Tsinghua Science and Technology
Doi: xxxx(mandatory)

Research Article

Graph Representation Learning with a Multi-View Hybrid Model for Class-Imbalanced Node Classification

Longqing Du†, Zhekai Mao†, Zihang Wang, Yuanhua Zhou (✉), Mingxia Bi, Nguyen Khac An, Bui Quang Vinh, Guangquan Lu

Abstract Prevalent techniques for class-imbalanced graph learning predominantly rely on synthetic data generation to augment training samples for Graph Neural Networks (GNNs). Despite their popularity, these generative paradigms often incur substantial computational costs, introduce artifacts that distort intrinsic node relationships, and may destabilize model convergence. To address these limitations, we propose GraphMVHM (Graph Representation Learning with a Multi-View Hybrid Model), a versatile and efficient framework for node classification on imbalanced graph data. Specifically, the model constructs three complementary views: an original view, a generative view that supplements minority class samples, and a semantic view that captures node-topology correlations. Through cross-view collaborative learning, the framework mitigates the dominance of majority classes over minority classes, yielding a more discriminative node representation set compared to prior methods. The semantic view is constructed by integrating node features with topological attributes to capture fine-grained semantic information at the node level, enabling a deeper understanding of each node's intrinsic properties in relation to its neighborhood structure. Extensive experiments conducted on eight datasets—including three balanced, three imbalanced, and two large-scale graph benchmarks—demonstrate that the proposed method achieves state-of-the-art performance. Notably, on the CiteSeer dataset, accuracy improvements of 7% are achieved with both GCN and GAT backbones; on the Photo and Computers datasets, accuracy gains of 6% and 8% are attained with GAT, respectively.

Keywords node classification; class imbalance; graph neural network; data augmentation

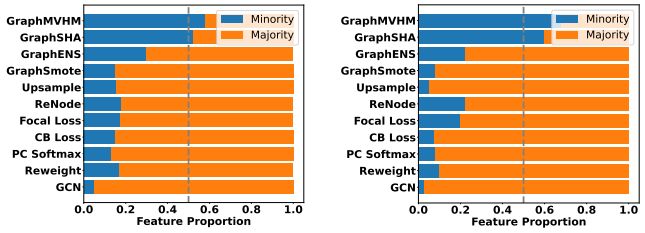
1 Introduction

Node classification constitutes a cornerstone capability within graph neural network frameworks, underpinning critical applications spanning from financial anomaly detection and agricultural monitoring systems to personalized recommendation engines [1–3].

In recent years, as real-world data collection has improved, there has been a growing trend of encountering unbalanced data distributions, often referred to as long-tail distributions[4–7]. This indicates that traditional node classification methods, designed for processing balanced data, do not adequately address real-world data needs. Therefore, research on handling imbalanced data has become a primary focus[8–10].

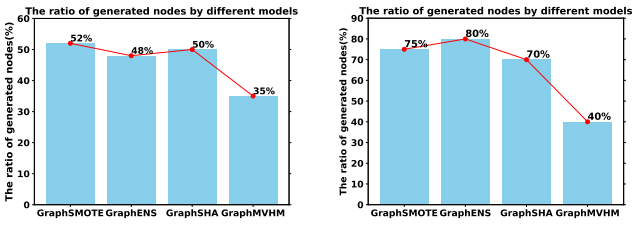
Many studies have employed GCN [11–13] for node classification research with the widespread adoption of Graph Convolutional Networks (GCN). However, as most data exhibits a long-tail distribution, the learning outcomes of GCN often favor categories with a larger number of nodes, thus resulting in biased classification results[14–16]. To address this issue, existing methods can be categorized into two groups based on whether they generate new nodes: non-generative models and generative models[17, 18].

• Longqing Du, Zhekai Mao, Zihang Wang, Yuanhua Zhou, Guangquan Lu are with Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, Guilin 541004, China, and also with Guangxi Key Lab of Multi-Source Information Mining and Security, Guangxi Normal University, Guilin 541004, China, and also with Guangxi Academy of Artificial Intelligence, Nanning 540000, China. E-mail: dulongqing@stu.gxnu.edu.cn; 18136093383@stu.gxnu.edu.cn; 927113914@qq.com; zhouyuanhua@mailbox.gxnu.edu.cn; lugq@mailbox.gxnu.edu.cn.
• Mingxia Bi, Qingdao West Coast New Area Sino-German Applied Technology School, Shandong 266500, China. E-mail: bi_mingxia@163.com
• Nguyen Khac An, Bui Quang Vinh, Hanoi National University of Education, Hanoi 100000, Vietnam. E-mail: annk@hnue.edu.vn; vinhbq@hnue.edu.vn
* To whom correspondence should be addressed.
† These authors contributed equally to this work.
Manuscript received: 2026-12-10; revised: 2026-02-04; accepted: 2026-03-06



(a) Cora-LT **(b)** Computers-ST

Fig. 1 Most models exhibit majority-class feature dominance due to limited diversity and high instance counts, resulting in significant underrepresentation of minority-class features. GraphMVHM mitigates this class imbalance more effectively than GraphSHA.



(a) Cora-LT **(b)** Computers-ST

Fig. 2 GraphMVHM demonstrates superior efficiency and graph quality over prior models, achieving higher quality with fewer nodes than GraphSHA. This indicates enhanced effectiveness in the aggregation process.

In non-generative model selection, feature enhancements and topological modifications are applied to the original data to minimize the aggregation impact of classes with a large number of nodes during message transmission, while increasing attention to classes with a small number of nodes. However, this method often fails to address the inherent data problem of minority class nodes when dealing with extreme imbalances. In such cases, where the number of minority class nodes is few, reducing the aggregation impact of majority class nodes does not alleviate the fundamental gap in magnitude. As shown in Fig 1, our experimental results on the Cora and Computers datasets.

Generation-based models typically utilize labeled nodes to generate new instances for the minority class, aiming to mitigate the dominance of the majority class[19]. Although this method can more directly and effectively address the fundamental issue of insufficient minority class data, it often introduces noise into the newly generated instances, disrupts the topological structure of the original data, alters semantic information, and incurs a certain degree of computational overhead[20]. In Fig 2, we analyzed this situation on the Cora and Computers datasets.

Therefore, we propose a semi-supervised multi-view hybrid model based on the GNN models. This model addresses the challenges of data scarcity in non-generative methods and the high computational overhead of generative models by generating an appropriate number of high-quality minority-class nodes. Additionally, it mitigates the issue of generative models altering the semantic information of the original data at the node level by combining the node attributes and topological information.

Our framework constructs multiple graph perspectives through specialized transformation operators. The architectural backbone employs parameter-shared GNN modules to extract complementary representations from distinct views, enabling cross-view knowledge distillation while maintaining model efficiency: the original aggregated view and the generative aggregated view, which are obtained from the original graph and the generated graph, respectively. This approach not only counterbalances the dominance of majority class features over minority class features but also ensures that the model remains aligned with the distribution inherent to the original graph data. Then, the attribute and topology information of the enhanced original node are combined to generate a semantic view, allowing the learning of unstructured information while preserving the representative feature at the node level. Finally, connections between the three views are established through shared parameter classifiers and contrastive learning, aiming to achieve a comprehensive and balanced decision-making result.

This article contributions can be summarized as follows:

- We further investigate the limitations of existing models in terms of computational efficiency and the quality of generated nodes, highlighting the challenges faced by current approaches in handling class imbalance in graph data.
- It was found that having more generated nodes does not necessarily enhance the alleviation of the feature compression problem. The message-passing mechanism based on the GNN model leads to feature aggregation that is not only related to node features but also to the topological structure.
- This article proposes a semi-supervised multi-view hybrid model based on GNNs. The model not only addresses the challenge of severe data scarcity in minority classes, which non-generative models fail to alleviate but also considers the noise and semantic damage caused by generating inferior nodes in generative models. This novel strategy strikes a balance between traditional non-generative models and generative models.
- This article assesses the performance of the proposed model and strategy using three GNN backbones as the core architecture. The approach was validated across

eight real-world datasets and compared to ten benchmark models, demonstrating substantial improvements over the current state-of-the-art in generative modeling.

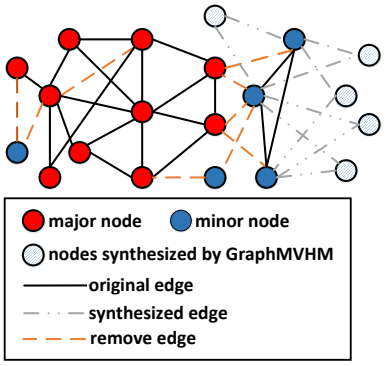


Fig. 3 Unlike previous methods, our approach (GraphMVHM) mitigates class imbalance by: (1) weighting adjacent nodes to synthesize new nodes; (2) logically removing edges between majority and minority classes to reduce negative aggregation effects; and (3) eliminating unreliable intra-class edges to minimize redundant information aggregation.

2 Related Work

2.1 Class Imbalance Problem

The class imbalance problem refers to the situation where the number of samples in different categories in the dataset varies greatly, with one or more categories having significantly fewer samples than others[21–26]. This issue has significant consequences in various domains. For instance, in medical diagnosis, there are usually far more samples from healthy individuals than from those who are sick, leading to a bias toward misclassifying sick individuals as healthy[27–29]. Similarly, in financial fraud detection, the number of fraudulent transaction samples is typically much lower than that of normal transactions, potentially causing the model to overlook a large number of fraudulent activities and resulting in significant economic losses[30–32]. The class imbalance problem also affects fields such as industrial quality control[33]. For example, in industrial production, there are usually many more samples of high-quality products than defective ones[34–36]. This class imbalance may lead to the model failing to identify defective products, thereby reducing production efficiency and product quality. Currently, researchers are developing various methods to address the class imbalance problem, including traditional methods and generative methods based on graph neural networks. These methods are expected to enhance model performance on imbalanced datasets, leading to more accurate and reliable classification and prediction in various domains.

