Neural Networks 190 (2025) 107612

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Full Length Article

Frequency Self-Adaptation Graph Neural Network for Unsupervised Graph Anomaly Detection

Ming Gu [a], Gaoming Yang [d], Zhuonan Zheng [a], Meihan Liu [a], Haishuai Wang [a,b], Jiawei Chen [a], Sheng Zhou [b,c,*], Jiajun Bu [a,b]

[a] College of Computer Science and Technology, Zhejiang University, Hangzhou, 310027, China
[b] Zhejiang Key Laboratory of Accessible Perception and Intelligent Systems, Zhejiang University, Hangzhou, 310027, China
[c] School of Software Technology, Zhejiang University, Ningbo, 315048, China
[d] Taobao and Tmall Group, Hangzhou, 310052, China

ARTICLE INFO

Keywords:
Unsupervised Graph Anomaly Detection
Graph Neural Networks
Graph filtering

ABSTRACT

Unsupervised Graph Anomaly Detection (UGAD) seeks to identify abnormal patterns in graphs without relying on labeled data. Among existing UGAD methods, Graph Neural Networks (GNNs) have played a critical role in learning effective representation for detection by filtering low-frequency graph signals. However, the presence of anomalies can shift the frequency band of graph signals toward higher frequencies, thereby violating the fundamental assumptions underlying GNNs and anomaly detection frameworks. To address this challenge, the design of novel graph filters has garnered significant attention, with recent approaches leveraging anomaly labels in a semi-supervised manner. Nonetheless, the absence of anomaly labels in real-world scenarios has rendered these methods impractical, leaving the question of how to design effective filters in an unsupervised manner largely unexplored. To bridge this gap, we propose a novel **F**requency Self-**A**daptation **G**raph Neural Network for Unsupervised Graph **A**nomaly **D**etection (**FAGAD**). Specifically, FAGAD adaptively fuses signals across multiple frequency bands using full-pass signals as a reference. It is optimized via a self-supervised learning approach, enabling the generation of effective representations for unsupervised graph anomaly detection. Experimental results demonstrate that FAGAD achieves state-of-the-art performance on both artificially injected datasets and real-world datasets. The code and datasets are publicly available at https://github.com/eaglelab-zju/FAGAD.

1. Introduction

A graph anomaly refers to a deviation from the expected normal patterns within a graph, based on its intrinsic properties or structural characteristics (Kim, Lee, Shin, & Lim, 2022). Recently, Unsupervised Graph Anomaly Detection (UGAD) has attracted significant attention in both academic research and industrial applications due to its broad applicability in real-world graph data. Key domains include network security (Ten, Hong, & Liu, 2011), fraud detection (Ngai, Hu, Wong, Chen, & Sun, 2011), and health monitoring (Bao, Tang, Li, & Zhang, 2019). Graph Neural Networks (GNNs) have been widely adopted in UGAD (Liu, Dou, et al., 2022) due to their ability to learn discriminative representations, enabling effective anomaly detection.

Despite their successes, most existing Graph Anomaly Detection (GAD) methods (Ding, Li, Bhanushali, & Liu, 2019; Fan, Zhang, & Li, 2020; Jin, Liu, Zheng, Chi, Li, & Pan, 2021; Zhou, Tan, Xu, Huang, & Chung, 2021) rely on traditional GNNs with neural message-passing schemas for representation learning, such as Graph Convolutional Networks (GCN) (Kipf & Welling, 2016) and Graph Attention Networks (GAT) (Veličković, Cucurull, Casanova, Romero, Liò, & Bengio, 2017). However, these approaches often overlook the potential impact of graph anomalies on these GNNs. From a spectral perspective, GNNs employing neural message-passing mechanisms can be interpreted as *low-pass filters*. Anomalies, characterized by features distinct from their local neighborhoods, can shift the graph signal's frequency band toward the high-frequency region (Tang, Li, Gao, & Li, 2022). Consequently, GNNs as low-pass filters may fail to capture high-frequency signals, leading to the loss of key characteristics that differentiate anomalous nodes. This limitation hinders the ability to distinguish anomalous

* Correspondence to: Zhejiang Key Laboratory of Accessible Perception and Intelligent Systems, College of Computer Science and Technology, Zhejiang University, Hangzhou, 310027, China.

E-mail addresses: guming444@zju.edu.cn (M. Gu), yanggaoming.ygm@taobao.com (G. Yang), zhengzn@zju.edu.cn (Z. Zheng), lmh_zju@zju.edu.cn (M. Liu), haishuai.wang@zju.edu.cn (H. Wang), sleepyhunt@zju.edu.cn (J. Chen), zhousheng_zju@zju.edu.cn (S. Zhou), bjj@zju.edu.cn (J. Bu).

https://doi.org/10.1016/j.neunet.2025.107612
Received 24 January 2025; Received in revised form 30 April 2025; Accepted 11 May 2025
Available online 27 May 2025

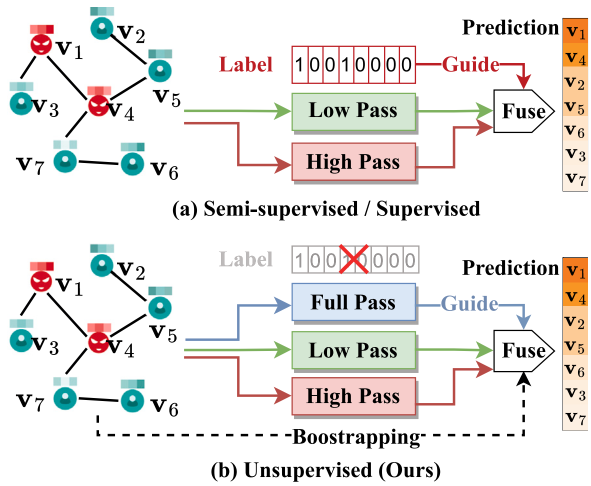

Fig. 1. Comparison of GNN-based graph anomaly detection methods in handling different frequency bands.

nodes from normal ones in the representation space. Hence, developing novel graph filters and GNNs that yield more discriminative representations is crucial for advancing anomaly detection.

Recent studies (Chai, You, Yang, Pu, Xu, Cai, & Jiang, 2022; Gao, Wang, He, Liu, Feng, & Zhang, 2023; Tang et al., 2022) have begun addressing this challenge. Specifically, these approaches employ multi-channel filters to capture signals across diverse frequency bands, *utilizing semi-supervised labels*, as depicted in Fig. 1-(a). However, obtaining anomaly labels in practice is often infeasible, limiting the applicability of these methods. This raises a critical and practical question: *How can we adaptively leverage different frequency bands across the entire spectrum in an unsupervised manner to learn effective representations for graph anomaly detection?* Addressing this question is significantly more challenging due to the absence of explicit supervision and remains an open research problem.

To bridge this gap, we propose a novel **F**requency Self-**A**daptation **G**raph Neural Network for Unsupervised Graph **A**nomaly **D**etection (FAGAD). Specifically, FAGAD derives three distinct variants of the generalized Laplacian smoothing filter — namely, full-pass, low-pass, and high-pass filters. Using the full-pass signals as a reference, FAGAD adaptively fuses low- and high-frequency signals through a carefully designed self-attentional module, optimizing the process in a self-supervised manner. To further enhance joint training and anomaly detection, we adopt a bootstrapping strategy. An auxiliary target encoder, updated via momentum, provides explicit guidance and iteratively bootstraps the training of FAGAD. Fig. 1-(b) illustrates the paradigm of the proposed method. We evaluate FAGAD on seven datasets, including both artificially injected and real-world datasets, against 15 baseline models. Experimental results demonstrate the effectiveness and superiority of FAGAD, achieving state-of-the-art performance. Our main contributions are summarized as follows:

- We explore adaptive utilization of different frequency bands in an unsupervised manner, facilitating the learning of distinguishable representations for both normal and anomalous nodes in graph data.
- We propose a novel Frequency Self-Adaptation Graph Neural Network (FAGAD) that effectively fuses signals across the entire frequency spectrum using a self-supervised approach with bootstrapping.
- Extensive experiments validate the effectiveness of FAGAD, achieving state-of-the-art performance on both synthetic and real-world datasets.

2. Related work

Graph Neural Networks. Spatial GNNs, such as GCN (Kipf & Welling, 2016), GAT (Veličković et al., 2017), and GraphSAGE (Hamilton, Ying, & Leskovec, 2017), employ message passing to aggregate local node information, enabling the update of node representations. In contrast, spectral GNNs, including ChebyNet (Defferrard, Bresson, & Vandergheynst, 2016), BernNet (He, Wei, Xu, et al., 2021), and JacobiConv (Wang & Zhang, 2022), utilize graph filters, performing convolution in the spectral domain to approximate arbitrary filters. In the domain of contrastive learning for graph representation, methods (Ai, Yan, Wang, & Li, 2024; Chen, Lei, & Wei, 2024; Ekbote, Deshpande, Iyer, Sellamanickam, & Bairi, 2023) following the training scheme of DGI (Veličković, Fedus, Hamilton, Liò, Bengio, & Hjelm, 2018) pioneered the approach of maximizing mutual information between local patches and the global graph summary. This enables the capture of global information; however, its reliance on negative sampling leads to significant memory overhead. Inspired by BYOL (Grill, Strub, Altché, Tallec, Richemond, Buchatskaya, Doersch, Avila Pires, Guo, Gheshlaghi Azar, et al., 2020), BGRL (Thakoor, Tallec, Azar, Azabou, Dyer, Munos, Veličković, & Valko, 2021) overcomes this limitation by learning node representations without requiring negative samples. For a comprehensive overview of advanced GNN models, we refer readers to recent surveys (Wu, Pan, Chen, Long, Zhang, & Philip, 2020; Zhou, Cui, Hu, Zhang, Yang, Liu, Wang, Li, & Sun, 2020).

