

# Dynamic Spectral Graph Anomaly Detection

Jianbo Zheng<sup>1,2</sup>, Chao Yang<sup>1\*</sup>, Tairui Zhang<sup>1</sup>, Longbing Cao<sup>2\*</sup>, Bin Jiang<sup>1</sup>, Xuhui Fan<sup>2</sup>, Xiao-ming Wu<sup>2,3</sup>, Xianxun Zhu<sup>2,4</sup>

<sup>1</sup>College of Computer Science and Electronic Engineering, Hunan University, China

<sup>2</sup>School of Computing, Macquarie University, Australia

<sup>3</sup>School of Computer Science and Engineering, Sun Yat-sen University, China

<sup>4</sup>School of Communication and Information Engineering, Shanghai University

zhengjianbo@hnu.edu.cn, yangchaoedu@hnu.edu.cn, zhangtairui@hnu.edu.cn, longbing.cao@mq.edu.au, jiangbin@hnu.edu.cn, xuhui.fan@mq.edu.au, wuxm65@mail2.sysu.edu.cn, zhuxianxun@shu.edu.cn

## Abstract

Graph anomaly detection is crucial for identifying anomalous nodes within graphs and addressing applications like financial fraud detection and social spam detection. Recent spectral graph neural networks advance graph anomaly detection by focusing on anomalies that notably affect the distribution of graph spectral energy. Such spectrum-based methods rely on two steps: graph wavelet extraction, and feature fusion. However, both steps are hand-designed, capturing incomprehensive anomaly information of wavelet-specific features and resulting in their inconsistent feature fusion. To address these problems, we propose a dynamic spectral graph anomaly detection framework DSGAD to adaptively capture comprehensive anomaly information and perform consistent feature fusion. DSGAD introduces dynamic wavelets, consisting of trainable wavelets to adaptively learn anomalous patterns and capture wavelet-specific features with comprehensive anomaly information. Furthermore, the consistent fusion of wavelet-specific features achieves dynamic fusion by combining wavelet-specific feature extraction with energy difference and channel convolution fusion using location correlation. Experimental results on four datasets substantiate the efficacy of our DSGAD method, surpassing state-of-the-art methods on both homogeneous and heterogeneous graphs.

**Code** — <https://github.com/IWantBe/Dynamic-Spectral-Graph-Anomaly-Detection>

## 1 Introduction

Graph anomaly detection (GAD) is crucial for identifying anomalous nodes within graphs (Pang et al. 2021; Wang et al. 2023a; He et al. 2024; Kumagai, Iwata, and Fujiwara 2021) and finds applications in diverse fields such as network security (Ten, Hong, and Liu 2011), social spam detection (Yu, He, and Liu 2015), and financial fraud detection (Han et al. 2022; Chai et al. 2023). Recent studies have revealed that, as the proportion of anomalous nodes in a graph increases, there is an observable shift of spectral energy from low to high frequencies, which is termed as a “right-shift” (Tang et al. 2022). In response to this phenomenon, spectral graph neural networks (GNNs), also

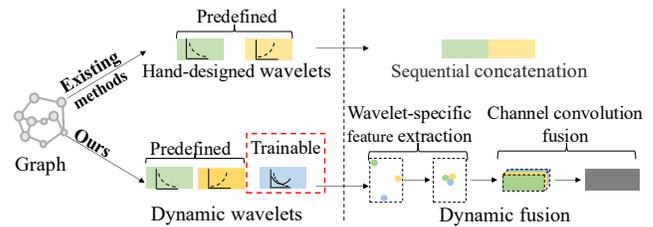


Figure 1: A framework comparison of our method with existing methods. The bottom portion illustrates our proposed method, which initially employs dynamic wavelets, followed by dynamic fusion. In contrast, the top portion depicts existing hand-designed methods, which first apply predefined wavelets, and followed by feature fusion through sequential concatenation.

known as graph wavelets/ filters, have garnered significant research attention (Chen, Ma, and Wang 2022).

Many spectral GNN methods for GAD typically involve two procedures: graph wavelet extraction and feature fusion. In graph wavelet extraction, a mother wavelet, consisting of multiple hand-designed graph wavelets, extracts wavelet-specific features across different frequency bands (Hammond, Vandergheynst, and Gribonval 2011). Each wavelet is hand-designed for a predefined frequency band, such as Beta wavelets (Tang et al. 2022) and hybrid wavelets (Xu et al. 2024). In the subsequent feature fusion step, wavelet-specific features are sequentially concatenated in a hand-designed manner to form fusion features for downstream anomaly classification (Dong, Zhang, and Wang 2023; Xu et al. 2024; Dong et al. 2024).

However, the current spectral GNN methods with two hand-designed steps exhibit several limitations. These include incomprehensive anomaly information and inconsistent feature fusion, which result in suboptimal performance. **(1) Incomprehensive anomaly information.** Anomaly information spans multiple frequency bands and varies with different datasets but does not adhere to fixed distributions like Beta or other predefined ones (De Oliveira and De Araújo 2015). This limitation arises because hand-designed wavelets are unable to capture anomalous patterns across varying datasets adaptively. **(2) Inconsistent feature fusion.** Wavelet-specific features are projections of node

\*Corresponding Author.

features in varying frequency band viewpoints (spaces), which exhibit energy differences. Inconsistent feature fusion stems from sequential concatenation without considering energy differences and location correlations among wavelet-specific features.

To address these issues, we propose a Dynamic Spectral Graph Anomaly Detection (DSGAD) framework, which consists of dynamic wavelets and dynamic fusion, as illustrated in Figure 1. **Dynamic wavelets** comprise both trainable wavelets that adaptively learn anomalous patterns within graphs and hand-designed wavelets to capture wavelet-specific features. They involve more comprehensive anomaly information compared to hand-designed wavelets alone. **Dynamic fusion** combines wavelet-specific feature extraction with energy differences and channel convolution fusion using location correlations. They ensure the consistent fusion of wavelet-specific features and obtain fusion features with higher discrimination capability. Additionally, we conduct extensive experiments across four datasets. The overall comparison results demonstrate that our DSGAD method outperforms state-of-the-art (SOTA) methods in both homogeneous and heterogeneous graph scenarios.

## 2 Preliminary

**Static attributed graphs.** A static attributed graph (Gao et al. 2023a) is usually represented as  $\mathbf{G} = \{\mathbf{V}, \mathbf{E}, \mathbf{X}\}$ , where  $\mathbf{V} = \{v_1, v_2, \dots, v_n\}$  refers to a set of  $n$  nodes,  $\mathbf{E}$  refers to the set of edges, and  $\mathbf{E}_{ij}$  represents an unweighted edge connecting nodes  $\mathbf{V}_i, \mathbf{V}_j$ , and  $\mathbf{X} \in \mathbb{R}^{n \times d}$  is a feature matrix, in which the  $i$ -th row  $\mathbf{X}_i$  denotes the  $d$ -length attribute vector characterizing node  $\mathbf{V}_i$ . For calculation convenience, we may transform  $\mathbf{E}$  into a binary adjacency matrix  $\mathbf{A} \in \{0, 1\}^{n \times n}$ , where  $\mathbf{A}_{ij} = 1$  indicates the presence of edge  $\mathbf{E}_{ij}$ , and vice versa. Further, a degree matrix  $\mathbf{D}$  is calculated from  $\mathbf{A}$  as a diagonal matrix with the  $i$ -th diagonal entry  $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$ . It is noted that the adjacency matrix  $\mathbf{A}$  and the degree matrix  $\mathbf{D}$  collectively facilitate an understanding of the connectivity and structural characteristics of graphs.