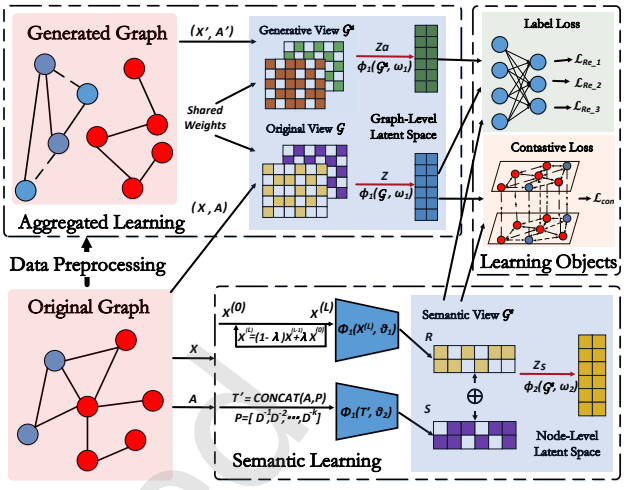


Fig. 4 The GraphMVHM model generates a synthetic graph from minority nodes, creates a semantic view by enhancing the original graph, and learns via shared parameters between original and generative views. It extracts semantic information and computes contrast loss by comparing original and semantic views. The final loss is a weighted average of contrast and classification label losses.

2.2 Previous Method

Traditional methods for addressing class imbalance in node classification include reweighting, parameter adjustments, loss function modifications, and resampling[37–40]. PC Softmax [41] modifies the probability distribution using temperature and category parameters. CB Loss [42] and Focal Loss [43] introduce category balance and scaling factors to focus on difficult-to-classify samples. Resampling methods like ReNode and Upsample [44]address topological imbalances by duplicating minority class samples. However, these approaches struggle to capture complex relationships and topological information in node classification tasks.

Graph Neural Networks have shown promise in addressing unbalanced node classification by effectively capturing local and global graph structure information[45**?** –47]. Despite their success, GNNs have limited efficacy in handling generated unbalanced node classification problems. Recent techniques for generated unbalanced node classification utilize generative models to create minority class nodes, enhancing classification via discriminator training For instance, GraphSHA enhances the decision boundary for minority classes by synthesizing challenging minority class samples. Nevertheless, these generative approaches face challenges such as low generation quality and unstable model training.

3 Methodologies

3.1 Data Preprocessing

This research focuses on the imbalanced node classification problem in undirected and unweighted graphs. We begin by

defining the initial graph structure as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{v_1, ..., v_N\}$ represents the set of nodes and $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$ denotes the set of edges. The graph contains $|\mathcal{V}| = N$ nodes and $|\mathcal{E}| = M$ edges. We represent the graph structure using an adjacency matrix $A \in \{0,1\}^{N\times N}$ and capture node characteristics through a feature matrix $X \in \mathbb{R}^{N\times d}$.

To alleviate the domination of majority class node features over minority class features during aggregation and mitigate the semantic confusion caused by the intermingling of majority class node features from different classes, we first perform logical neighbor edge removal.

$$Q^{(2\times M)} = \begin{cases} (v_i, v_j) & \text{if } \exists j \in \mathcal{N}_{(i)} \text{ s.t. } \cos(v_i, v_j) < \delta \\ & \quad \vee Y_j^* \neq Y_i^* \\ \emptyset & \text{otherwise} \end{cases} \tag{1}$$

while Q denotes the edge set of the modified graph, the labeled training label set $Y^* \subseteq$ the labeled label set Y, and the $\cos(v_i, v_j)$ represents the Features cosine similarity between v_i and v_j, and $v_{j\in\mathcal{N}(i)}$ represents the neighbor node of v_i.

Furthermore, empirical observations frequently indicate that the attributes of real-world datasets not only display a degree of uniformity but also predominantly align with a Gaussian distribution. Through grid search pre-experiments, the probability parameter δ is typically assumed to be 0.6.

To further alleviate the learning bias caused by the scarcity of minority class nodes, we adopt logical neighbor generation to generate edges and nodes. In order to enhance the quality of newly generated nodes, we select minority class nodes and their neighbors of the same class with high similarity to generate new nodes and edges. The number of synthetic nodes generated for each minority class is set to the difference between the average number of labeled nodes per class across the entire dataset and the current number of labeled nodes in that minority class, ensuring balanced class distribution. The generation process is defined as follows:

$$\begin{aligned} &X'_{(i,j)} = \tfrac{X_i + X_j \cdot \cos(v_i, v_j)}{1+\cos(v_i, v_j)}, \forall j \in \mathcal{N}_{(i)}, \\ &X' = \text{CONCAT}(X, \{X'_{(i,j)}\}_{j\in\mathcal{N}_{(i)}}, \text{dim} = 0), \\ &Y' = \text{CONCAT}(Y^*, \{y'_{(i,j)}\}_{j\in\mathcal{N}_{(i)}}, \text{dim} = 0), y'_{(i,j)} = Y_i^* \\ &Q = \text{CONCAT}(Q, (v_i, \{v'_{(i,j)}\}_{j\in\mathcal{N}_{(i)}}), \text{dim} = 1), \end{aligned} \tag{2}$$

where $v'_{i,j}$ is generated by v_i and v_j. Finally, original view $\mathcal{G} = (X, A)$ and generative view $\mathcal{G}^a = (X', A')$ where $A' = Sparse(Q)$, and $Sparse(Q)$ is to get the sparse matrix of Q. The training nodes in the generative view are denoted by Y'. The overall effect is shown in Fig 3.

To capture semantic information from distant nodes during aggregation, we enhance the topological information by performing topological coding based on the random walk diffusion process before constructing the semantic view:

$$\begin{aligned} T' &= \text{CONCAT}(A, P, \text{dim} = 1), \\ P &= [D_i^{-1}, D_i^{-2}, D_i^{-3}, \ldots, D_i^{k_t}], \end{aligned} \tag{3}$$

while D_i represents the degree of v_i and $k_t (k_t \geqslant 1,\ k_t \in \mathbb{N})$ represents the number of hops from v_i. Thus, P represents a probability random walk of order 1 to kt on v_i. Pre-experimental evaluations demonstrate that a three-hop ($k_t = 3$) distance yields optimal performance in most scenarios, aligning with the observed phenomenon of semantic information attenuation in node relationships.

In order to strengthen the focus on node-specific characteristics, while also learning the influence of neighboring features, we adopt a feature enhancement strategy:

$$X^{(L)} = (1-\lambda)AX^{(L-1)} + \lambda X^{(0)}, \tag{4}$$

where $L (L \geqslant 1,\ L \in \mathbb{N})$ is the number of feature enhancements, $\lambda (1 \geqslant \lambda \geqslant 0)$ is the weight and $X^{(0)}$ means origin feature X. In order to retain the distinctive characteristics of nodes while integrating the features of their neighbors, and acknowledging that nodes at a greater distance tend to influence semantic information more than feature information. Pre-experimental studies systematically established the configuration $\lambda = 0.8$, $L = 2$ as Pareto-optimal parameters. This pre-optimization protocol effectively decouples architectural hyperparameter tuning from downstream experimental validation phases. Finally, we generate the semantic view $\mathcal{G}^s = (X^{(L)}, T')$, which is used to learn the semantic information and unstructured information between nodes.

3.2 Multi-view Learning

In order to analyze the overall information of view nodes, we utilize the message-passing mechanism of the GNN model for aggregation learning. Furthermore, to ensure that the model learns from the original view and addresses the scarcity of minority class data, we allow the model to share learning parameters ω_1 between the generative view and the original view:

$$\begin{aligned} \Psi_1(\mathcal{G}; \omega_1) : Z &= \sigma(\text{MPGNN}(X, A)), \\ \Psi_1(\mathcal{G}^a; \omega_1) : Z_a &= \sigma(\text{MPGNN}(X', A')), \end{aligned} \tag{5}$$

where Z is the embedding of the origin view $\mathcal{G}$, σ is the activation function ,and Z_a is the embedding of the generative view $\mathcal{G}^a$.

To study the semantic information between individual nodes, we first map the nodes' attributes and topological information to the same dimension. Subsequently, we concatenate them into a single dimension and learn the semantic relationships between nodes using a multi-layer perceptron:

Table 1 Training time (seconds) comparison across different models, datasets, and backbones.

Model	Cora-LT			CiteSeer-LT			Photo-ST			Computers-ST		
	GCN	GAT	SAGE	GCN	GAT	SAGE	GCN	GAT	SAGE	GCN	GAT	SAGE
GraphSMOTE	42	50	39	46	52	44	57	62	53	62	67	59
GraphENS	41	49	40	47	51	45	56	61	54	61	66	60
GraphSHA	40	48	38	45	50	43	55	60	50	60	65	55
GraphMVHM (ours)	38	45	35	42	48	40	50	58	52	55	62	58

$$
\begin{aligned}
&\Phi_1(X^{(L)},\theta_1): R=\sigma(\mathrm{MLP}(X^{(L)})),\\
&\Phi_1(T',\theta_2): S=\sigma(\mathrm{MLP}(T')),\\
&\Psi_2(\mathcal{G}^s;\omega_2): Z_s=\sigma(\mathrm{MLP}(\mathrm{CONCAT}(R\oplus S)),
\end{aligned}
\tag{6}
$$

where Z_s = (R, S) is the embedding of the semantic view $\mathcal{G}^s$, and $\oplus$ represents the concatenation process at $\dim = 1$.

