NAAST-GNN: Neighborhood Adaptive Aggregation and Spectral Tuning for Graph Anomaly Detection

Ronghui Guo , **Xiaowang Zhang** , **Zhizhi Yu**[*] , **Minghui Zou** , **Sai Zhang** and **Zhiyong Feng**
College of Intelligence and Computing, Tianjin University, Tianjin, China
{ronghui_guo, xiaowangzhang, yuzhizhi, minghuizou, zhang_sai, zyfeng}@tju.edu.cn

Abstract

Heterophily emerges as a critical challenge in Graph Anomaly Detection (GAD). Recent studies reveal that neighborhood distributions, rather than heterophily itself, are the fundamental factor for the expressive power of Graph Neural Networks (GNNs). However, two key challenges remain unresolved. First, the overlap in neighborhood distributions between anomalous and normal nodes poses significant difficulties in distinguishing them effectively. Second, the dispersion in neighborhood distributions within the same class prevents the application of a fixed aggregation strategy to accommodate the diverse patterns within the class. To tackle the aforementioned challenges, we propose a novel Graph Neural Network model called Neighborhood Adaptive Aggregation and Spectral Tuning (NAAST-GNN). Specifically, we first design a neighborhood adaptive aggregation module that adjusts the message passing mechanism based on the predicted probabilities for different node classes, ensuring that nodes from distinct classes but with similar neighborhood distributions derive unique aggregated neighborhood information. We then present a spectral tuning module that dynamically selects and combines spectral filters based on the predicted neighborhood distribution, ensuring adaptability to the diverse neighborhood distributions of nodes within the same class. Comprehensive experimental results demonstrate that our method outperforms state-of-the-art baselines.

1 Introduction

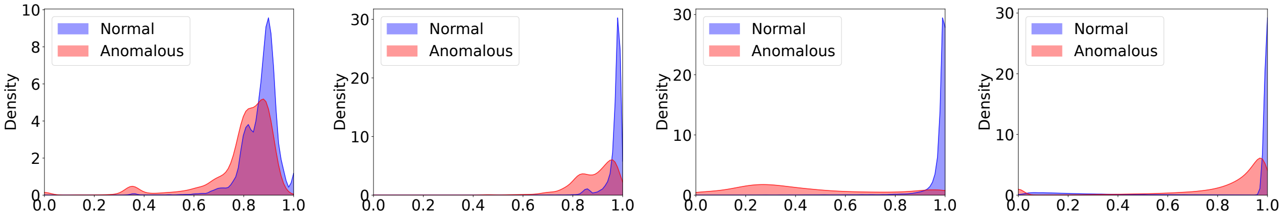
With the rapid advancements in information technology, anomaly detection has become crucial in areas such as network security [Zhang *et al.*, 2019], financial transactions [Lin *et al.*, 2021], and social networks [Deng *et al.*, 2023]. However, traditional anomaly detection methods struggle to keep pace with the growing complexity and scale of graph-structured data. Graphs, by effectively capturing intricate relationships between data points, have emerged as a powerful tool for anomaly detection. In particular, the rise of Graph Neural Networks (GNNs) has attracted significant attention, as they offer strong representation learning capabilities, making them ideal for addressing the difficulties of Graph Anomaly Detection (GAD).

Recent studies [Gao *et al.*, 2023a; Wang *et al.*, 2023; Xu *et al.*, 2024] emphasize the significance of heterophily in anomaly graphs. They observe that anomalous nodes in such graphs exhibit a high degree of heterophily, which makes GNNs based on the homophily assumption [McPherson *et al.*, 2001] unsuitable for capturing this characteristic, resulting in suboptimal performance. However, some works [Ma *et al.*, 2022; Luan *et al.*, 2022; Chen *et al.*, 2023a] argue that heterophily does not necessarily constrain the expressive power of GNNs and may not always be detrimental. Further studies [Luan *et al.*, 2023; Gao *et al.*, 2024; Wang *et al.*, 2024] analyze heterophily through the lens of neighborhood distributions, which typically quantified as the probability distribution of neighbor categories, revealing a positive correlation between node distinguishability after graph convolution and the Euclidean distance between neighborhood distributions. In essence, the core challenge of heterophily stems from neighborhood distributions, which fundamentally hinder GNNs performance in GAD. This limitation arises from the message-passing mechanism of GNNs, where the semantic information learned for a node is inherently influenced by the category distribution of its neighbors.

Building on this foundation, we revisit GAD methodologies from the perspective of neighborhood distributions in both spatial and spectral domains. Spatial methods [Zhuo *et al.*, 2024; Liu *et al.*, 2021] focus on edge trimming or neighbor aggregation mechanisms, which essentially aim to enhance the distinguishability of neighborhood distrubutions between anomalous and normal nodes during the message-passing process of GNNs, enabling the learning of distinguishable representations for different classes (normals and anomalies). Spectral methods [Xu *et al.*, 2024; Tang *et al.*, 2022], since nodes with similar neighborhood distrubutions but different classes exhibit different frequency components [Gao *et al.*, 2024], employ polynomial filters to retain the high-frequency signals of anomalous nodes and the low-frequency signals of normal nodes.