**Graph anomaly detection.** GAD is commonly studied in numerous GNN-based approaches (Tang et al. 2022; Gao et al. 2023a; Huang et al. 2022; Liu et al. 2021; Shi et al. 2022; Xu et al. 2023). In this paper, we treat all edges  $\mathbf{E}$  in graph  $\mathbf{G}$  as trusted and focus on node anomaly. The nodes within graph  $\mathbf{G}$  are categorized into labeled nodes  $\mathbf{V}_l$  and unlabeled nodes  $\mathbf{V}_u$ . Anomalies are labelled as  $\mathbf{1}$ , while normal nodes are labelled as  $\mathbf{0}$ . Let  $\mathbf{V}_l^a$  and  $\mathbf{V}_l^n$  denote two subsets of labeled nodes, where  $\mathbf{V}_l^a$  contains all anomalous nodes and  $\mathbf{V}_l^n$  contains all normal nodes, with  $\mathbf{V}_l^a \cap \mathbf{V}_l^n = \emptyset$  and  $\mathbf{V}_l^a \cup \mathbf{V}_l^n = \mathbf{V}_l$ . GAD aims to classify whether unlabeled nodes  $\mathbf{V}_u$  are anomalous based on the information provided by the labeled nodes  $\mathbf{V}_l$ , graph structure  $\mathbf{E}$ , and nodes attributes  $\mathbf{X}$ .

**Spectral decomposition.** In spectral decomposition/analysis (Balcilar et al. 2021; Gao et al. 2021), the normalized Laplacian matrix  $\mathbf{L}$  of graph  $\mathbf{G}$  is defined as  $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$ , where  $\mathbf{A}$  and  $\mathbf{D}$  are the adjacency matrix and the degree matrix, respectively. We decompose  $\mathbf{L}$

as  $\mathbf{L} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top$  via a spectral decomposition, where  $\mathbf{\Lambda}$  is a diagonal matrix  $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_n)$  with eigenvalues as the diagonal entries and  $\mathbf{U}$  is composed of corresponding eigenvectors as  $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_n]$ . These eigenvectors  $\{\mathbf{u}_i\}_{i \in [1, n]}$  can be regarded as the basis functions of graph  $\mathbf{G}$  in the frequency domain.

Further, graph features  $\mathbf{X}$  can be transformed into their frequency-domain representation  $\mathbf{X}^f$ , i.e.,  $\mathbf{X}^f = \mathbf{U}^\top \mathbf{X}$  (Dong et al. 2021), through a graph Fourier transformation. We then process  $\mathbf{X}^f$  with wavelets to extract the features within a specific frequency band, yielding the wavelet-specific representation  $\hat{\mathbf{X}}^f = g(\mathbf{\Lambda}) \mathbf{X}^f = g(\mathbf{\Lambda}) \mathbf{U}^\top \mathbf{X}$ , where  $g(\mathbf{\Lambda})$  denotes the kernel function which targets the desired frequency band. Finally, a graph inverse transformation  $\mathbf{F} = \mathbf{U} \hat{\mathbf{X}}^f = \mathbf{U} g(\mathbf{\Lambda}) \mathbf{U}^\top \mathbf{X}$  can be used to revert the wavelet-specific representation to the spatial domain, where  $\mathbf{F}$  denotes wavelet-specific features. To expedite spectral analysis, we follow previous studies (Tang et al. 2022; Xu et al. 2024) on employing the polynomial kernel function  $g_{\text{poly}}$  to directly map  $\mathbf{L}$  to a new space, resulting in wavelet-specific features  $\mathbf{F} = g^{\text{poly}}(\mathbf{L}) \mathbf{X}$ .

**Beta mother wavelet.** In graphs, anomalous nodes cause the ‘‘right-shift’’ phenomenon in the frequency domain. BetaWavelet (Tang et al. 2022) effectively leverages this phenomenon to improve GAD tasks. A Beta mother wavelet  $\mathbf{W}^\beta$  in BetaWavelet includes all the hand-designed Beta wavelets as follows:

$$\mathbf{W}^\beta = (\{\mathbf{W}_{p,q}^\beta\}_{p+q=C}) = (\mathbf{W}_{0,C}^\beta, \mathbf{W}_{1,C-1}^\beta, \dots, \mathbf{W}_{C,0}^\beta). \quad (1)$$

Here,  $\mathbf{W}_{p,q}^\beta$  denotes the Beta wavelets used as the basis functions, with  $p$  and  $q$  being the parameters of the graph wavelets ( $p + q = C$ ) and  $C$  is a constant integer.

This Beta mother wavelet captures wavelet-specific features across different frequency bands through various  $p, q$  values of  $\mathbf{W}_{p,q}^\beta$ , thus facilitating spectral analysis. Based on the probability density function of Beta distribution  $\beta_{p,q}$ , the formula for Beta wavelets  $\mathbf{W}_{p,q}^\beta$  is defined as follows:

$$\mathbf{W}_{p,q}^\beta = \mathbf{U} \frac{1}{2} \beta_{p,q} \left( \frac{\mathbf{\Lambda}}{2} \right) \mathbf{U}^\top = \frac{(\frac{\mathbf{I}}{2})^p (\mathbf{I} - \frac{\mathbf{I}}{2})^q}{2B(p+1, q+1)}, \quad (2)$$

where  $B(p+1, q+1) = \frac{p! \times q!}{(p+q+1)!}$  is a normalized constant in Beta distribution,  $\mathbf{I}$  is the identity matrix.

## 3 The DSGAD Method

The DSGAD framework is illustrated in Figure 2. DSGAD advances graph anomaly detection by capturing comprehensive anomaly information and performing consistent feature fusion. Initially, DSGAD introduces dynamic wavelets, consisting of trainable wavelets to adaptively learn anomalous patterns and capture wavelet-specific features with comprehensive anomaly information. Furthermore, the consistent fusion of wavelet-specific features achieves dynamic fusion by combining wavelet-specific feature extraction with energy difference and channel convolution fusion using location correlation. This process results in fusion features

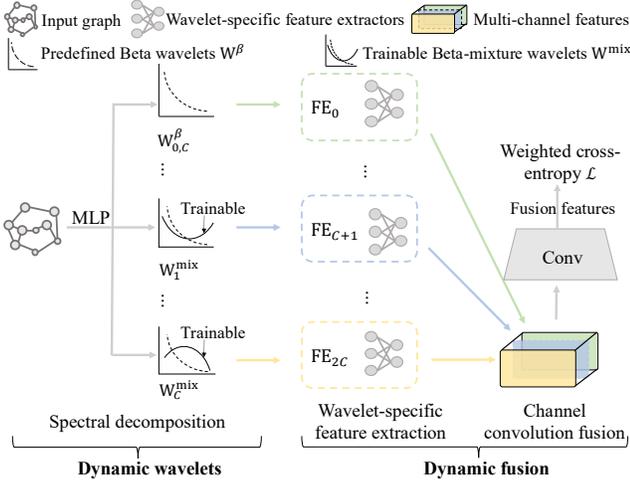


Figure 2: The overall framework of DSGAD involves dynamic wavelets and dynamic fusion. Initially, dynamic wavelets are used for spectral decomposition to extract the wavelet-specific features with comprehensive anomaly information adaptively. Subsequently, dynamic fusion combines wavelet-specific feature extraction and channel convolution fusion to integrate wavelet-specific features with energy differences and location correlations consistently, thus generating fusion features with higher discrimination capability. Finally, the fusion features are used by the weighted cross-entropy loss function  $\mathcal{L}$  for binary classification.

with higher discrimination capability. Wavelet-specific feature extraction maps wavelet-specific features from different frequency band spaces to adjacent latent spaces, while channel convolution fusion effectively integrates wavelet-specific features by leveraging their location correlations.

