Electronic Research Archive

http://www.aimspress.com/journal/era

ERA, 34(3): 1742–1784.
DOI: 10.3934/era.2026079
Received: 10 November 2025
Revised: 21 January 2026
Accepted: 26 January 2026
Published: 27 February 2026

Survey

Imbalanced node classification on graphs with graph neural networks: A survey

Changhai Wang[1], **Zhe Huang**[1], **Yuwei Xu**[2,3] **and Yaoli Xu**[1,*]

[1] Software Engineering College, Zhengzhou University of Light Industry, Zhengzhou 450000, China
[2] School of Cyber Science and Engineering, Southeast University, Nanjing 210000, China
[3] Purple Mountain Laboratories for Network and Communication Security, Nanjing 210000, China

* **Correspondence:** Email: yaolixu@zzuli.edu.cn.

Abstract: Imbalanced node classification is an important research topic in graph learning. With the rise of deep learning, powerful models like graph neural networks (GNNs) have been widely used in node classification tasks and achieved promising performance. However, real-world graphs are usually imbalanced, characterized by some classes having an adequate amount of data while others lack data. This imbalance presents the suboptimal classification performance of the model. Therefore, research on the GNNs-based imbalanced node classification is crucial. This article aims to systematically summarize the development status of GNNs in imbalanced node classification. First, the related concepts of GNNs and imbalanced node classification are introduced to establish a solid foundation for readers. Second, the methods are divided into data-level methods and algorithm-level methods, then subdivided into seven subcategories. Especially, we discuss the key thoughts, relative strengths, and weaknesses of classic methods in each subcategory. Then, datasets and common evaluation metrics are compiled to provide a wealth of useful resources. Finally, future research directions for imbalanced node classification on graphs are introduced to promote the boom of this field.

Keywords: imbalanced node classification; class imbalance; graph neural networks; graph learning; survey

1. Introduction

With the rapid development of technologies, a mass of data is captured and stored by computers [1]. The data obtained can be shown as graphs that show the connections between the various entities [2]. For instance, online social networks can be shown as graphs, with people acting as nodes and the social relationship between them being as edges [3]. Graphs are typical non-euclidean structures with arbitrary size and complex topology [4]. Graphs have performed remarkably in various fields, such as social sciences [5] and knowledge graphs [6].

Deep learning is a powerful method that is frequently used for images [7–10]. Convolutional neural networks (CNNs) [11] are extensively employed in computer vision [12], such as semantic segmentation [13] and image processing [14]. Recurrent neural networks (RNNs) [15] or long short-term memory networks (LSTMs) [16] are utilized in natural language processing [17], involving machine translation [18] and sentiment analysis [19]. While deep neural networks can process euclidean structure data such as images [20, 21], text, and audio, they exhibit suboptimal performance in graphs [22].

Graph neural networks (GNNs) have emerged as effective tools for dealing with non-euclidean structure data, garnering increasing attention in recent years [23, 24]. GNNs are widely used in various graph analysis tasks like link prediction [25], text classification [26], graph classification [27], and networks synthesis [28]. The prevalent application case for GNNs is the node classification task, in which a model is trained to predict unknown node labels by comparing or contrasting them with known nodes [29].

However, in the real world, graphs are usually imbalanced, where the number of one class significantly exceeds that of others [30], a pervasive issue in high-stakes applications such as financial fraud detection [31], disease prediction [32], crime forecasting [33], and spam filtering [34]. In these scenarios, the target nodes (e.g., fraudsters) constitute only a minute fraction of the population. Standard GNNs struggle in these settings not merely due to data scarcity, but due to fundamental conflicts between the message-passing paradigm and skewed distributions. From an optimization perspective, the sheer volume of majority nodes dominates the gradient updates, which leaves minority nodes with insufficient training participation [35, 36]. Simultaneously, regarding topological propagation, GNNs rely on aggregating neighborhood information. Consequently, during the message-passing process, the discriminative features of minority nodes are prone to being assimilated by the dominant majority signals, leading to indistinguishable node embeddings [37]. This structural sparsity eventually results in representation collapse, where the model over-learns the robust features of the majority class while overlooking the subtle, structure-dependent patterns of the minority class [38, 39]. Therefore, developing specialized GNN-based methods that explicitly address these topological and optimization biases is crucial. The diagram of imbalanced node classification on graphs is shown in Figure 1.

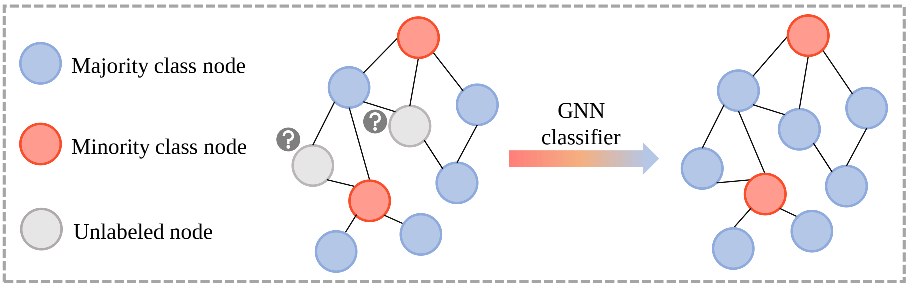

Figure 1. An illustration of imbalanced node classification on graphs. Nodes with known labels are shown as blue and red, while unlabeled nodes are gray. The goal of imbalanced node classification is assign a label to each unknown node and determine its category.

The differences between this article and existing research are as follows.

Previous studies focus on node classification GNNs-based on particular scenarios, and there is a lack of a comprehensive review that contrasts current methods and directs improvement directions for GNNs. For instance, [40] explored the application of GNNs in recommendation systems. [41,42] investigated GNNs-based anomaly detection tasks, [43] examined the use of GNNs in traffic flow prediction. [44] classified the previous class-imbalanced learning methods into two categories, but did not provide a detailed introduction to potential avenues for future development. [45] studied the strengths and weaknesses of GNNs in node classification from four perspectives: class distribution, edge connection ratio, attribute values, and graph size. Another study [46] talked less about the advantages and disadvantages of the current methods. However there are some extensive studies on GNNs [47–49], that touched on imbalanced node classification in brief as one of the application scenarios. This article provides an overview of the research state and development of GNNs-based imbalanced node classification methods, classifies and summarizes various methods, and proposes a comprehensive assessment system. As well, three future research directions are pointed out to provide guidelines for researchers in this field. The main contributions of our work are as follows:

- A detailed summary of the background and related research on graph neural networks and imbalanced node classification is provided in Section 2, which highlights the characteristics of imbalanced node classification on graphs.
- The GNN-based imbalanced classification methods are classified into data-level methods and algorithm-level methods in Section 3. Data-level methods are further divided into data interpolation, generating adversarial, generating pseudo-label, and methods based on large language models. Algorithm-level methods are categorized into loss modification, enhancement strategy, and model optimization. The methods under each classification standard are introduced and contrasted in detail.
- A comprehensive evaluation framework is summarized in Section 4, including benchmark datasets and a critical assessment of performance metrics.
- The shortcomings of existing methods in this field and the directions for future research are identified in Section 5.

2. Related works

2.1. Node classification on graphs

This section gives the definitions of graph and node classification. The graph is defined as $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{A}, \boldsymbol{X})$, where $\boldsymbol{V}$ is the set of nodes, $|\boldsymbol{V}| = n$ is the total number of nodes, $\boldsymbol{A}$ is the set of edges, and $\boldsymbol{X}$ is the node feature matrix corresponding to $\boldsymbol{V}$. The relationship between node i and j is represented by edge a_{ij} in the graph. $\boldsymbol{N}(v)$ represents the set of neighbour nodes with node v. The node classification on graphs can be defined as follows: Given $\boldsymbol{G}$ and a set of labeled nodes $\boldsymbol{V}_l \subseteq \boldsymbol{V}$, the eventual task is to learn an unbiased classifier f that accurately classifies nodes and assigns a label to each unlabeled node in $\boldsymbol{V}_u = \boldsymbol{V} \backslash \boldsymbol{V}_l$.

Node classification uses a message-passing mechanism that learns node properties from the graph to compare neighbors and determine the category of the central node [50]. Node classification plays an essential role in many fields, such as inferring the research topic to which each article belongs in the citation networks [51], ratiocinating the categories of products in co-purchasing networks [52], and identifying the effect of proteins in protein-protein interaction graphs [53–55]. In the case of social networks, users may not expose all the information about themselves, including their hobbies and interests [56]. These attributes can be viewed as labels connected to the users [57, 58]. Labels can categorize users in various ways, such as by interests [59]. Node classification tasks predict these labels so that online platforms can assist users to suggest friends or recommend new products [60].

Currently, a few studies are focusing on graph classification [27], and most of them concentrate on node classification, which is the subject of this paper as well.

2.2. *Class imbalance problem*

The class imbalance problem exists in various datasets [29, 61]. It refers to the significant difference in sample numbers between classes, where the majority class has a larger number of samples than the minority class [62].

Suppose there is a training set $\boldsymbol{x} = \{\boldsymbol{x}_1,...,\boldsymbol{x}_n\}$ and its corresponding label set $\boldsymbol{y} = \{y_1,...,y_n\}$. $\boldsymbol{C}_i$ denotes the set of nodes belonging to class i. The imbalance ratio $h = \frac{\min_i(|\boldsymbol{C}_i|)}{\max_i(|\boldsymbol{C}_i|)}$ is used to represent the ratio of the smallest number of samples to the largest number of samples among different classes. Here, $|\boldsymbol{C}_i|$ indicates the total number of nodes in class i. When $h < 1$, it indicates class imbalance, and a smaller h signifies a more severe imbalance.

The node classification on imbalanced datasets has additional difficulties due to the unique graph structure.

- The degree and category of nodes in the graph typically exhibit a long-tail distribution [63], meaning that there are more nodes in some categories than in others and few high-degree nodes have more links than most low-degree nodes. As a result, high-degree nodes have more influence, making them more likely to attract the attention of the model, while low-degree nodes are overlooked [64].
- The connections between nodes in a graph show a locality structure [65]. It means that nodes in one category are more likely to connect with nodes of specific other categories, leading to an imbalanced class distribution in local regions [66, 67].
- The information propagation in a graph can be affected by class imbalance [68, 69]. When certain categories have fewer nodes, the information spread from these nodes may be limited [70], impacting the local or global structure of the entire graph. Therefore, it is essential to design algorithms that can adapt to various types of graphs.

The existing methods are divided into data-level methods and algorithm-level methods. Data-level methods involve over-sampling the minority class [71], under-sampling the majority class [72], and hybrid sampling [73]. Algorithmic-level methods aim to modify the underlying learning [74] or decision-making process [65, 75] of the model to address the class imbalance. Training a model for node classification on an imbalanced graph is difficult because the imbalance can lead to the bias of the model [76].

2.3. *Graph neural networks*

Traditional machine learning methods, such as graph feature engineering [77] and neighbor relationship-based [78], are the primary means of early graph data processing. These methods typically transform graphs into a vector or matrix format, which is subsequently processed using conventional machine learning models [79]. However, the information contained in graphs cannot be fully exploited utilizing these strategies.

Analogous to convolutional neural networks (CNNs) used for image data, [80] put forward graph convolutional networks (GCNs), which introduce convolution operations to handle graphs. This is a significant breakthrough in the domain of graph processing. GCNs have demonstrated impressive performance in tasks including node classification [81] and graph classification [27], demonstrating their ability to learn node relationships in graphs. The fundamental principle of GCNs is to take the adjacency matrix of graphs as input and learn node representations through multi-layer convolution and pooling operations [23]. As graph learning advanced, graph attention networks (GATs) were proposed by [82]. This work introduced a dynamic attention mechanism based on GNNs that adjusts the relationship weight between nodes to better capture local and global information. [83] put forward the graph sample and aggregate (GraphSAGE), which concentrates on inductive learning and uses sampling and aggregation to learn the low-dimensional representation of nodes. It can generalize to unseen nodes during the training. [84] proposed a powerful message-passing scheme based on deep sets and graph isomorphism networks (GIN), enhancing the expressive power of GNNs. [85] generated the graph self-attention networks to enhance the capacity of the model and represent graphs by taking into account both the information of nodes and their relationships. Additionally, graph neural networks [86] and graph generation models [87] have also emerged, which are employed to handle a wider range of intricate and varied graphs.

GNNs aggregate information from neighboring nodes and edges to update node representations [88]. After k rounds of aggregation iterations, the structural information from the node k-hop neighbors is captured in the node representation, defined as follows:

$$\boldsymbol{u}_v^k = aggr(\boldsymbol{u}_v^{k-1}, \boldsymbol{u}_{\boldsymbol{N}(v)}^{k-1}), \tag{2.1}$$

where $\boldsymbol{u}_v^k$ is the feature vector of node v after k-hop aggregation, and $\boldsymbol{N}(v)$ denotes the neighbor set of node v. The information propagation mechanism in Eq (2.1) is crucial because it generates high-level embeddings that capture complex topological dependencies. These node representations are subsequently encoded and fed into subsequent prediction modules or classification layers to predict node labels. Technically, the function $aggr(\cdot, \cdot)$ serves as a unified abstraction that is instantiated differently across various GNN architectures. Specifically, GCNs implement it as a fixed weighted sum based on node degrees [80], while GATs employ learnable attention coefficients to dynamically weigh the importance of edges [82]. Through this mechanism, the propagation fuses node attributes with complex topological dependencies, generating high-level embeddings fed into subsequent classification layers. However, it is worth noting that when k is set too high, the receptive field $\boldsymbol{N}(v)$ of each node grows exponentially, leading to over-squashing [89]. This typically results in information loss during transmission between distant nodes. Simultaneously, as GNNs increase in depth, repeated aggregation of local information often causes embedded vectors to lose their ability to distinguish between different categories of nodes. This phenomenon, known as over-smoothing [90], prevents GNNs from accurately conveying information from distant nodes.

In Figure 2, the traditional GNN framework is shown. In class-imbalanced scenarios, minority nodes are typically surrounded by majority neighbors. Consequently, during aggregation, the discriminative features of minority nodes are easily overwhelmed by the dominant majority signals, resulting in generated node embeddings that are difficult to distinguish.

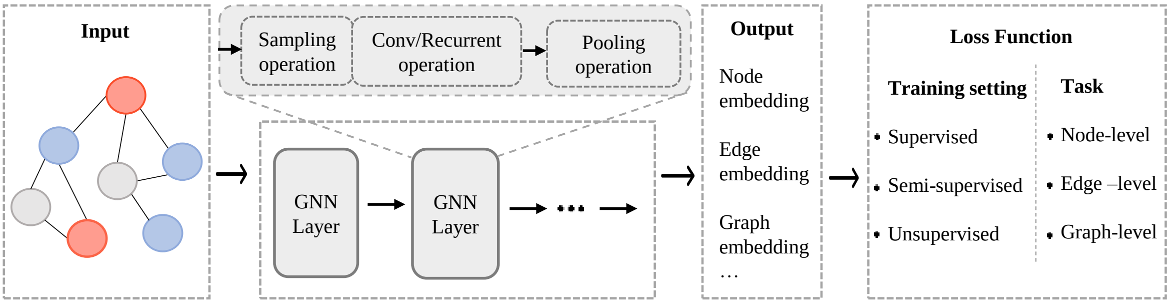

Figure 2. Gray indicates unlabeled nodes, while blue and red indicate positive and negative classes with known labels, respectively. Convolution and recurrent operations are used to collect neighbor information from the input graph. The sampling module can be used to sample a large graph for tasks like prediction, classification, and others.

3. The methods for imbalanced node classification

This section reviews the imbalanced node classification methods based on GNNs, categorizing them into data-level methods and algorithm-level methods. Data-level methods are further divided into data interpolation, generating adversarial, generating pseudo-label, and methods based on large language models. Algorithm-level methods are divided into loss modification, enhancement strategy, and model optimization. Additionally, several popular algorithms within each category are introduced. Figure 3 shows a summary of imbalanced node classification methods.

