Addressing Heterophily in Graph Anomaly Detection: A Perspective of Graph Spectrum

Yuan Gao
yuanga@mail.ustc.edu.cn
University of Science and Technology of China
Hefei, China

Xiang Wang*
xiangwang1223@gmail.com
University of Science and Technology of China
Hefei, China

Xiangnan He*
xiangnanhe@gmail.com
University of Science and Technology of China
Hefei, China

Zhenguang Liu
liuzhenguang2008@gmail.com
Zhejiang University
Hangzhou, China

Huamin Feng
oliver_feng@yeah.net
Beijing Electronic Science and Technology Institute
Beijing, China

Yongdong Zhang
zhyd73@ustc.edu.cn
University of Science and Technology of China
Hefei, China

ABSTRACT

Graph anomaly detection (GAD) suffers from heterophily — abnormal nodes are sparse so that they are connected to vast normal nodes. The current solutions upon Graph Neural Networks (GNNs) blindly smooth the representation of neiboring nodes, thus undermining the discriminative information of the anomalies. To alleviate the issue, recent studies identify and discard inter-class edges through estimating and comparing the node-level representation similarity. However, the representation of a single node can be misleading when the prediction error is high, thus hindering the performance of the edge indicator.

In graph signal processing, the smoothness index is a widely adopted metric which plays the role of frequency in classical spectral analysis. Considering the ground truth $\mathbf{Y}$ to be a signal on graph, the smoothness index is equivalent to the value of the heterophily ratio. From this perspective, we aim to address the heterophily problem in the spectral domain. First, we point out that heterophily is positively associated with the frequency of a graph. Towards this end, we could prune inter-class edges by simply emphasizing and delineating the high-frequency components of the graph. Recall that graph Laplacian is a high-pass filter, we adopt it to measure the extent of 1-hop label changing of the center node and indicate high-frequency components. As GAD can be formulated as a semi-supervised binary classification problem, only part of the nodes are labeled. As an alternative, we use the prediction of the nodes to estimate it. Through our analysis, we show that prediction errors are less likely to affect the identification process. Extensive empirical evaluations on four benchmarks demonstrate the effectiveness of the indicator over popular homophilic, heterophilic, and tailored fraud detection methods. Our proposed indicator can effectively reduce the heterophily degree of the graph, thus boosting the overall GAD performance. Codes are open-sourced in https://github.com/blacksingular/GHRN.

CCS CONCEPTS

• **Security and privacy → Web application security**; • **Computing methodologies → Neural networks**.

KEYWORDS

Misinformation Detection, Graph Anomaly Detection, Graph Neural Network, Heterophily

ACM Reference Format:
Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. 2023. Addressing Heterophily in Graph Anomaly Detection: A Perspective of Graph Spectrum. In *Proceedings of the ACM Web Conference 2023 (WWW '23), May 1–5, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3543507.3583268

1 INTRODUCTION

Anomaly detection is the task of identifying some rare objects (*aka.* anomalies) that deviate significantly from the majority of the corpus data (*aka.* normals) [27]. These objects carry vital information to support the analysis of fraudsters' behaviors. Hence anomaly detection has attracted considerable attention, such as identifying spam in reviews [12, 14, 15], misinformation in social networks [7], and frauds in financial transactions [24]. In the Web era, rich relationships between abnormal and normal objects become ubiquitous, which can be naturally organized as graphs [12]. Wherein, nodes represent these objects, and edges interpret their relationships. On such graphs, graph anomaly detection (GAD) is formulated as the semi-supervised classification problem, *i.e.,* transferring the discriminative information learned from a fraction of labeled anomalies to the vast remaining test data. As graph neural networks (GNNs) are powerful tools to address this issue, there has been interest in leveraging GNNs to solve GAD problems.

However, recent works [12, 24] realize that anomalies tend to have a high edge heterophily degree. **Heterophily** [32] indicates the phenomenon that edges connect the nodes from different classes

[*] Corresponding authors

WWW '23, May 1–5, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9416-1/23/04…$15.00
https://doi.org/10.1145/3543507.3583268

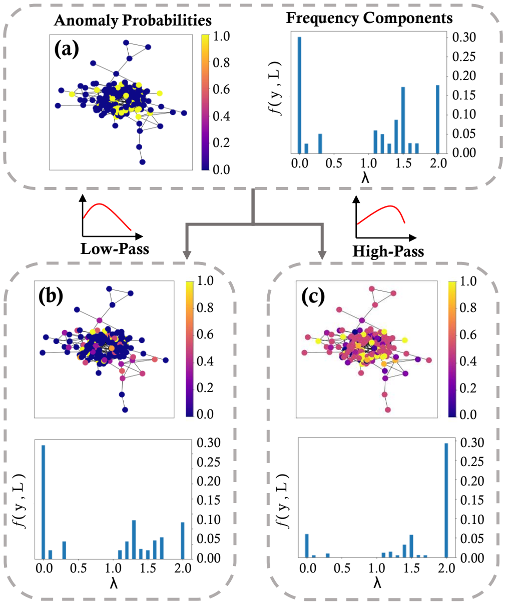


Figure 1: The Frequency Component Visualization, where the heatmap value indicates the probability of an anomaly node. The amplitude of each frequency component (λ_k) is measured by the corresponding *Spectral Energy Distribution* introduced on Page 4.

(*i.e.,* the anomaly and normal classes). Due to GAD's class imbalance nature, anomalies are always submerged in huge amounts of the normal neighborhood. As the plain GNNs force the representation of neighboring nodes to be similar, they could undermine the discriminative information of the anomalies. As shown in Figure 1(b), with the existence of inter-class edges, the decision boundary between the two classes becomes closer after aggregation through the plain GNNs (*i.e.,* low-pass filter). We attribute this performance degradation to the false mixing of nodes in different classes, which makes the anomalies indistinguishable and inundates the crucial cues from them. Hence it is of great importance to address the heterophily problem in GAD. Note that heterophily edges connect the nodes in different classes, for the rest of the paper, we use inter-class edges and heterophily edges interchangeably.

To alleviate the issue, the key lies in designing different strategies for homophilic connections and heterophilic connections [34, 36] or pruning inter-class edges directly [12, 24]. Several models [9, 12, 16, 23, 24, 34, 47] have been proposed, which can be roughly categorized into spatial and spectral branches. In the spatial domain, GNN models adopt multifarious mechanisms to delete the inter-class edges, including: graph attention mechanism [9, 23], task-specific optimization objectives [16], edge prediction models [16, 47], to name a few. However, most of the methods are based on the node-level representation similarity and are not robust. For example, in the case of adopting the Manhattan distance of two predicted representations as the similarity measure, the prediction error will be accumulated to the final result. Differently, from the perspective of graph spectrum, spectral GNNs [3, 8, 36] design adaptive filters or band-pass filters, and identify connections to which they assign different weights, regardless of the extent to which the node labels are homophilic or heterophilic [8]. However, existing works don't explicitly exhibit the exact semantics of high-frequency and low-frequency signals, which results in poor explainability and generalization. In addition, we show that heterophilic connections are useless even harmful in GAD, indicating denoising is better than adapting in this task.

In graph signal processing, the Rayleigh quotient (*i.e.,* $\frac{\mathbf{X}^T\mathbf{L}\mathbf{X}}{\mathbf{X}^T\mathbf{X}}$) defined on the laplacian matrix $\mathbf{L}$ and the signal $\mathbf{X}$ is widely adopted as the smoothness index which plays the role of frequency in classical spectral analysis [35]. As shown in Figure 1, we consider the ground-truth $\mathbf{Y}$ to be a signal on the graph where the smooth index equals to the heterophily ratio. In Figure 1(a), the anomalies are submerged in normal nodes, leading to high heterophily and large smoothness index. Since larger values of the smoothness index indicate faster-changing signals (high-frequency signals), we bridge the gap between the heterophilic connection and the vertex-frequency. From this perspective, we seek to explore the opportunity to prune inter-class edges with the spectral indication in a localized window of the node, rather than the node itself.