Although previous methods have made some progress, as

[*] Corresponding author



(a) YelpChi (b) Amazon (c) T-Finance (d) T-Social

Figure 1: The neighborhood distribution when the neighboring nodes are normal, *i.e.*, ND_0^v.

shown in Figure 1, two key challenges remain: (1) **Overlap of Inter-class Neighborhood Distrubutions**: There is significant overlap in the neighborhood distrubutions between normal and anomalous nodes, making it difficult for GNNs to learn distinguishable representations for different classes. Existing methods [Liu *et al.*, 2021; Dou *et al.*, 2020] that determine message passing between nodes based on features can easily lead to errors being propagated to the results, causing suboptimal performance. (2) **Dispersion of Intra-class Neighborhood Distrubutions**: The intra-class neighborhood distribution exhibit significant dispersion in some cases. Fixed GNN aggregation strategies or filters struggle to adapt to such complex scenarios, causing learned representations within the same class to still demonstrate differences. Current approaches [Tang *et al.*, 2022; Gao *et al.*, 2024] fail not specifically address the diverse neighborhood distrubutions within the same class of nodes. Overall, the impact of these two challenges varies across different datasets. For instance, for YelpChi, it is primarily the former, whereas for T-Finance, it is mainly the latter.

To address the aforementioned challenges, we propose the GNN with **N**eighborhood **A**daptive **A**ggregation and **S**pectral **T**uning for graph anomaly detection (NAAST-GNN). The core idea is to dynamically adjust graph convolution propagation mechanisms based on node-specific predicted probabilities. Each layer consists of two key modules: The neighborhood adaptive aggregation module operates in the spatial domain, addressing the challenge of overlapping neighborhood distributions between anomalous and normal nodes. It constructs edge weights based on the predicted probabilities of nodes from the previous epoch, emphasizing a dynamic adjustment of edge weights. Specifically, this module assigns higher weights to homophilous edges and lower weights to heterophilous ones, ensuring that the aggregated neighborhood information remains distinct and discriminative, even when nodes from different classes share similar neighborhood distributions. The spectral tuning module operates in the spectral domain to address the dispersion in neighborhood distributions within the same class. By dynamically selecting and combining spectral filters based on the predicted neighborhood distribution, this module ensures adaptability to the diverse neighborhood characteristics within the same class. Finally, the learned spatial and spectral representations are integrated as inputs for the next layer.

The main contributions of this paper are summarized as:

- We identify two key challenges in GAD: the overlap of intra-class neighborhood distributions and the dispersion of inter-class neighborhood distributions.
- We propose the GNN with neighborhood adaptive aggregation and spectral tuning for graph anomaly detection (NAAST-GNN), which dynamically adjusts the graph convolution propagation mechaism based on predicted node probabilities.
- Extensive experiments on four public datasets demonstrate that our proposed method NAAST-GNN outperforms the state-of-the-art baselines.

2 Preliminaries

We present the notations and problem definition, and then delve into the concepts of homophily and heterophily, along with an analysis of neighborhood distributions.

2.1 Notations and Problem Definition

An anomaly graph is denoted as $\mathcal{G} = \{\mathcal{V}, X, \mathcal{E}\}$, where $\mathcal{V}$ is the node set with size $N = |\mathcal{V}|$, $\mathcal{E}$ is the edges set, and $X \in \mathbb{R}^{N \times d}$ denotes the node attributes matrix, with $X_v \in \mathbb{R}^d$ representing the attribute vector of node v.

The GAD is usually formalized as a binary node classification task (normal and anomalous), where normal nodes are labeled as 0 and anomalous nodes as 1. Given a graph $\mathcal{G}$, the target of GAD is to train a classifier $f(\cdot)$ that can infer the labels of unlabeled nodes based on the known labels of some nodes. This process can be represented as:

$$f(\mathcal{G}, Y_{Train}) \rightarrow \hat{Y}_{Test}. \tag{1}$$

2.2 Homophily/Heterophily and Neighborhood Distribution

The homophily coefficient H^v_{node} is usually used to measure the degree of homophily of the node v, defined as:

$$H^v_{\text{node}} = \frac{|\{u \mid u \in \mathcal{N}_v, y_u = y_v\}|}{|\mathcal{N}_v|}, \tag{2}$$

where $\mathcal{N}_v$ denotes the set of neighbors of node v. Thus, the heterophily of the node v can be defined as $1 - H^v_{\text{node}}$.

Then, the Neighborhood Distribution (ND) of a node v, which represents the probability of its neighboring nodes u belonging to category c, can be calculated as:

$$ND^v_c = \frac{|\{u \mid u \in \mathcal{N}_v, y_u = c\}|}{|\mathcal{N}_v|}. \tag{3}$$

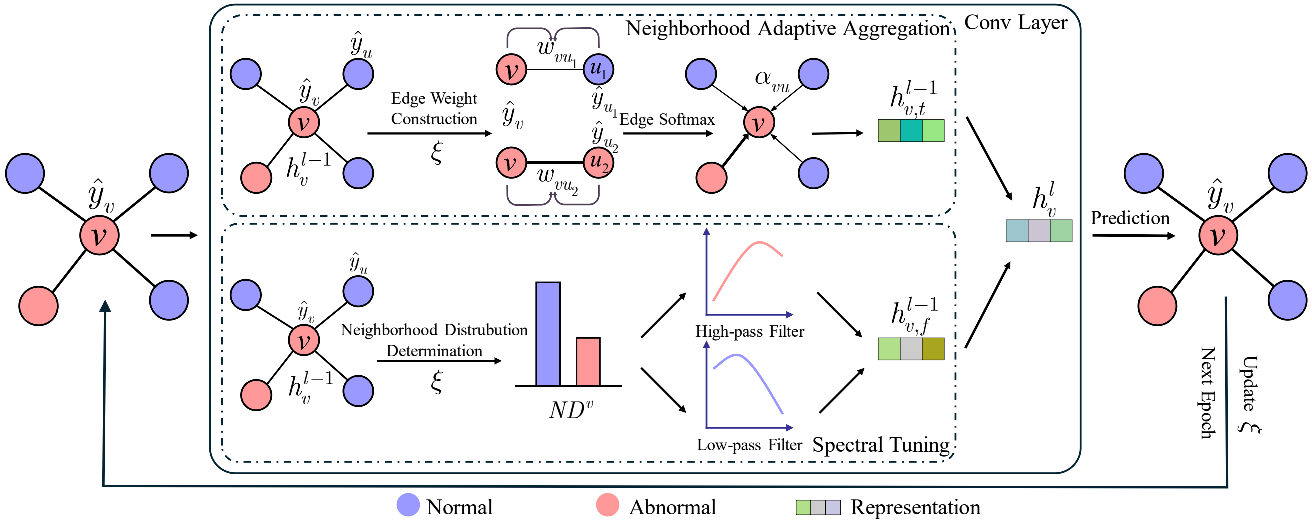


Figure 2: The architecture of the proposed NAAST-GNN.

