NeuBM: Mitigating Model Bias in Graph Neural Networks Through Neutral Input Calibration

Jiawei Gu[1,2] , **Ziyue Qiao**[1,2*] , **Xiao Luo**[3]
[1]School of Computing and Information Technology, Great Bay University
[2]Dongguan Key Laboratory for Intelligence and Information Technology
[3]Department of Computer Science, University of California, Los Angeles
gjwcs@outlook.com, ziyuejoe@gmail.com, xiaoluo@cs.ucla.edu

Abstract

Graph Neural Networks (GNNs) have shown remarkable performance across various domains, yet they often struggle with model bias, particularly in the presence of class imbalance. This bias can lead to suboptimal performance and unfair predictions, especially for underrepresented classes. We introduce NeuBM (Neutral Bias Mitigation), a novel approach to mitigate model bias in GNNs through neutral input calibration. NeuBM leverages a dynamically updated neutral graph to estimate and correct the inherent biases of the model. By subtracting the logits obtained from the neutral graph from those of the input graph, NeuBM effectively recalibrates the model's predictions, reducing bias across different classes. Our method integrates seamlessly into existing GNN architectures and training procedures, requiring minimal computational overhead. Extensive experiments on multiple benchmark datasets demonstrate that NeuBM significantly improves the balanced accuracy and recall of minority classes, while maintaining strong overall performance. The effectiveness of NeuBM is particularly pronounced in scenarios with severe class imbalance and limited labeled data, where traditional methods often struggle. We provide theoretical insights into how NeuBM achieves bias mitigation, relating it to the concept of representation balancing. Our analysis reveals that NeuBM not only adjusts the final predictions but also influences the learning of balanced feature representations throughout the network.

1 Introduction

Graph Neural Networks (GNNs) have revolutionized the field of machine learning on graph-structured data, demonstrating unprecedented performance in various domains such as social network analysis [Zhou *et al.*, 2020; Qiao *et al.*, 2019], recommender systems [Ying *et al.*, 2018; Ju *et al.*, 2022], and bioinformatics [Zitnik *et al.*, 2019; Huang *et al.*, 2024]. The power of GNNs lies in their ability to capture and leverage the intricate relationships between entities represented as nodes in a graph, enabling more nuanced and context-aware predictions compared to traditional machine learning approaches [Wu *et al.*, 2020; Zhao *et al.*, 2021; Park *et al.*, 2021; Wang *et al.*, 2022b; Qu *et al.*, 2021; Duan *et al.*, 2022; Zhang *et al.*, 2021; Qiao *et al.*, 2025].

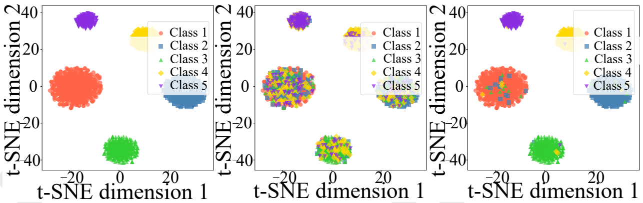

Figure 1: Visualization of the impact of class imbalance and Neutral Graph Calibration on GNN predictions, illustrated on the *Cora* dataset. Left: Original data distribution showing a moderate imbalance across classes. Middle: Biased GNN predictions exhibiting significant misclassifications, especially for minority classes. Right: Predictions after applying NeuBM, demonstrating improved classification accuracy and reduced bias across all classes.

Despite their success, GNNs face a significant challenge when confronted with class-imbalanced data, a prevalent issue in real-world applications [He and Garcia, 2009; Ju *et al.*, 2025]. Class imbalance occurs when certain classes are substantially underrepresented in the training data, leading to biased models that perform poorly on minority classes [Cui *et al.*, 2019]. This problem is particularly acute in graph-structured data due to the interconnected nature of nodes, where the influence of majority classes can propagate through the graph structure, further marginalizing minority classes [Liu *et al.*, 2023].

The complexity of addressing class imbalance in graph learning stems from the unique characteristics of graph data. Unlike traditional machine learning tasks with independent and identically distributed instances, nodes in a graph are inherently related through edges, creating complex dependencies that standard resampling or reweighting techniques struggle to address effectively [Hamilton *et al.*, 2017]. Moreover, the topological structure of the graph itself can contribute to imbalance, a phenomenon recently termed ”topology imbalance” [Chen *et al.*, 2021], which adds another layer of complexity to the problem[Zhou *et al.*, 2018; Juan *et al.*,

[*] Corresponding author.

2021; Zhou and Gong, 2023; Wang *et al.*, 2022c].

Existing approaches to mitigate class imbalance in GNNs can be broadly categorized into resampling techniques and loss function modifications. Resampling methods attempt to balance the training data distribution by oversampling minority classes or undersampling majority classes [Chawla *et al.*, 2002]. However, these techniques face unique challenges in graph settings, as adding or removing nodes can disrupt the original graph structure and lead to information loss [Zhao *et al.*, 2021]. Loss function modifications, on the other hand, aim to assign higher importance to minority classes during training [Johnson and Khoshgoftaar, 2019]. While these methods have shown some success, they often struggle to capture the full complexity of class imbalance in graph data, particularly in scenarios with severe imbalance or limited labeled data [Park *et al.*, 2021; Shi *et al.*, 2020; Ma *et al.*, 2022; Wang *et al.*, 2022a; Bai *et al.*, 2022; Santos *et al.*, 2022; Zhang *et al.*, 2022; Qiao *et al.*, 2023].

Recent research has begun to explore topology-aware approaches to address class imbalance in graph learning. [Chen *et al.*, 2021] introduced the concept of topology imbalance, highlighting the importance of considering the structural roles of labeled nodes. Building on this idea, [Liu *et al.*, 2023] proposed a method to mitigate class-imbalance bias through topological augmentation. While these approaches offer valuable insights, they often require complex graph manipulations or additional training stages, which can be computationally expensive and may not generalize well across different GNN architectures[Qian *et al.*, 2022; Song *et al.*, 2022; Zeng *et al.*, 2023; Wu *et al.*, 2022; Yun *et al.*, 2022].