### 3.1 Dynamic Wavelets

Normal and anomalous nodes are anticipated to exhibit distinctive energy distribution behaviors, which may significantly diverge from predefined distributions such as the Beta distribution. Moreover, their behaviors may vary with changes in the dataset. Current hand-designed wavelets, represented explicitly by the Beta mother wavelet described in Eq. 1, overlook these characteristics, thereby failing to extract essential anomaly information.

To more comprehensively extract anomaly information, the new Beta-mixture mother wavelet  $\mathbf{W}^{\text{mix}}$  is proposed to complement the existing Beta mother wavelet  $\mathbf{W}^\beta$ . Assuming we have a Beta-mixture mother wavelet  $\mathbf{W}^{\text{mix}}$  comprising  $C$  trainable Beta-mixture wavelets to adaptively learn anomalous patterns, the  $j$ -th Beta-mixture wavelet is defined as a normalized linear sum of existing Beta wavelets  $\mathbf{W}_j^{\text{mix}} = \sum_{i=0}^C p_{j,i}^{\text{mix}} \mathbf{W}_{i,C-i}^\beta$ , where  $\sum_i p_{j,i}^{\text{mix}} = 1$ . Furthermore, the weight  $p_{j,i}^{\text{mix}}$  for the  $j$ -th Beta-mixture wavelet is calculated as  $\mathbf{p}_j^{\text{mix}} = \text{Softmax}(\sum_{k=1}^n (\theta_{j,k,:}; \mathbf{X}_{k,:}^T)) \in \mathbb{R}^{C+1}$ . Here,  $\theta \in \mathbb{R}^{C \times n \times (C+1) \times d}$  is the feature-to-weight transformation matrix, and  $\theta_{j,k,i,d}$  represents the  $j$ -th Beta-mixture wavelet for the  $k$ -th node over the  $i$ -th Beta wavelet

in the  $d$ -th attribute.

The proposed dynamic wavelets  $\mathbf{W}$  is structured as:

$$\mathbf{W} = \{\mathbf{W}^\beta, \mathbf{W}^{\text{mix}}\} = \{\{\mathbf{W}_{i,C-i}^\beta\}_i, \{\mathbf{W}_j^{\text{mix}}\}_{j \in [1,C]}\}. \quad (3)$$

The left panel of Figure 3 visually illustrates the effectiveness of dynamic wavelets when  $C$  is set to 2. It shows spectral behaviors of three hand-designed Beta wavelets and two trained Beta-mixture wavelets on the T-finance dataset, which are detailed in Section 4.1. The trained Beta-mixture wavelets exhibit significantly different behaviors compared to the hand-designed Beta wavelets, enabling them to capture anomaly information that hand-designed Beta wavelets miss. Thus, our proposed dynamic wavelets, which include both trainable Beta-mixture wavelets and hand-designed Beta wavelets, capture more comprehensive anomaly information than using hand-designed Beta wavelets alone.

Finally, based on the dynamic wavelets  $\mathbf{W}$ , we conduct spectral analysis on the original graph features  $\mathbf{X}$ , thereby extracting wavelet-specific features  $\mathbf{F}$  with comprehensive anomaly information. The process can be expressed as:

$$\mathbf{F} = \mathbf{W}(\text{MLP}(\mathbf{X})) = \{\{\mathbf{F}_{p,C-p}^\beta\}_{p \in [0,C]}, \{\mathbf{F}_j^{\text{mix}}\}_{j \in [1,C]}\}, \quad (4)$$

where dynamic wavelets  $\mathbf{W}$  is defined in Eq. 3,  $\text{MLP}(\cdot)$  denotes a simple Multi-Layer Perceptron (Taud and Mas 2018), and there are  $2C + 1$  wavelet-specific features from different frequency band spaces.

### 3.2 Dynamic Fusion

**Wavelet-specific feature extraction** We observe that wavelet-specific features  $\mathbf{F}$ , derived from dynamic wavelet processing, are projections of feature embeddings  $\text{MLP}(\mathbf{X})$  in varying frequency band spaces and exhibit energy differences. Furthermore, the spectral energy distribution is used to describe the energy distribution among wavelet-specific features quantitatively.

Given the original features  $\mathbf{X} \in \mathbb{R}^{n \times d}$ , the frequency-domain representation may be obtained through the process as  $\mathbf{X}^f = [X_1^f, \dots, X_n^f]^T = \mathbf{U}^T \mathbf{X}$ , with eigenvalues  $\{\lambda_1, \dots, \lambda_n\}$  corresponding to eigenvectors of  $\mathbf{U}$  and the rows of  $\mathbf{X}^f$  respectively. As a result, the *spectral energy distribution*  $S_E$  of  $\mathbf{X}$  at  $\lambda_k (1 \leq k \leq n)$  is defined as:

$$S_E(\lambda_k) = \frac{(X_k^f)^2}{\sum_{i=1}^n (X_i^f)^2}. \quad (5)$$

Using the T-finance dataset as an example, the spectral energy distribution of wavelet-specific features, obtained through dynamic wavelets processing, can be calculated using Eq. 5 by incorporating the eigenvalue  $\Lambda$ . As depicted in Figure 3 (middle), the colored histograms illustrate the total spectral energy distribution  $ST$  across various eigenvalue intervals. Specifically, the histogram for the interval  $[0, 0.5]$  is calculated as  $ST_{[0,0.5]} = \sum_{\lambda \in [0,0.5]} S_E(\lambda)$ , where  $S_E(\lambda)$  represents the spectral energy distribution corresponding to the eigenvalue  $\lambda$ . It is evident that wavelet-specific features exhibit significant energy differences, depending on the specific wavelets employed during processing.