Graph Anomaly Detection. Dominant (Ding et al., 2019) was the first to utilize a graph autoencoder for Graph Anomaly Detection (GAD). This approach encodes the graph using a GCN and reconstructs node attributes and structure through independent decoders, assigning anomaly scores to nodes based on reconstruction errors. CoLA (Liu, Li, Pan, Gong, Zhou, & Karypis, 2021) adopts a node-subgraph contrastive learning framework, contrasting a node's subgraph with both its own subgraph and those of other nodes. This method performs anomaly detection by leveraging differences between positive and negative samples. ComGA (Luo, Wu, Beheshti, Yang, Zhang, Wang, & Xue, 2022) integrates community information through the modularity matrix, progressively incorporating it into GCN encoding at each layer. Anomaly scores are evaluated in a manner similar to Dominant. Semi-supervised GAD methods, such as AMNet (Chai et al., 2022), BWGNN (Tang et al., 2022), and GHRN (Gao et al., 2023), enhance anomaly detection by designing filters to incorporate multi-band signals. However, their reliance on fixed filters constrains their capacity for adaptive learning. Although these models have achieved significant progress, the challenge of adaptively integrating signals across different frequency bands in an unsupervised manner remains largely unexplored.

3. Methodology

In this section, we present our proposed method FAGAD in detail as depicted in Fig. 2. We introduce frequency self-adaptation graph neural network, efficiently integrating multiple frequency bands through self-attentional adaptation, and propose a self-supervised learning framework via bootstrapping for concurrent graph representation learning and anomaly detection.

3.1. Notations

Consider an attribute graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{A}\}$, where $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ represents the node set with N instances. The node attributes are denoted as $\mathbf{X} \in \mathbb{R}^{N\times D}$. The edge set $\mathcal{E}$ is represented by the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$, where $a_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$, and $a_{ij} = 0$ otherwise. $\mathbf{D} = \mathrm{diag}(d_1, d_2, \dots, d_N)$ is the degree matrix of $\mathbf{A}$, where $d_i = \sum_{j=1}^{N} a_{ij}$. By adopting the renormalization trick (Kipf & Welling, 2016) $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, the symmetrically normalized graph Laplacian matrix is given by $\mathbf{L} = \mathbf{I} - \widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}$, where $\widetilde{\mathbf{D}}$ is the degree matrix of $\widetilde{\mathbf{A}}$. Its eigendecomposition is expressed as $\mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\intercal}$, where $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_N)$ denotes the diagonal matrix of eigenvalues, and the columns of $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N] \in \mathbb{R}^{N\times N}$ denote the orthonormal eigenvectors of $\mathbf{L}$.

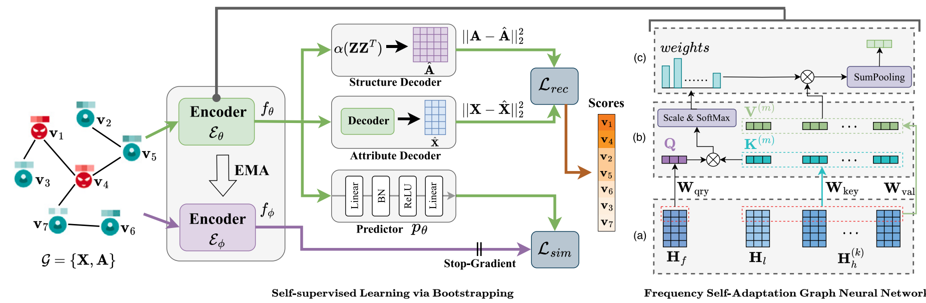

Fig. 2. Overview of FAGAD. (1) A frequency self-adaptation graph neural network is proposed to adaptively learn the optimal combination of different frequency bands with the guidance of the full-pass signals; (2) An unsupervised strategy via bootstrapping is employed to jointly learn node representations and graph anomaly detection.

3.2. Overview of the fagad pipeline

As showed in Fig. 2, FAGAD consists of two key modules: (1) a frequency self-adaptation graph neural network, and (2) a bootstrapped self-supervised learning strategy for joint node representation learning and anomaly detection.

Frequency Self-Adaptation Graph Neural Network: As discussed in the Introduction, traditional GNNs act predominantly as low-pass filters, making them prone to overlook high-frequency signals associated with anomalies. To overcome this, we designed frequency self-adaptation graph neural networkto derive three generalized Laplacian smoothing filters — full-pass, low-pass, and high-pass — that decompose the graph signals across the frequency spectrum. This enables FAGAD to capture and preserve critical anomalous patterns typically encoded in high-frequency components, while still retaining the low-frequency information necessary for modeling the overall graph structure.

Self-attentional Frequency Adaptation Module: Simply concatenating signals from different bands risks introducing irrelevant information and noise. Therefore, we propose a self-attentional fusion module that adaptively weighs low- and high-frequency signals with respect to the full-pass reference. This design allows FAGAD to dynamically emphasize the most informative frequency components for each node, rather than relying on fixed or heuristic weighting, thus enhancing the expressiveness and anomaly sensitivity.

Self-supervised Learning via Bootstrapping (without data augmentation): Inspired by the success of bootstrapping strategies like BYOL, we incorporate a target encoder updated via an Exponential Moving Average (EMA) mechanism. This allows the model to gradually stabilize its node representations and improve detection robustness. Importantly, unlike traditional contrastive learning, we omit data augmentation, as augmentations (e.g., edge dropout, feature masking) can introduce artificial anomalies, which are undesirable in unsupervised anomaly detection tasks. Removing augmentation preserves the graph's integrity and focuses learning on real anomalies.

Dual Decoder for Attribute and Structure Reconstruction: To simultaneously capture anomalies in both node attributes and graph structure, we employ dual reconstruction decoders — one for attributes and one for the adjacency matrix. This dual-objective design ensures that FAGAD can comprehensively detect diverse anomaly types (e.g., nodes with anomalous features, nodes connected inconsistently with their local community).

In summary, each architectural choice in FAGAD is carefully motivated by the unique challenges of unsupervised GAD — particularly the need to capture multi-pass signals, adaptively weigh signals, stabilize representations without supervision, and handle both attribute and structural irregularities.

3.3. Frequency self-adaptation graph neural network

According to the graph signal processing theory (Shuman, Narang, Frossard, Ortega, & Vandergheynst, 2012), the eigenvectors $\mathbf{U}$ of the graph Laplacian matrix $\mathbf{L}$ serve as the *graph Fourier basis*, and the eigenvalues $\boldsymbol{\Lambda}$ are referred to as *frequencies*. Therefore, signals $\mathbf{x} \in \mathbb{R}^{N\times 1}$ can be decomposed using the graph Fourier basis $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N]$:

$$\mathbf{x} = \mathbf{U}\mathbf{p} = \sum_{i=1}^{N} p_i \mathbf{u}_i, \tag{1}$$

where $\mathbf{p} \in \mathbb{R}^{N\times 1}$ is the coefficient for the decomposition. Since the eigenvectors are orthonormal $\mathbf{U}^\intercal\mathbf{U} = \mathbf{I}$, the signals after the graph filtering process can be represented as:

$$\begin{aligned}\tilde{\mathbf{x}} &= g_\theta \star \mathbf{x} = \mathbf{U} g_\theta(\boldsymbol{\Lambda})\mathbf{U}^\intercal \mathbf{x} = \boldsymbol{\Theta}\mathbf{x} \\ &= \mathbf{U} g_\theta(\boldsymbol{\Lambda})\mathbf{U}^\intercal\mathbf{U}\mathbf{p} = \mathbf{U} g_\theta(\boldsymbol{\Lambda})\mathbf{p} = \sum_{i=1}^{N} g_\theta(\lambda_i) p_i \mathbf{u}_i,\end{aligned} \tag{2}$$

where g_θ is a function of the eigenvalues $\boldsymbol{\Lambda}$, and the matrix $\boldsymbol{\Theta} = \mathbf{U} g_\theta(\boldsymbol{\Lambda})\mathbf{U}^\intercal$ can be seen as the matrix form of the filter. To measure the smoothness of the signals, the *Rayleigh quotient* (Horn & Johnson, 2012) over $\tilde{\mathbf{x}}$ is calculated as:

$$R(\tilde{\mathbf{x}}, \mathbf{L}) = \frac{\tilde{\mathbf{x}}^\intercal \mathbf{L} \tilde{\mathbf{x}}}{\tilde{\mathbf{x}}^\intercal \tilde{\mathbf{x}}} = \frac{\sum_{i=1}^{N} [g_\theta(\lambda_i) p_i]^2 \lambda_i}{\sum_{i=1}^{N} [g_\theta(\lambda_i) p_i]^2}. \tag{3}$$

For *low-pass* filters, the aim is to attain smoother signals by filtering out the high-frequency components while preserving low-frequency components. In this context, according to (3), $[g_\theta(\lambda_i)]^2$ should decrease as the frequency λ_i increases. This characteristic ensures the suppression of higher frequencies. Conversely, the filter is referred to as a *high-pass* filter (Cui, Zhou, Yang, & Liu, 2020).