To establish the connection between the three views, we obtain the label loss of the three views by sharing the classifier parameters θ_3. This means that we use the same set of parameters for the classifier in all three views, allowing us to compute the label loss for each view using the shared parameters. By doing this, we ensure that the classifier learns to extract meaningful features from all three views, leading to a more robust and accurate model:

$$
\begin{aligned}
&f_n(\hat{y})=\textstyle\sum_{i=1}^{\mathcal{V}^*}\mathbb{I}(Y_i^*=\hat{y}),\\
&\hat{F}_{\mathrm{MSE}}(\hat{y},y)=\frac{1}{N}\sum_{i=0}^{N-1}\frac{1}{f_n(\hat{y})}(\hat{y}_i-y_i)^2,\\
&\Phi 2(Z,\theta_3):\mathcal{L}_{re1}=\hat{F}_{\mathrm{MSE}}(\mathrm{Max}(\sigma(\mathrm{MLP}(Z),\mathrm{dim}{=}0),Y^*),\\
&\Phi 2(Z_a,\theta_3):\mathcal{L}_{re2}=\hat{F}_{\mathrm{MSE}}(\mathrm{Max}(\sigma(\mathrm{MLP}(Z_a),\mathrm{dim}{=}0),Y'),\\
&\Phi_2(Z_s,\theta_3):\mathcal{L}_{re3}=\hat{F}_{\mathrm{MSE}}(\mathrm{Max}(\sigma(\mathrm{MLP}(Z_s),\mathrm{dim}{=}0),Y^*),\\
&\mathcal{L}_{re}=cl_1\mathcal{L}_{re1}+cl_2\mathcal{L}_{re2}+(1-cl_1-cl_2)\mathcal{L}_{re3},
\end{aligned}
\tag{7}
$$

Among them, cl_1 and cl_2 are both manually initialized hyperparameters within the range $[0, 1.0]$, typically satisfying cl_1+$cl_2 \leqslant 1$. In the hyperparameter analysis section of experiments, we conducted a comprehensive 3D visualization analysis to examine the impact of label losses from three different views. Through this analysis, we identified the most balanced parameter configuration that optimizes the model performance

In addition, in order to better learn the semantic information between nodes, we use Z and Z_s for comparative learning [49] and perform a weighted sum of label loss $\mathcal{L}_{re}$ and contrastive loss $\mathcal{L}_{con}$:

$$
\begin{aligned}
\mathcal{L}_{\mathrm{con}}=-\frac{1}{2|V|}\sum_{v\in V}\Bigg[&\log\frac{\phi(Z_v,Z_{s_v})}{\sum_{u\neq v}\phi(Z_v,Z_{s_u})}\\
&+\log\frac{\phi(Z_{s_v},Z_v)}{\sum_{u\neq v}\phi(Z_{s_v},Z_u)}\Bigg],\\
J(\Theta)=(1-\alpha)\cdot\mathcal{L}_{\mathrm{re}}&+\alpha\cdot\mathcal{L}_{\mathrm{con}}
\end{aligned}
\tag{8}
$$

where $\phi(a,b)=\exp(\mathrm{sim}(a,b)/\tau)$ denotes the similarity kernel, α is a weighted hyper-parameter for semi-supervised contrastive loss to balance learning, in the experiments section, we analyze the impact of the change α on the result of the model. The overall model process is shown in the Fig 4.

3.3 Complexity Analysis

The proposed GraphMVHM framework introduces only modest computational overhead beyond base GNN models, with complexity dominated by two components: (1) Feature generation for synthesized nodes incurs $\mathcal{O}(\theta^2 N_{\text{train}}^2 d)$ time, where θ denotes the majority class proportion, N_{train} the training set size, and d the feature dimension; (2) Edge construction and pruning operations require $\mathcal{O}(N_{\text{train}}^2|\mathcal{E}|)$ time for synthesizing connections and filtering unreliable edges. Crucially, since $\mathcal{O}(\theta^2 N_{\text{train}}^2) \ll \mathcal{O}(N_{\text{train}}^2) \ll \mathcal{O}(N)^2$ in semi-supervised node classification paradigms, the total overhead $\mathcal{O}(\theta^2 N_{\text{train}}^2 d + N_{\text{train}}^2|\mathcal{E}|)$ remains lightweight compared to full-graph computations. Notably, dependence on θN_{train} rather than N yields substantial efficiency gains on large graphs, where $\theta^2 N_{\text{train}}^2 d$ significantly outperforms N-scaled alternatives in memory and time costs.

To empirically validate the efficiency of GraphMVHM, we compare the training time with three representative baseline methods across four datasets using three GNN backbones. The results are presented in Table 1. Experiments were conducted on a single NVIDIA RTX 3090 GPU with 24GB memory. As shown, GraphMVHM demonstrates competitive computational efficiency in most settings, requiring less time than the strong generative baseline GraphSHA on most configurations. While GraphENS and GraphSMOTE generally show comparable efficiency to each other (with slight variations), they typically require more time than GraphMVHM and GraphSHA in most cases.

4 Experimental

In this section, we conduct extensive experiments to evaluate the effectiveness of GraphMVHM for class-imbalanced node classification by addressing the following research questions:

RQ1(**Performance**): How does GraphMVHM perform compared to baseline methods?

Table 2 Statistics of datasets used in the paper.

Dataset	#Nodes	#Edges	#Features	#Classes
Cora	2,708	5,429	1,433	7
CiteSeer	3,327	4,732	3,703	6
PubMed	19,717	44,338	500	3
Photo	7,650	238,162	745	8
Computers	13,752	491,722	767	10
CS	18,333	163,788	6,805	15
obgn-arxiv	169,343	1,166,243	128	40
CoraFull	19,793	65,311	8,710	70

RQ2(Robustness): How robust is GraphMVHM in handling different levels of class imbalance?

RQ3(Impaction): What impact do different hyperparameters and view settings have on the performance of GraphMVHM?

RQ4(Explanatory): How to explain the robustness of GraphMVHM?

4.1 Experimental Setup

This configuration allows us to evaluate model performance under different degrees of class imbalance across distinct network types, and the degree of imbalance is quantified by the ratio $\rho = \frac{max_i|Y_i|}{min_j|Y_j|}$ in Table 2. Our experiments explore long-tail class imbalance settings on the Cora, CiteSeer, and PubMed datasets with a 6:2:2 split for training, validation, and test sets, and an imbalance ratio ρ set to 100 [50]. For the step-tail class imbalance setting on the Photo, Computers, and CS datasets, we use a 1:1:8 split for training, validation, and test sets, and an imbalance ratio ρ set to 20. Here, half of the classes are designated as major classes, sharing an equal number of training samples (n_{major}), while the other half are minor classes, sharing the same number of training samples ($n_{min} = n_{major}/\rho$). To evaluate the performance of GraphMVHM in handling large imbalanced graph data, we conduct tests on the CoraFull with 70 classes [51] and obgn-arxiv with 40 classes [52].

We compare GraphMVHM with various methods for handling class imbalance. For loss modification, we examine (1) Reweighting, which adjusts class weights proportionally to the number of samples; (2) PC Softmax and (3) Class Balanced Loss (CB Loss), two general approaches to modifying loss functions for imbalanced data; (4) Focal Loss, which prioritizes hard samples in the loss function; and (5) ReNode, which addresses topological imbalance in graph data alongside class imbalance. In terms of generation methods, we consider (1) Upsample, which duplicates minority nodes; (2) GraphSMOTE, an oversampling technique for imbalanced graph data; and (3) GraphENS, an ensemble learning method

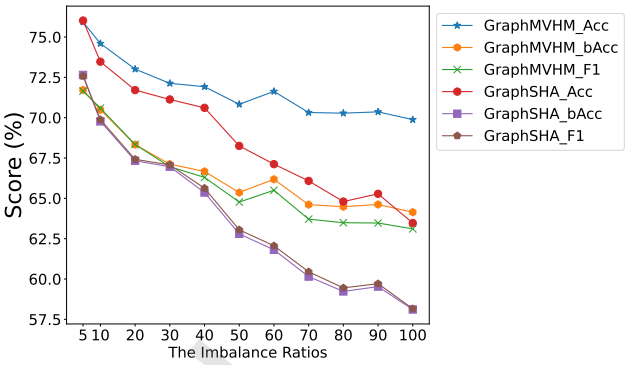


Fig. 5 Imbalance Ratios Experiments for CiteSeer-LT

tailored for graph-structured data classification. Additionally, we evaluate GraphSHA, a confidence-based method for generating nodes and edges, which currently demonstrates superior performance as a generative model[38, 53–56].

4.2 Results on Imbalanced Datasets

From Table 3 and 5, we can observe that GraphMVHM exhibits significant improvements compared to almost all other competitors utilizing different GNN backbones. Particularly, on the CiteSeer dataset, the accuracy score achieved using GCN and GAT as base models improved by 7% compared to the state-of-the-art generative technique (GraphSHA). Additionally, on the Photo and Computer datasets, the accuracy score using GAT as the base model improved by 6% and 8%, respectively. Furthermore, the consistency in performance across different methods utilizing various GNN backbones suggests that the inherent characteristics of models primarily influence the observed performance differences. And our F1 score has always been higher than all baseline models, which proves the stability of the model. Although GraphSAGE as the backbone generally performs weaker than GCN and GAT, GraphMVHM often achieving optimal or near-optimal performance with only a small gap from the best results. We also analyze classification accuracy for each class on the Computers dataset in LT and ST split in Table 4. We can see that the labels with the most nodes did not show the best performance which is related to the topological distribution of the nodes as we found, and our model can still achieve good results on classes with only 1% labeled nodes.

And, as demonstrated in Table 6, our GraphMVHM framework exhibits robust performance across three fundamental GNN architectures (GCN, GAT and GraphSAGE) on the Cora-LT dataset. The framework achieves exceptional performance for rare classes while maintaining strong majority class recognition, particularly evident in Label6 (0.6% samples) where GCN backbone attains 88.2% accuracy. Notably, our

Table 3 Node classification results in the long-tailed class-imbalanced setting with three backbones for 10 runs. The best is highlighted in boldface, and the runner-up is highlighted in underline.