In addition, one recent work [36] discovers the "right-shift" phenomenon that the frequency content shifts to a higher frequency when the anomaly degree becomes larger. In light of this observation, a natural question is *"Shall heterophily be the implicit factor bridging the frequency and anomaly degree – The higher the fraction of the anomalies, the higher the heterophily, the higher the frequency of the graph?"* To validate the assumption, we explicitly represent heterophily in the spectral domain with the help of the smoothness index. Once the positive association of heterophily and frequency is confirmed, we can delete inter-class edges by simply delineating high-frequency components.

In this paper, we first take a closer look into the negative effect of heterophily, empirically showing that the GNN performance monotonically increases with the decrease of heterophily most of the time. We thus conjecture that a high-frequency edge indicator is quite useful when addressing heterophily for GAD. To support this argument, we theoretically prove that heterophily is positively associated with frequency, hence identifying inter-class edges resorts to extracting high-frequency signals. Guided by this idea, we propose Graph Heterophily Resistant Network (GHRN), which is equipped with a label-aware high-frequency indicator. Specifically, the indicator measures the extent of the 1-hop label changing of the center node. Due to the inaccessibility of the test node label, without loss of generality, our analysis of the indicator with the existence of prediction error shows that it is less likely to be affected. On four benchmarks, we test the indicator over popular homophilic, heterophilic, and tailored fraud detection methods, all of which show a great boost in the overall GAD performance. Our method can either work in an end-to-end manner, or prune inter-class edges given a set of reliable predictions and fix the input graph.

Our main contributions can be summarized as:

- We formulate the task of identifying edges between normal nodes and anomalies for GAD. We find heterophily is useless even harmful in class-imbalance problems like GAD.
- We explicitly fill the gap between the heterophily in the spatial domain and the frequency in the spectral domain.

- We devise a label-aware high-frequency component indicator, and prove its robustness to prediction error. Extensive empirical evaluation validates the effectiveness of the method in deleting heterophilous edges.

2 PRELIMINARIES

In this section, we illustrate the task of GAD. Then we introduce the heterophily property and graph Fourier transform.

Graph Anomaly Detection. Conventional anomaly detection techniques always consider isolated data instances while ignoring the relationship between instances which carries complementary information [2]. Differently, GAD treats $\mathcal{V}_a$ and $\mathcal{V}_n$ as two sets of abnormal and normal nodes respectively, and define the whole network as $\mathcal{G} = \{\mathcal{V}, \{\mathcal{E}\}, \mathbf{X}\}$. $\mathcal{V}$ is the union of anomaly and normal nodes (*i.e.,* $\mathcal{V} = \mathcal{V}_n \cup \mathcal{V}_a$); $\mathcal{E}$ stands for edges that either belong to the same relation or one of the multi-relations; $\mathbf{X}$ is the attribute matrix, each row of which is a d-dimensional vector representing the features of the corresponding node.

Typically, anomalies are regarded as positive with label 1, while normal nodes are negative with label 0 [12, 48]. GNN methods always cast GAD as a semi-supervised task, that is, given the information of the labeled nodes $\mathcal{V}_{train}$ along with their labels $\mathbf{Y}_{train}$, the classifier assigns the class $\hat{\mathbf{Y}}_{test}$ to the unlabeled nodes $\mathcal{V}_{test}$:

$$f(\mathcal{G}, \mathbf{Y}_{train}) \rightarrow \hat{\mathbf{Y}}_{test}. \tag{1}$$

Heterophilic Connections. Given a set of labeled nodes along with the edges between them, the edge is called a heterophilic connection if its source node and destination have distinct labels (*i.e.,* abnormal and normal), then the edge heterophily of a node v and the graph $\mathcal{G}$ could be respectively defined as:

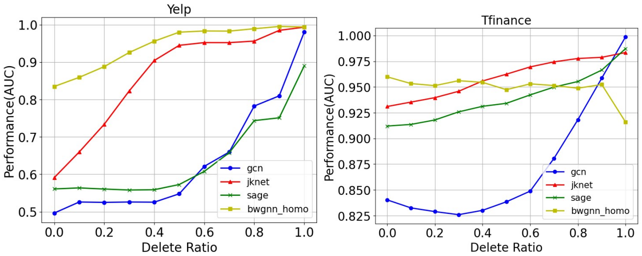
$$\begin{aligned} hetero(v) &= \frac{1}{|\mathcal{N}(v)|} |\{u : u \in \mathcal{N}(v), y_u \neq y_v\}| \\ hetero(\mathcal{G}) &= \sum_{(i,j)\in\mathcal{E}} \mathbb{I}\{\mathbf{y}_i \neq \mathbf{y}_j\}/|\mathcal{E}|, \end{aligned} \tag{2}$$

where $|\mathcal{E}|$ is the total number of edges and $\mathbb{I}$ is an indicator function. In GAD, anomalies have high heterophily and normal nodes have relatively low heterophily thanks to the imbalance nature.

Graph Fourier Transform. Let $\mathbf{A}$ be the adjacency matrix, then graph laplacian $\mathbf{L}$ can be expressed as $\mathbf{D} - \mathbf{A}$ or as $\mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ (symmetric normalized) or as $\mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$ (random walk normalized) [26], where $\mathbf{I}$ is the identity matrix and D is the diagonal degree matrix. Since $\mathbf{L}$ is positive semi-definite and symmetric, it has an eigendecomposition $\mathbf{L} = \mathbf{U}\Lambda\mathbf{U}^T$, where $\mathbf{\Lambda} = \{\lambda_1, \cdots, \lambda_N\}$ are eigenvalues and $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_N]$ are corresponding unit eigenvectors [36]. Assume $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]$ is a graph signal, then we call spectrum $\mathbf{U}^T\mathbf{X}$ as the graph Fourier transform of signal $\mathbf{X}$. In graph signal processing (GSP), the frequency is associated with $\mathbf{\Lambda}$ [6], thus the objective of spectral methods is to identify a response function $g(\cdot)$ on Λ to learn graph representation $\mathbf{Z}$ [5]:

$$\mathbf{Z} = g(\mathbf{L})\mathbf{X} = \mathbf{U}[g(\mathbf{\Lambda}) \odot (\mathbf{U}^T\mathbf{X})] = \mathbf{U}g(\mathbf{\Lambda})\mathbf{U}^T\mathbf{X} \tag{3}$$

As $\mathbf{L}$ have eigenvalues $\in [0, 2)$, and $\mathbf{A} = \mathbf{I} - \mathbf{L}$ with $g(\mathbf{\Lambda}) = \mathbf{I} - \mathbf{\Lambda}$ have eigenvalues $\in (-1, 1]$, $\mathbf{A}$ and $\mathbf{L}$ are treated as low-pass filters and high-pass filters, respectively [41].



(a) YelpChi **(b) T-Finance**

Figure 2: GNN Performance *w.r.t.* Edge Heterophily.

3 METHODOLOGY

In this section, we present the theoretical and empirical support for our proposed method. We first take a closer look into the influence of heterophily, finding it is harmful and thus should be reduced. Assisted by the smooth index defined on the graph Laplacian $\mathbf{L}$ and the ground-truth $\mathbf{Y}$, it is theoretically proved that heterophily is positively associated with frequency. We emphasize that identifying inter-class edges resorts to extracting high-frequency signals, in light of this key finding, we devise a novel model equipped with a label-aware high-frequency indicator, which measures the extent of 1-hop label changing of the center node. Since our method involves the ground truth of nodes which is partially masked in the real-world, as an alternative, we utilize the prediction of the model to estimate it. Without loss of generality, we verify the effectiveness of the indicator with the existence of prediction error.

3.1 Heterophily in Spectral Field

Most of GNNs aggregate neighborhood information based on the inductive bias (*aka.* homophily assumption) – "connected nodes tend to have similar labels", hence they are intrinsically low-pass filters [6, 41]. Apparently, this setting fails when the graph represents high heterophily, where neighboring nodes have distinct labels. This problem triggers the emergence of heterophily GNNs [3, 8, 50]. Among the branches, spectral GNNs [8, 36] adaptively learn filters with appropriate response functions or adopt a band-pass filter to absorb graph signals with different frequencies.

3.1.1 Influence of Heterophily. Heterophily is known to be harmful to the aggregation process of GNNs [26] because blindly mixing the features of nodes in different classes leads the nodes to be indistinguishable. Beyond the GAD scenario, some works [26, 28] find that the performance curves of GNNs are of "v" shape *w.r.t.* heterophily, which means the performance of GNNs will first increase then decrease when the heterophily ratio continues to increase. Intuitively, the worst heterophily occurs around the reciprocal of the number of classes, and we argue that the intrinsic reason is that the neighborhood distributions of nodes in distinct classes are the least distinguishable when their heterophily are similar.

