Clustering-Augmented Fraud Detection on Graphs Using Label-Aware Feature Aggregation

Shixiong Jing JING@PSU.EDU
Pennsylvania State University, University Park, PA 16802, USA
Lingwei Chen LINGWEI.CHEN@WRIGHT.EDU
Wright State University, Dayton, OH 45435, USA
Dinghao Wu DINGHAO@PSU.EDU
Pennsylvania State University, University Park, PA 16802, USA

Editors: Vu Nguyen and Hsuan-Tien Lin

Abstract

Fraud detection has emerged as a pivotal process in different fields (e.g., e-commerce, social networks). Since interactions among entities provide valuable insights into fraudulent activities, such behaviors can be naturally represented as graphs, where graph neural networks (GNNs) have been developed as prominent models to boost the efficacy of fraud detection. However, the application of GNNs in this domain encounters significant challenges, primarily due to class imbalance and a mixture of homophily and heterophily of fraud graphs. To address these challenges, in this paper, we propose LACA, which implements fraud detection on graphs using Label-Aware feature aggregation to advance GNN training, which is regularized by Clustering-Augmented optimization. Specifically, label-aware feature aggregation simplifies adaptive aggregation in homophily-heterophily mixed neighborhoods, preventing gradient domination by legitimate nodes and mitigating class imbalance in message passing. Clustering-augmented optimization provides fine-grained subclass semantics to improve detection performance, and yields additional benefit in addressing class imbalance. Extensive experiments on four fraud datasets demonstrate that LACA can significantly improve fraud detection performance on graphs with different imbalance ratios and homophily ratios, outperforming state-of-the-art GNN models.

Keywords: Fraud Detection, Graph Neural Networks, Heterophily, Imbalance, Clustering

1. Introduction

Increasing connectivity of devices and individuals to the Internet has dramatically reshaped various aspects of our daily lives. While these advancements offer considerable benefits, they also create expanding opportunities for fraudsters to exploit these networks for economic, social, or political gains (Hooi et al., 2017). The surge in fraudulent activities has thus underscored the importance of fraud detection in the fields of e-commerce (Chen et al., 2021a), healthcare (Bauder et al., 2017), online reviews (Wang et al., 2019; Li et al., 2019), and social networks (Feng et al., 2022). Fraudulent entities, such as accounts, reviews, and transactions, often disguise their malicious intent by blending in genuine information, making them difficult to detect based solely on individual attributes. However, their interactions with others provide crucial clues that can be analyzed to reveal their deceitful nature (Pourhabibi et al., 2020). Representing fraudulent activities using graphs thus enables a more intuitive and effective analysis, highlighting suspicious patterns at the graph level, and facilitating more accurate and comprehensive fraud detection (Qin et al., 2022).

Due to their exceptional learning capabilities (Chen et al., 2021b; Li et al., 2022), graph neural networks (GNNs) (Hamilton et al., 2017; Kipf and Welling, 2017) have emerged as prevalent and powerful tools to enhance fraud detection (Dou et al., 2020; Li et al., 2019; Shi et al., 2022; Wang et al., 2019). In this line of research work, fraud detection is reduced to a node classification problem, where GNN models are designed to follow the message-passing paradigm, enabling the propagation of information from labeled nodes to unlabeled ones through the graph structure. Nevertheless, when applying GNNs for fraud detection, two primary challenges arise. (1) Fraudsters tend to intentionally establish relationships between fraudulent and legitimate entities for camouflage and evasion (Dou et al., 2020). This results in a high degree of heterophily among fraud nodes, and their embeddings are significantly smoothed by neighborhood aggregation using GNNs designed for homophily (Yan et al., 2022; Zhu et al., 2021, 2020), rendering them indistinguishable from legitimate ones. (2) Graphs used in fraud detection commonly exhibit a natural imbalance among labeled nodes due to the typical rarity of fraudulent activities. For example, in four real-world fraud detection datasets (detailed in Section 4.1), most imbalance ratios (as defined in Section 2) are below 0.1, indicating that fraudulent instances are significantly outnumbered by legitimate ones. When conventional GNN models are trained on such class-imbalanced graphs, accurately identifying frauds becomes challenging, leading to biased predictions.

Regarding the first challenge, recent methods have been proposed to mitigate the impact of graph heterophily during the aggregation process, which broadly falls into three categories: neighbor extension (Liu et al., 2019; Pei et al., 2020), inter-layer connections (Zhu et al., 2020; Liu et al., 2021a), and adaptive message aggregation (Bo et al., 2021; Jing et al., 2024b; Du et al., 2022). The first two approaches utilize either long-range dependencies or residuals to complement node representations, while their efficacy is limited when tackling graphs with high levels of heterophily. In contrast, the adaptive message aggregation method trains bi-filters to collect neighborhood features that is more adept at managing local homophily and heterophily. However, it may significantly compromise on class-imbalanced graphs due to the use of shared filters across neighbors (Alon and Yahav, 2021), which is thus unsatisfactory for fraud detection with intrinsic imbalance. To tackle the second challenge of class imbalance, data augmentation using oversampling techniques has recently been developed for graphs (Zhao et al., 2021; Park et al., 2022; Duan et al., 2022; Ashmore and Chen, 2023). These methods synthesize new nodes in the embedding space and then generate edges to connect them with existing nodes. However, they may suffer from two major limitations: (1) due to the significant variation in fraudulent activities, there is a natural scattering among fraud nodes, where nodes synthesized based on the assumption of intra-class similarity may lead to distribution shifts; (2) synthesized edges may not accurately represent real-world relationships between nodes due to constraints, while adding such edges may inadvertently introduce noise, potentially undermining the intended enhancement of neighborhood information and diminishing the efficacy of message passing and the resulting node embeddings.

This naturally raises the following question: "*Can we build up a graph-based fraud detection model that can effectively address both heterophily and class imbalance inherent in fraud graphs?*" To answer this question, in this paper, we accordingly propose LACA, which implements fraud detection on graphs using Label-Aware feature aggregation to advance GNN training, which is further regularized by Clustering-Augmented optimization.