		Cora-LT			CiteSeer-LT			PubMed-LT		
	$\rho = 100$	Acc.	bAcc.	F1	Acc.	bAcc .	F1	Acc.	bAcc .	F1
GCN	Vanilla	70.85±0.45	58.95±0.72	58.87±0.91	51.22±0.49	44.32±0.39	37.52±0.68	51.34±0.61	43.95±0.49	34.87±0.74
	Reweight	77.64±0.15	72.55±0.18	73.22±0.18	63.32±0.28	56.62±0.26	54.98±0.18	76.75±0.17	72.15±0.20	71.68±0.16
	PC Softmax	76.52±0.17	71.88±0.33	71.49±0.26	61.98±0.49	58.84±0.28	57.94±0.33	74.12±0.68	72.26±0.38	71.44±0.18
	CB Loss	77.19±0.22	72.59±0.28	72.92±0.20	61.31±0.39	55.01±0.59	53.22±0.66	76.37±0.23	71.75±0.19	72.68±0.23
	Focal Loss	77.64±0.25	72.95±0.26	73.63±0.24	59.50±0.44	53.21±0.35	51.56±0.40	75.43±0.20	71.12±0.24	71.61±0.23
	ReNode	78.15±0.27	73.01±0.24	74.29±0.28	62.20±0.36	55.50±0.30	53.89±0.25	75.75±0.16	69.76±0.18	70.98±0.16
	Upsample	74.75±0.14	68.52±0.18	68.10±0.18	54.92±0.22	48.30±0.15	45.07±0.14	71.40±0.10	63.95±0.14	64.95±0.11
	GraphSMOTE	74.62±0.44	68.80±0.52	69.20±0.52	56.02±0.90	50.19±0.29	47.79±0.39	74.42±0.13	68.26±0.13	70.98±0.13
	GraphENS	75.32±0.24	71.00±0.38	70.70±0.49	63.02±0.38	56.85±0.39	55.49±0.43	76.99±0.18	71.11±0.24	72.78±0.26
	ImGCL	76.21±0.36	71.90±0.34	70.99±0.44	62.51±0.34	57.89±0.37	55.79±0.33	77.45±0.23	71.49±0.28	73.36±0.26
	GraphSHA	79.11±0.36	73.76±0.60	75.05±0.39	63.73±0.29	58.35±0.30	58.25±0.24	78.75±0.17	74.23±0.19	75.03±0.33
	GraphMVHM	**80.18±0.41**	**75.78±0.35**	**76.17±0.31**	**70.53±0.18**	**64.67±0.25**	**63.39±0.11**	**80.30±0.22**	**79.23±0.25**	**79.53±0.26**
GAT	Vanilla	64.26±0.65	54.10±0.80	55.22±0.79	49.04±0.28	42.32±0.20	35.55±0.27	47.55±1.68	38.89±1.33	29.89±2.12
	Reweight	77.47±0.34	71.90±0.58	73.22±0.18	61.70±0.63	54.22±0.62	53.52±0.69	74.00±0.44	69.86±0.77	69.48±0.77
	PC Softmax	68.49±0.84	63.89±0.94	64.00±0.77	56.51±1.69	56.10±1.33	55.09±1.52	76.53±0.36	73.21±0.17	73.13±0.34
	CB Loss	77.10±0.28	71.90±0.72	72.52±0.40	61.21±0.51	54.93±0.56	53.43±0.58	74.57±0.29	69.40±0.65	70.25±0.64
	Focal Loss	77.64±0.15	72.26±0.24	72.99±0.28	59.50±0.44	53.21±0.35	51.56±0.40	73.98±0.31	70.19±0.34	70.52±0.26
	ReNode	77.90±0.27	71.61±0.34	73.30±0.38	60.60±0.35	53.75±0.37	51.80±0.43	73.90±0.33	69.05±0.46	69.39±0.39
	Upsample	72.37±0.30	62.21±0.38	64.90±0.34	53.22±0.25	46.70±0.25	42.91±0.44	64.42±1.02	57.12±0.61	54.91±1.02
	GraphSMOTE	74.45±0.31	67.50±0.41	68.90±0.44	57.22±0.28	51.12±0.34	49.20±0.49	73.93±0.43	68.90±0.39	70.60±0.45
	GraphENS	76.89±0.27	71.89±0.39	71.90±0.48	61.79±0.36	57.65±0.36	54.20±0.40	76.40±0.14	70.23±0.24	71.10±0.24
	ImGCL	77.21±0.35	72.47±0.37	71.50±0.39	62.04±0.31	58.41±0.33	56.39±0.31	76.95±0.29	70.79±0.30	71.57±0.30
	GraphSHA	78.31±0.26	72.94±0.37	74.10±0.36	64.20±0.39	58.46±0.40	58.14±0.44	78.26±0.22	72.93±0.24	73.89±0.21
	GraphMVHM	**80.04±0.31**	**75.35±0.35**	**75.60±0.30**	**70.81±0.43**	**61.34±0.36**	**60.05±0.43**	**81.18±0.40**	**81.50±0.46**	**81.14±0.45**
SAGE	Vanilla	73.00±0.14	61.57±0.15	63.05±0.17	47.74±0.25	41.82±0.48	36.90±0.37	58.60±0.14	47.89±0.13	42.21±0.11
	Reweight	76.53±0.19	68.50±0.36	70.09±0.42	57.15±0.58	50.82±0.52	49.12±0.62	65.90±0.58	59.79±1.39	58.69±1.19
	PC Softmax	76.81±0.27	72.95±0.33	73.27±0.33	58.20±0.29	55.90±0.24	56.49±0.23	71.32±0.16	73.61±0.17	70.11±0.14
	CB Loss	76.90±0.34	69.92±0.39	71.12±0.35	57.44±0.41	50.99±0.36	48.53±0.42	67.57±0.39	60.49±0.49	61.29±0.53
	Focal Loss	76.99±0.20	69.62±0.33	70.59±0.29	57.00±0.78	50.65±0.80	48.30±0.91	70.39±0.40	65.44±0.44	66.02±0.46
	ReNode	77.05±0.19	69.02±0.27	70.93±0.28	57.60±0.55	51.12±0.54	48.90±0.48	67.35±0.56	60.45±0.83	60.56±0.82
	Upsample	73.67±0.17	63.21±0.24	65.59±0.19	50.12±0.15	44.10±0.15	41.30±0.19	63.90±0.09	54.45±0.10	53.11±0.09
	GraphSMOTE	74.05±0.24	65.95±0.44	67.70±0.39	52.72±0.74	46.92±0.75	44.09±0.88	64.93±0.43	56.60±0.53	56.70±0.58
	GraphENS	76.49±0.24	69.87±0.27	70.20±0.35	62.45±0.36	55.95±0.39	53.95±0.41	77.40±0.17	72.23±0.25	72.99±0.20
	ImGCL	76.87±0.35	70.25±0.37	70.80±0.36	62.79±0.33	56.82±0.37	54.30±0.36	77.80±0.20	72.61±0.27	73.47±0.23
	GraphSHA	77.96±0.36	72.57±0.41	73.57±0.41	62.71±0.41	58.39±0.39	58.26±0.35	78.11±0.24	74.85±0.25	75.51±0.27
	GraphMVHM	**79.00±0.33**	**74.55±0.88**	**75.20±0.39**	**65.41±0.49**	**59.83±0.42**	**59.30±0.37**	**78.94±0.31**	**77.27±0.21**	**76.65±0.27**

Table 4 Node classification result of different labels on Computer-LT and Computer-ST datasets with GCN by GraphMVHM.

Computers-LT	Acc.	bAcc.	F1	C0 3%	C1 6%	C2 10%	C3 4%	C4 38%	C5 2%	C6 4%	C7 6%	C8 16%	C9 12%
GCN	83.88	82.95	87.66	92.0	82.3	85.2	82.6	88.4	82.8	92.2	90.9	86.9	93.9
Computers-ST	Acc.	bAcc.	F1	C0 19%	C1 19%	C2 19%	C3 19%	C4 19%	C5 1%	C6 1%	C7 1%	C8 1%	C9 1%
GCN	84.78	86.46	83.78	89.7	83.0	85.3	85.5	84.9	84.8	85.2	84.2	84.9	85.1

Table 5 Node classification results in the step-tailed class-imbalanced setting with three backbones for 10 runs. The best is highlighted in boldface, and the runner-up is highlighted in underline.

