Mitigating Over-Smoothing in Graph Neural Networks via Separation Coefficient Guided Adaptive Graph Structure Adjustment

Hanyang Meng, **Jielong Yang***, **Li Peng**
School of Internet of Things Engineering, Jiangnan University, Wuxi, China
7241905013@stu.jiangnan.edu.cn, jyang022@e.ntu.edu.sg, pengli@jiangnan.edu.cn

Abstract

As the number of layers in Graph Neural Networks (GNNs) increases, over-smoothing becomes more severe, causing intra-class feature distances to shrink, while heterogeneous representations tend to converge. Most existing methods attempt to address this issue by employing heuristic shortcut mechanisms or optimizing objectives to constrain inter-class feature differences. However, these approaches fail to establish a theoretical connection between message passing and the variation in inter-class feature differences, making it challenging to design methods that target the key influencing factors. To address this gap, this paper first introduces the concept of the separation coefficient, which quantifies the contraction of feature distances between classes during multi-layer message passing. Based on this theory, we propose a low-complexity, pluggable, pseudo-label-based adaptive graph structure adjustment method. This approach effectively enhances the separation coefficient of inter-class features while maintaining intra-class compactness, thereby alleviating the convergence of heterogeneous representations caused by multi-layer aggregation. Experimental results demonstrate that the proposed method significantly improves the discriminability of node representations and enhances node classification performance across various datasets and foundational models.

1 Introduction

Graph Neural Networks (GNNs) are a class of deep learning models specifically designed to handle graph-structured data. In recent years, they have achieved significant progress in tasks such as node classification [Chen *et al.*, 2023], link prediction [Wang *et al.*, 2019], and graph classification [Xu *et al.*, 2019]. The core idea of GNNs is to leverage a message passing mechanism that enables each node to iteratively aggregate the features of its neighboring nodes over multiple layers, thereby learning discriminative node or graph-level representations. However, with the increase in network depth, GNNs face a common challenge in practical applications: over-smoothing. When the network becomes too deep, node features become progressively indistinguishable due to repeated aggregation, making it difficult to distinguish nodes of different classes, and severely impairing classification or clustering performance.

To alleviate over-smoothing, several strategies have been proposed. These strategies mainly focus on either introducing specific mechanisms or adjusting the loss function to limit the feature convergence that results from excessive aggregation. For instance, GCNII [Chen *et al.*, 2020], introduces inter layer skip connections, effectively preserving local node features, and enhancing class-level feature differences during multi-layer information propagation, thereby mitigating the over-smoothing issue. The DropEdge [Rong *et al.*, 2020] method randomly removes some edges in the graph to limit the excessive aggregation of features from heterogeneous node classes during propagation, making the information flow more flexible. The Allen-Cahn method [Wang *et al.*, 2022] draws inspiration from the Allen-Cahn equation in physics by introducing a smoothing term in the model's loss function, which controls the inter-class feature differences and mitigates excessive feature aggregation.

Although these methods help alleviate the over-smoothing problem to some extent, they typically rely on empirical rules or heuristic designs, failing to theoretically reveal the relationship between the message passing process and the variation in inter-class node feature differences. This lack of theoretical guidance makes it difficult to effectively optimize the key factors that lead to over-smoothing, resulting in unstable performance on complex datasets and limiting their generalizability and scalability. Moreover, these methods often have high computational complexity, especially in large-scale graph data or high-dimensional feature cases, leading to significant increases in computational cost and inefficiency in practical applications.

To address these challenges, we first analyze the theoretical causes of over-smoothing in GNNs. We introduce the concept of separation coefficient and, through analyzing the changes in the expected representation distance between classes, propose that the separation coefficient is a key factor in quantifying the contraction of feature distances between classes during multi-layer message propagation. Further analysis shows that the separation coefficient is closely related to the graph

*Corresponding Author.

structure, and specific graph structures can effectively amplify inter-class distances and improve intra-class compactness, thereby fundamentally mitigating the phenomenon of heterogeneous representations converging. Based on this theoretical insight, we propose a pseudo-label-based adaptive graph structure adjustment method. First, we use a well-trained GNN model to obtain node predicted pseudo-labels, then selectively add or delete edges based on this pseudo-label information, while fine-tuning the GNN parameters to adaptively adjust the graph structure and mitigate the impact of over-smoothing on model performance. Unlike some traditional methods that rely on end-to-end joint training, our approach directly adjusts the graph structure based on interpretable rules, avoiding additional loss function design and large-scale training costs, while enhancing the interpretability and usability of the method. Moreover, we conduct experiments on multiple benchmark datasets, and our method demonstrates its ability to maintain effective class separation even under deep aggregation conditions, outperforming other methods designed to alleviate over-smoothing.

