



Universiteit
Leiden
The Netherlands

Heterophily-based graph neural network for imbalanced classification

Liang, Z.; Li, Y.; Huang, T.; Saxena, A.; Pei, Y.; Pechenizkiy, M.; ... ; Donduran, M.

Citation

Liang, Z., Li, Y., Huang, T., Saxena, A., Pei, Y., & Pechenizkiy, M. (2024). Heterophily-based graph neural network for imbalanced classification. *Studies In Computational Intelligence*, 74-86. doi:10.1007/978-3-031-53468-3_7

Version: Accepted Manuscript

License: [Licensed under Article 25fa Copyright Act/Law \(Amendment Taverne\)](#)

Downloaded from: <https://hdl.handle.net/1887/3748314>

Note: To cite this publication please use the final published version (if applicable).

Heterophily-Based Graph Neural Network for Imbalanced Classification

Zirui Liang¹, Yuntao Li¹, Tianjin Huang¹, Akрати Saxena², Yulong Pei¹, and Mykola Pechenizkiy¹

¹ Eindhoven University of Technology, Eindhoven, The Netherlands
`{y.pei.1, m.pechenizkiy}@tue.nl`

² Leiden Institute of Advanced Computer Science,
Leiden University, Leiden, The Netherlands
`a.saxena@liacs.leidenuniv.nl`

Abstract. Graph neural networks (GNNs) have shown promise in addressing graph-related problems, including node classification. However, in real-world scenarios, data often exhibits an imbalanced, sometimes highly-skewed, distribution with dominant classes representing the majority, where certain classes are severely underrepresented. This leads to a suboptimal performance of standard GNNs on imbalanced graphs. In this paper, we introduce a unique approach that tackles imbalanced classification on graphs by considering graph heterophily. We investigate the intricate relationship between class imbalance and graph heterophily, revealing that minority classes not only exhibit a scarcity of samples but also manifest lower levels of homophily, facilitating the propagation of erroneous information among neighboring nodes. Drawing upon this insight, we propose an efficient method, called **Fast Im-GBK**, which integrates an imbalance classification strategy with heterophily-aware GNNs to effectively address the class imbalance problem while significantly reducing training time. Our experiments on real-world graphs demonstrate our model’s superiority in classification performance and efficiency for node classification tasks compared to existing baselines.

Keywords: Graph neural networks, Imbalanced classification, Heterophily

1 Introduction

GNNs have gained popularity for their accuracy in handling graph data. However, their accuracy, like other deep learning models, is highly dependent on data quality. One major challenge is class imbalance, where some classes have far fewer examples than others. This can lead to biased classification results, favoring the majority class while neglecting the minority classes [5]. The issue of imbalanced datasets commonly arises in classification and recognition tasks, where accurate classification of minority classes is critical. Graph imbalance classification has real-world applications, like identifying spammers in social networks [18] and detecting fraud in financial networks [8]. In these cases, abnormal nodes are

rare, making graph imbalance classification very challenging. Finding effective solutions to this problem is valuable for both research and practical applications.

The class-imbalanced problem has been extensively studied in machine learning and deep learning, as evident by prior research [6]. However, these methods may not effectively handle imbalanced graph data due to the interconnected nature of nodes within graphs. Graph nodes are characterized not only by their own properties but also by the properties of their neighboring nodes, introducing non-i.i.d. (independent and identically distributed) characteristics. Recent studies on graph imbalance classification have focused on data augmentation techniques, such as GraphSMOTE [19] and GraphENS [12]. However, our observations indicate that class imbalance in graphs is often accompanied by heterophilic connections of minority nodes, where minority nodes have more connections with nodes of diverse labels than the majority class nodes. This finding suggests that traditional techniques may be insufficient in the presence of heterophily.

To address this challenge, we propose incorporating a graph heterophily handling strategy into graph imbalanced classification. Our approach builds upon the bi-kernel design of GBK-GNN [2] to capture both homophily and heterophily within the graph. Additionally, we introduce a class imbalance-aware loss function, such as logit adjusted loss, to appropriately reweight minority and majority nodes. The complexity of GBK-GNN makes training computationally challenging. To overcome this, we propose an efficient version of the GBK-GNN that achieves both efficacy and efficiency in training.