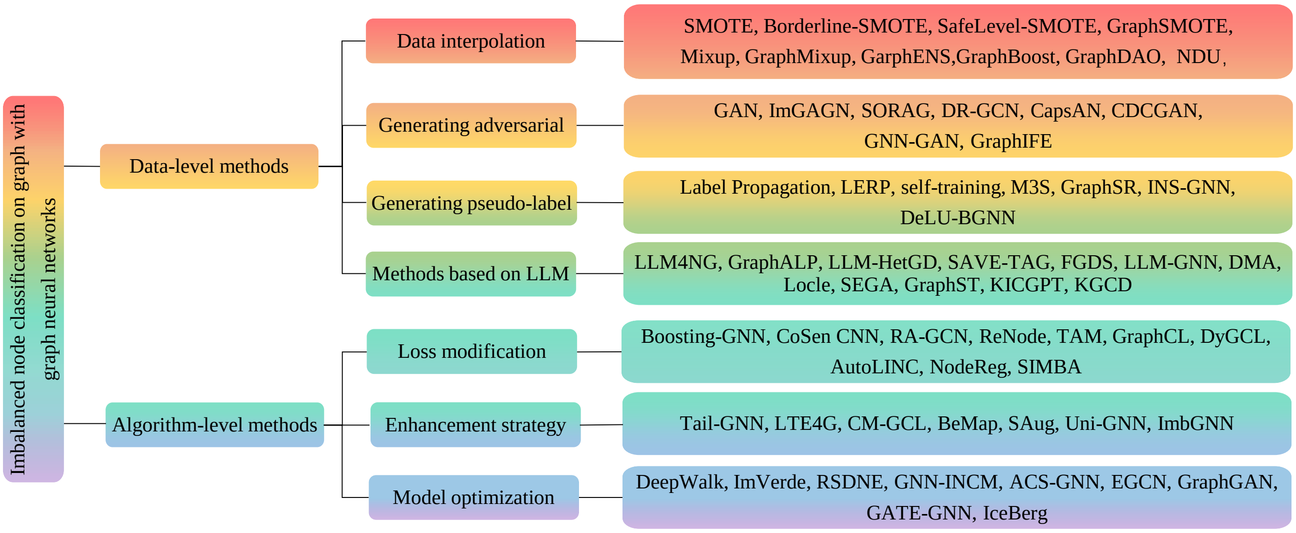
Imbalanced node classification on graph with graph neural networks
Data-level methods
Algorithm-level methods
Data interpolation
Generating adversarial
Generating pseudo-label
Methods based on LLM
Loss modification
Enhancement strategy
Model optimization
SMOTE, Borderline-SMOTE, SafeLevel-SMOTE, GraphSMOTE, Mixup, GraphMixup, GarphENS,GraphBoost, GraphDAO, NDU,
GAN, ImGAGN, SORAG, DR-GCN, CapsAN, CDCGAN, GNN-GAN, GraphIFE
Label Propagation, LERP, self-training, M3S, GraphSR, INS-GNN, DeLU-BGNN
LLM4NG, GraphALP, LLM-HetGD, SAVE-TAG, FGDS, LLM-GNN, DMA, Locle, SEGA, GraphST, KICGPT, KGCD
Boosting-GNN, CoSen CNN, RA-GCN, ReNode, TAM, GraphCL, DyGCL, AutoLINC, NodeReg, SIMBA
Tail-GNN, LTE4G, CM-GCL, BeMap, SAug, Uni-GNN, ImbGNN
DeepWalk, ImVerde, RSDNE, GNN-INCM, ACS-GNN, EGCN, GraphGAN, GATE-GNN, IceBerg

Figure 3. Classification and summary of imbalanced node classification methods based on GNNs.

3.1. Data-level methods

Data-level methods primarily adjust the node distribution through preprocessing or generating new samples, which helps the model better learn the relationships and features among nodes of various classes.

3.1.1. Data interpolation

Data interpolation synthesizes nodes of the minority class with a linear combination of existing samples, as shown in Figure 4.

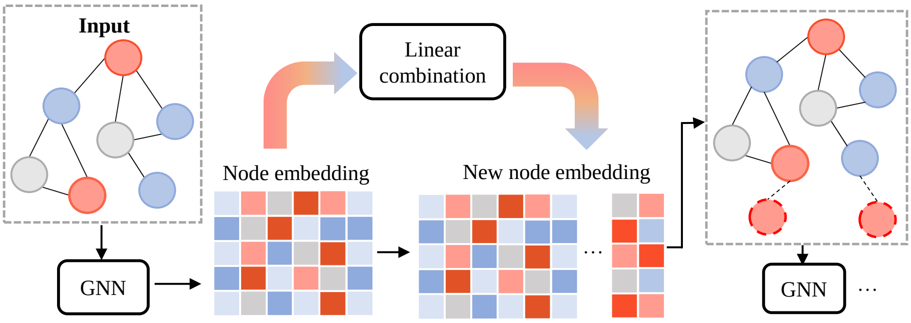

Figure 4. The framework of the data interpolation method. In the original imbalanced network, minority and majority nodes are represented in red and blue, respectively. The synthetic minority nodes generated by data interpolation are indicated by red dashed circles. Edges between original nodes are shown as solid lines, while edges between synthetic minority nodes and original minority nodes are shown as dashed lines.

One classic data interpolation method for addressing class imbalance is the synthetic minority oversampling technique (SMOTE) [81]. Since the random oversampling method directly resamples the minority class, there will be many repeated samples in the training set, which can cause overfitting [71]. SMOTE [81] addresses this problem via interpolating between minority samples to synthesize samples. Neighboring samples are randomly selected based on the required oversampling rate to sample each minority class. The new nodes are synthesized by

$$\tilde{\boldsymbol{x}} = \boldsymbol{x}_i + \beta|\boldsymbol{x}_j - \boldsymbol{x}_i|, \tag{3.1}$$

where $\boldsymbol{x}_i$ represents a random minority class node, $\beta \in [0, 1]$, and $\boldsymbol{x}_j$ is one of the neighbors from the same minority class as $\boldsymbol{x}_i$.

The main disadvantage of SMOTE is that it does not consider the majority class in the neighborhood when generating new synthetic samples, which leads to class overlap and introduces additional noise. Borderline-SMOTE [91] focuses on sampling minority samples that are on the border, synthesizing new samples in these critical areas. SafeLevel-SMOTE [92] samples within safe regions computed from the nearest neighbors of the minority class, aiming to prevent oversampling in overlapping or noisy regions. However, these methods present the following challenges when used in graphs. First, the complicated information of the graph is not fully taken into account by the nodes synthesized in the original feature space. Second, the synthesized nodes are unable to establish edges with existing nodes, which makes it difficult for GNNs to make full use of the information of synthesized nodes.

By creating an embedding space to encode node similarity, GraphSMOTE [73] overcomes these difficulties. New synthetic nodes are formed in the embedding space between two minority nodes. Furthermore, an edge predictor is built to establish links between the prior nodes and the synthetic nodes. However, GraphSMOTE synthesizes new nodes by interpolating nodes within the same minority class, which can limit the diversity of the synthetic nodes.

Mixup [93] creates a synthetic sample training set and interpolates all class samples rather than concentrating on the minority class. Concretely, it generates node set by

$$\tilde{\boldsymbol{x}} = \boldsymbol{x}_i + \beta|\boldsymbol{x}_j - \boldsymbol{x}_i|,\ \tilde{\boldsymbol{y}} = \boldsymbol{y}_i + \beta|\boldsymbol{y}_j - \boldsymbol{y}_i|, \tag{3.2}$$

where $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ and $(\boldsymbol{x}_j, \boldsymbol{y}_j)$ are any two nodes of different types, $\beta \in [0, 1]$. Due to the extremely sparse distribution of minority class, performing mixups directly in the original space or feature space can result in out-of-distribution samples. GraphMixup [29] constructs semantic relation space, which enables the mixup of features to be processed at the semantic level.

When synthesizing nodes, the connection relationship between nodes in the graph is prone to neighbor memory problems. To address this problem, GraphENS [94] synthesizies an ego network for the minority class by neighbor sampling and saliency-based node mixing. Specifically, the synthesized nodes $\boldsymbol{v}_{mixed}$ can be formulated as follows given minor nodes $\boldsymbol{v}_{minor}$, target node $\boldsymbol{v}_{target}$ from entire classes, and hyperparameter K:

$$\begin{aligned} \boldsymbol{v}_{mixed} &= (1-\boldsymbol{\Lambda}_K) \odot \boldsymbol{v}_{minor} + \boldsymbol{\Lambda}_K \odot \boldsymbol{v}_{target}, \\ \boldsymbol{\Lambda}_K &= \varphi \boldsymbol{M}_K, \end{aligned} \tag{3.3}$$

where $\boldsymbol{M}_K$ is a binary matrix, which masks the $K\%$ of node features to 0, $\odot$ indicates element-wise multiplication, and φ is a random mixing ratio. GraphBoost [95] starts from the difficult-to-classify nodes and generates the topology and features of synthetic nodes by minimizing the distance of synthetic nodes and misclassified nodes from training stages. GNN-CL [50] puts forward a two-step curriculum learning method for the node imbalance issue. In the first step, it identifies confidence labeled nodes and edges using an innovation oversampling method. Graph-DAO [96] first obtains node representations and synthesizes minority nodes through multilayer perceptron (MLP), and then aggregates information from GNNs. It also introduces the sum-product network (SPN) to better learn minority class information. Sun et al. [97] proposed F_factor to measure the importance of majority class samples in the overlapping region and applied it to the nearest neighbors and density-based undersampling (NDU) strategy. NDU is employed for the majority class, while SMOTE is applied to the minority class, ultimately balancing the number of samples between the two classes and effectively alleviating the class overlap problem.

3.1.2. Generating adversarial

The generating adversarial method combines various variants of GANs to synthesize minority class nodes, improving the learning effect of the model on the minority class, as shown in Figure 5. GAN [98] generates realistic nodes by training two adversarial neural networks (a generator and a discriminator), training D to maximize the probability of assigning the correct label to both trai ning examples and samples from G [98].

$$\begin{aligned} &\min_G \max_D V(D, G) = E_{\boldsymbol{x} \sim P_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \\ &E_{z \sim p_z(\boldsymbol{z})}[\log(1 - D(G(z)))], \end{aligned} \tag{3.4}$$

where G is a differentiable function represented by a multilayer perceptron, and D is a second multilayer perceptron that outputs a single scalar. $p_z(z)$ represents a prior over the input noise variables, $D(\boldsymbol{x})$ is the probability that the discriminator discerns $\boldsymbol{x}$ is a real node, and $p_{data}(\boldsymbol{x})$ denotes the distribution of real nodes. $E_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})]$ represents the mathematical expectation of the probability that a node taken from the real data distribution is classified as a real node by the discriminator. $E_{z\sim p_z(z)}[\log(1 - D(G(z)))]$ is the expectation of the negative logarithm of the probability predicted by the discriminator. The accuracy of the discriminator in identifying real or false data grows with the size of this value.

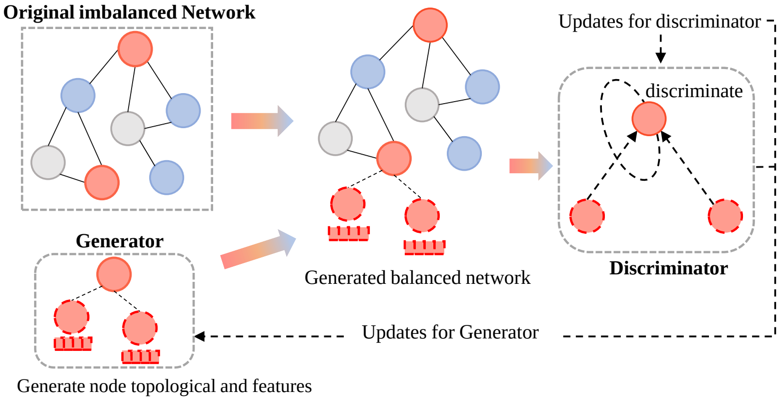
Original imbalanced Network
Generator
Generate node topological and features
Generated balanced network
Updates for discriminator
discriminate
Discriminator
Updates for Generator

Figure 5. The framework of the generating adversarial method. The generator creates samples that resemble the training data, while the discriminator distinguishes original data and generated samples. Through the adversarial training, the generator can generate samples that are very close to the real data. Thus, the number of nodes can be balanced.

ImGAGN [99] is the first to apply GANs to graphs, introducing the GraphGenerator to simulate the distribution of minority class and network topology. However, since each generator can only generate new nodes for a single class, while it achieves good results in binary classification, it cannot effectively extend to multi-class classification. SORAG [100] integrates GAN and cGAN as a node generator and is used for multi-label classification in graphs. GAN is used to generate unlabeled synthetic minority nodes, while cGAN is used to create labeled synthetic minority nodes. Two kinds of regularization are created by DR-GCN [101] and combined with a conditional adversarial training process to separate the labeled node representations from various classes. CapsAN [102] propose a lightweight capsule adversarial network that effectively prevents overfitting while generating new samples for the minority class. CDCGAN [103] is proposed as a class distribution-aware conditional generative adversarial network, in which the model consists of a class distribution-aware module and a minority class enhancement module. The embedding representations of the nodes are first extracted, followed by the computation of the class distribution in the latent space. Finally, diverse minority class samples that conform to the class distribution are generated by the enhancement module.

Zhou et al. [104] proposed GNN-GAN, which jointly trains GAN and GNN, effectively addressing the limitation of traditional sequential training where models cannot dynamically adapt. Zeng et al. [105] drew inspiration from GANs to propose the GraphIFE framework, addressing the non-distributional (OOD) problem prevalent in existing node synthesis methods. This framework comprises two key components: the invariant feature extractor (IFE) and the environmental feature extractor (EFE). The IFE ensures consistency between synthesized and original nodes at the feature level, effectively mitigating OOD bias. The EFE introduces controlled perturbations, and it enhances the model's ability to distinguish minority class samples.

3.1.3. Generating pseudo-label

In real-world graphs, there are numerous unlabeled background nodes [106], and their information has not been utilized efficiently. Generating pseudo-labels for background nodes to balance the training data is a feasible direction, as shown in Figure 6.

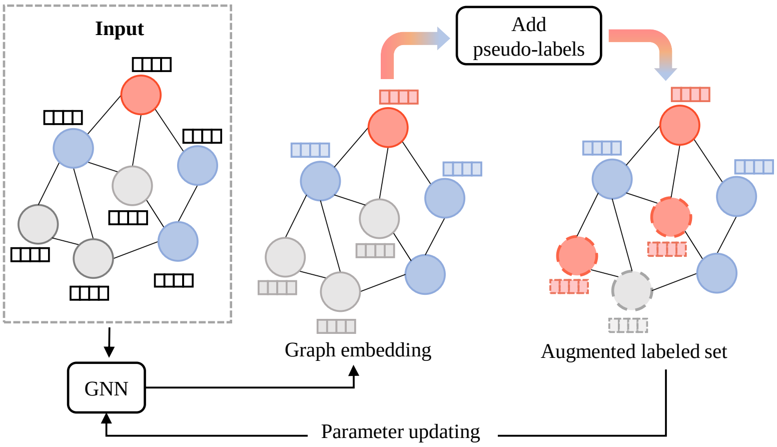

Figure 6. The framework of the generating pseudo-label method. First, input the graph to obtain node embeddings. Second, add pseudo-labels to the unlabeled nodes. Finally, combine these nodes with the original node to update the initial model. These processes yield optimal pseudo-labeled nodes, which are used to train the subsequent node classification model.

The classic graph-based pseudo-label generating method uses label propagation (LP) [107]. Each node label is propagated to its neighboring nodes based on similarity [108]. LP employs a weighted averaging method for label updates.

$$\boldsymbol{Y}_i^{(t+1)} = \frac{1}{\sum_j \boldsymbol{w}_{ij}} \sum_{j \in \boldsymbol{N}(i)} \boldsymbol{w}_{ij} \boldsymbol{Y}_j^{(t)}, \tag{3.5}$$

where $\boldsymbol{Y}_i^{(t)}$ represents the label of node i at the t-th iteration, $\boldsymbol{N}(i)$ denotes the neighbor set of node i, and $\boldsymbol{w}_{ij}$ is the weight between node i and j. The similarity metric between nodes determines how well LP performs, and it is quite sensitive to the original labels. An inappropriate number of iterations can result in either overfitting or underfitting [109]. The label efficient regularization and propagation (LERP) method was put forward by [110]. In each round of alternation, reliable pseudo-labeled nodes are adaptively determined through their distance to labeled nodes.

