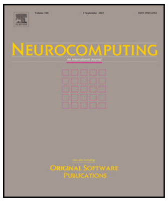
Contents lists available at ScienceDirect

Neurocomputing

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



Graph Neural Network with curriculum learning for imbalanced node classification

Xiaohe Li [a], Zide Fan [a,*], Feilong Huang [a], Xuming Hu [b], Yawen Deng [a], Lei Wang [a], Xinyu Zhao [a]

[a] Aerospace Information Research Institute, Chinese Academy of Sciences & Key Laboratory of Network Information System Technology (NIST), Beijing, 100094, China
[b] School of Software, Tsinghua University, Beijing, 100084, China

ARTICLE INFO

Communicated by J. Andreu-Perez

Keywords:
Graph neural network
Imbalanced classification
Curriculum learning

ABSTRACT

Graph Neural Network (GNN) stands as an emerging methodology for graph-based learning tasks, particularly for node classification. This study elucidates the susceptibility of GNN to discrepancies arising from imbalanced node labels. Conventional solutions for imbalanced classification, such as resampling, falter in node classification task, primarily due to their negligence of graph structure. Worse still, they often exacerbate the model's inclination towards overfitting or underfitting, especially in the absence of adequate priori knowledge. To circumvent these limitations, we introduce a novel **G**raph **N**eural **N**etwork framework with **C**urriculum **L**earning (GNN-CL). This framework integrates two pivotal components. Initially, leveraging the principles of smoothness and homophily, we endeavor to procure dependable interpolation nodes and edges via adaptive graph oversampling. For another, we combine the Graph Classification Loss with the Metric Learning Loss, thereby refining the spatial proximity of nodes linked to the minority class in the feature space. Drawing inspiration from curriculum learning, the parameters of these components are dynamically modulated during the training phase to accentuate generalization and discrimination capabilities. Comprehensive evaluations on several widely used graph datasets affirm the superiority of our proposed model, which consistently outperforms the existing state-of-the-art methods.

1. Introduction

Graph neural network (GNN), an innovative approach for mining graph-structured data in non-Euclidean spaces, has received extensive research attention in recent years [1–3]. GNN adeptly tackles complex challenges within network topologies, such as node classification [4,5], edge prediction [6–8], and clustering [9]. These methods find extensive applications in various intrinsic areas such as social network recommendations, molecular structure analysis, and developing early warning systems for financial risks. Semi-supervised node classification epitomizes a prototypical application scenario for GNN models, yet its performance is considerably modulated by the data distributions. Currently, GNN models are predominantly based on the propagation–aggregation mechanism like GCN [4] and GraphSAGE [10]. A significant challenge emerges in scenarios with limited labeled samples and imbalanced class distributions within the training dataset. When the mentioned GNN models meet such imbalanced situations, minority samples fail to influence others effectively due to sparse connectivity. Additionally, the deep graph classifiers' accuracy is hampered by the dual constraints of data scarcity and class imbalance.

Regrettably, these issues are pervasive in real-world graph data. In Fig. 1, we illustrate the long-tail distribution of node categories in the BlogCategory and Citeseer datasets, highlighting the model's susceptibility to class imbalances through empirical evidence.

Multidisciplinary contemporary research has proposed various strategies to tackle imbalanced class distributions. A prevalent approach is the resampling mechanism, incorporating both oversampling and undersampling techniques, aiming at equalizing data distributions across minority and majority classes. Interpolation-based methods, like SMOTE [11], originate from this strategy. Nonetheless, such techniques might inadvertently affect the accuracy of class evaluations, often due to potential overfitting or underfitting, where improper resampling scales can result in an excessive focus on minority samples or the inadvertent omission of valuable information from majority samples. Another noteworthy approach is cost-sensitive learning, which augments the weight assigned to the classification loss of the minority class. However, determining the precise weightings for different classes

* Corresponding author.
E-mail addresses: lixiaohe@aircas.ac.cn (X. Li), fanzd@aircas.ac.cn (Z. Fan), huangfeilong22@mails.ucas.ac.cn (F. Huang), hxm19@mails.tsinghua.edu.cn (X. Hu), dengyawen@aircas.ac.cn (Y. Deng), wanglei002931@aircas.ac.cn (L. Wang), zhaoxinyu@aircas.ac.cn (X. Zhao).

https://doi.org/10.1016/j.neucom.2023.127229
Received 13 March 2023; Received in revised form 29 November 2023; Accepted 30 December 2023
Available online 5 January 2024
0925-2312/© 2024 Elsevier B.V. All rights reserved.

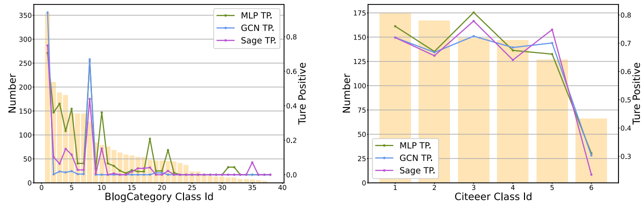


Fig. 1. Imbalanced class distributions of training data (from majority to minority) in the BlogCategory and Citeseer datasets. It includes the True Positive (TP.) test result of MLP, GCN and GraphSAGE. From this case, we can observe a significantly high correlation between accuracy and class proportion.

remains challenging, given the absence of prior knowledge regarding the datasets.

Current research has not extensively addressed class imbalance in graph node classification. In reflecting upon previous endeavors, Zhao et al. [12] extend oversampling algorithms and train an edge generator to enhance the graph's structure. However, they ignore the complex relationships and feature interactions between nodes, so that the quality of the generated edges cannot be guaranteed. In some studies previously proposed for addressing the class imbalance in node classification, there indeed lies potential utility to improve the quality of node representations, but it also has some limitations: (1) In high-dimensional feature spaces, owing to the non-Euclidean characteristics of graph structures, features derived from newly sampled subgraphs may not necessarily be dependable. Features of minority categories generated during the initial training stages tend to diffuse globally, resulting in the original node features being permeated with commingled information. (2) Maintaining a balanced distribution throughout the training process will cause a negative effect on generalization since the classifiers in GNN overemphasize the minority nodes, especially for overly imbalanced datasets. (3) The structural connectivity information between nodes remains inadequately leveraged after data distribution transformation.

To address the limitations of existing GNN models, we propose a **G**raph **N**eural **N**etwork framework with **C**urriculum **L**earning (GNN-CL). Delving into specifics, we initially employ an acceptable sampling strategy for addressing the imbalance issue and meticulously devise a graph-based adaptive oversampling method. This technique enhances the effective representations of nodes and edges within the graph. We aim to derive reliable interpolated minority nodes by leveraging the existing embeddings from the intermediate layers. It is imperative to connect the newly generated nodes with the extant components of the graph in pursuit of preserving the integrity of the graph structure, enhancing the volume of information propagation within the graph network, and attenuating the noise. Inspired by [13], we lean on two quantifiable metrics – smoothness and homophily, both rooted in node features and labels – to safeguard the volume and veracity of information gain. This strategy, in turn, bolsters the classifier's reliability for the minority class.

Furthermore, appropriate feature representations facilitate delineating clear classification boundaries within the feature space. Consequently, we are also committed to enhancing the representation quality of both original and synthetic minority class nodes. We pay attention to metric learning for incremental rectification and add a neighbor-based triplet loss, which discovers sparse boundaries of minority class samples. Ulteriorly, nodes of the minority class, which yield high-confidence scores within the classifier, are selected to serve as anchor points. It looks like the class rectification loss (CRL) function introduced by Chen et al. [14]. Motivated by this intuition, we combine the **G**raph **C**lassification **L**oss (GCL) for assigning labels with the **N**eighbor-based **T**riplet **L**oss (NTL) that separates different samples associated with the minority class in the feature space by adjusting the distance between nodes.

Ultimately, to overcome the aforementioned overfitting issues and prevent loss of majority class information, we advocate an easy-to-hard training paradigm inspired by the principles of curriculum learning [15]. At the initial stage of training, to circumvent excessive concern about the minority class that could hurt the model's generalizability, we reduce the oversampling ratio. It ensures that the system can procure accurate classification outcomes on an integral part of the samples. Concurrently, by adopting relative relaxation confidence conditions, appropriate representations of minority features are fleetly obtained. As the training progresses, nodes and edges are then generated incrementally to amplify the influence of the minority class in the graph, which makes the classifier focus more on the classification accuracy of minority nodes. During the realization, we modulate the threshold parameters in the two loss functions, ensuring that the framework first acquires suitable feature representations and produces high-quality samples to optimize the classifier appropriately. These dual processes can be parameterized by the overall curriculum learning strategy, wherein their opposite tendency should be coordinated together.

The main contributions of this work are summarized as follows:

- We propose two associated components to tackle the imbalance problem of node classification. For one thing, we design a novel adaptive graph oversampling method based on smoothness and homophily, capable of generating more reliable nodes and edges. For another, we develop a neighbor-based metric learning loss based on a specialized triplet loss to adjust the distance between nodes related to the minority class in the feature space.
- We propose a novel graph neural network framework with curriculum learning (GNN-CL), incorporating two newly proposed loss functions. During the training process, we dynamically adjust the parameters of different modules and control the learning process from easy to hard in order to improve the generalization ability and the convergence rate of the model. Subsequent demonstrations indicate that this integrated approach substantially promotes representation learning and classifier training, particularly in the context of imbalanced data distribution.
- To show the effectiveness of node embeddings learned by our model, we compare the proposed GNN-CL with many state-of-the-art baselines across five real-world datasets for semi-supervised node classification tasks. Further analysis and visualization intuitively reveal the superiority of the proposed model.