But in GAD, normal nodes naturally have links between each other, while anomalies are sparse and connected to many normal nodes. This nature makes both normals and anomalies submerged in the normal nodes, hence their neighborhood label distributions tend to be the same. As a result, heterophily is quite harmful for the detection of anomalies, and the GNN performance should decrease with the increase of the edge heterophily. The empirical results shown in Figure 2 support our argument. We analyze the GNN performance with different heterophily edge deleting ratios (*i.e.,* from 0 to 1). Yelp and T-Finance are two real-world datasets with heterophily 0.23 and 0.03, respectively. This observation empirically shows that deleting heterophily edges could achieve an absolute boost in the overall GAD performance.

3.1.2 Heterophily and Frequency. Previous analysis shows the necessity to remove inter-class edges. Unfortunately, the ground-truth labels of test nodes are inaccessible, which limits our capability of directly deleting inter-class edges. Furthermore, the node-level prediction becomes unreliable when the prediction error is large. Considering the Manhattan distance of two predicted representations as the similarity measure, the prediction error will be accumulated to the final result. Recall that graph Laplacian denotes the local difference approximation of nodes, based on which the smoothness index is defined in the form of Rayleigh quotient (*i.e.,* $\frac{\mathbf{X}^T\mathbf{L}\mathbf{X}}{\mathbf{X}^T\mathbf{X}}$). Considering the one-hot ground-truth $\mathbf{Y}$ to be a signal on the graph, the smoothness index is equivalent to the rate of change of the labels, that is, the value of the heterophily ratio. Hence intuitively, there should be a relationship between the heterophily in the spatial domain and the frequency in the spectral domain. Towards this end, we aim to explore heterophily filtering in the spectral field. To bridge the gap, we need to efficiently capture the main frequency component of the ground-truth signal Y. Here we quantify the frequency distribution of the signal by introducing the metric below:

Definition 1 (*Spectral Energy Distribution* [36]) *Given the spectrum $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}^T$ of an input signal $\mathbf{Y}$, the spectral energy distribution at λ_k is:*

$$f_k(\mathbf{Y}, \mathbf{L}) = \alpha_k^2 / \textstyle\sum_{n=1}^{N} \alpha_i^2, \tag{4}$$

where f is a probability distribution with $\sum_{k=1}^{N} f_k = 1$, $\mathbf{L}$ is the Laplacian matrix of the graph. Since $\alpha_k = \mathbf{u}_k^T\mathbf{Y}$, $f_k(\mathbf{Y}, \mathbf{L})$ measures the weight of $\mathbf{u}_k$ in $\mathbf{Y}$, and a larger f_k indicates that the spectral energy concentrates more on λ_k.

Proposition 2 *For a binary classification problem, given graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$, and a one-hot input signal $\mathbf{Y}$, the expectation of spectral energy distribution $\mathbb{E}[f(\mathbf{Y}, \mathbf{L})]$ is monotonically increasing with the heterophily degree of the graph, and also affected by the total number of edges and nodes in the graph*:

$$\mathbb{E}[f(\mathbf{y}, \mathbf{L})] = \frac{|\mathcal{E}| \cdot hetero(\mathcal{G})}{N}, \tag{5}$$

where $|\mathcal{E}|$ is the total number of edges. The proof based on the smooth index can be found in Appendix A. In Proposition 2, we explicitly represent heterophily in the spectral domain. We know the frequency of the label is positively associated with the graph heterophily. This interesting finding verifies our conjecture, and is also consistent with the previous work [6]: a low-pass filter is empirically obtained for a homophilic graph, while a high-pass filter should be assigned if the graph is heterophilic. Also, a recent work [36] finds that f shifts following the degree of anomaly, which is called “right-shift". We further validate the phenomenon and claim that heterophily is the implicit factor that bridges f and the anomaly degree.

3.2 Heterophily Edge Denoising

3.2.1 Post-aggregation (PA) score. The analysis in Section 3.1.2 shows that a high-pass filter is of importance to detect heterophily edges. Recall that the eigenvalues λ_i of $\mathbf{L}$ exhibits high frequency of a graph, the k-th power of the normalized graph Laplacian is a commonly used high-pass filter. Note that the heterophily of a node is defined within the 1-hop neighborhood, we then employ the aggregation of 1-hop neighborhood label distribution as:

$$\mathbf{S} = \hat{\mathbf{L}}\mathbf{Y}, \tag{6}$$

where i-th row of $\mathbf{S}$ is called Post-aggregation (PA) similarity score for node i [26]. $\hat{\mathbf{L}}$ is the random walk normalized graph Laplacian (self-loop added), *i.e.,* $\hat{\mathbf{L}} = \mathbf{I} - \tilde{\mathbf{D}}^{-1}\tilde{\mathbf{A}}$, which can be written as:

$$\begin{pmatrix} \frac{d_1}{d_1+1} & -\frac{1}{d_1+1} & \cdots & -\frac{1}{d_1+1} \\ -\frac{1}{d_2+1} & \frac{d_2}{d_2+1} & \cdots & -\frac{1}{d_2+1} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{d_N+1} & -\frac{1}{d_N+1} & \cdots & \frac{d_N}{d_N+1} \end{pmatrix}$$

where the diagonal elements are $\hat{\mathbf{L}}_{ii} = \frac{d_i}{d_i+1}$, while non-diagonal elements are $\hat{\mathbf{L}}_{ij} = -\frac{1}{d_i+1}$ if there is an edge between node i and node j else 0. Then PA score (*i.e.,* $\mathbf{H}_i$) can be expressed as:

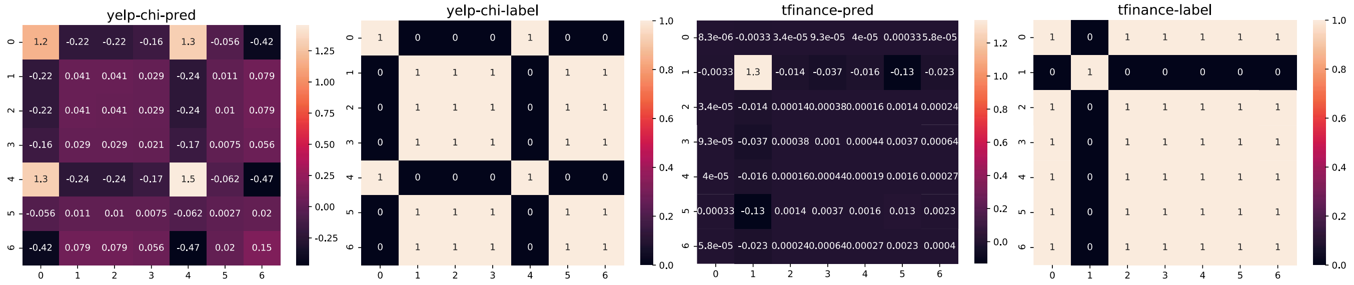
$$\mathbf{S}_i = SIGN * [\frac{d_i}{d_i+1} hetero(i), -\frac{d_i}{d_i+1} hetero(i)] \tag{7}$$

where $SIGN$ equals 1 and -1 for the normal nodes and anomalies respectively. Inspecting these equations, we have some observations: (1) The result is proportional to the node heterophily $hetero(i)$, which supports the opinion that the high-pass filter emphasizes node heterophily. (2) Anomalies and the normal nodes have exactly opposite directions after label aggregation so that the inner product could reveal the difference between them. (3) The inner product is proportional to the multiplication of the heterophily of two nodes. As analyzed before, anomalies have high heterophily while normal nodes have low heterophily, hence the value rank of the inner product of the PA score is:

$$\mathbf{S}_{v\in\mathcal{V}_a} \cdot \mathbf{S}_{v\in\mathcal{V}_a} > \mathbf{S}_{v\in\mathcal{V}_n} \cdot \mathbf{S}_{v\in\mathcal{V}_n} > 0 > \mathbf{S}_{v\in\mathcal{V}_a} \cdot \mathbf{S}_{v\in\mathcal{V}_n}, \tag{8}$$

from empirical results in Figure 3, we can observe this phenomenon. We randomly choose 7 nodes from YelpChi and T-Finance datasets and present their PA score: the row and column mean the node index, and the heatmap value is the PA score. For a more detailed version of the heatmap, please refer the Appendix B. This finding is valuable since we are unlikely to delete anomaly-anomaly edges when deleting edges with the least similarity scores. Besides, the heterophily for YelpChi and T-Finance is 0.23 and 0.03, respectively. As a result, the values of YelpChi are larger than that of T-Finance, which is consistent with the result in Equation (7).