3.3.1. Multi-pass graph filter

As highlighted in the introduction, a fundamental necessity for GNNs in GAD is the ability to effectively capture signals of varying frequencies. To address this requirement, we employ three distinct types of graph filters, each derived from different variants of the generalized Laplacian smoothing filter (Taubin, 1995), namely the full-pass, low-pass, and high-pass filters:

$$\boldsymbol{\Theta}_f = \mathbf{I}, \boldsymbol{\Theta}_l = 2\mathbf{I} - \mathbf{L}, \boldsymbol{\Theta}_h = \mathbf{L} \tag{4}$$

where $\mathbf{I}$ is an identity matrix. Thus the filtered signals $\tilde{\mathbf{X}} = \boldsymbol{\Theta}\mathbf{X} \in \mathbb{R}^{N\times D}$ where $\boldsymbol{\Theta}$ can be one of $\boldsymbol{\Theta}_f$, $\boldsymbol{\Theta}_l$ and $\boldsymbol{\Theta}_h$.

It should be noted that $\boldsymbol{\Theta}_f$ simply passes all the signals without applying any filtering to the original features, thus being a full-pass filter. Meanwhile, we can prove that $\boldsymbol{\Theta}_l$ and $\boldsymbol{\Theta}_h$ are low-pass and high-pass with the corresponding Rayleigh quotients as shown below:

$$R(\tilde{\mathbf{x}}, \boldsymbol{\Theta}_l) = \frac{\sum_{i=1}^{N} (2-\lambda_i)^2 p_i^2 \lambda_i}{\sum_{i=1}^{N} (2-\lambda_i)^2 p_i^2}, \tag{5}$$

$$R(\tilde{\mathbf{x}}, \boldsymbol{\Theta}_h) = \frac{\sum_{i=1}^{N} \lambda_i^2 p_i^2 \lambda_i}{\sum_{i=1}^{N} \lambda_i^2 p_i^2}. \tag{6}$$

Proof. According to (5), $[g_\theta(\lambda_i)]^2$ of the filter $\boldsymbol{\Theta}_l$ is $(2-\lambda_i)^2$. Considering that the eigenvalues λ_i of $\mathbf{L}$ fall within the range $[0,2]$ (Shuman, Narang, Frossard, Ortega, & Vandergheynst, 2013), and $(2-\lambda_i)^2$ decreases as λ_i increases within the range, the filter $\boldsymbol{\Theta}_l$ functions as a low-pass filter. Conversely, according to (6), $[g_\theta(\lambda_i)]^2 = \lambda_i^2$ for $\boldsymbol{\Theta}_h$ results in an increase as λ_i increases. Therefore, $\boldsymbol{\Theta}_h$ serves as a high-pass filter. □

To more effectively capture the distinct high frequencies that are closely associated with anomalies, we expand the high-pass filter Θ_h into a series of variants, each possessing different high-pass capabilities:

$$\boldsymbol{\Theta}_h^{(k)} = \mathbf{L}^k, k \in \{1,2,\dots,K\}, \tag{7}$$

where $[g_\theta(\lambda_i)]^2 = (\lambda_i^k)^2$, and their Rayleigh quotients are:

$$R(\tilde{\mathbf{x}}, \boldsymbol{\Theta}_h^{(k)}) = \frac{\sum_{i=1}^N \lambda_i^{2k} p_i^2 \lambda_i^k}{\sum_{i=1}^N \lambda_i^{2k} p_i^2}. \tag{8}$$

By increasing the value of k, $\boldsymbol{\Theta}_l^{(k)}$ will progressively emphasize the capture of higher frequencies. This can be observed as $\lambda_i^k \geq \lambda_i$ within the range of high frequencies $\lambda_i \in [1,2]$.

3.3.2. Self-attentional frequency adaptation

After capturing signals of different frequencies, the subsequent challenge lies in selectively integrating these signals in a manner that optimally preserves the most relevant information, instead of employing a simplistic summarization approach. To address this challenge, a self-attention mechanism is employed. Prior to that, a multilayer perceptron (MLP) is utilized for feature extraction from the filtered signals: $\mathbf{H} = \widetilde{\mathbf{X}}\mathbf{W}$, where $\mathbf{H} \in \mathbb{R}^{N\times d}$ represents the latent embedding, and $\mathbf{W} \in \mathbb{R}^{D\times d}$ denotes the weight matrix. We refer to the embeddings obtained from the full-pass, low-pass, and high-pass signals as $\mathbf{H}_f$, $\mathbf{H}_l$, and $\mathbf{H}_h^{(k)}$, respectively.

Next, we utilize the full-pass embedding as the *query* matrix $\mathbf{Q}$ and use the low- and high-pass embedding to form the *key* matrices $\mathbf{K}^{(m)}$ and *value* matrices $\mathbf{V}^{(m)}$:

$$\mathbf{Q} = \mathbf{H}_f \mathbf{W}_{\text{qry}}, \tag{9}$$

$$\mathbf{K}^{(m)} = \begin{cases} \mathbf{H}_l \mathbf{W}_{\text{key}}, & \text{if } m = 0, \\ \mathbf{H}_h^{(k)} \mathbf{W}_{\text{key}}, & \text{if } m = k \in \{1,2,\dots,K\}, \end{cases} \tag{10}$$

$$\mathbf{V}^{(m)} = \begin{cases} \mathbf{H}_l \mathbf{W}_{\text{val}}, & \text{if } m = 0, \\ \mathbf{H}_h^{(k)} \mathbf{W}_{\text{val}}, & \text{if } m = k \in \{1,2,\dots,K\}, \end{cases} \tag{11}$$

where $\mathbf{W}_{\text{qry}}, \mathbf{W}_{\text{key}}, \mathbf{W}_{\text{val}} \in \mathbb{R}^{d\times d'}$ are the weight matrices. The value K represents the maximum power of the high-pass filters defined in (8). Each row $\mathbf{q}_i$, $\mathbf{k}_i^{(m)}$, and $\mathbf{v}_i^{(m)}$ of $\mathbf{Q}$, $\mathbf{K}^{(m)}$, and $\mathbf{V}^{(m)}$ denotes the query, key and value embedding of node v_i, respectively.

Subsequently, the self-attention score $\alpha_i^{(m)}$ of node v_i with the mth key $\mathbf{k}_i^{(m)}$ can be calculated as follows:

$$\alpha_i^{(m)} = \frac{\exp(w_i^{(m)})}{\sum_{m=0}^K \exp(w_i^{(m)})}, \quad w_i^{(m)} = \frac{\mathbf{q}_i \mathbf{k}_i^{(m)}}{\sqrt{d_i}}, \tag{12}$$

where d_i is the dimension of node embedding. Finally, by combining the value vectors $\mathbf{v}_i$ with the obtained attention weights, the embedding $\mathbf{z}_i$ can be obtained by:

$$\mathbf{z}_i = \sum_{m=0}^K \alpha_i^{(m)} \mathbf{v}_i^{(m)}. \tag{13}$$

In this self-attention mechanism, the keys associated with the largest attention weights tend to preserve the most information from the signals, as they exhibit the highest similarity to the full-pass signals. *Therefore, this characteristic enables the final embedding $\mathbf{z}_i$ to adaptively learn the optimal combination of different frequency bands.*

3.4. Self-supervised learning via bootstrapping

With the adaptive capture of different frequency bands, the challenge of jointly learning node representations and anomaly detection arises. Here, we employ the bootstrapping strategy for self-supervised training, drawing inspiration from BYOL (Grill et al., 2020). However, we have implemented a crucial modification by excluding data augmentation from the strategy, which aims to prevent the introduction of additional anomalous information during the training process.