2. Related work

2.1. Graph neural network

Graph neural network (GNN) has been a widely used classical model in the last few years, which transforms the complicated input graph-structure data into meaningful representations for downstream mining tasks by passing information and aggregation according to network dependencies. Among all GNNs, graph convolutional network (GCN) is believed to become a dominating solution, falling into two categories: the spectral and the spatial method. With respect to the spectral domain, Bruna et al. [16] proposed to utilize the Fourier-based vector to perform convolution in the spectral domain. ChebNet [17] introduced that smooth filters in spectral convolutions can be well-approximated by K-order Chebyshev polynomials. Kipf et al. [4] presented a convolutional architecture via a localized first-order approximation of spectral graph convolutions, which further constrains and simplifies the parameters of ChebNet [17]. In contrast, spatial methods are defined directly on the graph, operating on the target node and its topological neighbors, so as to realize the aggregation operation on the graph structure. For instance, Hamilton et al. proposed GraphSAGE [10], which generated embeddings by sampling and aggregating features from the nodes' local neighbors. In addition, many works utilize attention layers in neural networks, such as GAT [18], which leverages

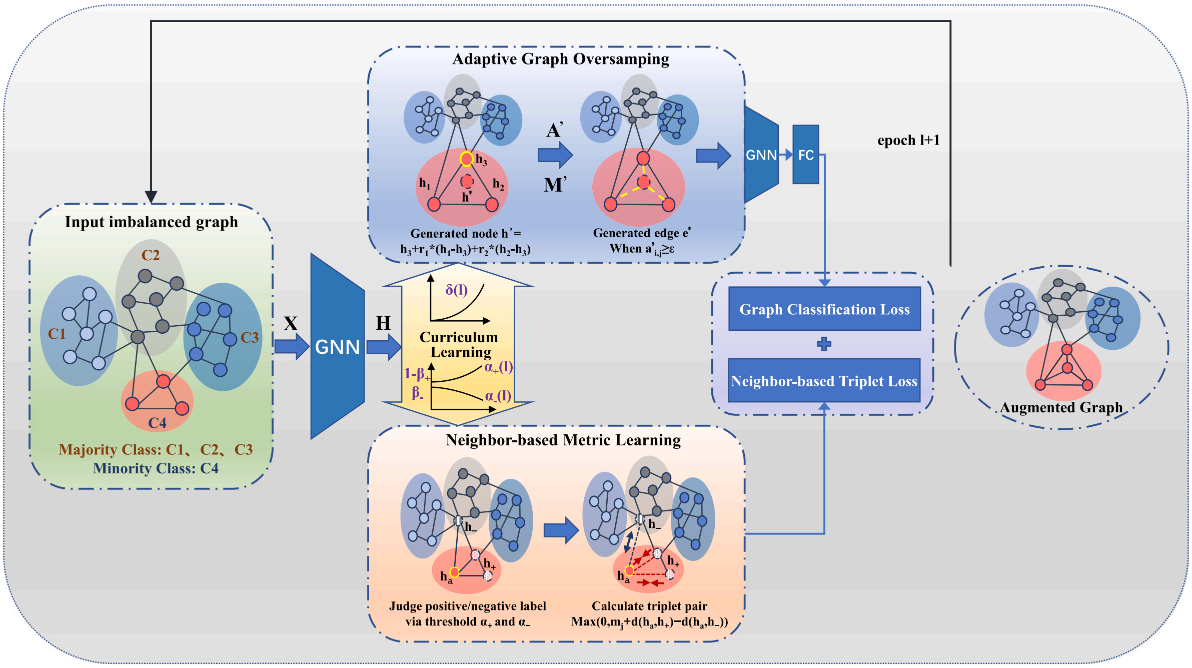

Input imbalanced graph
C1
C2
C3
C4
Majority Class: C1、C2、C3
Minority Class: C4
X
GNN
H
Adaptive Graph Oversamping
h1
h2
h3
h'
A'
M'
Generated node h'= h3+r1*(h1-h3)+r2*(h2-h3)
Generated edge e' When a'i,j≥ε
GNN
FC
epoch l+1
δ(l)
Curriculum Learning
1-β+
β-
α+(l)
α-(l)
Graph Classification Loss
Neighbor-based Triplet Loss
Augmented Graph
Neighbor-based Metric Learning
h−
h+
ha
Judge positive/negative label via threshold α+ and α−
Calculate triplet pair Max(0,mj+d(ha,h+)−d(ha,h−))

Fig. 2. The proposed model framework. It includes two loss functions: Graph Classification Loss (GCL) and Neighbor-based Triplet Loss (NTL). And they are scheduled through a unified curriculum learning framework.

masked self-attention to specify different weights to different nodes in the neighbors.

However, these methods do not address the bias caused by majority class nodes in the implementation process, for which they are unsuitable for imbalanced node classification problems.

2.2. Imbalanced learning

There are currently different groups of methods for reducing the bias [19] caused by disequilibrium of class proportion by intervening in the class distribution when training. (1) Resampling is the process of transferring the data into a balanced distribution [20,21], which can be disassembled into two types. One is oversampling, which adjusts the proportion of data samples by simply copying samples from the minority class. An advanced sampling method called SMOTE [11] also expands artificial samples by interpolating between similar samples. Another is undersampling, which balances the sample proportions by weeding out the majority classes of samples. However, such methods may cause overfitting or underfitting problems due to repeated visits to duplicate samples or forgoing important information. (2) Reweighting is another kind of method for maintaining balance in the training process by adjusting the objective functions. In cost-sensitive learning, researchers intend to assign varying weights to different classes, such as a higher loss for samples [22] of the minority class. In contrast, the threshold adjustment technique changes the decision threshold when testing [23]. Unfortunately, the lack of prior knowledge about different datasets and backgrounds makes it difficult to guarantee how to set the weights properly. (3) Hybrid methods are devoted to combining the above categories. For example, EasyEnsemble and BalanceCascade propose a committee of classifiers on undersampled subsets [24]. SMOTEBoost [25] combines the boosting technology and SMOTE oversampling. Furthermore, researchers introduce some novel methods, such as metric learning [26] and meta-learning. Moreover, there are also neural-network-based methods for imbalanced classification learning.

For all that, few studies have worked on the imbalanced classification problem on graphs. GraphSMOTE [12] devises a new framework that applies SMOTE algorithm for graph-based data. It constructs a node generator according to node similarity and trains an edge generator for isolated synthetic nodes. Approximately, GraphMixup [27] synthesizes additional nodes in the semantic space and exploits a reinforcement mixing mechanism to adaptively determine how many samples need to be generated for those minority classes. But, they do not make full use of the topological information of the graph. Moreover, maintaining a balanced distribution throughout the training process will have a detrimental effect on generalization since the GNN classifier will give too much attention to the minority class.

3. The proposed model

In order to solve the problem of semi-supervised node classification from imbalanced graph-based data, we propose to construct a graph neural network model with curriculum learning, which has the ability to infer the type of unknown nodes. Before a detailed introduction, we first give some necessary definitions. In an imbalanced graph, denoted as $G = (V, E, F)$, where $V = \{v_i\}_{i=1}^{N}$ indicates the node set and $E = \{e_{i,j}\}$ indicates the edge set which connect v_i and v_j. Let $F \in \mathbb{R}^{n \times d}$ denotes the feature matrix of samples and let $Y = \{1, 2, \ldots, C\}$ be its corresponding label, where C is the number of classes. In imbalanced datasets, the number of nodes $|N_i|_{i=1}^{C}$ belonging to different labels varies greatly.

Intending to jointly learn node features and the label classifier from the imbalanced training set in an end-to-end procedure, we propose a novel Graph Neural Network framework with Curriculum Learning (GNN-CL) for imbalanced node classification, consisting of two novel components shown in Fig. 2. The first is an adaptive graph oversampling, the fundamental idea of which is to interpolate the most significant samples related to the original structure. The primary purpose is to dynamically reset the data distribution from imbalance to balance. The second is neighbor-based metric learning. By this way, the distances between nodes and their neighbors are regularized according to pseudo positive/negative labels, so as to dynamically adjust the coordinates of minority class nodes in the feature space. The proposed model leverages two losses: Graph Classification Loss and Neighbor-based Triplet Loss in the whole learning procedure. Above them, we put up an overall curriculum scheduling strategy consisting of two opposite learning curves. In the early stage of the training process, our proposed framework focuses more on optimizing feature propagation and reducing biased noise in the latent feature space. As training continues, it gradually pays more attention to the average accuracy in each class.