Figure 1 shows the neighborhood distribution of each dataset when the neighboring nodes are normal, *i.e.*, ND_0^v. Since GAD involves only two types of nodes, the neighboring anomalous node case, *i.e.*, ND_1^v, equals $1 - ND_0^v$, and its illustration is omitted here due to space constraints.

3 Methodology

We first provide an overview of the proposed NAAST-GNN, followed by a detailed description of its two key components: the neighborhood adaptive aggregation and spectral tuning module. Finally, we discuss the model optimization.

3.1 Overview

Figure 2 illustrates the detailed architecture of our proposed NAAST-GNN. The model operates iteratively, where the message-passing mechanism of the graph convolution relies on the node prediction probabilities from the previous epoch. Each convolutional layer consists of two core modules: the neighborhood adaptive aggregation and spectral tuning module. To address the issue of overlapping neighborhood distributions between anomalous and normal nodes, the neighborhood adaptive aggregation module constructs edge weights in the graph based on the predicted node probabilities. This ensures that nodes with similar neighborhood distributions but belonging to different classes aggregate distinct neighborhood information. Then, to tackle the challenge posed by significant intra-class variations in neighborhood distributions, the spectral tuning module dynamically combines different spectral information. This approach adapts to the diverse neighborhood characteristics of nodes, enabling the model to handle various neighborhood situations effectively. The two modules collaboratively address these challenges from spatial and spectral perspectives, respectively. Finally, the aggregated representations are transformed through a multilayer perceptron (MLP), generating updated representations that serve as input for the subsequent layer.

3.2 Neighborhood Adaptive Aggregation Module

Most traditional GNNs are based on the homophily assumption [McPherson *et al.*, 2001], which indiscriminately aggregates information from all neighbors. However, the neighborhood distributions of normal and anomalous nodes overlap significantly in GAD. This causes traditional GNNs to learn representations that fail to effectively distinguish between normal and anomalous nodes. To this end, we propose a neighborhood adaptive aggregation strategy that selectively aggregates information from relevant neighbors. Specifically, we utilize the classification probabilities $\hat{y}$ of the nodes at both ends of an edge from the previous epoch to compute the edge weights. This approach ensures that, for any given node, edges connecting neighbors of the same class are assigned larger weights, while edges connecting neighbors of different classes are assigned smaller weights. As a result, our method facilitates the learning of more distinguishable representations, even when the neighborhood distributions of different classes exhibit significant similarity.

First, for any central node v, we determine its category based on the comparison between its predicted probability $\hat{y}_v$ in the previous epoch and the threshold ξ. If the probability is less than the adaptive adjustment parameter ξ, the instance is classified as normal; otherwise, it is considered an anomaly. Then, we adopt different strategies to construct edge weights based on their categories:

$$w_{vu} = \begin{cases} \hat{y}_v \hat{y}_u, & \hat{y}_v \geqslant \xi \\ (1 - \hat{y}_v)(1 - \hat{y}_u), & \hat{y}_v < \xi \end{cases}. \tag{4}$$

Empirically, anomalous nodes tend to exhibit $\hat{y} \approx 1$, while normal nodes usually have $\hat{y} \approx 0$. When a central node v is classified as anomalous (*i.e.*, $\hat{y}_v \geq \xi$), the edge weight is defined as $w_{vu} = \hat{y}_v \hat{y}_u$, which assigns higher weights to edges connecting v with anomalous neighbors (where $\hat{y}_u \approx 1$), and lower weights to edges connecting v with normal neighbors (where $\hat{y}_u \approx 0$). Conversely, when v is classified as a normal node (*i.e.*, $\hat{y}_v < \xi$), the weight is defined as

$w_{vu} = (1-\hat{y}_v)(1-\hat{y}_u)$, thereby strengthening the connections with other normal neighbors and weakening those with anomalous ones. By constructing edge weights in this manner, even if the neighborhood distrubutions of anomalous and normal nodes overlap, the edge weights built for these two types of nodes are completely different, enabling the GNN to effectively handle different types of node scenarios.

Since $\hat{y} \in (0,1)$, to prevent gradient vanishing and improve training robustness, we normalize the edge weights of the central node v using edge softmax:

$$\alpha_{vu} = \frac{\exp(w_{vu})}{\sum_{u\in\mathcal{N}(v)} \exp(w_{vu})}. \tag{5}$$

Finally, based on the normalized edge weights α_{vu}, the final spatial domain representation is obtained:

$$h_{v,t}^{l-1} = h_v^{l-1} + W^{l-1} \sum_{u\in\mathcal{N}_v} \alpha_{vu} h_u^{l-1}, \tag{6}$$

where h_v^{l-1} represents the embedding of node v in layer $l-1$, with $h_v^0 = X_v$, and W^{l-1} is a learnable parameter matrix.

3.3 Spectral Tuning Module

Considering that even among nodes belonging to the same category, substantial differences in neighborhood distributions may exist. Previous approaches [Tang *et al.*, 2022; Zhuo *et al.*, 2024] often rely on predefined spectral convolution kernels or fixed aggregation strategies, which fail to adapt effectively to the unique neighborhood distributions of individual nodes. To this end, we propose a method to design convolution kernels in the spectral domain, tailored to the neighborhood distributions of the nodes.