3.4.1. Self-supervised graph representation learning

To begin with, we utilize two frequency self-adaptation graph neural networks followed by a projection head as encoders with different parameter sets to obtain node representations, namely an online encoder $\mathcal{E}_\theta$ and a target encoder $\mathcal{E}_\phi$:

$$\mathbf{Z}_o = f_\theta(\mathcal{E}_\theta(\mathbf{X},\mathbf{A})), \ \mathbf{Z}_t = f_\phi(\mathcal{E}_\phi(\mathbf{X},\mathbf{A})), \tag{14}$$

where θ and ϕ represent the encoder and projection head parameters. The online representation $\mathbf{Z}_o$ is then fed into a node-level predictor p_θ, which outputs a prediction of the target representation:

$$\widetilde{\mathbf{Z}}_o = p_\theta(\mathbf{Z}_o). \tag{15}$$

Updating the online encoder $\mathcal{E}_\theta$. The parameters θ of the online encoder are updated by minimizing the mean squared error loss between the online prediction representation $\widetilde{\mathbf{Z}}_o$ and target representation $\mathbf{Z}_t$. By applying normalization, the loss can be simplified into a cosine similarity-related loss:

$$\mathcal{L}_{sim} = \|\widetilde{\mathbf{Z}}_o - \mathbf{Z}_t\|_2^2 = 2N - \frac{2}{N}\sum_{i=0}^{N-1} \frac{\widetilde{\mathbf{Z}}_{o,i} \cdot \mathbf{Z}_{t,i}}{\|\widetilde{\mathbf{Z}}_{o,i}\|_2 \cdot \|\mathbf{Z}_{t,i}\|_2}. \tag{16}$$

The Adam optimizer is used to optimize the θ parameter with a learning rate η: $\theta \leftarrow \text{optimize}\left(\theta, \eta, \partial_\theta \mathcal{L}_{sim}\right)$.

Updating the target encoder $\mathcal{E}_\phi$. The parameters ϕ of the target encoder are updated using momentum (Haynes, Corns, & Venayagamoorthy, 2012) with an exponential moving average (EMA) of θ in the online encoder:

$$\phi \leftarrow \tau\phi + (1-\tau)\theta, \tag{17}$$

where τ is the decay rate.

3.4.2. Anomaly detection

In our approach, we follow the approach of Dominant (Ding et al., 2019) for anomaly scoring, based on graph structure and node attribute reconstruction:

$$R_s = \|\mathbf{A} - \widehat{\mathbf{A}}\|_2^2, \ R_a = \|\mathbf{X} - \widehat{\mathbf{X}}\|_2^2, \tag{18}$$

where $\widehat{\mathbf{A}} = \delta(\mathbf{Z}_o\mathbf{Z}_o^\intercal)$ and $\widehat{\mathbf{X}} = \mathcal{D}_\sigma(\mathbf{Z}_o, \mathbf{A})$. δ is the sigmoid function, and $\mathcal{D}_\sigma$ stands for a decoder that shares the same architecture as the encoders. Then the objective function of our anomaly detection module is expressed as:

$$\mathcal{L}_{rec} = (1-\alpha)R_s + \alpha R_a, \tag{19}$$

where α controls the balance between the two reconstructions. Finally, the anomaly score $\mathbf{s}_i$ is defined as:

$$\mathbf{s}_i = (1-\alpha)\left\|\mathbf{a}_i - \hat{\mathbf{a}}_i\right\|_2 + \alpha\left\|\mathbf{x}_i - \hat{\mathbf{x}}_i\right\|_2, \tag{20}$$

where higher scores indicate greater anomalousness.

3.4.3. Joint optimization

We jointly optimize each component of the FAGAD model as follows:

$$\mathcal{L} = \beta\mathcal{L}_{sim} + (1-\beta)\mathcal{L}_{rec}, \tag{21}$$

where β controls the balance between the impacts of reconstruction loss and bootstrapping loss.

As the self-supervised learning progresses, the target encoder adapts to the evolving representations of the online encoder according to the momentum update in (17), which in turn enhances the anomaly detection module's ability in the online encoder with $\mathcal{L}_{sim}$ to identify anomalous nodes based on the learned target representations.

3.5. Parameter count and complexity analysis

3.5.1. Parameter count analysis

Self-attentional Frequency Adaptation Module (Encoder). The self-attentional frequency adaptation module mainly introduces learnable parameters through multilayer perceptrons (MLPs) and attention-related transformations. **(1) Feature Projection MLP:** Given the input filtered signal $\widetilde{\mathbf{X}} \in \mathbb{R}^{N\times D}$, a one-layer MLP projects it into a latent space: $\mathbf{H} = \widetilde{\mathbf{X}}\mathbf{W}$, where $\mathbf{W} \in \mathbb{R}^{D\times d}$ is the trainable weight matrix. **Parameter count:** $D\times d$. **(2) Attention Transformation:** For computing the query, key, and value embeddings, the following learnable matrices are introduced: $\mathbf{W}_{\text{qry}}, \mathbf{W}_{\text{key}}, \mathbf{W}_{\text{val}} \in \mathbb{R}^{d\times d'}$. **Parameter count:** $3\times(d\times d')$. Thus, the total number of parameters in this module is: $\mathrm{P}_1 = D\times d+3\times(d\times d')$.

Self-supervised Learning via Bootstrapping. The self-supervised framework contains two encoders (online and target), each comprising: **(1) Decoder (i.e., frequency self-adaptation graph neural network):** The decoder consists of the same architecture of the encoder: $\text{Params}_{\text{de}} = d'\times d + 3\times(d\times D)$. **(2) Projection Head f_θ and f_ϕ:** The projection head is: $f : \mathbb{R}^{d'} \to \mathbb{R}^{d_p}$, with parameter count: $d'\times d_p$. **(3) Predictor p_θ:** Similarly, the predictor is: $p_\theta : \mathbb{R}^{d'} \to \mathbb{R}^{d_p}$, with parameter count: $d'\times d_p$. Since the target encoder $\mathcal{E}_\phi$ is updated via EMA and not directly optimized, we only count parameters once for the online branch. Thus, the total number of parameters in this module is: $\mathrm{P}_2 = d'\times d + 3\times(d\times D) + 2(d'\times d_p)$.

Therefore, the total parameter count of FAGAD is:

$$\text{Params}_{\text{Total}} = P_1 + P_2 = 4(Dd + dd') + 2(d'd_p). \tag{22}$$

3.5.2. Complexity analysis

Multi-pass Graph Filtering: Each graph filter requires multiplying the graph Laplacian $\mathbf{L}^k$ with node features $\mathbf{X}$, which takes $O(N^2D)$ per filter. With K high-pass filters and one low-pass filter, the total cost is $O_1 = O((K+1)N^2D)$.

Frequency Self-Adaptation Graph Neural Network: First, the cost of the feature extraction from the filtered signals before frequency self-adaptation graph neural network is $O((K+1)NDd)$. Second, projecting filtered features into queries, keys, and values requires $O(Ndd')$ for each projection. The attention computation across $K+1$ frequency bands takes $O(N(K+1)d')$. Combined, this stage costs $O_2 = O((K+1)NDd + Ndd' + N(K+1)d')$.

Self-supervised Learning: Adjacency reconstruction incurs $O(N^2d')$, while attribute reconstruction with a decoder of the same architecture as the encoders adds $O((K+1)Nd'd + NdD + N(K+1)D)$. The bootstrapping loss $\mathcal{L}_{sim}$ contributes $O(Nd'd_p)$ for projections. Combined, this stage costs $O_3 = O(N^2d'+(K+1)Nd'd+NdD+N(K+1)D+Nd'd_p)$.

The overall time complexity per iteration is dominated by:

$$\begin{aligned} O_{\text{total}} =& O_1 + O_2 + O_3 \\ =& N^2\left[(K+1)D + d'\right] + \\ & N\left[(K+1)(Dd + d' + d'd + D) + dd' + d'd_p + Dd\right]. \end{aligned} \tag{23}$$

The quadratic terms $O(N^2(K+1)D)$ and $O(N^2d')$ arise from dense graph filtering and adjacency reconstruction, which can be optimized via sparse matrix operations for real-world graphs. The linear terms $O(N)$ ensures scalability to large node sets. Compared to standard GCNs ($O(|\mathcal{E}|Dd)$), FAGAD trades spatial complexity for enhanced frequency awareness, justified by its anomaly detection gains. *Efficiency evaluation compared to various baselines across different datasets in Section 4.7 also showed the competitive efficiency of FAGAD.*

4. Experiments

In this section, we present extensive experiments to evaluate the effectiveness of the proposed FAGAD method.

4.1. Datasets and baselines

We conduct a series of experiments encompassing two artificially injected social network datasets, namely Flickr and BlogCatalog (Tang & Liu, 2009), along with six real-world datasets: Amazon (Sánchez, Müller, Laforet, Keller, & Böhm, 2013), Enron (Sánchez et al., 2013), Reddit, Wiki (Kumar, Zhang, & Leskovec, 2019), and Facebook (Xu, Huang, Zhao, Dong, & Li, 2022). For the statistics information regarding these datasets, please refer to Table 2. In the case of the injected datasets, we adhere to the injection methodology outlined in Ding et al. (2019).