3.1. Adaptive graph oversampling

As mentioned in the previous section, we first need an innovative oversampling strategy to handle imbalanced graph classification by reasonably acquiring a balanced augmented graph. Intuitively, the superiority of Euclidean-based GNN models can be attributed to the computing paradigm, which effectively integrates the abundant information from neighboring nodes [28]. Wherein spatial-based GNN models mainly employ designed aggregators to assimilate information from neighboring nodes, such as the mean aggregator [10], the sum aggregator [29], or the attention aggregator [18]. Spectral-based GNN models, such as GCN, employ the Laplacian operator for convolution operation over neighboring nodes. Drawing inspiration from [30], we employ the theory of sensitivity analysis and influence functions. Within GCN framework, the influence $I(i,k)$ of node v_i at the Lth layer on node v_k during the training process is defined based on gradient calculation as follows:

$$
\begin{aligned}
I(i,k) &= \|\mathbb{E}(\partial h_{v_i}^{(L)}/\partial h_{v_k})\| \\
&= \|\sqrt{d_i d_k} \cdot \sum_{p=1}^{\psi} \mathbb{E}(\prod_{l=L}^{0} \frac{1}{d_{p^l}} diag(\mathbb{1}_{\sigma_l}) \cdot W^l)\| \propto \sqrt{d_i}.
\end{aligned}
\tag{1}
$$

Where d_i and d_k represent the degrees of node v_i and v_k respectively, and ψ signifies all L-length paths extending from v_i to v_k. $diag(\mathbb{1}_{\sigma_l})$ is the diagonal mask matrix representing the activation result and p^l represents the lth node along the specific path p. Assuming the aggregation process along the path is driven through random walks, $\| \sum_{p=1}^{\psi} \mathbb{E}(\prod_{l=L}^{0} \frac{1}{d_{p^l}} diag(\mathbb{1}_{\sigma_l}) \cdot W^l)\|$ in the above formula is constant and the degree of the node is directly related to its influence. From this, it can be concluded that the node influence level is directly proportional to the number of its neighbors. Analogous assertions can be made for other spatial-domain models. Building upon this understanding, by augmenting the quantity of node features pertaining to the minority class, we can enhance the inter-nodal influence within the minority class. This strategy mitigates the challenges arising from the minority class nodes not being predominant within the overall graph topology, which could otherwise result in the classification boundaries for the minority class being obscured by the predominant features of the majority class nodes.

Specifically, the SMOTE method proposed by Chawla et al. [11] is one of the most widely used resampling methods at present, which is realized by adding synthetic samples between the feature representations of minority class samples. However, the SMOTE method is not suitable for being used directly since the relation information is contained in the graphs, so we tune it based on the features of the graph and propose a novel adaptive graph oversampling module consisting of node and edge generators.

Regarding the node generator, we reformulate the origin SMOTE method by using the k-nearest neighbor nodes of the same class in the feature space to guide the model to interpolate new minority class nodes. In particular, if all nodes in the neighbors belong to the same or different class, we will not synthesize new nodes based on such nodes inspired by the practice in SMOTE−*Boardline* [31]. Here are two reasons: (1) Nodes belonging to the same class are close in the feature space. In this case, we use multiple neighbors to construct highly reliable similar nodes. (2) Selecting the minority nodes that are easily misclassified as a sample set makes the model focus on directly optimizing the classification boundary to improve the stability of the model. Explicitly, we firstly acquire the middle layer representations $h_v \in \mathbb{R}^d$ of the node $v \in V$ obtained from the general GNN backbone model, in which node feature and structure information are fused. The formula of h_v in lth layer is as follows:

$$
h_v^{(l)} = GNN^{(l)}(h_v^{(l-1)}, (h_u^{(l-1)} : u \in N(v))),
\tag{2}
$$

where $N(v)$ denotes the set of neighbors of node v. Since the same class of nodes usually forms a community in the feature space [32], we use k-nearest neighbor method to select the candidate neighbors for generating the embeddings of synthetic minority class nodes inspired by SMOTE−*boardline* [31]. We select node $v_i \in V$ from the minority class C^- with interventive probability function $\delta(l)$, where l refers to the current training epoch. Then we calculate its k-nearest neighbors KNN(v_i) from the whole training set T. Suppose the number of the same class examples among the above neighbors is k'. If $0 < k' < k$, v_i' is considered to be easy misclassification, which we denote as a danger node. For each danger sample v_i', we select the sample set P belonging to the same class with v_i' in its k-nearest neighbors KNN(v_i') and then calculate the differences D using Euclidean distance between v_i' and its neighbors in P. After that, we can generate $|P|$ new synthetic minority nodes $\hat{v}_i$ with the following interpolation:

$$
h_{\hat{v}_i}^{(l)} = h_{v_i'}^{(l)} + r_j \times D_j, \;\; s.t.\ 0 < r_j < 1.
\tag{3}
$$

Here, $r_j(1 \le j \le |P|)$ is a random number. These synthetic nodes obtained through the oversampling process make the weight of the minority class higher in the training process.

Next, we introduce the second part: edge generator. In order to effectively adapt GNN models, it is necessary to generate new edges for synthetic nodes. Hou et al. [13] propose that some quantitative indicators, such as smoothness and homophily, can measure the quality of information obtained from graph-based data based on information entropy theory. Within GNN models, the efficacy of information aggregation is intimately related to both the quantity and quality of information from neighbors. This perspective can also be employed to elucidate the performance enhancements observed in GNN models. According to [13], assuming context vector $c_{v_i}^l$ represents the true signal of node v_i, the following formula can be used to calculate the true aggregated information in the lth iteration:

$$
\begin{aligned}
\sum_{v_j \in N(v_j)} a_{i,j}^{l-1} \cdot c_{v_j}^{l-1} = &\sum_{v_j \in N(v_j)} \mathbb{1}(y_{v_i} = y_{v_j}) a_{i,j}^{l-1} \cdot c_{v_j}^{l-1} \\
&+ \sum_{v_j \in N(v_j)} (1 - \mathbb{1}(y_{v_i} = y_{v_j})) a_{i,j}^{l-1} \cdot c_{v_j}^{l-1}.
\end{aligned}
\tag{4}
$$

Where the item $\mathbb{1}(\cdot)$ serves as an indicator denoting the label consistency between v_i and v_j. $a_{i,j}$ refers to the ratio of information propagated from node v_j to v_i. Thus, in the context of node classification tasks, it is plausible to postulate that neighboring nodes sharing identical labels proffer beneficial information, while disparate labeled neighbors introduce adverse perturbations.

From the equation described above, it becomes evident that the propagation efficacy of the GNN models is related to the subsequent two factors. One is the quantity of information proffered by the surrounding nodes, and another is the quality of the information passed for a given task. To quantify these two aspects, we have defined two graph feature metrics for node v in normalized space $[0,1]^d$. Primarily, **Smoothness** can be discerned by calculating the similarity of features, represented as $(\sum_{v' \in N(v)} (h_v - h_{v'})^2)/(|N(v)| \cdot d)$. Secondarily, **Homophily** is determined by computing the proportion of interconnected nodes that share the same labels $(\sum_{e_{v,v'} \in E} \mathbb{1}(y_v = y_{v'}))/|N(v)|$.

Hence, within the scope of this work, the genesis of the edge generator endeavors to elevate the quality of the graph structure, striving to maximize both smoothness and homophily, thereby achieving an optimal information transmission within the topological framework. Based on it, we design an indicator to measure the quality of graph structure, so that we can obtain the optimum generated edges via the designed generator and then gain an augmented edge set E'. In this way, the optimized graph is more suitable for executing the GNN model. To compute the existing probabilities of latent edges related to synthetic nodes for each round, we leverage a multiplicative attention mechanism to get the coefficients in round k similar to [33]. It provides guidance on how to better use the surrounding information based on smoothness for context-surrounding as follows:

$$a_{i,j}^{(l)\prime} = \frac{exp(\sigma((W_1^{(l)} h_{v_i})^T \cdot (W_1^{(l)} h_{v_i} - W_2^{(l)} h_{v_j})))}{\sum_{v_m \in N(v_i)} exp(\sigma((W_1^{(l)} h_{v_i})^T \cdot (W_1^{(l)} h_{v_i} - W_2^{(l)} h_{v_m})))}. \tag{5}$$

Here, W_1 and W_2 are two learnable matrices. Unlike the traditional GAT [18] model, $(W_1^{(l)} h_{v_i} - W_2^{(l)} h_{v_j})$ is used here to calculate the feature smoothness. A large $a_{i,j}^{(l)\prime}$ indicates that edges should be generated between node v_i and v_j. Since a larger context difference means the features of a node and its neighbors are more dissimilar, the neighbors can contribute greater information gain. It can improve the effectiveness of feature propagation and avoid over-smoothing problems. Finally, we give the loss function L_{edge} for training the edge generator:

$$L_{edge} = \|A' - A\| + \|M' - M\|. \tag{6}$$

In the formula, $A \in \{0,1\}^{N \times N}$ refers to the factual adjacency matrix, and $M \in \{0,1\}^{N \times N}$ refers to the homophily matrix. $M_{i,j} = 1$ means node v_i and node v_j are connected in the training set and belong to the same class. M' is the predicted homophily matrix for labeled nodes, while A' represents the predicted adjacency matrix for existing nodes. In detail, the neighborhood provides both positive information and negative disturbance for a particular task. Aggregating neighbors' features simply cannot acquire the optimal results for graph embedding.