		Photo-ST			Computer-ST			CS-ST		
	$\rho = 20$	Acc.	bAcc .	F1	Acc.	bAcc .	F1	Acc.	bAcc .	F1
GCN	Vanilla	70.85±0.45	59.10±0.70	59.02±0.89	55.62±1.34	40.72±0.79	26.57±0.66	36.15±1.25	53.19±0.79	31.21±1.43
	Reweight	77.64±0.14	72.56±0.18	73.22±0.18	77.45±0.28	84.09±0.23	72.99±0.26	90.65±0.14	89.75±0.10	81.62±0.20
	PC Softmax	76.53±0.17	71.89±0.33	71.50±0.26	72.00±0.89	59.91±1.69	53.90±2.35	86.15±0.56	86.27±0.38	73.15±0.70
	CB Loss	77.20±0.22	72.60±0.28	72.93±0.20	81.11±0.25	85.49±0.08	75.63±0.15	89.90±0.09	89.65±0.09	72.68±0.23
	Focal Loss	77.64±0.25	72.96±0.26	73.64±0.24	80.30±0.23	85.57±0.35	74.57±0.20	89.75±0.14	89.03±0.14	78.52±0.90
	ReNode	78.15±0.27	73.01±0.24	74.30±0.28	72.51±0.97	77.50±0.91	65.80±1.16	90.80±0.26	89.56±0.18	81.45±1.07
	Upsample	85.25±0.14	68.52±0.18	68.10±0.18	78.90±0.23	83.88±0.15	73.52±0.13	84.90±0.16	85.50±0.12	74.65±0.18
	GraphSMOTE	83.89±0.24	86.32±0.23	81.56±0.25	75.59±0.18	83.18±0.19	68.18±0.29	85.80±0.23	85.25±0.24	68.92±0.77
	GraphENS	86.90±0.13	86.52±0.09	84.44±0.13	78.39±0.18	85.48±0.18	73.20±0.13	91.82±0.13	91.59±0.24	75.90±0.45
	ImGCL	86.55±0.23	87.00±0.29	84.20±0.23	78.40±0.24	85.33±0.23	73.55±0.21	90.22±0.22	90.56±0.24	77.01±0.41
	GraphSHA	87.23±0.30	**88.72±0.19**	84.97±0.39	80.81±0.33	86.10±0.27	75.31±0.55	**91.93±0.21**	**91.66±0.24**	77.11±0.42
	GraphMVHM	**89.11±0.23**	87.89±0.14	**87.81±0.20**	**84.78±0.32**	**86.46±0.27**	**83.78±0.32**	90.79±0.24	88.28±0.18	**85.95±0.33**
GAT	Vanilla	36.66±0.39	44.60±0.43	28.62±0.48	57.64±0.75	42.42±0.40	26.45±0.24	34.31±0.65	49.89±0.75	24.59±1.11
	Reweight	79.57±1.25	81.90±0.64	75.63±1.16	72.50±0.52	75.62±0.44	63.62±0.48	88.20±0.43	87.06±0.46	71.48±0.66
	PC Softmax	51.35±3.21	61.15±1.85	51.10±2.58	31.43±2.88	51.70±2.72	37.59±2.41	78.65±0.67	77.51±0.65	65.33±0.69
	CB Loss	81.60±0.89	85.20±0.61	79.20±1.06	79.41±0.77	84.92±0.36	74.01±0.68	89.55±0.30	89.37±0.19	74.98±0.83
	Focal Loss	81.84±0.65	84.76±0.64	79.05±0.85	79.31±0.49	84.85±0.35	73.99±0.59	88.48±0.27	87.81±0.24	72.92±0.76
	ReNode	76.30±1.23	80.65±1.18	72.56±0.93	71.60±0.65	75.05±0.37	63.81±0.83	87.60±0.33	85.29±0.36	69.55±0.49
	Upsample	77.76±1.15	79.81±0.44	72.70±0.64	72.25±0.45	76.70±0.39	63.91±0.34	82.12±0.20	82.42±0.10	65.41±0.24
	GraphSMOTE	80.45±0.61	81.20±0.41	75.75±0.39	79.22±0.28	84.44±0.22	70.03±0.41	83.20±0.23	82.40±0.19	66.71±0.35
	GraphENS	83.99±0.38	86.23±0.21	79.90±0.32	77.66±0.26	84.60±0.21	71.91±0.50	89.65±0.37	89.32±0.24	79.55±0.44
	ImGCL	83.00±0.34	85.12±0.23	79.01±0.35	77.80±0.27	84.69±0.26	72.91±0.30	88.70±0.37	89.15±0.29	77.42±0.39
	GraphSHA	83.71±0.97	86.24±0.88	80.59±0.81	77.20±0.29	82.90±0.24	79.73±0.59	**91.47±0.22**	**91.22±0.24**	76.35±0.21
	GraphMVHM	**89.41±0.37**	**88.69±0.33**	**88.50±0.45**	**85.30±0.27**	**87.10±0.23**	**84.08±0.22**	89.43±0.21	87.07±0.22	**86.48±0.23**
SAGE	Vanilla	59.00±2.32	59.67±1.76	46.89±2.90	63.16±0.05	49.84±0.08	29.90±0.13	73.60±0.34	78.49±0.23	63.37±0.55
	Reweight	84.33±0.27	86.86±0.26	82.67±0.22	82.85±0.37	87.54±0.14	75.40±0.47	89.83±0.37	88.65±0.36	74.18±0.36
	PC Softmax	85.94±0.15	86.51±0.17	83.27±0.13	80.60±0.19	80.09±0.85	70.05±0.63	89.35±0.25	89.99±0.37	76.68±0.54
	CB Loss	82.50±0.33	85.37±0.29	80.12±0.39	83.03±0.21	86.96±0.14	74.83±0.27	89.64±0.19	89.49±0.14	77.69±0.87
	Focal Loss	82.06±0.40	85.00±0.33	78.97±0.39	81.85±0.27	86.95±0.09	74.30±0.20	88.89±0.20	89.54±0.20	78.22±0.28
	ReNode	84.51±0.17	86.02±0.22	81.63±0.26	79.55±0.55	86.91±0.23	74.34±0.28	89.75±0.34	89.91±0.44	79.96±0.45
	Upsample	81.97±0.34	84.54±0.14	79.03±0.29	82.22±0.25	86.60±0.13	74.87±0.39	86.91±0.19	87.73±0.13	75.05±0.27
	GraphSMOTE	79.99±0.29	84.45±0.34	78.85±0.41	82.90±0.33	87.72±0.24	73.80±0.37	85.98±0.13	85.40±0.13	67.88±0.18
	GraphENS	86.89±0.14	89.42±0.16	86.00±0.15	82.50±0.40	88.10±0.12	74.55±0.54	91.55±0.17	92.05±0.24	76.91±0.24
	ImGCL	87.25±0.16	89.15±0.18	86.35±0.19	82.75±0.42	88.05±0.15	74.70±0.55	91.80±0.20	91.95±0.26	77.55±0.27
	GraphSHA	**88.58±0.16**	**90.25±0.14**	**87.14±0.23**	**83.57±0.51**	**88.98±0.14**	74.91±0.42	92.65±0.19	**92.52±0.19**	78.32±0.28
	GraphMVHM	87.35±0.19	87.01±0.14	86.36±0.34	83.30±0.49	86.84±0.45	**82.60±0.34**	**93.05±0.39**	91.01±0.36	**89.05±0.42**

Table 6 Node classIfication result of different labels on Cora-LT dataset with GCN, GAT, and GraphSAGE by GraphMVHM.

Cora-LT	Label0 5.3%			Label1 1.1%			Label2 11.6%			Label3 54.0%			Label4 25.0%			Label5 2.4%			Label6 0.6%		
	Acc.	bAcc.	F1	Acc.	bAcc .	F1	Acc.	bAcc .	F1	Acc.	bAcc.	F1	Acc.	bAcc.	F1	Acc.	bAcc.	F1	Acc.	bAcc.	F1
GCN	82.0	81.8	90.0	79.6	58.4	76.1	83.3	77.8	88.2	85.8	83.1	90.3	84.0	79.5	87.6	73.6	71.4	83.1	88.2	77.3	87.0
GAT	83.6	85.2	97.7	76.1	63.7	79.1	79.5	72.4	83.9	84.6	82.2	90.1	81.5	77.1	86.2	81.9	77.4	86.7	82.4	72.0	82.9
SAGE	85.2	85.5	91.9	73.9	61.9	78.3	84.6	78.6	88.2	84.6	82.7	89.5	84.0	80.6	88.3	80.6	76.2	85.8	85.3	88.3	93.5

Table 7 Classification results on obgn-arxiv and CoraFull datasets with GCN backone. OOM indicates Out-Of-Memory on a 24GB GPU.

	obgn-arxiv		CoraFull	
Method	Acc	bAcc	Acc	bAcc
ImGCL	45.97±0.55	45.64±0.82	63.63±0.73	62.29±0.71
GraphSMOTE	OOM	OOM	58.34±0.62	57.47±0.65
GraphENS	OOM	OOM	62.67±0.88	62.78±0.86
GraphSHA	47.03±0.45	45.82±0.87	63.84±0.60	63.03±0.58
GraphMVHM	**49.84±0.27**	**49.72±0.85**	**65.67±0.56**	**64.81±0.52**

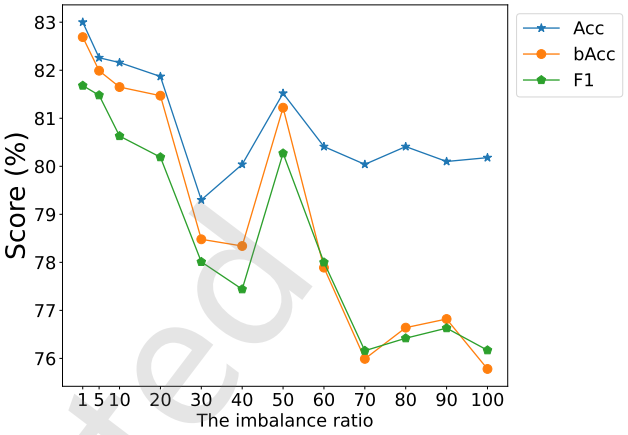
method sustains balanced performance across class distributions, with GCN achieving 85.8% accuracy for the majority Label3 (54.0% samples) while simultaneously delivering 79.6% accuracy for the extremely imbalanced Label1 (1.1% samples).

This once again confirms our findings: with many nodes, the learning performance of the model may not be good, which is related to the topological distribution of the nodes.

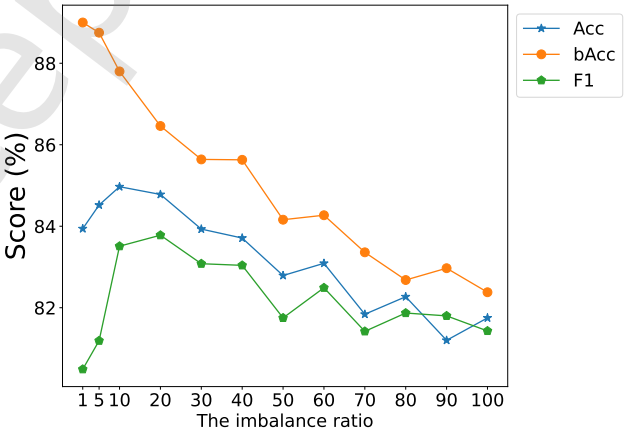
To further validate the scalability and practical utility of GraphMVHM in real-world scenarios, we evaluate its performance on large-scale, naturally imbalanced graphs. As shown in Table 7, GraphMVHM demonstrates strong scalability and robustness on both obgn-arxiv and CoraFull datasets. It achieves the best accuracy (Acc) and balanced accuracy (bAcc) on both datasets, while avoiding the out-of-memory (OOM) failures encountered by GraphSMOTE and GraphENS on the obgn-arxiv dataset. Although GraphSHA shows improved performance compared to other baselines, it still falls short of GraphMVHM in all metrics. These results underscore that GraphMVHM not only excels on curated benchmark splits but also maintains superior performance and efficiency on large-scale graphs with organic, long-tailed class distributions, highlighting its practicality for real-world applications.