We compare FAGAD with 15 state-of-the-art unsupervised baselines, categorized into four groups: (1) *Autoencoder-based UGAD methods:* Dominant (Ding et al., 2019), AnomalyDAE (Fan et al., 2020), GUIDE (Yuan, Zhou, Yu, Huang, Chen, & Xia, 2021), DONE (Bandyopadhyay, Lokesh, Vivek, & Murty, 2020), DMGD (Bandyopadhyay, Vishal Vivek, & Murty, 2020) and GADNR (Roy, Shu, Li, Yang, Elshocht, Smeets, & Li, 2024). (2) *Contrastive Learning-based UGAD methods:* CoLA (Liu et al., 2021), SL-GAD (Zheng, Jin, Liu, Chi, Phan, & Chen, 2021), ANEMONE (Jin et al., 2021), and CONAD (Xu et al., 2022). (3)*Other UGAD methods:* ComGA (Luo et al., 2022), AAGNN (Zhou et al., 2021), GAAN (Chen, Liu, Wang, Dai, Lv, & Bo, 2020), TAM (Qiao & Pang, 2023), and OCGNN (Wang, Jin, Du, Cui, Tan, & Yang, 2021).

4.2. Experimental settings and implementation details

We adopted the Area Under the Curve (AUC) as our performance measurement. To ensure the reliability of the outcomes, we conducted each experiment five times using different seeds. The results are presented as the mean and standard deviation, as depicted in Table 2. Hyperparameter settings are detailed in Table 1. The source code and datasets are publicly available at https://github.com/eaglelab-zju/FAGAD. To provide a comprehensive understanding of our framework, we elaborate here on the architecture and hyperparameter settings used in our experiments as follows.

Detailed Pipeline Architecture: (1) *Encoder Structure:* The encoder is a frequency self-adaptation graph neural network. The hidden dimension h varies per dataset, as detailed in Table 1, ranging from 128 (Reddit) to 2048 (Enron and Facebook). (2) *Attribute Decoder:* The same architecture as the encoder is employed to reconstruct node attributes $\mathbf{X}$. (3) *Structure Decoder:* Node embeddings are used to approximate the adjacency matrix via inner products. (4) *Predictor $\mathcal{P}_\theta$:* MLPs with batch normalization (BN) and ReLU activation project the online encoder outputs before computing the bootstrapping loss. (5) *Loss Functions:* The model jointly minimizes the reconstruction loss $\mathcal{L}_{\text{rec}}$ and the bootstrapping loss $\mathcal{L}_{\text{sim}}$, balanced by a coefficient β.

Detailed Architecture of the Frequency Self-adaptation Graph Neural Network: (1) *Multi-pass Graph Filters:* The input embeddings are filtered into multiple frequency bands: low-pass (H_l), full-pass (H_f), and multiple high-pass ($H_h^{(k)}$) frequencies. The maximum order K of high-pass filters is dataset-specific (e.g., $K=5$ for BlogCatalog, $K=2$ for Wiki). (2) *Self-attentional Frequency Adaptation:* Multi-frequency features are first transformed by matrices $\mathbf{W}_{qry}$, $\mathbf{W}_{key}$, and $\mathbf{W}_{val}$. The query-key attention computes similarity scores across different frequency components, normalized via scale and softmax operations. The

Table 1
Hyperparameter Settings. Blog. is short for BlogCatalog. h is the dimension of the encoder output, while τ denotes the decay rate of updating the target encoder. K represents the maximum order of the high-pass Laplacian filters, and lr is short for learning rate. α denotes the rate to control the balance between structure and attribute reconstruction. β is the rate to control the balance between reconstruction loss and bootstrapping loss.

	Flickr	Blog.	Amazon	Reddit	Wiki	Enron	Facebook
h	512	1024	512	128	1024	2048	2048
τ	0.9	0.9999	0.99	0.99	0.999	0.9999	0.99
K	2	5	1	3	2	2	8
lr	1e−3	1e−3	1e−3	1e−4	1e−5	1e−3	1e−3
α	0.99	0.99	0.2	0.99	0.001	0.99	0.1
β	0.9	0.1	0.9	0.5	0.9	0.5	0.9

Table 2
AUC (%) of All Baselines. The best results are in **bold**, and the second-best results are underlined. *A.R.* denotes the anomaly rates.

		Injected anomaly		Ground-truth anomaly					
		Flickr	BlogCatalog	Amazon	Reddit	Wiki	Enron	Facebook	Avg. Rank
Dataset	#nodes	7575	5196	1418	10,984	8227	13,533	1081	
	#edges	239,738	171,743	7390	168,016	752,879	353,974	55,104	
	#feats	12,047	8189	21	64	64	18	576	
	#A.R.	5.94%	5.77%	1.97%	3.33%	2.64%	0.04%	2.31%	
Unsupervised Methods	DOMINANT	61.70 ± 0.69	57.80 ± 1.30	48.33 ± 5.62	55.99 ± 0.09	49.30 ± 0.12	71.59 ± 0.35	48.02 ± 0.18	12.43
	AnomalyDAE	73.37 ± 0.55	63.19 ± 0.44	53.56 ± 1.64	55.81 ± 0.30	52.38 ± 0.61	57.77 ± 1.79	83.13 ± 1.89	9.0
	ComGA	65.93 ± 0.57	62.86 ± 1.24	55.06 ± 3.58	55.26 ± 0.21	51.92 ± 0.42	61.19 ± 0.91	82.03 ± 0.38	10.0
	CoLA	59.03 ± 0.15	62.43 ± 1.02	46.45 ± 3.83	55.14 ± 1.55	56.23 ± 0.24	52.67 ± 1.68	**96.73 ± 0.55**	10.14
	SLGAD	73.11 ± 0.33	71.63 ± 0.27	46.74 ± 9.93	60.02 ± 0.87	52.97 ± 0.47	57.87 ± 1.00	84.62 ± 7.52	7.86
	ANEMONE	49.60 ± 1.59	49.67 ± 2.26	48.92 ± 2.04	52.85 ± 0.99	54.50 ± 0.58	50.84 ± 9.81	92.86 ± 1.55	11.71
	AAGNN	75.26 ± 0.76	73.76 ± 1.33	38.50 ± 8.19	54.31 ± 0.86	43.80 ± 0.60	49.79 ± 9.42	63.72 ± 3.32	11.86
	GUIDE	74.07 ± 0.44	74.81 ± 1.16	53.94 ± 0.21	56.18 ± 0.44	51.18 ± 0.12	49.19 ± 0.06	54.34 ± 0.74	9.86
	DONE	73.22 ± 0.97	74.69 ± 1.04	57.98 ± 3.84	56.28 ± 1.13	48.72 ± 0.73	73.36 ± 0.05	91.09 ± 0.21	6.57
	CONAD	<u>76.39 ± 0.69</u>	<u>78.03 ± 0.15</u>	55.12 ± 4.49	57.08 ± 3.59	47.32 ± 1.99	73.84 ± 1.04	59.42 ± 12.95	6.57
	GAAN	63.47 ± 7.72	56.35 ± 10.71	58.83 ± 2.21	56.21 ± 1.07	53.24 ± 0.99	73.11 ± 0.00	58.20 ± 5.63	9.14
	DMGD	57.69 ± 1.37	58.17 ± 1.93	54.18 ± 3.52	58.88 ± 2.48	52.12 ± 1.09	71.95 ± 2.55	66.83 ± 10.66	9.71
	OCGNN	66.36 ± 0.35	59.58 ± 0.54	<u>63.03 ± 1.74</u>	55.67 ± 6.89	<u>58.16 ± 1.95</u>	<u>77.27 ± 3.91</u>	90.44 ± 0.82	6.57
	TAM	68.78 ± 4.66	75.37 ± 0.03	56.15 ± 2.55	<u>60.20 ± 0.3</u>	53.90 ± 0.49	56.60 ± 4.31	<u>93.33 ± 1.60</u>	<u>5.29</u>
	GADNR	67.88 ± 1.16	66.00 ± 1.31	53.04 ± 2.06	56.95 ± 2.62	52.59 ± 2.73	73.56 ± 1.52	76.21 ± 7.26	8.0
	FAGAD (Ours)	**76.82 ± 0.07**	**79.03 ± 0.29**	**70.22 ± 0.30**	**63.97 ± 0.37**	**63.77 ± 0.01**	**78.41 ± 0.06**	93.08 ± 0.01	**1.29**

attention weights are applied to value vectors and aggregated by sum pooling. (3) *Final Prediction Layer:* An MLP layer maps the aggregated representation to the task-specific output.

Training: (1) *Optimizer:* Adam optimizer is used throughout with a learning rate (lr) specified per dataset (e.g., 1e−3 for Flickr, 1e−5 for Wiki). Weight decays are set to 0 for most datasets, except for Reddit and Amazon with 0.00001 as specified. (2) *Dropout:* A dropout rate of 0.1 or 0.2 is applied in the predictor MLP as indicated in the training scripts. (3) *Number of Layers:* Encoder and predictor MLPs each consist of two linear layers. The attention module utilizes one-layer attention heads over frequency-filtered features.

The complete list of hyperparameters per dataset, including the dimension of hidden features h, the EMA decay rate τ, filter order K, learning rates lr and balance factors α and β, is provided in Table 1.

4.3. Performance analysis

In this subsection, we evaluate the performance of the proposed FAGAD, with the results summarized in Table 2. Several key observations can be drawn from the analysis.

