Neurocomputing 654 (2025) 131280

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Adaptive imbalanced node classification graph contrastive learning

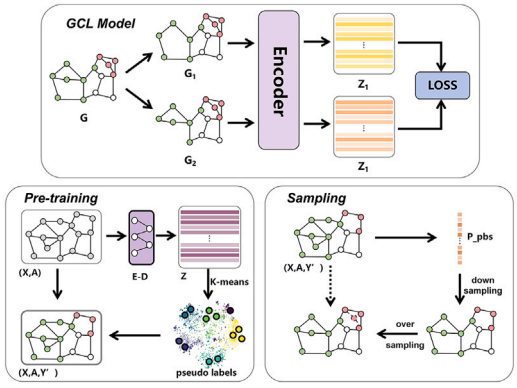
Xinyu Hu, Donghai Guan*, Weiwei Yuan, Qi Zhu, Çetin Kaya Koç

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, China

HIGHLIGHTS

- We introduce a VGAE with an encoder-decoder structure to learn graph structure and node features, improving robustness and generalization for pseudo-label generation.
- We design a graph-tailored resampling strategy. We prioritize removing low-importance nodes during undersampling and synthesizing high-importance nodes during oversampling.
- We create a multi-level, step-by-step sampling strategy. A hybrid sampling method adjusts class proportions, and an augmentation function retains minority node information.
- We develop a new data augmentation technique that prioritizes keeping minority class node information while masking majority class nodes, helping the model capture minority features.

GRAPHICAL ABSTRACT



This figure shows a Graph Contrastive Learning process, including three main modules. The GCL Model module demonstrates our graph augmentation method for retaining more minority class information and uses a graph encoder and loss function to learn graph representations. The Pre-training module uses the trained Encoder–Decoder model to generate high-quality embeddings and generate pseudo-labels through K-means clustering. The Resampling module balances the node class distribution through mixed sampling techniques, including oversampling and undersampling.

ARTICLE INFO

Communicated by M. Wu

Keywords:
Imbalanced node classification
Graph contrastive learning
Pseudo-labels

ABSTRACT

Graph Contrastive Learning (GCL) is a powerful self-supervised technique for learning node and graph representations. However, real-world graph data often exhibit imbalanced class distributions, which pose significant challenges to GCL's effectiveness. Our experiments show that current state-of-the-art (SOTA) methods perform poorly under imbalanced settings. To address this, we propose a novel GCL framework called AIGCL for imbalanced node classification. This framework automatically and adaptively balances the node representations learned by GCL. Specifically, we introduce a new data augmentation method that retains more information from minority class nodes during graph augmentation. Additionally, we use an imbalance rate adaptive sampling strategy to balance the data. We also incorporate a Variational Graph Autoencoder (VGAE) with an encoder–decoder structure to pretrain the data and generate high-quality pseudo-labels. Our experiments demonstrate that under imbalanced settings, our model improves classification accuracy by 4 %-12 % compared to baseline models, significantly enhancing the performance of minority class nodes.

* Corresponding author.
Email addresses: huxinyuabc@nuaa.edu.cn (X. Hu), dhguan@nuaa.edu.cn (D. Guan), yuanweiwei@nuaa.edu.cn (W. Yuan), zhuqi@nuaa.edu.cn (Q. Zhu), cetinkoc@ucsb.edu (Ç. Kaya Koç).

https://doi.org/10.1016/j.neucom.2025.131280
Received 24 May 2025; Received in revised form 16 July 2025; Accepted 13 August 2025
Available online 18 August 2025

1. Introduction

Graphs are a popular and powerful data structure for representing complex relational systems such as social networks, citation networks, and knowledge graphs. In these systems, nodes represent entities, while edges denote the relationships between them. Recently, Graph Contrastive Learning (GCL) has become the de facto standard for self-supervised graph learning due to its outstanding performance in unsupervised node/graph representation learning [1]. Most GCL models generate different views of nodes or graphs through augmentation functions, capture feature and structural information using encoders, and learn discriminative representations by maximizing the consistency between views with a contrastive loss function.

Graph Contrastive Learning combines the structural characteristics of graph data with the benefits of contrastive learning and has achieved notable results in a variety of practical applications, such as fraud detection and molecular property prediction [2,3]. However, real-world graph data often suffers from class imbalance, where the number of nodes in certain classes (head classes) far exceeds that in other classes (tail classes). This imbalanced class distribution challenges the effectiveness of GCL methods, as it can result in node representations biased towards the majority class while neglecting the features of the minority class, thereby degrading the quality of learned representations. Moreover, since GCL does not have access to label information during training, this important characteristic of class imbalance remains implicit and is largely ignored by most current GCL methods. To comprehensively evaluate the performance of the GCL model in imbalanced environments, we conducted numerous experiments under various imbalance categories and proportions. The results indicate that the quality of node representations significantly affects the experimental outcomes.

Current class-imbalance learning on graphs (CILG) can be divided into data-level methods and algorithm-level methods. Data-level methods generally attempt to integrate class-imbalance learning strategies with graph structural information to enhance model performance. For example, GraphSMOTE [4] is the first graph data interpolation method, inspired by the Synthetic Minority Over-sampling Technique (SMOTE) [5]. GraphSMOTE generates synthetic minority nodes by interpolating between two real minority nodes in the embedding space and pre-trains an edge predictor on real nodes and edges to determine the connectivity between synthetic and real nodes. Additionally, other methods such as GraphENS [6] and ImGAGN [7] also combine class-imbalance learning with graph structural information and offer improvements. In this research, we adopt resampling techniques from class imbalance research and combine them with graph structural information to balance node classes. Algorithm-level methods focus on enhancing the model or loss function. For network embedding methods, RSDNE [8] and ImVerde [9] modify DeepWalk [10], while ACS-GNN [11] and EGCN [12] adjust the aggregation operation in the standard GNN architecture. Cost-sensitive strategies assign greater weights to the losses of the minority class [13], incorporating graph topological information into the loss function design [14]. These model refinements are primarily based on the encoder and loss function, while our research focuses on the augmentation module unique to contrastive learning. Existing node masking functions generally mask both minority and majority class nodes with the same probability, which can lead to the loss of information from the minority class. To address this issue, we design a data augmentation technique based on node masking that prioritizes retaining information from minority class nodes. Class imbalance learning generally relies on known labels, but the GCL method we employ is label-free. Existing research often uses pseudo-labels to apply supervised methods to unsupervised or semi-supervised learning. ImGCL [15] generates pseudo-labels through online clustering of node representations in GCL, but this method is unstable during training, as pseudo-labels and node representations influence each other. In our research, we use an Encoder–Decoder architecture for pre-training to generate high-quality pseudo-labels.

In this paper, we propose AIGCL, a new GCL framework for imbalanced node classification, which employs a step-by-step balancing strategy to automatically and adaptively balance the node representations learned from GCL in an unsupervised manner. Our main contributions are as follows:

1. We introduce a Variational Graph Autoencoder (VGAE) based on an Encoder–Decoder architecture. The synergy between the encoder and decoder enables the model to effectively learn both the structural information of the graph and node features. Furthermore, the randomness and diversity introduced by the encoder's variational inference, the reconstruction capability of the decoder, and the enhancement of the encoder's generative ability by the discriminator further improve the model's robustness and generalization. This design lays a solid foundation for generating accurate pseudo-labels.
2. We design a resampling strategy tailored for graph structures. We assign each node a probability based on its importance in the graph and resample according to these probabilities. Specifically, during undersampling, we prioritize removing nodes with lower probabilities, while during oversampling, we focus on synthesizing nodes with higher probabilities.
3. We develop a new data augmentation technique that uses pseudo-label information during node masking to prioritize the retention of information from minority class nodes while masking majority class nodes. This method helps the model better capture the features of the minority class in imbalanced datasets.
4. We conduct extensive experiments to demonstrate the effectiveness and superiority of our method in addressing imbalanced node classification problems. Additionally, we analyze the proposed strategies to validate the improvements.

2. Related work

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph, where $\mathcal{V} = \{v_i\}_{i=1}^{N}$ is the set of N nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges between nodes. $X = [x_1, x_2, \dots, x_N]^T \in \mathbb{R}^{N \times d}$ represents the node feature matrix, with $x_i \in \mathbb{R}^d$ being the feature vector of node i, where d is the dimension of features. $Y = [y_1, y_2, \dots, y_N]^T \in \mathbb{R}^N$ denotes the labels of nodes in the graph. The adjacency matrix A is defined such that if there is an edge between node i and node j, $A_{ij} = 1$; otherwise, $A_{ij} = 0$.

2.1. Graph contrastive learning (GCL)

Graph Contrastive Learning (GCL) is an unsupervised learning method that aims to maximize the consistency of representations between different views in the latent space using a contrastive loss, thereby learning discriminative node representations. This method leverages the structural characteristics of graph data and the advantages of contrastive learning to effectively capture rich information. Its core consists of three modules: augmentation functions, encoders, and contrastive loss functions. The augmentation function is the first step in Graph Contrastive Learning. It generates different views of nodes or graphs to provide rich positive and negative sample pairs for contrastive learning, enhancing the model's generalization ability. We denote the set of these augmentation functions as $\mathcal{T}$. Common examples include random perturbations, data augmentation, and graph structural transformations [1]. The encoder is the core component of Graph Contrastive Learning. We typically use Graph Neural Networks (GNNs) [16] to encode both node feature information and graph structural information. GNNs update node representations by aggregating information from neighboring nodes, capturing both local and global characteristics of the graph. The final module of Graph Contrastive Learning is the loss function, which calculates the similarity between different node views and trains the model by maximizing the similarity between positive sample pairs and minimizing the similarity between negative sample pairs. A commonly used contrastive loss function is InfoNCE [17].