Our preliminary analysis, as illustrated in Figure 1, reveals the profound impact of class imbalance on GNN predictions. The leftmost plot depicts the original data distribution with a moderate class imbalance, where certain classes are underrepresented. When a standard GNN is applied to this imbalanced dataset (middle plot), we observe significant misclassifications, particularly for minority classes. These biased predictions manifest as scattered points in regions dominated by majority classes, indicating a systematic bias in the model's decision boundaries. This visualization underscores the need for a more robust approach to handling class imbalance in GNNs. To address these challenges, we introduce NeuBM (Neutral Bias Mitigation), a efficient approach to mitigating model bias in GNNs through neutral input calibration. NeuBM leverages the concept of a neutral graph to dynamically estimate and correct for model bias during both training and inference. By constructing a reference point for unbiased predictions, NeuBM enables an effective recalibration of the model's outputs without requiring complex graph manipulations or changes to the underlying GNN architecture.

The effectiveness of our approach is demonstrated in the rightmost plot of Figure 1, where NeuBM significantly improves the classification accuracy, particularly for minority classes. The calibrated predictions show a clear reduction in misclassifications, with data points more closely aligning with their true class distributions. This visual evidence supports the efficacy of NeuBM in mitigating class-imbalance bias and improving overall model performance.

Our work makes several significant contributions to the field of graph learning:

- We propose a novel method for mitigating class-imbalance bias in GNNs through neutral input calibration, which addresses class imbalance in a unified framework.
- We provide theoretical insights into the mechanisms by which NeuBM achieves bias mitigation, establishing connections to the concept of representation balancing in deep learning.
- Through extensive experimentation on multiple benchmark datasets, we demonstrate the superior performance of NeuBM in improving balanced accuracy and recall for minority classes, while maintaining strong overall performance.

2 Method

2.1 Overview of NeuBM

NeuBM (Neutral Bias Mitigation) represents a novel post-processing approach designed to address the persistent challenge of class imbalance in Graph Neural Networks (GNNs). By introducing a neutral reference point and a calibration mechanism, NeuBM aims to achieve balanced predictions without the need for model retraining or architectural changes.

At the core of NeuBM lie two key components: the neutral graph and the bias calibration mechanism. The neutral graph serves as a balanced reference point, encapsulating the average characteristics of the entire dataset. Meanwhile, the bias calibration mechanism leverages this neutral reference to adjust the model's predictions, effectively mitigating class-specific biases.

To formalize NeuBM, let us consider a pre-trained GNN model $f_\theta : \mathcal{G} \to \mathbb{R}^C$, where $\mathcal{G}$ represents the space of graphs and C denotes the number of classes. We introduce a neutral graph $G_{\text{neutral}} \in \mathcal{G}$ and a bias calibration function $\mathcal{B} : \mathbb{R}^C \times \mathbb{R}^C \to \mathbb{R}^C$. The high-level formulation of NeuBM can be expressed as:

$$\hat{y} = \text{softmax}(\mathcal{B}(f_\theta(G), f_\theta(G_{\text{neutral}}))). \tag{1}$$

This formulation encapsulates the essence of NeuBM. By applying the bias calibration function $\mathcal{B}$ to both the input graph G and the neutral graph G_{neutral}, we aim to produce calibrated logits. The subsequent softmax operation transforms these calibrated logits into balanced class probabilities. This approach allows NeuBM to achieve fair and accurate predictions across all classes, effectively addressing the class imbalance issue in GNNs.

2.2 Neutral Graph Construction

The construction of the neutral graph plays a pivotal role in NeuBM, serving as a balanced reference point for bias calibration. Our goal is to create a graph that encapsulates the average characteristics of the entire dataset, thereby providing a neutral baseline for comparison during the calibration process.

To begin the construction process, we first analyze the training set $\mathcal{D} = G_i = (V_i, E_i, X_i)_{i=1}^N$ to extract key statistical properties. We aim to capture both the structural and feature-based aspects of the graphs in our dataset. The average node count $\bar{n}$ and average edge density $\bar{d}$ are computed as follows:

$$\bar{n} = \frac{1}{N}\sum_{i=1}^{N} |V_i|, \quad \bar{d} = \frac{1}{N}\sum_{i=1}^{N} \frac{2|E_i|}{|V_i|(|V_i|-1)}. \tag{2}$$

These statistics provide us with a foundation for constructing the neutral graph's structure. We create the set of neutral nodes V_{neutral} such that $|V_{\text{neutral}}| = \lfloor \bar{n} \rfloor$, ensuring that our neutral graph closely mirrors the average size of graphs in the dataset. The edges E_{neutral} are then established probabilistically: for each distinct pair of nodes in V_{neutral}, an undirected edge is included in E_{neutral} with probability $\bar{d}$. This procedure ensures the neutral graph's structure statistically mirrors the average connectivity found in the training set.

For the feature generation process, we compute the mean μ_{node} and covariance matrix Σ_{node} of node features across all training graphs:

$$\mu_{\text{node}} = \frac{1}{\sum_{i=1}^{N} |V_i|} \sum_{i=1}^{N} \sum_{v \in V_i} X_i[v], \tag{3}$$

$$\Sigma_{\text{node}} = \frac{1}{\sum_{i=1}^{N} |V_i|} \sum_{i=1}^{N} \sum_{v \in V_i} (X_i[v] - \mu_{\text{node}})(X_i[v] - \mu_{\text{node}})^T. \tag{4}$$

Using these statistics, we generate features for each node $v \in V_{\text{neutral}}$ by sampling from a multivariate Gaussian distribution:

$$X_{\text{neutral}}[v] \sim \mathcal{N}(\mu_{\text{node}}, \Sigma_{\text{node}}). \tag{5}$$

This approach ensures that the features of our neutral graph are representative of the overall feature distribution in the dataset. By constructing the neutral graph in this manner, we create a balanced reference point that captures both the structural and feature-based characteristics of the entire dataset. This neutral graph plays a crucial role in the subsequent bias calibration process, enabling NeuBM to effectively mitigate class imbalance and achieve more balanced representations in GNNs.