Other methods include enhancing the training samples for minority class by utilizing self-training [111] to generate pseudo-labels. Self-training utilizes labeled nodes to train an initial model, generates pseudo-labels for unlabeled nodes, and combines them with the original nodes for model updates [112]. High-confidence unlabelled nodes are more likely to be predicted correctly in self-training, whereas discovering them incorrectly leads to mislabelling [113]. Thus, correctly discerning high-confidence unlabeled nodes is a critical issue.

Most of the present methods [31,76] aim to improve the reliability of pseudo-labels by combining self-training with other algorithms. Deep clustering and self-training are combined in M3S [76], which concentrates on graphs with fewer labeled nodes. GraphSR [31] incorporates reinforcement learning and similarity selection to enhance the precision of choosing high-confidence nodes and adjusts the sampling scale adaptively. To obtain pseudo-labels for unlabeled nodes, INS-GNN [114] utilizes self-training based on the node and topologyical attributes of the graph, and it enhances the edges of minority nodes, increasing their participation. DeLU-BGNN [115] investigates graph imbalance learning from both quantitative and topological perspectives. It combines a self-training mechanism with Bayesian graph neural networks to optimize the topological structure of specific nodes, thereby facilitating message propagation.

3.1.4. Methods based on large language models (LLM)

Large language models (LLM) pre-trained on massive corpora demonstrate exceptional performance in knowledge generation and capturing long-term dependencies [116]. Consequently, they offer novel approaches to address class imbalance in graph learning.

Yu et al. [117] pioneered the application of LLM to node generation in text-attributed graphs (TAGs), proposing LLM4NG (LLM for node generation). LLM4NG guides the LLM to generate labeled samples through prompts and label information, denoted as:

$$V' = LLM\left(Prompt_{node}(C)\right), \tag{3.6}$$

where C denotes the category label, and V' represents the generated sample. To integrate the generated sample into the original dataset, the authors first feed both the generated sample V' and the original dataset sample V into an embedding model to extract their vector representations H' and H. Subsequently, cosine similarity is used to select suitable node pairs, which are then input into a multi-layer perceptron (MLP)-based edge predictor, as shown below:

$$\hat{y}_e(H', H) = MLP(H' \parallel H), \tag{3.7}$$

where $\hat{y}_e$ represents the probability of an edge existing between two nodes, and $H' \parallel H$ denotes the feature concatenation of node pairs. Finally, the top-k edges with the highest selection probability are added to the original graph.

Additionally, GraphALP [118] and LLM-HetGDT [119] also employ LLMs to synthesize minority class nodes. Unlike LLM4NG, GraphALP introduces a pseudo-labeling mechanism and further adopts an LLM-based secondary oversampling strategy to mitigate potential category distribution offset caused by pseudo-labels. On the other hand, when synthesizing nodes using LLMs, LLM-HetGDT does not rely on category labels. Instead, it takes the corresponding data T as input and uses the LLM to generate node V' and its corresponding data T'. Subsequently, the LLM is further utilized to generate edges between the synthetic node V' and the original node V, namely $edge = LLM\left(Prompt_{edge}(V', V)\right)$.

Based on the vicinal risk minimization (VRM) principle, SAVE-TAG [120] performs LLM-based semantic interpolation on pairs of vicinal twins within minority class nodes, defined as: $\hat{t} = L_\tau(t_i, t_j|C)$, where L_τ is a temperature controlled LLM, t_i, t_j are text attributes of minority-class nodes in the same category, and C is the class label. FGDS [121] integrated three key factors tailored to the structure of streaming graphs, enabling LLMs to better understand graph structural information and generate samples that better align with temporal and topological relationships. LLM-GNN [122] designed a difficulty-aware heuristic to select suitable nodes for generating pseudo labels using LLM. Sheng et al. [123] revealed that pseudo-labels generated by LLMs often contain noise and semantic bias. To address this issue, they designed the Dataset- and LLM-aware graph active learning framework (DMA), which utilizes noise distributions to measure node reliability and adaptively selects nodes with the most reliable information content. Locle [124] selects representative nodes for annotation through subspace clustering, ensuring that minority clusters also have representative nodes chosen.

SEGA [125] guides the LLM synthesis process by establishing explicit principles to ensure enhanced data integrity, respectfulness, and diversity in sensitive domains such as mental health. GraphST [126] employs the multimodal model CLIP to generate class semantic descriptions and optimizes node representations through contrastive learning, thereby effectively improving the discriminative capability for minority-class nodes. Addressing cost and privacy concerns in large models, Pan et al. [127] propose a two-stage distillation framework. First, an LLM (teacher model) identifies node features and generates reasoning evidence to train an intermediate interpreter. Subsequently, the TAG model (student model) distills knowledge from the interpreter through semantic and structural alignment.

Traditional knowledge graph completion methods often suffer from performance degradation due to sparse long-tail entity structural information. KICGPT [128] employs LLMs as external common-sense knowledge bases, encoding knowledge through knowledge prompt to refine retriever candidate results and prevent unrestrained generation. KGCD [129] addresses LLM generation bias and data imbalance by synergizing rethink prompts with open prompts.

Unlike traditional data interpolation (e.g., SMOTE) or GAN-based methods that operate in numerical feature space or latent space, the uniqueness of LLM-based methods lies in their ability to utilize semantic reasoning and open-world knowledge. LLMs can generate high-quality synthetic nodes with rich textual attributes by understanding the semantic context of the graph, rather than merely fitting the statistical distribution of the minority class. This allows for zero-shot or few-shot generation capabilities that purely structural methods lack. However, these advantages come with substantial costs. LLM-based methods incur extremely high computational overhead due to the massive parameter of models (requiring significant GPU memory and inference time) or financial costs associated with commercial APIs.

3.2. *Algorithm-level methods*

Algorithm-level methods mostly focus on enhancing the weights of a minority class or modifying current methods to make them appropriate for imbalanced datasets.

3.2.1. Loss modification

By modifying the objective function, loss modification methods address the issue of class imbalance. Typically, there are two methods. First, methods may set greater weights to the loss of a minority class nodes [32, 71, 130]. This adjustment ensures the model is more attentive to the loss of minority class, mitigating the problem of the majority class dominating the loss function [64]. Second, methods can expand the decision boundary between the minority class and the majority class by modifying the loss function, which reduces the probability of misclassification [65, 75]. To enhance the performance of the model, these techniques can be applied singly or in combination.

Boosting-GNN [71] reduces the weights for training nodes that the classifier properly classifies while assigning higher weights to those that are misclassified. CoSen CNN [130] optimizes neural network parameters and the misclassification cost together to mitigate class imbalance. [32] proposed the reweighted adversarial graph convolutional network (RA-GCN), which reweights the nodes to change the importance of each node to the classifier, helping the classifier better fit both the majority and minority class.

However, in graphs, training samples of different classes not only differ in quantity but may also occupy distinct positions in the topological structure. Graph topology information is incorporated into the loss function design of ReNode [65] and topology-aware margin (TAM) [75]. In node representation learning, nodes near class boundaries are at risk of information conflict, while nodes far from labeled nodes lack sufficient information [70].

ReNode [65] decreases the weight of nodes near class boundaries and raises the weight of nodes near class centers. The design of loss function is

$$L_{\mathrm{Re}Node} = \frac{1}{|\boldsymbol{V}_l|}\sum_{v\in\boldsymbol{V}_l}\boldsymbol{\omega}_v\frac{|\bar{\mathbf{C}}|}{|\mathbf{C}_{y_v}|}L(\boldsymbol{l}_v, y_v), \tag{3.8}$$

where $\boldsymbol{\omega}_v$ is the trainable weight of node v, $\boldsymbol{l}_v$ denotes the logit of node v, $|\bar{\boldsymbol{C}}|$ represents the average size of the class training samples, $|\boldsymbol{C}_{y_v}|$ signifies the total number of training nodes in the class that node v is a part of, and $L(,)$ denotes the initial loss function.

TAM [75] observes a higher false positive rate around a few nodes in regions with high heterophily. Therefore, TAM suggests that when a small number of nodes possess an unusually large number of major neighbors, their weights should be reduced. TAM designs the anomalous connectivity-aware margin (ACM) and anomalous distribution-aware margin (ADM), and calculates the confusion level based on the average statistics of neighboring nodes. The final loss function is shown as

$$L_{TAM} = \frac{1}{|\boldsymbol{V}_l|}\sum_{v\in\boldsymbol{V}_l}L(\boldsymbol{l}_v + \alpha\boldsymbol{m}_v^{ACM} + \beta\boldsymbol{m}_v^{ADM}, y_v), \tag{3.9}$$

where $L(,)$ represents the initial loss function, $\boldsymbol{l}_v$ is the logit of node v. $\boldsymbol{m}_v^{ACM}$ denotes the adjustment of class margins by calibrating the deviation in the neighbors label distribution (NLD) for node v. $\boldsymbol{m}_v^{ADM}$ indicates the adjustment of the margin based on the relative distance computed using the average NLD of the target class and the node's own class. α and β are hyperparameters.

Another method is graph contrastive learning (GCL) [131]. GCL randomly samples a batch of graphs and performs data augmentation on each graph (such as deleting several nodes) to obtain enhanced views. To determine whether the generated views are identical to the original graphs, GNN is utilized to encode the views and compute the InfoNCE loss. [132] showed that the message-passing mechanism of GNNs and the contrastive loss are strongly related, building a connection between contrastive learning and GNNs. [133] proposed a new dynamic graph contrastive learning framework (DyGCL) that captures the temporal consistency of dynamic graphs by constructing a GCL trainer. In unlabeled or sparsely labeled datasets, it can also perform effectively. However, nodes evolve over time in real-world circumstances, and the subgraph sampler of DyGCL is unable to capture large time spans. One avenue for future enhancement is to identify better long time interval subgraph sampling methods [134]. Motivated by curriculum learning, [50] fuses node classification and metric learning loss to modify the distance with distinct nodes linked to minority classes. This mechanism can improve the capacity to discriminate and generalize. AutoLoss for imbalanced node classification (AutoLINC) was proposed by Guo et al. [135], in which poorly performing loss functions are actively filtered out through task evaluation with a loss-checking strategy, and Monte Carlo tree search algorithms are optimized to accelerate the searching process. It was found by Yang et al. [136] that an imbalance in the norm of node feature representations leads to poor performance of GNNs in node imbalance and out-of-distribution generalization problems. To address this issue, NodeReg was proposed to ensure the consistency of the node representation norm, thereby improving classification performance in both cases.

In multi graphs classification tasks also plagued by imbalance, Qin et al. [137] proposed an energy-guided structural smoothing framework (SIMBA). This framework introduces an energy-based graph reweighting module that calculates the final energy score for each graph through an energy belief propagation mechanism. Subsequently, a cosine annealing mechanism assigns training weights to each graph based on the propagated energy values, thereby achieving smoothing of structural differences in local regions.

3.2.2. Enhancement strategy

The enhancement strategy aims to improve model training by incorporating additional information. Most real-world graphs exhibit long-tail distributions in both categories and node degrees, i.e., the node degrees follow a power-law distribution [4, 70]. GNNs cannot generalize to the minority class and favor the majority class in terms of categories [138]. Regarding node degree, high-degree nodes (majority nodes) often possess plentiful connections, while the low-degree nodes (minority nodes) have fewer connections, causing GNNs to be biased towards majority nodes [139].

Tail-GNN [70] addresses the long-tail distribution problem by simulating connections between nodes and their neighborhoods, where information flows from high-order nodes to low-order nodes to enhance their representations. But, it does not account for the long-tail distribution of the class. Yun et al. [63] proposed long-tail experts for GNNs (LTE4G), which takes into account the long tail of both classes and node degrees. The graph is divided into four balanced subsets and trains an expert model on each subset. Next, using a prototype-based method and a knowledge distillation technique [140], each test node is allocated to the class that has the highest similarity score based on how similar it is to the class prototype. [74] put forward the co-modality graph contrastive learning (CM-GCL). CM-GCL jointly trains encoders with different modalities to enhance node representation learning. Contrastive pairs are generated from unlabeled data automatically, which facilitates the learning of more balanced node representations. The cross-modal contrastive loss can be expressed as

$$\mathcal{L}_{GCL} = -\log \sum_{v_i \in V} \frac{\exp(\mathrm{sim}(z_i, z_i^+)/\tau)}{\sum_{j=1}^{2n} \exp(\mathrm{sim}(z_i, z_j^+)/\tau)}, \tag{3.10}$$

where z_i and z_i^+ represent the embeddings of the original node, respectively. The term z_j^+ denotes the embeddings of other nodes in the batch, which serve as negative samples. The function $sim(\cdot, \cdot)$ computes the cosine similarity between two vectors, n is the size of the mini-batch, and τ is a temperature hyperparameter that regulates the discrimination sensitivity.

BeMap [141] is designed to achieve a balanced graph structure by discarding points in the local neighborhood of a node (i.e., through edge deletion), thereby facilitating the fair delivery of messages. SAug [142] is proposed with a PageRank-based strategy to distinguish hub nodes and tail nodes. Noisy neighbors are removed for hub nodes, potential neighbors are discovered for tail nodes, and pseudo-neighbors are generated to alleviate the structural imbalance between the two types of nodes.

To better handle structural information, Uni-GNN [143] integrates structural and semantic coders. This design enables message propagation to non-adjacent yet similar nodes, thereby generating comprehensive node embeddings. A pseudo-label generation strategy is also introduced to enhance the robustness and effectiveness of the framework. ImbGNN [144] proposes that structural imbalance and class imbalance often co-occur. To mitigate structural imbalance, ImbGNN designed a degree-aware optional augmentation method that samples augmentation techniques from distinct distributions for both degree-head and degree-tail graphs.

3.2.3. Model optimization

Model optimization is the process of updating the underlying architecture of graph learning. It can be achieved by integrating specific modules to change the overall network change or modifying the learning paradigm.

For network embedding methods, deepwalk [145] treats random walk sequences as sentences and nodes as words, employing the skip-gram model to learn latent representations. Specifically, for a random walk sequence $\mathcal{W} = \{w_1, \ldots, w_L\}$, it aims to maximize the co-occurrence probability of nodes within a context window w. The objective function is formulated as minimizing the negative log-likelihood:

$$\mathcal{L} = -\sum_{v \in \mathcal{W}} \sum_{u \in N_w(v)} \log P(u|\Phi(v)), \tag{3.11}$$

where $\Phi(v)$ is the embedding of the central node v, and $N_w(v)$ represents its context neighbors in the walk. To ensure computational efficiency on large graphs, hierarchical softmax is utilized to approximate the probability $P(u|\Phi(v))$, reducing the complexity from $O(|V|)$ to $O(\log|V|)$. However, deepwalk performs less well on graphs with intricate local patterns or community structures since it is unable to extract the local structural information of graphs [146]. ImVerde [147] adjusts the transition probabilities of random walks to ensure that minority nodes are more likely to remain within the same class, thereby producing better node representations. In RSDNE [148], auxiliary learning objectives are incorporated to ensure intra-class similarity and inter-class dissimilarity in the embedding space.

For optimizing GNNs, GNN-INCM [62] designs embedding clustering-based optimization (ECO) and graph reconstruction-based optimization (GRO). Node embeddings are obtained using a two-layer GCN in ECO, followed by clustering analysis to improve the representativeness of node embeddings and simplify classification. In GRO, an inner-product decoder is utlized to reconstruct the graph, thereby reducing information loss. The graph reconstruction loss is formulated as follows:

$$L_{g-re} = \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} \mathbf{A}_{ij} \log \widehat{\mathbf{A}}_{ij} + (1 - \mathbf{A}_{ij}) \log \left(1 - \widehat{\mathbf{A}}_{ij}\right). \tag{3.12}$$

L_{g-re} aims to compute the binary cross-entropy, which measures the difference between the original matrix $\mathbf{A}$ and the reconstructed matrix $\widehat{\mathbf{A}}$.