4.3 Influence of Imbalance Ratio

To assess the robustness and generalization capability of GraphMVHM, we conducted extensive experiments on three datasets with different imbalance ratios. As depicted in Fig 5, we measured key performance metrics for both GraphMVHM and GraphSHA as the imbalance ratio increased from 5 to 100. Notably, GraphMVHM's accuracy starts at around 76% and remains above 70% even as the imbalance ratio reaches 100, with only a marginal decline in performance. In contrast, GraphSHA's performance deteriorates significantly with increasing imbalance, as accuracy falls from approximately 75.5% at lower imbalance ratios to below 63% when the imbalance ratio reaches 100. This stark difference highlights the robustness of GraphMVHM, which maintains stable accuracy, balanced accuracy, and F1 scores, even under challenging imbalanced conditions.



(a) Cora-LT



(b) Computers-ST

Fig. 6 Different Imbalance ratios for GraphMVHM on GCN branch model in Cora-LT and Computers-ST Datasets

To further validate the generalizability of GraphMVHM, we evaluated its performance on the Computer and Cora datasets with the same range of imbalance ratios, shown in Fig. 6(a) and 6(b), respectively. GraphMVHM once again demonstrated stability across these datasets; for instance, on the Cora and Computer datasets, accuracy declined by only 3% and 6%, respectively, as the imbalance ratio increased from 1 to 100. Across all metrics, the fluctuations in performance for GraphMVHM remained within a 10% range on both datasets, similar to its performance on CiteSeer. These results demonstrate that GraphMVHM not only exhibits robustness across varying imbalance ratios but also generalizes effectively across datasets with diverse structural characteristics and sensitivity to imbalance.

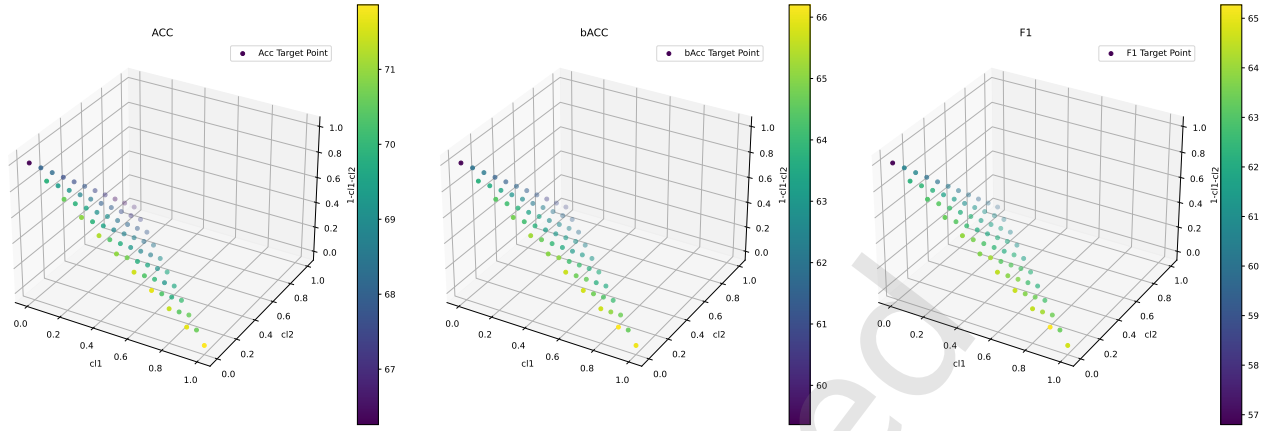
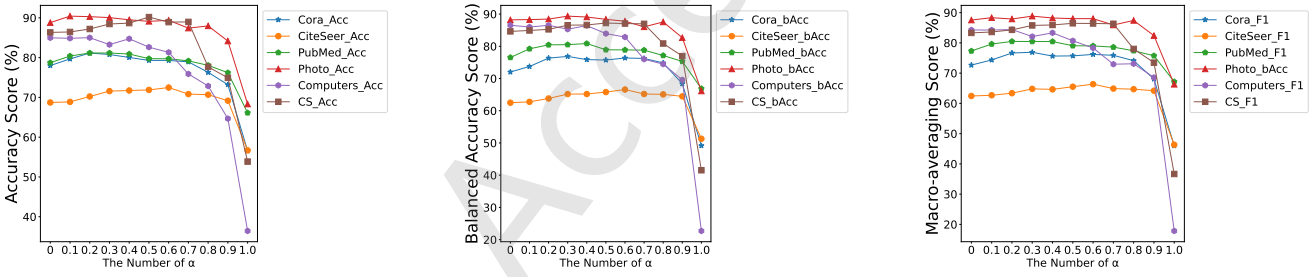


Fig. 7 Hyperparameter cl_1, cl_2 Experiments for CiteSeer-LT



(a) (b) (c)

Fig. 8 Analysis of GraphMVHM datasets with parameter α on GCN backbone

Table 8 Grid search results of δ on six datasets with GCN backbone. Best results are in **bold**.

	Cora-LT			CiteSeer-LT			PubMed-LT			Photo-ST			Computers-ST			CS-ST		
δ	Acc	bAcc	F1	Acc	bAcc	F1	Acc	bAcc	F1	Acc	bAcc	F1	Acc	bAcc	F1	Acc	bAcc	F1
0.1	77.3	72.1	72.5	65.2	59.8	59.3	75.8	70.5	70.1	85.1	80.3	80.7	80.4	76.2	75.8	86.5	81.8	81.4
0.2	78.1	73.0	73.3	66.8	61.2	60.8	76.9	71.6	71.2	86.3	81.5	81.9	81.7	77.5	77.1	87.2	82.5	82.1
0.3	79.0	74.1	74.5	68.5	62.8	62.3	78.2	72.9	72.5	87.1	82.3	82.7	82.5	78.3	77.9	88.0	83.3	82.9
0.4	79.5	74.8	75.2	69.3	62.6	63.1	79.0	73.8	73.4	87.8	83.0	83.4	83.3	79.1	78.7	88.7	84.0	83.6
0.5	79.9	75.2	75.6	70.0	64.2	62.8	79.8	74.6	74.3	88.5	83.7	84.1	84.0	79.8	79.4	89.3	84.6	84.2
0.6	**80.2**	**75.8**	**76.2**	**70.5**	**64.7**	**63.4**	**80.3**	**79.2**	**79.6**	**89.1**	**87.9**	**87.8**	**84.8**	**86.5**	**83.8**	**90.8**	**88.3**	**85.9**
0.7	80.0	75.5	75.9	70.3	64.5	63.1	80.1	75.0	74.7	88.9	84.1	84.5	84.6	80.4	80.0	90.5	84.8	84.4
0.8	79.6	75.1	75.5	69.8	64.0	63.6	79.5	74.4	74.1	88.3	83.5	83.9	83.9	79.7	79.3	89.9	84.2	83.8
0.9	78.9	74.4	74.8	68.9	63.1	62.7	78.8	73.7	73.4	87.6	82.8	83.2	83.1	78.9	78.5	89.1	83.4	83.0

4.4 Ablation Study

We perform systematic component ablation to quantify the individual and synergistic contributions of our architectural elements. Table 9 presents a granular decomposition analysis, revealing that the full model configuration—integrating all three perspective encoders with the contrastive alignment objective—achieves optimal performance, substantially exceeding variants with partial component subsets. This empirically validates the complementary nature of our multi-view learning paradigm. Notably, our combinatorial analysis demonstrates that arbitrary view concatenations fail to consistently improve performance over specialized single-view configurations, underscoring the importance of our carefully designed view interaction mechanisms rather than naive ensemble strategies.

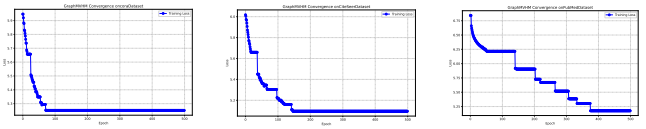
Table 9 Ablation study of GraphMVHM on CiteSeer-LT with GCN.

Method	Acc (%)	bAcc (%)	F1 (%)
Original View $\mathcal{G}$	56.80±0.26	50.87±0.29	49.99±0.29
Generative View $\mathcal{G}^a$	66.74±0.27	61.09±0.27	60.05±0.23
Semantic View $\mathcal{G}^s$	20.70±0.20	17.22±0.27	9.16±0.20
$\mathcal{G}$+$\mathcal{G}^a$	66.22±0.27	60.82±0.26	60.00±0.29
$\mathcal{G}$+$\mathcal{G}^s$	34.15±0.31	28.90±0.29	23.45±0.33
$\mathcal{G}^a$+$\mathcal{G}^s$	58.33±0.35	53.12±0.31	51.67±0.30
$\mathcal{G}$+$\mathcal{G}^a$+$\mathcal{G}^s$	65.44±0.23	59.81±0.27	58.85±0.24
$\mathcal{G}^a$+$\mathcal{L}_{con}$	67.15±0.24	61.88±0.28	60.92±0.25
$\mathcal{G}$+$\mathcal{G}^a$+$\mathcal{L}_{con}$	68.40±0.21	62.75±0.23	61.63±0.19
Three Views + $\mathcal{L}_{con}$	**70.53±0.18**	**64.67±0.25**	**63.39±0.11**

We conduct extensive hyperparameter sensitivity analysis across key architectural parameters, including view-specific loss coefficients (cl_1, cl_2) and the contrastive learning intensity (α). As illustrated in Fig. 7, performance metrics exhibit bounded fluctuations (5-8% relative variance) across the parameter space, confirming the operational robustness of our framework and reducing deployment tuning overhead.

Further evaluation in six datasets (Fig. 8) shows the robustness of the model in a range of α values, with consistent scores for Acc, bAcc, and F1. Performance remained strong until α exceeded 0.7, where Computers and CS exhibited significant decreases, while Cora, CiteSeer, PubMed and Photo maintained stability. Although a severe performance degradation occurred as the model transitioned to unsupervised learning at α = 1.0, GraphMVHM maintained robust performance on the Photo and PubMed datasets under these unsupervised conditions.