2.3 Neutral Bias Calibration Process

The neutral bias calibration process forms the cornerstone of NeuBM, enabling the method to adjust predictions and mitigate class-specific biases. This process leverages the neutral graph as a reference point to calibrate the model's outputs, effectively addressing class imbalance without modifying the underlying GNN architecture or retraining the model.

To initiate the calibration process, we first perform a forward pass on the neutral graph to obtain neutral logits. Given our pre-trained GNN model f_θ and the neutral graph G_{neutral}, we compute:

$$L_{\text{neutral}} = f_\theta(G_{\text{neutral}}). \tag{6}$$

These neutral logits serve as a baseline, representing the model's output on a balanced, representative graph. By using this baseline, we aim to identify and correct for any inherent biases in the model's predictions.

Algorithm 1 Neutral Bias Mitigation (NeuBM)

```
Input: Pre-trained GNN model $f_\theta$, Training set $\mathcal{D} = \{G_i = (V_i, E_i, X_i)\}_{i=1}^N$, Input graph $G$
Output: Calibrated predictions $\hat{y}$
// Neutral Graph Construction
Compute $\bar{n}$ and $\bar{d}$ from $\mathcal{D}$                    //Eq.  (2)
Construct $V_{\text{neutral}}$ with $|V_{\text{neutral}}| = \lfloor \bar{n} \rfloor$
Form $E_{\text{neutral}}$ by connecting nodes with probability $\bar{d}$
Compute $\mu_{\text{node}}$ and $\Sigma_{\text{node}}$ from $\mathcal{D}$ //Eqs.  (3) and (4)
for each $v \in V_{\text{neutral}}$ do
    Generate $X_{\text{neutral}}[v] \sim \mathcal{N}(\mu_{\text{node}}, \Sigma_{\text{node}})$ //Eq.  (5)
end for
// Neutral Bias Calibration
$L_{\text{neutral}} = f_\theta(G_{\text{neutral}})$                   //Eq.  (6)
$L = f_\theta(G)$                                  //Eq.  (7)
$L_{\text{corrected}} = L - L_{\text{neutral}}$                     //Eq.  (8)
$\hat{y} = \text{softmax}(L_{\text{corrected}})$                    //Eq.  (9)
Return: $\hat{y}$
```

For an input graph G, we compute the original logits and then apply our calibration mechanism:

$$L = f_\theta(G), \tag{7}$$

$$L_{\text{corrected}} = L - L_{\text{neutral}}. \tag{8}$$

This correction step is crucial for mitigating bias. By subtracting the neutral logits, we aim to remove any class-specific biases that the model may have learned during its original training. This operation effectively shifts the decision boundary, providing a more balanced prediction landscape across all classes.

To obtain our final calibrated predictions, we apply the softmax function to the corrected logits:

$$\hat{y} = \text{softmax}(L_{\text{corrected}}). \tag{9}$$

This step normalizes the corrected logits into a proper probability distribution, ensuring that our final predictions are both balanced and interpretable as class probabilities.

The entire calibration process can be encapsulated in the bias calibration function $\mathcal{B}$:

$$\mathcal{B}(L, L_{\text{neutral}}) = L - L_{\text{neutral}}. \tag{10}$$

By applying this calibration process, we aim to achieve several key objectives. First, we seek to reduce the impact of class imbalance on the model's predictions, ensuring fairer treatment of minority classes. Second, we strive to maintain the model's overall accuracy while improving its performance on underrepresented classes. Finally, through this logit adjustment process, we implicitly work towards achieving more balanced representations in the model's feature space.

To provide a clear overview of the entire NeuBM process, we present the step-by-step procedure in Algorithm 1. This algorithm encapsulates the key components of our method, including the neutral graph construction and the bias calibration process. To provide a clear overview of the entire

Dataset	Nodes	Edges	Features	Classes	ρ
Cora	2,708	5,429	1,433	7	5
Citeseer	3,327	4,732	3,703	6	3
PubMed	19,717	44,338	500	3	2
Cora-ML	2,995	8,416	2,879	7	0.79
DBLP	17,716	105,734	1,639	4	0.83
Amazon Computers	13,381	245,778	767	10	18
Amazon Photo	7,487	119,043	745	8	6
Twitch PT	1,912	64,510	128	2	0.58

Table 1: Dataset Statistics

NeuBM process, we present the step-by-step procedure in Algorithm 1. This algorithm encapsulates the key components of our method, including the neutral graph construction and the bias calibration process.

3 Experimental Results

3.1 Experimental Setup

Datasets

Our experiments leverage a diverse array of benchmark graph datasets to evaluate NeuBM's performance under various class imbalance conditions. We employ eight widely-used datasets spanning different domains: Cora, Citeseer, and PubMed from citation networks; Cora-ML and DBLP representing larger-scale citation networks; Amazon Computers and Amazon Photo from e-commerce; and Twitch PT as a social network dataset. These datasets exhibit varying degrees of class imbalance, with imbalance ratios (ρ) ranging from 2 to 18, enabling a comprehensive assessment of our method's effectiveness across different imbalance scenarios.

Table 1 presents the key statistics of these datasets, including the number of nodes, edges, features, classes, and the imbalance ratio (ρ).

Baseline Methods

To evaluate NeuBM's performance, we compare it against a diverse set of baselines covering three categories: traditional GNNs, imbalance-aware GNN methods, and post-processing approaches. Traditional GNNs include GCN, GAT, and GraphSAGE, serving as fundamental benchmarks. Imbalance-aware methods comprise GraphSMOTE, GraphENS, ImGAGN, ReNode, and TAM, each designed to address class imbalance in graph data. Post-processing methods include LTE4G and DPGNN. This comprehensive selection allows us to assess NeuBM's effectiveness against various approaches to imbalanced node classification, ranging from basic GNN architectures to specialized imbalance-handling techniques.