According to [Gao *et al.*, 2024; Wang *et al.*, 2024], the distinguishability of node representations of different classes after applying GNNs is positively correlated with the Euclidean distance between their neighborhood distrubutions. Simultaneously, nodes of different classes with similar neighborhood distrubutions will retain different frequency components. Therefore, we can infer that for a given node, if there are more nodes of the same class in its neighborhood, more low-frequency information should be retained. Conversely, if there are more nodes of different classes in its neighborhood, more high-frequency information should be retained. Thus, the spectral tuning module will adjust the low- and high-frequency components in the spectral filters according to the node-level neighborhood distrubutions.

Specifically, we first determine the neighborhood distribution of each node v based on the predicted probability of the node in the previous epoch:

$$ND^v = \begin{cases} ND_{c=1}^v = \frac{\sum_{u\in\mathcal{N}_v} \mathbb{I}(\hat{y}_u \geqslant \xi)}{|\mathcal{N}_v|} \\ ND_{c=0}^v = \frac{\sum_{u\in\mathcal{N}_v} \mathbb{I}(\hat{y}_u < \xi)}{|\mathcal{N}_v|} \end{cases}, \tag{7}$$

where the indicator function $\mathbb{I}(\cdot)$ equals 1 if the condition inside the parentheses is true, and 0 otherwise. Here again, we utilize the relationship between the node probability $\hat{y}_u$ and the adaptive adjustment parameter ξ to determine the neighborhood distrubution of nodes.

Then, based on the previous analysis, we adjust the spectral information of the filter according to the category of node v and its neighborhood distrubution. The final learned node representation in the spectral domain is:

$$h_{v,f}^{l-1} = \begin{cases} ND_{c=0}^v HP(h_v^{l-1}) + ND_{c=1}^v LP(h_v^{l-1}), & \hat{y}_v \geqslant \xi \\ ND_{c=0}^v LP(h_v^{l-1}) + ND_{c=1}^v HP(h_v^{l-1}), & \hat{y}_v < \xi \end{cases}, \tag{8}$$

where normalized Laplacian matrix L and its complement $I-L$ are employed here as high- and low-pass filters [Wang and Zhang, 2022], respectively. It is worth mentioning that other high- and low-pass filters can also be utilized in this context.

We analyze the case when the central node v is anomalous, *i.e.*, when $\hat{y}_v \geqslant \xi$. The more neighbors of the same type it has, *i.e.*, the higher the anomalous distribution of the neighborhood distrubution ND_1^v is, according to the previous analysis, the more low-frequency information should be retained, making node v more similar to its neighbors. For the node v being normal, the situation is similar, but the retained spectral information is exactly the opposite. Through the spectral tuning module, we can ensure that nodes of the same class are well-tuned to their respective spectral information even when they have different neighborhood distrubutions.

Finally, the spatial representation $h_{v,t}^{l-1}$ from the neighborhood adaptive aggregation module and the spectral representation $h_{v,f}^{l-1}$ from the spectral tuning module are transformed as the input for the next layer:

$$h_v^l = \text{MLP}\left[h_{v,t}^{l-1} || h_{v,f}^{l-1}\right], \tag{9}$$

where $[\cdot||\cdot]$ denotes the concatenation operation, and MLP transforms the concatenated representations.

3.4 Model Optimization

We utilize the node representation h_v^L from the final layer of the model to perform classification, generating a predicted value $\hat{y}_v$, which represents the probability of node v being classified as anomalous:

$$\hat{y}_v = \text{Sigmoid}\left(h_v^L W + b\right), \tag{10}$$

where W and b represent a learnable parameter matrix and bias, respectively. Our model adopts an iterative optimization strategy, where $\hat{y}_v$ serves as both the output of the current epoch and the input for the next epoch.

Then, to mitigate the class imbalance between normal and anomalous nodes, we minimize the weighted cross-entropy loss, defined as:

$$Loss = -\sum_{v\in\mathcal{V}} [\gamma y_v \log \hat{y}_v + (1-y_v)\log(1-\hat{y}_v)]. \tag{11}$$

where γ is the ratio of the number of normals to anomalies.

During the model optimization process, the classification probabilities $\hat{y}_v$ of the nodes are not available during the first epoch of training. For this purpose, we initialize the probabilities randomly for all nodes. In subsequent epochs, as the training loss decreases, the prediction probabilities are iteratively refined, becoming increasingly accurate with each epoch. The predictive objectives and the model's sophisticated design synergize effectively: higher prediction accuracy enhances the functionality of the neighborhood adaptive

Dataset	#Nodes	Anomaly(%)	#Edges	#Features
YelpChi	45,954	14.53	3,846,979	32
Amazon	11,944	6.87	4,398,392	25
T-Finance	39,357	4.58	21,222,543	10
T-Social	5,781,065	3.01	73,105,508	10

Table 1: Statistics of datasets.

aggregation and spectral tuning modules, while improved capabilities in these modules, in turn, boost the model's overall prediction accuracy. This reciprocal relationship ultimately enhances the model's performance.

Finally, for the parameter ξ, we initialize it to 0.5 at the start of the first epoch. During model training, its value is dynamically adjusted to the threshold that maximizes the F1-Macro score on the validation set [Tang *et al.*, 2022].

4 Experiments

In this section, we perform experiments on four datasets and report results of our models and state-of-the-art baselines to show the effectiveness of NAAST-GNN.