For this reason, we introduce the homophily matrix to amplify positive information and reduce negative disturbance. Poor homogeneity means that nodes with different labels tend to connect, where the surrounding context negatively interferes with the task. High homogeneity, however, implies that nodes within the same community tend to have connected edges and gain much positive information from neighbors. Some work [13] has proven that homophily is the key to improving the performance of GNN models based on the propagation–aggregation mechanism.

We hope that the generated edges can maintain the original graph's topological structure and resolve the class imbalance dilemma. With the help of the edge generator, we then offer the integral complement by adding the generated edges into the augmented edge set, which are determined by a threshold ϵ:

$$\hat{A}'_{i,j} = \begin{cases} 1, & a'_{i,j} \geq \epsilon, \\ 0, & a'_{i,j} < \epsilon. \end{cases} \tag{7}$$

Here $\hat{A}' \in \{0,1\}^{|P| \times (|P|+N)}$ is the adjacency matrix. It said the connection relationship between the new enhanced sampling nodes and other candidate nodes in the graph. Because we are considering an undirected graph, the reverse connection relationship can be directly obtained by the transpose matrix of $\hat{A}'$. Concatenating the adjacency matrix A of the original graph with the above two partial adjacency matrices while the overlapping part is only counted once to obtain the complete adjacency matrix $\hat{A}$ of the enhanced graph.

In the specific implementation, we can limit the range of candidate nodes that may have edges with synthetic nodes when calculating Eq. (7). For example, only one-hop neighbors of nodes in sampling set P of synthetic v_i can be used to filter whether edges are connected. According to target probability function $\delta(l)$, the minority class samples are re-oversampled in different epochs to confirm inclining to balance gradually. It will be explained in detail below. Then we adopt another GNN backbone block, appended by a linear layer for node classification as follows:

$$p_v = softmax(GNN(h_v, (h_{v'} : v' \in \hat{N}(v)))), \tag{8}$$

where $\hat{N}(\cdot)$ represents the augmented neighbor set corresponding to $\hat{A}$. p_v is the probability distribution on class labels for node v. After that, we give the loss function L_{node} for node classification.

$$L_{node} = -\sum_{v \in \hat{V}_l} \sum_{c=1}^{C} Y_v[c] \cdot \log(p_v[c]), \tag{9}$$

where $\hat{V}_l$ is the set of labeled nodes, Y_v is the one-hot vector that indicates the ground-truth labels of nodes. Finally, the Graph Classification Loss (GCL) is defined as:

$$L_{GCL} = L_{node} + \lambda \cdot L_{edge}. \tag{10}$$

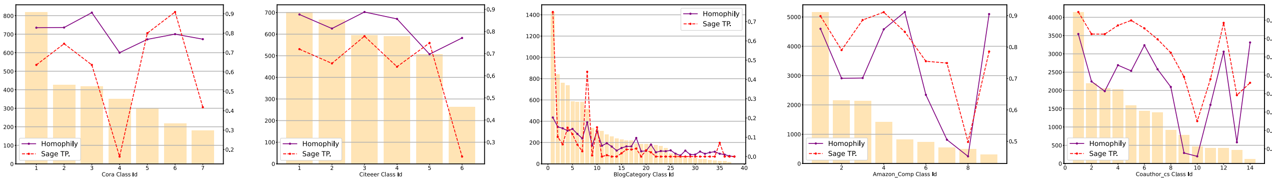
With the guide of labeled data, we can optimize the model via backpropagation and learn the embeddings of nodes.

3.2. Neighbor-based metric learning

In addition to the above oversampling method and loss function L_{GCL}, we also propose an imbalance-oriented method based on metric learning. Metric learning mainly adjusts the embedding positions of the selected minority classes in each mini-batch of training data to obtain better effectiveness of downstream tasks. It is realized with specific loss functions [26], and one of the typical selections is triplet loss, introduced by CRL [34] with hard mining. Define minor samples as anchors, and then adjust the distance between anchors and hard samples with high prediction scores on the wrong class via triplet loss pair. This resolvent can accelerate the convergence speed of the learning process and make it more effective with less training data for imbalanced data classification. Inspired by their ideas, for nodes in the minority class of a certain graph, we hope that they can avoid the dominant effect of majority classes when performing semi-supervised classification tasks.

In graph networks satisfying independent identically distributed, minority class nodes have less chance to meet the same class neighbors. The key consideration is that the aggregation mechanism of the GNN model makes nodes get much confusing information, leading to a decrease in performance or even over-smoothing issues. We observe this phenomenon on various datasets, and the details are shown in Fig. 3, where we display the GraphSAGE model performance True Positive (TP.) in different classes and the corresponding homophily score of each class set in five datasets. The histogram represents the points in each class from high to low. We can find that in these five datasets, when the homophily value decreases, the representation quality of the GNN model will be lessened, especially for tail classes in Fig. 3(c)(d)(e). However, the decline of the performance is not only influenced by homophily. For example, in Cora and Citeseer, the homophily value is maintained at a high level, but the effect of tail nodes decreases because the classifier cannot be trained effectively by limited samples. Here, we mainly consider the impact of homophily and its related information propagation process on the classification results.

As for the over-smoothing issue precipitated by the undue mixing of information and noise, an analytical approach founded upon the ratio between information content and noise can be employed, as elucidated in [32]. Within the context of node classification tasks, interactions from homogeneous classes can enhance information extraction, leading to increasingly convergent representations. It subsequently elevates the likelihood of nodes being categorized into identical classes. Conversely, interactions with nodes from heterogeneous classes introduce perturbations or noise into the system. For example, in the node classification task, interaction between nodes of the same class brings useful information, making their representations more similar to each other and increasing the probability of being classified into the same class. On the contrary, the contact of nodes from other classes brings noise. Obviously, the above issues are common in the minority class node. In order to solve the above problems, we use the novel triplet loss function to constrain the distances between minority class nodes and neighbors by drawing the same class neighbors and pushing away the different class neighbors. We explore graph sample mining strategy to enhance



(a) Cora (b) Citeseer (c) BlogCategory (d) Amazon Comp. (e) Coauthor CS

Fig. 3. Homophily score (purple) and True Positive (TP.) score (red) via GraphSAGE model of different classes in each dataset.

minority class manifold rectification by selectively "borrowing" different class samples from class decision boundary marginal (border) regions.

As for node distance function $d(h_1, h_2)$, where two node features $h_1, h_2 \in \mathbb{R}^d$, we compute the cosine distance between each node pair:

$$d(h_1, h_2) = 1 - \frac{h_1 \cdot h_2}{|h_1| \cdot |h_2|}, \tag{11}$$

where cosine distance is not affected by the absolute value of the node vector.

Define node samples in the minority classes as "anchor" samples. Then we start from each anchor's middle representation $h_{a,j}$ (we use the hidden representations of the final layer) of attribute label j and take its 1-hop neighbors as positive samples $h_{+,j}$ or negative samples $h_{-,j}$ to construct neighbor-based triplet loss pairs. The correlative Neighbor-based Triplet Loss function is defined as follows:

$$L_{NTL} = \frac{\sum_T max(0, m_j + d(h_{a,j}, h_{+,j}) - d(h_{a,j}, h_{-,j}))}{|T|}, \tag{12}$$

where $h_{+,j}$ and $h_{-,j}$ represent positive and negative samples with high prediction confidence in the neighbors of the central anchor, respectively. $|T|$ refers to the number of triplet pairs. We use the label of the anchor as a landmark. To evaluate the reliability, when the prediction scores of the anchor's surrounding samples on the landmark label exceed the credibility threshold $\alpha_+(l)$, we regard them as positive samples. And if the predicted scores of the samples on the landmark label are lower than the threshold $\alpha\_(l)$, we regard them as negative samples [35].

As shown in Fig. 2, we select minority class nodes as anchors and regularize the relative distances, which pulls the positive samples closer and pushes the negative samples further. The number of positive and negative samples to be selected is determined by the loss curriculum function $\alpha(l)$, where l refers to the current training epoch. Our proposed method can effectively deal with the over-smoothing problem of minority class nodes by pulling all the samples to the well-classified side.