The key idea behind LACA is to learn informative and distinguishable node embeddings by independently aggregating information from fraudulent and legitimate neighbors, and incorporating the principle of clustering to regularize cost-sensitive learning and optimize fraudulent and legitimate node distributions in the embedding space. More specifically, instead of applying shared filters across all neighbors that potentially intensifies the impact of class imbalance, LACA designs node-specific filters to dynamically capture the intricate interactions between a node and its fraudulent and legitimate neighbors. These filters are customized to reflect each node's distinct behaviors and characteristics, thus determining the amount of information it may obtain from its neighbors with differing labels. For labeled neighbors, such a node-specific filter—either fraudulent or legitimate—is activated to exclusively aggregate information from the corresponding neighborhood. For unlabeled neighbors, the prediction scores α are calculated to quantify their uncertainty, indicating the likelihood of belonging to the fraudulent class, which, in turn, enables the model to trade off the fraudulent and legitimate implications from unlabeled neighbors and activate both filters to perform gated aggregations. This paradigm not only simplifies adaptive feature aggregation from homophily-heterophily mixed neighborhood, but also avoids the gradients being dominated by the legitimate (i.e., majority class) nodes, thereby mitigating the impact of class imbalance on message passing through the graph structure.

Furthermore, a weighted cross-entropy loss is leveraged to optimize the GNN model against class imbalance. Due to feature diversity and topology complexity, nodes in either fraudulent or legitimate class may still exhibit significantly different semantics (Yang et al., 2023). It is thus crucial to sufficiently explore such semantic divergence within them to enhance label-based optimization. We achieve this by clustering nodes of each class into multiple semantically coherent subclasses in the learned embedding space, ensuring that the node sizes across subclasses are comparable to each other, and then assigning the remaining nodes to the corresponding subclasses to take advantage of unlabeled information. A new clustering score is devised and calculated to evaluate these clusters, which further regularizes model training. This provides fine-grained subclass semantics to improve detection performance, and yields additional benefit in addressing class imbalance. In summary, our major contributions are listed as follows:

- A novel label-aware feature aggregation is designed to address heterophily and class imbalance in graph-based fraud detection.

- A simple yet effective clustering is leveraged to augment model optimization for better detection performance and imbalance mitigation.

- Extensive experiments are conducted to demonstrate LACA's state-of-the-art fraud detection performance on graphs with different imbalance and homophily ratios.

2. Preliminaries

2.1. Notations

A given fraud graph is denoted as $G = (V, E, \mathbf{X})$, where $V(n = |V|)$ is the set of entities (e.g., accounts, reviews, and transactions), E is the set of edges indicating reciprocal links between entities, and $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the feature matrix. Edges E can be further encoded as an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{A}_{ij} = \{0, 1\}$, where if $(v_i, v_j) \in E$, then $\mathbf{A}_{ij} = 1$; otherwise, $\mathbf{A}_{ij} = 0$. The neighbors for v_i is represented as $\mathcal{N}(v_i) = \{v_j | (v_i, v_j) \in E)\}$. Each labeled node is associated with a ground truth $y \in Y = \{0 : \text{legitimate}, 1 : \text{fraudulent}\}$.

2.2. Graph Neural Networks

In this paper, fraud detection is cast as node classification, which aims to learn a GNN model $f_{\mathbf{W}} : (\mathbf{A}, \mathbf{X}) \to \mathbf{y}$ where $\mathbf{W}$ is the model parameters and $\mathbf{y}$ is the set of labels. Generally, GNN models enforce each node to aggregate information from its neighbors and generate higher-level node embedding in a form as follows:

$$\mathbf{H}^{(l)} = \text{aggregate}\left(\mathbf{H}^{(l-1)}, \mathbf{A}, \mathbf{W}^{(l)}\right) \tag{1}$$

where $\mathbf{H}^{(l-1)}$ and $\mathbf{H}^{(l)}$ are the input and output at layer l $(l \geq 1)$, $\mathbf{W}^{(l)}$ is a learnable weight matrix, and $\mathbf{H}^{(0)} = \mathbf{X}$. The final output $\mathbf{Z}$ of GNNs with L layers is computed as:

$$\mathbf{Z} = f_{\mathbf{W}}(\mathbf{A}, \mathbf{X}) = \text{softmax}\left(\mathbf{H}^{(L)}\right) \tag{2}$$

We focus on transductive inferences in this paper where all node connections and features are accessible during training.

2.3. Class Imbalance on Graphs

Fraud graphs exhibit a nature of class imbalance due to the fact that fraudulent entities are often rare. We define the imbalance ratio to quantify such nature to better understand the data challenge for fraud detection on graphs. Specifically, given a fraud graph G where N_{minor} represents the number of fraudulent entities and N_{major} signifies the number of legitimate entities, the imbalance ratio can be written as $r_i = N_{minor}/N_{major}$.

2.4. Homophily and Heterophily

When generalizing to graphs, homophily suggests that nodes tend to connect with others sharing similar features (Zhu et al., 2021). This paper focuses on homophily in class labels (Zhu et al., 2020), where a graph with good homophily indicates that connected nodes share the same label with a high probability. Traditionally, homophily ratio can be defined as the proportion of edges in G that connect nodes sharing the same labels. In fraud graphs, we are more interested in the homophily ratio for the minority (fraudulent) class to better understand the impact of heterophily on fraud nodes. r_h can be thus constrained by

$$r_h = \frac{|\{(v_i, v_j) : (v_i, v_j) \in E \wedge y_{v_i} = y_{v_j} = 1\}|}{|\{(v_i, v_j) : (v_i, v_j) \in E \wedge (y_{v_i} = 1 \vee y_{v_j} = 1)\}|} \tag{3}$$

Heterophily is the opposite of homophily to describe the status of connected nodes belonging to different labels. Graphs with high homophily have $r_h \to 1$, while graphs with high heterophily exhibit low homophily with $r_h \to 0$.