Our main contributions are as follows: (1) We provide comprehensive insights into the imbalance classification problem in graphs from the perspective of graph heterophily and investigate the relationship between class imbalance and heterophily. (2) We present a novel framework that integrates graph heterophily and class-imbalance handling based on the insights and its fast implementation that significantly reduces training time. (3) We conduct extensive experiments on various real-world graphs to validate the effectiveness and efficiency of our proposed framework in addressing imbalanced classification on graphs.

2 Related Work

Imbalanced Classification Efforts to counter class imbalance in classification entail developing unbiased classifiers that account for label distribution in training data. Existing strategies fall into three categories: loss modification, post-hoc correction, and re-sampling techniques. Loss modification adjusts the objective function by assigning greater weights [5] to minority classes. Post-hoc correction methods [11] adapt logits during inference to rectify underrepresented minority class predictions. Re-sampling employs techniques, such as sampling strategies [13] or data generation [1], to augment minority class data. The widely utilized Synthetic Minority Over-sampling Technique (SMOTE) [1] generates new instances by merging minority class data with nearest neighbors.

To tackle class imbalance in graph-based classification, diverse approaches harness graph structural information to mitigate the challenge. GraphSMOTE

[19] synthesizes minor nodes by interpolating existing minority nodes, with connectivity guided by a pretrained edge predictor. The Topology-Aware Margin (TAM) loss [16] considers each node’s local topology by comparing its connectivity pattern to the class-averaged counterpart. When nearby nodes in the target class are denser, the margin for that class decreases. This change enhances learning adaptability and effectiveness through comparison. GraphENS [12] is another technique that generates an entire ego network for the minor class by amalgamating distinct ego networks based on similarity. These methods effectively combat class imbalance in graph-based classification, leveraging graph structures and introducing inventive augmentation techniques.

Heterophily Problem In graphs, homophily [10] suggests that nodes with similar features tend to be connected, and heterophily suggests that nodes with diverse features and class labels tend to be connected. Recent investigations have analyzed the impact of heterophily on various tasks, emphasizing the significance of accounting for attribute information and devising methods attuned to heterophily [20]. Newly proposed models addressing heterophily can be classified into two categories: non-local neighbor extension and GNN architecture refinement [20]. Methods grounded in non-local neighbor extension seek to alleviate this challenge through neighborhood exploration. The H2GCN method [22], for instance, integrates insights from higher-order neighbors, revealing that two-hop neighbors frequently encompass more nodes of the same class as the central ego node. NLGNN [9] follows a pathway of employing attention mechanisms or pointer networks to prioritize prospective neighbor nodes based on attention scores or their relevance to the ego node. Approaches enhancing GNN architectures aspire to harness both local and non-local neighbor insights, boosting model capacity by fostering distinct and discerning node representations. A representative work is GBK-GNN (Gated Bi-Kernel Graph Neural Network) [2], which employs dual kernels to encapsulate homophily and heterophily details and introduces a gate to ascertain the kernel suitable for a given node pair.

3 Motivation

Node classification on graphs, such as one performed by the Graph Convolutional Network (GCN), differs fundamentally from non-graph tasks due to the interconnectivity of nodes. In imbalanced class distributions, minority nodes may have a higher proportion of heterophilic edges in their local neighborhoods, which can negatively impact classification performance.

To investigate the relationship between homophily and different classes, especially minorities, we conducted a small analysis on four datasets: Cora, CiteSeer, Wiki, and Coauthor CS (details about datasets can be found in Section 5.1). Our analysis involves computing the average homophily ratios and calculating node numbers across different categories. In particular, the average homophily ratio for nodes with label y is defined as:

$$h(y, \mathcal{G}_V) = \frac{1}{|\mathcal{V}_y|} \sum_{i \in \mathcal{V}_y} \frac{|\mathcal{N}_{i^s}|}{|\mathcal{N}_i|} \quad (1)$$

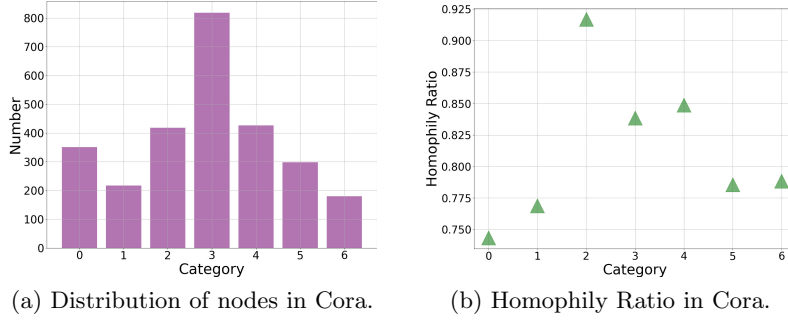


Fig. 1: Category distributions (left) and average homophily ratios (right) of Cora.

where $\mathcal{V}_y$ represents a set of nodes with label y , $\mathcal{N}i$ is the set of neighbors of node v_i (excluding v_i) in graph $\mathcal{G}$, and $\mathcal{N}_i^s$ is the set of neighbors (excluding v_i) whose class is the same as v_i .