Specifically, we use two augmentation functions $t_1, t_2 \sim \mathcal{T}$ to generate two augmented views $\mathcal{G}_1 = (X_1, A_1) = t_1(\mathcal{G})$ and $\mathcal{G}_2 = (X_2, A_2) = t_2(\mathcal{G})$. The node representations of these two views are obtained through a (parameter-sharing) GNN encoder $f(\cdot)$, denoted as $Z_1 = f(X_1, A_1)$ and $Z_2 = f(X_2, A_2)$, respectively. Using these latent representations, we optimize the parameters of the GNN encoder through a predefined contrastive loss. For any node u in $\mathcal{G}_1$, our goal is to have the score of the positive pair (u, u^+) higher than that of other negative pairs (u, u^-). Typically, the negative sample u^- is drawn from other nodes in the augmented graph views $\mathcal{G}_2$ of the same batch. The commonly used InfoNCE loss [17] can be defined as:

$$\mathcal{L}_{\text{NCE}} = -\frac{1}{N}\sum_{u=1}^{N}\log\frac{\exp(z_u \cdot z_u/\tau)}{\sum_{v=1}^{N}\exp(z_u \cdot z_v/\tau)}, \tag{1}$$

where τ is a temperature hyperparameter.

Graph Contrastive Learning (GCL) has rapidly become one of the mainstream paradigms in graph self-supervised learning since 2019. Early research primarily focused on foundational explorations of two core issues: contrastive objectives and data augmentation strategies [18,19]. In recent years, researchers have continuously introduced new perspectives, significantly enhancing the performance of unsupervised graph representation learning. For example, [20] improves graph embedding expressiveness and effectively captures latent features in the graph structure, thereby enhancing the performance of downstream tasks such as node classification. [21] optimizes the generation process of graph representations, ensuring consistency between different views and further improving the robustness and generalization ability of the model. [22] proposes a community-invariant contrastive learning method that maintains the stability of the community structure in the graph, thus enhancing the performance of graph representations in community-aware tasks. Furthermore, to address the issue of temporal and topological augmentation, researchers have combined temporal information with topological features and proposed a cross-view contrastive learning model [23]. By enhancing temporal and topological features, this approach significantly improves the accuracy of temporal link prediction.

Meanwhile, the field of Graph Contrastive Learning has also started to address issues related to deconfounding and fairness. For example, [24] proposes a deconfounding representation learning method, which eliminates latent confounding effects to improve the accuracy of personalized recommendations. [25] introduces a disentangled contrastive learning approach aimed at removing sensitive attribute biases in graph representations, ensuring fairness in the learning process of graph representations.

2.2. Class imbalance learning (CIL)

Class imbalance learning aims to address the problem of learning from imbalanced class distributions, where head classes (majority classes) have more training instances than tail classes (minority classes), and the model is evaluated on a balanced test set [26]. For a node classification problem with K classes, let $(v_i, y_i)_{i=1}^{N}$ be an imbalanced training set with a total number of samples $N = \sum_{k=1}^{K} N_k$, where N_k denotes the number of samples in the k-th class. When the class distribution is highly skewed, a class imbalance problem arises.

To quantify the severity of class imbalance, we use the imbalance ratio, defined as the ratio of the number of samples between the head and tail classes:

$$\rho = \frac{\max_k(N_k)}{\min_k(N_k)}. \tag{2}$$

Traditional CIL methods are generally categorized into two types: (1) data-level methods, which modify the distribution of training data to balance class distributions, such as undersampling of majority classes [27], oversampling of minority classes [28], and combined sampling [29]; (2) algorithm-level methods, which modify the learning algorithm itself to better handle class imbalance, such as cost-sensitive learning [30], ensemble learning [31], and loss function engineering [32].

2.3. Class imbalance learning on graphs (CILG)

Class imbalance learning on graphs (CILG) refers to learning from imbalanced graphs with skewed class distributions, which makes it challenging to train models that effectively classify nodes, predict relationships, or classify entire graphs. CILG is typically divided into imbalanced node classification and imbalanced graph classification. This paper primarily focuses on the problem of imbalanced node classification.

Existing CILG methods are divided into two main categories: data-level methods and algorithm-level methods. Data-level methods are further classified into data interpolation [4], adversarial generation [7], and pseudo-labeling [8]. Algorithm-level methods include model refinement [11,12], loss function engineering [13], and post-processing adjustments [14].

In recent years, growing attention has been directed toward class-imbalanced learning on graphs (CILG). For instance, [33] introduces a method that integrates Bayesian graph local extrema convolution with a long-tail learning strategy to tackle data imbalance in misinformation detection. By emphasizing minority class nodes, the model significantly improves its performance under skewed class distributions. To further alleviate the dominance of majority classes, [34] proposes the insertion of buffer nodes that disrupt the flow of information from majority to minority classes, thereby enhancing the model's ability to learn from underrepresented nodes. Additionally, [35] combines self-supervised pretraining with edge-level data augmentation to boost the involvement and representation quality of minority class nodes, effectively mitigating the challenges posed by class imbalance.

3. Model

In this paper, we propose a new GCL framework, AIGCL, which aims to learn node embeddings under class imbalance in a self-supervised manner. As shown in Fig. 1, the framework consists of three core components:

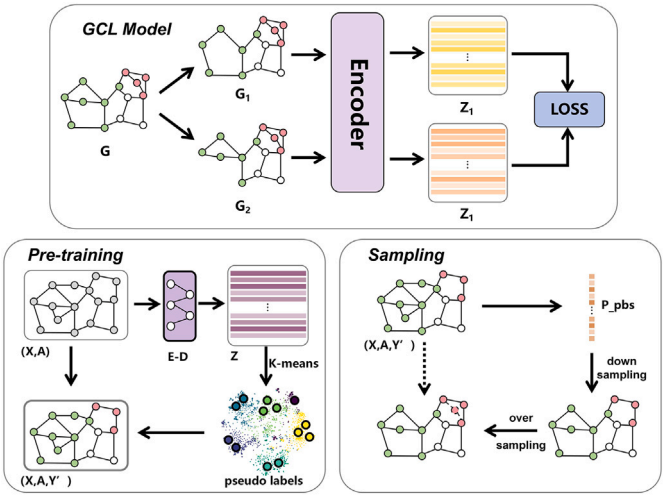


Fig. 1. This figure shows a Graph Contrastive Learning process, including three main modules. The GCL Model module demonstrates our graph augmentation method for retaining more minority class information and uses a graph encoder and loss function to learn graph representations. The Pre-training module uses the trained Encoder-Decoder model to generate high-quality embeddings and generate pseudo-labels through K-means clustering. The Resampling module balances the node class distribution through mixed sampling techniques, including oversampling and undersampling.

(1) A pre-training module for generating high-quality pseudo-labels.
(2) A re-sampling module for initially balancing the node class distribution.
(3) A graph contrastive model that incorporates an augmentation method.

3.1. Pre-training of encoder–decoder model

Existing methods for handling imbalanced data mostly rely on label information to adjust sample weights or perform re-sampling, which become significantly limited in unsupervised environments. Motivated by the observation in [36] that 'nodes with similar features tend to connect and share the same label,' we introduce a pseudo-labeling approach to enable class-imbalance learning in unsupervised scenarios. To this end, we introduce a Variational Graph Autoencoder (VGAE), a sophisticated encoder–decoder framework designed to jointly capture the structural context and semantic attributes of the graph. Through variational inference, each node is mapped to a Gaussian distribution in the latent space—where the mean anchors the most reliable representation, and the variance quantifies the uncertainty in our understanding. The variance not only captures the legitimate diversity inherent in multimodal data but also explicitly encodes the irreducible uncertainty caused by observational noise and information bottlenecks. As a result, the generated latent variables retain both global context and local detail, while maintaining robustness in noisy environments—laying a solid foundation for generating high-quality pseudo-labels.

The encoder of VGAE generates the latent representations using a two-layer Graph Convolutional Network (GCN), formulated as follows:

$$H_v = ReLU(A^* X W_v^{(1)}), \tag{3}$$

$$\mu_v = A^* H_v W_v^{(2)}, \tag{4}$$

$$\log \sigma_v^2 = A^* H_v W_v^{(3)}, \tag{5}$$

$$Z_v = \mu_v + \epsilon \odot \sigma_v, \tag{6}$$

where X and A^* are the input node features and the normalized adjacency matrix, respectively; $W_v^{(1)}$, $W_v^{(2)}$, $W_v^{(3)}$ are the weight matrices of different graph convolutional layers in the model; H_v is the hidden feature output; Z_v is the latent variable generated through variational inference; and μ_v, $\log \sigma_v^2$ are the mean and the logarithm of the variance output by the encoder, respectively.

Then, using the output of the encoder to train the discriminator, we sample random noise from a standard normal distribution $\mathcal{N}(0,1)$ as real samples and treat the encoder's output as generated samples. We design the following loss function to distinguish between real and generated samples:

$$\mathcal{L}_{dis} = \frac{1}{N}\sum_{i=1}^{N} BCE(1, D_{real,i}) + \frac{1}{M}\sum_{j=1}^{M} BCE(0, D_{fake,j}), \tag{7}$$

$$BCE(y, \hat{y}) = -(y \log(\hat{y}) + (1-y)\log(1-\hat{y})), \tag{8}$$

where $D_{\text{real},i}$ is the output of the discriminator for the i-th real sample; $D_{\text{fake},j}$ is the output for the j-th generated sample; $\mathbf{1}$ and $\mathbf{0}$ denote all-one and all-zero vectors shaped like $D_{\text{real},i}$ and $D_{\text{fake},j}$, respectively; N and M are the numbers of real and generated samples; BCE denotes the Binary Cross-Entropy loss function, where y and $\hat{y}$ represent the true label and the predicted probability, respectively.