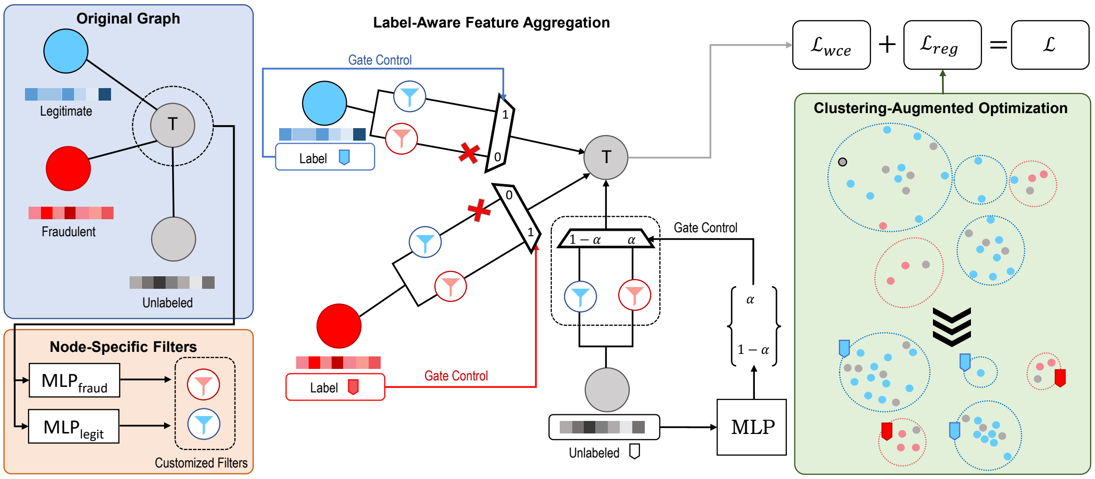


Figure 1: Overview of our proposed model LACA for fraud detection on graphs.

3. Proposed Model: LACA

In this section, we present the technical details of our proposed graph-based fraud detection model LACA, the overview of which is illustrated in Fig. 1.

3.1. Label-Aware Feature Aggregation

3.1.1. Motivation

To retain the discrimination capability in low-homophily graphs, GNNs must harness nodes' surrounding dissimilarities, which prompts the use of high-pass filters (Ekambaram, 2014) to extract neighborhood differences and address heterophily. Following this idea, different adaptive aggregation methods (Bo et al., 2021; Jing et al., 2024b; Du et al., 2022; Pei et al., 2020) have been deployed that train separate filters to extract similar and dissimilar signals from neighbors, enriching node representations while mitigating the impact of heterophily on graph learning. Nevertheless, these approaches have two limitations: (1) filters are shared across all neighbors, risking dominance by majority classes and potentially oversmoothing node embeddings in class-imbalanced graphs; and (2) they are inherently edge-centric, relying on edge properties and neighbor features, which complicates feature aggregation due to the need for multiple combinatorial operations. To address these limitations, we propose a label-aware feature aggregation method that elaborates node-specific filters to separately gather fraudulent and legitimate information from their respective neighbors with the assistance of labeling. This yields two significant advantages: (1) the gradients used to update node-specific filters are less influenced by majority class nodes, enhancing message passing; and (2) the center node determines the aggregation of similar and dissimilar signals, simplifying adaptive aggregation in a mixed homophily-heterophily neighborhood.

3.1.2. Gate Control with Prediction Score

For each center node v_i, label-aware feature aggregation applies two node-specific filters—one fraudulent and one legitimate—for the aggregation of information from its fraudulent and legitimate neighbors, respectively. To adaptively adjust the influence of the two filters on neighborhood aggregation, a gate-control mechanism is employed, which intuitively utilizes neighbor characteristics and their reliable label information to guide the aggregation process. More specifically, when a neighbor v_j is labeled, its label value (either 0 or 1) straightforwardly acts as a binary gate to determine which filter is allowed for aggregating information from that neighbor. For an unlabeled neighbor v_j, the label is uncertain, resulting in mixed information available for aggregation; to account for this uncertainty, a prediction score is calculated on v_j using its embedding $\mathbf{h}_{v_j}^{(l-1)}$, which enables the information aggregation to interpolate between its fraudulent and legitimate implications.

Formally, the gate-control signal, represented by the prediction score α_j for neighbor v_j at aggregation layer l can be defined as follows:

$$\alpha_j = \begin{cases} y_j, & \text{if } v_j \text{ is labeled} \\ \sigma(\mathrm{MLP}(\mathbf{h}_{v_j}^{(l-1)})), & \text{otherwise} \end{cases} \tag{4}$$

where y_j represents the label of v_j if v_j is labeled, with 0 indicating legitimate entities and 1 indicating fraudulent entities. When v_j is unlabeled, its node embedding $\mathbf{h}_{v_j}^{(l-1)}$ is fed into an MLP to derive α_j, with σ as an activation function mapping the output from the MLP to a value between [0, 1]. Different from the clear-cut fraudulent and legitimate label values, the resulting value α_j indicates the likelihood of neighbor node v_j being fraudulent, which naturally serves as a gate-control signal to determine the influence of the node-specific filters and the amount of information the center node v_i may obtain from its neighbor v_j of mixed nature that exhibits characteristics with both fraudulent and legitimate categories.

3.1.3. Feature Aggregation with Node-Specific Filters

As previously discussed, the signal α_j exclusively controls the impact of two node-specific filters on neighborhood aggregation. However, the actual feature aggregation operations depend heavily on the design of these filters, which determine the specific fraudulent or legitimate information that is gathered to contribute to the behavior of the center node. Due to the message passing designed in GNNs that first aggregates the neighbor information and then performs feature learning using $\mathbf{AXW}$, where $\mathbf{W}$ is shared across all center nodes, the influence from differing labeled neighbors is regulated by $\mathbf{W}$ that is accordingly adjusted by the variability of center node attributes. In this respect, we propose to learn these filters with respect to fraudulent and legitimate neighbors using the unique characteristics of the center node to reflect its intrinsic dynamics.