Observations. The results for Cora datasets are shown in Fig. 1. It is evident that the average homophily ratios of minority classes are relatively smaller suggesting higher proportions of heterophilic edges in their local neighborhoods. This could negatively affect classification performance when using existing imbalance strategies like data augmentation and loss reweight. This finding highlights the importance of considering the homophily and heterophily properties of nodes when designing graph classification models. We propose a novel approach that addresses imbalance issue effectively, considering node heterophily.

Problem Formulation. An attributed graph is denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, where $\mathcal{V} = \{v_1, \dots, v_n\}$ represents the set of n nodes in $\mathcal{G}$, and $\mathcal{E}$ is the set of edges with edge e_{ij} connecting nodes v_i and v_j . For simplicity, we consider undirected graphs, but the argument could be generalized for directed graphs. The node attribute matrix is represented by $\mathbf{X} = [\mathbf{x}_1^\top, \dots, \mathbf{x}_n^\top] \in \mathcal{R}^{n \times d}$, where $\mathbf{x}_i$ indicates the feature of node v_i . $\mathcal{Y}$ denotes the label information for nodes in $\mathcal{G}$ and each node v_i is associated with a label $y_i \in \mathcal{R}^n$. During the training, only a subset of the classes, denoted by $\mathcal{Y}_{\mathcal{L}}$, is available containing the labels for node subset $\mathcal{V}_{\mathcal{L}}$. Moreover, $\mathcal{C} = \{c_1, c_2, \dots, c_m\}$ represents the set of m classes assigned to each node, with $|C_i|$ denoting the size of the i -th class, which indicates the number of nodes with class c_i . To quantify the degree of imbalance in the graph, we use the imbalance ratio, defined as $\mathbf{r} = \frac{\max_i(|C_i|)}{\min_i(|C_i|)}$. For imbalanced graphs, the imbalance ratio of $\mathcal{Y}_{\mathcal{L}}$ is high (Table 1 of Section 5 shows that the first two datasets have relatively balanced classes and the last two have imbalanced classes).

Imbalance Node Classification on Graphs. In the context of an imbalanced graph $\mathcal{G}$ with a high imbalance ratio $\mathbf{r}$, the aim is to develop a node classifier $f : f(\mathcal{V}, \mathcal{E}, \mathbf{X}) \rightarrow \mathcal{Y}$ that can work well for classifying nodes belonging to both the majority and minority classes.

4 Methodology

In this section, we present our solution to address class imbalance that incorporates heterophily handling and imbalance handling components (Section 4.1). We also propose a fast version that effectively reduces training time (Section 4.2). The main objective of our model is to minimize the loss of minority classes while ensuring accurate information exchange during the message-passing process.

4.1 Im-GBK

Heterophily Handling We build our model based on the GBK-GNN [2] model, which is a good model for graph classification, though not able to handle class imbalance. GBK-GNN is designed to address the lack of distinguishability in GNN, which stems primarily from the incapability to adjust weights adaptively for various node types based on their distinct homophily properties. As a consequence, a bi-kernel feature transformation method is employed to capture either homophily or heterophily information. In this work, we, therefore, introduce a learnable kernel-based selection gate that aims to distinguish if a pair of nodes are similar or not and then selectively choose appropriate kernels, i.e., homophily or heterophily kernel. The formal expression for the input transformation is presented below.