To address the challenge posed by significant energy differences among wavelet-specific features, we propose multiple wavelet-specific Feature Extractors  $\{\text{FE}_i\}_{i \in [0,2C]}$  for

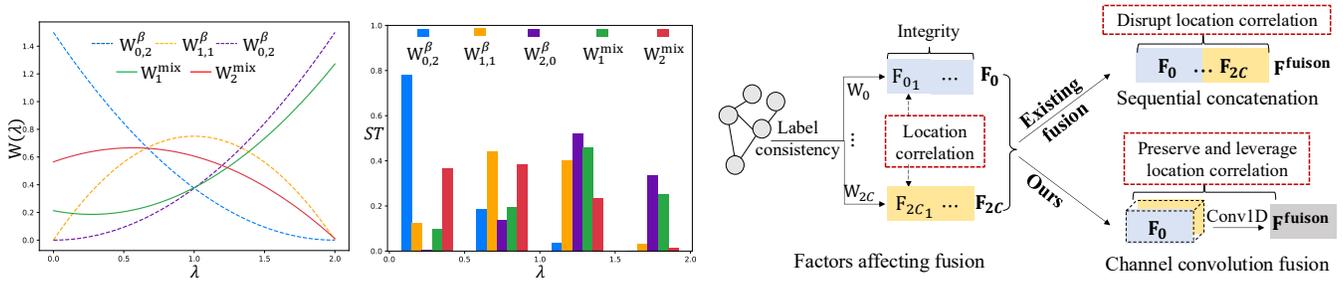


Figure 3: **Left**, spectral behaviors of three hand-designed Beta wavelets  $\mathbf{W}_{p,q}^\beta$  and two trained Beta-mixture wavelets  $\mathbf{W}_j^{\text{mix}}$  on dataset T-finance. The trained Beta-mixture wavelets exhibit different behaviors compared to Beta wavelets, enabling the extraction of anomaly information that Beta wavelets miss. Thus, our dynamic wavelets, combining trainable Beta-mixture wavelets and hand-designed Beta wavelets, capture more comprehensive anomaly information. **Middle**, energy difference between wavelet-specific features using various wavelets on the T-finance dataset. **Right**, factors affecting the consistent fusion of wavelet-specific features, along with illustrations showing existing hand-designed sequential concatenation at the top and our proposed channel convolution fusion at the bottom.

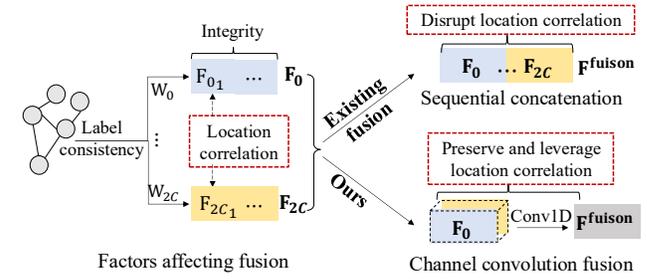
wavelet-specific feature extraction. All feature extractors share an identical structure based on Multi-Layer Perceptron (MLP), but their parameters distinct. Each feature extractor’s parameters are customized to process its corresponding wavelet-specific feature. This process generates deep wavelet-specific features, effectively mapping wavelet-specific features  $\mathbf{F}$  from different frequency band spaces into adjacent latent spaces, as illustrated in Figure 4. The ensemble of deep wavelet-specific features with space consistency  $\mathbf{F}^{\text{WFE}}$  is represented as:

$$\mathbf{F}^{\text{WFE}} = \text{FE}(\mathbf{F}) = \{\text{FE}_0(\mathbf{F}_{0,C}^\beta), \dots, \text{FE}_{2C}(\mathbf{F}_C^{\text{mix}})\}. \quad (6)$$

**Channel convolution fusion** Drawing inspiration from the orthographic view in geometry and conducting a thorough analysis of Eq. 4, we recognize that the consistent fusion of wavelet-specific features depends not only on energy difference but also on the following factors:

- **Label Consistency:** The labels of wavelet-specific features for the same node are consistent.
- **Integrity:** All information encompassed within wavelet-specific features is indispensable for precisely delineating normal or anomalous patterns within the corresponding frequency band space.
- **Location Correlation:** Wavelet-specific features processed using various wavelets exhibit correlations in the same location, denoting values on the same attribute.

Existing methods, as depicted in the top portion of Figure 3 (right), often directly concatenate wavelet-specific features sequentially. While such concatenation may leverage the integrity and label consistency within wavelet-specific features, it disrupts their location correlation, resulting in inconsistent fusion of wavelet-specific features. To leverage location correlations effectively while preserving integrity and label consistency, we propose a channel convolution fusion method as illustrated in the bottom portion of Figure 3 (right). This method integrates channel concatenation and convolution fusion, thereby obtaining fusion features  $\mathbf{F}^{\text{fusion}}$  with higher discrimination capability.



(1) **Channel concatenation.** Just as RGB channels are used to preserve feature integrity and maintain label consistency across individual color channels, we concatenate multiple deep wavelet-specific features  $\mathbf{F}^{\text{WFE}}$  along the channel dimension to create multi-channel features  $\mathbf{F}^C$ . This channel concatenation strategy maintains feature integrity and label consistency across all channels while avoiding irrelevant sequential information among wavelet-specific features. As a result, this approach facilitates the consistent utilization of location correlations among wavelet-specific features in the subsequent feature fusion.

The process for channel concatenation of deep wavelet-specific features can be expressed as:

$$\mathbf{F}^C = \text{Channel-Concate}(\mathbf{F}_0^{\text{WFE}}, \dots, \mathbf{F}_{2C}^{\text{WFE}}). \quad (7)$$

(2) **Convolution fusion.** Considering the location correlations across channels, we use one-dimensional convolutional (Conv1D) operations to fuse features across channels consistently. The formulation for the convolution fusion of multi-channel features  $\mathbf{F}^C$  is described as follows:

$$\begin{aligned} \mathbf{Y}[c_{\text{out}}, l_{\text{out}}] &= \text{Conv1D}(\mathbf{F}^C) \\ &= \sum_{c_{\text{in}}=1}^{C_{\text{in}}} \sum_{l=0}^{L_{\text{kernel}}-1} \mathbf{F}^C[c_{\text{in}}, l_{\text{out}} + l] \times \mathbf{K}[c_{\text{out}}, c_{\text{in}}, l], \end{aligned} \quad (8)$$

where  $\mathbf{Y}[c_{\text{out}}, l_{\text{out}}]$  represents the value at location  $l_{\text{out}}$  of the  $c_{\text{out}}$ -th output channel in the output tensor  $\mathbf{Y}$ .  $c_{\text{out}}$  is the output channel index.  $l_{\text{out}}$  is the location index in the output sequence.  $c_{\text{in}}$  denotes the number of input channels.  $L_{\text{kernel}}$  is the size of the convolutional kernel.  $\mathbf{F}^C[c_{\text{in}}, l_{\text{out}} + l]$  represents the value at location  $l_{\text{out}} + l$  of the  $c_{\text{in}}$ -th input channel in the input tensor.  $\mathbf{K}[c_{\text{out}}, c_{\text{in}}, l]$  denotes the weight value at location  $l$  of the  $c_{\text{out}}$ -th output channel and the  $c_{\text{in}}$ -th input channel in the convolutional kernel.

In this study, we apply two Conv1D operations along with normalization to the multi-channel features, resulting in the fusion features  $\mathbf{F}^{\text{fusion}} = \text{Conv1D}(\text{Conv1D}(\mathbf{F}^C))$ . By considering location correlations, the fusion features  $\mathbf{F}^{\text{fusion}}$  produced through consistent convolution fusion ex-

Dataset	Nodes	Edges	Edge types	Anomaly	Features
T-finance	39357	21222543	1	4.58%	10
Tolokers	11758	519000	1	21.8%	10
YelpChi	45954	3846979	3	14.53%	32
Amazon	11944	4398392	3	6.87%	25

Table 1: Statistics of four datasets

hibit higher discrimination. Finally, the weighted cross-entropy loss function  $\mathcal{L}$  is used to train the GNN model for the GAD task (Tang et al. 2022).