Finally, To ensure the robustness of the edge removal process, we performed a comprehensive grid search over the cosine similarity threshold $\delta \in 0.1, 0.2, \ldots, 0.9$ across all six



(a) Cora-LT **(b)** CiteSeer-LT **(c)** PubMed-LT

Fig. 9 Convergence analysis for GraphMVHM on GCN

datasets using the GCN backbone. The complete results (Acc, bAcc, F1) are detailed in Table 8. The analysis confirms that $\delta = 0.6$ delivers optimal or near-optimal performance consistently, establishing it as a robust and generally applicable parameter choice.

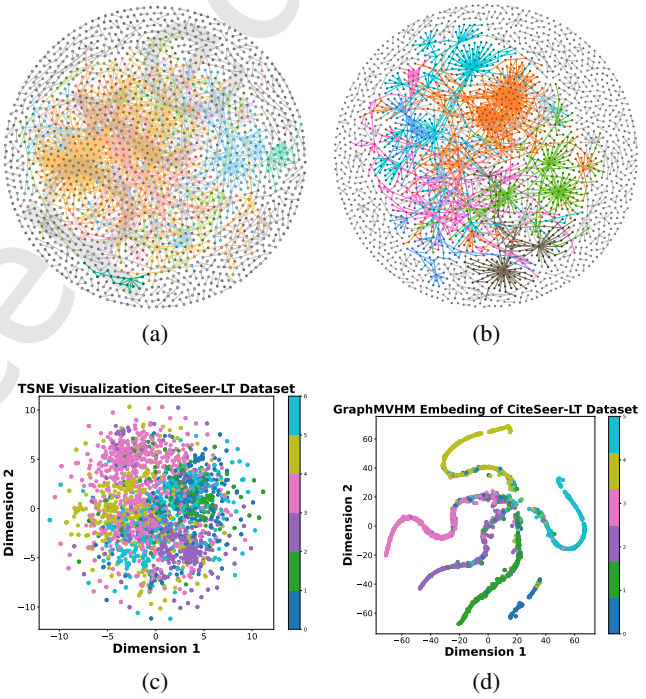


Fig. 10 Robustness analysis for GraphMVHM on GCN

4.5 Robustness Analysis

To comprehensively evaluate the robustness of GraphMVHM, we conduct analyses from the perspectives of optimization stability and representation quality. As illustrated in Fig. 9, our model exhibits a consistent and monotonic reduction in training loss, reaching a stable plateau within 400 epochs across all evaluated datasets. This reliable convergence behavior underscores the optimization stability and generalizability of the proposed multi-view learning framework.

We further investigate the quality of the learned node representations by analyzing inter-class separability and intra-class compactness. A community structure analysis on the CiteSeer-LT dataset (containing seven classes) provides a qualitative assessment. Fig. 10(a) visualizes the node em-

beddings without our topological enhancement, where nodes form only three discernible communities with significant inter-class entanglement. In contrast, as shown in Fig. 10(b), applying GraphMVHM leads to a clear separation of all seven ground-truth classes with high intra-community label consistency, demonstrating the method's efficacy in refining graph topology to improve class discrimination.

This qualitative observation is quantitatively supported by analyzing the feature distribution. A comparison of the node embedding spaces before training (Fig. 10(c)) and after training with GraphMVHM (Fig. 10(d)) reveals a marked increase in both the distance between different class clusters (inter-class separability) and the density of nodes within the same class (intra-class compactness).

In summary, the consistent optimization convergence, the enhanced separation of community structures, and the improved feature space geometry collectively demonstrate the robustness of the GraphMVHM framework. These results validate its capability to learn stable and discriminative representations across diverse imbalanced graph datasets.

5 Conclusions

This work presents GraphMVHM, an innovative multi-perspective learning framework that addresses fundamental limitations in class-imbalanced graph representation learning for homogeneous graphs. By synergizing topology-aware data augmentation with contrastive multi-view alignment, our approach effectively mitigates majority class dominance while preserving semantic fidelity—addressing critical gaps in conventional generative and reweighting strategies. Comprehensive evaluations across six benchmark datasets and two large-scale datasets demonstrate that GraphMVHM consistently surpasses ten contemporary baselines under three representative GNN architectures, establishing new state-of-the-art performance in imbalanced node classification. Beyond accuracy gains, our framework exhibits exceptional robustness across varying imbalance ratios and graph typologies, highlighting its practical applicability in real-world scenarios. Future work will extend GraphMVHM to heterogeneous graphs to address more complex real-world network structures.

6 Acknowledgements

The work was supported partly by Guangxi Natural Science Foundation under Grant (Nos. 2025GXNSFFA069015), National Natural Science Foundation of China (Nos.62372119 and 62166003), and in part by the Guangxi Academy of Artificial Intelligence, and in part by the Guanex Bagui Young Talent Training Program, Innovation Project of Guangxi Graduate Education (Nos.JGY2024080 and JGY2022046), the Guangxi Collaborative Innovation Center of Multi-Source Information Integration, China. Innovation Project of Guangxi Graduate Education (YCSW2025168) .Finally, I would also like to express my deepest gratitude to my fiancée, Liu Fuhang, for her unwavering support and encouragement throughout my research.

Reference

[1] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, A comprehensive survey on graph neural networks, *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[2] G. Lu, L. Wang, and J. Li, Aspect sentiment analysis with heterogeneous graph neural networks, *Inf. Process. Manage.*, vol. 59, no. 4, p. 102953, 2022.

[3] X. Wang, L. Peng, R. Hu, P. Hu, and X. Zhu, Unsupervised multiplex graph representation learning via maximizing coding rate reduction, *Pattern Recognit.*, vol. 165, p. 111557, 2025.

[4] Z. Wu, P. Zhou, G. Wen, and X. Zhu, A pseudo-labeling approach based on knowledge distillation for graph few-shot learning, *Inf. Process. Manage.*, vol. 62, no. 6, p. 104268, 2025.

[5] S. Qi, B. Lin, X. Hu, C. Zhang, L. Zheng, L. Qian, and Y. Wu, Throughput optimization for multi-UAV-assisted offshore internet of things: A hypergraph approach, *Tsinghua Sci. Technol.*, vol. 30, no. 6, pp. 2452–2466, 2025.

[6] M. Sun, X. Wu, Y. Zhou, J.-K. Hao, and Z.-H. Fu, Efficient backbone network construction in wireless artificial intelligent computing systems, *Tsinghua Sci. Technol.*, vol. 30, no. 5, pp. 2300–2319, 2025.

[7] B. Liu, Y. Qian, and J. Wang, Rodent arena multi-view monitor (RAMM): A camera synchronized photographic control system for multi-view rodent monitoring, *Tsinghua Sci. Technol.*, vol. 30, no. 5, pp. 2195–2214, 2025.

[8] W. L. Hamilton, R. Ying, and J. Leskovec, Inductive representation learning on large graphs, in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, pp. 1024–1034, 2017.

[9] K. Guo, Y. Hu, Z. Qian, H. Liu, and Z. Wang, Hierarchical graph convolution network for traffic forecasting, in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 35, no. 4, pp. 3353–3361, 2021.

[10] Y. Li, S. Zhang, G. Zhang, and D. Cheng, Community-centric graph unlearning, in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 39, no. 17, pp. 18548–18556, 2025.

[11] B. Jiang, Z. Zhang, D. Lin, J. Tang, and B. Luo, Semi-supervised learning with graph learning-convolutional networks, in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 11313–11320, 2019.

[12] X. Wang, J. Lv, M. O. Alassafi, F. E. Alsaadi, B. D. Parameshachari, L. Zou, G. Feng, and Z. Liu, Deep bi-directional adaptive gating graph convolutional networks for spatio-temporal

traffic forecasting, *Tsinghua Sci. Technol.*, vol. 30, no. 5, pp. 2060–2080, 2025.

[13] D. Wang, L. Chen, F. Gong, and Q. Zhu, Maximizing depth of graph-structured convolutional neural networks with efficient pathway usage for remote sensing, *Tsinghua Sci. Technol.*, vol. 30, no. 5, pp. 1940–1953, 2025.

[14] L. He, D. Cheng, G. Zhang, and S. Zhang, Leveraging long-range nodes in multi-view graph contrastive learning, *Inf. Fusion*, vol. 122, p. 103186, 2025.

[15] J. Lu, C. Wu, L. Wang, R. Li, and X. Shen, Dual-modality integration attention with graph-based feature extraction for visual question and answering, *Tsinghua Sci. Technol.*, vol. 30, no. 5, pp. 2133–2145, 2025.

[16] L. Zhou, Q. Mao, and M. Dong, Objective class-based micro-expression recognition through simultaneous action unit detection and feature aggregation, *Tsinghua Sci. Technol.*, vol. 30, no. 5, pp. 2114–2132, 2025.

[17] R. Li, Z. Liu, Y. Ma, D. Yang, and S. Sun, Internet financial fraud detection based on graph learning, *IEEE Trans. Comput. Soc. Syst.*, vol. 10, no. 3, pp. 1394–1401, 2022.

[18] H. Zhang, G. Lu, M. Zhan, and B. Zhang, Semi-supervised classification of graph convolutional networks with Laplacian rank constraints, *Neural Process. Lett.*, pp. 1–12, 2022.

[19] Y. Wang, C. Wu, Z. Shi, and L. Zhang, GraphGAN: Graph representation learning with GANs, in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 34, no. 1, pp. 6630–6637, 2020.

[20] R. Sauber-Cole and T. M. Khoshgoftaar, The use of generative adversarial networks to alleviate class imbalance in tabular data: A survey, *J. Big Data*, vol. 9, no. 1, p. 98, 2022.

[21] N. Japkowicz and S. Shaju, The class imbalance problem: A systematic study, *Intell. Data Anal.*, vol. 6, no. 5, pp. 429–449, 2002.

[22] Y. Yang and Z. Xu, Rethinking the value of labels for improving class-imbalanced learning, in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, pp. 19290–19301, 2020.