The main contributions of this paper are as follows:

- We describe the causes of over-smoothing in GNNs from the perspective of inter-class expected representations. By introducing the concept of the separation coefficient, we quantify the contraction of feature distances between classes during multi-layer message propagation and reveal the close relationship between this contraction and graph structure, providing a theoretical basis for optimizing graph structures.
- Based on this theory, we propose a pseudo-label-based adaptive graph structure adjustment method. This method utilizes the predicted pseudo-labels from a well-trained GNN to adaptively adjust the graph structure while fine-tuning the GNN model parameters, effectively alleviating the impact of over-smoothing on GNN performance.
- We propose a low-complexity, plug-and-play graph structure optimization algorithm to enhance inter-class separation and intra-class compactness during message passing. Experimental results demonstrate that our method significantly improves node classification performance on various benchmark datasets for both homogeneous and heterogeneous graphs.

2 Proposed Method

To address the over-smoothing problem in Graph Neural Networks (GNNs), we propose a pseudo-label-based adaptive graph structure adjustment method. Our approach is based on a theoretical analysis of the relationship between node representation changes during message passing and the graph structure, with the goal of optimizing the graph structure to maintain intra-class feature compactness and enhance inter-class feature separation, thereby alleviating the negative effects of over-smoothing.

In Section 2.1, we first present the problem statement, clearly defining the research problem and objectives. Section 2.2 provides an in-depth discussion of the causes of over-smoothing in GNNs, introducing the separation coefficient as a key factor influencing the convergence of heterogeneous representations during message passing, and revealing the relationship between the separation coefficient and the graph structure, which serves as a theoretical foundation for graph structure optimization. Section 2.3 presents a detailed description of our algorithm, which generates pseudo-labels based on the non-uniformity of node label predictions and adjusts homogeneous and heterogeneous connections according to the pseudo-label information, thereby optimizing the graph structure and effectively mitigating the over-smoothing problem.

2.1 Problem Statement

In graph neural networks (GNNs), node classification performance is fundamentally constrained by the underlying graph structure and suffers from the over-smoothing problem where node representations become increasingly similar as the network depth increases. Let $G = (V, E)$ be an undirected graph, where each node $v \in V$ is associated with a feature vector $\mathbf{x}_v \in \mathbb{R}^l$ and a label $y_v \in \mathcal{Y}$. Given an L-layer GNN model $f_\theta : (G, \mathbf{X}) \to \mathbf{Y}$ with parameters θ, where $\mathbf{X} \in \mathbb{R}^{|V| \times l}$ is the node feature matrix, the node representations $\mathbf{h}_v^{(L)}$ tend to converge:

$$\lim_{L \to \infty} \|\mathbf{h}_i^{(L)} - \mathbf{h}_j^{(L)}\|_2 \to 0, \quad \forall i, j \in V. \tag{1}$$

Our goal is to optimize the graph structure while maintaining its essential properties:

$$\max_{G' \in \mathcal{G}(G)} \mathrm{Acc}\big(f_\theta(G')\big) \quad \text{s.t.} \quad G' = (V, E'),\ \mathrm{sim}(G', G) \geq \delta, \tag{2}$$

where $\mathcal{G}(G)$ represents possible graph structures and $\mathrm{sim}(G', G) \geq \delta$ ensures structural similarity between the original and optimized graphs.

2.2 Theoretical Analysis

In this section, we first model the interactions between node features, labels, and their neighbors based on Assumption 1. Building on this, we derive the formula for calculating the expected class representation under a general message passing mechanism. Next, we analyze the changes in the expected inter-class feature distances before and after message passing and introduce the separation coefficient to quantify the variation in inter-class feature distances. Finally, we propose the conditions for enhancing the separation coefficient while ensuring intra-class compactness.

