Rethinking graph anomaly detection: A self-supervised Group Discrimination paradigm with Structure-Aware

Junyi Yan †
College of Software
Xinjiang University
Xinjiang, China
yjy@stu.xju.edu.cn

Enguang Zuo †
College of Information Science and Engineering
Xinjiang University
Xinjiang, China
zeg@stu.xju.edu.cn

Chen Chen
College of Information Science and Engineering
Xinjiang University
Xinjiang, China
chen_chen@stu.xju.edu.cn

Cheng Chen
College of Software
Xinjiang University
Xinjiang, China
chenchengoptics@gmail.com

Jie Zhong
College of Software
Xinjiang University
Xinjiang, China
zhongjiehk@163.com

Tianle Li
College of Information Science and Engineering
Xinjiang University
Xinjiang, China
107552103632@stu.xju.edu.cn

Xiaoyi Lv *
College of Software
Xinjiang University
Xinjiang, China
xjuwawj01@163.com

***Abstract*—Structural anomalies are the core problem in graph anomaly detection. However, the current mainstream self-supervised graph anomaly detection models do not directly model structural anomalies and their expensive time consumption limits the efficiency of graph anomaly detection. For this reason, we rethink graph anomaly detection and propose a self-supervised Group Discrimination paradigm with Structure-Aware (GDSA). Our model can be explicitly aware of the graph topology changes by multi-view structure disturbance. Moreover, GDSA transforms graph anomaly detection into discriminating the scalar summaries of positive and negative group nodes. The results of extensive experiments on four benchmark datasets show that GDSA outperforms current state-of-the-art methods, with the most significant AUC performance improvement of 28.7%. Notably, in scalability testing on a large-scale dataset, the training time and testing time of GDSA are 1181.0× and 5064.7× faster than the baseline, respectively, with 61.9% savings in memory usage.**

***Index Terms*—Graph anomaly detection, Self-supervised learning, Group discrimination, Structure-aware**

I. Introduction

Graph anomaly detection has received increasing attention in multimedia security and data mining. It has an essential impact on security-related applications such as fraud detection, spam fraud and financial transactions. Structural anomalies are a common problem in graph anomaly detection and are used to describe anomalous topological relationships between nodes, as shown in Fig.1. Incomplete links lead to the existence of abnormal nodes because of missing links between nodes as they should be, and wrong links lead to the existence of abnormal nodes because of the addition of improper inter-node links. It has been shown that the performance of graph neural networks (GNNs) can fluctuate substantially due to imperceptible structural anomalies [1], and inadequate structure-aware will affect the performance of graph anomaly detection. Therefore, it is essential to establish the robust structure disturbance mechanism to capture the complex structural information in the graph to accomplish anomaly detection. Contrastive self-supervised learning is widely used for graph anomaly detection with its excellent performance. It constructs positive and negative instance pairs in different ways, then maximizes the similarity between positive instance pairs and minimizes the similarity between negative instance pairs to learn feature representations, and finally computes the anomaly score for each node using contrastive learning [2]. However, this approach still has the following challenges: (1) Insufficient detection effect. It does not directly model the anomaly structure, making it difficult to be efficiently aware of the potential information in complex structural anomalies.

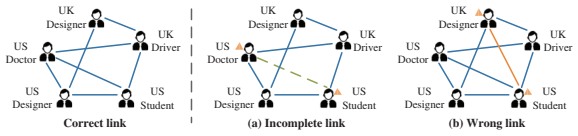


Fig. 1. Toy examples to describe the structural anomalies in graph topology using social networks. In subgraph (a), the relationship link between two American engineers is missing, resulting in the incomplete graph structure. In subgraph (b), the relationship link between an American engineer and a British manager is wrongly added, resulting in the wrong graph structure.

This work was supported by the Major Science and Technology Projects of Xinjiang Uygur Autonomous Region (2020A03001, 2020A03001-3, 2020A03001-1).

Code is publicly available at https://github.com/zeg-datamining/GDSA.

†Equal contribution

*Corresponding author

1945-788X/23/$31.00 ©2023 IEEE
DOI 10.1109/ICME55011.2023.00465

This motivates us to think about improving graph anomaly detection performance and efficiency by directly modeling the graph topology. (2) Inefficient calculation. The calculation of contrastive loss relies on the similarity calculation of nodes. The time complexity of a node loss calculation is at least $O(M)$, where M denotes the embedding dimension.