First, FAGAD demonstrates consistent performance across the majority of datasets, achieving an average ranking of 1.29, significantly outperforming the second-best method, which achieves an average ranking of 5.29. Specifically, in graphs with injected anomaly labels, FAGAD surpasses all baseline methods, highlighting its ability to effectively identify nodes with manually injected structural and attribute anomalies. Furthermore, FAGAD exhibits substantial improvements over state-of-the-art methods on most real-world datasets, demonstrating its capability to detect unknown types of anomalous nodes within graphs. Notably, on the Amazon dataset, FAGAD achieves an AUC of 70.22%, outperforming the second-best method (OCGNN, 63.03%) by 11.4%. Similarly, on the Wiki dataset, FAGAD achieves an AUC of 63.77%, representing a significant improvement of 9.6% over the next-best method (58.16%). This stability is particularly pronounced in datasets with varying anomaly rates (A.R.), underscoring the robustness of FAGAD's multi-pass signal integration architecture.

In contrast, baseline methods exhibit inconsistent performance across injected and real-world anomaly datasets. For example, CONAD achieves strong results on the BlogCatalog dataset with an AUC of 78.03%, but its performance declines significantly on real-world datasets. Similarly, while CoLA and DMGD excel on the real-world Facebook datasets, achieving top-tier AUCs, they struggle on injected datasets such as Flickr and BlogCatalog. In comparison, FAGAD consistently adapts to both injected and real-world anomaly datasets, emphasizing its balanced and versatile capabilities.

A key differentiating factor of FAGAD lies in its use of multi-pass signal fusion, which sets it apart from existing graph anomaly detection methods. By effectively aggregating and integrating information across multiple channels, FAGAD captures complex neighborhood relationships that other models fail to exploit. This ability to leverage multi-pass signals enables FAGAD to uncover subtle anomalies and ensures generalizability across diverse graph structures, making it particularly well-suited for graph anomaly detection tasks.

4.4. Visualization of attention distribution

We conducted an analysis of attention weight distribution on the Reddit dataset. This analysis aims to confirm whether the model can effectively combine multiple frequency bands to capture comprehensive information regarding anomalies. As depicted in Fig. 3, as the filter leans toward high-pass, the attention weights of anomalous nodes progressively surpass those of normal nodes. This signifies our model's ability to adaptively harness signals from various frequency bands,

Table 3
AUC (%) of Ablation Studies for Filter Designs on Different Datasets. The best results are in bold, and the second-best results are underlined.

Dataset	Injected anomaly		Ground-truth anomaly			
	Flickr	BlogCatalog	Amazon	Reddit	Wiki	Enron
FAGAD-GCN	75.76 ± 0.34	75.45 ± 0.75	53.60 ± 1.01	46.49 ± 0.93	49.25 ± 0.21	72.22 ± 7.35
FAGAD-GAT	76.13 ± 0.00	76.02 ± 0.00	49.68 ± 0.50	57.32 ± 0.01	52.32 ± 0.04	60.04 ± 0.83
FAGAD-Cheb	76.09 ± 0.00	76.02 ± 0.00	65.14 ± 0.00	57.78 ± 0.01	48.11 ± 0.01	47.53 ± 10.60
FAGAD-Bern	76.35 ± 0.03	75.40 ± 0.14	38.01 ± 0.42	55.47 ± 0.11	53.57 ± 0.05	27.91 ± 0.51
FAGAD	**77.02 ± 0.00**	**78.34 ± 0.05**	**70.86 ± 0.01**	**64.36 ± 0.39**	**63.77 ± 0.01**	**78.39 ± 0.22**

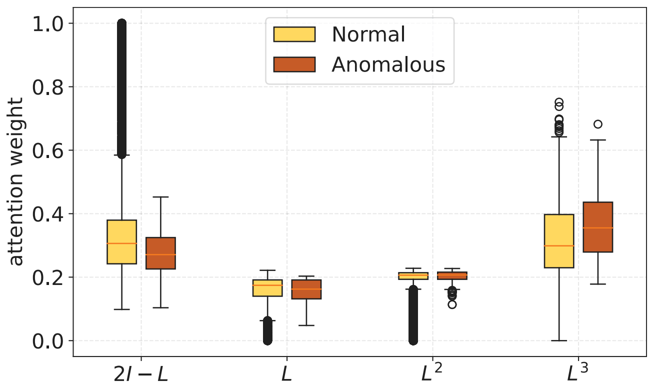

Fig. 3. Analysis of attention distribution on Reddit.

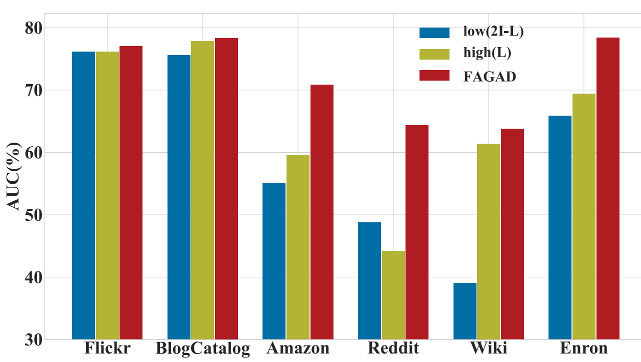

Fig. 4. AUC(%) of low- and high-pass filters.

tailoring its learning to different nodes and thereby enhancing its anomaly detection proficiency.

The observed trends align with the understanding that normal nodes favor low-frequency signals, while anomalies exhibit higher weights in the high-frequency range. These results affirm FAGAD's ability to capture intricate patterns across frequency domains.

4.5. Ablation studies

Effectiveness of Frequency Self-Adaptation Graph Neural Network. To validate the efficacy of our proposed frequency self-adaptation graph neural network , we compare it to four variants, including GCN, GAT, ChebyNet (Cheb), and BernNet (Bern), across two injected datasets and four real-world datasets. As presented in Table 3, our proposed gnn consistently outperforms the other variants across all datasets. The diminished performance of GCN and GAT methods can be attributed to their inclination toward low-frequency signals, which proves detrimental to accurate anomaly detection. Although ChebyNet and BernNet can capture both high and low frequencies, they still fall short of achieving results comparable to our model. This is due to the constrained nature of their fixed polynomial filters, which restricts the flexibility and expressive capacity in effectively selecting diverse frequencies of critical information. In contrast, our model excels in adaptively capturing the most pertinent information from various frequency bands for each individual node.

Table 4
AUC values(%) of ablation studies for the effectiveness of bootstrapping.

Dataset	w/o bootstrapping	w/ bootstrapping
BlogCatalog	76.70 ± 0.02	**78.34 ± 0.05**
Reddit	63.92 ± 0.03	**64.36 ± 0.39**
Wiki	63.25 ± 0.01	**64.77 ± 0.01**

Table 5
AUC (%) with Different Data Augmentation. FAGAD-AE,FAGAD-MF, and FAGAD-RN denote the model with adding edges, masking node features and removing nodes.

Augmentation	BlogCatalog	Reddit	Wiki
FAGAD-AE	76.60 ± 0.34	51.41 ± 0.52	63.39 ± 0.01
FAGAD-MF	64.19 ± 4.00	40.83 ± 1.64	62.13 ± 1.17
FAGAD-RN	75.15 ± 0.72	54.46 ± 3.99	63.58 ± 0.09
FAGAD (Ours)	**78.34 ± 0.05**	**64.36 ± 0.39**	**63.77 ± 0.01**

Moreover, we perform experiments comparing our approach with pure high-pass and low-pass filters. The outcome depicted in Fig. 4 highlights that our model adeptly learns node representations in an adaptive manner, tailored to the unique input data characteristics of different frequency bands. This adaptability contributes to notable improvements in graph anomaly detection.

Effectiveness of Self-supervised Learning via Bootstrapping. We conduct experiments by removing the self-supervised learning via bootstrapping module to verify the necessity of it. The outcomes are illustrated in Table 4. Module removal leads to a significant AUC reduction on some datasets. This decline serves as an indication that the inclusion of the bootstrapping strategy contributes to the model's efficacy. The effectiveness of the bootstrapping strategy stems from its utilization of exponential moving averages, which bootstraps the target encoder to obtain more stable and anomaly-discernible node representations. As a result, these enhanced target node representations provide improved guidance to the online encoder. Consequently, this synergistic effect leads to notable performance improvements.

The impact of refining the bootstrapping strategy. BYOL (Grill et al., 2020) employs a data augmentation strategy within the bootstrapping strategy, while FAGAD excludes the data augmentation module to prevent the introduction of additional noisy information. To assess the impact, we conducted a comparative analysis involving three variant models that incorporated the data augmentation module. We report the AUC values on an injected anomalous dataset, BlogCatalog, and a real-world dataset, Reddit (see Table 5). Removing the data augmentation module resulted in optimal performance compared to variants using it. This indicates that the data augmentation module may introduce additional anomalous information, diminishing overall model performance.