## 4 Experiment Settings

### 4.1 Datasets and Metrics

**Datasets.** Our experiments use four datasets: T-finance (Tang et al. 2022), Tolokers (Likhobaba, Pavlichenko, and Ustalov 2023), YelpChi (Rayana and Akoglu 2015), and Amazon (McAuley and Leskovec 2013), as detailed in Table 1. The T-finance and Tolokers datasets are both examples of homogeneous graphs. The T-finance dataset focuses on detecting anomalous accounts within transaction networks. The Tolokers dataset represents a network of workers from the crowdsourcing platform Toloka. In contrast, the YelpChi and Amazon datasets are heterogeneous graphs but can be treated as homogeneous graphs by ignoring edge types. The YelpChi dataset comprises spam and legitimate reviews of hotels and restaurants from Yelp.com, featuring three types of relationships (edge types). Amazon contains product reviews under the music instrument category of Amazon.com, with three types of relationships.

**Metrics.** This paper uses two widely recognized metrics: AUC and F1-macro (Wang et al. 2023b; Gao et al. 2023b). AUC, which stands for the area under the receiver operating characteristic curve, evaluates a model’s ability to differentiate between classes. On the other hand, F1-macro is the macro-averaged F1 score to assess the balance between precision and recall across all classes.

### 4.2 Implementation Details

In this paper, to ensure fairness, the ratio of training set/validation set/test set for all methods is fixed at 0.4/ 0.3/ 0.3. All methods are trained using the Adam optimizer with a learning rate of 0.01 for 100 epochs. Each method is executed 10 times, with the model’s performance evaluated based on the mean and standard deviation of the evaluation metrics at the 100-th epoch. The parameter  $C$ , crucial for determining the number of wavelets, is set to 2. Hidden layers in all methods are set to 64 dimensions. The Conv1D layer has a convolutional kernel size of 3 and a stride of 1. All methods are executed on a cloud server virtual machine equipped with 8 vCPUs (32G RAM) and one NVIDIA T4 Tensor Core GPU. Our method leverages the Deep Graph Library (DGL 2.0.0) within PyTorch 2.2.1 with Cuda 11.8. Additionally, we integrate the public source codes of other methods into our framework to ensure fair comparisons.

## 5 Experiment Results

### 5.1 Overall Performance Comparison

**Baselines.** Our baselines are categorized into two groups. The first group comprises traditional graph neural network (GNN) methods, including GCN (Kipf and Welling 2016), GAT (Veličković et al. 2017), GIN (Xu et al. 2019b), and ChebyNet (Defferrard, Bresson, and Vandergheynst 2016). The second group encompasses SOTA methods for GAD on homogeneous or heterogeneous graphs, including CARE-GNN (Dou et al. 2020), GPR-GNN (Chien et al. 2020), BWGNN (Tang et al. 2022), and GHRN (Gao et al. 2023a).

To demonstrate the superiority of our proposed DSGAD method, we conduct comprehensive comparisons with two groups of baseline methods. We evaluate the performance (measured by the average AUC and average F1-macro) of our method against all baseline methods on homogeneous graphs from four datasets. Additionally, to confirm the versatility of our proposed method on heterogeneous graphs, we compare it with three SOTA methods capable of handling heterogeneous graphs on the YelpChi and Amazon datasets. The performance comparison for both homogeneous and heterogeneous graphs is presented in Table 2.

**Results analysis.** On homogeneous graphs, our DSGAD (homo) significantly surpasses traditional methods such as GCN, GAT, GIN, and ChebyNet with improvements of over 3.44% in AUC and over 5.09% in F1-macro. The reasons for this are as follows: GCN, GAT, and GIN can be seen as low-pass filters (wavelets), focusing on low-frequency information within the graph. ChebyNet utilizes learnable Chebyshev kernels, which acts as a low-pass or band-pass filter, leveraging both low-frequency and mid-frequency information, thus performing better than GCN, GAT, and GIN. Our method not only considers low-frequency, mid-frequency, and high-frequency information but also incorporates adaptive mixture-frequency information learned during model training, thereby capturing more comprehensive anomaly information and further enhancing model performance.

Similarly, our DSGAD (homo) method outperforms recent methods such as CARE-GNN (homo), GPR-GNN, BWGNN (homo), and GHRN (homo) with improvements of over 1.47% in AUC and over 2.60% in F1-macro, for several reasons. First, we use dynamic wavelets to extract comprehensive anomaly information. Second, by considering the energy difference between wavelet-specific features, we map these features from different frequency spaces into adjacent latent spaces, facilitating subsequent feature fusion. Third, we use channel convolution fusion to consistently integrate wavelet-specific features, exploiting label consistency, feature integrity, and location correlation among these features. Overall, our method produces fusion features with comprehensive anomaly information and higher discrimination capability, thereby improving overall performance.

Furthermore, on heterogeneous graphs from YelpChi and Amazon, our DSGAD (hete) outperforms recent methods such as CARE-GNN (hete), BWGNN (hete), and GHRN (hete) with improvements of over 0.63% in AUC and over 0.85% in F1-macro. These findings underscore the superiority of the DSGAD framework, showcasing its effectiveness