3.3. Curriculum learning framework

To solve the imbalanced issue on the graph efficiently, we first explore the appropriate data generation and design a special edge generator based on homophily and smoothness. After that, we utilize the classification loss L_{GCL}. Next, we propose a special metric loss L_{NTL} according to the type relationship between the target node and its neighbor nodes with the help of pseudo positive/negative labels to improve the quality of the generated nodes. The final objective function is as follows:

$$\min_{\delta,\alpha,\lambda} L_{GCL} + \gamma \cdot L_{NTL}. \tag{13}$$

Inspired by the idea of curriculum learning [14], it is shown that learning from easy to hard significantly improves the generalization of the deep model. In order to leverage the training process, we design two opposite curriculum schedulers for loss functions:

The first is the curriculum probability scheduler $\delta(l)$, which helps define the sampling scale in a single batch and makes the distribution from imbalance to balance. This scheduler determines the sampling strategy for the proposed Graph Classification Loss (GCL) function, where L refers to expected total training epochs:

$$\delta(l) = \mu \cdot (1 - cos(\frac{l}{L} \cdot \frac{\pi}{2})). \tag{14}$$

Here, μ is the upper bound of sampling probability ranging from 0 to 1. The second is the curriculum loss scheduler $\alpha(l)$, which controls the thresholds for judging anchors, positives and negatives for the neighbor-based triplet loss (NTL). Especially for imbalanced data learning, we want the model first to learn a suitable feature representation to promote synthetic samples and benefit the classification. So that we hope the proposed model can assign more accurate positive/negative labels in the training process with the following scheduler:

$$\alpha_+(l) = (1 - \beta_+ \cdot cos(\frac{l}{L} \cdot \frac{\pi}{2})), \tag{15}$$

$$\alpha\_(l) = \beta\_ \cdot cos(\frac{l}{L} \cdot \frac{\pi}{2}). \tag{16}$$

Metric loss occupies a more significant proportion in the early stage of the training process. On the one hand, it plays the role of the "teacher" to guide the high-quality node embeddings and speed up the training process. On the other hand, it can help ensure a better quality of oversampling. While in the later stage, the system emphasizes classification loss more to learn the optimized classifier. Finally, the model is optimized by the objective in Eq. (13).

The optimization procedure of our GNN-CL is summarized in Algorithm 1. Line 2 is the calculation of two opposite curriculum schedules. After acquiring the middle layer node representations using the generic GNN model in Line 3, we select the minority class nodes with probability $\delta(l)$ and then generate synthetic nodes with the proposed interpolation method (Line 4). Subsequently, the augmented edge set is obtained in Line 5. We use optimized topology to calculate Graph Classification Loss in Line 6. In another part introduced in Lines 7–8, we use minority anchor samples and filtered positive/negative samples to calculate Neighbor-based Triplet Loss. Finally, the GNN-CL framework completes parameter updating on imbalanced datasets in Lines 9–10.

Herein, we delve into a comprehensive analysis of the time complexity associated with the proposed method. Viewed holistically, the curriculum learning framework solely involves probabilistic alterations and does not introduce any additional time overhead. The primary temporal expenditures of the algorithm are distributed across the following components: Initially, we focus on the process of employing the GNN network to compute feature representations for all nodes. To facilitate computation, we contemplate the canonical aggregation-based GNN computational paradigm. For a single layer of the GNN model that accomplishes the feature transformation from F dimensions to F' dimensions, its computational procedure predominantly encompasses a matrix transformation for all nodes V. The time complexity can be denoted as $O(|V| \times |F| \times |F'|)$. Subsequently, it becomes requisite to execute an information propagation procedure across all edges in the graph. The computational time complexity for this segment is represented as $O(|E| \times |F'|)$. Further considering Graph Oversampling, the generation of nodes entails calculating the K-nearest neighbors for a subset of minority nodes denoted as V^-. This process involves traversing all nodes, resulting in a time complexity of $O(|V| \times |V^-|)$. Upon completion, interpolation operations are performed using the selected nodes. The time cost of this operation can be negligible compared to the preceding item. Furthermore, when contemplating the edge generator, computations involve the newly generated nodes in conjunction with a limited set of candidate nodes within the graph. The time overhead for this operation is less than $O(|V| \times |V^{-}|)$. Regarding the Neighbor-based Metric Learning, one selects all the minority anchors and conducts triplet distance calculations with their neighbors. Given that this only involves a subset of nodes, the computational overhead is also less than $O(|V| \times |V^{-}|)$. In summary, the total computational complexity of the proposed GNN-CL is given by $O(|V| \times |F| \times |F'|) + O(|E| \times |F'|) + O(|V| \times |V^{-}|)$. This does not exceed the complexities of other classical methods, such as GraphSmote [12] and Graphmixup [27].

Algorithm 1: Training procedure of GNN-CL.

```
Input: training graph data $G = (V, E, F)$ and label set $Y$,
       initialized parameters of GNN $\theta$, learning rate $\alpha$, train
       epoches $L$.
Output: trained parameters of GNN $\theta^*$.
for $l = 1, \ldots, L$ do
    Calculate the curriculum probability scheduler $\delta(l)$ and the
      curriculum loss scheduler $\alpha_+(l)$ and $\alpha_-(l)$;
    Obtain the middle layer node representations $h_v$;
    Select minority class nodes with probability $\delta(l)$ and
      generate synthetic nodes $h_{\hat{v}}$;
    Obtain the augmented adjacency matrix $\hat{A}$;
    Compute Graph Classification Loss $L_{GCL}$;
    Select positive samples $h_{+,j}$ or negative samples $h_{-,j}$ with
      probability $\alpha_+(l)$ and $\alpha_-(l)$ being neighbors with minority
      class anchors $h_{a,j}$;
    Compute Neighbor-based Triplet Loss $L_{NTL}$;
    Compute objective loss: $L(\theta) = L_{GCL} + \gamma \cdot L_{NTL}$;
    Update $\theta \leftarrow \theta - \alpha \nabla_\theta L(\theta)$.
end
```

4. Experiments

In this section, we design several experiments on five real-world datasets to test the effect of GNN-CL. We resolve three problems in the following:

- **RQ1:** How is the performance of GNN-CL compared with the state-of-the-art imbalanced classification methods?
- **RQ2:** How do the L_{GCL} loss and L_{NTL} loss affect the classifier performance?
- **RQ3:** How do different factors (imbalanced ratio, oversampling scale, base model, etc.) affect the results of GNN-CL?

4.1. Experimental settings

4.1.1. Datasets

For our experiments, we choose 5 widely used node classification datasets belonging to 4 types for experimental comparison, including two well-known citation graphs: Citeseer and Cora [14], co-purchase graph: Amazon computers [13], co-authorship graph: Coauthor CS [13] and co-authorship graph: BlogCatalog [36]. To describe the dataset in detail, we list the statistics of the datasets in Table 1.

(1) We first use the two well-known citation network datasets: Cora and Citeseer. Edges in these networks represent the citation relationship between two papers (undirected), node features are the bag-of-words embeddings of the papers and labels are the fields of papers. Among them, Cora contains 140 labeled training nodes with balanced class distributions, so the factor $imbalance\_ratio$ is used to disequilibrate the data by downsampling half of random classes. For each minority class, the number is 20×$imbalance\_ratio$. Likewise, there is a mild class imbalance problem in Citeseer's training set.

(2) Amazon computer is built from fragments in the Amazon co-purchase graph. The nodes in the graph represent products, and their features are obtained through the bag-of-words model of consumers' comments. The edges represent that the products are purchased at the same time, and the category label is obtained by the category of the product. It contains 9 types of samples and the head majority class is 16 times more than the tail minority class.

(3) Coauthor CS is a co-authorship graph based on the Microsoft academic graph. The nodes symbolize the authors and the edges represent the co-authorship relationships. These features come from the keywords of each author's paper. At the same time, 14 different labels indicate the active fields of study for each author suffering from a large imbalance problem.

(4) BlogCatalog is also an analogous co-authorship graph based on the Microsoft academic graph. The features originate from the keywords for each author's paper. Classes in this dataset meet a genuine imbalanced distribution, with 14 classes smaller than 100 and 8 classes larger than 500.

4.1.2. Compared methods

In the following experiments, we compare GNN-CL with representative and state-of-the-art approaches for handling imbalanced class distribution problem, which consists of the conventional methods, Oversampling and Reweighting, deep learning method Deep OverSampling, graph neural network method GraphSMOTE and the Graphmixup model based on semantic feature space. We describe these baselines in detail as follows:

- **Oversampling** Oversampling is a classical method that improves the classifier's performance through the repetition of minority classes. In the implementation, we duplicate n_s minority samples and edges connected with them on the graph.
- **Reweighting** This is a kind of method to adjust the category weight of loss function, primarily by increasing the importance of a few categories in supervision information.
- **Deep OverSampling** To counteract the class imbalance problem, this method utilizes a synthetic embedding target in the deep feature space, which is sampled from the linear subspace of in-class neighbors.
- **GraphSMOTE** GraphSMOTE [12] synthesizes similar node embeddings in the feature space to assure genuineness. An edge generator is also trained simultaneously to model the relationship information and provide it for new samples.
- **Graphmixup** Graphmixup [27] transfers the mixup of embeddings from the feature space to the semantic space, and proposes a reinforcement mixup mechanism to adaptively determine the number of newly generated samples of those minority classes.

In order to verify the effectiveness of each part of our proposed method, GNN-CL and its ablation study models are tested:

- **GNN-CL** Our proposed graph neural network with curriculum learning on Graph Classification Loss and Neighbor-based Triplet Loss.
- **GNN-CL$_O$** It removes the oversampling strategy from the proposed model so that synthetic nodes and corresponding edges will not be generated.
- **GNN-CL$_M$** It removes the metric loss part from the proposed model and ignores the regularization between neighbors.
- **GNN-CL$_C$** It removes the curriculum learning mechanism from the proposed model, and the ratio of two losses is determined by the fixed optimal parameters.

Table 1
Statistics of the datasets.