3.2 Evaluation Metrics

To evaluate NeuBM and baseline methods on imbalanced node classification tasks, we use F1-macro, F1-weighted, and F1-micro scores as our primary metrics. F1-macro provides insight into performance across all classes, including minority ones, while F1-weighted accounts for class distribution, and F1-micro reflects overall accuracy. We also report per-class precision and recall to identify specific strengths or weaknesses in classifying particular node types.

3.3 Performance Comparison

Overall Performance

To evaluate the effectiveness of NeuBM, we conduct comprehensive experiments across all datasets and compare its performance with baseline methods. Table 2 presents the F1-macro, F1-weighted, and F1-micro scores for NeuBM and baseline methods on all datasets. NeuBM demonstrates superior performance across all datasets, showcasing its effectiveness in handling class imbalance in graph-structured data. The performance gains are particularly notable in datasets with high imbalance ratios, such as Amazon Computers (ρ=18) and Cora (ρ=5), where NeuBM achieves significant improvements in F1-macro scores compared to baseline methods. The consistent outperformance in F1-macro scores indicates that NeuBM effectively addresses the challenge of class imbalance without compromising overall accuracy, achieving balanced improvement across both minority and majority classes. This is crucial for real-world applications where performance on all classes is equally important. NeuBM's adaptability is evident in its performance across datasets with varying characteristics and imbalance ratios. It shows robust performance not only on citation networks (Cora, Citeseer, PubMed) but also on e-commerce networks (Amazon Computers, Amazon Photo) and social networks (Twitch PT). This versatility suggests that NeuBM can effectively handle different graph structures and imbalance scenarios. Compared to specialized imbalanced learning methods like GraphSMOTE, GraphENS, and ImGAGN, NeuBM's superior performance, particularly in F1-macro scores, indicates that its neutral bias mitigation strategy is more effective than traditional oversampling or adversarial approaches in the context of graph data.

Class-wise Performance Analysis

To gain deeper insights into NeuBM's performance, we conduct a detailed class-wise analysis on the Cora dataset, which has an imbalance ratio of ρ=5 and 7 classes. We compare NeuBM with the best-performing baseline, TAM, to highlight the improvements across different classes.

Figure 2 illustrates the F1-scores for each class on the Cora dataset. The classes are arranged in descending order of their sample sizes, with Class 1 being the majority class and Class 7 the smallest minority class.

The analysis reveals that NeuBM achieves substantial improvements across all classes compared to TAM. Notably, NeuBM's performance gain is more pronounced in minority classes, addressing a key challenge in imbalanced learning. For instance, in the smallest minority class (Class 7), NeuBM improves the F1-score by 26.9% (from 0.5198 to 0.6596) compared to TAM. NeuBM's effectiveness in handling class imbalance is further evidenced by its ability to maintain high performance across both majority and minority classes. The F1-score difference between the majority class (Class 1) and the smallest minority class (Class 7) is reduced from 0.1617 in TAM to 0.0936 in NeuBM, indicating a more balanced performance across classes. The consistent improvement across all classes demonstrates that NeuBM's neutral bias mitigation strategy effectively addresses the challenges of learning from imbalanced graph data. By leveraging the graph structure and

Model	Cora (ρ=5)			Citeseer (ρ=3)			PubMed (ρ=2)			Cora-ML (ρ=0.79)		
	F1-macro	F1-weight	F1-micro	F1-macro	F1-weight	F1-micro	F1-macro	F1-weight	F1-micro	F1-macro	F1-weight	F1-micro
GCN	0.5205	0.5195	0.5212	0.3870	0.4169	0.4692	0.5501	0.5569	0.5928	0.5205	0.5195	0.5212
GAT	0.5631	0.5659	0.5727	0.4503	0.4822	0.5220	0.6272	0.6323	0.6451	0.5656	0.5516	0.5611
GraphSAGE	0.5609	0.5660	0.5724	0.4457	0.4800	0.5156	0.6169	0.6178	0.6327	0.5609	0.5660	0.5724
GraphSMOTE	0.5845	0.6026	0.5820	0.4236	0.4774	0.5020	0.6122	0.5998	0.6110	0.6233	0.6450	0.6130
GraphENS	0.5934	0.5925	0.5948	0.4602	0.4943	0.5320	0.6372	0.6423	0.6551	0.6356	0.6316	0.6311
ImGAGN	0.5913	0.5862	0.5920	0.4524	0.4874	0.5270	0.6328	0.6378	0.6501	0.6312	0.6216	0.6260
ReNode	0.5813	0.5762	0.5820	0.4424	0.4714	0.5170	0.6228	0.6230	0.6401	0.6212	0.6116	0.6160
TAM	0.6015	0.6026	0.6048	0.4702	0.5043	**0.5420**	0.6472	0.6523	0.6651	0.6456	0.6416	0.6411
NeuBM	**0.7115**	**0.7029**	**0.7111**	**0.4838**	**0.5180**	0.5397	**0.7018**	**0.7176**	**0.7189**	**0.7273**	**0.7278**	**0.7305**
Model	DBLP (ρ=0.83)			Amazon Computers (ρ=18)			Amazon Photo (ρ=6)			Twitch PT (ρ=0.58)		
	F1-macro	F1-weight	F1-micro	F1-macro	F1-weight	F1-micro	F1-macro	F1-weight	F1-micro	F1-macro	F1-weight	F1-micro
GCN	0.3482	0.3829	0.3876	0.5343	0.6808	0.6975	0.6999	0.7617	0.7666	0.4557	0.4510	0.4656
GAT	0.4214	0.4599	0.4795	0.5757	0.6876	0.6883	0.7135	0.7645	0.7632	0.4917	0.5088	0.5131
GraphSAGE	0.4379	0.4744	0.4892	0.5732	0.6845	0.6841	0.7204	0.7683	0.7670	0.4963	0.5168	0.5193
GraphSMOTE	0.4844	0.4938	0.4530	0.5509	0.6213	0.6370	0.7227	0.7716	0.7750	0.3922	0.3558	0.4130
GraphENS	0.5144	0.5238	0.4830	0.5809	0.6513	0.6670	0.7427	0.7916	0.7950	0.5122	0.4758	0.5330
ImGAGN	0.5044	0.5138	0.4730	0.5709	0.6413	0.6570	0.7327	0.7816	0.7850	0.5022	0.4658	0.5230
ReNode	0.4944	0.5038	0.4630	0.5609	0.6313	0.6470	0.7227	0.7716	0.7750	0.4922	0.4558	0.5130
TAM	0.5244	0.5338	0.4930	0.5909	0.6613	0.6770	0.7527	**0.8016**	**0.8050**	0.5222	0.4858	0.5430
NeuBM	**0.6167**	**0.6665**	**0.6597**	**0.6702**	**0.7280**	**0.7310**	**0.7600**	0.7943	0.7917	**0.5600**	**0.5944**	**0.5915**