To alleviate the above challenges, we innovatively propose an end-to-end self-supervised **G**roup **D**iscrimination paradigm with **S**tructure-**A**ware (GDSA) for graph anomaly detection. Specifically, to address challenge (1), we designed the multi-view structure disturbance mechanism to model the graph topology's structural information explicitly. *View 1* performs a row transformation of the attribute matrix without changing the adjacency matrix, indirectly modeling incomplete and wrong links in the anomalous structure. In addition, *View 2* fully models the different wrong links directly by changing the number of elements within the adjacency matrix. To address challenge (2), we transform graph anomaly detection into the group discrimination problem, where anomaly detection is accomplished by discriminating the scalar summaries of positive and negative group nodes. In this session, firstly, we use numerical multiplication between the nodes' corresponding scalars and the self-supervised learning labels (instead of vector multiplication between node pairs) to complete the model training, significantly reducing the computational burden of forward and backward propagation. Secondly, we directly use a binary cross entropy (BCE) loss to distinguish the scalar information of two group nodes, and the computational time complexity of the loss for each node is only $O(1)$.

In general, the contributions of our work can be summarized as follows: (1) An efficient self-supervised group discrimination model for graph anomaly detection, GDSA, is proposed by us. To the best of our knowledge, this is the first time that graph anomaly detection has been transformed into the problem of distinguishing the scalar information of positive and negative group nodes. (2) A new graph anomaly detection approach is proposed, multi-view structure disturbance, which simulates various structural anomalies from multiple views, enabling the model to be effectively aware of the topological information between nodes, thus directly modeling the graph structure to complete the graph anomaly detection efficiently. (3) The experimental results on four benchmark datasets show that GDSA achieves state-of-the-art (SOTA) performance. The most prominent AUC value is improved by 28.7% over the suboptimal model. Notably, in scalability testing on a large-scale dataset, with 9.7% AUC performance improvement, GDSA is 1181.0× and 5064.7× faster than the baseline in training time and testing time, respectively, and achieves 61.9% savings in memory usage.

II. Related work

A. Contrastive self-supervised learning

Contrastive learning is an essential component of self-supervised learning. In graph representation learning, contrastive self-supervised learning shows competitive performance. DGI [3] accomplishes learning by maximizing the mutual information between the global graph representation and the local patch representation, which is the first attempt of GNNs to adapt contrastive learning. Based on this, GMI [4] explores a more comprehensive node representation by combining edge-level and node-level contrastive learning. MVGRL [5] learns node and graph level representations by comparing the structural views of the graphs. However, these efforts require computing similarity between two nodes, resulting in the time complexity of at least $O(M)$. In contrast, we use group discrimination to compute the scalar information of nodes directly.

B. Graph anomaly detection

Graph anomaly detection identifies abnormal nodes that are different from the majority of nodes in the graph by focusing on their attribute and topological structure information. AdONE [6] uses unsupervised adversarial learning to learn network embeddings. To avoid excessive network smoothing, ResGCN [7] exploits an attention-based deep residual modeling approach for anomaly detection. CoLA [8] effectively captures the relationship between the node and its neighboring substructures by a type of contrastive instance pair, which is the first application of contrastive self-supervised learning to graph anomaly detection. Based on this, ANEMONE [9] uses a multi-scale contrastive learning approach to capture anomalies from different perspectives. Sub-CR [10] adopts the joint optimization of a multi-view contrastive learning-based module and an attribute reconstruction-based module to complete anomaly detection. Although these algorithms perform the graph anomaly detection task differently, they do not directly target the anomaly structure modeling. In contrast, we design a multi-view structure disturbance mechanism that makes the model aware of structural anomalies.

III. Proposed method

A. Preliminary

In this paper, we focus on the graph anomaly detection problem. For the graph $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ with node set $\mathcal{V} = \{v_1, \ldots, v_m\}$, where $\mathbf{A} \in \mathbb{R}^{m \times m}$ denotes the two-dimensional adjacency matrix, $\mathbf{X} \in \mathbb{R}^{m \times n}$ denotes the attribute matrix, and n is the attribute dimension of the node v_i. Using the above symbolic representation, we define graph anomaly detection as follows.