Subsequently, the internal decoder reconstructs the adjacency matrix $\tilde{A}$ from the latent variable representation Z_v. A reconstruction loss $\mathcal{L}_{\text{recon}}$ is then defined by combining the discriminator loss $\mathcal{L}_{\text{dis}}$ with the KL divergence loss $\mathcal{L}_{\text{KL}}$. We train the Variational Graph Autoencoder (VGAE) using $\mathcal{L}_{\text{recon}}$, encouraging it to both reconstruct the input graph and generate new data. The loss functions are defined as follows:

Algorithm 1: Pre-training model.

Input: Graph data G, encoder–decoder $\mathcal{F}$, classifier $\mathcal{M}$
Output: Trained encoder–decoder $\mathcal{F}$, classifier $\mathcal{M}$

```
Initialize encoder–decoder parameters θ_F, classifier parameters θ_M;
Data normalization;
for each epoch do
    Use the encoder in the encoder–decoder to generate latent variable representation z;
    Use the decoder in the encoder–decoder to reconstruct the adjacency matrix Â from the latent variable representation z;
    for j = 0, 1, 2, ..., 10 do
        Generate real samples z_real from the standard normal distribution N(0, 1);
        Calculate the discriminative loss L_dis between real samples D_{z_real} and generated samples D_z based on Eq. (7);
        Backpropagate L_dis and update the parameters of classifier M: θ_M ← θ_M − η_M · ∇_{θ_M} L_dis;
    Calculate the reconstruction loss L_recon based on Eq. (10);
    Update the parameters of F based on L_recon: θ_F ← θ_F − η_F · ∇_{θ_F} L_recon;
```

$$\mathcal{L}_{KL} = -\frac{1}{2N}\sum_{i=1}^{N}\left(1 + 2\log\sigma_i - \mu_i^2 - \sigma_i^2\right), \tag{9}$$

$$\mathcal{L}_{recon} = \mathcal{L}_{KL} + \text{norm} \cdot \mathcal{L}_{dis}, \tag{10}$$

where μ_i and $\log \sigma_i$ are the mean and the logarithm of the standard deviation output by the encoder, respectively. The synergy between the encoder and decoder enables the model to effectively learn both the structural information of the graph and node features. Furthermore, the randomness and diversity introduced by the encoder's variational inference, the reconstruction capability of the decoder, and the enhancement of the encoder's generative ability by the discriminator further improve the model's robustness and generalization. This design lays a solid foundation for generating accurate pseudo-labels. The specific training process of VGAE is shown in Algorithm 1.

3.2. Imbalance rate adaptive sampling strategy

According to [37], the probability of sampling a node from class k in a given graph is defined as:

$$p_k = \frac{N_k^q}{\sum_{i=1}^{k} N_i^q}, \tag{11}$$

where $q \in [0,1]$. Different sampling strategies correspond to different values of q. When $q = 0$, the same number of nodes is sampled from each class, meaning the model pays equal attention to all classes during training. However, this also leads to sampling only a few nodes from the majority class, resulting in a significant loss of information. Conversely, when $q = 1$, the proportion of sampled nodes reflects the original class distribution, which preserves majority class information but is unfavorable for the minority class. Therefore, neither pure equal-number sampling nor pure proportion-based sampling can effectively address the class imbalance problem.

Therefore, to learn high-quality representations for each class from imbalanced datasets, we combine both sampling strategies and balance them using a hyperparameter α. The sampling probability is defined as:

$$p_{k_1} = \alpha \frac{N_k}{\sum_{k=1}^{K} N_k} + \frac{(1-\alpha)}{K}, \tag{12}$$

where $p_{k_1} \in \mathbb{R}$ is the sampling probability of class k, N_k is the number of nodes in class k, and K is the number of classes. The hyperparameter $\alpha \in [0, 1]$ is generally negatively correlated with the imbalance ratio. Moreover, we believe that Eq. (12) only samples based on classes and ignores the graph structure. Degree is a fundamental measure of node connectivity in a graph, reflecting the importance of nodes, the connectivity of the graph, and the structural characteristics of the network. It is crucial for understanding the properties of the graph and optimizing graph learning tasks. Therefore, we adopt a degree-based sampling strategy to balance the class distribution across all classes. For each node class, we sample nodes with higher degrees at a higher probability to better preserve the inherent structure of the graph in the learning representation:

$$p_{k_2} = \lambda p_{k_1} + \frac{1-\lambda}{D_k}, \tag{13}$$

where $p_{k_2} \in \mathbb{R}^{N_k}$ denotes the sampling probability vector for all nodes in class k, $\overline{D_k}$ is the normalized degree vector of nodes in class k, and λ is a hyperparameter that controls the balance between degree-based and class-based sampling.

To avoid severe information loss caused by excessive downsampling of the majority class, as well as information distortion resulting from excessive oversampling of the minority class, we use the average value to determine the sampling quantity:

$$\text{avg} = \frac{\sum_{k=1}^{K} N_k}{K}. \tag{14}$$

We define minority classes as those whose number of nodes is less than the average number of nodes across all classes, while the remaining classes are considered majority classes. The number of nodes to be retained after downsampling is given by:

$$Less_k = \begin{cases} (N_k - \min(N_k/2, N_k - \text{avg})) \times \beta_{\text{less}}, & \text{if } N_k > \text{avg} \\ N_k, & \text{if } N_k \le \text{avg} \end{cases}, \tag{15}$$

where β_{less} is the hyperparameter controlling the undersampling ratio. We perform undersampling based on p_{k_2}, retaining a specified number of nodes $Less_k$. After undersampling, we assess whether further sampling is necessary. Specifically, we define an imbalance threshold τ, and consider the dataset balanced if the post-undersampling imbalance ratio $\rho' = \frac{\max_k |N_k|}{\min_k |N_k|}$ is less than τ. If the imbalance condition is not met, we proceed to oversample the remaining minority classes. The number of nodes to be generated during oversampling is given by:

$$More_k = \begin{cases} (\min(N_k, \text{avg} - N_k)) \times \beta_{\text{more}}, & \text{if } N_k < \text{avg} \\ 0, & \text{if } N_k \ge \text{avg} \end{cases}, \tag{16}$$

where β_{more} is the hyperparameter controlling the oversampling ratio. We select nodes with a specified quantity $More_k$ based on p_{k_2}, and apply the SMOTE method [5] to generate new nodes for these selected samples.

Using the computed sampling quantity and sampling probability, we construct an oversampling mask. For each target node v_{target} that requires oversampling, we identify its nearest neighbor v_{neigh}, and generate a new node according to the following formula:

$$v_{\text{new}} = \delta v_{\text{target}} + (1-\delta) v_{\text{neigh}}, \tag{17}$$

where $\delta \sim \mathcal{N}(0, 1)$ is a random variable introduced to increase the diversity of the training data and prevent overfitting. The newly generated node v_{new} is initially isolated, so it is necessary to establish edges connecting it with other nodes in the graph to preserve structural integrity and maintain graph connectivity. To this end, we first identify the set of k_{neigh} nearest neighbors of v_{new}, denoted as V_{neigh}:

$$V_{\text{neigh}} = \arg\min_{V' \subseteq V} \sum_{v \in V'} \|v_{\text{new}} - v\|, \tag{18}$$

where V is the set of all nodes, and $V' \subseteq V$ denotes the subset containing the k_{neigh} nearest neighbors of v_{new}. From V', we select n_e nodes to establish edges between them and the newly generated node v_{new}. In addition, we add a self-loop to v_{new} to enhance its self-representation.

3.3. Pseudo-label-based augmentation method

Graph augmentation is a key component of Graph Contrastive Learning. It generates new, realistic data by applying specific transformations that preserve the semantic meaning of the original graph. Currently, node dropping and edge perturbation, as proposed by Ref. [38], are widely adopted augmentation strategies. These methods alter the graph structure or node features to create diverse views, thereby providing abundant positive and negative sample pairs for contrastive learning and enhancing model generalization. However, these augmentation methods face limitations when applied to imbalanced node classification tasks. In imbalanced datasets, the number of nodes in certain classes (minority classes) is significantly lower than that in majority classes. During node dropping, nodes from both majority and minority classes are masked with equal probability, making it difficult to ensure a balanced class distribution in the remaining graph. This often results in the loss of critical information from minority class nodes. To address this issue, we propose a node masking strategy guided by pseudo-labels, which prioritizes the retention of information from minority class nodes. We define the expected number of nodes retained after node masking as:

$$\text{avg}^* = \frac{\sum_{k=1}^{K} N_k \times (1 - p_v)}{K}. \tag{19}$$

Where N_k denotes the number of nodes in class k, calculated based on pseudo-labels, K is the total number of node classes, and $p_v \in [0, 1]$ is the node masking probability. In our model, classes with a node count greater than the mean are defined as majority classes, while those with fewer nodes are considered minority classes. We argue that, in imbalanced data, minority class nodes carry more informative value, and retaining their complete information is beneficial for improving model performance. Therefore, we propose a class-aware augmentation method that prioritizes the preservation of minority class nodes. This method adjusts the masking probability for different classes to control the number of masked nodes. Let $S = \{N_k \mid 1 \le k \le K \text{and } N_k < \text{avg}^*\}$ be the set of node counts corresponding to minority classes. The adjusted masking probability is defined as:

$$p'_v = \begin{cases} \varphi \cdot p_v & , N_k < \text{avg}^* \\ \frac{\sum_{k=1}^{K} N_k}{\sum_{k=1}^{K} N_k - \varphi \cdot \sum_{N_k \in S} N_k} \cdot p_v & , N_k \ge \text{avg}^* \end{cases}. \tag{20}$$

Specifically, we adjust the probability of retaining minority class nodes using a hyperparameter $\varphi \in [0, 1]$. When $\varphi = 0$, it indicates that no minority class nodes are masked during node masking, and the distribution of masked nodes becomes more balanced. Conversely, when $\varphi = 1$, the masking function degenerates into the standard (class-agnostic) node masking function. We show that the imbalance ratio after node masking using the adjusted probability p'_v, denoted as $\rho' = \frac{\max_k |N_k|}{\min_k |N_k|}$, is no greater than the original imbalance ratio $\rho = \frac{\max_k |N_k|}{\min_k |N_k|}$ (see Proof 1 for details). This adjustment allows the model to better preserve the information from minority class nodes, which in turn enhances performance in imbalanced node classification tasks. Subsequently, we follow the training procedure described in Algorithm 2. The augmented graph is passed through the encoder to obtain node embeddings $z_1, z_2 \in \mathbb{R}^{B \times D}$, where

Algorithm 2: The learning algorithm of AIGCL.