4.6. Sensitivity of hyperparameters

In this section, we delve into the impact of parameters within FAGAD. Fig. 5 illustrates all experimental results. First, on the Flickr, BlogCatalog, and Wiki datasets, using 4-th or 5-th order Laplacian matrices yields superior results. For the Amazon and Enron datasets,

Table 6
Average Training Time (in Seconds) Over 10 Epochs. The best are in **bold** with the second best underlined. *A.R.* denotes average rank.

Dataset	FAGAD	DONE	TAM	OCGNN	SLGAD	CONAD
Reddit	0.82 ± 0.56	16.98 ± 1.96	41.56 ± 0.72	**0.48 ± 0.06**	433.79 ± 2.78	0.76 ± 0.36
Flickr	15.44 ± 0.79	7.70 ± 0.86	34.79 ± 0.29	**0.92 ± 0.21**	386.75 ± 7.40	2.60 ± 0.26
A.R.	3.5	3.5	5	**1**	6	2

optimal performance occurs when the parameter k is 1 or 2. The varying optimal k underscores dataset characteristics, influencing suitable frequency band combinations. Second, experimental results for β show most datasets have consistent effects across different β values, except Enron and Reddit. Optimal results for them happen around $\beta = 0.5$. As β increases, AUC performance declines due to higher dependency on reconstruction loss. Third, the FAGAD excels on Flickr, BlogCatalog, Enron, and Reddit datasets, with $\alpha = 0.99$ prioritizing node attribute reconstruction. For Amazon, performance stays stable with $\alpha \leq 0.7$ but declines when $\alpha > 0.7$, emphasizing the need to balance graph structure and attribute reconstruction. Finally, MLP predictor p_θ in FAGAD impacts anomaly detection by predicting target encoder representations, influencing signal fusion across frequency bands. Experimental results for the Reddit dataset show FAGAD achieves optimal results with a smaller embedding dimension h, suggesting fewer parameters required.

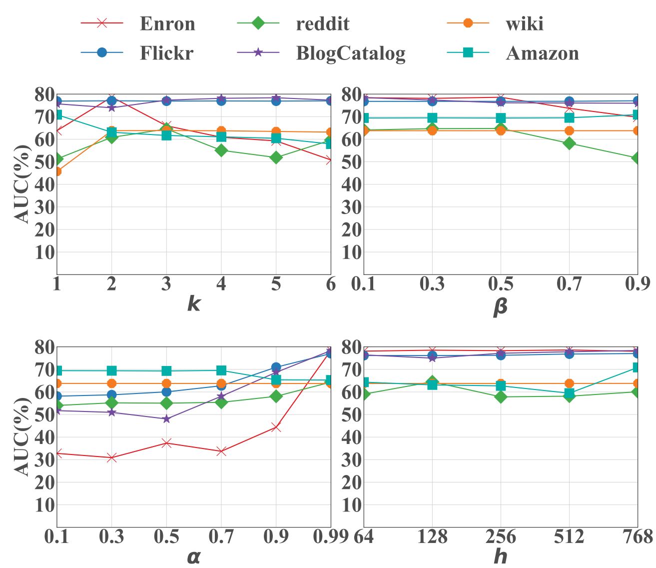

Fig. 5. Parameter analysis for anomaly detection.

4.7. Efficiency evaluation

To evaluate the efficiency of the proposed method, we report the average training time (in seconds) over 10 epochs and 3 independent runs across two representative datasets: Reddit (real-world) and Flickr (synthetic). The results, shown in Table 6, include comparisons with five state-of-the-art baseline models achieving top 5 average ranks in Table 2: TAM, DONE, CONAD, OCGNN, and SLGAD. The best and second-best results are highlighted in bold and underlined, respectively.

Among all methods, SLGAD incurs the highest training cost due to its combined generative and contrastive discrimination mechanisms, which are inherently resource-intensive. TAM ranks as the second most time-consuming method across both datasets, largely because of its complex node distance preprocessing steps. DONE also exhibits high training times, particularly as the number of nodes increases.

In contrast, our proposed method, FAGAD, demonstrates strong efficiency characteristics. It ranks third in average training time across datasets, exhibiting low training cost on sparse graphs such as Reddit. On denser graphs like Flickr, the training time is moderately higher due to the use of high-order Laplacian-based filters, which can introduce additional computational overhead as the matrix sparsity decreases.

Overall, FAGAD achieves a favorable balance between effectiveness and efficiency, offering scalable performance with competitive training times.

5. Discussion of limitation and future work

While the proposed FAGAD framework demonstrates strong performance across a wide range of datasets, we acknowledge several limitations that suggest promising directions for future research.

Limitations: First, FAGAD relies on the availability of meaningful graph signal information. In cases where the graph structure or node attributes are extremely noisy or sparse, the effectiveness of multi-frequency signal integration may be reduced, potentially affecting anomaly detection performance.

Second, although FAGAD shows robust results on medium-to-large graphs such as Reddit, applying the model to extremely large-scale graphs (e.g., graphs with hundreds of millions of nodes) could introduce computational challenges. The multi-pass graph filtering and self-attentional frequency adaptation steps, while effective, may increase memory and computation costs in such scenarios.

Third, while our hyperparameter sensitivity analysis indicates that FAGAD is generally robust, the model's performance still benefits from careful tuning of hyperparameters such as the Laplacian filter order K and the balance coefficients α and β. In some settings, suboptimal hyperparameter choices could lead to performance degradation.

Future Work: To address these limitations, future research could explore several promising directions:

- *Robust Signal Extraction:* Designing more robust pre-processing, structure learning techniques or noise-resilient graph filters to better handle highly noisy or incomplete graphs (Fatemi, El Asri, & Kazemi, 2021; Gu, Yang, Zhou, Ma, Chen, Tan, Liu, & Bu, 2023; Liu, Zheng, et al., 2022).
- *Scalability Improvements:* Developing scalable variants of FAGAD through techniques such as pre-computing, graph sampling, mini-batch training, or sparse frequency attention to enable efficient processing of ultra-large graphs (Chang, Rong, Xu, Huang, Sojoudi, Huang, & Zhu, 2021; Liu, Ren, & Chen, 2025).
- *Automated Hyperparameter Tuning:* Investigating meta-learning or self-tuning strategies to automatically adapt hyperparameters to different datasets without the need for extensive manual search (Spinelli, Scardapane, & Uncini, 2022; Zhu, Tao, Li, & Li, 2021).
- *Broader Applications:* Extending FAGAD to anomaly detection in dynamic graphs, heterogeneous graphs, or heterophilic graphs, which present even richer structures and challenges (Gu, Zheng, Zhou, Liu, Chen, Qiao, Li, & Bu, 2024; Luan, Hua, Lu, Zhu, Zhao, Zhang, Chang, & Precup, 2022).

Through these efforts, we aim to further enhance the robustness, scalability, and versatility of FAGAD for broader real-world deployments.

6. Conclusion

In this work, we have addressed the crucial challenges in the context of unsupervised graph anomaly detection based on GNNs. Specifically, existing works either have disregarded the reciprocal influence of anomalies, which results in the distortion of high-frequency signals, or have captured high-frequency signals through a semi-supervised approach, leaving the unsupervised setting largely unexplored. Our solution involves a frequency self-adaptation graph neural network that aptly fuses signals across multiple frequency bands under the guidance of the full-pass signals. We also harness self-supervised learning via bootstrapping to optimize this fusion process effectively. The results show the prowess of the FAGAD, which outperforms prevailing methods on both injected and real-world datasets. In future work, we will further refine filters to better capture various signals. Subsequently, we have plans to integrate community structure analysis with graph anomaly detection, enhancing the detection of anomalous groups for practical industrial use.

CRediT authorship contribution statement

Ming Gu: Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Gaoming Yang:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Zhuonan Zheng:** Writing – review & editing, Visualization. **Meihan Liu:** Writing – review & editing. **Haishuai Wang:** Writing – review & editing. **Jiawei Chen:** Writing – review & editing. **Sheng Zhou:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization. **Jiajun Bu:** Writing – review & editing, Resources, Funding acquisition.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used GPT-4o in order to improve language and readability. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 62476245), Zhejiang Provincial Natural Science Foundation of China (Grant No. LTGG23F030005).

Data availability

I have shared the link to may data/code in the manuscript.

References

Ai, G., Yan, H., Wang, H., & Li, X. (2024). A2GCN: Graph convolutional networks with adaptive frequency and arbitrary order. *Pattern Recognition*, *156*, Article 110764.

Bandyopadhyay, S., Lokesh, N., Vivek, S. V., & Murty, M. N. (2020). Outlier resistant unsupervised deep architectures for attributed network embedding. In *WSDM '20: the thirteenth ACM international conference on web search and data mining*.

Bandyopadhyay, S., Vishal Vivek, S., & Murty, M. N. (2020). Integrating network embedding and community outlier detection via multiclass graph description. In *ECAI 2020* (pp. 976–983). IOS Press.

Bao, Y., Tang, Z., Li, H., & Zhang, Y. (2019). Computer vision and deep learning–based data anomaly detection method for structural health monitoring. *Structural Health Monitoring*, *18*(2), 401–421.