$$\mathbf{h}_i^{(l)} = \sigma \left(\mathbf{W}_f \mathbf{h}_i^{(l-1)} + \frac{1}{|\mathcal{N}(v_i)|} \sum_{v_j \in \mathcal{N}(v_i)} \alpha_{ij} \mathbf{W}_s \mathbf{h}_j^{(l-1)} + (1 - \alpha_{ij}) \mathbf{W}_d \mathbf{h}_j^{(l-1)} \right) \quad (2)$$

$$\alpha_{ij} = \text{Sigmoid} \left(\mathbf{W}_g \left[\mathbf{h}_i^{(l-1)}, \mathbf{h}_j^{(l-1)} \right] \right) \quad (3)$$

$$\mathcal{L} = \mathcal{L}_0 + \lambda \sum_l^L \mathcal{L}_g^{(l)} \quad (4)$$

where $\mathbf{W}_s$ and $\mathbf{W}_d$ are the kernels for homophilic and heterophilic edges, respectively. The value of α_{ij} is determined by $\mathbf{W}_g$ and the embedding layer of nodes i and j . The loss function consists of two parts: $\mathcal{L}_0$, a cross-entropy loss for node classification, and $\mathcal{L}_g^{(l)}$, a label consistency-based cross-entropy, i.e., to discriminate if labels of a pair of nodes are consistent for each layer l to guide the training of the selection gate or not. A hyper-parameter λ is introduced to balance the two losses. The original GBK-GNN method does not explicitly address the class imbalance issue, which leads to the model being biased toward the majority class. Our method employs a class-imbalance awareness for the GBK-GNN design, and therefore it mitigates the bias.

Class-Imbalance Handling with Logit Adjusted Loss When traditional softmax is used on such imbalanced data, it can result in biases toward the majority class. This is because the loss function used to train the model typically treats all classes equally, regardless of their frequency. As a result, the model tends to optimize for overall accuracy by prioritizing the majority classes

while performing poorly on the minority classes. This issue can be addressed by adjusting the logits (i.e., the inputs to the softmax function) for each class to be inversely proportional to their frequencies in the training data, which effectively reduces the weight of the majority classes and increases the weight of the minority classes [11]. In this study, we calculate logit-adjusted loss as follows:

$$\mathcal{L}_{\text{logit-adjusted}} = -\log \frac{e^{f_y(x) + \tau \cdot \log \pi_y}}{\sum_{y' \in [L]} e^{f_{y'}(x) + \tau \cdot \log \pi_{y'}}} \quad (5)$$

where π_y is the estimate of the class prior. In this approach, a label-dependent offset is added to each logit, which differs from the standard softmax cross-entropy approach. Additionally, the class prior offset is enforced during the learning of the logits rather than being applied post-hoc, as in other methods.

Class-Imbalance Handling with Balanced Softmax Another approach called balanced softmax [13] focuses on assigning larger weights to the minority classes and smaller weights to the majority class, which encourages the model to focus more on the underrepresented classes and improve their performance. The traditional softmax function treats all classes equally, which can result in a bias towards the majority class in imbalanced datasets. Balanced softmax, on the other hand, adjusts the temperature parameter of the softmax function to balance the importance of each class, effectively reducing the impact of the majority class and increasing the impact of the minority classes. Formally, given N_k , l_v , y_v , l_{v,y_v} represent k -th of class N , logit and the label of node v , logit associated with the true label y_v for the input node v , respectively. In this study, the balanced softmax is calculated as:

$$\mathcal{L}_{\text{balanced-softmax}} = -\log \frac{e^{l_{v,y_v} + \log N_{y_v}}}{\sum_{k \in \mathcal{Y}} e^{l_{v,k} + \log |N_k|}}. \quad (6)$$

Loss Function We design our loss function to combine these two components that handle heterophily and class imbalance in the proposed Im-GBK model. The learning objective of our model consists of (i) reducing the weight of the majority classes and increasing the weight of the minority classes in the training data, and (ii) improving the model’s ability to select the ideal gate. To achieve this objective, we incorporate two loss components into the loss function:

$$\mathcal{L} = \mathcal{L}_{im} + \lambda \sum_l^L \mathcal{L}_g^{(l)}. \quad (7)$$

The first component, from the class-imbalance handler, denoted as $\mathcal{L}_{im}$, applies either the *logit adjusted loss* $\mathcal{L}_{\text{logit-adjusted}}$ or the *Balanced Softmax* $\mathcal{L}_{\text{balanced-softmax}}$ approach to reduce the impact of the majority class and focus on underrepresented classes. The second component, denoted as $\mathcal{L}_g^{(l)}$, applies cross-entropy loss to each layer l of the heterophily handler to improve the model’s discriminative power and adaptively select gate to emphasize homophily or heterophily. The hyper-parameter λ balances the two losses in the overall loss function.