4.1 Experimental Setup

Dataset. We conduct experiments on four datasets related to graph anomaly detection, the statistics of which are shown in Table 1. Among them, YelpChi [Rayana and Akoglu, 2015] and Amazon [McAuley and Leskovec, 2013] are two public spam review datasets for opinion fraud detection. YelpChi aims to identify reviews that unfairly promote or disparage specific products or businesses. Amazon focuses on detecting users who provide deceptive product reviews in the musical instruments category on the Amazon website and get paid for it. T-Finance and T-Social [Tang *et al.*, 2022] aim to identify anomalous accounts in transaction networks and social networks, respectively. They share 10-dimensional features related to the number of days since registration, login activity, and interaction frequency. In T-Finance, edges represent transaction records between accounts. If a node is associated with categories such as fraud, money laundering, or online gambling, human experts label it as anomalous. In T-Social, nodes are connected if they have maintained a friendship for over three months.

Baselines. We evaluate NAAST-GNN against state-of-the-art baselines, which are categorized into three groups. The first group comprises conventional GNNs, including GCN [Kipf and Welling, 2017], GraphSAGE [Hamilton *et al.*, 2017], and GAT [Velickovic *et al.*, 2018]. The second group focuses on spectral heterophilic methods, such as AMNet [Chai *et al.*, 2022], LSGNN [Chen *et al.*, 2023b], BWGNN [Tang *et al.*, 2022], GHRN [Gao *et al.*, 2023a], SEC-GFD [Xu *et al.*, 2024], and BioGNN [Gao *et al.*, 2024]. The third group encompasses spatial heterophilic methods, including CARE-GNN [Dou *et al.*, 2020], PC-GNN [Liu *et al.*, 2021], H2-FDetector [Shi *et al.*, 2022], GDN [Gao *et al.*, 2023b], GAGA [Wang *et al.*, 2023], CONSISGAD [Chen *et al.*, 2024], and PMP [Zhuo *et al.*, 2024].

Metrics. Since GAD is a class-imbalanced classification task, we select two widely used evaluation metrics: F1-Macro and AUC. F1-Macro calculates the mean F1 score across all classes, ensuring equal significance to each class by balancing precision and recall. AUC quantifies a model's capacity to differentiate between positive and negative examples through the area under the ROC curve. Higher values for these two metrics indicate better performance.

Implementation Details. The training, validation, and test sets for all datasets are split in a 0.4/0.2/0.4 ratio, with experimental results reported as the average of five runs. For our proposed NAAST-GNN, the layer number is set to 2. The hidden layer dimension is set to 32 for YelpChi, Amazon, and T-Finance. Due to the large scale of the T-Social, its hidden layer dimension is set to 10. The choice of parameter ξ is discussed at the end of Section 3.4. We use the Adam optimizer and tune the hyperparameters using grid search, with the learning rate ranging from {5e-4, 1e-3, 5e-3, 1e-2} and the weight decay ranging from {1e-3, 1e-4, 1e-5}. For GCN, GAT, and GraphSAGE, we implement these models using DGL [Wang *et al.*, 2019b]. For other baseline methods, we utiliz the source code provided by their respective authors. Due to the unavailability of the source code for SEC-GFD and BioGNN at the time of manuscript submission and our experimental settings are consistent with those outlined in their original publications, we present the results directly from the respective papers.

4.2 Experimental Results

We conduct a comprehensive comparison of our method against homophilous GNN models, state-of-the-art spectral and spatial heterophilous GNNs. The results are summarized in Table 2. Traditional GNNs based on the homophily assumption perform poorly on heterophilous anomaly graphs due to their inability to effectively model heterophilous structures. Recent spectral and spatial heterophilous GNNs introduce mechanisms to address heterophily, achieving some degree of performance improvement. However, they fail to fully consider the key factor of neighborhood distribution, which led to suboptimal performance.

Our proposed model consistently achieves superior performance in most cases. Specifically, NAAST-GNN outperforms the best baseline methods on YelpChi, T-Finance, and T-Social by 1.48%, 1.20%, and 4.26% in terms of Fl-macro, and by 1.19%, 0.74%, and 2.17% in terms of AUC, respectively. These results underscore the efficacy of our neighborhood adaptive aggregation and spectral tuning modules in addressing the challenges posed by inter-class neighborhood distribution overlap and intra-class neighborhood distribution dispersion. Notably, the performance of NAAST-GNN on the Amazon dataset is relatively less competitive, ranking as the second best. This may be attributed to the complex structural characteristics of Amazon, where excessive interaction complexity partially hampers the model's ability to distinguish neighborhood distributions effectively.

4.3 Ablation Study

To better verify the effectiveness of each module in NAAST-GNN, we conduct ablation studies on the neighborhood adap-

Dataset	YelpChi		Amazon		T-Finance		T-Social	
Metric	F1-Macro	AUC	F1-Macro	AUC	F1-Macro	AUC	F1-Macro	AUC
GCN	55.32	58.15	64.61	82.11	70.58	63.44	63.35	79.78
GraphSAGE	60.44	69.98	66.53	78.85	53.70	69.15	61.71	74.01
GAT	57.07	60.15	68.67	89.59	55.77	76.61	73.60	88.54
AMNet	69.31	84.17	91.82	88.62	88.19	93.72	77.76	93.76
LSGNN	72.27	82.49	91.05	96.45	87.96	94.37	90.28	96.94
BWGNN	76.21	90.16	91.51	97.48	88.34	95.13	90.10	95.96
GHRN	77.95	90.97	92.32	97.63	89.64	96.70	91.53	96.55
SEC-GFD*	77.73	91.39	92.35	**98.23**	89.86	96.32	87.74	96.11
BioGNN*	76.32	89.20	**93.68**	97.48	90.47	96.39	81.40	93.25
CARE-GNN	64.20	79.73	87.03	92.27	73.81	92.26	55.20	85.03
PC-GNN	65.79	89.55	88.43	96.44	62.69	91.28	47.76	87.63
H2-FDetector	69.37	88.33	83.87	96.34	OOM		OOM	
GDN	76.34	90.08	90.13	96.89	84.89	93.27	74.25	93.52
GAGA	80.02	91.85	91.58	96.75	90.25	95.46	OOT	
CONSISGAD	76.43	90.57	92.26	97.41	91.08	96.21	83.60	95.77
PMP	80.17	91.77	91.95	97.61	91.26	96.29	OOT	
NAAST-GNN	**81.36**	**92.95**	92.48	97.82	**92.36**	**97.11**	**95.43**	**98.65**