In graph-structured data, node features typically originate from some distribution, and node labels are often correlated with the labels of their neighboring nodes. This relationship is crucial for node classification tasks in GNNs. To effectively model this dependency, it is generally assumed that node features are independently sampled from a particular distribution. Additionally, the labels of neighboring nodes influence the learning process of the target node. Therefore, modeling the interactions between node features, node labels, and their neighbors' labels is key to the theoretical analysis of graph learning.

Given the above considerations, we assume the following distribution for node features and labels in our framework:

Assumption 1. *We consider a graph G, where each node i has features $\mathbf{x}_i \in \mathbb{R}^l$ and label y_i. The assumptions are as follows:*

(1) The feature vector $\mathbf{x}_i$ of node i is sampled from the feature distribution $\mathcal{F}_{y_i}$.

(2) The dimensions of the feature vector $\mathbf{x}_i$ are independent of each other.

(3) All elements of the node feature matrix $\mathbf{X}$ are bounded by a positive number B, i.e., $\max_{i,j} |\mathbf{X}[i,j]| \leq B$.

(4) The feature of node i is independent of the labels of its neighbors.

(5) The labels of node i's neighbors are independently sampled from the neighbor distribution $\mathcal{D}_{y_i}$, with the sampling process occurring $\deg(i)$ times, where $\deg(i)$ is the degree of node i. During the sampling process, the proportion of nodes belonging to the same class is p, and the proportion of nodes belonging to different classes is $\frac{1-p}{K-1}$, where K is the total number of classes.

Similar to [Ding *et al.*, 2024][Liao *et al.*, 2021], we model node features as being sampled from label-specific distributions, reflecting the similarity of features among nodes of the same class, while maintaining flexibility for intra-class natural differences. Moreover, the independence of neighbor labels aligns with the Markov property and conforms to the local dependency assumptions of graph data [Bohde *et al.*, 2024]. Although the independence between features and neighbor labels is a simplification, it does not affect practical applications, as Graph Neural Networks (GNNs) can implicitly model these complex relationships during message passing. These assumptions provide a solid foundation for theoretical analysis while ensuring the generalizability and applicability of the method.

Under Assumption 1, we perform a theoretical analysis of the message passing mechanism. Specifically, the representation h_i of node i after message passing can be expressed as:

$$h_i = \sum_{j \in \mathcal{N}(i)} \frac{1}{\deg(i)} W x_j,$$

where $\mathcal{N}(i)$ is the set of neighbors of node i, $\deg(i)$ is the degree of node i, W is the weight matrix, and x_j is the original feature of node j. Based on this formula, we can obtain the representation of node i after message passing. Furthermore, in Lemma 1, we derive the formula for the expected representation of class c, denoted as $\mathbb{E}_c[h]$.

Lemma 1. *Under Assumption 1, the expected representation h of class c can be expressed as:*

$$\mathbb{E}_c[h] = \mathbf{W}(p\mathbb{E}_c[x] + \frac{1-p}{K-1} \sum_{k \in \mathcal{Y}, k \neq c} \mathbb{E}_k[x]),$$

where $\mathbf{W}^{(k)} \in \mathbb{R}^{l \times l}$ is the transformation matrix, $\mathcal{Y}$ is the set of classes, and $\mathbb{E}_k[x]$ is the expected feature of nodes in class k.

Based on Lemma 1, our goal is to analyze the changes in class representations during message passing using the expected class representation. We prove that message passing leads to the convergence of representations from different classes. Moreover, in Theorem 1, we establish the distance relationship between the expected representations of classes c and d before and after message passing.

Theorem 1. *Under Assumption 1, the distance between the expected representations of different classes satisfies the following equation:*

$$\|\mathbf{E}_c[h] - \mathbf{E}_d[h]\|_2 \geq \sigma_{\min}(\mathbf{W}) \frac{pK-1}{K-1} \|\mathbf{E}_c[x] - \mathbf{E}_d[x]\|_2,$$

$$\|\mathbf{E}_c[h] - \mathbf{E}_d[h]\|_2 \leq \sigma_{\max}(\mathbf{W}) \frac{pK-1}{K-1} \|\mathbf{E}_c[x] - \mathbf{E}_d[x]\|_2,$$

where $\mathbb{E}_c[h]$ and $\mathbb{E}_d[h]$ are the expected representations of classes c and d, respectively, $\mathbb{E}_c[x]$ and $\mathbb{E}_d[x]$ are the expected features of nodes in classes c and d, and $\sigma_{\min}(\mathbf{W})$ is the smallest singular value of matrix $\mathbf{W}$, $\sigma_{\max}(\mathbf{W})$ is the largest singular value of matrix $\mathbf{W}$.