Input: Graph data $\mathcal{G}$, encoder $\mathcal{F}$, graph augmentation functions $t_1, t_2 \sim \mathcal{T}$
Output: Pre-trained GNN encoder $\mathcal{F}$

```
Generate pseudo labels Ŷ for graph data G based on the training process described in Section 3.1;
Sample graph data G to obtain G' based on Section 3.2;
Initialize encoder parameter θ;
for each epoch do
    Calculate the node masking probabilities p'_v for each category based on Eq. (21);
    Augment the graph to obtain two augmented graphs G_1 = t_1(G', p'_v), G_2 = t_2(G', p'_v);
    Compute node embeddings Z_1 = F(G_1), Z_2 = F(G_2);
    Calculate L based on Eq. (21);
    Update the parameters of F with L;
```

B is the number of nodes and D is the embedding dimension. We adopt the contrastive loss function proposed in [39] for model training:

$$\mathcal{L}(z_1, z_2) = \sum_i \left(1 - \frac{1}{B} z_{1,i}^{\top} z_{2,i}\right)^2 + \sum_{i \neq j} \left(\frac{1}{B} z_{1,i}^{\top} z_{2,j}\right)^2. \tag{21}$$

Proof. It is easy to know that

$$\min_k |N_k| \cdot p'_v = \min_k |N_k| \cdot (1 - \varphi \cdot p_v),$$

and

$$\max_k |N_k| \cdot p'_v = \max_k |N_k| \cdot \left(1 - \frac{\sum_{k=1}^{K} N_k}{\sum_{k=1}^{K} N_k - \varphi \sum_{N_k \in S} N_k} \cdot p_v\right),$$

because

$$\frac{\sum_{k=1}^{K} N_k}{\sum_{k=1}^{K} N_k - \varphi \sum_{N_k \in S} N_k} \geq 1 \geq \varphi \geq 0.$$

Also, let

$$\eta_{\text{more}} = 1 - \frac{\sum_{k=1}^{K} N_k}{\sum_{k=1}^{K} N_k - \varphi \sum_{N_k \in S} N_k} \cdot p_v,$$

and

$$\eta_{\text{less}} = 1 - \varphi \cdot p_v.$$

It follows that

$$\eta_{\text{more}} \leq \eta_{\text{less}}.$$

Therefore, the post-hiding imbalance rate

$$\rho' = \frac{\max_k |N_k|}{\min_k |N_k|} \cdot \rho \cdot \frac{\eta_{\text{more}}}{\eta_{\text{less}}},$$

which proves that

$$\rho' \leq \rho.$$

Theoretical Analysis 1: From Proof 1, it can be seen that the imbalance rate calculated based on pseudo-labels is effectively reduced when using the node masking method for the minority class. However, pseudo-labels cannot assign accurate labels to every node. We conduct a detailed analysis of the experimental results. By examining the distribution of pseudo-labels, we observe that most minority samples are misclassified as the majority class, but the correctly classified minority nodes tend to have higher confidence. Let $acc_A = \frac{A_1}{A_1+B_1}$ and $acc_B = \frac{B_0}{A_0+B_0}$ represent the accuracy of the majority and minority classes, respectively. We simplify the problem into a binary classification task, where $N_0 = A_0 + B_0$ and $N_1 = A_1 + B_1$ denote the number of nodes in the majority and minority classes. A_1 and B_0 are the numbers of correctly classified nodes, while A_0 and B_1 are the misclassified ones. The true imbalance rate is defined as $\rho_{\text{true}} = \frac{A_0+A_1}{B_0+B_1}$, and the imbalance rate after node masking is $\rho'_{\text{true}} = \frac{A_1\eta_{\text{more}}+A_0\eta_{\text{less}}}{B_0\eta_{\text{less}}+B_1\eta_{\text{more}}}$. We aim to prove that $\rho'_{\text{true}} \leq \rho_{\text{true}}$.

$$(A_1\eta_{\text{more}} + A_0\eta_{\text{less}}) \cdot (B_0 + B_1) \leq (B_0\eta_{\text{less}} + B_1\eta_{\text{more}}) \cdot (A_0 + A_1)$$

Expanding and simplifying, we get:

$$A_1 B_0 \eta_{\text{more}} + A_0 B_1 \eta_{\text{less}} \leq A_1 \cdot B_0 \eta_{\text{less}} + A_0 B_1 \eta_{\text{more}}$$

Rearranging and factoring, we obtain:

$$A_1 B_0 (\eta_{\text{more}} - \eta_{\text{less}}) \leq A_0 B_1 (\eta_{\text{more}} - \eta_{\text{less}})$$

Since $\eta_{\text{more}} - \eta_{\text{less}} \leq 0$, it follows that $A_1 B_0 \geq A_0 B_1$. That is, we need to prove $acc_A \cdot acc_B \geq (1-acc_A)(1-acc_B)$, which simplifies to $acc_A + acc_B > 1$. According to the experimental results, the nodes correctly classified in the minority class have higher confidence, so $acc_B > 0.5$. The majority class has a higher recall rate (the probability that nodes labeled as the majority class are correctly classified), and since the number of nodes in the majority class is much greater than that in the minority class, the experimental results show $A_1 > B_1$, that is, $acc_A > 0.5$. In summary, $acc_A + acc_B > 1$ holds true under the experimental settings of this paper. Therefore, it can be proven that $\rho'_{\text{true}} \leq \rho_{\text{true}}$. □

4. Experiment

In this section, we present empirical results to validate the effectiveness of our AIGCL framework. We conduct extensive experiments on imbalanced graph datasets, aiming to answer the following key questions:

(1) Can AIGCL outperform state-of-the-art (SOTA) methods in imbalanced node classification?
(2) Does AIGCL maintain consistently strong performance across different types of class imbalances and varying imbalance ratios?
(3) What are the advantages of AIGCL for each class in the datasets?

4.1. Datasets and experimental settings

Datasets. We used six widely adopted benchmark datasets—Cora, Citeseer, Pubmed, Photo, Computers, and Coauthor CS—to comprehensively evaluate node classification performance under imbalanced settings. Cora, Citeseer [40], and Pubmed [41] are widely used citation network datasets in Graph Neural Network research. In these datasets, nodes represent papers, edges indicate citation relationships between papers, node features are high-dimensional bag-of-words vectors, and class labels correspond to the academic field of each paper. Photo and Computers [42] are two Amazon co-purchase networks. Here, nodes represent products, edges indicate frequently co-purchased items, node features are bag-of-words encoded product reviews, and class labels denote product categories. Coauthor CS is a collaboration network in the field of computer science. In this dataset, nodes represent authors, edges connect co-authors of at least one paper, node features are constructed from the bag-of-words of paper keywords, and class labels correspond to the research topics of the authors' publications. The detailed statistics of these datasets are presented in Table 1.

Baseline methods. We selected representative baseline methods from both traditional and deep learning approaches:

Table 1
Statistics of datasets. The imbalance ratio is calculated on the entire graph dataset.

Dataset	Cora	Citeseer	Pubmed	Photo	Computers	CS
#Nodes	2708	3327	19717	7650	13752	18333
#Edges	5278	4552	44324	119081	245861	81894
#Features	1433	3703	500	745	767	6805
#Classes	7	6	3	10	8	15
#Train	556	495	1821	760	1371	1824
#Val per class	10	10	10	10	10	10
#Test per class	100	100	100	100	100	100
Imbalanced ratio	4.54	2.66	1.92	5.86	17.73	35.05

(1) Traditional methods include raw features (using raw features as input without considering the graph topology) and DeepWalk [10].
(2) Deep learning methods include DGI [18], GRACE [19], BGRL [43], GBT [39], CCA-SSG [44], HomoGCL [45], PolyGCL [46], and ImGCL [15]. Note that for the four baselines, DGI [18], GRACE [19], BGRL [43], and GBT [39], we used the models provided by the PyGCL [1] open-source library for GCL model training. The remaining baselines were trained using the source code provided in their respective papers. We used the linear classifier from the PyGCL [1] open-source library to compare our model with two classic graph self-supervised learning models and eight state-of-the-art graph self-supervised learning methods.
(3) Supervised methods include GCN [47] and GAT [48].

Imbalance types. To comprehensively evaluate the performance of models under different imbalance types, we introduced two types of imbalances [49]: Exp and Step, parameterized by an imbalance factor. The higher the imbalance factor, the more imbalanced the graph is. The Exp-type class-imbalance distribution follows an exponential decay pattern, forming a smoothly decreasing long-tailed distribution. This distribution more closely resembles real-world scenarios, where the degree of imbalance gradually diminishes across classes. It is typically used to evaluate a model's ability to adapt to continuous and progressive imbalance. In contrast, the Step-type distribution divides all classes into two distinct groups: the first half contains an equal number of nodes per class, while the second half consists of classes with increasing node counts. The extent of this increase is determined by both the number of nodes in the first half and a predefined imbalance factor. This distribution is characterized by a sharp transition, with a significant disparity between the two segments. It is commonly used to assess a model's robustness in scenarios with abrupt class gaps, such as when extremely dominant and underrepresented classes coexist.