To avoid over-complicating the feature aggregation model, we simply apply two MLPs on the node embedding $\mathbf{h}_{v_i}^{(l-1)}$ of the center node v_i to generate two node-specific filters for the aggregation of fraudulent and legitimate neighbor information. These filters are represented as $\mathbf{W}_{\text{fraud},i}^{(l)}$ and $\mathbf{W}_{\text{legit},i}^{(l)}$ at layer l, which can be calculated as follows:

$$\mathbf{W}_{\text{fraud},i}^{(l)} = \mathrm{MLP}_{\text{fraud}}(\mathbf{h}_{v_i}^{(l-1)}),\ \mathbf{W}_{\text{legit},i}^{(l)} = \mathrm{MLP}_{\text{legit}}(\mathbf{h}_{v_i}^{(l-1)}) \tag{5}$$

The resulting $\mathbf{W}^{(l)}_{\text{fraud},i}$ and $\mathbf{W}^{(l)}_{\text{legit},i}$ are then used as weight matrices for feature aggregation performed on node v_i, which can be specified as follows:

$$\begin{aligned}\mathbf{h}^{(l)}_{v_i} = \sigma(\mathbf{h}^{(l-1)}_{v_i}\mathbf{W}^{(l)}_{\text{center}} + \frac{1}{\sum_{v_j \in \mathcal{N}(v_i)} \alpha_j} \sum_{v_j \in \mathcal{N}(v_i)} \alpha_j \mathbf{h}^{(l-1)}_{v_j}\mathbf{W}^{(l)}_{\text{fraud},i} + \\ \frac{1}{\sum_{v_j \in \mathcal{N}(v_i)} 1-\alpha_j} \sum_{v_j \in \mathcal{N}(v_i)} (1-\alpha_j)\mathbf{h}^{(l-1)}_{v_j}\mathbf{W}^{(l)}_{\text{legit},i})\end{aligned} \tag{6}$$

where at aggregation layer $l \in L$, $\mathbf{W}^{(l)}_{\text{center}}$ is the weight matrix directly applied to the center node embedding; $\mathbf{W}^{(l)}_{\text{fraud},i}$ and $\mathbf{W}^{(l)}_{\text{legit},i}$ are customized to process the fraudulent and legitimate neighbor information for center node v_i. During feature aggregation, local information, fraudulent neighbor information, and legitimate neighbor information is first processed separately, each gated by α_j, and then integrated to form the new center node embedding. To control the scale of aggregated neighborhood information across different categories, we further perform a weighted average based on α_j or $1-\alpha_j$ on the processed fraudulent or legitimate neighbor embeddings for normalization. It is worth noting that our designed node-specific filters capture the distinct features of center nodes while sharing the learning function, striking a balance between learning effectiveness and training cost.

3.2. Clustering-Augmented Optimization

Both fraudulent and legitimate entities often exhibit significant scattered distributions, where instances within each category may not share substantial similarities. Unfortunately, this node feature diversity is further aggravated by complex graph topology, making it challenging for conventional classification to establish clear decision boundaries between fraudulent and legitimate entities. To address this challenge, we propose to explore semantic divergence among nodes and leverage the inherent grouping capabilities of clustering to handle the dispersed nature of nodes. By clustering fraudulent or legitimate nodes into a larger number of clusters, we can isolate smaller groups exhibiting similar behaviors, and enhance the model's sensitivity to subtle patterns of fraud that may be overlooked by simpler classification techniques. Moreover, beyond improving fraud detection performance through regularization, clustering also assists in addressing class imbalance during optimization, as it relies less on the node label information.

3.2.1. Label-Guided Node Clustering

Unlike typical clustering problems, node embeddings in fraud detection scenarios are partially labeled and plagued by severe data imbalance. The goal of clustering in this context goes beyond traditional clustering objectives, which aims not only to uncover the substructures among fraudulent and legitimate nodes, but also to leverage additional information from unlabeled nodes to enrich node semantics and alleviate the impact of class imbalance on model training using subclass information. With this in mind, we design a label-guided node clustering pipeline, which proceeds with the following unique steps:

- Cluster nodes within each class into multiple semantically coherent subclasses in the learned embedding space. Technically, considering a class k (fraudulent or legitimate) and its corresponding nodes $V_k \in V$, we use a selected clustering algorithm, such as k-means, to partition V_k into multiple subclasses. To ensure a balanced distribution of nodes across subclasses, the node sizes across subclasses are comparable to each other.

- Assign unlabeled nodes to the corresponding subclasses. Once all labeled nodes are in place, each unlabeled node is then assigned to the nearest cluster based on the similarity between its embedding and cluster centroid.

The hyperparameters c_{fraud} and c_{legit} serve to regulate the number of clusters for the fraudulent and legitimate classes, respectively. Our clustering pipeline (1) ensures the number of nodes is approximately equal across resulting fraudulent and legitimate subclass, thus mitigating the class-imbalance issue; (2) accumulates abundant node semantic information from different subclasses and by incorporating unlabeled nodes; and (3) forms compact and distinct groupings of distributions in the embedding space with each tightly clustered and separated from others that further facilitates classification training and inference.

3.2.2. Cluster Evaluation

To evaluate the clustering results, we borrow the silhouette score (Rousseeuw, 1987) to refine the general metrics of intra-cluster and inter-cluster distances. However, given that many clusters with the same labels could potentially be merged, enforcing distinct separations among these clusters is unnecessary. Therefore, rather than treating all clusters equally, our evaluation focuses only on differentiating clusters with different labels. More specifically, given node v_i with label y_i that belongs to cluster c_i, we define the inter-cluster distance score as follows:

$$S_{v_i}^{\text{inter}} = \min_{\forall c_j \in \mathbf{C} \setminus c_i, y_i \neq y_j} \text{Dist}(\mathbf{h}_{v_i}, \mathbf{h}_c(v_j)) \tag{7}$$

where Dist() is a distance function, $\mathbf{h}_{v_i}$ refers to the node representation of node v_i, and $\mathbf{h}_c(v_j)$ refers to the centroid of the cluster which v_j belongs to. Similarly, the intra-cluster distance score can be formulated as follows:

$$S_{v_i}^{\text{intra}} = \text{Dist}(\mathbf{h}_{v_i}, \mathbf{h}_c(v_i)) \tag{8}$$