ACS-GNN [149] and EGCN [150] modify the aggregation operations within standard GNN architectures. An attention mechanism is performed by ACS-GNN to set various weights to different nodes. EGCN limits inter-class edge aggregation using estimated density and pays more attention to the minority class. GraphGAN [151] updates node representations using the generative adversarial framework, with graph softmax employed to compute the estimated connection distribution. GraphGAN performs a breadth-first search (BFS) for each node, which consumes additional space complexity. However, it is thought that applying GAN to network representation learning is a worthwhile direction to explore [152]. GATE-GNN [153] is proposed with the graph ensemble weight attention (GEWA) mechanism. GEWA is designed to integrate inputs from graph node feature extraction, node ensemble, and transfer learning. The most informative features can receive priority attention, while the overbearing influence of the majority class is minimized. This enables a balanced representation of each class to be provided by the model. IceBerg [154] is proposed with double balancing (DB) to address the Matthew effect and label distribution bias that occur in self-training. To mitigate the over-smoothing problem caused by increasing the number of model layers, the framework is designed to decouple propagation and transformation, thereby redesigning the architecture to optimize message passing.

3.3. Comparative analysis

Although the aforementioned methods address the imbalance problem from different perspectives, a thorough understanding of their inherent trade-offs is crucial for practical model selection. We analyze various approaches from two dimensions: mechanism and applicability.

1) Mechanism analysis

Most data interpolation and GAN-based methods typically rely on the homophily assumption, where edges tend to connect nodes with the same category labels [155]. This approach often fails to construct heterogeneous edges, introducing structural bias that degrades model performance. Additionally, GAN methods face risks of training instability and mode collapse, where the generator may fail to capture the diversity of minority class samples, leading to generated samples becoming monotonous. Pseudo-labeling methods typically select high-confidence nodes to expand the training set, but early mispredictions by the model may be reinforced as true labels, causing cumulative errors. LLMs possess robust semantic reasoning capabilities and extensive external knowledge, excelling at text attributes but typically exhibiting weak understanding of graph topology. Moreover, the hallucination risk inherent in large models is difficult to eliminate, potentially leading to generated node features or labels containing erroneous information.

Loss modification avoids altering graph structure by adjusting sample weights or boundaries, thereby preserving the integrity of the original topology. However, as a passive adjustment strategy, it cannot physically increase the information content of minority classes and has limited effectiveness for structurally isolated nodes. Enhancement and model optimization enhance representational capabilities by introducing auxiliary tasks (e.g., GCL) or designing specialized message-passing mechanisms. While these approaches extract more robust features, they often introduce additional hyperparameters and complex optimization objectives, increasing model training complexity.

2) Applicability analysis

First, the core mechanism of loss modification solely involves sample reweighting within the objective function, with computational complexity determined by the loss function and reweighting algorithm design. Compared to other methods, it exhibits lower computational costs and more manageable implementation difficulty. This also grants it superior scalability when handling large-scale graph data. Enhancement strategies and model optimization typically require introducing additional training processes or redesigning the model architecture, incurring computational costs and resulting in steeper implementation curves. Consequently, scalability on large graphs is reduced.

Second, data interpolation and pseudo-labeling methods are technically mature with moderate implementation difficulty. They primarily rely on interpolation algorithms and LP algorithms, resulting in low computational costs on small-scale graphs. GAN methods require training additional network structures (generator and discriminator), entailing significant computational overhead and involving stability debugging for adversarial training, making implementation more challenging. For LLM approaches, local deployment incurs substantial GPU memory and inference latency, while commercial API calls entail economic costs. Since LLM methods primarily utilize pretrained LLMs to generate node or label information and comprehensive deployment guides are available, implementation difficulty is relatively low. It is worth noting that these generative methods often struggle to be directly applied to large-scale graph data, frequently requiring complex sampling strategies. Consequently, scalability in large-scale scenarios is poor.

In Table 1, the core ideas of the different approaches, their relative advantages and their limitations are clearly compared.

Table 1. Summary of methods for addressing imbalanced node classification in GNNs, highlighting their core mechanisms, relative strengths, and potential limitations.

Categories	Subcategories	Key concepts	Relative strengths	Weaknesses
Data-level	Data interpolation	Generation of synthetic samples by linear combination	Simple and easy to implement, low computational cost	May lead to overfitting
	Generating adversarial	Generation of synthetic samples by GANs	Higher quality generation, avoiding oversampling	High training complexity high computational cost
	Generating pseudo-label	Generating pseudo-labels using unlabeled data	Effective use of unlabeled data to reduce labeling costs	Sensitive to initial labeling and high risk of mislabeling
	Methods based on LLM	Semantic-Level reasoning	Strong semantic comprehension ability	High computational cost; Prompt sensitivity
Algorithm-level	Loss modification	Adjusting the loss function or sample weights	Direct optimization of the loss function with high flexibility	Possible overfocus on a few classes leading to overfitting
	Enhancement strategy	Enhancing model training with additional information	Improves model generalization capabilities	Dependence on the quality of additional information
	Model optimization	Adaptation of network architectures or training paradigms	Improved model architecture to enhance the quality of minority class representation	Increased model complexity and computational overhead

4. Assessment setup

We offer a thorough overview of the imbalanced node classification assessment setup in this section, including commonly used datasets and evaluation metrics.

4.1. Datasets

This section introduces the public datasets for GNN-based imbalanced node classification. These datasets are summarized in Table 2, which includes imbalance rates, label rates, numbers of nodes, edges, classes, and feature information. The datasets are categorized based on their sources.

Table 2. A detailed summarization of benchmark datasets.

Domain	Dataset	Imbalance rate	Label rate	Nodes	Edges	Features	Classes
Citation datasets	Cora [156]	24%	5.2%	2708	5429	1433	7
	Citeseer [156]	38%	0.56%	3327	4732	3703	6
	Pubmed [80]	40%	0.03%	19717	44,338	500	3
protein-protein interaction datasets	PPI [157]	–	–	56,944	818,716	50	121
	ogbn-proteins [158]	13.88%	100%	132,534	39,561,252	8	2
Knowledge graph	Wiki-CS [159]	2.2%	0.17%	11,701	216,123	300	10
Co-authorship datasets	Coauthor-CS [160]	22.6%	0.16%	18,333	81,894	6805	15
	Coauthor-Physics [160]	–	0.03%	34,493	247,962	8415	5
Co-purchase datasets	Amazon-Photo [161]	–	2.14%	7487	119,043	745	8
	Amazon-Computers [161]	–	1.49%	13,381	245,778	767	10
Social datasets	Flickr [162]	–	4.9%	80,513	5,899,882	1282	195
	Weibo [163]	–	10.3%	8405	407,963	400	–
	YouTube [164]	–	–	1,157,827	4,945,382	4000	47
	BlogCatalog [165]	0.2%	–	10,312	333,983	–	39
	Github [166]	–	0.01%	37,700	289,003	4005	2
Financial dataset	DGraphFin [106]	1.3%	23.2%	3,700,550	4,300,999	17	3
	Elliptic++ [167]	10.82%	22.85%	203,769	234,355	183	2

1) Citation datasets

Cora [156], Citeseer [156], and Pubmed [80] are typical examples of citation networks, where documents and their citation relationships are represented by nodes and edges, respectively. The node features are derived from the bag-of-words representation of the documents. The study domain of each document node is indicated by a label. These datasets can be utilized for tasks such as studying the relationships between academic documents, information retrieval and recommendation [6], and graph data mining [168].

2) Knowledge graph

The Wiki-CS dataset [159] is a reference network based on wikipedia. Nodes represent articles, and edges are the links between article nodes. Node features are obtained from the embeddings of words [169] in each computer science article. Each node is labeled with one of ten categories, representing a different branch of computer science. This dataset is used to construct knowledge graphs in the computer science domain, aiding research in question answering systems [6] and other related tasks.

3) Protein-protein interaction datasets

Protein-protein interaction (PPI) [157] refers to the graph formed by the binding process of two or more proteins. Each node has a feature vector with a length of 50, including positional gene sets, motifs, and immunological features. This dataset provides a crucial foundation for biological research and medical applications, facilitating the understanding of the structure and function of protein interaction networks. It is used for tasks such as drug design and development [170], disease diagnosis and treatment [32], and protein structure prediction [157]. The ogbn-proteins dataset is an undirected, weighted, multi-task protein-protein interaction network provided by the Open Graph Benchmark [158]. Nodes represent proteins, while edges denote different types of biological associations between proteins, such as physical interactions, co-expression, or homology [171]. This dataset comprises 112 classification tasks, many of which exhibit extremely imbalanced sample distributions, with an average imbalance rate of 13.88%.

4) Co-authorship datasets

The Coauthor-CS and Coauthor-Physics datasets [160] are academic networks composed of coauthors from Microsoft. Nodes represent authors, and edges are formed if they have coauthored a paper. A node is characterized by a sparse bag-of-words feature based on the keywords of the author's papers and is labeled with the research field in which the corresponding author is most active. These datasets are utilized for tasks such as coauthor analysis [172] and disciplinary field analysis [173], promoting the development of academic networks.

5) Co-purchase datasets

Amazon-Computers [161] and Amazon-Photo [161] are subsets of the Amazon co-purchase graph. Products are represented as nodes and frequent co-purchase relationships are represented as edges. These datasets provide valuable resources for research and applications in the e-commerce field, facilitating the mining of association rules between products, prediction of user behavior [174], and market trends [175].

6) Social datasets

Flickr [162], Weibo [163], YouTube [164], BlogCatalog [165], and Github [166] are social networks where nodes represent individual users and edges represent relationships between different users. Flickr [162] is an online photo management and sharing dataset, where the interests of users are identified with node labels. Weibo [163] is a platform where users can post topic tags, with edge weights representing the number of specific tags posted by users and temporal information marking the users. YouTube [164] is a social network of YouTube users, with labels indicating groups of users who favor similar video types. BlogCatalog [165] represents a social relation network of bloggers, where labels infer the interests of bloggers from the metadata provided. In Github [166], node features are detailed information about developers, edges represent that two people follow each other on Github, and node labels show the areas in which developers are mainly engaged.

7) Financial datasets

DGraphFin [106] is a real financial dataset where labels indicate whether a user has defaulted or not. Rich information and contexts are provided by these datasets, which can be utilized for tasks such as social network analysis [176], default prediction [31], and understanding user behavior [174]. Elliptic++ [167] is a cryptocurrency graph dataset consisting of approximately 203,000 Bitcoin transaction records and 822,000 wallet addresses. By leveraging the underlying graph structure, this dataset enables the detection of fraudulent transactions and illicit addresses (actors) within the Bitcoin network.

4.2. Evaluation metrics

4.2.1. Performance evaluation metrics

The evaluation metrics commonly used in imbalanced node classification are summarized in this section.

Appropriate evaluation metrics are crucial for optimizing model performance [177]. Using a binary classification problem as an example, the concepts can be extended to multi-class classification problems. The correct or incorrect classification results for each class are recorded in the confusion matrix, as shown in Table 3. In these binary classification problems, the minority class is defined as the positive class and the majority class as the negative class. True Positive (TP) and True Negative (TN) denote the number of correctly classified positive and negative instances, respectively. False Positive (FP) and False Negative (FN) represent the misclassified positive and negative instances.

Table 3. Confusion matrix for a two-class problem.

	Positive prediction	Negative prediction
Positive class	True positive (TP)	False negative (FN)
Negative class	False positive (FP)	True negative (TN)

1) *ACC*

Accuracy [178] is a typical method for evaluating classifier performance, representing the proportion of correctly classified samples to the total number of samples in the test set. It is one of the most commonly used evaluation metrics in classification tasks. The calculation formula is as follows:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}. \tag{4.1}$$

ACC is affected by the model tends to predict the majority class in imbalanced settings. Thus, it is crucial to combine accuracy with other evaluation criteria in real-world applications.

2) *bACC*

Balanced Accuracy [75] is the arithmetic mean of sensitivity and specificity. It takes into account the differences in sample sizes between classes, making it suitable for handling imbalanced datasets. The calculation method is as follows:

$$bACC = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{FP + TN}\right). \tag{4.2}$$

The *bACC* better reflects the performance of the classifier for each category and avoids bias in most categories.

3) *GM*

Geometric mean [179] is the geometric average of *Precision* and *Recall*, used to balance the impact of both metrics. *Precision* is the proportion of true positives among the samples predicted as positive [180]. *Recall* represents the proportion of actual positive samples correctly predicted by the model among all actual positives [180].

$$Precision = \frac{TP}{TP + FP}, \tag{4.3}$$

$$Recall = \frac{TP}{TP + FN}, \tag{4.4}$$

$$GM = \sqrt{Precision \times Recall}. \tag{4.5}$$

Compared to accuracy, the *GM* can better assess the performance of classification methods on imbalanced datasets. It has become one of the common evaluation metrics in the field of imbalanced data processing.

4) $F1 - measure$

$F1 - measure$ [181] is the weighted average of precision and recall, considering the classification accuracy of the model on both positive and negative examples. It effectively evaluates the performance of the model when there is a significant difference in the number of positive and negative samples. The calculation formula is as follows:

$$F1 - measure = \frac{2TP}{2TP + FN + FP}. \tag{4.6}$$

The value range of $F1$ is $[0, 1]$, which is suitable for datasets with a contradiction between accuracy and recall rate.

5) $AUC - ROC$

Area under the receiver operating characteristic curve describes the relationship between the true positive rate (TPR or *recall*) and the false positive rate (FPR) under different thresholds for a classification model [182]. TPR represents the proportion of actual positive samples that are correctly predicted by the model, while FPR indicates the proportion of negative samples that are incorrectly classified as positive. The ROC curve is plotted with FPR on the x-axis and TPR on the y-axis, where each point on the curve corresponds to the TPR and FPR values at different classification thresholds.

$$TPR = \frac{TP}{TP + FN}, \tag{4.7}$$

$$FPR = \frac{FP}{FP + TN}. \tag{4.8}$$

The $AUC - ROC$ measures the overall performance of the model across various thresholds. A higher $AUC - ROC$ value indicates better model performance.

6) MCC

The range of *Matthews correlation coefficient* [183] is $[-1, 1]$. A higher MCC value indicates better predictive performance, while a lower or negative value suggests poorer performance or significant bias in the predictions. It can be calculated using the following formula:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \tag{4.9}$$

Selecting appropriate metrics is critical in imbalanced scenarios [184, 185], as performance metrics can be deceptive. The most significant pitfall lies in ACC, which yields misleadingly high scores by simply predicting the majority class, masking the model's failure to detect minority nodes; thus, it should never be used as the sole metric. While $AUC - ROC$ effectively assesses ranking quality, it can be overly optimistic when the majority class is dominant, as the large number of TN suppresses the FPR. Regarding the $F1$-*measure*, when dealing with multiclass classification, it is not recommended to directly compute the $F1$-*measure* (*micro*-$F1$) [186] using the overall sample. Instead, the $F1$-*measure* should be calculated independently for each class, and then averaged (*macro*-$F1$) [186]. This is because *micro*-$F1$ is heavily biased toward the majority class, exhibiting behavior similar to ACC. MCC, GM, and $bACC$ are recommended as robust benchmarks. MCC, in particular, is regarded as a balanced measure even if classes are of very different sizes, ensuring that a high score reflects accurate predictions for both majority and minority classes.

4.2.2. Homophily-aware measures

Homophily refers to the tendency for nodes in a graph to connect with nodes bearing the same label [187]. In the context of imbalanced node classification, the interaction between category distribution and structural topology is crucial. For instance, minority class nodes residing in heterophilous neighborhoods are susceptible to being overwhelmed by majority class neighbors, leading to the neighborhood assimilation problem. Therefore, relying solely on predictive performance metrics is insufficient to fully characterize the difficulty of a learning task. To comprehensively evaluate structural bias and topological difficulty, introducing homophily-aware metrics is essential.

We first define the symbols used to evaluate homophily. Given an undirected, unweighted graph $G = (V, E)$ with $|V| = n$ nodes and $|E|$ edges. Let y_v denote the label of node v, $N(v)$ be the set of neighbors of v, and $d(v) = |N(v)|$ be the degree of v. Finally, let D_k denote the total degree of the k_{th} class, namely $D_k = \sum_{v:y_v=k} d(v)$.