Definition 1 (Graph anomaly detection). Given a graph $\mathcal{G} = (\mathbf{A}, \mathbf{X})$, the goal is to learn a function $f(\cdot)$: $f(\mathbf{A}, \mathbf{X}) \rightarrow s_i$, where $s_i \in \mathbb{R}^{1 \times 1}$. Specifically, under self-supervised learning, $\mathcal{G}$ is taken as the input, and the output is a vector $\mathbf{s}$ consisting of the anomaly values of all nodes. The i-th element s_i in $\mathbf{s}$ reflects the abnormality of node v_i, which is positive when v_i is an abnormal node and negative when v_i is a normal node. The relevant theorem in this paper is as follows, as discussed in Section IV-D and derived in *Supplemental material I*.

Theorem 1. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ containing structural anomalies, where the node set $\mathcal{V} = \{v_1, \ldots, v_m\}$ and the edge set $\mathcal{E} = \{e_1, \ldots, e_t\}$, the model performs a random sampling with put-back of edges in $\mathcal{E}$, taking k edges per round, for

a total of r rounds of sampling. After r rounds of sampling, the relationship between the number of edges k sampled by the model in each round and the expectation $E(m)$ of the proportion of different nodes sampled in total in r rounds as: $E(m) = 1 - \left(\frac{t-k}{t}\right)^r$.

B. Proposed model

In this paper, we propose a model called GDSA for detecting graph structure anomalies. The framework of GDSA is shown in Fig.2. Given a graph $\mathcal{G} = (\mathbf{A}, \mathbf{X})$, For l-th epoch training, firstly, we generate the attribute matrix $\mathbf{X}_{Al}$ of the positive group based on the $\mathbf{X}$ of $\mathcal{G}$. Moreover, through the multi-view structure disturbance mechanism, we generate the attribute matrix $\mathbf{X}_{Cl}$ of the negative group based on $\mathbf{X}_{Al}$ (by *view 1*) and the adjacency matrix $\mathbf{A}_l$ based on the $\mathbf{A}$ of $\mathcal{G}$ (by *view 2*). After that, $\mathbf{A}_l$ is combined with $\mathbf{X}_{Al}$ and $\mathbf{X}_{Cl}$ as the positive and negative group input to the Encoder-Readout module for feature extraction and obtaining the anomaly values s_i for two group nodes. Finally, anomaly detection is completed by discriminating s_i.

Feature augmentation. The purpose of this technique is to help the model effectively exploit the contextual information of $\mathcal{G}$ and avoid homogenization of training. Inspired by [11], we obtain the positive group attribute matrix $\mathbf{X}_{Al}$ by feature augmentation of the attribute matrix $\mathbf{X}$ in $\mathcal{G}$. Specifically, randomly mask (set to 0) the predefined proportion of w attributes of the nodes. Considering the validity of the operation, the w in GDSA is set to 0.2. This operation is clearly different from augmentation in contrastive learning to facilitate the establishment of contrast, which increases the difficulty of self-supervised model training in each epoch.

Structure disturbance mechanism. We perform the multi-view structure disturbance mechanism to model the structural anomaly adequately. Notably, to increase the robustness of the model, the structure disturbance mechanism performs r rounds of training, with r denoting the epoch value. After each round disturbance, the next round of reset disturbance will be performed. For l-th epoch training, to address the structural anomalies in Fig.1, the *view 1* operation makes structure disturbance without changing the adjacency matrix, indirectly simulating the incomplete and wrong links. Specifically, Inspired by [3], we perform a randomized row transformation based on the attribute matrix $\mathbf{X}_{Al}$ of positive group obtain the attribute matrix $\mathbf{X}_{Cl}$ of negative group. This operation helps the model avoid overfitting. To fully simulate the structural anomalies in Fig.1 (b), the *view 2* operation directly simulates the wrong links by changing the number of edges. Specifically, the adjacency matrix $\mathbf{A}_l$ is generated by performing a disturbance with randomly added edges to the original adjacency matrix $\mathbf{A}$. We add edges to $\mathbf{A}$ by randomly selecting a nodes per epoch and making them fully connected, repeating this process b times, and we denote the total number of disturbed edges per epoch time as k. For different datasets, we determined the number of edges k for each round of structure disturbance based on the parameter analysis in Section IV-D. This *view 2* operation increases the richness of structure disturbance. The multi-view structure disturbance mechanism indirectly and directly simulate structural anomalies from different views so that the model effectively perceives the anomalous links between nodes in the graph. Moreover, it significantly widens the information difference between the positive and negative group, so that the model can extract more distinguishable feature information of two groups.