Finally, the clustering score for a given set of nodes will be defined by the average difference between the intra-cluster and inter-cluster distance scores, divided by the maximum of these two, to evaluate how dense the clusters are:

$$S = \frac{1}{|\mathbf{V}|} \sum_{v_i \in \mathbf{V}} \frac{S_{v_i}^{\text{intra}} - S_{v_i}^{\text{inter}}}{\max\{S_{v_i}^{\text{inter}}, S_{v_i}^{\text{intra}}\}} \tag{9}$$

The aforementioned score $\mathcal{S}$ is calculated separately on labeled and unlabeled nodes, as we have less confidence in the cluster assignments of unlabeled nodes. As such, the overall clustering score that evaluates the clustering results can be written as follows:

$$S_{\text{cluster}} = S_{\text{labeled}} + \delta S_{\text{unlabeled}} \tag{10}$$

where δ is a hyperparameter; S_{labeled} and $S_{\text{unlabeled}}$ refers to the result of scoring function in Eq. (9) on labeled nodes and unlabeled nodes, correspondingly.

3.3. Model Optimization

To combine label-aware feature aggregation and clustering-augmented optimization, the overall training procedure of LACA can be described as follows: the given graph G is first fed into the GNN model to perform label-aware feature aggregation and generate node embeddings, which are then used to perform label-guided clustering and calculate the resulting clustering score; based on the predicted results and clustering results, the final loss to be minimized is formulated, which consists of a supervised loss and a regularization loss. The supervised loss is a weighted cross-entropy loss function $\mathcal{L}_{\text{wce}}$, which provides cost-sensitive learning to mitigate class imbalance:

$$\mathcal{L}_{\text{wce}} = \frac{-\sum_{v_i \in V} |V_{\text{legit}}| y_i \log(p_f) + |V_{\text{fraud}}|(1-y_i)\log(1-p_f)}{|V_{\text{legit}}| + |V_{\text{fraud}}|} \tag{11}$$

where p_f is the predicted probability of node i to be fraudulent; y_i is the ground truth label of node i; $|V_{\text{fraud}}|$ and $|V_{\text{legit}}|$ are the number of labeled fraudulent nodes and legitimate nodes in the training set, where the ratios of them are used as an estimate for the overall imbalance ratio. The division of the total number of labeled nodes controls the scale of the weighted cross-entropy loss. The overall clustering score discussed in Section 3.2.2 is used as the regularization loss $\mathcal{L}_{\text{reg}} = \mathcal{S}_{\text{cluster}}$ to enforce more distinguishable decision boundaries. The final loss can then be expressed as:

$$\mathcal{L} = \mathcal{L}_{\text{wce}} + \lambda \mathcal{L}_{\text{reg}} \tag{12}$$

where λ is a hyperparameter controlling the influence of the clustering-based regularization. In summary, with the label-aware aggregation scheme mitigating the issues of heterophily and class imbalance existing in fraud graphs, weighted cross-entropy loss mitigating the bias introduced by class imbalance, and clustering-augmented optimization enforcing better decision boundary, LACA allows for a more nuanced analysis of the dataset, facilitating the development of a more accurate and reliable fraud detection model.

4. Experimental Results and Analysis

4.1. Experimental Setup

Datasets. We evaluate LACA on four real-world fraud detection datasets:

- **YelpChi** (Rayana and Akoglu, 2015): This dataset identifies anomalous reviews on Yelp.com that promote or demote products or businesses. It features a graph with three types of edges: R-U-R, R-S-R, and R-T-R.
- **Amazon** (McAuley and Leskovec, 2013): This dataset detects compensated users who write counterfeit reviews for musical instruments on Amazon.com. It includes three types of relations: U-P-U, U-S-U, and U-V-U.
- **T-Finance** (Tang et al., 2022): This dataset identifies anomalous user accounts within transaction networks. Node features include registration duration, login activities, and interaction rates, while edges signify transactional account interactions.

Table 1: Statistics of the datasets (Hom. Ratio reports r_h for minority class)

Dataset	**#Nodes**	**#Edges**	**#Features**	**Imb. R**	**Hom. R**
Amazon	11,944	4,398,392	25	0.07	0.04
YelpChi	45,954	3,846,979	32	0.17	0.10
T-Finance	39,357	42,445,086	10	0.05	0.30
Elliptic	203,769	234,355	166	0.02	0.12

- **Elliptic** (Weber et al., 2019): The dataset categorizes Bitcoin transactions into legal entities (e.g., wallet providers and miners) and illegal entities (e.g., scams, malware, and terrorists). It is structured as a graph where nodes are transactions and edges represent the flow of Bitcoin.

We use 40%-30%-30% data-split across all four datasets to train, validate, and test models, respectively. The data statistics are summarized in Table 1. In the experiments, all relationships in YelpChi and Amazon datasets are considered the same to align with homogeneous GNNs. The timestamp in the Elliptic dataset is excluded from data splitting and removed before being fed into the model.

Baselines. We select 15 different models as our baselines to compare with LACA. These baselines include four conventional GNNs (**GCN** (Kipf and Welling, 2017), **GAT** (Veličković et al., 2017), **GIN** (Xu et al., 2018a), and **GraphSAGE** (Hamilton et al., 2017)), two heterophily-aware GNNs (**JKGCN** (Xu et al., 2018b), **GPRGNN** (Chien et al., 2021)), and nine advanced models for imbalanced graphs or fraud detection (**GraphENS** (Park et al., 2022), **GraphSMOTE** (Zhao et al., 2021), **Graph-Consis** (Liu et al., 2020), **Care-GNN** (Dou et al., 2020), **PC-GNN** (Liu et al., 2021b), **DOS-GNN** (Jing et al., 2024a), **H2-Fdetector** (Shi et al., 2022), **BWGNN** (Tang et al., 2022)), **GDN** (Gao et al., 2023). Some of the reported results in this paper are taken from their original papers.