Chai, Z., You, S., Yang, Y., Pu, S., Xu, J., Cai, H., et al. (2022). Can abnormality be detected by graph neural networks? In L. D. Raedt (Ed.), *Proceedings of the thirty-first international joint conference on artificial intelligence* (pp. 1945–1951). International Joint Conferences on Artificial Intelligence Organization, http://dx.doi.org/10.24963/ijcai.2022/270, Main Track.

Chang, H., Rong, Y., Xu, T., Huang, W., Sojoudi, S., Huang, J., et al. (2021). Spectral graph attention network with fast eigen-approximation. In *Proceedings of the 30th ACM international conference on information & knowledge management* (pp. 2905–2909).

Chen, J., Lei, R., & Wei, Z. (2024). PolyGCL: GRAPH CONTRASTIVE LEARNING via learnable spectral polynomial filters. In *The twelfth international conference on learning representations*. URL: https://openreview.net/forum?id=y21ZO6M86t.

Chen, Z., Liu, B., Wang, M., Dai, P., Lv, J., & Bo, L. (2020). Generative adversarial attributed network anomaly detection. In *Proceedings of the 29th ACM international conference on information & knowledge management* (pp. 1989–1992).

Cui, G., Zhou, J., Yang, C., & Liu, Z. (2020). Adaptive graph encoder for attributed graph embedding. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 976–985).

Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in Neural Information Processing Systems*, *29*.

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.

Ekbote, C., Deshpande, A., Iyer, A., Sellamanickam, S., & Bairi, R. B. (2023). FiGURe: Simple and efficient unsupervised node representations with filter augmentations. In *Thirty-seventh conference on neural information processing systems*. URL: https://openreview.net/forum?id=yh0OkiUk5h.

Fan, H., Zhang, F., & Li, Z. (2020). Anomalydae: Dual autoencoder for anomaly detection on attributed networks. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing* (pp. 5685–5689). IEEE.

Fatemi, B., El Asri, L., & Kazemi, S. M. (2021). SLAPS: Self-supervision improves structure learning for graph neural networks. *Advances in Neural Information Processing Systems*, *34*, 22667–22681.

Gao, Y., Wang, X., He, X., Liu, Z., Feng, H., & Zhang, Y. (2023). Addressing heterophily in graph anomaly detection: A perspective of graph spectrum. In *Proceedings of the ACM web conference 2023* (pp. 1528–1538).

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., et al. (2020). Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, *33*, 21271–21284.

Gu, M., Yang, G., Zhou, S., Ma, N., Chen, J., Tan, Q., et al. (2023). Homophily-enhanced structure learning for graph clustering. In *Proceedings of the 32nd ACM international conference on information and knowledge management* (pp. 577–586).

Gu, M., Zheng, Z., Zhou, S., Liu, M., Chen, J., Qiao, T., et al. (2024). Universal inceptive GNNs by eliminating the smoothness-generalization dilemma. arXiv preprint arXiv:2412.09805.

Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & et al. (Eds.), *Advances in neural information processing systems: vol. 30*, Curran Associates, Inc., URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf.

Haynes, D., Corns, S., & Venayagamoorthy, G. K. (2012). An exponential moving average algorithm. In *2012 IEEE congress on evolutionary computation* (pp. 1–8). http://dx.doi.org/10.1109/CEC.2012.6252962.

He, M., Wei, Z., Xu, H., et al. (2021). Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. *Advances in Neural Information Processing Systems*, *34*, 14239–14251.

Horn, R. A., & Johnson, C. R. (2012). *Matrix analysis*. Cambridge University Press.

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).

Kim, H., Lee, B. S., Shin, W.-Y., & Lim, S. (2022). Graph anomaly detection with graph neural networks: Current status and challenges. *IEEE Access*.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. http://dx.doi.org/10.48550/ARXIV.1609.02907, URL: https://arxiv.org/abs/1609.02907.

Kumar, S., Zhang, X., & Leskovec, J. (2019). Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1269–1278).

Liu, K., Dou, Y., Zhao, Y., Ding, X., Hu, X., Zhang, R., et al. (2022). Bond: Benchmarking unsupervised outlier node detection on static attributed graphs. *Advances in Neural Information Processing Systems*, *35*, 27021–27035.

Liu, Y., Li, Z., Pan, S., Gong, C., Zhou, C., & Karypis, G. (2021). Anomaly detection on attributed networks via contrastive self-supervised learning. CoRR abs/2103.00113, arXiv:2103.00113.

Liu, W., Ren, Z., & Chen, L. (2025). Knowledge reasoning based on graph neural networks with multi-layer top-p message passing and sparse negative sampling. *Knowledge-Based Systems*, Article 113063.

Liu, Y., Zheng, Y., Zhang, D., Chen, H., Peng, H., & Pan, S. (2022). Towards unsupervised deep graph structure learning. In *Proceedings of the ACM web conference 2022* (pp. 1392–1403).

Luan, S., Hua, C., Lu, Q., Zhu, J., Zhao, M., Zhang, S., et al. (2022). Revisiting heterophily for graph neural networks. *Advances in Neural Information Processing Systems*, *35*, 1362–1375.

Luo, X., Wu, J., Beheshti, A., Yang, J., Zhang, X., Wang, Y., et al. (2022). Comga: Community-aware attributed graph anomaly detection. In *Proceedings of the fifteenth ACM international conference on web search and data mining* (pp. 657–665).

Ngai, E. W., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*, *50*(3), 559–569.

Qiao, H., & Pang, G. (2023). Truncated affinity maximization: One-class homophily modeling for graph anomaly detection. In *Advances in neural information processing systems*.

Roy, A., Shu, J., Li, J., Yang, C., Elshocht, O., Smeets, J., et al. (2024). Gad-nr: Graph anomaly detection via neighborhood reconstruction. In *Proceedings of the 17th ACM international conference on web search and data mining* (pp. 576–585).

Sánchez, P. I., Müller, E., Laforet, F., Keller, F., & Böhm, K. (2013). Statistical selection of congruent subspaces for mining attributed graphs. In *2013 IEEE 13th international conference on data mining* (pp. 647–656). http://dx.doi.org/10.1109/ICDM.2013.88.

Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., & Vandergheynst, P. (2012). Signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular data domains. CoRR abs/1211.0053, arXiv:1211.0053.

Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., & Vandergheynst, P. (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, *30*(3), 83–98.

Spinelli, I., Scardapane, S., & Uncini, A. (2022). A meta-learning approach for training explainable graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(4), 4647–4655.

Tang, J., Li, J., Gao, Z., & Li, J. (2022). Rethinking graph neural networks for anomaly detection. arXiv preprint arXiv:2205.15508.

Tang, L., & Liu, H. (2009). Relational learning via latent social dimensions. In J. F. E. IV, F. coise Fogelman-Soulié, P. A. Flach, & M. J. Zaki (Eds.), *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 817–826). ACM, http://dx.doi.org/10.1145/1557019.1557109.

Taubin, G. (1995). A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on computer graphics and interactive techniques* (pp. 351–358).

Ten, C.-W., Hong, J., & Liu, C.-C. (2011). Anomaly detection for cybersecurity of the substations. *IEEE Transactions on Smart Grid*, *2*(4), 865–873.

Thakoor, S., Tallec, C., Azar, M. G., Azabou, M., Dyer, E. L., Munos, R., et al. (2021). Large-scale representation learning on graphs via bootstrapping. arXiv preprint arXiv:2102.06514.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2017). Graph attention networks. http://dx.doi.org/10.48550/ARXIV.1710.10903, URL: https://arxiv.org/abs/1710.10903.

Veličković, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., & Hjelm, R. D. (2018). Deep graph infomax. arXiv preprint arXiv:1809.10341.

Wang, X., Jin, B., Du, Y., Cui, P., Tan, Y., & Yang, Y. (2021). One-class graph neural networks for anomaly detection in attributed networks. *Neural Computing and Applications*, *33*, 12073–12085.

Wang, X., & Zhang, M. (2022). How powerful are spectral graph neural networks. *ICML*.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(1), 4–24.

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.

Yuan, X., Zhou, N., Yu, S., Huang, H., Chen, Z., & Xia, F. (2021). Higher-order structure based anomaly detection on attributed networks. In *2021 IEEE international conference on big data* (pp. 2691–2700). IEEE.

Zheng, Y., Jin, M., Liu, Y., Chi, L., Phan, K. T., & Chen, Y.-P. P. (2021). Generative and contrastive self-supervised learning for graph anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., et al. (2020). Graph neural networks: A review of methods and applications. *AI Open*, *1*, 57–81.

Zhou, S., Tan, Q., Xu, Z., Huang, X., & Chung, F.-l. (2021). Subtractive aggregation for attributed network anomaly detection. In *Proceedings of the 30th ACM international conference on information & knowledge management* (pp. 3672–3676).

Zhu, R., Tao, Z., Li, Y., & Li, S. (2021). Automated graph learning via population based self-tuning GCN. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval* (pp. 2096–2100).