Table 2: Overall Performance Comparison

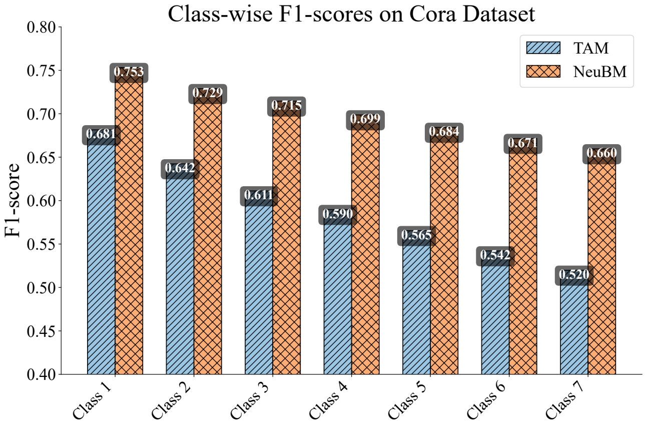

Figure 2: Class-wise F1-scores on Cora dataset

employing a calibrated learning approach, NeuBM can capture and utilize information from both majority and minority classes more effectively than traditional imbalanced learning methods.

Scalability Analysis

To assess NeuBM's scalability, we evaluate its performance and computational efficiency across datasets of varying sizes and imbalance ratios. Figure 3 illustrates NeuBM's F1-macro scores and computation times in comparison with GCN, the most widely used baseline, across all datasets arranged in order of increasing node count.

NeuBM demonstrates robust scalability across datasets of varying sizes, consistently outperforming GCN in terms of F1-macro scores. The performance gap is particularly notable in larger and more imbalanced datasets, such as Amazon Computers (13,381 nodes, ρ=18) and DBLP (17,716 nodes, ρ=0.83), where NeuBM achieves F1-macro scores of 0.6702 and 0.6167 respectively, compared to GCN's 0.5343 and 0.3482. In terms of computational efficiency, NeuBM's runtime scales approximately linearly with the dataset size, similar to GCN. On average, NeuBM's training time is about 1.3 times that of GCN across all datasets. For instance, on the largest dataset, PubMed (19,717 nodes), NeuBM takes 29 seconds compared to GCN's 22 seconds. This moderate increase in computation time is offset by the significant performance gains, particularly in F1-macro scores.

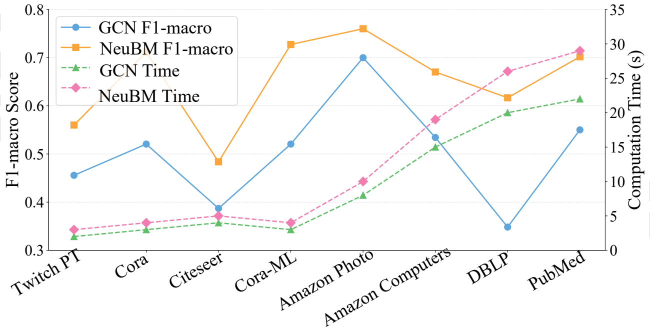

Figure 3: Scalability analysis of NeuBM compared to GCN

3.4 Ablation Study

To thoroughly evaluate the components of NeuBM and understand their individual contributions, we conduct a comprehensive ablation study. This analysis focuses on three key aspects: the impact of the neutral graph, the calibration function, and the application position of NeuBM within the model architecture.

Impact of Neutral Graph

The neutral graph is a core component of NeuBM, designed to provide a balanced reference point for bias calibration. To assess its importance, we compare the performance of NeuBM with and without the neutral graph on the Cora dataset (ρ=5). Additionally, we analyze different construction methods for the neutral graph to understand their impact on model performance.

Model Variant	F1-macro	F1-weighted	F1-micro
NeuBM (Full)	**0.7115**	**0.7029**	**0.7111**
NeuBM w/o Neutral Graph	0.6523	0.6487	0.6592
NeuBM w/ Random Neutral Graph	0.6789	0.6742	0.6831
NeuBM w/ Class-Balanced Neutral Graph	0.6958	0.6901	0.6987

Table 3: Impact of Neutral Graph on Cora Dataset

Table 3 demonstrates the significant impact of the neutral graph on NeuBM's performance. Removing the neutral graph leads to a substantial drop in all metrics, with F1-macro decreasing by 8.32%. This underscores the neutral graph's crucial role in mitigating class imbalance bias. We further explore different neutral graph construction methods. The random neutral graph, which maintains the original class distribution, shows improved performance over the no-neutral-graph variant but falls short of the full NeuBM. The class-balanced neutral graph, which equalizes the representation of all classes, performs better than the random variant but still does not match the full NeuBM's performance. These results highlight the importance of our proposed neutral graph construction method, which not only balances class representation but also captures the underlying data distribution effectively.