Implementation Details. The number of aggregation layers L is set to 2 for LACA. All models are trained for 2,000 epochs with a patience of 200 using Adam optimizer with learning rate $lr = 0.01$ and $5e-4$ L2 regularization. AUC (Area Under the Receiver Operating Characteristic Curve) and F1-Macro are the primary evaluation metrics to provide insight into a model's effectiveness on class-imbalanced fraud detection. All MLP models will be tuned between 1 and 2 layers, and the coefficients δ and λ are tuned in the range of $[0.1, 0.5]$ and $[0.1, 0.7]$, correspondingly. The experiment is run on AMD EPYC 7643 48-core processors with an NVIDIA A100-sxm4-80GB GPU.

4.2. Comparison with Baselines

In this section, we would like to evaluate the effectiveness of LACA for class-imbalanced fraud detection on graphs by comparing our model with 15 selected baselines over four different public fraud datasets. The comparative results are reported in Table 2. From Table 2, traditional GNN models exhibit poor performance in class-imbalanced fraud graphs. In contrast, advanced models designed for addressing heterophily, imbalanced data, or fraud detection show significant improvement in detection performance, due to their enhancement in handling heterophilic neighborhood and class imbalance, preventing fraud nodes from be-

Table 2: Comparison of different Fraud Detection methods (%) on benchmark datasets. Some GNN models are highlighted: **bold** statistics denote the best results. OOM indicates that the machine runs out of memory before the algorithm terminates.

Dataset	**Amazon**		**YelpChi**		**T-Finance**		**Elliptic**	
	AUC	**F1**	**AUC**	**F1**	**AUC**	**F1**	**AUC**	**F1**
GCN	74.34±1.2	67.47±7.2	52.47±0.6	54.31±0.7	64.43±0.7	70.74±1.0	90.47±1.1	83.13±0.9
GAT	75.16±1.8	83.18±4.1	56.24±0.3	54.64±2.3	73.00±1.2	53.86±1.0	63.87±0.8	47.43±0.8
GIN	80.56±2.8	69.26±5.5	74.09±0.8	62.85±1.1	80.02±0.8	65.23±1.3	93.71±1.0	88.78±0.9
GraphSAGE	75.27±0.8	74.17±0.6	54.00±0.2	65.49±0.8	67.12±0.3	52.71±0.5	94.35±0.6	89.04±0.5
JKGCN	89.63±0.8	72.52±0.5	80.74±0.4	59.75±1.6	93.92±0.2	85.39±0.6	93.60±0.5	85.98±0.5
GPRGNN	92.79±0.6	85.47±1.8	81.03±0.7	65.46±0.8	94.25±0.9	87.73±0.4	94.49±0.7	89.71±0.7
GraphENS	80.01±1.8	52.81±1.9	60.12±1.5	47.63±3.3	OOM	OOM	83.43±1.1	51.35±1.3
GraphSMOTE	90.79±1.0	88.36±1.5	76.74±0.9	65.22±2.1	OOM	OOM	91.46±0.9	86.81±1.5
Graph-Consis	87.41±0.4	75.12±0.5	69.83±0.3	58.70±0.7	91.42±0.5	73.46±0.8	93.93±0.3	89.28±0.4
PC-GNN	95.86±0.1	89.56±0.8	79.87±0.2	63.00±2.3	91.23±0.6	63.18±0.5	94.39±0.4	91.02±0.5
Care-GNN	89.73±1.5	86.39±1.8	75.70±3.0	63.32±1.2	92.16±1.0	77.55±0.9	94.12±1.1	90.55±1.1
H2FDetector	96.11±0.8	86.86±0.9	89.62±1.3	74.39±2.5	94.55±0.7	73.87±0.8	95.91±1.0	90.82±0.7
DOS-GNN	96.55±1.1	92.10±0.9	81.15±1.2	70.46±2.5	96.01±0.6	88.53±0.9	96.32±0.9	92.75±0.8
BWGNN	97.41±0.5	91.72±0.9	90.61±0.6	76.89±0.9	95.82±0.5	88.90±0.6	96.35±0.2	90.95±0.5
GDN	97.02±0.2	90.23±0.4	90.32±0.8	75.99±0.6	95.61±0.9	88.92±2.1	95.80±0.8	90.72±0.7
LACA	**97.71±0.4**	**91.77±0.9**	**93.28±0.3**	**80.64±0.6**	**97.00±0.3**	**91.11±0.6**	**97.03±0.5**	**92.99±0.6**

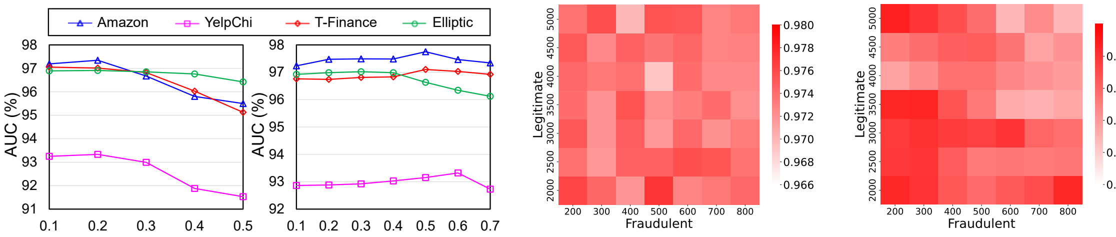
ing overshadowed by many legitimate nodes. Even so, our proposed LACA still manages to advance the state-of-the-art performance to a higher level. Compared to the best results of traditional GNN models, LACA improves the AUC and F1 by 23.37% and 24.30% for Amazon, 40.81% and 26.33% for YelpChi, 32.57% and 30.37% for T-Finance, 6.56% and 9.86% for Elliptic. Compared to the best results of enhanced models for heterophily, imbalanced data, or fraud detection, LACA further improves the results for all tested datasets.

In summary, LACA achieves state-of-the-art performance across all four public benchmarks and outperforms the leading GNN models. This comparative study confirms that the combination of label-aware feature aggregation and clustering-augmented regularization loss can effectively extract and propagate neighborhood information to derive discriminative node embedding, while forcing clearer decision boundaries, thus boosting model discrimination capability in the graphs with heterophily and class imbalance.