Encoder-Readout module. The module design aims to extract the features of the positive and negative group and convert them into scalar information of nodes for group discrimination. Specifically, for l-th epoch training, the Encoder is used to extract spatial features in the graph, and various GNNs can be selected, which we set to GCN [12] as follows:

$$\mathbf{E}_l = GCN\left(\mathbf{A}_l, \mathbf{X}_l\right) \tag{1}$$

where $\mathbf{X}_l \in \mathbb{R}^{m \times n}$ represents the attribute matrix input to the Encoder, $\mathbf{E}_l \in \mathbb{R}^{m \times h}$ represents the embeddings output of the Encoder, and h is the hidden size of $\mathbf{E}_l$. Specifically, we combine $\mathbf{A}_l \in \mathbb{R}^{m \times m}$ with $\mathbf{X}_{Al}$ and $\mathbf{X}_{Cl}$ as a positive and negative group into the Encoder, and its outputs are $\mathbf{E}_{Al}$ and $\mathbf{E}_{Cl}$, which are the embeddings of positive and negative group. The Readout component reduces the dimensionality of the positive and negative group embeddings, which can choose various dimensionality reduction methods. Considering the actual efficiency, we use a custom MLP:

$$\begin{cases} \mathbf{e}_l = \sigma\left(\mathbf{E}_l\right) \\ s_i = MLP\left(\mathbf{E}_l\right) = \sum_{j=1}^{h} \mathbf{e}_l[i,j] \end{cases} \tag{2}$$

where $\sigma(\cdot)$ is a linear function that processes $\mathbf{E}_l$ as feature information $\mathbf{e}_l \in \mathbb{R}^{m \times h}$. $\mathbf{e}_l[i,j]$ represents the j-th attribute information of the node v_i in the $\mathbf{e}_l$. The scalar information $s_i \in \mathbb{R}^{1 \times 1}$ of the positive and negative group nodes is calculated by summing the attribute information of the $\mathbf{e}_l$ of the node v_i. We enter the $\mathbf{E}_{Al}$ and $\mathbf{E}_{Cl}$ into this component, respectively, to get the s_i of the nodes in the positive and negative group.

Group discrimination. Inspired by the fact that the group discrimination approach effectively reduces the time cost in graph representation learning without affecting the performance [13], we use simple BCE loss to discriminate the scalar information of two group nodes as follows:

$$\mathcal{L}_{BCE} = \frac{1}{2m}\left(\sum_{i=1}^{2m} y_i \log s_i + (1 - y_i)\log\left(1 - s_i\right)\right) \tag{3}$$

where $y_i \in \mathbb{R}^{1 \times 1}$ of each node of the positive and negative group corresponds to 0 and 1, and the total number of nodes is $2m$. When GDSA is trained, the model is optimized by correctly identifying whether the node v_i is abnormal according to the s_i. In contrast, we replace the vector multiplication operation in contrastive learning with scalar computation between s_i and y_i, and the loss calculation of node v_i is only $O(1)$, effectively reducing the time and memory consumption in graph anomaly detection. In an ideal state, the s_i of abnormal nodes behaves positively, and the s_i of normal nodes behaves