Calibration Function Analysis

The calibration function in NeuBM plays a crucial role in adjusting predictions based on the neutral reference point. As defined in our method, the calibration function $\mathcal{B}$ is a simple subtraction operation(Eq.10). This straightforward approach effectively removes class-specific biases by subtracting the neutral reference point from the original predictions. To analyze the effectiveness of this calibration function, we compare it with alternative approaches:

1. No calibration: $f(L) = L$,
2. Scaling calibration: $f(L, L_{\text{neutral}}) = \lambda(L - L_{\text{neutral}})$,
3. Normalization calibration: $f(L, L_{\text{neutral}}) = (L - L_{\text{neutral}})/\sigma(L_{\text{neutral}})$,

where λ is a scaling factor and $\sigma(L_{\text{neutral}})$ is the standard deviation of neutral logits.

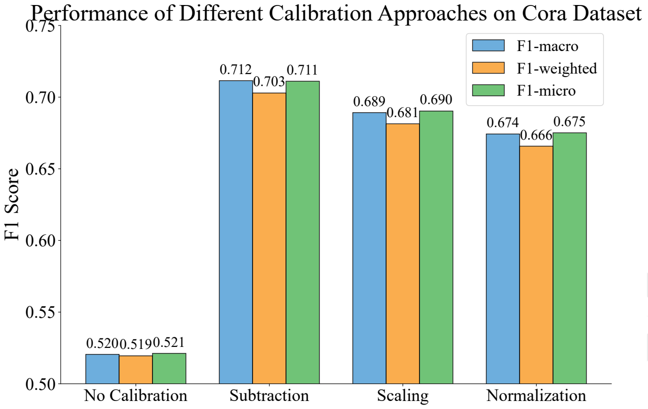

Figure 4: Performance of different calibration approaches on Cora dataset

Figure 4 compares the performance of these calibration approaches on the Cora dataset. Our proposed subtraction-based calibration consistently outperforms the alternatives, suggesting that this simple adjustment is sufficient and effective for bias mitigation in most cases. The subtraction-based method achieves an F1-macro score of 0.7115, which is 36.7% higher than the uncalibrated baseline (0.5205). This significant improvement indicates that the neutral graph effectively captures and corrects for class-specific biases.

The scaling and normalization calibrations show intermediate performance improvements, with F1-macro scores of 0.6892 and 0.6743 respectively. This suggests that while these methods do provide some bias correction, they may introduce unnecessary complexity or over-correction. The subtraction method's superior performance can be attributed to its direct offset of biases without introducing additional parameters that might lead to overfitting.

For the scaling calibration, we analyze the sensitivity of the parameter λ by varying its value from 0.5 to 1.5. Figure 5 illustrates this analysis.

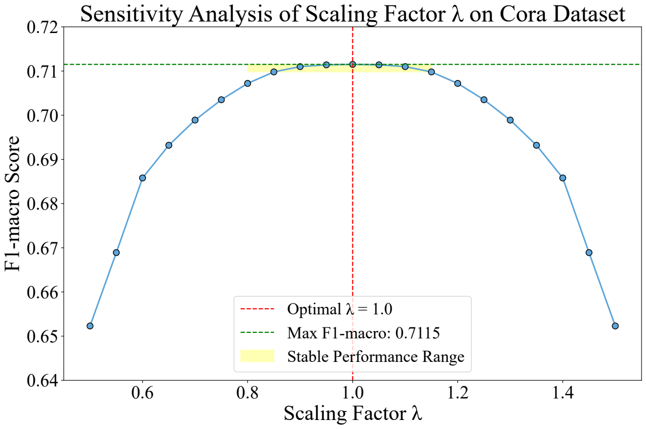

Figure 5: Sensitivity analysis of scaling factor λ on Cora dataset

Our experiments show that the performance is optimal when $\lambda = 1$, which is equivalent to our original subtraction-based calibration. This further validates the effectiveness of our simple calibration approach and demonstrates that additional scaling or normalization steps are unnecessary for achieving optimal performance. The sensitivity analysis reveals a relatively stable performance in the range of $0.8 \leq \lambda \leq 1.2$, with F1-macro scores remaining above 0.70. This stability indicates that our method is robust to small variations in the calibration process, which is advantageous in real-world scenarios where exact calibration might be challenging.

The peak performance at $\lambda = 1$ (F1-macro = 0.7115) and

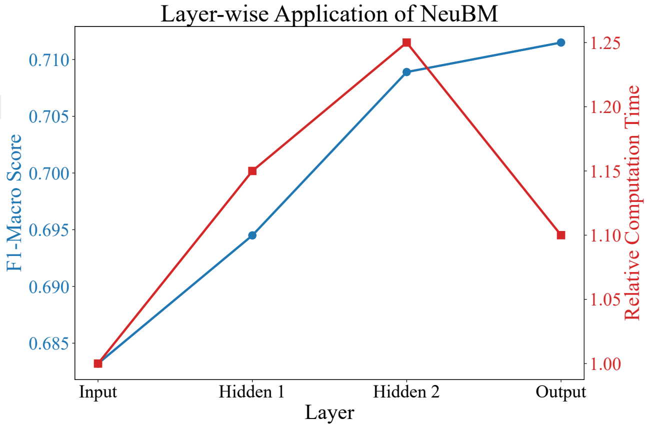

Figure 6: Performance and computational complexity of NeuBM applied at different GNN layers.

the symmetric decline on either side suggest that the neutral graph provides an unbiased reference point. Deviating from $\lambda = 1$ either under-corrects ($\lambda < 1$) or over-corrects ($\lambda > 1$) the biases, leading to suboptimal performance. This behavior underscores the effectiveness of our neutral graph construction in capturing the intrinsic biases of the model without introducing additional skew.

Application Position Study

The position at which NeuBM is applied within the model architecture can significantly impact its effectiveness. We compare applying NeuBM at different stages of the model, focusing on the logits layer and post-softmax layer.