4.2 Fast Im-GBK

The major limitation of Im-GBK lies in efficiency, as the additional time is required to compute the gate result from the heterophily handling component. The challenge arises from the need to label edges connecting unknown nodes, which is typically addressed by learning an edge classifier. Specifically, in the message-passing process, the gate acts as an edge classifier that predicts the homophily level of a given edge. As a result, this process requires additional time proportional to the number of edges in the graph, expressed as $T_{\text{extra}} = |\mathbf{E}| \times \text{time}(e_{ij})$, where $\text{time}(e_{ij})$ represents the time to compute one edge. Therefore, we propose to use a graph-level homophily ratio [21] instead of the pair-wise edge classifier. This removes the kernel selection process before training and significantly reduces training time. We use Eq. 2 to aggregate and update the embedding layers similarly. The formal definition of the gate generator is:

$$H(\mathbf{G}) = \frac{\sum_{(v_i, v_j) \in \mathbf{E}} \mathbb{I}(y_i = y_j)}{|\mathbf{E}|} \quad (8)$$

$$\alpha_{ij} = \begin{cases} \max((1 - \epsilon)\mathbb{I}(y_i = y_j), \epsilon) & y_i, y_j \in V_{\text{train}} \\ H(\mathbf{G}) & \text{otherwise} \end{cases} \quad (9)$$

$$\mathcal{L}_{\text{Fast Im-GBK}} = \mathcal{L}_{\text{im}} \quad (10)$$

where the hyper-parameter ϵ will serve as the minimum similarity threshold, and $\mathcal{L}_{\text{im}}$ is aforementioned Class-Imbalance Handling loss.

5 Experiments

We address four questions to enhance our understanding of the model: **RQ1:** How do Im-GBK and fast-Im-GBK models perform in comparison to baselines in node classification for an imbalanced scenario? **RQ2:** How is the performance efficiency of fast-Im-GBK compared to other baseline models? **RQ3:** What is the role of each component in our Im-GBK model, and do they positively impact the classification performance?

5.1 Experiment Settings

Datasets. We conduct experiments on four datasets from PyTorch Geometric [3], which are commonly used in graph neural network literature, to evaluate the efficiency of our proposed models for the node classification task when classes are imbalanced. Table 1 presents a summary of the datasets. In addition, using CiteSeer and Cora datasets, we generate two extreme instances in which each minority class with random selection has only five training examples.

Experiment environment. For each dataset, we randomly select 60% of total samples for training, 20% for validation, and the remaining 20% for testing. We use Adam optimizer and set learning rate lr , $weight\ decay$ as 0.001 and $5e^{-4}$, respectively. All baselines follow the same setting except for the layer number of

Table 1: Statistics of the node classification datasets.

	Hom.Ratio	Imbalance Ratio	Nodes	Edges	Features	Classes
CITESEER [14]	0.736	2.655	3327	9104	3703	6
CORA [14]	0.810	4.544	2708	10556	1433	7
WIKI [14]	0.712	45.111	2405	17981	4973	17
COAUTHOR CS [15]	0.808	35.051	18333	163788	6805	15

128. We run each experiment on the same random seeds to ensure reproducibility. Model training is done on NVIDIA GeForce RTX 3090 (24GB) GPU with 90GB memory. The code depends on PyTorch 1.7.0 and PyG 2.0.4.

Evaluation metric. The quality of classification is assessed by average accuracy, AUC-ROC, and F1 scores. Each experiment is repeated five times to avoid randomness and compute the final value.

Baselines. We evaluate our approach with representative and state-of-the-art approaches, including three classic GNN models (GCN [7], GAT [17], GraphSage [4]) using three traditional imbalance techniques (Over-sampling, Re-weight, SMOTE [1]), original GBK-GNN [2], GraphSMOTE [19], and TAM [16] (we choose the combination of GCN+TAM+Balanced Softmax and GCN+TAM+ENS, referred to as GCN-TAM-BS and GCN-TAM-ENS, respectively).