Proof. For the proof, please refer to Appendix. □

Since $\sigma_{\max}(\mathbf{W}) \frac{pK-1}{K-1} < 1$, Theorem 1 demonstrates that for graph neural networks satisfying Assumption 1, after multiple message passing iterations, the distance between the representations of nodes from different classes will gradually decrease. Specifically, when the proportion p of same-class neighbors in the graph remains constant, and the maximum and minimum singular values of the matrix $\mathbf{W}$ do not change significantly, the degree of this distance contraction is primarily related to the proportion p. As the number of message passing iterations increases, the representations of nodes from different classes will become increasingly similar, leading to a gradual loss of discriminability between node representations, which may eventually result in the over-smoothing problem. To more precisely describe this phenomenon, we introduce the concept of the *separation coefficient* to quantify the contraction process of the inter-class representation distance.

Definition 1 (Separation Coefficient)**.** *We define the separation coefficient as:*

$$S = \frac{\|\mathbb{E}_c[h] - \mathbb{E}_d[h]\|_2}{\|\mathbb{E}_c[x] - \mathbb{E}_d[x]\|_2}, \quad \forall c, d \in \mathcal{Y},\ c \neq d,$$

where $\mathbb{E}_c[h]$ and $\mathbb{E}_d[h]$ are the expected representations of classes c and d, respectively, $\mathbb{E}_c[x]$ and $\mathbb{E}_d[x]$ are the expected features of nodes in classes c and d, and $\|\cdot\|_2$ denotes the 2-norm.

Based on Theorem 1 and the definition of the separation coefficient, we derive a lower bound for the separation coefficient in the general message passing mechanism of Graph Neural Networks (GNNs). Specifically, the separation coefficient satisfies the following inequality:

$$\sigma_{\min}(\mathbf{W})\frac{pK-1}{K-1} \leq \mathrm{S} \leq \sigma_{\max}(\mathbf{W})\frac{pK-1}{K-1},$$

where $\sigma_{\min}(\mathbf{W})$ is the smallest singular value of the weight matrix $\mathbf{W}$, $\sigma_{\max}(\mathbf{W})$ is the largest singular value of matrix $\mathbf{W}$, p is the proportion of same-class neighbors, and K is the total number of classes.

This inequality shows that, for fixed classes, the lower bound of the separation coefficient is influenced by weight matrix $\mathbf{W}$ and the proportion p of same-class neighbors. When the maximum and minimum singular values of $\mathbf{W}$ do not decrease, increasing the proportion p of neighboring nodes from the same class will improve the lower bound of the separation coefficient. This means that a graph structure with a higher proportion of same-class neighbors helps enhance the class separability of node representations, thereby better maintaining the inter-class differences during message passing.

However, in the node classification task of GNNs, simply enhancing class separability without considering intra-class compactness may compromise classification accuracy while mitigating over-smoothing. While increasing the separation coefficient effectively improves inter-class discriminability, if the representations of nodes within the same class are too dispersed, intra-class compactness will be insufficient, potentially harming classification performance. Especially in high-dimensional feature spaces, if the inter-class representation differences increase but the representations of same-class nodes lack compactness, the classifier may struggle to distinguish same-class nodes accurately, thus reducing classification accuracy. Therefore, in Theorem 2, we provide the conditions for ensuring intra-class compactness.

Theorem 2. *For any $t > 0$, if node i belongs to class c, the probability bound that the distance between the observed feature aggregation h_i and the expected feature aggregation of class c, $\mathbb{E}_c[h]$, exceeds t is given by:*

$$P\left(\|h_i - \mathbb{E}_c[h]\|_2 \geq t\right) \leq 2l \exp\left(-\frac{\deg(i)t^2}{8p\sigma^2_{\max}(\mathbf{W})B^2}\right) + 2l \exp\left(-\frac{\deg(i)t^2}{8(1-p)\sigma^2_{\max}(\mathbf{W})B^2}\right),$$

where $\sigma_{\max}(\mathbf{W})$ denotes the largest singular value of the matrix $\mathbf{W}$, l is the feature dimensionality, and $\deg(i)$ is the degree of node i. Additionally, p represents the proportion of nodes in the graph that belong to the same class as node i, as specified in Assumption 1.