Evaluation scheme. For each experiment, we followed the commonly used GCL linear evaluation scheme introduced in [43]. First, we trained the model in a self-supervised manner, and then input the embeddings learned by the self-supervised model into a simple linear classifier for training and testing. Typically, the dataset for Graph Contrastive Learning is divided into training, validation, and test sets in a ratio of 10 %:10 %:80 % [49]. However, this division results in a bias toward the majority class. Therefore, we randomly selected 10 % of the data for training, and the validation and test sets were sampled equally from each class. The specific experimental statistics are summarized in Table 1. For the results in this section, we trained each model with 10 different data splits and reported the average performance along with the corresponding standard deviation to ensure a fair comparison. In the following sections, model performance is evaluated using accuracy and F1 scores.

4.2. Experimental results on node classification

Tables 2 and 3 summarize the experimental performance of imbalanced node classification using the Exp imbalance type with an imbalance factor of 100. Table 2 reports the average and standard deviation of the F1 scores over 10 experiments, while Table 3 presents the average and standard deviation of the accuracy values.

From Tables 2 and 3, we can draw the following conclusions:

(1) The recently proposed universal GCL methods [39,43,45,46] were only evaluated on balanced test sets. However, they experienced significant performance degradation under our imbalanced node classification setting. The ImGCL model [15], which was specifically designed for imbalanced node classification, achieved noticeable performance improvements on multiple datasets. However, there were cases where its performance declined on individual datasets. Our AIGCL model outperformed all baseline models in the imbalanced setting, achieving performance gains of [10.74 %, 10.52 %, 14.2 %, 4.75 %, 5.54 %, 5.38 %] in average accuracy and [11.29 %, 8.3 %, 12.74 %, 4.58 %, 5.11 %, 5.08 %] in average F1 scores across the six widely-used datasets: Cora, Citeseer, Pubmed, Photo, Computers, and CS. Additionally, based on the experimental results and the hypothesis proposed in [15] that the embeddings generated by the GBT model are more suitable for imbalanced node classification, we conducted our study of imbalance methods using the GBT baseline model.
(2) Traditional methods, such as raw features and DeepWalk, performed extremely poorly under imbalanced conditions. The former relies solely on node attribute information, which may overlap significantly across different classes. The latter tends to generate more node sequences from the majority class, thereby ignoring minority class nodes. As a result, the learned embeddings are insensitive to minority class distinctions.
(3) Compared with the supervised learning methods GCN and GAT, the AIGCL framework achieves superior or competitive performance on all datasets, demonstrating its strong generalization ability. This further corroborates the effectiveness of our proposed

Table 2
F1 results of unbalanced node classification. Among them, we bold the best effect, italicize the second best, and underline the third best.

Method	Available data	Cora	Citeseer	Pubmed	Photo	Computers	CS
Features	X	22.65±1.95	20.60±0.96	31.62±2.18	24.86±0.71	23.93±1.47	56.84±2.26
Deepwalk	A	9.02±1.75	9.03±1.54	17.49±1.70	5.54±0.96	4.00±0.49	3.76±1.03
DGI	X,A	46.61±1.14	30.19±2.88	25.01±3.31	21.42±2.04	20.82±1.14	72.62±1.50
GBT	X,A	64.40±4.72	38.77±3.17	53.57 ± 5.52	71.68±2.17	71.10±3.05	78.54±0.87
BGRL	X,A	64.99 ± 2.42	43.81 ± 3.99	53.35±4.03	63.44±4.22	44.93±2.16	81.10 ± 1.13
GRACE	X,A	35.82±2.92	33.27±2.02	21.34±3.98	18.30±0.83	15.93±1.18	51.99±1.37
CCA-SSG	X,A	54.78±3.72	39.26±2.37	48.37±4.50	74.49 ± 4.06	74.47 ± 3.11	68.92±1.07
HomoGCL	X,A	66.67 ± 2.96	39.29±2.78	36.96±2.17	5.66±1.37	3.42±0.59	54.31±2.48
PolyGCL	X,A	61.15±17.79	44.15 ± 2.94	41.39±3.18	67.36±2.71	50.28±1.67	78.75±1.30
ImGCL	X,A	62.36±2.91	43.11±2.66	64.33 ± 4.35	77.42 ± 3.74	74.09 ± 2.83	83.86 ± 1.43
GCN	X,A	45.75±1.96	32.36±1.13	36.92±2.07	19.17±2.05	19.31±0.71	42.86±0.47
GAT	X,A	37.36±3.60	29.61±1.96	20.64±3.07	20.88±5.20	20.54±1.49	42.24±0.72
ours	X,A	**77.41 ± 2.47**	**54.67 ± 2.46**	**78.53 ± 2.42**	**82.17 ± 3.63**	**80.01 ± 3.22**	**89.24 ± 0.96**
↑	X,A	10.74	10.52	14.2	4.75	5.54	5.38

Table 3
ACC results of unbalanced node classification. Among them, we bold the best effect, italicize the second best, and underline the third best.

Method	Available data	Cora	Citeseer	Pubmed	Photo	Computers	CS
Features	X	31.16±1.76	29.12±0.83	41.90±1.51	34.73±1.02	30.91±1.53	**61.65 ± 1.61**
Deepwalk	A	15.50±1.26	17.30±0.63	33.67±0.80	13.04±0.48	10.36±0.46	7.20±0.69
DGI	X,A	53.23±1.08	40.97±2.52	37.63±1.82	32.71±1.80	30.00±1.25	74.07±1.28
GBT	X,A	65.31±3.79	44.82±2.54	58.60 ± 2.94	74.96±2.06	71.88±2.60	79.27±0.81
BGRL	X,A	66.14 ± 2.14	48.17 ± 3.57	56.77±3.17	66.14±3.67	50.27±1.60	81.66 ± 1.01
GRACE	X,A	44.21±2.37	42.93±1.77	35.73±2.09	29.05±0.98	25.80±1.09	58.07±1.07
CCA-SSG	X,A	57.93±3.04	45.15±2.30	53.37±3.75	76.79 ± 3.04	74.20 ± 2.54	70.92±0.82
HomoGCL	X,A	66.44±2.64	46.80±2.11	45.67±1.69	13.21±0.65	10.47±0.59	59.67±1.61
PolyGCL	X,A	63.47±15.48	49.73±2.62	48.70±2.07	69.83±2.72	55.79±1.50	79.78±1.12
ImGCL	X,A	65.04±2.18	48.32±2.14	65.83 ± 3.53	79.15 ± 2.71	74.93 ± 2.25	84.22 ± 1.20
GCN	X,A	52.89±2.01	42.92±0.84	46.33±1.76	32.40±1.97	30.46±0.91	54.33±0.55
GAT	X,A	43.29±3.48	39.53±2.67	35.30±1.59	32.20±4.28	31.51±1.45	53.24±0.80
Ours	X,A	**77.73 ± 2.17**	**58.03 ± 2.09**	**78.57 ± 2.40**	**83.73 ± 2.53**	**80.04 ± 2.98**	**89.30 ± 0.91**
↑	X,A	11.29	8.3	12.74	4.58	5.11	5.08

Table 4
Accuracy under different Factor of exp imbalance type.

Factor	Few			Middle			More			All		
	GBT	ImGCL	ours	GBT	ImGCL	ours	GBT	ImGCL	ours	GBT	ImGCL	Ours
10	76.67	78.33	81.33	87.25	88.50	93.00	91.67	91.33	93.33	85.40	86.30	89.60
20	72.00	75.00	79.33	87.00	83.50	89.75	89.00	93.00	92.33	83.10	83.80	87.40
30	67.33	73.33	77.67	82.75	84.25	89.75	93.00	90.67	91.67	81.20	82.90	86.70
40	69.67	71.33	75.33	77.00	82.25	90.25	92.00	89.67	93.33	79.30	81.20	86.70
50	60.67	66.00	71.67	76.75	80.00	91.00	91.33	92.00	92.00	76.30	79.40	85.50
60	61.67	69.67	76.67	74.25	81.50	83.25	93.00	85.67	91.67	76.10	79.20	83.80
70	65.67	63.00	68.67	71.75	83.25	89.25	91.00	88.00	92.67	75.70	78.60	84.10
80	57.67	66.33	71.00	78.50	80.00	88.25	89.00	87.00	91.00	75.40	78.00	83.90
90	64.67	67.33	68.00	69.75	75.00	90.75	91.67	89.67	90.33	74.80	77.10	83.80
100	65.00	66.67	71.67	65.50	77.00	82.75	87.33	86.00	91.00	71.90	76.60	81.90

Table 5
Accuracy under different Factor for step imbalance type.

Factor	Few			More			All		
	GBT	ImGCL	ours	GBT	ImGCL	ours	GBT	ImGCL	ours
2	86.00	88.20	90.20	94.20	93.20	94.20	90.10	90.70	92.20
3	84.20	87.20	88.40	92.20	90.60	93.80	88.20	88.90	91.10
4	83.00	86.20	87.60	91.40	91.20	94.00	87.20	88.70	90.80
5	78.20	81.00	85.00	93.00	92.80	94.60	85.60	86.90	89.80

framework in handling class imbalance and makes it a preferred model for imbalanced node classification tasks.