3.2.2 Under Prediction Error. Typically, in real-world applications, only part of the labels are available to train the model. In this case, the PA score cannot be derived directly, since it requires the ground-truth label of the nodes. As an alternative, we utilize the prediction of the nodes to estimate it. In this section, we analyze the influence of prediction error on the similarity score. First, to



(a) YelpChi PA Score **(b) YelpChi Ground Truth** **(c) T-Finance PA Score** **(d) T-Finance Ground Truth**

Figure 3: Illustration of the ground-truth indicator and the ground-truth label. Considering S_i as the PA score, the rank of the edge values is $\mathrm{S}_{v\in\mathcal{V}_a}\cdot\mathrm{S}_{v\in\mathcal{V}_a} > \mathrm{S}_{v\in\mathcal{V}_n}\cdot\mathrm{S}_{v\in\mathcal{V}_n} > 0 > \mathrm{S}_{v\in\mathcal{V}_a}\cdot\mathrm{S}_{v\in\mathcal{V}_n}$.

unify the prediction error to be positive, we define the predicted label matrix $\hat{\mathrm{Y}}$ under prediction error as:

$$\hat{\mathrm{Y}}_v = \begin{cases} [1-\triangle_v, \triangle_v], y_v = 0, \\ [\triangle_v, 1-\triangle_v], y_v = 1, \end{cases} \tag{9}$$

where we denote $\triangle_v$ as the prediction error of node v. For easier expression, we denote PA score S_i and S_j as $[S_{i0}, S_{i1}]$ and $[S_{j0}, S_{j1}]$ for the i-th normal node and the j-th anomaly, respectively. Furthermore, we denote the expectation of prediction error for each class as $\mathbb{E}_k, k \in \{0,1\}$. On top of that:

$$\begin{aligned} S_{i0} &= \frac{d_i}{d_i+1}(1-\triangle_i) - \frac{1}{d_i+1}\sum_{j:y_i=y_j, i\neq j}(1-\triangle_j) - \frac{1}{d_i+1}\sum_{k:y_i\neq y_k}\triangle_k \\ &= \frac{d_i}{d_i+1}(1-\triangle_i) - \frac{d_i(1-h_i)}{d_i+1}(1-\mathbb{E}_0) - \frac{d_i h_i}{d_i+1}\mathbb{E}_1 \\ &= \frac{d_i h_i}{d_i+1}(1-\mathbb{E}_0-\mathbb{E}_1+\frac{\mathbb{E}_0-\triangle_i}{h_i}), \end{aligned} \tag{10}$$

$$\begin{aligned} S_{i1} &= -\frac{1}{d_i+1}\sum_{j:y_i\neq y_j}(1-\triangle_j) - \frac{1}{d_i+1}\sum_{j:y_i=y_k, i\neq k}\triangle_k + \frac{d_i}{d_i+1}\triangle_i \\ &= -\frac{d_i h_i}{d_i+1}(1-\mathbb{E}_1) - \frac{d_i(1-h_i)}{d_i+1}\mathbb{E}_0 + \frac{d_i}{d_i+1}\triangle_i \\ &= \frac{d_i h_i}{d_i+1}(-1+\mathbb{E}_1+\mathbb{E}_0-\frac{\mathbb{E}_0-\triangle_i}{h_i}), \end{aligned} \tag{11}$$

where h_i is the node heterophily of node i (*i.e., hetero*(i)). Similarly, for anomalies:

$$\begin{aligned} S_{j0} &= \frac{d_j h_j}{d_j+1}(-1+\mathbb{E}_1+\mathbb{E}_0-\frac{\mathbb{E}_1-\triangle_j}{h_j}), \\ S_{j1} &= \frac{d_j h_j}{d_j+1}(1-\mathbb{E}_0-\mathbb{E}_1+\frac{\mathbb{E}_1-\triangle_j}{h_j}), \end{aligned} \tag{12}$$

for a qualified classifier, the sum of average prediction errors should be less than 1. From the analysis above, we observe: (1) If the prediction error $\triangle_i$ is less than the same class average prediction error, the signs of PA score should be unaffected. (2) For abnormal nodes, due to their high heterophily, they are less likely to be affected by the prediction error. By aggregating the 1-hop neighborhood label with a high-pass filter, we alleviate the effect of prediction error. Towards this end, the inner product between inter-class nodes should be negative, while that between intra-class nodes is positive.

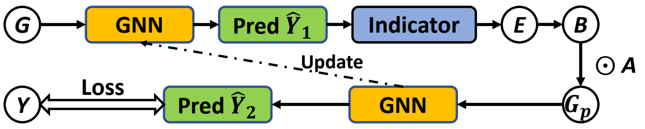


Figure 4: The architecture of the proposed model Graph Heterophily Resistance Network (GHRN).

3.2.3 *Heterophily Resistant Network.*

The overall framework is illustrated in Figure 4. GNN encoder encodes the node representation of the input graph G and gets prediction $\hat{\mathrm{Y}}_1$, according to which the high-frequency indicator gives edge score $\mathbf{E}$:

$$\mathbf{E} = \mathbf{L}\hat{\mathbf{Y}}\hat{\mathbf{Y}}^T\mathbf{L}^T, \tag{13}$$

From the previous analysis, we know the $\mathbf{E}$ value of the heterophilous edges is larger than that of homophilous edges, which paves the way for our pruning. We tease out the edges with top-K sampling and binarize the edge score $\mathbf{E}$ to $\mathbf{B}$, based on which we purify the graph into G_p:

$$G_p = \{\mathcal{V}, \mathbf{B}\odot\mathbf{A}, \mathbf{X}\}, \tag{14}$$

G_P shares the same nodes and features with the original G, but a different adjacency matrix. The GNN encoder is then updated with a new input graph G_P through cross-entropy loss:

$$\mathcal{L} = \sum_{v\in\mathcal{V}} -log(y_v\cdot\sigma(\hat{y}_v)). \tag{15}$$

4 EXPERIMENTS

In this section, we conduct experiments on real-world datasets and report the results of our models as well as some state-of-the-art baselines to show the effectiveness of our proposed model. Particularly, we mainly aim to answer the following research questions:

- **RQ1:** How does the proposed model perform compared to the homophilous, heterophilous, and SOTA GAD methods?
- **RQ2:** How does GHRN performs compared to alternative edge pruning methods? Can it successfully reduce heterophily?
- **RQ3:** Can the proposed model work well with prediction instead of the gold label? What's the effect of prediction error?
- **RQ4:** Is GHRN sensitive to the hyperparameter deleting ratio **r**?

Table 1: Performance Results. The best results of all methods are indicated in boldface, and the best results of each category (*i.e.,* homophilous, heterophilous, and GAD) are underlined. The T-Social dataset consists of 5 million nodes and up to 100 million edges and is a relatively large dataset. H2GCN needs to operate on the adjacency matrix, leading to the memory issue. T-finance and T-social only have one single relation, so they are treated as homo. The Hetero and Homo in BWGNN stand for heterogenous and homogenous respectively.