Theorem 2 proves that, for GNNs satisfying Assumption 1, the representations of nodes are close to their expected representations with a certain probability. The probability of this closeness is influenced by the node degree, the largest singular value of the weight matrix $\mathbf{W}$, and the proportion p of same-class neighbors. Specifically, as long as the number of same-class neighbors and the largest singular value of $\mathbf{W}$ do not decrease, increasing the proportion p of same-class neighbors will lead to a higher probability that the node representation is close to its expected class representation.

By combining Theorem 1 and Theorem 2, we can conclude that, when the maximum and minimum singular values of the weight matrix do not decrease, both the intra-class compactness and inter-class separability of the node representations after message passing are closely related to the graph structure. Specifically, we need to increase the proportion p of same-class neighbors while ensuring that the number of same-class neighbors does not decrease, i.e., by enhancing homogeneous connections and removing heterogeneous connections between nodes in the graph structure. This approach not only fosters greater intra-class compactness but also improves the separation coefficient, thereby enhancing the discriminability of the node representations. Through such structural adjustments, we aim to achieve the effects shown in Figure 1.

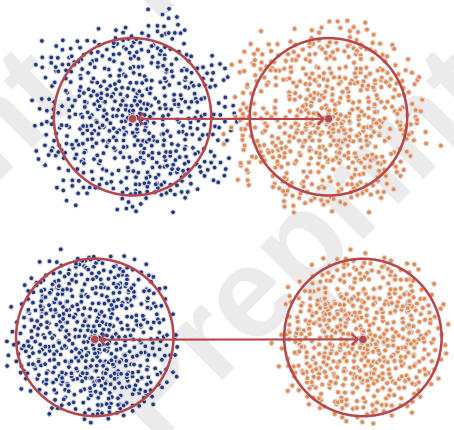

Figure 1: The figure illustrates the distribution of node representations from two classes in the feature space. The upper part shows the current distribution of node representations, where the two circles are centered at $\mathbb{E}_c[h]$ and $\mathbb{E}_d[h]$, with a radius of t, as shown in Theorem 2. The lower part depicts the expected changes in node representations after applying our method: the representations of the two classes become more concentrated within their respective circles, and the distance between the centers of the two circles increases, thereby enhancing the separability between the classes.

2.3 Method for Node Classification

In the previous section, we demonstrate that, while keeping the maximum and minimum singular values of the weight matrix invariant, adjusting the graph structure by increasing homophilic connections and reducing heterophilic connections can enhance intra-class node representation compactness and improve inter-class representation separability. Building on this theoretical insight, in this section, we propose an adaptive graph structure adjustment method based on pseudo-labels [Wang *et al.*, 2025], aimed at further optimizing the graph structure and improving downstream task performance.

This method consists of three core steps: (1) generating pseudo-labels based on label non-uniformity; (2) optimizing the graph structure according to the pseudo-labels, by increasing homophilic connections and removing heterophilic connections; (3) incorporating the optimized graph structure into the GNN training process, and performing singular value checks on the weight matrix during training. The overall framework of the algorithm is illustrated in the Figure 2.

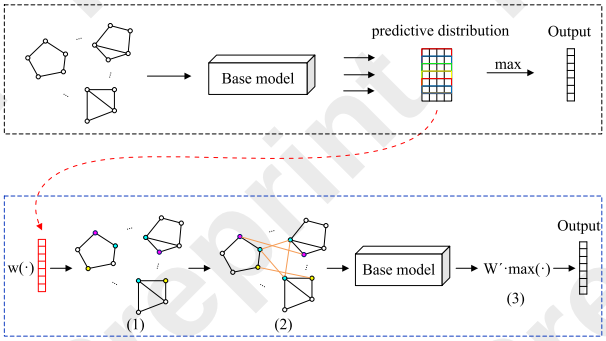

Figure 2: The figure illustrates The overall framework of our method. The upper part (black dashed box) shows the trained GNN model predicting the labels by selecting the category with the highest probability. The lower part (blue dashed box) outlines the subsequent steps of the pseudo-label-based adaptive graph structure optimization: (1) pseudo-labels are assigned based on node non-uniformity; (2) the graph structure is adjusted by adding homogeneous connections and removing heterogeneous ones; (3) during training, the weight matrix is adjusted as $\mathbf{W}' = \mathbf{W} \times \max\left(\frac{\sigma_{\min}(\mathbf{W})}{\sigma_{\min}(\mathbf{W}')}, \frac{\sigma_{\max}(\mathbf{W})}{\sigma_{\max}(\mathbf{W}')}\right)$.