5.2 Comparisons with baselines (RQ1)

In this section, we analyze the performance of classic graph neural networks (GNNs) and traditional imbalance learning approaches. As shown in the learning objective (Section 4.1), λ plays an important role in the tradeoff between classification error and consistency. Thus, we first explore the impact of hyperparameter λ on the performance. We set λ between 0 and 5 with an interval of 0.5 and the results are shown in Fig. 2. According to our analysis, the impact of λ on the results is insignificant if it is not 0. Therefore, in the following experiments, we always set $\lambda = 1$, considering the overall trends for both methods on all datasets. We examine the performance of different models. The results are reported in Table 2 and Table 3 on original datasets and extreme datasets, respectively. From the results, it can be observed that:

- The experiment demonstrates that Im-GBK models, which use logit adjust loss and balanced softmax (denoted as Im-GBK (LogitAdj) and Im-GBK (BLSM), respectively), achieve comparable or better results to state-of-the-art methods in most original datasets. Specifically, in Cora and CiteSeer, Im-GBK (LogitAdj) and Im-GBK (BLSM) are comparable to GraphSMOTE and GCN-TAM, while they outperform it on Wiki. Furthermore, Fast Im-GBK demonstrates superiority on Wiki and Coauthor-CS, and all proposed methods performed better than the baselines in extremely imbalanced cases.
- The models designed specifically for imbalance problems perform better in the extreme cases (Cora Extreme and CiteSeer Extreme); refer to Table 3.

Table 2: Comparison of different methods on original datasets.

	CORA			CITESEER			WIKI			COAUTHOR-CS		
	ACC	AUC	F-1	ACC	AUC	F-1	ACC	AUC	F-1	ACC	AUC	F-1
GCN	0.853	0.981	0.847	0.719	0.905	0.687	0.664	0.875	0.592	0.935	0.995	0.914
GCN+SMOTE	0.855	0.980	0.848	0.709	0.904	0.673	0.649	0.865	0.605	0.939	0.996	0.921
GCN+RE-WEIGHT	0.848	0.981	0.840	0.712	0.905	0.683	0.672	0.873	0.631	0.935	0.996	0.915
GAT	0.853	0.969	0.846	0.730	0.896	0.702	0.243	0.643	0.191	0.889	0.975	0.822
GAT+SMOTE	0.839	0.970	0.822	0.716	0.876	0.687	0.281	0.691	0.250	0.370	0.721	0.185
GAT+RE-WEIGHT	0.827	0.965	0.821	0.703	0.888	0.677	0.125	0.580	0.111	0.913	0.987	0.891
GRAPHSAGE	0.805	0.968	0.780	0.697	0.895	0.676	0.691	0.877	0.578	0.905	0.988	0.829
GRAPHSAGE+SMOTE	0.798	0.971	0.776	0.724	0.902	0.702	0.693	0.884	0.688	0.633	0.947	0.397
GRAPHSAGE+RE-WEIGHT	0.794	0.966	0.769	0.703	0.888	0.677	0.668	0.884	0.563	0.898	0.986	0.816
GRAPHSMOTE	0.872	0.984	0.864	0.769	0.929	0.741	0.569	0.859	0.440	0.940	0.996	0.926
GCN-TAM	0.878	0.931	0.868	0.752	0.840	0.727	0.651	0.823	0.563	0.929	0.956	0.914
GBK-GNN-BS	0.876	0.974	0.866	0.730	0.915	0.707	0.681	0.881	0.611	0.936	0.997	0.918
Im-GBK (LOGITAdj)	0.866	0.979	0.853	0.728	0.912	0.699	0.674	0.887	0.622	0.932	0.996	0.912
Im-GBK (BLSM)	0.861	0.979	0.846	0.721	0.909	0.700	0.677	0.888	0.615	0.933	0.996	0.914
FAST Im-GBK	0.876	0.988	0.863	0.766	0.926	0.738	0.723	0.911	0.655	0.951	0.997	0.939

Our models show superior performance in extreme circumstances, as indicated by achieving the best performance consistently w.r.t the most of metrics, whereas GCN-TAM and GraphSMOTE, as specifically designed models for the imbalanced classification task, also exhibit their capabilities in differentiating minority classes.

- Models designed to account for graph heterophily generally outperform classic GNNs and modified classic GNNs. This also empirically validates the rationality of using heterophilic neighborhoods for imbalanced classification to some extent.

Overall, the experiment demonstrates that the proposed methods are effective in imbalanced node classification and offer better or comparable performance to state-of-the-art. In extreme cases, our approaches outperform and show a better capability in differentiating minority classes.