Dataset	♯Node	♯Edge	♯Training	♯Validation	♯Test	Imbalance ratio(M:1)
Cora	2708	10 556	140	140	2428	–
Citeseer	3327	9228	831	831	1663	2.65
BlogCategory	10 312	667 966	2561	2561	5146	355.00
Amazon Comp	13 752	287 209	3434	3434	6875	16.74
Coauthor CS	18 333	163 788	4579	4579	9164	35.65

Table 2
Experiment results for the imbalanced node classification task. (bold: best, underline: runner-up).

Dataset Model	**Cora**		**Citeseer**		**BlogCategory**		**Amazon Comp.**		**Coauthor CS**	
Metric	cmA.	AUC-ROC	cmA.	AUC-ROC	cmA.	AUC-ROC	cmA.	AUC-ROC	cmA.	AUC-ROC
Origin	0.655±0.003	0.902±0.005	0.616±0.009	0.883±0.002	0.062±0.005	0.569±0.009	0.794±0.013	0.980±0.002	0.854±0.003	0.977±0.002
Oversampling	0.645±0.025	0.900±0.012	0.619±0.011	0.885±0.005	0.056±0.002	0.563±0.018	0.798±0.002	0.980±0.001	0.853±0.006	0.985±0.003
Reweighting	0.651±0.019	0.909±0.009	0.625±0.004	0.886±0.001	0.058±0.003	0.561±0.017	0.791±0.007	0.978±0.001	0.856±0.004	0.980±0.002
DOS.	0.651±0.012	0.901±0.006	0.595±0.015	0.875±0.005	0.056±0.001	0.556±0.011	0.781±0.022	0.977±0.003	0.850±0.004	0.976±0.002
GraphSMOTE	0.723±0.015	0.915±0.007	0.593±0.009	0.870±0.007	0.058±0.008	0.558±0.005	0.801±0.004	0.978±0.001	0.845±0.006	0.976±0.002
Graphmixup	0.740±0.007	0.926± 0.006	0.634±0.004	0.880± 0.005	**0.100±0.006**	**0.629±0.016**	0.735±0.006	0.971±0.002	0.842±0.027	0.966±0.032
GNN-CL	**0.742±0.006**	**0.936±0.002**	**0.637±0.005**	**0.889±0.005**	0.067±0.006	0.575±0.010	**0.806±0.005**	**0.980±0.001**	**0.869±0.006**	**0.989±0.001**
GNN-CL$_O$	0.669±0.018	0.911±0.007	0.627±0.011	0.884±0.006	0.052±0.001	0.561±0.011	0.799±0.007	0.979±0.001	0.862±0.007	0.988±0.001
GNN-CL$_M$	0.725±0.016	0.935±0.004	0.620±0.006	0.883±0.002	0.055±0.004	0.569±0.004	0.798±0.001	0.979±0.001	0.863±0.003	0.989±0.001
GNN-CL$_C$	0.710±0.009	0.920±0.010	0.627±0.006	0.881±0.001	0.059±0.003	0.565±0.007	0.791±0.004	0.977±0.002	0.858±0.010	0.985±0.002

4.1.3. Metrics

In order to comprehensively measure the effect of our proposed model, we adopt three commonly used criterias of the imbalance classification task: class-balanced mean accuracy (cmA) and average AUR-ROC score. cmA is computed on all testing examples at once. Following the standard profile, we apply the class-balanced accuracy as the average recall obtained in each class. It can be formulated as follows:

$$cmA = \frac{\sum_{i=1}^{|C|} \frac{TP_i}{TP_i+FN_i}}{|C|}. \tag{17}$$

While the AUC-ROC score indicates the probability that the predicted positive case is ranked higher than the case from other classes.

4.1.4. Settings and hyper-parameters

We initialize the parameters randomly for all methods in the experiment and use Adam to optimize the model with a maximum of 2000 epochs (adopting early stopping with patience of 100). In practice, we implement them with pytorch1.2[1] to train model parameters and also use mini-batch gradient descent, which divides the training data into several batches and updates parameters per batch. The source code and dataset can be found on the website.[2]

The GNN backbone model used in concrete realization can be varied. In the experiment, we default to choosing the GraphSAGE model because it can flexibly adapt to various topology structures, and the idea based on spatial-domain is more suitable for our implementation. The learning rate in models involved in comparison is initialized to 0.001 and the weight decay is set to 0.0005. Two hyper-parameters λ and γ are set to 0.002 and 1.0 by default, according to the actual function values in the function. As for parameters related to our graph oversampling, the oversampling parameter μ is set to 1.0. The edge generation threshold ϵ and the hyperparameter of neighbors k in KNN algorithm are defaulted to 0.5 and 7 based on experimental results. While the boundary parameter β_+ and β_- of pseudo positive/negative labels in L_{NTL} are set to 0.6 and 0.1, respectively. m in neighbor-based triplet loss is empirically set to 0.5. In addition, several sensitivity experiments are carried out to explore the proper range of parameters.

4.2. Overall performance (RQ1)

Here we compare the effectiveness of different methods by the imbalanced semi-supervised node classification task on various datasets. In order to eliminate variance, we repeat the process 5 times and report the averaged cmA. and AUC-ROC in Table 2. From the results, it can be concluded that our proposed GNN-CL model consistently achieves either the best or the second-best performance results across all datasets. It has around 1−3% performance gain over the best baseline in general, which indicates that oversampling and metric learning modules alleviate the adverse effects of long-tail distribution.

To commence, let us deliberate on the conventional imbalanced classification techniques. Oversampling and Deep OverSampling merely engage in rudimentary replication of node features, neglecting the unique information propagation characteristics inherent to graph structures. Therefore, they yield adverse outcomes on the Cora and BlogCategory datasets. Reweighting also merely adjusts the weight of the loss function, yielding no enhancement in outcomes when compared with the Origin. While these generic methods prove efficacious within Euclidean fields, they overlook the topological characteristic of graphs, thereby failing to confer training benefits.

The graph-based GraphSMOTE method has certain improvements in some datasets with small imbalance ratios, such as Cora and Amazon comp., but performs poorly in other datasets. Its primary limitation resides in the elementary expansion of nodes and edges within the graph without sufficiently contemplating the quality of information propagation. Graphmixup melds node features at the semantic level while devising a methodology for edge structure augmentation. Its efficacy is particularly pronounced on datasets with a large imbalance ratio, such as BlogCategory, attributable to its consideration of various semantic spaces and the implementation of reinforcement training strategies. But in other cases, the effect is inferior to our method. Nevertheless, it can also be seen that when faced with a large imbalance ratio, our oversampling approach exhibits a certain disparity compared to Graphmixup. This discrepancy primarily emanates from the paucity of long-tail samples because relying on node neighbors for interpolation fails to guarantee the efficacy of feature generation, resulting in a lack of effective training for classification boundaries. To some extent, Graphmixup uses the semantic level to avoid this shortcoming.

The ablation experiment shows that the performance of removing the oversampling module is significantly weakened. Across the various datasets, the cmA. index observes respective reductions of $\{7.3\%, 1.0\%, 1.5\%, 0.7\%, 0.7\%\}$. Concurrently, the variance of the results

[1] https://pytorch.org/

[2] https://github.com/seanlxh/GNN-CL

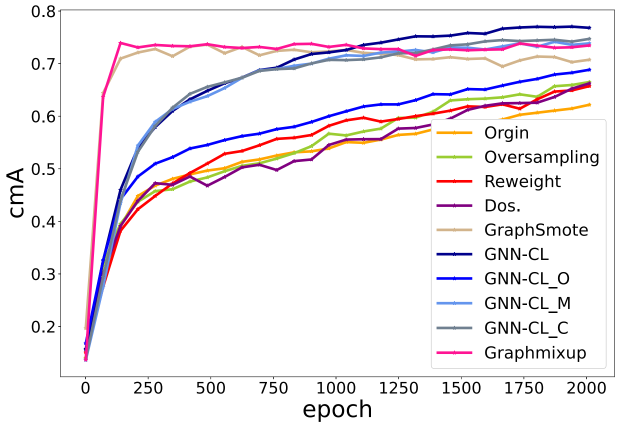


Fig. 4. Test results (cmA) of each method during training process on Cora dataset.

Table 3
Experiment results on different imbalance ratio. (bold: best, underline: runner-up).

	Imbalance Ratio				
Methods	0.1	0.3	0.5	0.7	0.9
Origin	0.354	0.554	0.634	0.677	0.711
Oversampling	0.442	0.559	0.659	0.676	0.713
Reweighting	0.464	0.579	0.661	0.699	0.702
DOS.	0.474	0.587	0.644	0.625	0.633
GraphSMOTE	0.591	0.708	0.723	0.731	0.755
Graphmixup	0.597	0.712	0.738	0.748	0.751
GNN-CL	0.595	0.712	**0.745**	**0.757**	**0.759**
GNN-CL$_O$	0.473	0.582	0.670	0.714	0.721
GNN-CL$_M$	**0.601**	**0.724**	0.740	0.745	0.746
GNN-CL$_C$	0.542	0.683	0.701	0.734	0.745

exhibited amplification, underscoring an augmented instability of the algorithm upon the removal of oversampling. For GNN-CL$_M$, given that metric learning mainly assists the classification task by improving the quality of node representations, the algorithm performance experiences a decrease after removing the relevant components. While there still have some good cases when omitting metric loss, which shows that in datasets with a lower imbalance ratio, the impact of Loss$_{NTL}$ is not conspicuously manifested. To verify the effectiveness of the curriculum learning framework, we conduct a detailed analysis on GNN-CL$_C$. The outcomes indicate that no enhancement in the training process is observed after removing the curriculum learning mechanism and directly deploying both the Adaptive Graph Oversampling and Neighbor-based Metric Learning modules. Indeed, on datasets such as Cora and Coauthor, the performance is even inferior compared to merely employing one of the aforementioned mechanisms. This is because at the beginning of the training process, emphasis should be placed on using metric loss to improve the quality of soft features and constrain the span of minority classes in the feature space. As the training process progresses, the model dynamically adjusts the label ratio based on superior sample features, thereby bolstering the classifier's discriminative capability for imbalanced samples.