1) *Edge homophily*

Edge homophily [188] measures the proportion of edges connecting nodes within the same category. The calculation formula is as follows:

$$h_{edge} = \frac{|\{\{u, v\} \in E : y_u = y_v\}|}{|E|}. \tag{4.10}$$

2) *Node homophily*

Node homophily [189] calculates the homophily of each node, then takes the average of all nodes' values:

$$h_{node} = \frac{1}{n} \sum_{v \in V} \frac{|\{u \in N(v) : y_u = y_v\}|}{d(v)}. \tag{4.11}$$

3) *Assortativity coefficient*

Assortativity coefficient [190] is defined as

$$h = \frac{h_{edge} - \sum_{k=1}^{m} D_k^2 / (2|E|)^2}{1 - \sum_{k=1}^{m} D_k^2 / (2|E|)^2}. \tag{4.12}$$

It is important to note that h_{edge} is highly sensitive to class imbalance and is typically dominated by the majority class. Both h_{node} and h represent global average metrics, which may obscure the extreme heterophily faced by minority class nodes. Therefore, a high homophily score does not necessarily guarantee superior classification performance [90].

4.3. Comparative results and setup configurations

To provide more intuitive references for model selection and enhance the practical value of this survey, Tables 4 and 5 present quantitative performance comparisons of representative methods on *F*1*-measure* and *ACC*. All methods use a two-layer GCN [80] backbone and hidden layer dimension of 128. The experimental setup includes the Adam optimizer (learning rate: 0.001, weight decay: 5e-4) and setting dropout to 0.3.

To facilitate reproducibility and practical application, access to standardized codebases and experimental protocols is essential. We summarize the recommended hyperparameter ranges and available code resources for several representative methods in Table 6. In terms of model implementation, general-purpose libraries such as PyG [191] and DGL [192] provide foundational support. Additionally, specialized resources like IGL-Bench [193] serve as the first open-source benchmark platform dedicated to graph-specific imbalanced learning.

Table 4. Quantitative comparison of representative methods in terms of ACC.

Methods\Datasets	Cora [156]	CiteSeer [156]	PubMed [80]	Computers [161]	Photo [161]
LTE4G [63]	80.85	63.85	83.15	78.90	88.65
ReNode [65]	82.15	71.55	83.92	80.15	90.12
Tail-GNN [70]	79.95	70.65	84.45	86.85	92.88
GraphSMOTE [73]	77.32	53.25	71.35	76.88	88.55
TAM [75]	80.80	64.25	81.55	81.15	90.52
DR-GCN [101]	71.65	54.85	64.12	69.25	86.95
GraphENS [94]	79.52	64.25	80.90	80.95	90.55
ImGAGN [99]	74.55	54.15	72.85	75.65	82.25

Table 5. Quantitative comparison of representative methods in terms of $F1\text{-}measure$.

Methods\Datasets	Cora [156]	CiteSeer [156]	PubMed [80]	Computers [161]	Photo [161]
LTE4G [63]	76.55	58.50	81.25	68.45	84.22
ReNode [65]	81.98	70.32	83.06	70.92	88.45
Tail-GNN [70]	77.40	65.98	84.30	72.60	89.55
GraphSMOTE [73]	72.08	51.25	68.95	55.48	78.80
TAM [75]	79.40	61.98	78.31	78.60	85.55
DR-GCN [101]	66.52	51.35	55.62	46.12	77.29
GraphENS [94]	78.30	59.92	82.08	76.45	85.70
ImGAGN [99]	68.65	51.75	72.98	69.85	75.83

5. Future research directions for imbalanced node classification on graphs

Our research shows that imbalanced node classification tasks on graphs confront multiple complicated challenges, due to the complexity of graphs and the infancy of graph learning. Additionally, there is a lack of study on the graph and edge levels in existing classification tasks on imbalanced datasets, which mostly concentrate on the node level. To further explore the classification tasks on imbalanced graphs, three future research directions are proposed.

5.1. Tasks at different levels - edge, graph

The tasks of edge- and graph-level have extensive applications in real scenarios, such as network security [194] and traffic control [195]. Although research in these areas is relatively sparse, it should not be overlooked.

Table 6. Hyperparameter search spaces and code resources for representative imbalanced graph learning algorithms.

Algorithm	Hyperparameter	Search space	Code
LLM4NG [117]	δ for the similarity threshold	0.1–0.8	Link
	k for the top-k edges	10, 50, 100, 300	
LTE4G [63]	α for focal loss	0.5, 0.6, 0.7, 0.8	Link
	γ for curve shape control	0, 1, 2	
ReNode [65]	PageRank teleport probability	0.05, 0.1, 0.15	Link
	lower bound of reweighting	0.25, 0.5, 0.75	
	upper bound of reweighting	1.25, 1.5, 1.75	
LLM-HetGDT [119]	δ for the node embedding weight	1e-2, 5e-2	Link
	λ for the weight loss coefficient	1e-3, 2e-3	
Tail-GNN [70]	μ for $\mathcal{L}_m$	0.01, 0.001	Link
	η for $\mathcal{L}_d$	0.1, 1.0	
GraphSMOTE [73]	λ for $\mathcal{L}_{edge}$	1e-6, 2e-6, 3e-6	Link
TAM [75]	ϕ for the class-wise temperature	0.8, 1.0, 1.2	Link
	α for the ACM term of node v	0.25, 0.5, 1.5	
	β for the ADM term of node v	0.125, 0.25, 0.5	
GATE-GNN [153]	λ for the weight loss coefficient	1e-6, 2e-6, 3e-6	Link
DR-GCN [101]	α for loss trade-off	0.5, 0.6, 0.7, 0.8	Link
GraphENS [94]	number of warming up epochs	1, 5	Link
	k for feature masking	1, 5, 10	
	τ for temperature	1, 2	
ImGAGN [99]	λ_1 for minority nodes ratio	0.2, 0.3, 0.4, 0.5	Link
	λ_2 for discriminator training steps	30, 40, 50, 60	

5.1.1. Learning tasks at the edge-level

In class-imbalanced scenarios, edge learning aims to predict the connection of nodes, categorize edges in a graph, and address problems caused by uneven edge distribution [196]. While edges between nodes in the majority class have a high connectivity density, those in the minority class have sparse connectivity [197]. This creates a unique structural sparsity problem where traditional link prediction methods tend to suppress connections involving minority nodes due to their low prior probability. Jo et al. [198] proposed the dual hypergraph transformation (DHT), in which original edges are transformed into nodes of a hypergraph, and message-passing of node representation are applied to the edges. After obtaining edge representations from the hypergraph, clustering or edge deletion is performed to obtain an overall representation. Future research should improve the edge pooling methods and extend them to graph- or node-level tasks. Topological entropy (TE) was introduced by Cheng et al. [199] to measure the topological imbalance of each edge. Topological reweighting and TE wedge-based mixup are developed and integrated into a new classification of edge strategies, referred to as TopoEdge. In the future, the impact of topological imbalance can be further quantified and other methods of measuring topological imbalance can be explored. A study on the topological imbalance in different types of graphs could be conducted, and the applicability of TE in different scenarios could be examined.

5.1.2. Learning tasks at the graph-level

An important learning task at the graph-level is the graph classification, which needs to consider the global information of graphs, including the structure of graphs and the attributes of nodes [200]. Given multiple graphs and their corresponding labels, the task involves learning a model that maps graphs to their respective labels. Node imbalance can be mitigated via propagating information from neighboring nodes to gain additional supervision. However, graphs are isolated with no interconnections between them. Therefore, learning an effective global representation of a graph is a challenge.

The graph-of-graph neural networks (G^2GNN) method proposed by [201] aggregates globally neighboring graph representations and generates locally random augmentations for each graph. Future graph classification methods can enhance the G^2GNN method to accommodate varying graph scales. For instance, incorporating attention mechanisms into G^2GNN propagation [202] adaptively aggregates information based on the topological similarity of neighboring graphs, thereby enhancing the application of graph-level tasks. To address the limitation of traditional GNNs in modeling higher-order dependencies between nodes, the hypergraph-enhanced dual framework (HEAL) was proposed by Ju et al. [203]. In HEAL, a low-rank strategy is employed to model the hypergraph structure matrix, while hypergraph convolution is designed to learn node representations. A line graph is then introduced to identify potential semantic structures within the hypergraph. Finally, graph representations from both perspectives are integrated to fully utilize unlabeled graphs and better optimize the model results. A promising direction is to improve the adaptivity and computational efficiency of hypergraph for large graph datasets. Meanwhile, integrating more information in hypergraphs to enhance expressiveness would be another research direction.

5.2. Node classification on complex graphs

The majority of imbalanced node classification on graphs focuses on simple graphs, while complex graphs are prevalent in practical applications. Dynamic and heterogeneous graphs are two types of complex graphs that can be influenced by imbalance.

5.2.1. Node classification on dynamic graphs

The nodes and edges change over time in dynamic graphs. In the context of imbalance, this manifests as a temporal distribution shift, where the imbalance ratio fluctuates over time [43] (e.g., a burst of minority class events), rendering static re-balancing strategies ineffective and hindering effective node classification. Data is continuously generated as streams in dynamic graphs. There are two tasks that must be considered. The first is the spatiotemporal information that is contained in each graph snapshot in different periods. The second is the evolution patterns of nodes and edges and the connections among their attributes.

Incremental learning [204] or online learning [205] offer promising avenues research. These methods allow the model to continuously receive new data and update its parameters while preserving the original graph information [206], thereby accommodating the dynamic changes of graphs. Both of these methods require far less time and computational cost by updating a few parameters instead of retraining the whole model.

5.2.2. Node classification on large-scale graphs

The scalability of node classification methods for high-dimensional, large-scale data has long been a major challenge. Large databases like DGraphFin [106] contain millions of users and connections, exhibiting substantial data scale in both graph size and node attributes. Some studies reduce graph structure through sampling to enhance learning efficiency [207]. However, this may inadvertently filter out sparse minority nodes, compromising structural integrity. Other research has redesigned node information propagation in large-scale graphs, proposing adaptive topology-aware propagation optimization strategies [208].

Future directions may explore hierarchical subgraph sampling or class-balanced partitioning algorithms to ensure minority nodes are retained in each training mini-batch without compromising training speed. Additionally, decoupling representation learning from classifier training (e.g., using precomputed embeddings) may offer a viable approach for large-scale imbalanced learning.

5.2.3. Node classification on heterogeneous graphs

In a homogeneous graph, one type of node and edge exists [209]. However, a heterogeneous graph has multiple types of nodes and edges, which permits different kinds of nodes to have features with various dimensions [210]. Learning on heterogeneous graphs presents the following challenges. First, heterogeneous graphs comprise multiple types of entities and relationships [211]. Most solutions [212, 213] rely on metapaths to show the complex structure of heterogeneous graphs, but it is difficult to capture information comprehensively, which can easily lead to the omission of semantic information [214]. Second, heterogeneous graphs consist of multiple relationships, each of which represents a different aspect. The different connections and node features drive the formation of diverse underlying graphs [209]. The efficiency of learning graphs is hampered if these interactions are trained uniformly [215].

Future studies can first distinguish meta paths based on their features [216], and then learn node embeddings for each of these meta paths separately. Higher-order neighbor information should be effectively utilized during the message propagation process [217]. Heterogeneous graphs and GNN parameters can be integrated to learn the classification objective in the training process. The internal relationships within the heterogeneous graph can be leveraged to generate feature similarity graphs, feature propagation graphs, and semantic graphs, allowing the optimal graph structure to be learned.

5.3. Node classification with topological imbalance

Unlike quantity imbalance which focuses on label ratios, topological imbalance refers to the structural asymmetry where nodes of different classes occupy distinct topological positions within the graph. This phenomenon presents two specific unresolved challenges. First, A minority of nodes typically reside near the periphery of the graph or close to class boundaries, while the majority often occupy central hubs. This structural bias can lead to learning decision boundary drift [65]. Second, When minority-class nodes are surrounded by majority-class neighbors, their distinguishing features can be overwhelmed by majority signals during aggregation. This leads to indistinguishable node embeddings and information inadequacy [75]. Current research on imbalance mainly focuses on quantity imbalance, thus it is crucial to thoroughly investigate topological imbalance and create suitable solutions.

Although ReNode [65] and TAM [75] shown promising outcomes for topological imbalance, the impact is less evident in situations when there is a low labeling ratio and inadequate connection. TopoEdge [199] uses TE to effectively measure the local class distribution variance on edges, and subsequent researchers can apply the idea to the domain of node topological imbalance. Active learning [218] and self-training [111] can be explored in the learning of labeled nodes, generating representations for unsupervised nodes, achieving a balanced labeled dataset. These methods can be used for node-, edge-, and graph-level tasks. Additionally, the characteristics of datasets in special scenarios, such as biomedical applications [219], should be considered.

6. Conclusions

This paper provides a comprehensive review of GNNs for classifying imbalanced nodes. We systematically review the evolution of GNN-based approaches, categorizing them into data-level methods (such as interpolation, generation, pseudo-labeling, and LLM methods) and algorithm-level strategies (such as loss modification, enhancement, and model optimization). Beyond hierarchical classification, we conduct an in-depth comparative analysis to highlight the inherent trade-offs among these techniques, revealing how the generation method brings the risk of topological noise while providing semantic richness, whereas algorithmic adjustments preserve structural integrity but may struggle with information sparsity. Looking forward, the field stands at a juncture where research must transcend simple data rebalancing. Future endeavors should focus on tackling complex challenges in dynamic and heterogeneous graph structures, explicitly addressing interactions between inherent graph properties like imbalance and heterophily and leveraging the semantic reasoning capabilities of emerging LLMs to overcome data scarcity. We hope this review serves not only as a reference for existing techniques but also as a roadmap to inspire the emergence of robust, scalable, and topology-aware solutions for learning on imbalanced graphs.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research was supported by the Major Science and Technology Project of Henan Province (241100210100), the Key Science and Technology Program of Henan Province (242102211068, 252102211083), the Natural Science Foundation of Henan Province (252300420392), the Key Scientific Research Projects of Henan Higher School (No.25B520013).

Conflict of interest

The authors declare there is no conflicts of interest.

References

1. Z. Zhang, P. Cui, W. Zhu, Deep learning on graphs: A survey, *IEEE Trans. Knowl. Data Eng.*, **34** (2020), 249–270. https://doi.org/10.1109/TKDE.2020.2981333

2. X. Wang, D. Bo, C. Shi, S. Fan, Y. Ye, S. Y. Philip, A survey on heterogeneous graph embedding: Methods, techniques, applications and sources, *IEEE Trans. Big Data*, **9** (2022), 415–436. https://doi.org/10.1109/TBDATA.2022.3177455

3. S. Bhagat, G. Cormode, S. Muthukrishnan, Node classification in social networks, in *Social Network Data Analytics*, (2011), 115–148. https://doi.org/10.1007/978-1-4419-8462-3_5

4. S. E. Schaeffer, Graph clustering, *Comput. Sci. Rev.*, **1** (2007), 27–64. https://doi.org/10.1016/j.cosrev.2007.05.001

5. N. Hafiene, W. Karoui, L. B. Romdhane, Influential nodes detection in dynamic social networks: A survey, *Expert Syst. Appl.*, **159** (2020), 113642. https://doi.org/10.1016/j.eswa.2020.113642

6. M. Yasunaga, H. Ren, A. Bosselut, P. Liang, J. Leskovec, QA-GNN: Reasoning with language models and knowledge graphs for question answering, preprint, arXiv:2104.06378.

7. Y. Apedo, H. Tao, A weakly supervised pavement crack segmentation based on adversarial learning and transformers, *Multimedia Syst.*, **31** (2025), 266. https://doi.org/10.1007/s00530-025-01850-1

8. K. Zhang, W. Luo, Y. Zhong, L. Ma, W. Liu, H. Li, Adversarial spatio-temporal learning for video deblurring, *IEEE Trans. Image Process.*, **28** (2018), 291–301. https://doi.org/10.1109/TIP.2018.2867733

9. Y. Wen, T. Gao, Z. Li, J. Zhang, K. Zhang, T. Chen, All-in-one weather-degraded image restoration via adaptive degradation-aware self-prompting model, *IEEE Trans. Multimedia*, **27** (2025), 3343–3355. https://doi.org/10.1109/TMM.2025.3535316

10. J. Zhang, W. Chen, J. Zhang, UAV-based quantitative crack measurement for bridges integrating four-point laser metric calibration and mamba segmentation, *Autom. Constr.*, **182** (2026), 106774. https://doi.org/10.1016/j.autcon.2026.106774

11. L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, et al., Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions, *J. Big Data*, **8** (2021), 1–74. https://doi.org/10.1186/s40537-021-00444-8

12. A. Aslam, E. Curry, Towards a generalized approach for deep neural network based event processing for the internet of multimedia things, *IEEE Access*, **6** (2018), 25573–25587. https://doi.org/10.1109/ACCESS.2018.2823590

13. N. Chen, B. Li, Y. Wang, X. Ying, L. Wang, C. Zhang, et al., Motion and appearance decoupling representation for event cameras, *IEEE Trans. Image Process.*, **34** (2025), 5964–5977. https://doi.org/10.1109/TIP.2025.3607632

14. C. Deng, Y. Li, F. Xiong, H. Liu, X. Li, Y. Zeng, Detection of rupture damage degree in laminated rubber bearings using a piezoelectric-based active sensing method and hybrid machine learning algorithms, *Struct. Control Health Monit.*, **2025** (2025), 6694610. https://doi.org/10.1155/stc/6694610

15. J. Koutnik, K. Greff, F. Gomez, J. Schmidhuber, A clockwork RNN, in *International Conference on Mmachine Learning*, (2014), 1863–1871.