Method	Dataset	YelpChi		Amazon		T-finance		T-Social	
	Metric	F1-Macro	AUC	F1-Macro	AUC	F1-Macro	AUC	F1-Macro	AUC
Homophilous GNNs	MLP	0.4614	0.7366	0.9010	0.9082	0.4883	0.8609	0.4406	0.4923
	GCN [18]	0.5157	0.5413	0.5098	0.5083	0.5254	0.8203	0.6550	0.7012
	GAT [37]	0.4614	0.5459	0.5675	0.7731	0.8816	0.9388	0.4921	0.4923
	JKNet [44]	0.5805	0.7736	0.8270	0.8970	0.8971	0.9554	0.4923	0.7226
	JK-GHRN (Ours)	0.6145	0.7765	0.8756	0.9206	0.9015	0.9559	0.4923	0.7016
GAD Models	CARE-GNN [12]	0.5015	0.7300	0.6313	0.8832	0.6115	0.8731	0.4868	0.7939
	PC-GNN [24]	0.6925	0.8118	0.8367	0.9555	0.5322	0.9182	0.4536	0.8917
	PC-GHRN (Ours)	0.7082	0.8230	0.8855	0.9519	0.6177	0.9238	0.6218	0.9035
Heterophilious GNNs	H2GCN [50]	0.6575	0.8406	0.9213	0.9693	0.8824	0.9553	-	-
	MixHop [1]	0.6534	0.8796	0.8093	0.9723	0.4880	0.9569	0.6471	0.9597
	GPRGNN [8]	0.6423	0.8355	0.8059	0.9358	0.8507	**0.9642**	0.5976	**0.9722**
	BWGNN(Homo) [36]	0.6935	0.8255	0.9194	0.9395	0.8899	0.9599	**0.9145**	0.9630
	BHomo-GHRN (Ours)	0.7532	0.8631	0.9203	0.9609	**0.8975**	0.9609	0.9118	0.9637
	BWGNN(Hetero) [36]	0.7568	0.8967	0.9204	0.9706	-	-	-	-
	BHetero-GHRN (Ours)	**0.7789**	**0.9073**	**0.9282**	**0.9728**	-	-	-	-

4.1 Experimental Setup

4.1.1 Baselines. Our baselines can be categorized into three groups. The first group considers some homophilous methods, including GCN [18], GAT [37] and JKNet [44]. Note that MLP can be regarded as an all-pass filter with all of the eigenvalues being 1. As our focus is GAD, the second group considers tailored GAD methods such as CARE-GNN [12] and PC-GNN [24]. The third group is heterophilous methods, most of which are designed in the spectral domain. They are H2GCN [50] and BWGNN [36]. More details are shown in Appendix C.4.

4.1.2 Metrics. Since GAD is always a class-imbalanced classification problem, we adopt two widely used measures for a fair comparison, namely F1-macro and AUC. F1-macro calculates F1-score for every class and finds their unweighted mean. AUC is the area under the ROC Curve, which depicts the relationship between the False Positive Rate (FPR) and the True Positive Rate (TPR). For both of the two metrics, the higher scores indicate a higher performance.

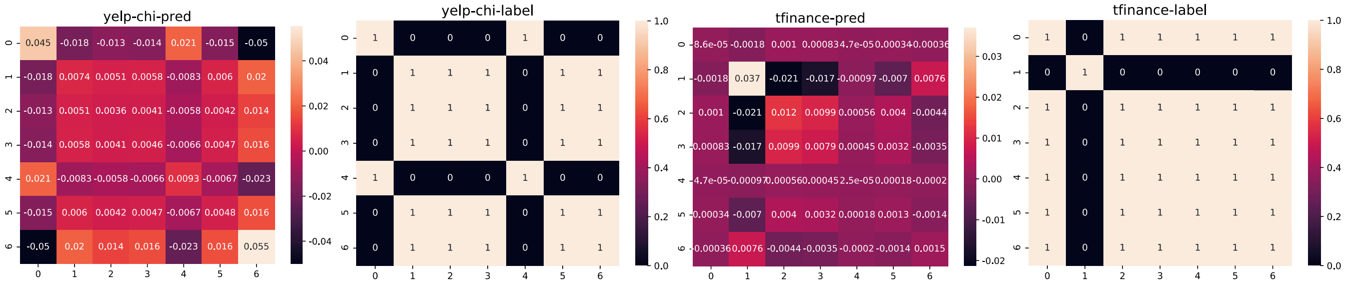
4.2 Main Comparison Results

To answer **RQ1**, we evaluate the performance of baselines and the proposed method, and the comparison results are reported in Table 1. Note that we don't adjust the threshold to achieve F1-Macro as previous works do, since we think AUC can reflect this performance. We suppose with a threshold 0.5 we can analyze more information such as model confidence. We have the following observations:

First of all, heterophilous GNNs consistently outperform homophilous GNNs. We ascribe this large performance gap to the harmfulness of heterophily. We report the heterophily degree for each dataset in Appendix C.2. Joining Table 1 and Table 4, we observe the gap is more huge when the heterophily for the dataset is extremely high. Especially on the Amazon dataset, where the anomaly heterophily is 0.9254, MLP outperforms most of the homophilous GNNs. Additionally, JKNet has the most competitive performance among the homophilous GNNs. We suppose the most possible reason is that it inherits the original information through max-pooling operation, hence is more robust to false mixing. All of these observations suggest that we need to address the heterophily problem carefully and properly in GAD. Another interesting observation is that on T-Finance and T-Social, methods could achieve a high AUC while maintaining a quite low F1-Macro (even lower than 0.5). The model can distinguish two classes but the prediction is skewed, which we suppose is unhealthy and unstable.

Secondly, CARE-GNN and PC-GNN are two tailored GAD classifiers built upon a multi-relation graph. They utilize RGCN as the backbone model and focus on modifying the adjacency matrix to prune noise edges. Our motivations are similar, yet different from them, we are aiming at distinguishing 1-hop label distribution difference instead of calculating node-level similarity, which can alleviate the effects of prediction error when deleting edges. Experimental results show that the proposed method consistently outperforms these two popular GAD methods on all the metrics across two datasets, which demonstrates the effectiveness of the proposed edge indicator. PC-GNN performs better than CARE-GNN, we attribute this improvement to its attempt of maintaining a balanced neighborhood label distribution before pruning.

Heterophilous methods including H2GCN, Mixhop, GPRGNN and BWGNN discover the harmfulness of inter-class edges. Hence,



(a) YelpChi PA Score **(b) YelpChi Ground Truth** **(c) T-Finance PA Score** **(d) T-Finance Ground Truth**

Figure 5: Illustration of the prediction indicator and the gold label.

Table 2: Performance comparison with alternative edge pruning methods. The threshold for F1-macro is 0.5.

	T-Finance		T-Social	
	F1-Macro	AUC	F1-Macro	AUC
DropEdge	0.8417	0.9240	0.6553	0.8495
AdaEdge(Drop)	0.8843	0.9298	0.6419	0.8407
GHRN (Ours)	**0.8975**	**0.9609**	**0.9118**	**0.9637**
Improvement	1.49%	3.34%	42.14%	14.63%

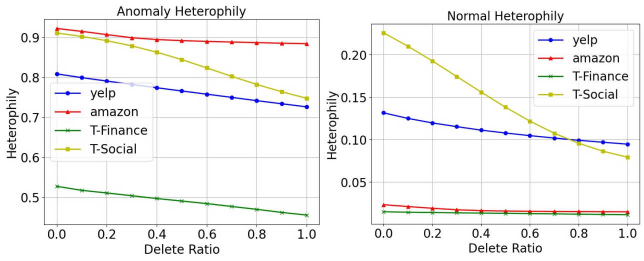
they aim to extract and deal with heterophily and homophily information separately. However, as stated in Section 3.1, heterophily could be treated as high-frequency noises in GAD. Instead of designing a proper response function to identify the signal, we suppose delineating is more straightforward and better. The result in the table supports our argument.

4.3 Effectiveness of the Indicator

To answer **RQ2**, we compare GHRN with several popular alternative edge pruning methods. DropEdge [33] randomly drops out a certain rate of edges of the input graph; AdaEdge [4] add and drop edges according to the similarity of two nodes and the confidence of the prediction, here we adapt it to the only-drop version; G-Aug [47] is another popular graph structure learning method, however, we don't adopt it as our baseline due to the high time complexity and space complexity of GVAE used in G-Aug. Pruning edge methods on graphs with higher density are more reasonable, in light of which we test three methods on two large-scale benchmarks: T-Finance and T-Social. As shown in Table 2, the performances of DropEdge and AdaEdge are similar, suggesting that node-level similarity can be unreliable and even misleading in GAD. The proposed method shows great superiority from this perspective. In addition, we seek to answer the question that "Can edge indicator successfully reduce the heterophily degree?" We plot the heterophily degree with a different deleting ratio for both the anomalies and the normal nodes in four benchmarks. As presented in Figure 6, the heterophily degree decreases with the increase of deleting ratio.