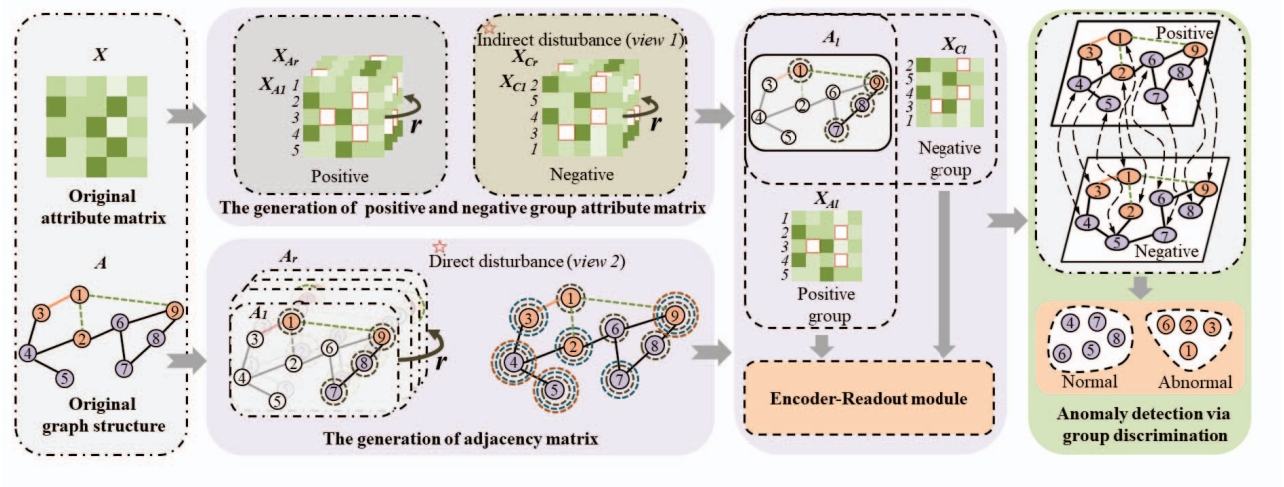


Fig. 2. The overall framework of GDSA. For l-th epoch training, Firstly, the generation of positive and negative group attribute matrix (performing indirect disturbance) and the generation of adjacency matrix (performing direct disturbance) give the positive and negative group information (positive group: $\mathbf{X}_{Al}$ and $\mathbf{A}_l$, negative group: $\mathbf{X}_{Cl}$ and $\mathbf{A}_l$). After that, the information of the two groups is input to the Encoder-Readout module to obtain the anomaly value s_i of positive and negative group nodes, respectively. Finally, anomaly detection is completed via group discrimination.

negatively, so s_i can directly distinguish abnormal nodes from normal nodes. Compared with using anomaly score for anomaly detection, this is simpler and more efficient, and the anomaly detection task can be done directly by discriminating the s_i obtained by the model. To get the anomaly scores of nodes, a sigmoid function can be used to process the s_i of positive and negative group nodes as follows:

$$c_i = \frac{1}{(1 + e^{-s_i})} \tag{4}$$

In an ideal state, the c_i of the abnormal node is close to 1 and the c_i of the normal node is close to 0.

The overall procedures and time complexity analysis of GDSA are summarized in *Supplemental material II* and *Supplemental material III*, respectively.

IV. Experiments

A. Experimental settings

Datasets. We evaluate the GDSA model on five datasets of different sizes, which include four benchmark datasets, two of which are citation network datasets (Cora and Citeseer [14]), two of which are social network datasets (BlogCatalog and Flickr [15]), and one large-scale dataset (ogbn-arxiv) provided by Open Graph Benchmark [16]. The total number of anomalies and statistics of these datasets are summarized in *Supplemental material IV*.

Implementation details. To ensure the reproducibility of the experiments, the parameter settings and computing infrastructure of the experiments are in *Supplemental material V*. In addition, we choose the evaluation metric AUC, which is used in most graph anomaly detection tasks, to measure the comprehensive performance of the model. Considering that the purpose of anomaly detection is to detect anomaly instances that are different from the majority of instances (in the graph, instances generally refer to nodes), we use F1 score to evaluate the precision of the model.

B. Evaluation on the benchmark datasets

On four benchmark datasets, we compare the GDSA model with six unsupervised SOTA models for graph anomaly detection, where CoLA [8], ANEMONE [9] and SL-GAD [17] are SOTA methods for graph anomaly detection based on contrastive learning. In addition, AdONE [6] is a deep autoencoder-based anomaly detection model, GANN [18] is a generative adversarial-based anomaly detection model, and ResGCN [7] accomplishes anomaly detection through attention-based deep residuals.

Tabel I shows the graph anomaly detection performance (i.e., AUC values) of the GDSA model with six baseline models in four benchmark datasets in the unsupervised case. The corresponding ROC curves are shown in Fig. 3 in turn. From Tabel I and Fig.3, we have the following observations: (1) Our proposed model GDSA outperforms recent graph anomaly detection models on the four datasets, which shows the efficiency of the multi-view structure disturbance mechanism to solve the graph anomaly detection problem by modeling the structure. In particular, GDSA achieves the AUC of 1.0000 on the Cora and Citeseer datasets, with the p values < 0.05 (highly significant) for the t-test with the suboptimal model, indicating that GDSA performance is also statistically significant in terms of advantages. (2) The existence of suboptimal performance of the three baseline models on four different datasets indicates that the current mainstream graph anomaly detection models have poor generalization capabilities, although they have great potential for anomaly detection. In contrast, our model shows optimal performance in all four datasets and significantly improves the AUC by 28.7% (0.1826) over the suboptimal model (CoLA [8]) on the Flickr dataset. (3) Compared with the baseline models, GDSA