4.3. Experimental analysis

Imbalance type analysis. To evaluate Graph Contrastive Learning models under class-imbalanced conditions, we construct training sets using two imbalance sampling strategies: long-tailed sampling and step sampling. Long-tailed sampling is inspired by the long-tailed CIFAR variants in computer vision [6,13], where the number of training samples per class decreases exponentially. Step sampling, following the established semi-supervised node classification protocol [47], sets a single threshold that splits classes into two groups—majority classes far above the threshold and minority classes far below—creating a sharp, cliff-like two-level distribution with no transition in between. Imbalanced learning [50] usually reports the performance on head, middle, and tail classes. Tables 4 and 5 show the performance on the Computers [42] dataset for the Exp and Step imbalance types, respectively. In the Exp imbalance setting, we first divide the training set of Computers into three disjoint groups: {Head, Middle, Tail}. The Head and Tail contain the top and bottom one-third of the classes, respectively. Since Computers has 10 classes, we divide the classes [1–3, 4–7, 8–10] into [Head (3 classes), Mid (4 classes), Tail (3 classes)] according to the exponentially decreasing order. For the Step imbalance type, which includes only two class sizes by design, we simply consider the class with fewer nodes as the minority class (Few) and the class with more nodes as the majority class (More). By jointly analyzing the two tables, we observe the following:

(1) GBT is the baseline model, and its performance drops most rapidly as the imbalance increases, especially in the minority and middle classes. This indicates that the GBT model has limitations when handling highly imbalanced data.
(2) ImGCL, a Graph Contrastive Learning model designed for imbalanced data, improves the performance of the minority class by sacrificing that of the majority class. On the Computers dataset, this strategy leads to a decline in majority-class performance, which in turn affects the overall generalization ability of the model.
(3) Ours refers to our proposed method. It significantly improves the performance of the minority and middle classes while maintaining the performance of the majority class. As a result, it achieves better overall balance and generalization, and outperforms both baselines by a large margin.

Imbalance rate analysis. We conducted a further analysis of the Exp type. Figs. 2 and 3 present the changes in accuracy (%) for [Head, Middle, Tail, All] classes on the Cora and Photo datasets as the imbalance

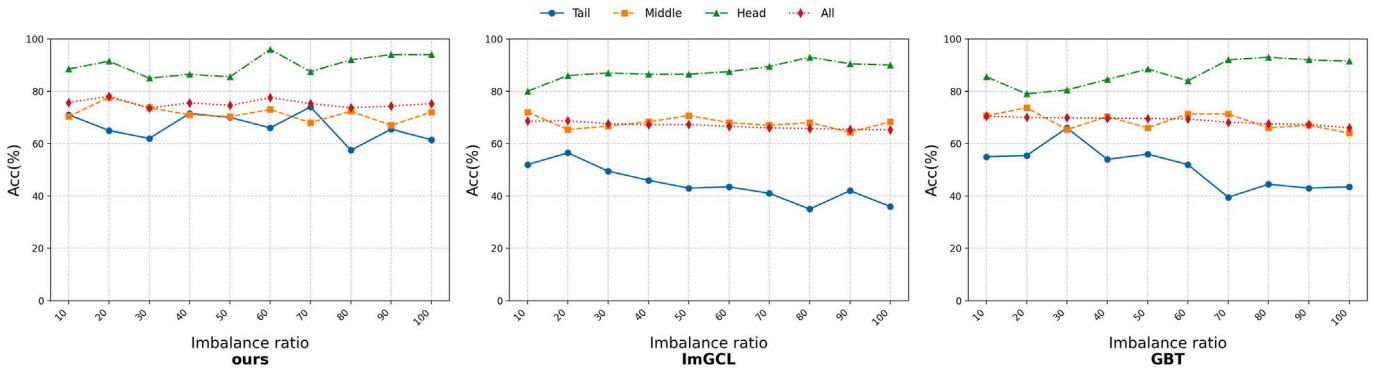


Fig. 2. The performance of different models on the Cora dataset in terms of accuracy (%) for head classes, middle classes, tail classes, and the entire dataset under various imbalance rates.

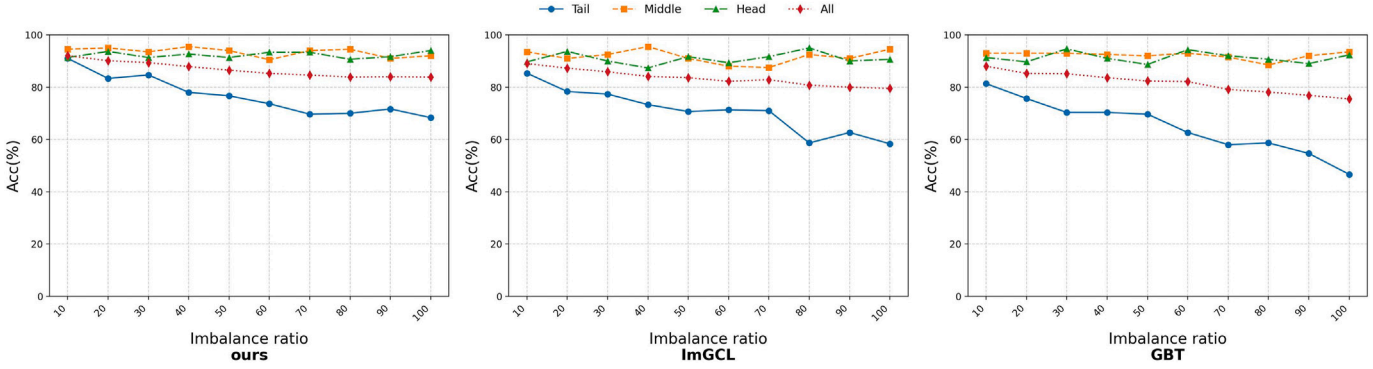


Fig. 3. The performance of different models on the Photo dataset in terms of accuracy (%) for head classes, middle classes, tail classes, and the entire dataset under various imbalance rates.

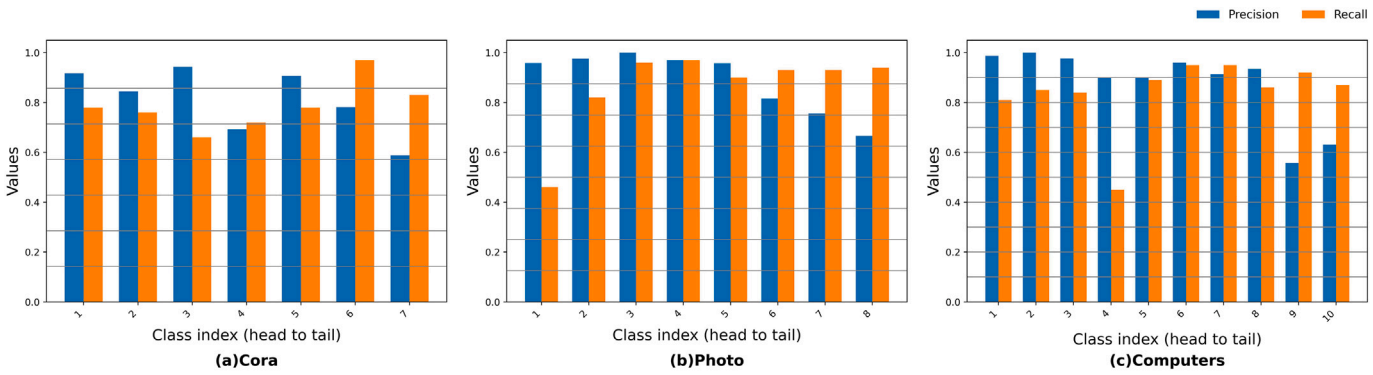


Fig. 4. In the imbalanced experimental setup, the precision and recall for each class on the Cora, Photo, and Computers datasets. Class indices are sorted in descending order based on the number of nodes in each class.

factor increases. The analysis reveals the following findings: (1) As the imbalance factor increases, the accuracy of the tail classes significantly decreases. However, since the head and middle classes maintain relatively high accuracy or even show an upward trend, the overall accuracy decreases slowly. (2) Our method outperforms both ImGCL and GBT on the Cora and Photo datasets. Specifically, our method performs better on the majority classes in Cora and the minority classes in Photo. We believe this is due to the two-stage training process of our GCL-based method, which involves training the graph contrastive model followed by the linear classifier. We apply imbalance handling only during the training of the linear classifier, while the input to the graph contrastive model is the entire graph. In this context, since the class distribution in Cora is relatively balanced, the model can already perform well on the minority classes, and our method further enhances this by retaining more minority class nodes—thereby better capturing their characteristics. On the other hand, the Photo dataset has a more imbalanced class distribution, and models such as ImGCL and GBT, which lack explicit mechanisms for addressing minority class imbalance, perform poorly. In contrast, our approach achieves better performance due to its targeted handling of imbalance.

Per-class analysis. We present the precision and recall for each class in the Cora, Photo, and Computers datasets in Fig. 4. High precision indicates that a large proportion of the samples predicted as positive by the model are indeed positive, while high recall suggests that the model successfully identifies most of the true positive samples. In the first few classes (minority classes), the model exhibits high precision

Table 6
Melting experiment.

	Cora		Citeseer		Pubmed		Photo		Computers		CS	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
Base	65.31	64.40	44.82	38.77	58.60	53.57	74.96	71.68	71.88	71.10	79.27	78.54
w/o pre + z	69.11	67.92	49.33	44.54	67.37	65.76	80.33	79.81	76.78	75.99	84.93	84.71
w/o z	70.17	68.95	53.93	49.17	71.33	70.70	82.85	80.54	78.15	77.76	87.61	87.51
w/o c	75.60	74.93	54.82	51.83	71.87	71.15	82.25	80.56	79.02	78.85	85.67	85.51
Ours	**77.73**	**77.73**	**58.03**	**58.03**	**78.57**	**78.57**	**83.73**	**83.73**	**80.04**	**80.04**	**89.30**	**89.30**

and recall, demonstrating that the AIGCL model can accurately identify samples from these classes. Many existing methods for handling class imbalance [15] improve the performance of minority classes at the cost of degrading the performance of majority classes. However, the AIGCL model achieves high precision and recall across all classes, indicating that our model possesses strong robustness. It effectively captures the characteristics of minority classes while maintaining the stability of majority class performance, thereby achieving a better balance between the two. Moreover, the model's strong performance on different datasets (Cora, Photo, Computers) demonstrates its strong generalization ability, enabling it to adapt to various data distributions and feature types.