4.3. Parameter Evaluation

In this section, we would like to study the sensitivity of the model on the most important hyperparameters: the coefficient δ which controls the impact of unlabeled data in the calculation of clustering loss; the coefficient λ which controls the influence of the clustering loss function in the training process; and the number of pre-set clusters c_{legit} for legitimate class and c_{fraud} for fraud class, which will influence how the nodes will be grouped. The performance is expected to vary depending on the chosen value of these parameters. From the experimental results reported in Fig. 2, we make the following observations:

Impact of λ. For coefficient λ, it can be observed that the performance of the model reaches its highest peak at values 0.5, 0.5, 0.3, and 0.2 for Amazon, YelpChi, T-Finance, and Elliptic respectively, indicating that the inclusion of clustering loss helps improve the overall performance of the model.



(a) δ **(b)** λ **(c) #Cluster-Amazon** **(d) #Cluster-YelpChi**

Figure 2: Impact of parameters on LACA in terms of AUC (%): (a) different δ to adjust clustering score; (b) different λ to balance loss function; (c) different cluster numbers for legitimate and fraud class on Amazon dataset; (d) different cluster numbers for legitimate and fraud class on YelpChi dataset.

Table 3: Ablation study in terms of AUC (%): LAFA denotes Label-Aware Feature Aggregation and CAO denotes Clustering-Augmented Optimization.

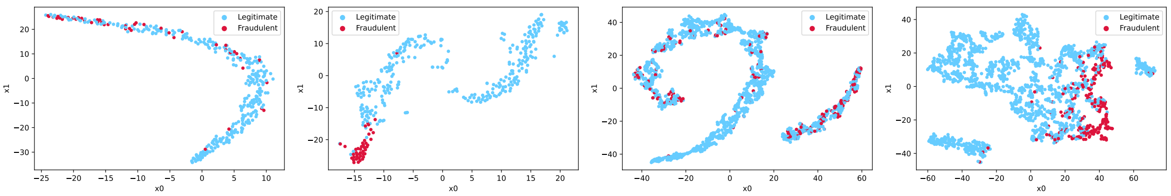
GCN	LAFA	CAO	Amazon	YelpChi	T-Finance	Elliptic
✓			74.34±1.23	52.47±0.59	64.43±0.72	90.47±1.12
	✓		97.14±0.17	92.89±0.18	96.56±0.25	95.98±0.16
	✓	✓	**97.71±0.41**	**93.28±0.33**	**97.00±0.39**	**97.03±0.50**

Impact of δ. For coefficient δ, the performance of the model reaches its highest peak at values 0.2, 0.2, 0.1, and 0.2 for Amazon, YelpChi, T-Finance, and Elliptic, respectively. It can be observed that the performance of our model starts to drop when δ exceeds 0.2 on most of the datasets, indicating that the weighted cross entropy loss still plays a crucial role during the training despite the support of clustering augmentation.

Impact of cluster numbers. The number of clusters is expected to influence how the node representations will be clustered and determine the final performance of the model. Here we provide a heat map to show the influence of the cluster number selection. As illustrated in Fig. 2(c) and (d), while the initial cluster number setting has a smaller impact on the Amazon dataset, the YelpChi heatmap has shown a clearer trend that a smaller number of clusters is preferred for the regularization term to provide support for the model. Nevertheless, It is worth noting that these fluctuations are relatively small, implying that our model exhibits high stability across different cluster number selections.

4.4. Ablation Study

LACA leverages two critical designs: label-aware feature aggregation to derive informative and distinguishable node embeddings and clustering-augmented optimization for regularization. To demonstrate their necessity and benefit, we conduct an ablation study to assess the contributions of these two components to our model's performance by including/excluding them and construct three alternative models. As illustrated in Table 3, both components contribute to the performance of LACA. Notably, label-aware feature aggregation emerges as the most significant contributor among two components. The findings show that even without clustering-augmented optimization, a model with label-aware feature aggregation



(a) GCN-Amazon **(b) LACA-Amazon** **(c) GCN-YelpChi** **(d) LACA-YelpChi**

Figure 3: Visualization of node embeddings derived by GCN and LACA.

alone registers a performance increase exceeding 20% on datasets such as Amazon, YelpChi, and T-Finance compared to the baseline GCN model. This outcome aligns with our expectations. The core functionality of LACA relies on the aggregation, while collaborating with clustering-based regularization further alleviates the impact of class imbalance on graphs, and boosts the model's performance.

4.5. Case Study

To validate our claim that LACA provides more distinguishable node embeddings in graphs on fraud datasets, we present a brief case study to showcase the difference between embeddings generated by GCN and LACA. Due to the page limit, we only present results on two datasets, Amazon and YelpChi. We map node embeddings into a two-dimensional space using t-SNE, and the resulting embeddings are visualized in Fig. 3. For Amazon, GCN fails to generate clear clusters between fraud and non-fraud points and suffers from fuzzy boundaries (Fig. 3(a)); in contrast, LACA exhibits better-distinguished boundaries with higher cohesion for fraud nodes(Fig. 3(b)). For YelpChi, GCN again fails to generate any associations with fraud points being scattered among legitimate points (Fig. 3(c)), while LACA redistributes the nodes, making them further apart (Fig. 3(d)). These observations reaffirm the effectiveness of LACA in learning more distinct node embeddings for fraud detection in graphs characterized by heterophily and class imbalance.

5. Conclusion

In this paper, we introduce LACA for fraud detection on graphs with class imbalance and a mixture of homophily and heterophily. LACA consists of label-aware feature aggregation and employs clustering-augmented optimization, both of which contribute to the solutions that address the issues of class imbalance and graph heterophily. Label-aware feature aggregation enables LACA to create informative node embeddings using node-specific filters that dynamically capture the interactions between a node and its varied neighbors, while clustering-augmented optimization leverages the inherent grouping capabilities of clustering algorithms to handle the dispersed nature of fraudulent and legitimate nodes. Overall, these strategies enhance the ability of LACA to discern subtle patterns indicative of fraud, making it a more robust tool in the detection of fraudulent activities within networked systems. Evaluation through extensive experiments demonstrates that our model achieves state-of-the-art performance, which affirms its effectiveness in class-imbalanced node classification and practical significance in handling fraud detection tasks on graphs with heterophily.