TABLE I

AUC VALUES ON FOUR BENCHMARK DATASETS (ALL MODELS RUN THREE TIMES WITH RANDOM INITIALIZATIONS AND REPORT THE MEAN RESULTS). THE BEST PERFORMANCE IS HIGHLIGHTED IN **BOLD**. THE SECOND-BEST PERFORMANCE IS UNDERLINED. * MARKS THE PERFORMANCE OF T-TEST WITH p VALUES < 0.05 BETWEEN GDSA AND THE SUBOPTIMAL MODEL.

Methods	Cora	Citeseer	BlogCatalog	Flickr
CoLA [8] (2021)	0.9338	0.9055	<u>0.6804</u>	<u>0.6365</u>
ANEMONE [9] (2021)	0.9706	0.9655	0.6681	0.6180
SL-GAD [17] (2021)	0.9035	0.9127	0.6477	0.6144
AdONE [6] (2020)	0.9525	<u>0.9922</u>	0.6144	0.3754
GANN [18](2020)	<u>0.9841</u>	0.9851	0.6051	0.6324
ResGCN [7] (2021)	0.6117	0.5135	0.6083	0.6113
GDSA	**1.0000***	**1.0000***	**0.7163**	**0.8191**

based on the multi-view structure disturbance mechanism fully considers different types of structural anomalies in the graph and directly models the structural anomalies in the graph, which is the main reason why GDSA can perform optimally in various types of datasets for detection performance.

Fig.4 shows the time consumption of the GDSA model compared with six mainstream graph anomaly detection models on four datasets, from which we can find that the GDSA model presents an exponential efficiency improvement while ensuring optimal detection performance on the four datasets. Most obviously, on the Citeseer dataset, GDSA is 87.8× faster than the baseline SL-GAD [17]. It verifies that processing graph anomaly detection by discriminating the scalar information of positive and negative group nodes can effectively reduce time consumption. Notably, GDSA transforms the loss computation object of nodes in the graph anomaly detection task from vector information to scalar information, which directly reduces the time complexity of the loss computation part from at least $O(M)$ to $O(1)$, which is the main reason why GDSA can guarantee the optimal performance of detection efficiency on datasets of different scales.

C. Ablation study

To further investigate the effect of different view's structure disturbance mechanism on the GDSA model, we explored the graph anomaly detection performance of GDSA without structure disturbance (GDSA w/o both), GDSA without *view 1*'s structure disturbance (GDSA w/o *view 1*), GDSA without *view 2*'s structure disturbance (GDSA w/o *view 2*) and GDSA on four benchmark datasets, respectively, as shown in Tabel II. The visualization of the initial feature distribution, the feature distribution after being processed by GDSA w/o both and the feature distribution after being processed by GDSA model for the four datasets is shown in *Supplemental material VI*. By comparison, we have the following findings: (1) The multi-view structure disturbance can be fully aware of various anomalous links in the graph and exhibits optimal performance on all four datasets. Although GDSA w/o *view 2* has shown better performance, it is still second to the GDSA model, probably because it does not directly and adequately simulate different types of wrong links in the graph. (2) Compared

TABLE II

AUC AND STANDARD DEVIATION ANALYSIS OF THE EFFECTS OF DIFFERENT VIEW'S STRUCTURE DISTURBANCE (ALL EXPERIMENTS RUN THREE TIMES WITH RANDOM INITIALIZATIONS AND REPORT THE MEAN RESULTS). THE BEST PERFORMANCE IS IN **BOLD**.