16. K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, J. Schmidhuber, LSTM: A search space odyssey, *IEEE Trans. Neural Networks Learn. Syst.*, **28** (2016), 2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924

17. J. Devlin, M. W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, preprint, arXiv:1810.04805.

18. Y. Yin, F. Meng, J. Su, C. Zhou, Z. Yang, J. Zhou, et al., A novel graph-based multi-modal fusion encoder for neural machine translation, preprint, arXiv:2007.08742.

19. Z. Jin, M. Tao, X. Zhao, Y. Hu, Social media sentiment analysis based on dependency graph and co-occurrence graph, *Cognit. Comput.*, **14** (2022), 1039–1054. https://doi.org/10.1007/s12559-022-10004-8

20. P. Li, H. Tao, H. Zhou, P. Zhou, Y. Deng, Enhanced multiview attention network with random interpolation resize for few-shot surface defect detection, *Multimedia Syst.*, **31** (2025), 36. https://doi.org/10.1007/s00530-024-01643-y

21. Z. Wang, H. Tao, H. Zhou, Y. Deng, P. Zhou, A content-style control network with style contrastive learning for underwater image enhancement, *Multimedia Syst.*, **31** (2025), 60. https://doi.org/10.1007/s00530-024-01642-z

22. J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, et al., A survey of uncertainty in deep neural networks, *Artif. Intell. Rev.*, **56** (2023), 1513–1589. https://doi.org/10.1007/s10462-023-10562-9

23. G. Corso, H. Stark, S. Jegelka, T. Jaakkola, R. Barzilay, Graph neural networks, *Nat. Rev. Methods Primers*, **4** (2024), 17. https://doi.org/10.1038/s43586-024-00294-7

24. S. Xiao, J. Li, J. Lu, S. Huang, B. Zeng, S. Wang, Graph neural networks for multi-view learning: A taxonomic review, *Artif. Intell. Rev.*, **57** (2024), 341. https://doi.org/10.1007/s10462-024-10990-1

25. P. Zhang, J. Chen, C. Che, L. Zhang, B. Jin, Y. Zhu, IEA-GNN: Anchor-aware graph neural network fused with information entropy for node classification and link prediction, *Inf. Sci.*, **634** (2023), 665–676. https://doi.org/10.1016/j.ins.2023.03.022

26. K. Wang, Y. Ding, S. C. Han, Graph neural networks for text classification: A survey, *Artif. Intell. Rev.*, **57** (2024), 190. https://doi.org/10.1007/s10462-024-10808-0

27. X. Ma, J. Wu, S. Xue, J. Yang, C. Zhou, Q. Z. Sheng, et al., A comprehensive survey on graph anomaly detection with deep learning, *IEEE Trans. Knowl. Data Eng.*, **35** (2021), 12012–12038. https://doi.org/10.1109/TKDE.2021.3118815

28. X. Guo, L. Zhao, A systematic survey on deep generative models for graph generation, *IEEE Trans. Pattern Anal. Mach. Intell.*, **45** (2022), 5370–5390. https://doi.org/10.1109/TPAMI.2022.3214832

29. 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*, (2022), 519–535. https://doi.org/10.1007/978-3-031-26412-2_32

30. A. Taherkhani, G. Cosma, T. M. McGinnity, Adaboost-CNN: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning, *Neurocomputing*, **404** (2020), 351–366. https://doi.org/10.1016/j.neucom.2020.03.064

31. M. Zhou, Z. Gong, GraphSR: A data augmentation algorithm for imbalanced node classification, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **37** (2023), 4954–4962. https://doi.org/10.1609/aaai.v37i4.25622

32. M. Ghorbani, A. Kazi, M. S. Baghshah, H. R. Rabiee, N. Navab, RA-GCN: Graph convolutional network for disease prediction problems with imbalanced data, *Med. Image Anal.*, **75** (2022), 102272. https://doi.org/10.1016/j.media.2021.102272

33. W. Liang, F. Lv, L. Chen, H. Tao, M. Shi, X. Zhu, et al., Citywide multi-step crime prediction via context-aware bayesian tensor decomposition, *IEEE Trans. Inf. Forensics Secur.*, **20** (2025), 8030–8042. https://doi.org/10.1109/TIFS.2025.3594131

34. L. He, G. Xu, S. Jameel, X. Wang, H. Chen, Graph-aware deep fusion networks for online spam review detection, *IEEE Trans. Comput. Soc. Syst.*, **10** (2023), 2557–2565. https://doi.org/10.1109/TCSS.2022.3189813

35. Z. Zhu, H. Xing, Y. Xu, Balanced neighbor exploration for semi-supervised node classification on imbalanced graph data, *Inf. Sci.*, **631** (2023), 31–44. https://doi.org/10.1016/j.ins.2023.02.064

36. C. Lv, X. Lv, Z. Wang, T. Zhao, W. Tian, Q. Zhou, et al., A focal quotient gradient system method for deep neural network training, *Appl. Soft Comput.*, **184** (2025), 113704. https://doi.org/10.1016/j.asoc.2025.113704

37. H. Cai, V. W. Zheng, K. C. C. Chang, A comprehensive survey of graph embedding: Problems, techniques, and applications, *IEEE Trans. Knowl. Data Eng.*, **30** (2018), 1616–1637. https://doi.org/10.1109/TKDE.2018.2807452

38. P. Branco, L. Torgo, R. P. Ribeiro, A survey of predictive modeling on imbalanced domains, *ACM Comput. Surv.*, **49** (2016), 1–50. https://doi.org/10.1145/2907070

39. Y. Zhang, B. Kang, B. Hooi, S. Yan, J. Feng, Deep long-tailed learning: A survey, *IEEE Trans. Pattern Anal. Mach. Intell.*, **45** (2023), 10795–10816. https://doi.org/10.1109/TPAMI.2023.3268118

40. S. Wu, F. Sun, W. Zhang, X. Xie, B. Cui, Graph neural networks in recommender systems: A survey, *ACM Comput. Surv.*, **55** (2022), 1–37. https://doi.org/10.1145/3535101

41. Z. Zhuang, K. M. Ting, G. Pang, S. Song, Subgraph centralization: A necessary step for graph anomaly detection, in *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, SIAM, (2023), 703–711. https://doi.org/10.1137/1.9781611977653.ch79

42. C. Gao, X. Wang, X. He, Y. Li, Graph neural networks for recommender system, in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, (2022), 1623–1625. https://doi.org/10.1145/3488560.3501396

43. W. Jiang, J. Luo, Graph neural network for traffic forecasting: A survey, *Expert Syst. Appl.*, **207** (2022), 117921. https://doi.org/10.1016/j.eswa.2022.117921

44. Y. Ma, Y. Tian, N. Moniz, N. V. Chawla, Class-imbalanced learning on graphs: A survey, preprint, arXiv:2304.04300.

45. S. Maekawa, K. Noda, Y. Sasaki, Beyond real-world benchmark datasets: An empirical study of node classification with GNNs, in *Advances in Neural Information Processing Systems*, **35** (2022), 5562–5574.

46. S. Xiao, S. Wang, Y. Dai, W. Guo, Graph neural networks in node classification: Survey and evaluation, *Mach. Vision Appl.*, **33** (2022), 4. https://doi.org/10.1007/s00138-021-01251-0

47. Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S. Y. Philip, A comprehensive survey on graph neural networks, *IEEE Trans. Neural Networks Learn. Syst.*, **32** (2020), 4–24. https://doi.org/10.1109/TNNLS.2020.2978386

48. H. Yuan, H. Yu, S. Gui, S. Ji, Explainability in graph neural networks: A taxonomic survey, *IEEE Trans. Pattern Anal. Mach. Intell.*, **45** (2022), 5782–5799. https://doi.org/10.1109/TPAMI.2022.3204236

49. Y. Xie, Z. Xu, J. Zhang, Z. Wang, S. Ji, Self-supervised learning of graph neural networks: A unified review, *IEEE Trans. Pattern Anal. Mach. Intell.*, **45** (2022), 2412–2429. https://doi.org/10.1109/TPAMI.2022.3170559

50. X. Li, Z. Fan, F. Huang, X. Hu, Y. Deng, L. Wang, et al., Graph neural network with curriculum learning for imbalanced node classification, *Neurocomputing*, **574** (2024), 127229. https://doi.org/10.1016/j.neucom.2023.127229

51. J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei, Line: Large-scale information network embedding, in *Proceedings of the 24th International Conference on World Wide Web*, (2015), 1067–1077. https://doi.org/10.1145/2736277.2741093

52. W. L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, C. J. Hsieh, Cluster-GCN: An efficient algorithm for training deep and large graph convolutional networks, in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, (2019), 257–266. https://doi.org/10.1145/3292500.3330925

53. Z. A. Shen, T. Luo, Y. K. Zhou, H. Yu, P. F. Du, NPI-GNN: Predicting ncRNA–protein interactions with deep graph neural networks, *Briefings Bioinf.*, **22** (2021), bbab051. https://doi.org/10.1093/bib/bbab051

54. A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, et al., Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles, *Proc. Natl. Acad. Sci. U.S.A.*, **102** (2005), 15545–15550. https://doi.org/10.1073/pnas.0506580102

55. J. Xu, Y. Li, Discovering disease-genes by topological features in human protein–protein interaction network, *Bioinformatics*, **22** (2006), 2800–2805. https://doi.org/10.1093/bioinformatics/btl467

56. S. Min, Z. Gao, J. Peng, L. Wang, K. Qin, B. Fang, Stgsn—a spatial–temporal graph neural network framework for time-evolving social networks, *Knowledge-Based Syst.*, **214** (2021), 106746. https://doi.org/10.1016/j.knosys.2021.106746

57. Z. Guo, K. Yu, A. Jolfaei, G. Li, F. Ding, A. Beheshti, Mixed graph neural network-based fake news detection for sustainable vehicular social networks, *IEEE Trans. Intell. Transp. Syst.*, **24** (2022), 15486–15498. https://doi.org/10.1109/TITS.2022.3185013

58. X. Li, L. Sun, M. Ling, Y. Peng, A survey of graph neural network based recommendation in social networks, *Neurocomputing*, **549** (2023), 126441. https://doi.org/10.1016/j.neucom.2023.126441

59. S. Xu, X. Liu, K. Ma, F. Dong, B. Riskhan, S. Xiang, et al., Rumor detection on social media using hierarchically aggregated feature via graph neural networks, *Appl. Intell.*, **53** (2023), 3136–3149. https://doi.org/10.1007/s10489-022-03592-3

60. K. Sharma, Y. C. Lee, S. Nambi, A. Salian, S. Shah, S. W. Kim, et al., A survey of graph neural networks for social recommender systems, *ACM Comput. Surv.*, **56** (2024), 1–34. https://doi.org/10.1145/3661821

61. J. He, S. Xiao, S. Huang, J. Li, D. Wang, Mokan: A multi-omics data analysis framework using kolmogorov-arnold networks, *IEEE J. Biomed. Health Inf.*, **30** (2026), 1830–1840. https://doi.org/10.1109/JBHI.2025.3596817

62. Z. Huang, Y. Tang, Y. Chen, A graph neural network-based node classification model on class-imbalanced graph data, *Knowledge-Based Syst.*, **244** (2022), 108538. https://doi.org/10.1016/j.knosys.2022.108538

63. S. Yun, K. Kim, K. Yoon, C. Park, Lte4g: Long-tail experts for graph neural networks, in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, (2022), 2434–2443. https://doi.org/10.1145/3511808.3557381

64. Z. Zhang, Y. Zhang, M. Dong, K. Ota, Y. Zhang, Y. Ren, Collaborative tag-aware graph neural network for long-tail service recommendation, *IEEE Trans. Serv. Comput.*, **3174** (2024), 2124–2138. https://doi.org/10.1109/TSC.2024.3349853

65. D. Chen, Y. Lin, G. Zhao, X. Ren, P. Li, J. Zhou, et al., Topology-imbalance learning for semi-supervised node classification, in *Advances in Neural Information Processing Systems*, **34** (2021), 29885–29897.

66. W. Liu, M. Gong, Z. Tang, A. K. Qin, K. Sheng, M. Xu, Locality preserving dense graph convolutional networks with graph context-aware node representations, *Neural Networks*, **143** (2021), 108–120. https://doi.org/10.1016/j.neunet.2021.05.031

67. J. Jusko, M. Rehak, Identifying peer-to-peer communities in the network by connection graph analysis, *Int. J. Network Manage.*, **24** (2014), 235–252. https://doi.org/10.1002/nem.1862

68. W. Yu, X. Lin, J. Liu, J. Ge, W. Ou, Z. Qin, Self-propagation graph neural network for recommendation, *IEEE Trans. Knowl. Data Eng.*, **34** (2021), 5993–6002. https://doi.org/10.1109/TKDE.2021.3076772

69. W. D. G. de Oliveira, L. Berton, A systematic review for class-imbalance in semi-supervised learning, *Artif. Intell. Rev.*, **56** (2023), 2349–2382. https://doi.org/10.1007/s10462-023-10579-0

70. Z. Liu, T. K. Nguyen, Y. Fang, Tail-GNN: Tail-node graph neural networks, in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, (2021), 1109–1119.

71. S. Shi, K. Qiao, S. Yang, L. Wang, J. Chen, B. Yan, Boosting-GNN: Boosting algorithm for graph networks on imbalanced node classification, *Front. Neurorob.*, **15** (2021), 775688. https://doi.org/10.3389/fnbot.2021.775688

72. R. M. Pereira, Y. M. Costa, C. N. Silla Jr, Toward hierarchical classification of imbalanced data using random resampling algorithms, *Inf. Sci.*, **578** (2021), 344–363. https://doi.org/10.1016/j.ins.2021.07.033

73. 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), 833–841. https://doi.org/10.1145/3437963.3441720

74. Y. Qian, C. Zhang, Y. Zhang, Q. Wen, Y. Ye, C. Zhang, Co-modality graph contrastive learning for imbalanced node classification, in *Advances in Neural Information Processing Systems*, **35** (2022), 15862–15874.

75. J. Song, J. Park, E. Yang, TAM: Topology-aware margin loss for class-imbalanced node classification, in *International Conference on Machine Learning*, (2022), 20369–20383.

76. 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*, **34** (2020), 5892–5899. https://doi.org/10.1609/aaai.v34i04.6048

77. L. Zheng, F. Chao, N. M. Parthaláin, D. Zhang, Q. Shen, Feature grouping and selection: A graph-based approach, *Inf. Sci.*, **546** (2021), 1256–1272. https://doi.org/10.1016/j.ins.2020.09.022

78. D. Cheng, Q. Zhu, J. Huang, L. Yang, Q. Wu, Natural neighbor-based clustering algorithm with local representatives, *Knowledge-Based Syst.*, **123** (2017), 238–253. https://doi.org/10.1016/j.knosys.2017.02.027

79. M. I. Jordan, T. M. Mitchell, Machine learning: Trends, perspectives, and prospects, *Science*, **349** (2015), 255–260. https://doi.org/10.1126/science.aaa8415

80. T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, preprint, arXiv:1609.02907.

81. N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote: Synthetic minority oversampling technique, *J. Artif. Intell. Res.*, **16** (2002), 321–357. https://doi.org/10.1613/jair.953

82. P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, preprint, arXiv:1710.10903. https://doi.org/10.48550/arXiv.1710.10903

83. W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in *Advances in Neural Information Processing Systems*, **30** (2017).