5.3 Comparison in Efficiency (RQ2)

Section 3 showed that the proposed Im-GBK model could be time-consuming due to its second loss function component. To address this, we replace the kernel selection process by using a graph-level homophily ratio. Table 4 presents training time comparisons, revealing that models using gate-selection mechanisms, like GBK-GNN and Im-GBK, require significantly more training time. For example, while most models could complete one epoch within 1 second, GBK-GNN and Im-GBK took over 10 seconds and around 160 seconds to train one epoch on the CS dataset. GraphSMOTE also requires more time than Fast Im-GBK because it has to generate several synthetic nodes for each minority class. However, the results demonstrate that our proposed Fast Im-GBK model shows a significant reduction in training time compared to GBK-GNN and GraphSMOTE.

Table 3: Comparison of different methods on extreme datasets.

	Cora Extreme			CiteSeer Extreme		
	ACC	AUC	F-1	ACC	AUC	F-1
GCN	0.746	0.929	0.647	0.697	0.877	0.607
GCN+SMOTE	0.745	0.930	0.648	0.699	0.877	0.610
GCN+Re-weight	0.756	0.934	0.667	0.700	0.878	0.612
GAT	0.679	0.878	0.532	0.694	0.880	0.605
GAT+SMOTE	0.653	0.861	0.427	0.677	0.847	0.593
GAT+Re-weight	0.689	0.869	0.516	0.681	0.873	0.596
GraphSAGE	0.657	0.873	0.498	0.682	0.866	0.596
GraphSAGE+SMOTE	0.671	0.884	0.494	0.680	0.871	0.593
GraphSAGE+Re-weight	0.674	0.878	0.509	0.687	0.873	0.598
GraphSMOTE	0.770	0.928	0.674	0.703	0.893	0.612
GCN-TAM-BS	0.829	0.888	0.797	0.718	0.801	0.649
GCN-TAM-ENS	0.790	0.884	0.769	0.680	0.797	0.659
GBK-GNN	0.692	0.905	0.521	0.696	0.892	0.607
Im-GBK (LogitAdj)	0.717	0.914	0.599	0.703	0.896	0.615
Im-GBK (BLSM)	0.800	0.931	0.761	0.705	0.897	0.655
Fast Im-GBK	0.727	0.941	0.570	0.737	0.899	0.641

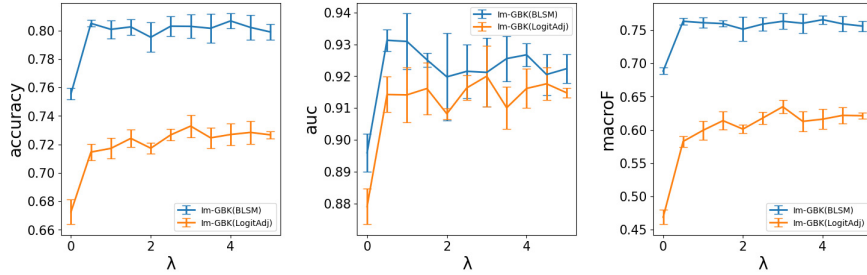


Fig. 2: Experimental results on Cora (extreme)

5.4 Ablation Analysis (RQ3)

Subsection 5.3 showed that the proposed method exhibits clear advantages in performance compared to other baselines in differentiating minority classes for extremely imbalanced graphs. To further investigate the fundamental factors underlying the performance improvements of our proposed method in Im-GBK, we conduct ablation analyses using one of the extreme cases, CiteSeer Extreme. We show the effectiveness of the model handling class imbalance classification by ablating the model Class-Imbalance Handler and Heterophily Handler, respectively. In Table 5, ‘Class-Imbalance Handling Loss’ represents two Class-Imbalance Handling losses introduced in Section 4.1. ‘Heterophily handling’ refers to the method introduced in Section 4.1 to capture graph heterophily, and ‘ $\times$ ’ means this part is ablated. Considering all strategies, it can be observed from Table 5

Table 5: Ablation Experiment Results

Class-Imbalance Handling Loss		Heterophily Handling	ACC	AUC	F1
Logit adjusted loss	Balanced Softmax				
×	×	×	0.697	0.877	0.607
×	×	✓	0.696	0.892	0.607
×	✓	×	0.710	0.871	0.618
×	✓	✓	0.705	0.897	0.655
✓	×	×	0.667	0.872	0.566
✓	×	✓	0.703	0.896	0.615

that either dropping ‘Class-Imbalance Handling Loss’ or ‘Heterophily handling’ components will result in a decrease in performance.