	Cora	Citeseer	BlogCatalog	Flickr
GDSA w/o both	0.0528±0.0069	0.2410±0.0734	0.6303±0.0024	0.7685±0.0176
GDSA w/o *view 1*	0.0550±0.0070	0.3004±0.1339	0.6325±0.0028	0.7739±0.0084
GDSA w/o *view 2*	**1.0000**±0.0000	**1.0000**±0.0000	0.7146±0.1343	0.8015±0.0569
GDSA	**1.0000**±0.0000	**1.0000**±0.0000	**0.7163**±0.0069	**0.8191**±0.0375

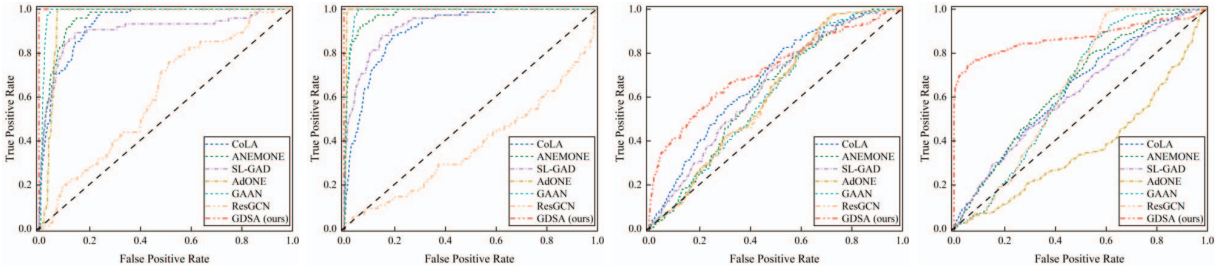
with the two social network datasets (BlogCatalog and Flickr), GDSA with the multi-view structure disturbance is more effective on two citation networks (Cora and Citeseer). The possible reason is that the average degree of the citation network nodes (mean degree = 2.51) is much smaller than that of the social network nodes (mean degree = 32.35). Therefore, their topological relationships are relatively simple, making it less difficult to be aware of the interrelationships among their nodes.

D. Parameter analysis

According to Theorem 1, we determined the number of edges k for each round of structure disturbance for different datasets. The derivation of this theorem is shown in *Supplemental material I*. To more intuitively observe the optimality of k selection, for four benchmark datasets, we visualized the relationship between the number of edges sampled k in each round, the expectation $E(m)$ of different node proportions perceived by r-round training and the final F1 score of the model. The visualization of the Cora dataset is shown in Fig. 5, and the visualization of the other datasets is shown in *Supplemental material VII*. We can find that as the k value increases, $E(m)$ gradually tends to 1, and the F1 value shows a saturation state or even a certain degree of decrease after the gradual increase, which indicates that it is not the case that the larger the k value and the closer $E(m)$ is to 1, the better structure disturbance of the model. Integrating efficiency and performance, we set the k value that makes $E(m)$ reach near 0.99 in the datasets as the final number of edges for each round of structure disturbance.

E. Scalability testing

To validate the efficiency of GDSA, we conducted scalability testing on the large-scale dataset ogbn-arxiv. We compared the GDSA model with the baseline CoLA [8], as shown in Table III. Combining the CoLA paper elaboration with its convergence degree performance on this dataset, we set the epoch of CoLA to 2000. From Table III, we can find that using only one epoch, the AUC value of the GDSA model is improved by 9.7% (0.0805) over that of CoLA with 2000 epoch training, with the memory usage saving of 61.9% and the significant improvement of 1181.0× and 5064.7× in training time and testing time, respectively. This indicates the GDSA model can handle large-scale datasets quickly and efficiently. In addition, when the epoch is set to 100, the AUC value of the GDSA model improves by 19.0% (0.1576)



(a) Cora (b) Citeseer (c) BlogCatalog (d) Flickr

Fig. 3. ROC curves comparison for GDSA and six baselines on four benchmark datasets.

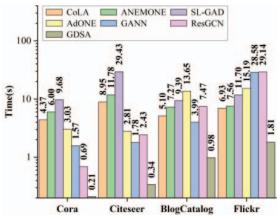


Fig. 4. Comparison of total time consumption on four datasets.

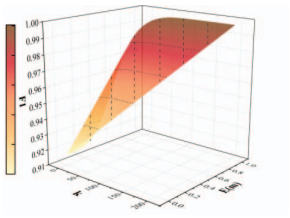


Fig. 5. The relationship between k, $E(m)$ and F1 on the Cora dataset.

TABLE III
COMPARISON OF ANOMALY DETECTION PERFORMANCE AND EFFICIENCY ON THE OGBN-ARXIV DATASET. 'EPO' REFERS TO EPOCH.