Table 2: Performance Results (%). The best results for all methods are in bold and the second best results are underlined. Results marked with "*" are reproduced from original publications. OOM: out of memory; OOT: out of time (running time > 1 day).

tive aggregation (w/o NAA) and spectral tuning (w/o ST) modules. Additionally, for our proposed neighborhood adaptive aggregation, we replace it with GAT (w/ GAT), which dynamically aggregates neighborhood information through an attention mechanism based on node features. For the spectral tuning module, we design two variants, which replace it with situations that only retain high-frequency information (w/ HP) or low-frequency information (w/ LP), respectively.

As shown in Table 3, NAAST-GNN achieves the best performance in its complete state, indicating that each module contributes to its effectiveness. Compared to YelpChi and T-Social, we observe that for YelpChi, the removal of the NAA module has a more significant impact on performance than the removal of the ST module. In contrast, for T-Social, the removal of the ST module has a greater impact than that of the NAA module. This disparity can be attributed to the differing primary factors influencing their performance: for YelpChi, it is the overlap of inter-class neighborhood distrubutions, whereas for T-Social, it is the dispersion of intra-class neighborhood distrubutions. These findings further validate that the proposed modules are specifically designed to address these distinct challenges.

4.4 Analysis of Training Ratio

In real-world scenarios, we may not obtain enough labeled data. Therefore, we explore the impact of the training ratio on model performance. Table 4 shows the performance of NAAST-GNN on YelpChi and T-Social with training data ranging from 1% to 40%. According to [Tang *et al.*, 2022], the validation and test set split ratio remains 1:2.

From the results in Table 4, it can be seen that as the training ratio increases, the model performance improves significantly for YelpChi. However, the improvement for T-Social is limited. We believe the possible reason is that in YelpChi, the neighborhood distrubutions of anomalous and normal nodes overlap too much, as shown in Figure 1(a), and the model requires more supervisory signals for training. The neighborhood distrubutions between classes in T-Social are quite different, as shown in Figure 1(d), making it relatively simpler. The main issue is the large intra-class neighborhood distrubution disparity, which our spectral tuning module can effectively address. Therefore, even with extremely limited training data (1%), NAAST-GNN still outperforms other graph anomaly detectors (using 40% training data in Table 2).

Dataset	YelpChi		T-Social	
Metric	F1-Macro	AUC	F1-Macro	AUC
NAAST-GNN	81.36	92.95	95.43	98.65
w/o NAA	75.88	85.76	93.28	98.73
w/ GAT	78.14	90.73	93.91	98.73
w/o ST	77.28	89.92	79.41	96.41
w/ HP	79.21	90.95	93.95	99.05
w/ LP	80.37	91.64	93.15	98.72

Table 3: The ablation study of NAAST-GNN.

4.5 Visualization and Discussion

To intuitively illustrate the effectiveness of NAAST-GNN, we present a visualization of a portion of the convolution process of our model in Figure 3. As shown in the figure, the weights assigned to heterophilous connections are notably smaller compared to those of homophilous connections during the neighborhood adaptive aggregation. This mechanism enables the model to efficiently aggregate information from similar

Dataset	YelpChi		T-Social	
Metric	F1-Macro	AUC	F1-Macro	AUC
1%	69.12	80.82	94.01	97.90
5%	73.21	85.34	94.35	98.16
10%	75.17	86.94	94.54	98.19
20%	77.66	89.54	94.61	98.24
30%	78.40	90.83	95.16	98.47
40%	81.36	92.95	95.43	98.65

Table 4: The analysis of training ratio.

nodes, effectively mitigating the issue of excessive overlap in inter-class neighborhood distributions. Furthermore, our approach accurately captures the neighborhood distributions of nodes, facilitating the integration of low- and high-frequency information based on the node's specific neighborhood characteristics. This design addresses the challenge of substantial intra-class neighborhood distribution discrepancies. In summary, our method strategically leverages node prediction probabilities, ensuring that the graph convolution process and prediction objectives complement each other. This synergistic design results in significant performance improvements.

5 Related Work

This section introduces related work on GAD, as well as recent research on heterophily and neighborhood distribution.

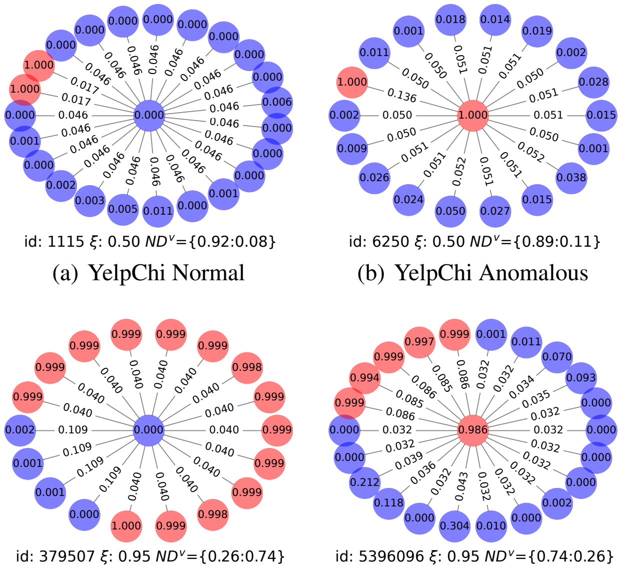
5.1 Graph Anomaly Detection

Early works [Wang *et al.*, 2019a; Liu *et al.*, 2020] focus on leveraging multiple relationships to aggregate information. However, indiscriminate aggregation of all information cannot lead to the differentiation of anomalous nodes. Furthermore, CARE-GNN [Dou *et al.*, 2020], PC-GNN [Liu *et al.*, 2021], H2-FDetector [Shi *et al.*, 2022], and GAGA [Wang *et al.*, 2023] selectively aggregate neighbor information by designing unique GNN propagation mechanisms.