84. K. Xu, W. Hu, J. Leskovec, S. Jegelka, How powerful are graph neural networks, preprint, arXiv:1810.00826. https://doi.org/10.48550/arXiv.1810.00826

85. C. Xu, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, F. Zhuang, et al., Graph contextualized self-attention network for session-based recommendation, in *IJCAI*, **19** (2019), 3940–3946.

86. G. Jin, M. Wang, J. Zhang, H. Sha, J. Huang, STGNN-TTE: Travel time estimation via spatial–temporal graph neural network, *Future Gener. Comput. Syst.*, **126** (2022), 70–81. https://doi.org/10.1016/j.future.2021.07.012

87. W. Jin, K. Yang, R. Barzilay, T. Jaakkola, Learning multimodal graph-to-graph translation for molecular optimization, preprint, arXiv:1812.01070. https://doi.org/10.48550/arXiv.1812.01070

88. F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, *IEEE Trans. Neural Networks*, **20** (2008), 61–80. https://doi.org/10.1109/TNN.2008.2005605

89. S. Akansha, Over-squashing in graph neural networks: A comprehensive survey, *Neurocomputing*, **2025** (2025), 130389. https://doi.org/10.1016/j.neucom.2025.130389

90. S. Luan, C. Hua, Q. Lu, L. Ma, L. Wu, X. Wang, et al., The heterophilic graph learning handbook: Benchmarks, models, theoretical analysis, applications and challenges, preprint, arXiv:2407.09618. https://doi.org/10.48550/arXiv.2407.09618

91. H. Han, W. Y. Wang, B. H. Mao, Borderline-smote: A new over-sampling method in imbalanced data sets learning, in *International Conference on Intelligent Computing*, (2005), 878–887. https://doi.org/10.1007/11538059_91

92. C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem, in *Advances in Knowledge Discovery and Data Mining: 13th Pacific-Asia Conference, PAKDD 2009 Bangkok, Thailand, April 27–30, 2009 Proceedings 13*, Springer, (2009), 475–482. https://doi.org/10.1007/978-3-642-01307-2_43

93. H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, preprint, arXiv:1710.09412. https://doi.org/10.48550/arXiv.1710.09412

94. J. Park, J. Song, E. Yang, Graphens: Neighbor-aware ego network synthesis for class-imbalanced node classification, in *International Conference on Learning Representations*, (2021), 1–20.

95. Y. Gao, S. Zhang, R. Song, GraphBoost: Adaptive boosting node generation for class-imbalanced graphs, in *Long Beach '23: ACM Symposium on Neural Gaze Detection, August 06–10, 2023, Long Beach, CA*, (2023), 1–8.

96. R. Xia, C. Zhang, Y. Zhang, X. Liu, B. Yang, A novel graph oversampling framework for node classification in class-imbalanced graphs, *Sci. China Inf. Sci.*, **67** (2024), 162101. https://doi.org/10.1007/s11432-023-3897-2

97. P. Sun, Y. Du, S. Xiong, Nearest neighbors and density-based undersampling for imbalanced data classification with class overlap, *Neurocomputing*, **609** (2024), 128492. https://doi.org/10.1016/j.neucom.2024.128492

98. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, et al., Generative adversarial networks, *Commun. ACM*, **63** (2020), 139–144. https://doi.org/10.1145/3422622

99. 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), 1390–1398. https://doi.org/10.1145/3447548.3467334

100. Y. Duan, X. Liu, A. Jatowt, H. T. Yu, S. Lynden, K. S. Kim, et al., Anonymity can help minority: A novel synthetic data over-sampling strategy on multi-label graphs, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, (2022), 20–36. https://doi.org/10.1007/978-3-031-26390-3_2

101. M. Shi, Y. Tang, X. Zhu, D. Wilson, J. Liu, Multi-class imbalanced graph convolutional network learning, in *Proceedings of the Twenty-ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*, (2020), 1–7.

102. P. Shamsolmoali, M. Zareapoor, L. Shen, A. H. Sadka, J. Yang, Imbalanced data learning by minority class augmentation using capsule adversarial networks, *Neurocomputing*, **459** (2021), 481–493. https://doi.org/10.1016/j.neucom.2020.01.119

103. B. Liu, C. Zheng, F. Sun, X. Wang, L. Pan, CDCGAN: Class distribution-aware conditional GAN-based minority augmentation for imbalanced node classification, *Neural Networks*, **183** (2025), 106933. https://doi.org/10.1016/j.neunet.2024.106933

104.X. Zhou, P. Shi, Graph neural network with generative adversarial training for node classification on class imbalanced data, *Eng. Appl. Artif. Intell.*, **161** (2025), 112264. https://doi.org/10.1016/j.engappai.2025.112264

105.F. Zeng, W. Gan, P. S. Yu, GraphIFE: Rethinking graph imbalance node classification via invariant learning, preprint, arXiv:2509.23616. https://doi.org/10.48550/arXiv.2509.23616

106.X. Huang, Y. Yang, Y. Wang, C. Wang, Z. Zhang, J. Xu, et al., Dgraph: A large-scale financial dataset for graph anomaly detection, in *Advances in Neural Information Processing Systems*, **35** (2022), 22765–22777. https://doi.org/10.1109/TITS.2022.3185013

107.X. Zhu, Z. Ghahramani, Learning from labeled and unlabeled data with label propagation, *IJCNN'02*, **3** (2002), 2803–2808. https://doi.org/10.1109/IJCNN.2002.1007592

108.C. Gong, D. Tao, W. Liu, L. Liu, J. Yang, Label propagation via teaching-to-learn and learning-to-teach, *IEEE Trans. Neural Networks Learn. Syst.*, **28** (2016), 1452–1465. https://doi.org/10.1109/TNNLS.2016.2514360

109.T. Xie, B. Wang, C. C. J. Kuo, Graphhop: An enhanced label propagation method for node classification, *IEEE Trans. Neural Networks Learn. Syst.*, **34** (2023), 9287–9301. https://doi.org/10.1109/TNNLS.2022.3157746

110.T. Xie, R. Kannan, C. C. J. Kuo, Label efficient regularization and propagation for graph node classification, *IEEE Trans. Pattern Anal. Mach. Intell.*, **45** (2023), 14856–14871. https://doi.org/10.1109/TPAMI.2023.3309970

111.L. Yu, L. Sun, B. Du, T. Zhu, W. Lv, Label-enhanced graph neural network for semi-supervised node classification, *IEEE Trans. Knowl. Data Eng.*, **35** (2023), 11529–11540. https://doi.org/10.48550/arXiv.2205.15653

112.D. C. G. Pedronette, L. J. Latecki, Rank-based self-training for graph convolutional networks, *Inf. Process. Manage.*, **58** (2021), 102443. https://doi.org/10.1016/j.ipm.2020.102443

113.Z. Zhou, J. Shi, S. Zhang, Z. Huang, Q. Li, Effective stabilized self-training on few-labeled graph data, *Inf. Sci.*, **631** (2023), 369–384. https://doi.org/10.1016/j.ins.2023.02.032

114.X. Juan, F. Zhou, W. Wang, W. Jin, J. Tang, X. Wang, INS-GNN: Improving graph imbalance learning with self-supervision, *Inf. Sci.*, **637** (2023), 118935. https://doi.org/10.1016/j.ins.2023.118935

115.X. Juan, M. Peng, X. Wang, Dynamic self-training with less uncertainty for graph imbalance learning, *Expert Syst. Appl.*, **271** (2025), 126643. https://doi.org/10.1016/j.eswa.2025.126643

116.M. Gao, S. Zhou, W. Gu, Z. Wu, H. Liu, A. Zhou, et al., MMGPT4LF: Leveraging an optimized pre-trained GPT-2 model with multi-modal cross-attention for load forecasting, *Appl. Energy*, **392** (2025), 125965. https://doi.org/10.1016/j.apenergy.2025.125965

117.J. Yu, Y. Ren, C. Gong, J. Tan, X. Li, X. Zhang, Leveraging large language models for node generation in few-shot learning on text-attributed graphs, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **39** (2025), 13087–13095. https://doi.org/10.1609/aaai.v39i12.33428

118.R. Xia, R. Wang, Y. Liu, A. Li, X. Liu, Y. Zhang, When noisy labels meet class imbalance on graphs: A graph augmentation method with LLM and pseudo label, preprint, arXiv:2507.18153. https://doi.org/10.48550/arXiv.2507.18153

119.T. Ma, Y. Qian, Z. Wang, Z. Zhang, C. Zhang, Y. Ye, LLM-empowered class imbalanced graph prompt learning for online drug trafficking detection, preprint, arXiv:2503.01900. https://doi.org/10.18653/v1/2025.findings-acl.725

120.L. Wang, Y. Wang, B. Ni, Y. Zhao, H. Wang, Y. Ma, et al., SaVe-TAG: Semantic-aware vicinal risk minimization for long-tailed text-attributed graphs, preprint, arXiv:2410.16882. https://doi.org/10.1145/3770854.3780311

121.X. Zhang, L. Zhang, D. Zhou, G. Xu, Fine-grainedly synthesize streaming data based on large language models with graph structure understanding for data sparsity, preprint, arXiv:2403.06139. https://doi.org/10.48550/arXiv.2403.06139

122.Z. Chen, H. Mao, H. Wen, H. Han, W. Jin, H. Zhang, et al., Label-free node classification on graphs with large language models (LLMS), preprint, arXiv:2310.04668. https://doi.org/10.48550/arXiv.2310.04668

123.Z. Sheng, W. Guo, Y. Shao, W. Zhang, B. Cui, LLMs are noisy oracles! LLM-based noise-aware graph active learning for node classification, in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, **2** (2025), 2526–2537. https://doi.org/10.1145/3711896.3737030

124.T. Zhang, R. Yang, Y. Lai, M. Yan, X. Ye, D. Fan, Leveraging large language models for effective label-free node classification in text-attributed graphs, in *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, (2025), 698–708. https://doi.org/10.1145/3726302.3730021

125.Z. Chen, J. Deng, J. Zhou, J. Wu, T. Qian, M. Huang, Depression detection in clinical interviews with LLM-empowered structural element graph, in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, (2024), 8181–8194. https://doi.org/10.18653/v1/2024.naacl-long.452

126.J. Wang, Q. Guan, L. Deng, M. Zhou, Z. Gong, GraphST: Class-imbalanced node classification with semantic relation transfer, *Pattern Recognit.*, **172** (2026), 112622. https://doi.org/10.1016/j.patcog.2025.112626

127.B. Pan, Z. Zhang, Y. Zhang, Y. Hu, L. Zhao, Distilling large language models for text-attributed graph learning, in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, (2024), 1836–1845. https://doi.org/10.1145/3627673.3679830

128.Y. Wei, Q. Huang, J. T. Kwok, Y. Zhang, KICGPT: Large language model with knowledge in context for knowledge graph completion, preprint, arXiv:2402.02389. https://doi.org/10.18653/v1/2023.findings-emnlp.580

129.W. Hou, W. Zhao, N. Jia, X. Liu, Low-resource knowledge graph completion based on knowledge distillation driven by large language models, *Appl. Soft Comput.*, **169** (2025), 112622. https://doi.org/10.1016/j.asoc.2024.112622

130.S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, R. Togneri, Cost-sensitive learning of deep feature representations from imbalanced data, *IEEE Trans. Neural Networks Learn. Syst.*, **29** (2017), 3573–3587. https://doi.org/10.1109/TNNLS.2017.2732482

131.Z. Jiang, T. Chen, T. Chen, Z. Wang, Improving contrastive learning on imbalanced data via open-world sampling, in *Advances in Neural Information Processing Systems*, **34** (2021), 5997–6009.

132.S. Li, L. Han, Y. Wang, Y. Pu, J. Zhu, J. Li, GCL: Contrastive learning instead of graph convolution for node classification, *Neurocomputing*, **551** (2023), 126491. https://doi.org/10.1016/j.neucom.2023.126491

133.P. Bao, J. Li, R. Yan, Z. Liu, Dynamic graph contrastive learning via maximize temporal consistency, *Pattern Recognit.*, **148** (2024), 110144. https://doi.org/10.1016/j.patcog.2023.110144

134.P. Ribeiro, P. Paredes, M. E. Silva, D. Aparicio, F. Silva, A survey on subgraph counting: Concepts, algorithms, and applications to network motifs and graphlets, *ACM Comput. Surv.*, **54** (2021), 1–36. https://doi.org/10.1145/3433652

135.X. Guo, K. Wu, X. Zhang, J. Liu, Automated loss function search for class-imbalanced node classification, in *International Conference on Machine Learning*, PMLR, (2024), 16958–16973. https://doi.org/10.48550/arXiv.2405.14133

136.S. Yang, J. Xia, J. Zhou, X. Yao, X. Zhang, NodeReg: Mitigating the imbalance and distribution shift effects in semi-supervised node classification via norm consistency, preprint, arXiv:2503.03211. https://doi.org/10.48550/arXiv.2503.03211

137.J. Qin, P. Huang, Q. Sun, C. Ji, X. Fu, J. Li, Graph size-imbalanced learning with energy-guided structural smoothing, in *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, (2025), 457–465. https://doi.org/10.1145/3701551.3703559

138.H. Guo, Y. Li, J. Shang, M. Gu, Y. Huang, B. Gong, Learning from class-imbalanced data: Review of methods and applications, *Expert Syst. Appl.*, **73** (2017), 220–239. https://doi.org/10.1016/j.eswa.2016.12.035

139.A. N. Tarekegn, M. Giacobini, K. Michalak, A review of methods for imbalanced multi-label classification, *Pattern Recognit.*, **118** (2021), 107965. https://doi.org/10.1016/j.patcog.2021.107965

140.Y. Yu, B. Li, Z. Ji, J. Han, Z. Zhang, Knowledge distillation classifier generation network for zero-shot learning, *IEEE Trans. Neural Networks Learn. Syst.*, **34** (2021), 3183–3194. https://doi.org/10.1109/TNNLS.2021.3112229

141.X. Lin, J. Kang, W. Cong, H. Tong, BeMap: Balanced message passing for fair graph neural network, in *Learning on Graphs Conference*, PMLR, (2024), 1–37. https://doi.org/10.48550/arXiv.2306.04107

142.K. J. Chen, W. Mu, Z. Liu, Z. Liu, SAug: Structural imbalance aware augmentation for graph neural networks, *ACM Trans. Intell. Syst. Technol.*, **16** (2025), 22. https://doi.org/10.1145/3712699

143.A. Alchihabi, H. Yan, Y. Guo, Overcoming class imbalance: Unified GNN learning with structural and semantic connectivity representations, preprint, arXiv:2412.20656. https://doi.org/10.48550/arXiv.2412.20656

144.W. Xu, P. Wang, Z. Zhao, B. Wang, X. Wang, Y. Wang, When imbalance meets imbalance: Structure-driven learning for imbalanced graph classification, in *Proceedings of the ACM Web Conference 2024*, (2024), 905–913. https://doi.org/10.1145/3589334.3645629

145.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), 701–710.