Methods	AUC	Memory usage	Training time	Testing time
CoLA (2000 epo)	0.8291	8961MB	78h24m25s	7h18m43s
GDSA (1 epo)	0.9096	3414MB|61.9%	3m59s|1181.0×	5.1973s|5064.7×
GDSA (100 epo)	0.9867	4850MB|45.9%	8h11m32s|9.6×	5.1973s|5064.7×

over the baseline, which further shows the potential of GDSA model for graph anomaly detection on large-scale datasets.

V. CONCLUSION

This paper proposes a novel self-supervised paradigm for graph anomaly detection called GDSA. Firstly, it efficiently models the graph topology based on the multi-view structure disturbance mechanism to be fully aware of the nodes' interrelationships. Moreover, it transforms the graph anomaly detection task into the problem of discriminating scalar information of positive and negative group nodes, reducing the time complexity of inter-node loss computation directly to $O(1)$. Experimental results on four benchmark datasets and scalability testing on a large-scale dataset show that the AUC performance, memory consumption and time consumption of GDSA in solving graph anomaly detection problems on different types and different sizes of datasets are significantly better than that of the current SOTA models. Our work verifies that the structure disturbance approach does have a significant impact on graph anomaly mining, while the group discrimination approach can significantly reduce the ability of computational redundancy in graph anomaly detection. GDSA provides a simple and efficient new idea for graph anomaly detection based on self-supervised learning, which will have promising applications for detecting anomalies in online social network interactions, financial fraud and academic citations.

REFERENCES

[1] Wei Jin, Yaxing Li, Han Xu, Yiqi Wang, Shuiwang Ji, Charu Aggarwal, and Jiliang Tang, "Adversarial attacks and defenses on graphs," in *SIGKDD*, 2021.

[2] Jing Ren, Feng Xia, Ivan Lee, Azadeh Noori Hoshyar, and Charu C Aggarwal, "Graph learning for anomaly analytics: Algorithms, applications, and challenges," *TIST*, 14(2): 1-29, 2022.

[3] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm, "Deep graph infomax," in *ICLR*, 2019.

[4] Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang, "Graph representation learning via graphical mutual information maximization," in *WWW*, 2020.

[5] Kaveh Hassani and Amir Hosein Khasahmadi, "Contrastive multi-view representation learning on graphs," in *ICML*, 2020.

[6] Sambaran Bandyopadhyay, Lokesh N, Saley Vishal Vivek, and M. N. Murty, "Outlier resistant unsupervised deep architectures for attributed network embedding," in *WSDM*, 2020.

[7] Yulong Pei, Tianjin Huang, Werner van Ipenburg, and Mykola Pechenizkiy, "Resgcn: Attention-based deep residual modeling for anomaly detection on attributed networks," in *DSAA*, 2021.

[8] Yixin Liu, Zhao Li, Shirui Pan, Chen Gong, Chuan Zhou, and George Karypis, "Anomaly detection on attributed networks via contrastive self-supervised learning," *TNNLS*, 33(6): 2378-2392, 2021.

[9] Ming Jin, Yixin Liu, Yu Zheng, Lianhua Chi, Yuan-Fang Li, and Shirui Pan, "Anemone: graph anomaly detection with multi-scale contrastive learning," in *CIKM*, 2021.

[10] Jiaqiang Zhang, Senzhang Wang, and Songcan Chen, "Reconstruction enhanced multi-view contrastive learning for anomaly detection on attributed networks," in *IJCAI*, 2022.

[11] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen, "Graph contrastive learning with augmentations," in *NeurIPS*, 2020.

[12] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[13] Yizhen Zheng, Shirui Pan, Vincent Cs Lee, Yu Zheng, and Philip S Yu, "Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination," in *NeurIPS*, 2022.

[14] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad, "Collective classification in network data," *AI Magazine*, 29(3): 93–93, 2008.

[15] Lei Tang and Huan Liu, "Relational learning via latent social dimensions," in *SIGKDD*, 2009.

[16] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec, "Open graph benchmark: Datasets for machine learning on graphs," in *NeurIPS*, 2020.

[17] Yu Zheng, Ming Jin, Yixin Liu, Lianhua Chi, Khoa T Phan, and Yi-Ping Phoebe Chen, "Generative and contrastive self- supervised learning for graph anomaly detection," in *TKDE*, 2021.

[18] Zhenxing Chen, Bo Liu, Meiqing Wang, Peng Dai, Jun Lv, and Liefeng Bo, "Generative adversarial attributed network anomaly detection," in *CIKM*, 2020.