4.4 The Role of Prediction Error

In real-world applications, not all of the ground-truth of nodes are seen. As an alternate, our pruning operations are based on the prediction. Towards this end, to answer **RQ3**, we measure the label-aware edge indicator with the existence of prediction error. Comparing Figure 3 and Figure 5, we observe the same phenomenon as discussed in Section 3.1. That is, the value rank of the inner product is $anomaly \cdot anomaly > normal \cdot normal > 0 > normal \cdot anomaly$. However, in rare cases, the algorithm may fail. For instance, the 6th node in YelpChi behaves as an anomaly – the self-loop value is large (0.055), and some related inter-class edges are positive, which is different from the gold label heatmap where the 6th node is normal. We suppose the effect of the prediction error is inevitable but acceptable in our proposed method.

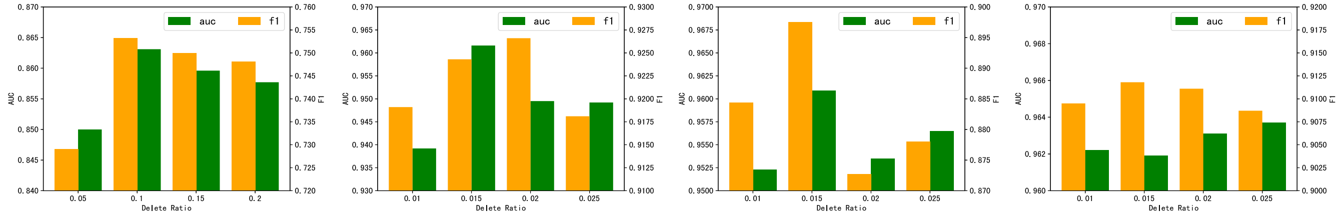


(a) Abnormal Heterophily **(b) Normal Heterophily**

Figure 6: Illustration of the heterophily change when part of the edges are deleted.

4.5 Sensitivity Analysis

As shown in Figure 2, the performance of GNNs varies with the change in deleting ratio. Hence, to answer **RQ4**, we want to explore the model's sensitivity to the most important hyper-parameter "**Deleting Ratio** r", which controls the ratio of deleting edges to the total. The performance of two metrics — F1-Macro and AUC with different deleting ratios are shown in Figure 7, respectively. We observe that (1) Generally when r continues to increase, the performance will first increase and then decrease. In the first stage, the performance gain is from the decrease of the heterophily ratio of the graph. However, as more edges are deleted, the probability of wrong deletions becomes higher which may cause the performance drop. (2) r should be around 0.015 for T-Social and Amazon datasets, while that for YelpChi is around 0.1. Joining this observation with 4, we claim r is better to be around half of the heterophily ratio. But as seen from Figure 7d, the model achieves good performance when r is small although T-Social has a high heterophily. We are



(a) YelpChi *w.r.t.* Deleting Ratio **(b) Amazon *w.r.t.* Deleting Ratio** **(c) T-Finance *w.r.t.* Deleting Ratio** **(d) T-Social *w.r.t.* Deleting Ratio**

Figure 7: Model performance (*i.e.,* AUC and F1-Macro) with a different deleting ratio. Since the two metrics are different in scale, we display the charts in a dual-Y style. The left-y axis represents AUC, while the right one represents F1-Macro.

confused about this phenomenon and couldn't give an explanation. (3) The performance is stable over the range near optimal.

5 RELATED WORK

In this section, we introduce some representative GAD networks and tailored Heterophilous GNNs.

5.1 GNN-based Graph Node Anomaly Detection

GNNs have been widely used as an effective method to leverage information about the graph attributes to learn to score anomalies appropriately [17]. In this work, we focus on node anomalies that are associated with static graphs.

Graph auto-encoder (GAE) is a popular branch of unsupervised-learning methods on static attributed graphs. DOMINANT [10] spots anomalies by measuring the reconstruction errors of nodes from both the structure and the attribute perspectives. SpecAE [21] is a spectral convolution and deconvolution based framework to project the attributed network into a tailored space to detect global and community anomalies. AnomalyDAE [13] captures the complex interactions between network structure and node attribute for high-quality node embeddings. With the advances of GNNs, GNN-based semi-supervised learning methods [11, 22] have been of focus. GraphUCB [11] adopts contextual multi-armed bandit technology, and transform graph anomaly detection to a decision-making problem. DCI [42] decouples representation learning and classification with the self-supervised learning task.

Recent methods realize the importance of incorporating multiple relationships into graph learning [12, 23–25, 38, 39, 46]. FdGars [39] and GraphConsis [25] construct a single homo-graph with multiple relations and leverage GNNs to aggregate neighborhood information. Similarly but differently, Semi-GNN [38], CARE-GNN [12], and PC-GNN [24] construct multiple homo-graphs based on node relation. Semi-GNN and IHGAT [23] employ hierarchical attention mechanism for interpretable prediction, while based on camouflage behaviors and imbalanced problems, CARE-GNN and PC-GNN prune edges adaptively according to neighbor distribution.

5.2 Heterophilous Graph Neural Networks

In GAD, nodes with different labels tend to be linked, which significantly limits the performance of vanilla GNNs. Hence, it is worth studying GNNs for heterophilic graphs in the community [49]. Mixhop [1] repeatedly mixes feature representations of neighbors at various distances to alleviate the negative effect of 1-hop heterophily. GPRGNN [8] adaptively learns the Generalized PageRank weights. FAGCN [3] adaptively fuses different signals in the process of message passing by employing a self-gating mechanism. H2GCN [50] identifies a set of key designs which are combined into a single graph neural network. FSGNN [29] treats the feature propagation and learning separately and proposes a simple GNN model with some considerations. ACM [26] study heterophily from the perspective of post-aggregation node similarity and adaptively exploits aggregation, diversification, and identity channels in each GNN layer. There are also some tailored-GAD heterophily GNNs [16, 34, 36]. AO-GNN [16] decouples the AUC maximization process on GNN into a classifier parameter searching and an edge pruning policy searching process to solve the label-imbalance problem as well as the heterophily issue. H2-FDetector [34] identifies the homophilic and heterophilic connections with the supervision of labeled nodes for both of which they design distinct aggregation strategies. BWGNN [36] observes the "right-shift" phenomenon and designs a band-pass filter to aggregate different frequency signals simultaneously.

6 CONCLUSION AND FUTURE WORK

In this work, we look closely at heterophily's negative effect on GAD. Explicitly bridging the heterophily in the spatial domain and the frequency in the spectral domain makes it possible to delineate the inter-class edges with guidance from the spectral domain. Towards this end, we devise a label (prediction)-aware edge indicator to calculate the post-aggregation similarity score based on which we prune possibly heterophily edges.

The method addresses the heterophily issue in GAD from the spectral domain. For future work, we suppose a few research directions deserve our attention: (1) Better unsupervised learning pruning techniques [20]. The performance of prediction-aware indicators relies highly on the quality of the prediction error although the error can be somehow reduced. From this perspective, an informative label-irrelevant statistic is preferred. (2) The generalization of Heterophily. A recent work [19] studies the problem that severe performance degradation occurs if a large heterophily gap exists between training and testing graphs. This direction is worth studying [43, 45], and our proposed method will likely contribute to the improvement in this branch. (3) Human-level Explanation. GHRN takes a step toward understanding the frequency in the graph. We hope to investigate on the mechanism behind spectral graph filtering.

7 ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China (2020AAA0106000), the National Natural Science Foundation of China (9227010114, U21B2026, 62121002), and the CCCD Key Lab of Ministry of Culture and Tourism.

REFERENCES

[1] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. 2019. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *ICML*. 21–29.

[2] Leman Akoglu, Hanghang Tong, and Danai Koutra. 2015. Graph based anomaly detection and description: a survey. *Data Min. Knowl. Discov.* 29, 3 (2015), 626–688.

[3] Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. 2021. Beyond low-frequency information in graph convolutional networks. In *AAAI*. 3950–3957.

[4] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. 2020. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *AAAI*. 3438–3445.