Application Position	F1-macro	F1-weighted	F1-micro
Logits Layer (Default)	**0.7115**	**0.7029**	**0.7111**
Post-Softmax Layer	0.6892	0.6814	0.6903
Multiple Layers	0.7043	0.6957	0.7032

Table 4: Performance comparison of NeuBM application positions on Cora dataset

Table 4 shows that applying NeuBM at the logits layer yields the best performance across all metrics. Applying NeuBM after the softmax function results in a slight decrease in performance, likely due to the loss of fine-grained calibration information in probability space. Interestingly, applying NeuBM at multiple layers (both logits and intermediate layers) does not lead to further improvements and slightly increases computational cost. This suggests that a single application at the logits layer is sufficient to capture and correct class imbalance biases.

Application at Different GNN Layers

To understand the impact of NeuBM at various stages of the graph neural network, we conducted experiments applying the method at the input layer, hidden layers, and output layer of a GCN model. Figure 6 illustrates the performance and computational complexity across these settings.

The analysis reveals that applying NeuBM at deeper layers of the GNN generally yields better performance, with the output layer application achieving the highest F1-Macro score of 0.7115. This trend suggests that calibration at later stages allows the model to learn more balanced representations throughout the network. However, the performance gains are not linear, with diminishing returns observed as we move to deeper layers. In terms of computational complexity, applying NeuBM at hidden layers incurs a moderate increase in computation time, with the second hidden layer application being 1.25 times slower than the baseline. Interestingly, output layer application shows only a 1.1x increase in computation time while providing the best performance, making it an attractive trade-off between effectiveness and efficiency. Comparing single-layer and multi-layer applications, we found that applying NeuBM at multiple layers does not necessarily lead to significant performance improvements. A dual-layer application (hidden layer 2 and output layer) achieved an F1-Macro score of 0.7143, only marginally better than the single output layer application (0.7115), while increasing the computation time by 1.35x. This suggests that the benefits of multi-layer application may not justify the additional computational cost in most cases.

While our results consistently show NeuBM's effectiveness across diverse benchmarks, it remains crucial to address several practical challenges. For instance, in extremely large-scale graphs with billions of nodes, building and processing a neutral graph may introduce additional overhead unless combined with efficient sampling techniques. Furthermore, our current neutral graph construction assumes relatively consistent feature distributions across classes, which could be compromised if outlier features heavily dominate certain minority classes. Investigating these aspects and refining NeuBM's calibration strategy for highly skewed feature distributions constitute promising directions for future work.

4 Conclusion

In this work, we introduced NeuBM, an approach to mitigating class imbalance in Graph Neural Networks through neutral bias calibration. NeuBM addresses a challenge in real-world graph learning scenarios, where imbalanced class distributions often lead to biased predictions and suboptimal performance for minority classes. Through a neutral graph and adaptive calibration mechanism, NeuBM effectively recalibrates model predictions while maintaining inherent graph structures. Experimental results demonstrate NeuBM's superiority over existing methods, particularly in scenarios with severe class imbalance and limited labeled data. The method's robustness to noise, varying imbalance ratios, and generalizability across different GNN architectures establish it as a practical solution for real-world graph learning applications where balanced class performance is crucial.

While NeuBM shows promising results, further exploration is needed on dynamic or heterogeneous graph scenarios, where node types and structures may evolve over time. Moreover, investigating the theoretical underpinnings of neutral bias calibration could yield deeper insights and guide refinements. As graph-based machine learning continues to mature, NeuBM holds promise for improving fairness and accuracy across a broad spectrum of real-world applications.

Acknowledgments

The work is partially supported by the National Natural Science Foundation of China (Grant No. 62406056), the Guangdong Basic and Applied Basic Research Foundation (Grant No.2024A1515140114), Guangdong Research Team for Communication and Sensing Integrated with Intelligent Computing (Project No. 2024KCXTD047). The computational resources are supported by SongShan Lake HPC Center (SSL-HPC) in Great Bay University.

References

[Bai *et al.*, 2022] Xiang-En Bai, Jing An, Zi-Bo Yu, Han-Qiu Bao, and Ke-Fan Wang. A kernel propagation-based graph convolutional network imbalanced node classification model on graph data. In *2022 IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pages 1–6. IEEE, 2022.

[Chawla *et al.*, 2002] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[Chen *et al.*, 2021] Deli Chen, Yankai Lin, Guangxiang Zhao, Xuancheng Ren, Peng Li, Jie Zhou, and Xu Sun. Topology-imbalance learning for semi-supervised node classification. *Advances in Neural Information Processing Systems*, 34:29885–29897, 2021.

[Cui *et al.*, 2019] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019.

[Duan *et al.*, 2022] Yijun Duan, Xin Liu, Adam Jatowt, Haitao Yu, Steven Lynden, Kyoung-Sook Kim, and Akiyoshi Matono. 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*, pages 20–36. Springer, 2022.

[Hamilton *et al.*, 2017] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.

[He and Garcia, 2009] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[Huang *et al.*, 2024] Xinlei Huang, Zhiqi Ma, Dian Meng, Yanran Liu, Shiwei Ruan, Qingqiang Sun, Xubin Zheng, and Ziyue Qiao. Praga: Prototype-aware graph adaptive aggregation for spatial multi-modal omics analysis. In *AAAI 2025*, 2024.

[Johnson and Khoshgoftaar, 2019] Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of big data*, 6(1):1–54, 2019.

[Ju *et al.*, 2022] Wei Ju, Yifang Qin, Ziyue Qiao, Xiao Luo, Yifan Wang, Yanjie Fu, and Ming Zhang. Kernel-based substructure exploration for next poi recommendation. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 221–230. IEEE, 2022.

[Ju *et al.*, 2025] Wei Ju, Zhengyang Mao, Siyu Yi, Yifang Qin, Yiyang Gu, Zhiping Xiao, Jianhao Shen, Ziyue Qiao, and Ming Zhang. Cluster-guided contrastive class-imbalanced graph classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 11924–11932, 2025.

[Juan *et al.*, 2021] Xin Juan, Meixin Peng, and Xin Wang. Exploring self-training for imbalanced node classification. In *Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part V 28*, pages 28–36. Springer, 2021.