146.T. Jin, H. Dai, L. Cao, B. Zhang, F. Huang, Y. Gao, et al., Deepwalk-aware graph convolutional networks, *Sci. China Inf. Sci.*, **65** (2022), 152104. https://doi.org/10.1007/s11432-020-3318-5

147.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)*, (2018), 871–880. https://doi.org/10.1109/BigData.2018.8622603

148.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*, **32** (2018). https://doi.org/10.1609/aaai.v32i1.11242

149.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)*, (2022), 1–6. https://doi.org/10.1109/ICNSC55942.2022.10004144

150.D. Wang, Z. Zhang, Y. Ma, T. Zhao, T. Jiang, N. V. Chawla, et al., Learning attribute-structure co-evolutions in dynamic graphs, preprint, arXiv:2007.13004. https://doi.org/10.48550/arXiv.2007.13004

151.H. Wang, J. Wang, J. Wang, M. Zhao, W. Zhang, F. Zhang, et al., GraphGAN: Graph representation learning with generative adversarial nets, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **32** (2018). https://doi.org/10.1609/aaai.v32i1.11872

152.Q. Wang, Y. Ji, Y. Hao, J. Cao, GRL: Knowledge graph completion with GAN-based reinforcement learning, *Knowledge-Based Syst.*, **209** (2020), 106421. https://doi.org/10.1016/j.knosys.2020.106421

153.A. J. Fofanah, D. Chen, L. Wen, S. Zhang, Addressing imbalance in graph datasets: Introducing gate-GNN with graph ensemble weight attention and transfer learning for enhanced node classification, *Expert Syst. Appl.*, **255** (2024), 124602. https://doi.org/10.1016/j.eswa.2024.124602

154.Z. Li, D. Chen, T. Zhao, D. Wang, H. Liu, Z. Zhang, et al., IceBerg: Debiased self-training for class-imbalanced node classification, preprint, arXiv:2502.06280. https://doi.org/10.48550/arXiv.2502.06280

155.J. Liu, M. He, G. Wang, N. Q. V. Hung, X. Shang, H. Yin, Imbalanced node classification beyond homophilic assumption, in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, (2023), 7206–7214. https://doi.org/10.48550/arXiv.2304.14635

156.P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, T. Eliassi-Rad, Collective classification in network data, *AI Mag.*, **29** (2008), 93. https://doi.org/10.1609/aimag.v29i3.2157

157.M. Zitnik, J. Leskovec, Predicting multicellular function through multi-layer tissue networks, *Bioinformatics*, **33** (2017), i190–i198. https://doi.org/10.1093/bioinformatics/btx252

158.W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, et al., Open graph benchmark: Datasets for machine learning on graphs, preprint, arXiv:2005.00687. https://doi.org/10.48550/arXiv.2005.00687

159.P. Mernyei, C. Cangea, Wiki-CS: A wikipedia-based benchmark for graph neural networks, preprint, arXiv:2007.02901. https://doi.org/10.48550/arXiv.2007.02901

160. Y. Chen, C. Ding, J. Hu, R. Chen, P. Hui, X. Fu, Building and analyzing a global co-authorship network using google scholar data, in *Proceedings of the 26th International Conference on World Wide Web Companion*, (2017), 1219–1224. https://doi.org/10.1145/3041021.3053056

161. J. McAuley, C. Targett, Q. Shi, A. Van Den Hengel, Image-based recommendations on styles and substitutes, in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, (2015), 43–52. https://doi.org/10.48550/arXiv.1506.04757

162. B. Sigurbjörnsson, R. Van Zwol, Flickr tag recommendation based on collective knowledge, in *Proceedings of the 17th International Conference on World Wide Web*, (2008), 327–336. https://doi.org/10.1145/1367497.1367542

163. T. Zhao, C. Deng, K. Yu, T. Jiang, D. Wang, M. Jiang, Error-bounded graph anomaly loss for GNNs, in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, (2020), 1873–1882. https://doi.org/10.1145/3340531.3411979

164. M. Wattenhofer, R. Wattenhofer, Z. Zhu, The youtube social network, in *Proceedings of the International AAAI Conference on Web and Social Media*, **6** (2012), 354–361. https://doi.org/10.1609/icwsm.v6i1.14243

165. H. Bisgin, N. Agarwal, X. Xu, Investigating homophily in online social networks, in *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, **1** (2010), 533–536. https://doi.org/10.1109/WI-IAT.2010.61

166. B. Rozemberczki, C. Allen, R. Sarkar, Multi-scale attributed node embedding, *J. Complex Netw.*, **9** (2021), cnab014. https://doi.org/10.1093/comnet/cnab014

167. Y. Elmougy, L. Liu, Demystifying fraudulent transactions and illicit nodes in the bitcoin network for financial forensics, preprint, arXiv:2306.06108. https://doi.org/10.1145/3580305.3599803

168. R. Longadge, S. Dongre, Class imbalance problem in data mining review, preprint, arXiv:1305.1707. https://doi.org/10.48550/arXiv.1305.1707

169. J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (2014), 1532–1543. https://doi.org/10.3115/v1/D14-1162

170. J. Xiong, Z. Xiong, K. Chen, H. Jiang, M. Zheng, Graph neural networks for automated de novo drug design, *Drug Discov. Today*, **26** (2021), 1382–1393. https://doi.org/10.1016/j.drudis.2021.02.011

171. D. Szklarczyk, A. L. Gable, D. Lyon, A. Junge, S. Wyder, J. Huerta-Cepas, et al., STRING v11: Protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets, *Nucleic Acids Res.*, **47** (2019), D607–D613. https://doi.org/10.1093/nar/gky1131

172. D. Pretolesi, D. Garbarino, D. Giampaoli, A. Vian, A. Barla, Geometric deep learning strategies for the characterization of academic collaboration networks, *IEEE Trans. Emerging Top. Comput.*, **12** (2024), 840–851. https://doi.org/10.1109/TETC.2023.3315954

173. R. Zhao, Z. Shao, W. Zhang, J. Zhang, C. Wu, A multi-channel multi-tower GNN model for job transfer prediction based on academic social network, *Appl. Soft Comput.*, **142** (2023), 110300. https://doi.org/10.1016/j.asoc.2023.110300

174. J. Liu, Y. Chen, X. Huang, J. Li, G. Min, GNN-based long and short term preference modeling for next-location prediction, *Inf. Sci.*, **629** (2023), 1–14. https://doi.org/10.1016/j.ins.2023.01.131

175. D. Cheng, F. Yang, S. Xiang, J. Liu, Financial time series forecasting with multi-modality graph neural network, *Pattern Recognit.*, **121** (2022), 108218. https://doi.org/10.1016/j.patcog.2021.108218

176. Z. Guo, H. Wang, A deep graph neural network-based mechanism for social recommendations, *IEEE Trans. Ind. Inf.*, **17** (2020), 2776–2783. https://doi.org/10.1109/TII.2020.2986316

177. J. Opitz, A closer look at classification evaluation metrics and a critical reflection of common evaluation practice, *Trans. Assoc. Comput. Ling.*, **12** (2024), 820–836. https://doi.org/10.1162/tacl_a_00675

178. J. A. Swets, Measuring the accuracy of diagnostic systems, *Science*, **240** (1988), 1285–1293. https://doi.org/10.1126/science.3287615

179. J. Du, C. M. Vong, C. M. Pun, P. K. Wong, W. F. Ip, Post-boosting of classification boundary for imbalanced data using geometric mean, *Neural Networks*, **96** (2017), 101–114. https://doi.org/10.1016/j.neunet.2017.09.004

180. G. M. Foody, Challenges in the real world use of classification accuracy metrics: From recall and precision to the matthews correlation coefficient, *PLoS One*, **18** (2023), e0291908. https://doi.org/10.1371/journal.pone.0291908

181. T. T. Wong, Linear approximation of f-measure for the performance evaluation of classification algorithms on imbalanced data sets, *IEEE Trans. Knowl. Data Eng.*, **34** (2020), 753–763. https://doi.org/10.1109/TKDE.2020.2986749

182. P. A. Jaskowiak, I. G. Costa, R. J. Campello, The area under the roc curve as a measure of clustering quality, *Data Min. Knowl. Discovery*, **36** (2022), 1219–1245. https://doi.org/10.1007/s10618-022-00829-0

183. W. Dong, J. Wu, Z. Bai, Y. Hu, W. Li, W. Qiao, et al., Mobilegcn applied to low-dimensional node feature learning, *Pattern Recognit.*, **112** (2021), 107788. https://doi.org/10.1016/j.patcog.2020.107788

184. D. Wang, L. Hu, H. Du, Y. Liu, J. Huang, Y. Xu, et al., Classification, experimental assessment, modeling methods and evaluation metrics of trombe walls, *Renewable Sustainable Energy Rev.*, **124** (2020), 109772. https://doi.org/10.1016/j.rser.2020.109772

185. Ž. Vujović, Classification model evaluation metrics, *Int. J. Adv. Comput. Sci. Appl.*, **12** (2021), 599–606. https://doi.org/10.14569/IJACSA.2021.0120670

186. K. Dembczynski, W. Waegeman, W. Cheng, E. Hüllermeier, An exact algorithm for f-measure maximization, in *Advances in Neural Information Processing Systems*, **24** (2011).

187. M. Mironov, L. Prokhorenkova, Revisiting graph homophily measures, preprint, arXiv:2412.09663. https://doi.org/10.48550/arXiv.2412.09663

188. J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, D. Koutra, Beyond homophily in graph neural networks: Current limitations and effective designs, in *Advances in Neural Information Processing Systems*, **33** (2020), 7793–7804.

189. H. Pei, B. Wei, K. C. C. Chang, Y. Lei, B. Yang, Geom-GCN: Geometric graph convolutional networks, preprint, arXiv:2002.05287. https://doi.org/10.48550/arXiv.2002.05287

190. O. Platonov, D. Kuznedelev, A. Babenko, L. Prokhorenkova, Characterizing graph datasets for node classification: Homophily-heterophily dichotomy and beyond, in *Advances in Neural Information Processing Systems*, **36** (2023), 523–548. https://doi.org/10.48550/arXiv.2209.06177

191. M. Fey, J. Sunil, A. Nitta, R. Puri, M. Shah, B. Stojanovič, et al., PYG 2.0: Scalable learning on real world graphs, preprint, arXiv:2507.16991. https://doi.org/10.48550/arXiv.2507.16991

192. M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, et al., Deep graph library: A graph-centric, highly-performant package for graph neural networks, preprint, arXiv:1909.01315. https://doi.org/10.48550/arXiv.1909.01315

193. J. Qin, H. Yuan, Q. Sun, L. Xu, J. Yuan, P. Huang, et al., IGL-bench: Establishing the comprehensive benchmark for imbalanced graph learning, in *The Thirteenth International Conference on Learning Representations*, (2025), 1–58. https://doi.org/10.48550/arXiv.2406.09870

194. X. Zhou, W. Liang, W. Li, K. Yan, S. Shimizu, I. Kevin, et al., Hierarchical adversarial attacks against graph-neural-network-based iot network intrusion detection system, *IEEE Internet Things J.*, **9** (2021), 9310–9319. https://doi.org/10.1109/JIOT.2021.3130434

195. J. Yoon, K. Ahn, J. Park, H. Yeo, Transferable traffic signal control: Reinforcement learning with graph centric state representation, *Transp. Res. Part C Emerging Technol.*, **130** (2021), 103321. https://doi.org/10.1016/j.trc.2021.103321

196. B. Pandey, P. K. Bhanodia, A. Khamparia, D. K. Pandey, A comprehensive survey of edge prediction in social networks: Techniques, parameters and challenges, *Expert Syst. Appl.*, **124** (2019), 164–181. https://doi.org/10.1016/j.eswa.2019.01.040

197. Z. Kang, H. Pan, S. C. Hoi, Z. Xu, Robust graph learning from noisy data, *IEEE Trans. Cybern.*, **50** (2019), 1833–1843. https://doi.org/10.1109/TCYB.2018.2887094

198. J. Jo, J. Baek, S. Lee, D. Kim, M. Kang, S. J. Hwang, Edge representation learning with hypergraphs, in *Advances in Neural Information Processing Systems*, **34** (2021), 7534–7546. https://doi.org/10.48550/arXiv.2106.15845

199. X. Cheng, Y. Wang, Y. Liu, Y. Zhao, C. C. Aggarwal, T. Derr, Edge classification on graphs: New directions in topological imbalance, preprint, arXiv:2406.11685. https://doi.org/10.1145/3701551.3703518

200. Y. Wang, H. Wang, H. Jin, X. Huang, X. Wang, Exploring graph capsual network for graph classification, *Inf. Sci.*, **581** (2021), 932–950. https://doi.org/10.1016/j.ins.2021.10.001

201. Y. Wang, Y. Zhao, N. Shah, T. Derr, Imbalanced graph classification via graph-of-graph neural networks, in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, (2022), 2067–2076. https://doi.org/10.48550/arXiv.2112.00238

202. S. Miao, M. Liu, P. Li, Interpretable and generalizable graph learning via stochastic attention mechanism, in *International Conference on Machine Learning*, (2022), 15524–15543. https://doi.org/10.48550/arXiv.2201.12987

203. W. Ju, Z. Mao, S. Yi, Y. Qin, Y. Gu, Z. Xiao, et al., Hypergraph-enhanced dual semi-supervised graph classification, preprint, arXiv:2405.04773. https://doi.org/10.48550/arXiv.2405.04773

204.Z. Tan, K. Ding, R. Guo, H. Liu, Graph few-shot class-incremental learning, in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, (2022), 987–996. https://doi.org/10.1145/3488560.3498455

205.C. Wang, Y. Zhang, X. Zhang, Z. Wu, X. Zhu, S. Jin, et al., Offline-online learning of deformation model for cable manipulation with graph neural networks, *IEEE Rob. Autom. Lett.*, **7** (2022), 5544–5551. https://doi.org/10.1109/LRA.2022.3158376

206.Q. Yuan, S. U. Guan, T. Luo, K. L. Man, E. G. Lim, IMCGNN: Information maximization based continual graph neural networks for inductive node classification, *Neurocomputing*, **624** (2025), 129362. https://doi.org/10.1016/j.neucom.2025.129362

207.J. Vatter, M. L. Rochau, R. Mayer, H. A. Jacobsen, Size does (not) matter? sparsification and graph neural network sampling for large-scale graphs, *Proc. VLDB Endow.*, **2150** (2024), 8097.

208.X. Li, J. Ma, Z. Wu, D. Su, W. Zhang, R. H. Li, et al., Rethinking node-wise propagation for large-scale graph learning, in *Proceedings of the ACM Web Conference 2024*, (2024), 560–569. https://doi.org/10.1145/3589334.3645450

209.N. Chairatanakul, X. Liu, N. T. Hoang, T. Murata, Heterogeneous graph embedding with single-level aggregation and infomax encoding, *Mach. Learn.*, **112** (2023), 4227–4256. https://doi.org/10.1007/s10994-022-06160-5

210.X. Wang, N. Liu, H. Han, C. Shi, Self-supervised heterogeneous graph neural network with co-contrastive learning, in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, (2021), 1726–1736. https://doi.org/10.1145/3447548.3467415

211.L. Chen, W. Liang, Disentangling multi-factor effects via graph contrastive learning for travel recommendation, *Appl. Soft Comput.*, **186** (2026), 114095. https://doi.org/10.1016/j.asoc.2025.114095

212.A. Noori, M. A. Balafar, A. Bouyer, K. Salmani, Review of heterogeneous graph embedding methods based on deep learning techniques and comparing their efficiency in node classification, *Social Network Anal. Min.*, **14** (2024), 17. https://doi.org/10.1007/s13278-023-01178-6

213.L. Chen, X. Zhu, G. Zhu, Exploiting attributes and keywords for session-based recommendation with multi-view graph neural network, *Expert Syst. Appl.*, **296** (2026), 128990. https://doi.org/10.1016/j.eswa.2025.128990

214.Y. Chen, F. Chen, Z. Wu, Z. Chen, Z. Cai, Y. Tan, et al., Heterogeneous graph embedding with dual edge differentiation, *Neural Networks*, **183** (2025), 106965. https://doi.org/10.1016/j.neunet.2024.106965

215.X. Fu, J. Zhang, Z. Meng, I. King, MAGNN: Metapath aggregated graph neural network for heterogeneous graph embedding, in *Proceedings of the Web Conference 2020*, (2020), 2331–2341. https://doi.org/10.1145/3366423.3380297

216.Y. Zhao, S. Wang, H. Duan, LSPI: Heterogeneous graph neural network classification aggregation algorithm based on size neighbor path identification, *Appl. Soft Comput.*, **171** (2025), 112656. https://doi.org/10.1016/j.asoc.2024.112656

217.Y. Li, C. Jian, G. Zang, C. Song, X. Yuan, Node classification oriented adaptive multichannel heterogeneous graph neural network, *Knowledge-Based Syst.*, **292** (2024), 111618. https://doi.org/10.1016/j.knosys.2024.111618

218. K. Madhawa, T. Murata, Active learning for node classification: An evaluation, *Entropy*, **22** (2020), 1164. https://doi.org/10.3390/e22101164

219. R. Li, X. Yuan, M. Radfar, P. Marendy, W. Ni, T. J. O'Brien, et al., Graph signal processing, graph neural network and graph learning on biological data: A systematic review, *IEEE Rev. Biomed. Eng.*, **16** (2021), 109–135. https://doi.org/10.1109/RBME.2021.3122522