Method	YelpChi		T-finance		Amazon		Tolokers		Average	
	AUC	F1-macro	AUC	F1-macro	AUC	F1-macro	AUC	F1-macro	AUC	F1-macro
GCN	57.24 $\pm$ 1.21	54.51 $\pm$ 1.27	86.58 $\pm$ 2.65	75.93 $\pm$ 5.04	78.91 $\pm$ 1.11	62.53 $\pm$ 2.00	70.67 $\pm$ 0.63	59.86 $\pm$ 0.61	73.35	63.21
GAT	53.75 $\pm$ 2.58	52.41 $\pm$ 1.71	51.18 $\pm$ 11.75	40.95 $\pm$ 6.62	65.51 $\pm$ 14.28	60.23 $\pm$ 9.91	68.44 $\pm$ 2.43	58.21 $\pm$ 1.70	59.72	52.95
GIN	72.66 $\pm$ 3.27	61.87 $\pm$ 2.59	81.86 $\pm$ 3.22	62.39 $\pm$ 4.84	88.11 $\pm$ 4.03	78.22 $\pm$ 6.98	73.80 $\pm$ 0.41	61.87 $\pm$ 0.29	79.13	66.09
ChebyNet	78.03 $\pm$ 0.35	65.34 $\pm$ 0.27	92.66 $\pm$ 1.72	79.57 $\pm$ 4.01	96.81 $\pm$ 0.24	92.26 $\pm$ 0.24	77.20 $\pm$ 0.26	65.15 $\pm$ 0.36	86.19	75.58
CARE-GNN (homo)	76.98 $\pm$ 0.27	64.81 $\pm$ 0.19	74.84 $\pm$ 11.14	56.40 $\pm$ 9.71	96.49 $\pm$ 0.33	91.75 $\pm$ 0.38	73.50 $\pm$ 0.14	61.52 $\pm$ 0.27	80.45	68.62
BWGNN (homo)	81.20 $\pm$ 0.50	68.41 $\pm$ 0.57	93.86 $\pm$ 1.04	80.92 $\pm$ 7.43	97.25 $\pm$ 0.31	92.14 $\pm$ 0.36	79.00 $\pm$ 0.23	66.63 $\pm$ 0.29	87.85	77.02
GHRN (homo)	81.14 $\pm$ 0.85	68.15 $\pm$ 0.69	94.92 $\pm$ 0.50	85.27 $\pm$ 1.57	97.43 $\pm$ 0.25	92.27 $\pm$ 0.16	79.10 $\pm$ 0.28	66.60 $\pm$ 0.47	<b>88.16</b>	<b>78.07</b>
GPR-GNN	82.98 $\pm$ 0.62	64.92 $\pm$ 0.99	94.06 $\pm$ 1.27	88.04 $\pm$ 4.12	97.79 $\pm$ 1.08	<b>93.06</b> $\pm$ 0.38	76.33 $\pm$ 0.16	50.19 $\pm$ 0.38	87.79	74.05
<b>Our DSGAD (homo)</b>	<b>84.77</b> $\pm$ 0.28	<b>71.86</b> $\pm$ 0.24	<b>96.51</b> $\pm$ 0.15	<b>91.80</b> $\pm$ 0.41	<b>97.91</b> $\pm$ 0.17	92.32 $\pm$ 0.22	<b>79.31</b> $\pm$ 0.48	<b>66.70</b> $\pm$ 0.67	<b>89.63</b>	<b>80.67</b>
CARE-GNN (hete)	80.77 $\pm$ 0.84	67.18 $\pm$ 0.73	-	-	94.92 $\pm$ 1.79	88.51 $\pm$ 1.32	-	-	87.85	77.85
BWGNN (hete)	87.19 $\pm$ 0.44	73.69 $\pm$ 0.59	-	-	97.03 $\pm$ 0.37	91.82 $\pm$ 0.55	-	-	92.11	82.76
GHRN (hete)	88.97 $\pm$ 0.48	75.09 $\pm$ 0.59	-	-	97.11 $\pm$ 0.46	91.43 $\pm$ 0.27	-	-	<u>93.04</u>	<u>83.26</u>
<b>Our DSGAD (hete)</b>	<b>89.30</b> $\pm$ 0.58	<b>75.49</b> $\pm$ 0.75	-	-	<b>97.73</b> $\pm$ 0.26	<b>92.14</b> $\pm$ 0.44	-	-	<b>93.67</b>	<b>84.11</b>

Table 2: The performance comparison results of our DSGAD method alongside two groups of comparison methods on both homogeneous (homo) and heterogeneous (hete) graphs across four datasets. In each comparison, the best performance is highlighted in bold, while the second-best performance is denoted with underlining.

across both homogeneous and heterogeneous graphs.

## 5.2 Ablation Analysis

**Ablation methods.** DSGAD incorporates dynamic wavelets (DW) and dynamic fusion, which combines wavelet feature extraction (WFE) with channel convolution fusion (CCF). To assess the contribution of each component to the overall performance of GAD, we systematically remove each component for the ablation study.

In addition to our DSGAD (homo) method, we introduce three ablation methods: **(1) WO/ (DW+WFE+CCF):** Omitting DW, WFE, and CCF components, the Beta mother wavelet is used for spectral decomposition. The obtained wavelet-specific features are concatenated sequentially to form fused features directly. **(2) WO/ (WFE+CCF):** Omitting WFE and CCF components, multiple wavelet-specific features obtained using dynamic wavelets are sequentially concatenated to obtain fusion features. **(3) WO/ CCF:** Omitting the CCF component, dynamic wavelets are initially employed to derive wavelet-specific features. Subsequently, corresponding feature extractors for these wavelet-specific features are utilized to obtain deep wavelet-specific features, which are then sequentially concatenated to form fusion features. All ablation results on four homogeneous graphs are depicted in Table 3.

**Results analysis.** According to the average AUC and F1-macro metrics in Table 3, removing one or more components leads to performance degradation in the WO/CCF, WO/(WFE+CCF), and WO/(DW+WFE+CCF) methods compared to our DSGAD method. Specifically, the reductions of AUC are 0.98%, 1.39%, and 1.78%, respectively, while the decreases of F1-macro are 1.64%, 2.56%, and 3.43%, respectively. These results highlight the critical importance of DW, WFE, and CCF components. Our DSGAD method achieves the best performance by extracting comprehensive anomaly information (DW), considering energy differences among wavelet-specific features (WFE), and leveraging their location correlations (CCF).

Next, Figure 4 illustrates the effectiveness of WFE us-

Wavelet-specific feature extraction (WFE)

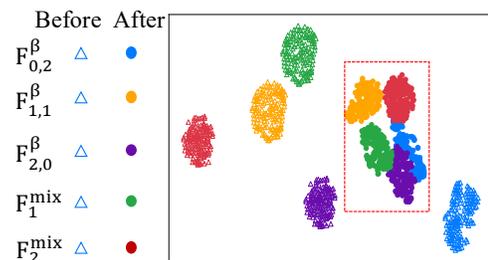


Figure 4: The t-SNE results for 150 nodes from the YelpChi dataset before and after applying WFE.

ing 150 nodes from the YelpChi dataset as an example. After WFE, the differences between wavelet-specific features across various frequency spaces are significantly reduced, which is attributed to WFE’s ability to map wavelet-specific features from different spaces into adjacent latent spaces.

Furthermore, the effectiveness of the DW can be attributed to two key components. First, the trained Beta-mixture wavelets and Beta wavelets complement each other, enabling the extraction of richer anomaly information. This is visually illustrated in Figure 3 (left) using the T-finance dataset. Second, the trainable coefficients of the Beta wavelets within the Beta-mixture wavelet are dynamically adjusted based on the input X, allowing the model to capture graph-specific anomalies. As a result, our DW, which integrates trainable Beta-mixture wavelets with hand-designed Beta wavelets, capture more comprehensive anomaly information than using hand-designed Beta wavelets alone.

## 5.3 Impact of Parameter $C$ on Performance

Parameter  $C$  directly determines the number of wavelets in dynamic wavelets, which can have a significant impact on the overall model performance. Therefore, it is imperative to comprehensively investigate the impact of  $C$  ( $1 \leq C \leq 7$ ) on the performance of our method. Figure 5 illustrates the

Method	YelpChi		T-finance		Amazon		Tolokers		Average	
	AUC	F1-macro	AUC	F1-macro	AUC	F1-macro	AUC	F1-macro	AUC	F1-macro
WO/(DW+WFE+CCF)	81.26 $\pm$ 0.50	68.41 $\pm$ 0.57	93.86 $\pm$ 1.03	80.92 $\pm$ 7.43	97.25 $\pm$ 0.31	92.14 $\pm$ 0.36	79.02 $\pm$ 0.23	66.63 $\pm$ 0.29	87.85	77.02
WO/(WFE+CCF)	82.12 $\pm$ 0.75	69.08 $\pm$ 0.80	94.53 $\pm$ 0.89	84.04 $\pm$ 5.67	97.26 $\pm$ 0.68	92.17 $\pm$ 0.31	79.06 $\pm$ 0.16	66.26 $\pm$ 0.54	88.24	77.89
WO/CCF	82.36 $\pm$ 1.00	69.52 $\pm$ 1.00	95.08 $\pm$ 0.52	86.60 $\pm$ 1.87	97.58 $\pm$ 0.13	92.18 $\pm$ 0.19	<b>79.58</b> $\pm$ 0.21	<b>66.95</b> $\pm$ 0.50	88.65	78.81
<b>Our DSGAD (homo)</b>	<b>84.77</b> $\pm$ 0.28	<b>71.86</b> $\pm$ 0.24	<b>96.51</b> $\pm$ 0.15	<b>91.80</b> $\pm$ 0.41	<b>97.91</b> $\pm$ 0.17	<b>92.32</b> $\pm$ 0.22	79.31 $\pm$ 0.48	66.70 $\pm$ 0.67	<b>89.63</b>	<b>80.67</b>

Table 3: Ablation experiment results on four homogeneous graphs.