[5] Zhiqian Chen, Fanglan Chen, Lei Zhang, Taoran Ji, Kaiqun Fu, Liang Zhao, Feng Chen, and Chang-Tien Lu. 2020. Bridging the Gap between Spatial and Spectral Domains: A Survey on Graph Neural Networks. *CoRR* abs/2002.11867 (2020).

[6] Zhixian Chen, Tengfei Ma, and Yang Wang. 2022. When Does A Spectral Graph Neural Network Fail in Node Classification? *CoRR* abs/2202.07902 (2022).

[7] Lu Cheng, Ruocheng Guo, Kai Shu, and Huan Liu. 2021. Causal understanding of fake news dissemination on social media. In *KDD*. 148–157.

[8] Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. 2021. Adaptive Universal Generalized PageRank Graph Neural Network. In *ICLR*.

[9] Limeng Cui, Haeseung Seo, Maryam Tabar, Fenglong Ma, Suhang Wang, and Dongwon Lee. 2020. Deterrent: Knowledge guided graph attention network for detecting healthcare misinformation. In *KDD*. 492–502.

[10] Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. 2019. Deep anomaly detection on attributed networks. In *SDM*. 594–602.

[11] Kaize Ding, Jundong Li, and Huan Liu. 2019. Interactive anomaly detection on attributed networks. In *WSDM*. 357–365.

[12] Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S Yu. 2020. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *CIKM*. 315–324.

[13] Haoyi Fan, Fengbin Zhang, and Zuoyong Li. 2020. AnomalyDAE: Dual autoencoder for anomaly detection on attributed networks. In *ICASSP*. 5685–5689.

[14] Yuan Gao, Xiang Wang, Xiangnan He, Huamin Feng, and Yongdong Zhang. 2022. Rumor Detection with Self-supervised Learning on Texts and Social Graph. *CoRR* (2022).

[15] Yuan Gao, Xiang Wang, Xiangnan He, Huamin Feng, and Yongdong Zhang. 2023. Alleviating Structrual Distribution Shift in Graph Anomaly Detection. In *WSDM*.

[16] Mengda Huang, Yang Liu, Xiang Ao, Kuan Li, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. 2022. AUC-oriented Graph Neural Network for Fraud Detection. In *WWW*. 1311–1321.

[17] Hwan Kim, Byung Suk Lee, Won-Yong Shin, and Sungsu Lim. 2022. Graph Anomaly Detection with Graph Neural Networks: Current Status and Challenges. *IEEE Access* (2022).

[18] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[19] Runlin Lei, Zhen Wang, Yaliang Li, Bolin Ding, and Zhewei Wei. 2022. EvenNet: Ignoring Odd-Hop Neighbors Improves Robustness of Graph Neural Networks. In *NIPS*.

[20] Sihang Li, Xiang Wang, An Zhang, Yingxin Wu, Xiangnan He, and Tat-Seng Chua. 2022. Let Invariant Rationale Discovery Inspire Graph Contrastive Learning. In *ICML*, Vol. 162. 13052–13065.

[21] Yuening Li, Xiao Huang, Jundong Li, Mengnan Du, and Na Zou. 2019. Specae: Spectral autoencoder for anomaly detection in attributed networks. In *CIKM*. 2233–2236.

[22] Jiongqian Liang, Peter Jacobs, Jiankai Sun, and Srinivasan Parthasarathy. 2018. Semi-supervised embedding in attributed networks with outliers. In *SDM*. 153–161.

[23] Can Liu, Li Sun, Xiang Ao, Jinghua Feng, Qing He, and Hao Yang. 2021. Intention-aware heterogeneous graph attention networks for fraud transactions detection. In *KDD*. 3280–3288.

[24] Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. 2021. Pick and choose: a GNN-based imbalanced learning approach for fraud detection. In *WWW*. 3168–3177.

[25] Zhiwei Liu, Yingtong Dou, Philip S Yu, Yutong Deng, and Hao Peng. 2020. Alleviating the inconsistency problem of applying graph neural network to fraud detection. In *SIGIR*. 1569–1572.

[26] Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. 2022. Revisiting Heterophily For Graph Neural Networks. In *NIPS*.

[27] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z Sheng, Hui Xiong, and Leman Akoglu. 2021. A comprehensive survey on graph anomaly detection with deep learning. *TKDE* (2021).

[28] Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. 2022. Is homophily a necessity for graph neural networks?. In *ICLR*.

[29] Sunil Kumar Maurya, Xin Liu, and Tsuyoshi Murata. 2021. Improving graph neural networks with simple architecture design. *CoRR* (2021).

[30] Julian John McAuley and Jure Leskovec. 2013. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *WWW*. 897–908.

[31] Shebuti Rayana and Leman Akoglu. 2015. Collective opinion spam detection: Bridging review networks and metadata. In *KDD*. 985–994.

[32] Everett M Rogers and Dilip K Bhowmik. 1970. Homophily-heterophily: Relational concepts for communication research. *Public opinion quarterly* 34, 4 (1970).

[33] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2020. DropEdge: Towards deep graph convolutional networks on node classification. In *ICLR*.

[34] Fengzhao Shi, Yanan Cao, Yanmin Shang, Yuchen Zhou, Chuan Zhou, and Jia Wu. 2022. H2-FDetector: A GNN-based Fraud Detector with Homophilic and Heterophilic Connections. In *WWW*. 1486–1494.

[35] Ljubiša Stanković, Danilo Mandic, Miloš Daković, Bruno Scalzo, Miloš Brajović, Ervin Sejdić, and Anthony G Constantinides. 2020. Vertex-frequency graph signal processing: A comprehensive review. *Digital signal processing* (2020), 102802.

[36] Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. 2022. Rethinking Graph Neural Networks for Anomaly Detection. In *ICML*.

[37] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.

[38] Daixin Wang, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, Shuang Yang, and Yuan Qi. 2019. A semi-supervised graph attentive network for financial fraud detection. In *ICDM*. 598–607.

[39] Jianyu Wang, Rui Wen, Chunming Wu, Yu Huang, and Jian Xion. 2019. Fdgars: Fraudster detection via graph convolutional networks in online app review system. In *WWW (Companion Volume)*. 310–316.

[40] Minjie Yu Wang. 2019. Deep graph library: Towards efficient and scalable deep learning on graphs. In *ICLR Workshop*.

[41] Xiyuan Wang and Muhan Zhang. 2022. How Powerful are Spectral Graph Neural Networks. In *ICML*.

[42] Yanling Wang, Jing Zhang, Shasha Guo, Hongzhi Yin, Cuiping Li, and Hong Chen. 2021. Decoupling representation learning and classification for gnn-based anomaly detection. In *SIGIR*. 1239–1248.

[43] Yingxin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. 2022. Discovering Invariant Rationales for Graph Neural Networks. In *ICLR*.

[44] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation Learning on Graphs with Jumping Knowledge Networks. In *ICML*. 5449–5458.

[45] An Zhang, Wenchang Ma, Xiang Wang, and Tat-Seng Chua. 2022. Incorporating Bias-aware Margins into Contrastive Loss for Collaborative Filtering. In *NeurIPS*.

[46] Ge Zhang, Jia Wu, Jian Yang, Amin Beheshti, Shan Xue, Chuan Zhou, and Quan Z Sheng. 2021. FRAUDRE: Fraud Detection Dual-Resistant to Graph Inconsistency and Imbalance. In *ICDM*. 867–876.

[47] Tong Zhao, Yozen Liu, Leonardo Neves, Oliver Woodford, Meng Jiang, and Neil Shah. 2021. Data augmentation for graph neural networks. In *AAAI*. 11015–11023.

[48] Tong Zhao, Bo Ni, Wenhao Yu, and Meng.Wang Jiang. 2020. Early Fraud Detection with Augmented Graph Learning. In *KDD Workshop*.

[49] Xin Zheng, Yixin Liu, Shirui Pan, Miao Zhang, Di Jin, and Philip S. Yu. 2022. Graph Neural Networks for Graphs with Heterophily: A Survey. *CoRR* abs/2202.07082 (2022).

[50] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. 2020. Beyond homophily in graph neural networks: Current limitations and effective designs. In *NIPS*. 7793–7804.