4.3. Process analyses (RQ2)

To further illustrate the effectiveness of GNN-CL, we draw a group of training process curves to make intuitive comparisons. Fig. 4 reveals the cmA. scores during the training process of each comparison method, where all models and variants are plotted with different colors. Based on this visualization, we can see that the proposed GNN-CL model achieves the best classification results along with a stable training procedure. Other traditional sampling methods perform better than the origin method, showing that the general sampling methods are also applicable to the graph.

However, compared with the special graph sampling method, it has obvious disadvantages in performance, which shows that GraphSMOTE, Graphmixup and GNN-CL have good effects when applied in imbalanced classification situations on graphs. Note that GraphSMOTE requires fewer rounds to reach its peak and then begins to decrease, proving that the generation of nodes and edges is obviously helpful early in the training process. However since the oversampling process remains unchanged, the problem of overfitting soon arises and the models cannot obtain the best result. Nevertheless, Graphmixup also achieves relatively good results quickly because of the pre-training process.

Our GNN-CL model prioritizes the feature embeddings' quality at the beginning of training through the curriculum learning mechanism and then turns to incrementally generating high-confidence nodes and edges. Compared with baseline methods, it not only ensures the speed of training but also continuously and significantly improves the effect. Ablation models GNN-CL$_M$ and GNN-CL$_C$ also have comparatively good results, but due to the lack of metric loss or curriculum learning, the effect falls behind significantly in the later stage of training. It is noteworthy that, as both GNN-CL$_C$ and Graphmixup have removed the curriculum mechanism, they can swiftly achieve optimal validation results during training. However, their test results after fitting are not as commendable as those achieved by the proposed GNN-CL, and they exhibit the phenomenon of overfitting. Certainly, this also reflects the potential room for enhancement in the fitting speed of our curriculum learning strategy. Such limitations are inherently attributable to the mechanism itself. In future endeavors, contemplating improvements in areas like dynamic learning rate might augment training efficiency.

4.4. In-depth analysis (RQ3)

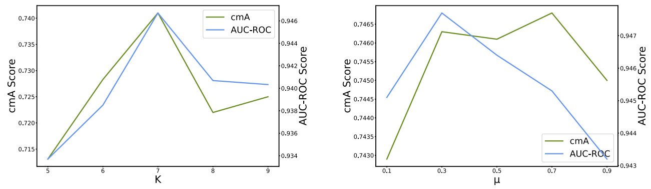
4.4.1. Study on imbalance ratio

The classification performances of all above models under different imbalance ratios are listed in Table 3. The severity of the imbalance problem is inversely proportional to the imbalance ratio value. We can see that both graph-based sampling and mixup methods are significantly effective, especially when the imbalance ratio value is low. For example, when the imbalance ratio $= 0.1$, there exists an increase of more than 20% compared with Origin. But in this extremity, there is no obvious distinction among Graphmixup, GraphSMOTE and GNN-CL, indicating that due to the serious imbalance problem, the neighbor-based triplet loss is not fully used. Overall, when observing GNN-CL$_O$ and GNN-CL$_M$, it can be seen that sampling plays a greater role than metric learning on the Cora dataset. When the imbalance ratio $= 0.9$, the dataset is almost in balance, so the sampling methods have little significance.

4.4.2. Study on parameters in proposed module

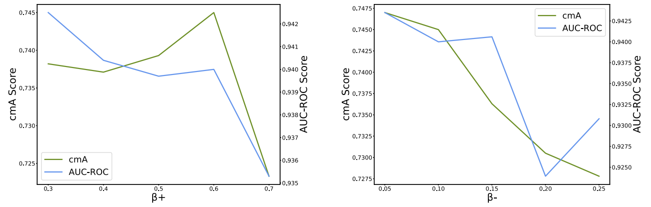
In this section, we verify the impact of different parameters for two proposed modules graph oversampling and metric learning on the results, shown in Figs. 5 and 6.

(1) Firstly, we count the performance indicators of the parameters related to the graph oversampling scale. The number k in the k-nearest neighbor algorithm determines the range of synthetic node generation candidates, and μ controls the upper bound on the probability of the graph oversampling scale. One can see from Fig. 5 that generating more synthetic nodes within a certain limit helps to improve the performance of the model. Furthermore, Cora has a relatively mild imbalance problem, so the quality of the generated nodes is high.



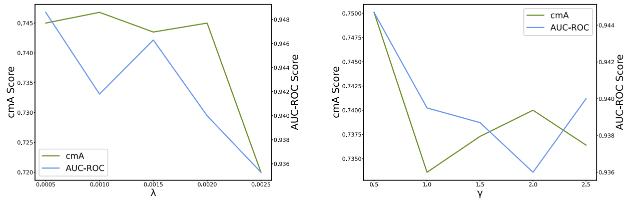
(a) Neighbor number k in KNN. (b) Oversampling scale μ.

Fig. 5. Experiment results on graph oversampling parameters.



(a) Threshold of positive label. (b) Threshold of negative label.

Fig. 6. Experiment results on metric learning parameters.



(a) Parameter in graph classification loss. (b) Parameter in final objective function.

Fig. 7. Experiment results on other hyper-parameters.

(2) We also test the parameter β used to judge pseudo positive/negative labels in the metric learning module. β_+ and β_- control the possibility of generating positive and negative sample labels. From Fig. 6, it can be seen that too many pseudo positive/negative labels are not conducive to the clear classification boundary. Only with appropriate parameters can the best model performance be achieved.

4.4.3. Study on parameters in loss function

In this section, we do sensitivity analysis to some essential parameters of the loss function in GNN-CL and Fig. 7 shows the test score curves on Cora.

(1) We first test the effect of the ratio of Graph Classification Loss and Edge Generator Loss L_{edge}, shown in Fig. 7(a). Performance initially holds steadily as the proportion of edge generator loss increases and then shows a continuous decrease. The optimal performance is obtained when $\lambda < 0.0015$.
(2) We also investigate the effect of the ratio of classification loss and metric loss reported in Fig. 7(b). Based on the results, we can find that limiting γ to a smaller range works best. In the future, we can finetune the proportion of metric loss more finely.

4.4.4. Study on backbone model

As shown in Table 4, we attempt to apply the proposed GNN-CL method to different backbone models to verify its generality. We select a classical spectral-domain GCN model and test it on two classic datasets Cora and Citeseer for verification. When all baseline methods employ GCN as the backbone model, adverse effects emerge to varying extents in comparison with the Origin. This can be attributed to the fact that spectral-based methods are particularly sensitive to the quality of graph topological structures, especially for GraphSMOTE and Graphmixup, which are specially designed for imbalanced graph data. Their ability to solve the imbalance problem is merely similar to that of the other generic methods. Whereas our proposed GNN-CL model has consistent applicability on test datasets, the performance of GNN-CL is 2–4% higher than the optimal baseline. Meanwhile, it can be seen from the ablation models GNN-CL$_O$ and GNN-CL$_M$ that both the oversampling and metric learning modules have their own values. The empirical outcomes from the GNN-CL$_C$ model indicate the universality of our proposed curriculum learning framework across diverse architectures.

5. Conclusion

In this work, our primary emphasis centers on addressing the imbalance challenge inherent in intricate node classification tasks. We introduce a pioneering graph neural network framework with curriculum learning (GNN-CL). Grounded on this paradigm, adaptive graph oversampling and neighbor-based metric learning are proposed, driven by the dual objectives of adjusting class proportions and optimizing classification boundaries. Extensive experiments prove that the proposed GNN-CL outperforms state-of-the-art methods across various domains with a consistent level of performance. Key merits of our method include the provision of a superior-quality graph oversampling technique based on sufficient theoretical analysis, the capability to mitigate imbalanced classification problems without succumbing to overfitting, strong generalization and support for Plug-and-Play, etc. Nevertheless, the model also has inevitable limitations, such as oversampling based on information propagation not explicitly incorporating semantic information, and the convergence speed of the model has not yet reached its optimal level. In the future, we will explore alternative dynamic graph sampling strategies and design an interpretable end-to-end learning framework.

CRediT authorship contribution statement

Xiaohe Li: Conceptualization, Data curation, Formal analysis, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. **Zide Fan:** Methodology, Project administration, Supervision. **Feilong Huang:** Software, Validation, Visualization. **Xuming Hu:** Data curation, Resources. **Yawen Deng:** Funding acquisition, Resources. **Lei Wang:** Project administration. **Xinyu Zhao:** Project administration, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the link to my data/code in the manuscript.