4.4. Ablation study

Table 6 displays the accuracy (ACC) and F1 scores of various model variants across multiple datasets. These variants were generated by removing or adding specific components to conduct ablation studies, thereby assessing the impact of each component on the model's performance. Specifically, the components include: *pre*, representing the use of a pre-trained model to generate pseudo-labels; *z*, representing the proposed augmentation function; and *c*, representing the use of a resampling strategy to balance the dataset. The Base model refers to the core model without any additional components, and we adopted the GBT model as the base. The Base model achieves relatively low ACC and F1 scores across all datasets, indicating that the additional components are beneficial for improving performance. The results of the ablation study demonstrate that each component—the pre-trained model, augmentation function, and resampling strategy—contributes positively to performance improvements. In particular, the resampling strategy plays a vital role in addressing class imbalance.

5. Conclusion

In this research, we investigated how to improve the representation of Graph Contrastive Learning (GCL) methods for imbalanced node classification, which is a highly practical yet rarely explored issue. We proposed the AIGCL framework for Graph Contrastive Learning, which automatically and adaptively balances the representations learned from GCL in a step-by-step manner without requiring label information. Experimental results demonstrate that this framework significantly outperforms existing methods on multiple real-world datasets, showcasing its superiority and robustness in handling imbalanced node classification problems. For future work, we will explore additional directions, such as edge- or graph-level imbalanced tasks.

CRediT authorship contribution statement

Xinyu Hu: Writing – review & editing, Writing – original draft, Software, Methodology. **Donghai Guan:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization. **Weiwei Yuan:** Supervision, Formal analysis, Data curation, Conceptualization. **Qi Zhu:** Supervision, Data curation, Conceptualization. **Çetin Kaya Koç:** Investigation, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Donghai Guan reports that financial support was provided by National Natural Science Foundation of China. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No. 62472220) and Jiangsu Province 100 Foreign Experts Introduction Plan(BX2022012).

Appendix A. Hyperparameter analysis

We conducted an analysis of the hyperparameters used in our experimental setup. Among them, p_f represents the node pruning probability, p_e represents the edge pruning probability, and α is the hyperparameter used in the sampling probability calculation. We found that all three hyperparameters exhibit a certain trend of first increasing and then decreasing. Specifically, both p_f and p_e achieve their peak performance at 0.1. We believe this is because when p_f and p_e are too low, too much of the original graph structure is retained, causing the model to rely excessively on the original structure rather than learning more abstract features, resulting in insufficient generalization capability. Conversely, when p_f and p_e are too high, important structural information may be lost, which affects the model's ability to learn the overall structure and key features. According to the equations, the closer α is to 1, the closer the sampled data distribution is to the original distribution; the closer α is to 0, the closer it is to a balanced distribution. In theory, we expect to obtain a balanced dataset for training, and thus, a smaller α is generally preferred. However, in practice, the more imbalanced the data, the more difficult it is to achieve true balance. Forcing balance may result in overfitting to the minority class or an excessive reduction of majority class samples, leading to severe information loss and reduced overall accuracy. As shown in Fig. A.1, on the Computers dataset with an imbalance rate of 17.73, α reaches its optimal value at 0.8. This analysis suggests that these hyperparameters can significantly impact model performance and should be carefully tuned to achieve optimal results.

Appendix B. Analysis of pseudo-label accuracy

To validate the high quality of the pseudo-labels generated by our pre-trained model, we conducted K-means clustering on the embeddings produced by all unsupervised models included in the experimental comparison. The clustering results were evaluated using accuracy (ACC) and F1-score metrics, as shown in Table B.1. The experimental outcomes demonstrate that the pseudo-labels generated by our pre-trained model achieved the highest quality among all the evaluated methods.

Appendix C. Robustness experiments

The experiments in Section 4 demonstrate that the proposed AIGCL model exhibits robustness to long-tail distributions. To further validate this property, we conducted additional edge robustness evaluations by

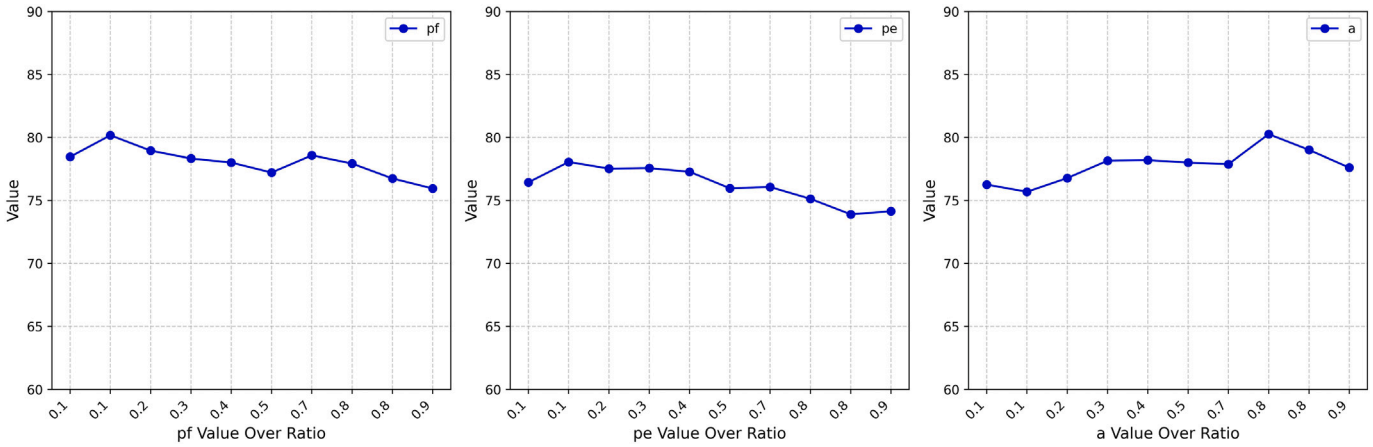


Fig. A.1. Hyperparametric analysis.

Table B.1
Pseudo-label clustering performance (ACC/F1).

Method	Cora		Citeseer		Pubmed		Photo		Computers		CS	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
Features	5.882	9.714	24.358	24.723	41.373	21.211	51.925	1.204	5.163	5.881	51.945	33.974
Deepwalk	22.526	15.352	19.928	14.559	39.083	27.099	17.778	12.649	18.092	10.277	11.215	7.094
DGI	21.529	15.813	29.336	25.818	39.504	32.677	20.484	15.855	24.804	11.656	36.966	25.342
GBT	56.721	52.612	44.034	37.778	51.078	44.574	33.987	18.071	40.954	15.752	68.685	56.661
BGRL	60.007	48.342	43.613	24.556	52.119	58.151	50.118	38.715	32.643	19.288	68.887	63.723
GRACE	55.539	49.398	44.665	43.153	52.270	52.009	34.026	27.272	30.105	17.586	51.208	50.357
CCA-SSG	43.279	34.733	30.778	16.148	60.461	58.579	18.902	14.055	19.699	10.404	12.829	7.086
HomoGCL	53.287	41.631	53.351	51.345	54.227	51.114	17.804	12.807	14.252	9.688	11.978	7.446
PolyGCL	31.979	25.478	23.084	18.006	39.565	34.650	21.190	18.158	20.070	11.662	33.950	24.340
ours	**76.625**	**75.502**	**69.522**	**64.199**	**69.990**	**69.280**	**80.484**	**78.510**	**56.610**	**43.967**	**75.989**	**68.085**

Table C.1
Robustness experiments: ACC and F1 under different noise ratios.

Noise Ratio	Cora		Citeseer		Pubmed	
	ACC	F1	ACC	F1	ACC	F1
0 %	77.73±2.17	77.41±2.47	58.03±2.09	54.67±2.46	78.57±2.40	78.53±2.42
20 %	77.63±3.53	77.61±3.57	58.21±1.81	54.97±2.65	80.06±2.87	80.03±2.97
40 %	78.86±1.56	78.84±1.52	58.25±2.76	56.74±2.99	81.07±2.04	81.09±2.02
60 %	78.85±2.87	78.72±2.95	57.63±2.67	55.77±2.66	80.57±2.27	80.61±2.23

introducing varying proportions of noisy edges during training and assessed the model using ACC and F1-score metrics. As shown in Table C.1, the results indicate that adding different proportions of noisy edges had minimal impact on model performance, thereby confirming the model's robustness to edge noise.

Data availability

Data will be made available upon request.

References

[1] Y. Zhu, Y. Xu, Q. Liu, S. Wu, An empirical study of graph contrastive learning, in: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2), 2021, https://openreview.net/forum?id=UuUbIYnHKO.

[2] Y. Ma, P. Gerard, Y. Tian, Z. Guo, N.V. Chawla, Hierarchical spatio-temporal graph neural networks for pandemic forecasting, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022, pp. 1481–1490.

[3] Y. Tian, C. Zhang, Z. Guo, Y. Ma, R. Metoyer, N.V. Chawla, Recipe2Vec: multi-modal recipe representation learning with graph neural networks, 2022, arXiv preprint arXiv:2205.12396.

[4] T. Zhao, X. Zhang, S. Wang, GraphSMOTE: imbalanced node classification on graphs with graph neural networks, in: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, 2021, pp. 833–841.

[5] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.

[6] J. Park, J. Song, E. Yang, GraphENS: neighbor-aware ego network synthesis for class-imbalanced node classification, in: International Conference on Learning Representations, 2021.