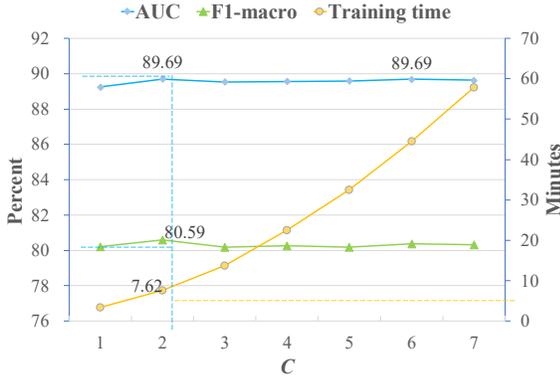


Figure 5: Effect of various  $C$  values on the average AUC, F1-macro, and training time of our DSGAD method in homogeneous graph scenarios across four datasets.

average AUC, average F1-macro, and average training time for our DSGAD method across various  $C$  values, evaluated on four datasets (Yelp, Amazon, T-Finance, and Tolokers) within homogeneous graph scenarios.

From Figure 5, it is evident that the performance of our method is relatively poor when  $C = 1$ . This deficiency can be attributed to the insufficient number of wavelets, leading to inadequate extraction of anomaly information across diverse frequency bands. Moreover, when  $C > 1$ , as the value of  $C$  increases, the required training time increases sharply, while the AUC and F1-macro metrics tend to stabilize. Balancing performance metrics constrained by training time,  $C = 2$  is used to realize optimal anomaly detection performance of our model.

## 6 Related Work

**Graph node classification.** Graph node classification is a fundamental task in graph learning, where GNNs have emerged as powerful models (Li et al. 2024; Ma et al. 2021; Qin et al. 2022; Luan et al. 2022; Wang et al. 2021). GNNs integrate node features and graph structure, achieving significant performance improvement. In a typical GNN architecture, each layer initiates message passing among node neighbors, followed by an aggregation step that updates node representations based on gathered information. Initially, Graph Convolutional Networks (GCNs) boosted the performance of graph node classification tasks (Kipf and Welling 2016). Subsequently, numerous GNN variants have been introduced, including GraphSAGE (Hamilton, Ying, and Leskovec 2017), GIN (Xu et al. 2019b),

CARE-GNN (Dou et al. 2020), FAGCN (Bo et al. 2021), BWGNN (Tang et al. 2022), GTAN (Xiang et al. 2023), and SEC-GFD (Xu et al. 2024).

**Spectral graph neural networks.** Spectral GNNs are a crucial branch of GNNs that treat graph data as signals on a graph and process them in the frequency domain (Xu et al. 2019a). They can be categorized into three categories: The first category, represented by GCN (Kipf and Welling 2016), operates as a linear filter. GCN is a first-order approximation of ChebyNet and functions akin to a low-pass filter. The second category includes polynomial filters like ChebyNet, which utilizes Chebyshev polynomials (Defferrard, Bresson, and Vandergheynst 2016). For instance, AMNet (Chai et al. 2022) leverages Bernstein polynomials to learn two filters that capture low-frequency and high-frequency features. Additionally, JacobiCov (Wang and Zhang 2022), BernNet (He et al. 2021), and NFGNN (Zheng et al. 2023) employ polynomial bases with learnable coefficients to design filters that adaptively extract features from the frequency domain. The third category employs graph wavelets for filtering (Hoang, Maehara, and Murata 2021; Dong et al. 2021). For instance, GWNN (Xu et al. 2018) utilizes graph wavelet transform to avoid the computational cost of matrix eigen-decomposition. BWGNN (Tang et al. 2022) proposes Beta wavelets as spectral and spatially localized filters to extract the information from each frequency band separately. Furthermore, building upon BWGNN, SEC-GFD (Xu et al. 2024) designs hybrid spectral filters to address the heterophily problem.

## 7 Conclusion and Future Work

This paper introduces a dynamic spectral graph anomaly detection (DSGAD) framework to integrate dynamic wavelets and dynamic fusion for graph anomaly detection. Dynamic wavelets consisting of trainable Beta-mixture wavelets extract comprehensive anomaly information adaptively. Dynamic fusion involves wavelet-specific feature extraction and channel convolution fusion to consistently integrate wavelet-specific features by considering their energy differences and location correlations. Extensive experimental results demonstrate that DSGAD delivers competitive performance. Future research will focus on enhancing graph anomaly detection performance by incorporating spatial-domain node features (attribute anomalies) alongside frequency-domain fusion features (structural anomalies). Additionally, dynamic wavelet computation on large-scale graphs may be computationally expensive, particularly for full-graph analysis, due to its high resource demands.

## 8 Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant Nos. 62172156 and 62072169, the China Scholarship Council Program under Grant No.202306130083, and the Postgraduate Scientific Research Innovation Project of Hunan Province under Grant No. QL20230099.