Acknowledgments

This work has been supported by the Strategic Priority Research Program of Chinese Academy of Sciences (Grant No. XDA0310502), Future Star of Aerospace Information Research Institute, Chinese Academy of Sciences, China (Grant No. E3Z10701) and Key Laboratory Fund of the Chinese Academy of Sciences (Grant No. CXJJ215032).

Table 4
Experiment results using GCN as backbone model. (bold: best, underline: runner-up).

Metrics	Cora		Citeseer	
	mcA	AUC-ROC	mcA	AUC-ROC
Origin	0.681 ± 0.024	0.907 ± 0.005	0.620 ± 0.020	0.859 ± 0.010
Oversampling	0.663 ± 0.029	0.914 ± 0.012	0.621 ± 0.006	0.864 ± 0.002
Reweighting	0.675 ± 0.005	0.904 ± 0.004	0.636 ± 0.008	0.867 ± 0.004
DOS.	0.689 ± 0.010	0.908 ± 0.008	0.609 ± 0.011	0.852 ± 0.006
GraphSMOTE	0.673 ± 0.008	0.905 ± 0.002	0.605 ± 0.009	0.852 ± 0.002
Graphmixup	0.692 ± 0.016	0.908 ± 0.010	0.608 ± 0.014	0.861 ± 0.007
GNN-CL	**0.703 ± 0.007**	0.911 ± 0.007	**0.646 ± 0.004**	**0.881 ± 0.004**
GNN-CL$_O$	0.686 ± 0.021	**0.916 ± 0.008**	0.636 ± 0.005	0.873 ± 0.002
GNN-CL$_M$	0.693 ± 0.003	0.909 ± 0.006	0.618 ± 0.002	0.858 ± 0.005
GNN-CL$_C$	0.698 ± 0.001	0.910 ± 0.004	0.625 ± 0.005	0.857 ± 0.004

References

[1] J. Scott, P.J. Carrington, The SAGE Handbook of Social Network Analysis, The SAGE handbook of social network analysis, 2011.

[2] J. Zhang, X. Liu, X. Zhou, X. Chu, Leveraging graph neural networks for point-of-interest recommendations, Neurocomputing 462 (2021) 1–13.

[3] Z. Liang, J. Du, Y. Shao, H. Ji, Gated graph neural attention networks for abstractive summarization, Neurocomputing 431 (2020).

[4] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016, arXiv preprint arXiv:1609.02907.

[5] K. Sun, Z. Lin, Z. Zhu, Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes, in: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34, no. 04, 2020, pp. 5892–5899.

[6] M. Zhang, Y. Chen, Link prediction based on graph neural networks, in: Advances in Neural Information Processing Systems, vol.31, 2018.

[7] H. Wang, F. Zhang, M. Hou, X. Xie, M. Guo, Q. Liu, Shine: Signed heterogeneous information network embedding for sentiment link prediction, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018, pp. 592–600.

[8] Q. Zhang, R. Wang, J. Yang, L. Xue, Structural context-based knowledge graph embedding for link prediction, Neurocomputing 470 (2022) 109–120.

[9] T. Wang, J. Wu, Z. Zhang, W. Zhou, S. Liu, Multi-scale graph attention subspace clustering network, Neurocomputing 459 (3) (2021).

[10] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Advances in neural information processing systems, vol.30, 2017.

[11] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, AI Access Found. (1) (2002).

[12] T. Zhao, X. Zhang, S. Wang, GraphSMOTE: Imbalanced Node Classification on Graphs with Graph Neural Networks, ACM, 2021.

[13] Y. Hou, J. Zhang, J. Cheng, K. Ma, R. Ma, H. Chen, M.C. Yang, Measuring and improving the use of graph information in graph neural networks, in: International Conference on Learning Representations, 2020.

[14] Y. Wang, W. Gan, J. Yang, W. Wu, J. Yan, Dynamic curriculum learning for imbalanced data classification, in: 2019 IEEE/CVF International Conference on Computer Vision, ICCV, 2020.

[15] X. Wang, Y. Chen, W. Zhu, A survey on curriculum learning, IEEE Trans. Pattern Anal. Mach. Intell. PP (99) (2021).

[16] J. Bruna, W. Zaremba, A. Szlam, Y. Lecun, Spectral networks and locally connected networks on graphs, in: International Conference on Learning Representations, 2013.

[17] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Advances in Neural Information Processing Systems, vol.29, 2016.

[18] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, et al., Graph attent. netw., stat 1050 (20) (2017) 10–48550.

[19] N. Japkowicz, S. Stephen, The Class Imbalance Problem: A Systematic Study, IOS Press, 2002.

[20] G.M. Weiss, Mining with rarity: A unifying framework, Acm Sigkdd Explor. Newslett. 6 (1) (2004) 7–19.

[21] C. Drummond, R.C. Holte, C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling, in: Proc of the Icml Workshop on Learning from Imbalanced Datasets II, 2003.

[22] M.T. Kai, A Comparative Study of Cost-Sensitive Boosting Algorithms, Morgan Kaufmann Publishers Inc, 2000.

[23] H. Yu, C. Sun, X. Yang, W. Yang, J. Shen, Y. Qi, Odoc-elm: optimal decision outputs compensation-based extreme learning machine for classifying imbalanced data, Knowl.-Based Syst. 92 (2016) 55–70.

[24] X.Y. Liu, J. Wu, Z.H. Zhou, Exploratory undersampling for class-imbalance learning, IEEE Trans. Syst. Man Cybern. B 39 (2) (2009) 539–550.

[25] N.V. Chawla, A. Lazarevic, L.O. Hall, K.W. Bowyer, SMOTEBoost: Improving prediction of the minority class in boosting, in: Knowledge Discovery in Databases: PKDD 2003: 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Cavtat-Dubrovnik, Croatia, September 22-26, 2003. Proceedings 7, Springer, 2003, pp. 107–119.

[26] H. Oh Song, Y. Xiang, S. Jegelka, S. Savarese, Deep metric learning via lifted structured feature embedding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4004–4012.

[27] L. Wu, J. Xia, Z. Gao, H. Lin, C. Tan, S.Z. Li, Graphmixup: Improving class-imbalanced node classification by reinforcement mixup and self-supervised context prediction, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2022, pp. 519–535.

[28] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, Graph neural networks: A review of methods and applications, AI Open 1 (2020) 57–81.

[29] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, R.P. Adams, Convolutional Networks on Graphs for Learning Molecular Fingerprints, MIT Press, 2015.

[30] X. Tang, H. Yao, Y. Sun, Y. Wang, J. Tang, C. Aggarwal, P. Mitra, S. Wang, Investigating and mitigating degree-related biases in graph convolutional networks, 2020.

[31] H. Hui, W.Y. Wang, B.H. Mao, Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning, Lecture Notes in Comput. Sci. (2005).

[32] D. Chen, Y. Lin, W. Li, P. Li, J. Zhou, X. Sun, Measuring and relieving the over-smoothing problem for graph neural networks from the topological view, in: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34, no. 04, 2020, pp. 3438–3445.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, vol.30, 2017.

[34] Q. Dong, S. Gong, X. Zhu, Imbalanced deep learning by minority class incremental rectification, IEEE Trans. Pattern Anal. Mach. Intell. (2018) 1.

[35] M. Lu, Z. Huang, Y. Zhao, Z. Tian, Y. Liu, D. Li, DaMSTF: Domain adversarial learning enhanced meta self-training for domain adaptation, in: The 61st Annual Meeting of the Association for Computational Linguistics, 2023.

[36] P. Sen, G. Namata, M. Bilgic, L. Getoor, T. Eliassi-Rad, Collective classification in network data, Ai Mag. (2008).

Xiaohe Li received the M.Sc degree from School of Software, Tsinghua University in 2021. He is currently an assistant research fellow at Aerospace Information Research Institute, Chinese Academy of Sciences. His research interests are focused on graph neural network, process mining and collaborative awareness.

Zide Fan received the Ph.D degree from Central South University in 2016. He is currently an associate research fellow at Aerospace Information Research Institute, Chinese Academy of Sciences. His research interests are focused on geographic data mining, information fusion.

Feikong Huang received the B.S. degree from Nanjing University of Science and Technology, China, in 2022. He is currently pursuing the master's degree in University of Chinese Academy of Sciences. His main research interests include graph data, data mining and machine learning.

Xuming Hu received the B.E. degree in Computer Science and Technology, Dalian University of Technology. He is working towards the Ph.D. degree at Tsinghua University. His research interests include natural language processing and information extraction.

Yawen Deng received the M.Sc degree from Beijing Institute of Technology in 2019. She is currently an assistant research fellow at Aerospace Information Research Institute, Chinese Academy of Sciences. Her research interests are focused on digital twin and multi-agent game.

Lei Wang received the Ph.D. degree from Institute of Electronics, Chinese Academy of Sciences. He is currently an research fellow at Aerospace Information Research Institute, Chinese Academy of Sciences. His research interests are focused on data mining and information fusion.

Xinyu Zhao received the Ph.D. degree from National University of Defense Technology. He is currently an research fellow at Aerospace Information Research Institute, Chinese Academy of Sciences. His research interests are focused on data mining, object detection and informatics.