Additionally, some works address this task from the spectral domain, with the mainstream solution being the design of various spectral filters for denoising [Guo *et al.*, 2024]. BWGNN [Tang *et al.*, 2022] discovers that as the degree of anomaly increases, the spectral energy distribution shifts to the right. Building on BWGNN, GHRN [Gao *et al.*, 2023a] trims heterophilous edges based on spectral energy components. SEC-GFD [Xu *et al.*, 2024] extends BWGNN by incorporating missing high-frequency information. BioGNN [Gao *et al.*, 2024] analyzes the relationship between heterophily and neighborhood distribution, suggesting that nodes with similar neighborhood distrubutions but different classes should retain different frequency components.

5.2 Heterophily and Neighborhood Distrubution

Recent studies on heterophilous graphs reveal that the poor performance of GNNs is not inherently due to heterophily but rather the indistinguishability of neighborhood distributions. Early studies [Ma *et al.*, 2022; Luan *et al.*, 2022; Chen *et al.*, 2023a] found that heterophily is not always detrimental and that the distinguishability of nodes by GNNs is related to neighborhood distrubutions. [Mao *et al.*, 2023] shows that GNNs often rely on majority-class structural distributions (homophily or heterophily), leading to poor performance for minority-class nodes in GAD. Moreover, [Luan *et al.*, 2023] demonstrates that the superior performance of GNNs is related to whether the intra-class node distinguishability is less than the inter-class node distinguishability. Some studies [Gao *et al.*, 2024; Wang *et al.*, 2024] further proves that the distinguishability of nodes after the graph convolution operation is positively correlated with the Euclidean distance of neighborhood distrubutions. Additionally, [Huang *et al.*, 2024] shows from a spectral perspective that high-frequency signals are more favorable on heterogeneous graphs, and the ideal graph signal bases are obligated to incorporate the heterophily of the graph.


id: 1115 ξ: 0.50 ND^v={0.92:0.08}
id: 6250 ξ: 0.50 ND^v={0.89:0.11}
id: 379507 ξ: 0.95 ND^v={0.26:0.74}
id: 5396096 ξ: 0.95 ND^v={0.74:0.26}

(a) YelpChi Normal

(b) YelpChi Anomalous

(c) T-Social Normal

(d) T-Social Anomalous

Figure 3: Graph convolution visualization: Blue and red represent normal and anomalous nodes, respectively. The labels on the nodes are $\hat{y}$, and the numbers on the edges indicate the edge weights α_{vu} calculated by Equation (5). ND^v indicates the neighborhood distrubution determined by ξ.

6 Conclusion and Future Work

In this paper, we highlight the critical role of neighborhood distribution in graph anomaly detection. To address the key challenges of inter-class neighborhood distribution overlap and intra-class neighborhood distribution dispersion, we propose NAAST-GNN. It integrates two modules: neighborhood ndaptive aggregation and spectral tuning, which dynamically adjust graph convolutions to align with node-specific neighborhood characteristics. Extensive experiments conduct on four public datasets validate the effectiveness of our approach, demonstrating superior performance compared to state-of-the-art methods. Future work will explore the broader interplay between heterophily and neighborhood distributions and extend the model to dynamic graph scenarios.

Acknowledgments

This work is supported by the National Natural Science Foundation of China No. 62402337, the Postdoctoral Fellowship Program of CPSF under Grant No. GZC20241207, and the China Postdoctoral Science Foundation under Grant No. 2024M752367.

References

[Chai *et al.*, 2022] Ziwei Chai, Siqi You, Yang Yang, Shiliang Pu, Jiarong Xu, Haoyang Cai, and Weihao Jiang. Can abnormality be detected by graph neural networks? In *Proceedings of the 31st International Joint Conference on Artificial Intelligence*, pages 1945–1951, 2022.

[Chen *et al.*, 2023a] Jie Chen, Shouzhen Chen, Junbin Gao, Zengfeng Huang, Junping Zhang, and Jian Pu. Exploiting neighbor effect: Conv-agnostic gnn framework for graphs with heterophily. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–14, 2023.

[Chen *et al.*, 2023b] Yuhan Chen, Yihong Luo, Jing Tang, Liang Yang, Siya Qiu, Chuan Wang, and Xiaochun Cao. LSGNN: towards general graph neural network in node classification by local similarity. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*, pages 3550–3558, 2023.

[Chen *et al.*, 2024] Nan Chen, Zemin Liu, Bryan Hooi, Bingsheng He, Rizal Fathony, Jun Hu, and Jia Chen. Consistency training with learnable data augmentation for graph anomaly detection with limited supervision. In *Proceedings of the 12th International Conference on Learning Representations*, 2024.

[Deng *et al.*, 2023] Leyan Deng, Chenwang Wu, Defu Lian, Yongji Wu, and Enhong Chen. Markov-driven graph convolutional networks for social spammer detection. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12310–12322, 2023.