[Liu *et al.*, 2023] Zhining Liu, Ruizhong Qiu, Zhichen Zeng, Hyunsik Yoo, David Zhou, Zhe Xu, Yada Zhu, Kommy Weldemariam, Jingrui He, and Hanghang Tong. Class-imbalanced graph learning without class rebalancing. In *Forty-first International Conference on Machine Learning*, 2023.

[Ma *et al.*, 2022] Chao Ma, Jing An, Xiang-En Bai, and Han-Qiu Bao. Attention and cost-sensitive graph neural network for imbalanced node classification. In *2022 IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pages 1–6. IEEE, 2022.

[Park *et al.*, 2021] Joonhyung Park, Jaeyun Song, and Eunho Yang. Graphens: Neighbor-aware ego network synthesis for class-imbalanced node classification. In *International conference on learning representations*, 2021.

[Qian *et al.*, 2022] Yiyue Qian, Chunhui Zhang, Yiming Zhang, Qianlong Wen, Yanfang Ye, and Chuxu Zhang. Co-modality graph contrastive learning for imbalanced node classification. *Advances in Neural Information Processing Systems*, 35:15862–15874, 2022.

[Qiao *et al.*, 2019] Ziyue Qiao, Yi Du, Yanjie Fu, Pengfei Wang, and Yuanchun Zhou. Unsupervised author disambiguation using heterogeneous graph convolutional network embedding. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 910–919. IEEE, 2019.

[Qiao *et al.*, 2023] Ziyue Qiao, Xiao Luo, Meng Xiao, Hao Dong, Yuanchun Zhou, and Hui Xiong. Semi-supervised domain adaptation in graph transfer learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 2279–2287, 2023.

[Qiao *et al.*, 2025] Ziyue Qiao, Junren Xiao, Qingqiang Sun, Meng Xiao, Xiao Luo, and Hui Xiong. Towards continuous reuse of graph models via holistic memory diversification. In *The Thirteenth International Conference on Learning Representations*, 2025.

[Qu *et al.*, 2021] Liang Qu, Huaisheng Zhu, Ruiqi Zheng, Yuhui Shi, and Hongzhi Yin. Imgagn: Imbalanced network embedding via generative adversarial graph networks. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 1390–1398, 2021.

[Santos *et al.*, 2022] Francisco Santos, Junke Ye, Farzan Masrour, Pang-Ning Tan, and Abdol-Hossein Esfahanian. Facs-gcn: Fairness-aware cost-sensitive boosting of graph convolutional networks. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022.

[Shi *et al.*, 2020] Min Shi, Yufei Tang, Xingquan Zhu, David Wilson, and Jianxun Liu. Multi-class imbalanced graph convolutional network learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*, 2020.

[Song *et al.*, 2022] Jaeyun Song, Joonhyung Park, and Eunho Yang. Tam: topology-aware margin loss for class-imbalanced node classification. In *International Conference on Machine Learning*, pages 20369–20383. PMLR, 2022.

[Wang *et al.*, 2022a] Kefan Wang, Jing An, and Qi Kang. Effective-aggregation graph convolutional network for imbalanced classification. In *2022 IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pages 1–5. IEEE, 2022.

[Wang *et al.*, 2022b] Yu Wang, Charu Aggarwal, and Tyler Derr. Distance-wise prototypical graph neural network for imbalanced node classification. In *Proceedings of the 17th International Workshop on Mining and Learning with Graphs (MLG)*, 2022.

[Wang *et al.*, 2022c] Yu Wang, Yuying Zhao, Neil Shah, and Tyler Derr. Imbalanced graph classification via graph-of-graph neural networks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2067–2076, 2022.

[Wu *et al.*, 2020] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[Wu *et al.*, 2022] Lirong Wu, Jun Xia, Zhangyang Gao, Haitao Lin, Cheng Tan, and Stan 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*, pages 519–535. Springer, 2022.

[Ying *et al.*, 2018] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 974–983, 2018.

[Yun *et al.*, 2022] Sukwon Yun, Kibum Kim, Kanghoon Yoon, and Chanyoung Park. Lte4g: Long-tail experts for graph neural networks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2434–2443, 2022.

[Zeng *et al.*, 2023] Liang Zeng, Lanqing Li, Ziqi Gao, Peilin Zhao, and Jian Li. Imgcl: Revisiting graph contrastive learning on imbalanced node classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11138–11146, 2023.

[Zhang *et al.*, 2021] Ge Zhang, Jia Wu, Jian Yang, Amin Beheshti, Shan Xue, Chuan Zhou, and Quan Z Sheng. Fraudre: Fraud detection dual-resistant to graph inconsistency and imbalance. In *2021 IEEE international conference on data mining (ICDM)*, pages 867–876. IEEE, 2021.

[Zhang *et al.*, 2022] Chunhui Zhang, Chao Huang, Yijun Tian, Qianlong Wen, Zhongyu Ouyang, Youhuan Li, Yanfang Ye, and Chuxu Zhang. Diving into unified data-model sparsity for class-imbalanced graph representation learning. *arXiv preprint arXiv:2210.00162*, 2022.

[Zhao *et al.*, 2021] Tianxiang Zhao, Xiang Zhang, and Suhang 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*, pages 833–841, 2021.

[Zhou and Gong, 2023] Mengting Zhou and Zhiguo Gong. Graphsr: a data augmentation algorithm for imbalanced node classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4954–4962, 2023.

[Zhou *et al.*, 2018] Dawei Zhou, Jingrui He, Hongxia Yang, and Wei Fan. Sparc: Self-paced network representation for few-shot rare category characterization. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2807–2816, 2018.

[Zhou *et al.*, 2020] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

[Zitnik *et al.*, 2019] Marinka Zitnik, Francis Nguyen, Bo Wang, Jure Leskovec, Anna Goldenberg, and Michael M Hoffman. Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities. *Information Fusion*, 50:71–91, 2019.