## References

- Balcilar, M.; Renton, G.; Héroux, P.; Gaüzère, B.; Adam, S.; and Honeine, P. 2021. Analyzing the expressive power of graph neural networks in a spectral perspective. In *Proceedings of the International Conference on Learning Representations*.
- Bo, D.; Wang, X.; Shi, C.; and Shen, H. 2021. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*.
- Chai, Z.; Yang, Y.; Dan, J.; Tian, S.; Meng, C.; Wang, W.; and Sun, Y. 2023. Towards learning to discover money laundering sub-network in massive transaction network. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Chai, Z.; You, S.; Yang, Y.; Pu, S.; Xu, J.; Cai, H.; and Jiang, W. 2022. Can abnormality be detected by graph neural networks? In *Proceedings of the International Joint Conference on Artificial Intelligence*.
- Chen, Z.; Ma, T.; and Wang, Y. 2022. When does a spectral graph neural network fail in node classification? *arXiv preprint arXiv:2202.07902*.
- Chien, E.; Peng, J.; Li, P.; and Milenkovic, O. 2020. Adaptive universal generalized pagerank graph neural network. *arXiv preprint arXiv:2006.07988*.
- De Oliveira, H.; and De Araújo, G. 2015. Compactly supported one-cyclic wavelets derived from beta distributions. *arXiv preprint arXiv:1502.02166*.
- Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in Neural Information Processing Systems*.
- Dong, X.; Zhang, X.; Sun, Y.; Chen, L.; Yuan, M.; and Wang, S. 2024. SmoothGNN: Smoothing-based GNN for unsupervised node anomaly detection. *arXiv preprint arXiv:2405.17525*.
- Dong, X.; Zhang, X.; and Wang, S. 2023. Rayleigh quotient graph neural networks for graph-level anomaly detection. *arXiv preprint arXiv:2310.02861*.
- Dong, Y.; Ding, K.; Jalaian, B.; Ji, S.; and Li, J. 2021. Adagnn: Graph neural networks with adaptive frequency response filter. In *Proceedings of the ACM international conference on information and knowledge management*.
- Dou, Y.; Liu, Z.; Sun, L.; Deng, Y.; Peng, H.; and Yu, P. S. 2020. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the ACM International Conference on Information and Knowledge Management*.
- Gao, X.; Dai, W.; Li, C.; Zou, J.; Xiong, H.; and Frossard, P. 2021. Message passing in graph convolution networks via adaptive filter banks. *arXiv preprint arXiv:2106.09910*.
- Gao, Y.; Wang, X.; He, X.; Liu, Z.; Feng, H.; and Zhang, Y. 2023a. Addressing heterophily in graph anomaly detection: A perspective of graph spectrum. In *Proceedings of the ACM Web Conference*.
- Gao, Y.; Wang, X.; He, X.; Liu, Z.; Feng, H.; and Zhang, Y. 2023b. Alleviating structural distribution shift in graph anomaly detection. In *Proceedings of the ACM International Conference on Web Search and Data Mining*.
- Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*.
- Hammond, D. K.; Vandergheynst, P.; and Gribonval, R. 2011. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*.
- Han, S.; Hu, X.; Huang, H.; Jiang, M.; and Zhao, Y. 2022. Adbench: Anomaly detection benchmark. *Advances in Neural Information Processing Systems*.
- He, J.; Xu, Q.; Jiang, Y.; Wang, Z.; and Huang, Q. 2024. ADA-GAD: Anomaly-Denoised Autoencoders for Graph Anomaly Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- He, M.; Wei, Z.; Xu, H.; et al. 2021. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. *Advances in Neural Information Processing Systems*.
- Hoang, N.; Maehara, T.; and Murata, T. 2021. Revisiting graph neural networks: Graph filtering perspective. In *Proceedings of the International Conference on Pattern Recognition*.
- Huang, M.; Liu, Y.; Ao, X.; Li, K.; Chi, J.; Feng, J.; Yang, H.; and He, Q. 2022. Auc-oriented graph neural network for fraud detection. In *Proceedings of the ACM Web Conference*.
- Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.
- Kumagai, A.; Iwata, T.; and Fujiwara, Y. 2021. Semi-supervised anomaly detection on attributed graphs. In *Proceedings of the International Joint Conference on Neural Networks*.
- Li, X.; Fan, Z.; Huang, F.; Hu, X.; Deng, Y.; Wang, L.; and Zhao, X. 2024. Graph neural network with curriculum learning for imbalanced node classification. *Neurocomputing*.
- Likhobaba, D.; Pavlichenko, N.; and Ustalov, D. 2023. Toloker graph: Interaction of crowd annotators.
- Liu, Y.; Ao, X.; Qin, Z.; Chi, J.; Feng, J.; Yang, H.; and He, Q. 2021. Pick and choose: A GNN-based imbalanced learning approach for fraud detection. In *Proceedings of the web conference*.
- Luan, S.; Hua, C.; Lu, Q.; Zhu, J.; Zhao, M.; Zhang, S.; Chang, X.-W.; and Precup, D. 2022. Revisiting heterophily for graph neural networks. *Advances in Neural Information Processing Systems*.

- Ma, X.; Wu, J.; Xue, S.; Yang, J.; Zhou, C.; Sheng, Q. Z.; Xiong, H.; and Akoglu, L. 2021. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering*.
- McAuley, J. J.; and Leskovec, J. 2013. From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews. In *Proceedings of the International Conference on World Wide Web*.
- Pang, G.; Shen, C.; Cao, L.; and Hengel, A. V. D. 2021. Deep learning for anomaly detection: A review. *ACM Computing Surveys*.
- Qin, Z.; Liu, Y.; He, Q.; and Ao, X. 2022. Explainable graph-based fraud detection via neural meta-graph search. In *Proceedings of the ACM International Conference on Information and Knowledge Management*.
- Rayana, S.; and Akoglu, L. 2015. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- Shi, F.; Cao, Y.; Shang, Y.; Zhou, Y.; Zhou, C.; and Wu, J. 2022. H2-fdetector: A GNN-based fraud detector with homophilic and heterophilic connections. In *Proceedings of the ACM web conference*.
- Tang, J.; Li, J.; Gao, Z.; and Li, J. 2022. Rethinking graph neural networks for anomaly detection. In *Proceedings of the International Conference on Machine Learning*.
- Taud, H.; and Mas, J.-F. 2018. Multilayer perceptron (MLP). *Geomatic Approaches for Modeling Land Change Scenarios*.
- Ten, C.-W.; Hong, J.; and Liu, C.-C. 2011. Anomaly detection for cybersecurity of the substations. *IEEE Transactions on Smart Grid*.
- Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.
- Wang, Q.; Pang, G.; Salehi, M.; Buntine, W.; and Leckie, C. 2023a. Cross-domain graph anomaly detection via anomaly-aware contrastive alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Wang, X.; Jin, B.; Du, Y.; Cui, P.; Tan, Y.; and Yang, Y. 2021. One-class graph neural networks for anomaly detection in attributed networks. *Neural Computing and Applications*.
- Wang, X.; and Zhang, M. 2022. How powerful are spectral graph neural networks. In *International conference on machine learning*.
- Wang, Y.; Zhang, J.; Huang, Z.; Li, W.; Feng, S.; Ma, Z.; Sun, Y.; Yu, D.; Dong, F.; Jin, J.; et al. 2023b. Label information enhanced fraud detection against low homophily in graphs. In *Proceedings of the ACM Web Conference*.
- Xiang, S.; Zhu, M.; Cheng, D.; Li, E.; Zhao, R.; Ouyang, Y.; Chen, L.; and Zheng, Y. 2023. Semi-supervised credit card fraud detection via attribute-driven graph representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Xu, B.; Shen, H.; Cao, Q.; Cen, K.; and Cheng, X. 2019a. Graph convolutional networks using heat kernel for semi-supervised learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*.
- Xu, B.; Shen, H.; Cao, Q.; Qiu, Y.; and Cheng, X. 2018. Graph wavelet neural network. In *Proceedings of the International Conference on Learning Representations*.
- Xu, F.; Wang, N.; Wen, X.; Gao, M.; Guo, C.; and Zhao, X. 2023. Few-shot Message-Enhanced Contrastive Learning for Graph Anomaly Detection. In *Proceedings of the IEEE International Conference on Parallel and Distributed Systems*.
- Xu, F.; Wang, N.; Wu, H.; Wen, X.; Zhao, X.; and Wan, H. 2024. Revisiting graph-based fraud detection in sight of heterophily and spectrum. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019b. How powerful are graph neural networks? In *Proceedings of the International Conference on Learning Representations*.
- Yu, R.; He, X.; and Liu, Y. 2015. Glad: group anomaly detection in social media analysis. *ACM Transactions on Knowledge Discovery from Data*.
- Zheng, S.; Zhu, Z.; Liu, Z.; Li, Y.; and Zhao, Y. 2023. Node-oriented spectral filtering for graph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.