6 Conclusion

In this paper, we studied the problem of imbalanced classification on graphs from the perspective of graph heterophily. We observed that if a model cannot handle heterophilic neighborhoods in graphs, its ability to address imbalanced classification will be impaired. To address the graph imbalance problem effectively, we proposed a novel framework, Im-GBK, and its faster version, Im-GBK, that simultaneously tackles heterophily and class imbalance. Our framework overcomes the limitations of previous techniques by achieving higher efficiency while maintaining comparable performance. Extensive experiments are conducted on various real-world datasets, demonstrating that our model outperforms most baselines. Furthermore, a comprehensive parameter analysis is performed to validate the efficacy of our approach. In future research, we aim to explore alternative methods for modeling graph heterophily and extend our approach to real-world applications, such as fraud and spammer detection.

Table 4: Average Execution Time (s) per epoch on CS.

	Time
GCN	0.0166
GAT	0.1419
GraphSage	0.0135
GraphSMOTE	5.309
Fast Im-GBK	0.5897
GBK-GNN	12.320
Im-GBK(LogitAdj)	11.594
Im-GBK(BLSM)	11.271

References

1. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research* **16**, 321–357 (2002)
2. Du, L., Shi, X., Fu, Q., Ma, X., Liu, H., Han, S., Zhang, D.: Gbk-gnn: Gated bi-kernel graph neural networks for modeling both homophily and heterophily. In: *Proceedings of the ACM Web Conference 2022*, pp. 1550–1558 (2022)

3. Fey, M., Lenssen, J.E.: Fast graph representation learning with pytorch geometric. arXiv preprint arXiv:1903.02428 (2019)
4. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs (2018)
5. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. *Intelligent data analysis* **6**(5), 429–449 (2002)
6. Johnson, J.M., Khoshgoftaar, T.M.: Survey on deep learning with class imbalance. *Journal of Big Data* **6**(1), 1–54 (2019)
7. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks (2017)
8. Liu, Y., Ao, X., Qin, Z., Chi, J., Feng, J., Yang, H., He, Q.: Pick and choose: a gnn-based imbalanced learning approach for fraud detection. In: *Proceedings of the Web Conference 2021*, pp. 3168–3177 (2021)
9. Liu, Y., Zheng, Y., Zhang, D., Chen, H., Peng, H., Pan, S.: Towards unsupervised deep graph structure learning. In: *Proceedings of the ACM Web Conference 2022*, pp. 1392–1403 (2022)
10. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: Homophily in social networks. *Annual review of sociology* **27**(1), 415–444 (2001)
11. Menon, A.K., Jayasumana, S., Rawat, A.S., Jain, H., Veit, A., Kumar, S.: Long-tail learning via logit adjustment (2021)
12. Park, J., Song, J., Yang, E.: Graphens: Neighbor-aware ego network synthesis for class-imbalanced node classification. In: *International Conference on Learning Representations* (2021)
13. Ren, J., Yu, C., Sheng, S., Ma, X., Zhao, H., Yi, S., Li, H.: Balanced meta-softmax for long-tailed visual recognition (2020)
14. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T.: Collective classification in network data. *AI magazine* **29**(3), 93–93 (2008)
15. Shchur, O., Mumme, M., Bojchevski, A., Günnemann, S.: Pitfalls of graph neural network evaluation. arXiv preprint arXiv:1811.05868 (2018)
16. Song, J., Park, J., Yang, E.: Tam: Topology-aware margin loss for class-imbalanced node classification. In: *International Conference on Machine Learning*, pp. 20,369–20,383. PMLR (2022)
17. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks (2018)
18. Wu, Y., Lian, D., Xu, Y., Wu, L., Chen, E.: Graph convolutional networks with markov random field reasoning for social spammer detection. In: *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 1054–1061 (2020)
19. Zhao, T., Zhang, X., Wang, S.: Graphsmote: Imbalanced node classification on graphs with graph neural networks. In: *Proceedings of the 14th ACM international conference on web search and data mining*, pp. 833–841 (2021)
20. Zheng, X., Liu, Y., Pan, S., Zhang, M., Jin, D., Yu, P.S.: Graph neural networks for graphs with heterophily: A survey. arXiv preprint arXiv:2202.07082 (2022)
21. Zhu, J., Rossi, R.A., Rao, A., Mai, T., Lipka, N., Ahmed, N.K., Koutra, D.: Graph neural networks with heterophily. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 11,168–11,176 (2021)
22. Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., Koutra, D.: Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems* **33**, 7793–7804 (2020)