Acknowledgments

L. Chen's work is partially supported by the NSF under grant CNS-2245968.

References

Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. *International Conference on Learning Representations (ICLR)*, 2021.

Bradley Ashmore and Lingwei Chen. Hover: Homophilic oversampling via edge removal for class-imbalanced bot detection on graphs. In *CIKM*, pages 3728–3732, 2023.

Richard Bauder, Taghi M Khoshgoftaar, and Naeem Seliya. A survey on the state of healthcare upcoding fraud analysis and detection. *HSOR*, 17:31–55, 2017.

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

Lingwei Chen, Yujie Fan, and Yanfang Ye. Adversarial reprogramming of pretrained neural networks for fraud detection. In *CIKM*, pages 2935–2939, 2021a.

Lingwei Chen, Xiaoting Li, and Dinghao Wu. Enhancing robustness of graph convolutional networks via dropping graph connections. In *ECML PKDD*, pages 412–428, 2021b.

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *ICLR*, 2021.

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 *CIKM*, pages 315–324, 2020.

Lun Du, Xiaozhou Shi, Qiang Fu, Xiaojun Ma, Hengyu Liu, Shi Han, and Dongmei Zhang. Gbk-gnn: Gated bi-kernel graph neural networks for modeling both homophily and heterophily. In *WWW*, pages 1550–1558, 2022.

Yijun Duan, Xin Liu, Adam Jatowt, Hai-tao Yu, Steven Lynden, Kyoung-Sook Kim, and Akiyoshi Matono. Anonymity can help minority: A novel synthetic data over-sampling strategy on multi-label graphs. In *ECML PKDD*, pages 20–36, 2022.

Venkatesan Nallampatti Ekambaram. *Graph-structured data viewed through a Fourier lens.* University of California, Berkeley, 2014.

Shangbin Feng, Zhaoxuan Tan, Rui Li, and Minnan Luo. Heterogeneity-aware twitter bot detection with relational graph transformers. In *AAAI*, pages 3977–3985, 2022.

Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. Alleviating structural distribution shift in graph anomaly detection. In *WSDM*, pages 357–365, 2023.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *NIPS*, 30, 2017.

Bryan Hooi, Kijung Shin, Hyun Ah Song, Alex Beutel, Neil Shah, and Christos Faloutsos. Graph-based fraud detection in the face of camouflage. *TKDD*, 11(4):1–26, 2017.

Shixiong Jing, Lingwei Chen, Quan Li, and Dinghao Wu. Dos-gnn: Dual-feature aggregations with over-sampling for class-imbalanced fraud detection on graphs. In *IJCNN*, pages 1–8, 2024a.

Shixiong Jing, Lingwei Chen, Quan Li, and Dinghao Wu. H2gnn: Graph neural networks with homophilic and heterophilic feature aggregations. In *DASFAA*, pages 342–352, 2024b.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ICLR*, 2017.

Ao Li, Zhou Qin, Runshi Liu, Yiqun Yang, and Dong Li. Spam review detection with graph convolutional networks. In *CIKM*, pages 2703–2711, 2019.

Quan Li, Xiaoting Li, Lingwei Chen, and Dinghao Wu. Distilling knowledge on text graph for social media attribute inference. In *SIGIR*, pages 2024–2028, 2022.

Songtao Liu, Lingwei Chen, Hanze Dong, Zihao Wang, Dinghao Wu, and Zengfeng Huang. Higher-order weighted graph convolutional networks. *arXiv:1911.04129*, 2019.

Xiaorui Liu, Jiayuan Ding, Wei Jin, Han Xu, Yao Ma, Zitao Liu, and Jiliang Tang. Graph neural networks with adaptive residual. *NeurIPS*, 34:9720–9733, 2021a.

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, 2021b.

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 *SIGIR*, pages 1569–1572, 2020.

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

Joonhyung Park, Jaeyun Song, and Eunho Yang. GraphENS: Neighbor-aware ego network synthesis for class-imbalanced node classification. In *ICLR*, 2022.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287*, 2020.

Tahereh Pourhabibi, Kok-Leong Ong, Booi H Kam, and Yee Ling Boo. Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems*, 133:113303, 2020.

Zidi Qin, Yang Liu, Qing He, and Xiang Ao. Explainable graph-based fraud detection via neural meta-graph search. In *CIKM*, pages 4414–4418, 2022.

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

Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

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 *WWW*, pages 1486–1494, 2022.

Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. Rethinking graph neural networks for anomaly detection. In *ICML*, pages 21076–21089, 2022.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Jianyu Wang, Rui Wen, Chunming Wu, Yu Huang, and Jian Xiong. Fdgars: Fraudster detection via graph convolutional networks in online app review system. In *WWW*, 2019.

Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I Weidele, Claudio Bellei, Tom Robinson, and Charles E Leiserson. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv:1908.02591*, 2019.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018a.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *ICML*, pages 5453–5462. PMLR, 2018b.

Yujun Yan, Milad Hashemi, Kevin Swersky, Yaoqing Yang, and Danai Koutra. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. In *ICDM*, pages 1287–1292. IEEE, 2022.

Xihong Yang, Yue Liu, Sihang Zhou, Siwei Wang, Wenxuan Tu, Qun Zheng, Xinwang Liu, Liming Fang, and En Zhu. Cluster-guided contrastive graph clustering network. In *AAAI*, pages 10834–10842, 2023.

Tianxiang Zhao, Xiang Zhang, and Suhang Wang. Graphsmote: Imbalanced node classification on graphs with graph neural networks. In *WSDM*, pages 833–841, 2021.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *NeurIPS*, 33:7793–7804, 2020.

Jiong Zhu, Ryan A Rossi, Anup Rao, Tung Mai, Nedim Lipka, Nesreen K Ahmed, and Danai Koutra. Graph neural networks with heterophily. In *AAAI*, 2021.