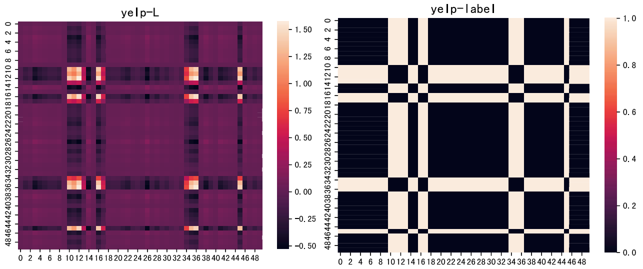


Figure 8: Illustration of the YelpChi indicator and the ground-truth (50 nodes)

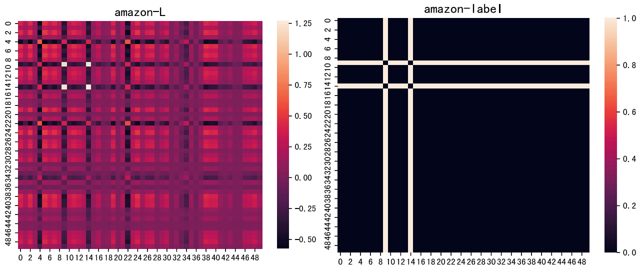


Figure 9: Illustration of the Amazon indicator and the ground-truth (50 nodes)

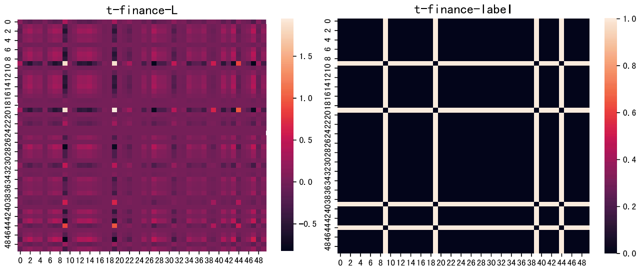


Figure 10: Illustration of the T-Finance indicator and the ground-truth (50 nodes)

A THE PROOF OF PROPOSITION 2

Proof of Proposition 2, partially from [19]:
The Rayleigh quotient of the label is:

$$
\begin{aligned}
E[\mathbf{y}] &= \mathbf{y}^T\mathbf{L}\mathbf{y} = \mathbf{y}^T\mathbf{D}\mathbf{y} - \mathbf{y}^T\mathbf{A}\mathbf{y} = \sum_{i=1}^{N} d_i \mathbf{y}_i^2 - \sum_{i,j=1}^{N} \mathbf{y}_i\mathbf{y}_j\mathbf{A}_{ij} \\
&= \frac{1}{2}\Big(\sum_{i=1}^{N} d_i\mathbf{y}_i^2 - 2\sum_{i,j=1}^{N}\mathbf{y}_i\mathbf{y}_j A_{ij} + \sum_{j=1}^{N} d_j\mathbf{y}_j^2\Big) \\
&= \frac{1}{2}\sum_{(i,j)\in\mathcal{E}} (\mathbf{y}_i - \mathbf{y}_j)^2 \\
&= \sum_{(i,j)\in\mathcal{E}} \mathbb{I}\{\mathbf{y}_i \neq \mathbf{y}_j\} \\
&= |\mathcal{E}| \cdot hetero(\mathcal{G})
\end{aligned}
\tag{16}
$$

Table 3: Statistics of Datasets

Dataset	#Nodes	#Edges	Relation	#Edges
YelpChi	45,954	3,846,979	R-U-R	49,315
			R-S-R	3,402,743
			R-T-R	573,616
Amazon	11,944	4,398,392	U-P-U	175,608
			U-S-U	3,566,479
			U-V-U	1,036,737
T-Finance	39,357	21,222,543	-	-
T-Social	5,781,065	73,105,508	-	-

Also the Rayleigh quotient of the label can be acquired as:

$$
\begin{aligned}
E[\mathbf{y}] &= \mathbf{y}^T\mathbf{U}\Lambda\mathbf{U}^T\mathbf{y} = \alpha^T\Lambda\alpha \\
&= \sum_{i=1}^{N}\lambda_i\alpha_i^2 \\
&= \sum_{i=1}^{N}\alpha_i^2\mathbb{E}[f(\mathbf{y},\mathbf{L})] \\
&= N\mathbb{E}[f(\mathbf{y},\mathbf{L})]
\end{aligned}
\tag{17}
$$

Joining Equations (16) and (17), we have:

$$
\mathbb{E}[f(\mathbf{y},\mathbf{L})] = \frac{|\mathcal{E}|\cdot hetero(\mathcal{G})}{N}
\tag{18}
$$

B A DETAILED VERSION OF PA SCORES

As shown in Figure 8, 9, 10, we present a more detailed version of comparison between PA score and the gold label: sampling 50 nodes from YelpChi, Amazon and T-Finance, respectively. The visualization is consistent with the result in Equation (7).

C REPRODUCIBLE DETAILS

This section presents more details about the dataset statistics, including size and heterophily degree.

C.1 Datasets

Following previous works [12], we conduct experiments on two tiny datasets and two large datasets to study the GNN-based fraud detection problem. The YelpChi dataset [31] includes hotel and restaurant reviews filtered and recommended by Yelp. The graph has three relations: R-U-R denotes the reviews posted by the same user; R-S-R denotes the reviews under the same product with the same star rating; (3) R-T-R denotes the reviews under the same product posted in the same month. The Amazon dataset [30] includes product reviews under the Musical Instruments category, which also have three relations: U-P-U connects users reviewing the same product; U-S-U connects users having the same star rating; U-V-U connects users with the top-5% mutual review. For raw node features, the YelpChi dataset has 32-dimensions while the Amazon dataset has 25-dimension features. Besides these two spam-review datasets, we also utilize two transaction datasets released recently in [36]. The T-Finance dataset aims to detect human-annotated anomaly accounts in a transaction network. The nodes are accounts with 10-dimension features whereas the edges connecting them denote

Table 4: Heterophily for Benchmarks

	normal	anomaly	heterophily of the graph
YelpChi	0.1317	0.8144	0.2268
Amazon	0.0234	0.9254	0.0456
T-Finance	0.0150	0.5280	0.0292
T-Social	0.2366	0.9161	0.3761

they have transaction records. The T-social dataset aims to detect human-annotated anomaly accounts in a social network. The node annotations and features are the same as T-Finance, whereas the edges connecting the nodes denote they maintain the friendship for more than 3 months. T-Finance and T-Social maintain a much larger size than YelpChi and Amazon. The detailed statistics of the datasets are reported in Table 3.

C.2 Heterophily Statistics

In this section, we calculate the heterophily degree for normal, abnormal, and all nodes respectively, reported in Table 4.

C.3 Implementation Details

For the YelpChi and Amazon fraudDataset, we use the official split from DGL [40]. For T-Finance and T-Social datasets, following previous work [36], our data splitting ratio is 40%, 20%, and 40% for training, validation, and test set. All of the hyperparameters are set to those reported in their paper if available, while the edge deleting ratio is chosen according to the edge heterophily of the dataset in Table 4. For all of the methods, we run 100 epochs, where Homophilous methods are implemented with the DGL library in Pytorch, and for the rest of the methods, we use their provided open-source code to implement them.

C.4 Baselines

This section presents more details about the baselines.

- **GCN** [18]: GCN is a traditional graph convolutional network in spectral space.
- **GAT** [37]: GAT leverages masked self-attentional layers to address the shortcomings of prior graph convolution methods.
- **JKNet** [44]: The jumping-knowledge network which concatenates or max-pooling the hidden representation from each layer.
- **Care-GNN** [12]: Care-GNN is a camouflage-resistant graph neural network that adaptively samples neighbors according to the feature similarity, and the optimal sampling ratio is found through an RL module.
- **PC-GNN** [24]: PC-GNN consists of two modules "pick" and "choose", and maintains a balanced label frequency around fraudsters by downsampling and upsampling.
- **H2GCN** [50]: H2GCN is a tailored heterophily GNN which identifies three useful designs.
- **MixHop** [1]: Mixhop repeatedly mixes feature representations of neighbors at various distances to learn relationships.
- **GPRGNN** [8]: GPRGNN learns a polynomial filter by directly performing gradient descent on the polynomial coefficients.
- **BWGNN** [36]: BWGNN is a tailored spectral filter for anomaly detection, aiming to address the "right-shift" phenomenon.