[23] J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi, and M. Asadpour, Boosting methods for multi-class imbalanced data classification: An experimental review, *J. Big Data*, vol. 7, pp. 1–47, 2020.

[24] A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, and S. Kumar, Long-tail learning via logit adjustment, in *Int. Conf. Learn. Represent. (ICLR)*, 2021.

[25] F. Kamalov, Kernel density estimation based sampling for imbalanced class distribution, *Inf. Sci.*, vol. 512, pp. 1192–1201, 2020.

[26] P. Kumar, R. Bhatnagar, K. Gaur, and A. Bhatnagar, Classification of imbalanced data: Review of methods and applications, in *Proc. IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 1099, no. 1, p. 012077, IOP Publishing, 2021.

[27] J. M. Johnson and T. M. Khoshgoftaar, Survey on deep learning with class imbalance, *J. Big Data*, vol. 6, no. 1, pp. 1–54, 2019.

[28] K. Tang, J. Huang, and H. Zhang, Long-tailed classification by keeping the good and removing the bad momentum causal effect, in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, pp. 1513–1524, 2020.

[29] V. Kumar, G. S. Lalotra, P. Sasikala, D. S. Rajput, R. Kaluri, K. Lakshmanna, M. Shorfuzzaman, A. Alsufyani, and M. Uddin, Addressing binary classification over class imbalanced clinical datasets using computationally intelligent techniques, *Healthcare*, vol. 10, no. 7, p. 1293, MDPI, 2022.

[30] J. Sun, H. Li, H. Fujita, B. Fu, and W. Ai, Class-imbalanced dynamic financial distress prediction based on adaboost-SVM ensemble combined with SMOTE and time weighting, *Inf. Fusion*, vol. 54, pp. 128–144, 2020.

[31] H. Faris, R. Abukhurma, W. Almanaseer, M. Saadeh, A. M. Mora, P. A. Castillo, and I. Aljarah, Improving financial bankruptcy prediction in a highly imbalanced class distribution using oversampling and ensemble learning: A case from the Spanish market, *Prog. Artif. Intell.*, vol. 9, pp. 31–53, 2020.

[32] C.-H. Cheng, Y.-F. Kao, and H.-P. Lin, A financial statement fraud model based on synthesized attribute selection and a dataset with missing values and imbalanced classes, *Appl. Soft Comput.*, vol. 108, p. 107487, 2021.

[33] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis, Decoupling representation and classifier for long-tailed recognition, in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.

[34] J. Lee, Y. C. Lee, and J. T. Kim, Fault detection based on one-class deep learning for manufacturing applications limited to an imbalanced database, *J. Manuf. Syst.*, vol. 57, pp. 357–366, 2020.

[35] J. Wang, T. Lukasiewicz, X. Hu, J. Cai, and Z. Xu, RSG: A simple but effective module for learning imbalanced datasets, in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 3784–3793, 2021.

[36] Y. Zhang, X.-S. Wei, B. Zhou, and J. Wu, Bag of tricks for long-tailed visual recognition with deep convolutional neural networks, in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 35, no. 4, pp. 3447–3455, 2021.

[37] C.-Y. Chuang, J. Robinson, Y.-C. Lin, A. Torralba, and S. Jegelka, Debiased contrastive learning, in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, pp. 8765–8775, 2020.

[38] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, Collective classification in network data, *AI Mag.*, vol. 29, no. 3, pp. 93–93, 2008.

[39] Y. Kalantidis, M. B. Sariyildiz, N. Pion, P. Weinzaepfel, and D. Larlus, Hard negative mixing for contrastive learning, in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, pp. 21798–21809, 2020.

[40] J. Xia, L. Wu, G. Wang, J. Chen, and S. Z. Li, ProgCL: Rethinking hard negative mining in graph contrastive learning, in *Proc. Int. Conf. Mach. Learn. (ICML)*, pp. 23959–23970, 2022.

[41] S. K. Prabhakar, R. Bharath, and K. S. Dec, Performance analysis of breast cancer classification with softmax discriminant classifier and linear discriminant analysis, in *Precision*

Medicine Powered by pHealth and Connected Health, ser. IFMBE Proc., vol. 66, Springer, 2018, pp. 171–175.

[42] Y. Cui, Y. Jia, T.-Y. Lin, Y. Song, and S. Belongie, Class-balanced loss based on effective number of samples, in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 9268–9277, 2019.

[43] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, Focal loss for dense object detection, in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pp. 2980–2988, 2017.

[44] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, Image-to-image translation with conditional adversarial networks, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 1125–1134, 2017.

[45] L. Qu, H. Zhu, R. Zheng, Y. Shi, and H. Yin, ImgAGN: Imbalanced network embedding via generative adversarial graph networks, in *Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Min. (KDD)*, pp. 1390–1398, 2021.

[46] M. Shi, Y. Tang, X. Zhu, D. Wilson, and J. Liu, Multi-class imbalanced graph convolutional network learning, in *Proc. 29th Int. Joint Conf. Artif. Intell. (IJCAI)*, pp. 2141–2147, 2020.

[47] W.-Z. Li, C.-D. Wang, H. Xiong, and J.-H. Lai, GraphSA: Synthesizing harder samples for class-imbalanced node classification, in *Proc. 29th ACM SIGKDD Conf. Knowl. Discov. Data Min. (KDD)*, pp. 1328–1340, 2023.

[48] L. Zeng, L. Li, Z. Gao, P. Zhao, and J. Li, ImgCL: Revisiting graph contrastive learning on imbalanced node classification, in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 37, no. 9, pp. 11138–11146, 2023.

[49] Q. Yu, L. Liu, P. Wang, and J. Zhang, Adversarial contrastive learning via asymmetric infonce, in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 13665, pp. 119–135, 2022.

[50] J. Chen, T. Ma, and C. Xiao, FastGCN: Fast learning with graph convolutional networks via importance sampling, in *Int. Conf. Learn. Represent. (ICLR)*, 2018.

[51] A. Bojchevski and S. Guennemann, Deep Gaussian embedding of graphs: Unsupervised inductive learning via ranking, in *Int. Conf. Learn. Represent. (ICLR)*, 2018.

[52] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, Open graph benchmark: Datasets for machine learning on graphs, in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, pp. 22118–22133, 2020.

[53] O. Shchur, M. Mumme, A. Bojchevski, and S. Guennemann, Pitfalls of graph neural network evaluation, in *Relational Represent. Learn. Workshop, NeurIPS*, 2018.

[54] D. Chen, Y. Lin, G. Zhao, X. Ren, P. Li, J. Zhou, and X. Sun, Topology-imbalance learning for semi-supervised node classification, in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, pp. 29885–29897, 2021.

[55] T. Zhao, X. Wang, and Z. Li, GraphSMOTE: Imbalanced node classification with GNNs, in *Proc. 14th ACM Int. Conf. Web Search Data Min. (WSDM)*, pp. 833–841, 2021.

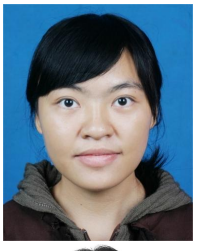
[56] J. Park, J. Song, and E. Yang, GraphENS: Neighbor-aware ego network synthesis for imbalanced node classification, in *Int. Conf. Learn. Represent. (ICLR)*, 2021.

Author biography

Longqing Du received the B.E. degree in software engineering from the Chengdu University of Information Technology, Chengdu, China, in 2023. He is currently pursuing the master's degree with the Guangxi Normal University, Guilin, China. His research interests include graph data mining. He was responsible for the model design, implementation, and paper writing.

Zhekai Mao received the B.E. degree (Bachelor of Engineering) in Data Science and Big Data Technology from Guangxi Normal University, Guilin, China, in 2025. He is currently pursuing a professional master's degree in Software Engineering at the same university. His research interests lie in multimodality and time series forecasting. He is responsible for collecting the experimental databases.

Zihang Wang received the Bachelor's Degree in Software Engineering from the Zhengzhou University of Light Industry, Zhengzhou, China, in 2025. He is currently pursuing the master's degree with the Guangxi Normal University, Guilin, China. His research interests in Graph Contrastive Learning. He is responsible for collecting experimental data.



Yuanhua Zhou is an assistant researcher at Guangxi Normal University. Her main research interests lie in bioinformatics and data mining, with a particular focus on the cultivation of talents in science and engineering and the design of related courses. She is responsible for guiding the writing of the paper.

Mingxia Bi Lecturer in the Department of Computer Science at the Qingdao West Coast New Area Sino-German Applied Technology School (Qingdao Sino-German Intelligent Manufacturing Technician College). She also holds the position of Lead Expert at the Qingdao Master Teacher Studio in Information Technology. Her research interests lie in professional talent development and curriculum design in computer science, with a specific focus on the applications of artificial intelligence and pattern recognition. She is responsible for verifying the effectiveness of the model.

Nguyen Khac An Lecturer at the Department of Computer Science, Faculty of Information Technology, Hanoi National University of Education (HNUE). His research focuses on applying optimization algorithms, machine learning, and deep learning techniques to solve complex problems in network optimization and intelligent system design. In addition, he is interested in developing and integrating artificial intelligence models into large-scale and complex computational systems to enhance their performance and reliability. His goal is to leverage AI-driven optimization and learning approaches to develop efficient, intelligent, and modern solutions that advance next-generation communication networks and smart technologies. He is responsible for guiding the designing experiments.

Bui Quang Vinh A Lecturer in Computer Science at HNUE, his research leverages optimization algorithms, machine learning, and deep learning to solve complex problems in network optimization and intelligent systems. His work focuses on integrating advanced AI into large-scale computational systems to improve performance and reliability, with the goal of developing intelligent solutions for next-generation networks and smart technologies. He also guides experimental design and provides research mentorship.

Guangquan Lu received the B.E. degree in computer science from Guangxi Normal University, Guilin, China, in 2011, and the Ph.D. degree in philosophy from Sun Yat-Sen University, China, in 2019. He is currently a professor with the Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, Guilin, China. His research interests include deep learning, data mining, graph representation learning and applications. He also guides experimental design and provides research mentorship.