[7] L. Qu, H. Zhu, R. Zheng, Y. Shi, H. Yin, ImGAGN: imbalanced network embedding via generative adversarial graph networks, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 1390–1398.

[8] Z. Wang, X. Ye, C. Wang, Y. Wu, C. Wang, K. Liang, RSDNE: exploring relaxed similarity and dissimilarity from completely-imbalanced labels for network embedding, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, 2018.

[9] J. Wu, J. He, Y. Liu, ImVerde: vertex-diminished random walk for learning imbalanced network representation, in: 2018 IEEE International Conference on Big Data (Big Data), IEEE, 2018, pp. 871–880.

[10] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: online learning of social representations, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 701–710.

[11] C. Ma, J. An, X.-E. Bai, H.-Q. Bao, Attention and cost-sensitive graph neural network for imbalanced node classification, in: 2022 IEEE International Conference on Networking, Sensing and Control (ICNSC), IEEE, 2022, pp. 1–6.

[12] Y. Han, K. Park, Y. Jung, L.-S. Kim, EGCN: an efficient GCN accelerator for minimizing off-chip memory access, IEEE Trans. Comput. 71 (12) (2022) 3127–3139.

[13] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, S. Belongie, Class-balanced loss based on effective number of samples, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9268–9277.

[14] J. Song, J. Park, E. Yang, TAM: topology-aware margin loss for class-imbalanced node classification, in: International Conference on Machine Learning, PMLR, 2022, pp. 20369–20383.

[15] L. Zeng, L. Li, Z. Gao, P. Zhao, J. Li, Imgcl: revisiting graph contrastive learning on imbalanced node classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, 2023, pp. 11138–11146.
[16] F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Trans. Neural Netw. 20 (1) (2008) 61–80.
[17] A. V. D. Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, 2018, arXiv preprint arXiv:1807.03748.
[18] P. Veličković, W. Fedus, W.L. Hamilton, P. Liò, Y. Bengio, R.D. Hjelm, Deep Graph Infomax, in: International Conference on Learning Representations, 2019, https://openreview.net/forum?id=rklz9iAcKQ.
[19] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, L. Wang, Graph contrastive learning with adaptive augmentation, in: Proceedings of the Web Conference 2021, 2021, pp. 2069–2080.
[20] S. Zu, L. Li, J. Shen, W. Tang, Contrastive graph auto-encoder for graph embedding, Neural Netw. 187 (2025) 107367.
[21] Y. Shou, H. Lan, X. Cao, Contrastive graph representation learning with adversarial cross-view reconstruction and information bottleneck, Neural Netw. 184 (2025) 107094.
[22] S. Tan, D. Li, R. Jiang, Y. Zhang, M. Okumura, Community-invariant graph contrastive learning, 2024, arXiv preprint arXiv:2405.01350.
[23] D. Li, S. Tan, Y. Wang, K. Funakoshi, M. Okumura, Temporal and topological augmentation-based cross-view contrastive learning model for temporal link prediction, in: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, 2023, pp. 4059–4063.
[24] G. Zhang, G. Yuan, D. Cheng, L. Liu, J. Li, Z. Xu, S. Zhang, Deconfounding representation learning for mitigating latent confounding effects in recommendation, Knowl. Inf. Syst 67 (7) (2025) 1–22.
[25] G. Zhang, G. Yuan, D. Cheng, L. Liu, J. Li, S. Zhang, Disentangled contrastive learning for fair graph representations, Neural Netw. 181 (2025) 106781.
[26] Y. Ma, Y. Tian, N. Moniz, N.V. Chawla, Class-imbalanced learning on graphs: a survey, ACM Comput. Surv. (2023).
[27] I. Mani, I. Zhang, Knn approach to unbalanced data distributions: a case study involving information extraction, in: Proceedings of Workshop on Learning from Imbalanced Datasets, vol. 126, ICML United States, 2003, pp. 1–7.
[28] S.S. Mullick, S. Datta, S. Das, Generative adversarial minority oversampling, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1695–1704.
[29] J.A. Sáez, J. Luengo, J. Stefanowski, F. Herrera, Smote–IPF: addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering, Inf. Sci. (NY) 291 (2015) 184–203.
[30] I. Araf, A. Idri, I. Chairi, Cost-sensitive learning for imbalanced medical data: a review, Artif. Intell. Rev. 57 (4) (2024) 80.
[31] P. Mahajan, S. Uddin, F. Hajati, M.A. Moni, Ensemble learning for disease prediction: a review, in: Healthcare, vol. 11, MDPI, 2023, p. 1808.
[32] K. Cao, C. Wei, A. Gaidon, N. Arechiga, T. Ma, Learning imbalanced datasets with label-distribution-aware margin loss, Adv. Neural Inf. Process. Syst. 32 (2019).
[33] G. Zhang, S. Zhang, G. Yuan, Bayesian graph local extrema convolution with long-tail strategy for misinformation detection, ACM Trans. Knowl. Discov. Data 18 (4) (2024) 1–21.
[34] Q. Wang, Z. Liu, Z. Zhang, B. He, Buffgraph: enhancing class-imbalanced node classification via buffer nodes, 2024, arXiv preprint arXiv:2402.13114.
[35] X. Juan, F. Zhou, W. Wang, W. Jin, J. Tang, X. Wang, INS-GNN: improving graph imbalance learning with self-supervision, Inf. Sci. (NY) 637 (2023) 118935.
[36] M. McPherson, L. Smith-Lovin, J.M. Cook, Birds of a feather: homophily in social networks, Annu. Rev. Sociol. 27 (1) (2001) 415–444.
[37] J. He, A. Kortylewski, S. Yang, S. Liu, C. Yang, C. Wang, A. Yuille, Rethinking re-sampling in imbalanced semi-supervised learning, 2021, arXiv preprint arXiv:2106.00209.
[38] H. Hafidi, M. Ghogho, P. Ciblat, A. Swami, GraphCL: contrastive self-supervised learning of graph representations, 2020, arXiv preprint arXiv:2007.08025.
[39] P. Bielak, T. Kajdanowicz, N.V. Chawla, Graph barlow twins: a self-supervised representation learning framework for graphs, Knowl. Based Syst. 256 (2022) 109631.
[40] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, T. Eliassi-Rad, Collective classification in network data, AI Mag. 29 (3) (2008) 93.
[41] G. Namata, B. London, L. Getoor, B. Huang, U. Edu, Query-driven active surveying for collective classification, in: 10th International Workshop on Mining and Learning with Graphs, vol. 8, 2012, pp. 1.
[42] O. Shchur, M. Mumme, A. Bojchevski, S. Günnemann, Pitfalls of graph neural network evaluation, 2018, arXiv preprint arXiv:1811.05868.
[43] S. Thakoor, C. Tallec, M.G. Azar, M. Azabou, E.L. Dyer, R. Munos, P. Veličković, M. Valko, Large-scale representation learning on graphs via bootstrapping, in: International Conference on Learning Representations (ICLR, 2022.
[44] H. Zhang, Q. Wu, J. Yan, D. Wipf, S.Y. Philip, From canonical correlation analysis to self-supervised graph neural networks, in: Thirty-Fifth Conference on Neural Information Processing Systems, 2021.
[45] W.-Z. Li, C.-D. Wang, H. Xiong, J.-H. Lai, HomoGCL: rethinking homophily in graph contrastive learning, in: Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining, 2023, pp. 1341–1352.
[46] J. Chen, R. Lei, Z. Wei, PolyGCL: graph contrastive learning via learnable spectral polynomial filters, in: The Twelfth International Conference on Learning Representations, 2024.
[47] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016, arXiv preprint arXiv:1609.02907.
[48] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, in: International Conference on Learning Representations (ICLR), 2018.
[49] Y. Zhang, X. Wei, B. Zhou, J. Wu, Bag of tricks for long-tailed visual recognition with deep convolutional neural networks, AAAI 35 (4) (2021) 3447–3455.
[50] Y. Zhang, B. Kang, B. Hooi, S. Yan, J. Feng, Deep long-tailed learning: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 45 (9) (2023) 10795–10816.

Author biography

Xinyu Hu received the bachelor's degree from Anhui University of Technology, in 2023. She is currently pursuing the master's degree with Nanjing University of Aeronautics and Astronautics, China. Her current research interests include Graph Contrastive Learning, and imbalanced learning.

Donghai Guan is an associate professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. He received the Ph.D. degree in Computer Engineering from Kyung Hee University (KHU), Suwon, Korea in 2009. He was a research professor in KHU during 2009–2011, assistant professor in KHU during 2012–2014. His research interests include machine learning, intelligent systems, and ubiquitous computing.

Weiwei Yuan is a professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. She received a Ph.D. from the Department of Computer Engineering, Kyung Hee University, South Korea, in 2010. She was a postdoc researcher in Kyung Hee University during 2012–2014. Her current research interests include pattern recognition, image processing, and social computing.

Qi Zhu received his B.S. degree, M.S. degree, and Ph.D. degree from Harbin Institute of Technology in 2007, 2010, and 2014, respectively. He is a professor at College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. He has published over 100 scientific articles in refereed international journal and conference proceedings. His interests include pattern recognition and multi-modal data fusion.

Çetin Kaya Koç (Life Fellow, IEEE) received a B.S. degree in Electrical Engineering from İstanbul Technical University (1980) and a Ph.D. in Electrical and Computer Engineering from the University of California Santa Barbara (1988). His research interests include cryptographic engineering, random number generators, finite field and ring arithmetic, homomorphic encryption and machine learning. Koç is the founding editor-in-chief of the Journal of Cryptographic Engineering, published by Springer since 2011, with focus on cryptographic hardware and software development. Koç was elected as an IEEE Fellow in 2007 and IEEE Life Fellow in 2023 for his contributions to cryptographic engineering.