAM
3. Xu K, Hu W, Leskovec J, Jegelka S (2018) How powerful are graph neural networks?. arXiv preprint arXiv:1810.00826
4. Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907
5. Bruna J, Zaremba W, Szlam A, LeCun Y (2013) Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203
6. Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y (2017) Graph attention networks. arXiv preprint arXiv:1710.10903
7. Chai Z, You S, Yang Y, Pu S, Xu J, Cai H, Jiang W (2022) Can abnormality be detected by graph neural networks. In: Proceedings of the Twenty-Ninth international joint conference on artificial intelligence (IJCAI), Vienna, Austria, pp 23–29
8. Tang J, Li J, Gao Z, Li J (2022) Rethinking graph neural networks for anomaly detection. In: International conference on machine learning, pp 21076–21089. PMLR
9. Liu Y, Ao X, Qin Z, Chi J, Feng J, Yang H, He Q (2021) Pick and choose: a GNN-based imbalanced learning approach for fraud detection. In: Proceedings of the web conference 2021, pp 3168–3177
10. Huang C, Li Y, Loy CC, Tang X (2016) Learning deep representation for imbalanced classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5375–5384
11. Zhao T, Zhang X, Wang S (2021) Graphsmote: imbalanced node classification on graphs with graph neural networks. In: Proceedings of the 14th ACM international conference on web search and data mining, pp 833–841
12. Wei C, Sohn K, Mellina C, Yuille A, Yang F (2021) Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10857–10866
13. Liu F, Ma X, Wu J, Yang J, Xue S, Beheshti A, Zhou C, Peng H, Sheng QZ, Aggarwal CC (2022) Dagad: Data augmentation for graph anomaly detection. In: 2022 IEEE international conference on data mining (ICDM), pp 259–268. IEEE
14. Perozzi B, Akoglu L (2016) Scalable anomaly ranking of attributed neighborhoods. In: Proceedings of the 2016 SIAM international conference on data mining, pp 207–215. SIAM
15. Liu Y, Li Z, Pan S, Gong C, Zhou C, Karypis G (2021) Anomaly detection on attributed networks via contrastive self-supervised learning. IEEE transactions on neural networks and learning systems
16. Hamilton W, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. Advances in neural information processing systems **30**
17. Li J-H, Huang L, Wang C-D, Huang D, Lai J-H, Chen P (2021) Attributed network embedding with micro-meso structure. ACM Trans Knowl Discovery Data (TKDD) 15(4):1–26
18. Liu Z, Huang C, Yu Y, Dong J (2021) Motif-preserving dynamic attributed network embedding. In: Proceedings of the web conference 2021, pp 1629–1638

19. Chen J, Zhong M, Li J, Wang D, Qian T, Tu H (2021) Effective deep attributed network representation learning with topology adapted smoothing. IEEE Transactions on Cybernetics 52(7):5935–5946
20. Kolesnikov A, Zhai X, Beyer L (2019) Revisiting self-supervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1920–1929
21. Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R (2019) Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942
22. Xia J, Wu L, Chen J, Hu B, Li SZ (2022) Simgrace: A simple framework for graph contrastive learning without data augmentation. In: Proceedings of the ACM web conference 2022, pp 1070–1079
23. You Y, Chen T, Wang Z, Shen Y (2020) When does self-supervision help graph convolutional networks? In: International conference on machine learning, pp 10871–10880. PMLR
24. Zhu P, Li J, Wang Y, Xiao B, Zhao S, Hu Q (2022) Collaborative decision-reinforced self-supervision for attributed graph clustering. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2022.3171583
25. Li J, Dani H, Hu X, Liu H (2017) Radar: Residual analysis for anomaly detection in attributed networks. In: IJCAI, pp 2152–2158
26. Peng Z, Luo M, Li J, Liu H, Zheng Q (2018) Anomalous: a joint modeling approach for anomaly detection on attributed networks. In: IJCAI, pp 3513–3519
27. Velickovic P, Fedus W, Hamilton WL, Liò P, Bengio Y, Hjelm RD (2019) Deep graph infomax. ICLR (Poster) 2(3):4
28. Huang T, Pei Y, Menkovski V, Pechenizkiy M (2022) Hop-count based self-supervised anomaly detection on attributed networks. In: Joint European conference on machine learning and knowledge discovery in databases, pp 225–241. Springer
29. Zheng Y, Jin M, Liu Y, Chi L, Phan KT, Chen Y-PP (2021) Generative and contrastive self-supervised learning for graph anomaly detection. IEEE Trans Knowl Data Eng
30. Jin M, Liu Y, Zheng Y, Chi L, Li Y-F, Pan S (2021) Anemone: graph anomaly detection with multi-scale contrastive learning. In: Proceedings of the 30th ACM international conference on information & knowledge management, pp 3122–3126
31. Xu Z, Huang X, Zhao Y, Dong Y, Li J (2022) Contrastive attributed network anomaly detection with data augmentation. In: Pacific-Asia conference on knowledge discovery and data mining, pp 444–457. Springer
32. Zhang J, Wang S, Chen S (2022) Reconstruction enhanced multi-view contrastive learning for anomaly detection on attributed networks. arXiv preprint arXiv:2205.04816
33. Duan J, Wang S, Zhang P, Zhu E, Hu J, Jin H, Liu Y, Dong Z (2023) Graph anomaly detection via multi-scale contrastive learning networks with augmented view. In: Proceedings of the AAAI conference on artificial intelligence, vol 37, pp 7459–7467
34. Xia L, Huang C, Xu Y, Zhao J, Yin D, Huang J (2022) Hypergraph contrastive collaborative filtering. In: Proceedings of the 45th international ACM SIGIR conference on research and development in information Retrieval, pp 70–79
35. Lin Z, Tian C, Hou Y, Zhao WX (2022) Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In: Proceedings of the ACM web conference 2022, pp 2320–2329
36. Tong H, Faloutsos C, Pan J-Y (2006) Fast random walk with restart and its applications. In: Sixth international conference on data mining (ICDM'06), pp 613–622. IEEE
37. Oord A v d, Li Y, Vinyals O (2018) Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748
38. Wang Y, Zhang J, Guo S, Yin H, Li C, Chen H (2021) Decoupling representation learning and classification for gnn-based anomaly detection. In: Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval, pp 1239–1248
39. You Y, Chen T, Sui Y, Chen T, Wang Z, Shen Y (2020) Graph contrastive learning with augmentations. Adv Neural Inf Process Syst 33:5812–5823
40. Hassani K, Khasahmadi AH (2020) Contrastive multi-view representation learning on graphs. In: International conference on machine learning, pp 4116–4126. PMLR
41. Zhou B, Cui Q, Wei X-S, Chen Z-M (2020) Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9719–9728
42. Wang P, Han K, Wei X-S, Zhang L, Wang L (2021) Contrastive learning based hybrid networks for long-tailed image classification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 943–952
43. Sen P, Namata G, Bilgic M, Getoor L, Galligher B, Eliassi-Rad T (2008) Collective classification in network data. AI Mag 29(3):93–93
44. Tang J, Zhang J, Yao L, Li J, Zhang L, Su Z (2008) Arnetminer: extraction and mining of academic social networks. In: Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining, pp 990–998
45. Tang L, Liu H (2009) Relational learning via latent social dimensions. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, pp 817–826