[Dou *et al.*, 2020] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S. Yu. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, pages 315–324, 2020.

[Gao *et al.*, 2023a] Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. Addressing heterophily in graph anomaly detection: A perspective of graph spectrum. In *Proceedings of the ACM Web Conference 2023*, pages 1528–1538, 2023.

[Gao *et al.*, 2023b] Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. Alleviating structural distribution shift in graph anomaly detection. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining*, pages 357–365, 2023.

[Gao *et al.*, 2024] Yuan Gao, Junfeng Fang, Yongduo Sui, Yangyang Li, Xiang Wang, Huamin Feng, and Yongdong Zhang. Graph anomaly detection with bi-level optimization. In *Proceedings of the ACM Web Conference 2024*, pages 4383–4394, 2024.

[Guo *et al.*, 2024] Ronghui Guo, Minghui Zou, Sai Zhang, Xiaowang Zhang, Zhizhi Yu, and Zhiyong Feng. Graph local homophily network for anomaly detection. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 706–716, 2024.

[Hamilton *et al.*, 2017] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 1024–1034, 2017.

[Huang *et al.*, 2024] Keke Huang, Yu Guang Wang, Ming Li, and Pietro Lio. How universal polynomial bases enhance spectral graph neural networks: Heterophily, oversmoothing, and over-squashing. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[Lin *et al.*, 2021] Wangli Lin, Li Sun, Qiwei Zhong, Can Liu, Jinghua Feng, Xiang Ao, and Hao Yang. Online credit payment fraud detection via structure-aware hierarchical recurrent neural network. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, pages 3670–3676, 2021.

[Liu *et al.*, 2020] Zhiwei Liu, Yingtong Dou, Philip S. Yu, Yutong Deng, and Hao Peng. Alleviating the inconsistency problem of applying graph neural network to fraud detection. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1569–1572, 2020.

[Liu *et al.*, 2021] Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. Pick and choose: A gnn-based imbalanced learning approach for fraud detection. In *Proceedings of the Web Conference 2021*, pages 3168–3177, 2021.

[Luan *et al.*, 2022] Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Revisiting heterophily for graph neural networks. In *Proceedings of the Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems*, 2022.

[Luan *et al.*, 2023] Sitao Luan, Chenqing Hua, Minkai Xu, Qincheng Lu, Jiaqi Zhu, Xiao-Wen Chang, Jie Fu, Jure Leskovec, and Doina Precup. When do graph neural networks help with node classification? investigating the homophily principle on node distinguishability. In *Proceedings of the Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems*, 2023.

[Ma *et al.*, 2022] Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. Is homophily a necessity for graph neural networks? In *Proceedings of the 10th International Conference on Learning Representations*, 2022.

[Mao *et al.*, 2023] Haitao Mao, Zhikai Chen, Wei Jin, Haoyu Han, Yao Ma, Tong Zhao, Neil Shah, and Jiliang Tang. Demystifying structural disparity in graph neural networks: Can one size fit all? In *Proceedings of the Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems*, 2023.

[McAuley and Leskovec, 2013] Julian J. McAuley and Jure Leskovec. From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd International World Wide Web Conference*, pages 897–908, 2013.

[McPherson *et al.*, 2001] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001.

[Rayana and Akoglu, 2015] Shebuti Rayana and Leman Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 985–994, 2015.

[Shi *et al.*, 2022] Fengzhao Shi, Yanan Cao, Yanmin Shang, Yuchen Zhou, Chuan Zhou, and Jia Wu. H2-fdetector: A gnn-based fraud detector with homophilic and heterophilic connections. In *Proceedings of the ACM Web Conference 2022*, pages 1486–1494, 2022.

[Tang *et al.*, 2022] Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. Rethinking graph neural networks for anomaly detection. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 21076–21089, 2022.

[Velickovic *et al.*, 2018] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

[Wang and Zhang, 2022] Xiyuan Wang and Muhan Zhang. How powerful are spectral graph neural networks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 23341–23362, 2022.

[Wang *et al.*, 2019a] Jianyu Wang, Rui Wen, Chunming Wu, Yu Huang, and Jian Xiong. Fdgars: Fraudster detection via graph convolutional networks in online app review system. In *Companion Proceedings of the 2019 World Wide Web Conference*, pages 310–316, 2019.

[Wang *et al.*, 2019b] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*, 2019.

[Wang *et al.*, 2023] Yuchen Wang, Jinghui Zhang, Zhengjie Huang, Weibin Li, Shikun Feng, Ziheng Ma, Yu Sun, Dianhai Yu, Fang Dong, Jiahui Jin, Beilun Wang, and Junzhou Luo. Label information enhanced fraud detection against low homophily in graphs. In *Proceedings of the ACM Web Conference 2023*, pages 406–416, 2023.

[Wang *et al.*, 2024] Junfu Wang, Yuanfang Guo, Liang Yang, and Yunhong Wang. Understanding heterophily for graph neural networks. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[Xu *et al.*, 2024] Fan Xu, Nan Wang, Hao Wu, Xuezhi Wen, Xibin Zhao, and Hai Wan. Revisiting graph-based fraud detection in sight of heterophily and spectrum. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, pages 9214–9222, 2024.

[Zhang *et al.*, 2019] Mingxue Zhang, Wei Meng, Sangho Lee, Byoungyoung Lee, and Xinyu Xing. All your clicks belong to me: Investigating click interception on the web. In *Proceedings of the 28th USENIX Security Symposium*, pages 941–957, 2019.

[Zhuo *et al.*, 2024] Wei Zhuo, Zemin Liu, Bryan Hooi, Bingsheng He, Guang Tan, Rizal Fathony, and Jia Chen. Partitioning message passing for graph fraud detection. In *Proceedings of the 12th International Conference on Learning Representations*, 2024.