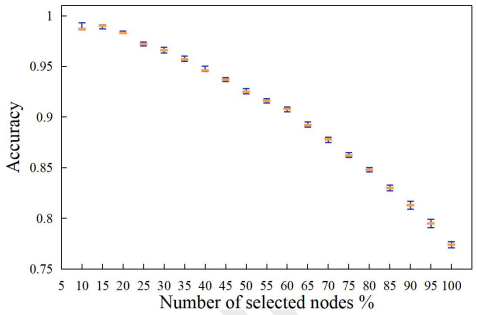

Figure 3: After the model training is completed, we sort the nodes based on their non-uniformity. Then, a subset $V' \subset V$ is selected from all nodes, with a size of $\lambda |V|$, where λ is a predefined proportion parameter. The figure illustrates the model prediction accuracy of the subset V' under different settings of λ.

By focusing on topological optimization in the post-training phase, our method aligns closely with the principles of Data-Centric AI (DCAI), emphasizing data-driven optimization to enhance downstream task performance.

To effectively generate pseudo-labels, we focus on the different dimensions of each node's predicted probability distribution and utilize label prediction non-uniformity to filter out nodes that are far from the class boundaries in classification tasks. Next, we introduce node prediction non-uniformity in Definition 2.

Definition 2 (Node Prediction Non-Uniformity)**.** *Let* $\mu_v = (\mu_v(y_1), \mu_v(y_2), \ldots, \mu_v(y_{|\mathcal{Y}|}))$ *be the predicted probability distribution vector of the base model trained on classifier* $C(\cdot)$*, with* $\mu_v(y_i) \in [0, 1]$ *for each* $y_i \in \mathcal{Y}$ *and each* $v \in \mathcal{V}$*. Moreover,* $\mathcal{Y}$ *is a finite set of labels, and* $\sum_{j=1}^{|\mathcal{Y}|} \mu_v(y_j) = 1$*. Therefore,* $\mu_v(y_j)$ *can be interpreted as the probability weight of node* v *having label* y_j*. Following [Ji* et al.*, 2023], the label non-uniformity at* v *is defined by*

$$w(v) = \sum_{j=1}^{|\mathcal{Y}|} \left| \mu_v(y_j) - \frac{1}{|\mathcal{Y}|} \right|.$$

The paper [Ji *et al.*, 2023] points out that there exists a positive correlation between the prediction label non-uniformity of nodes and classification accuracy. When the prediction label non-uniformity of nodes is high, the probability of correct classification increases significantly, as shown in Figure 3. Therefore, we assign pseudo-labels to those nodes with high prediction label non-uniformity, forming a set of nodes with pseudo-labels, denoted as V'. Based on Theorem 1 and Theorem 2, within V', we use pseudo-labels to increase homophilic connections and remove heterophilic connections between nodes, resulting in an optimized graph structure E'. Subsequently, we retrain the GNN using E' and apply a simple singular value stretching method to ensure that the maximum and minimum singular values of the weight matrix $\mathbf{W}'$ after retraining are no less than $\sigma_{\max}(\mathbf{W})$ and $\sigma_{\min}(\mathbf{W})$, respectively. The proposed pseudo-label-based adaptive graph structure adjustment algorithm is shown in Algorithm 1.

Algorithm 1 Graph Optimization with Pseudo-labeling

Input: Graph $G = (V, E, X)$, Weight matrix W trained on G, θ_1
Output: Optimized graph structure $G' = (V, E')$, Updated weight matrix W'

for each node $v \in V$ **do**
  $u_v = \text{GNN}(X(v))$
  $w(v) = \sum_{j=1}^{|\mathcal{Y}|} \left| \mu_v(y_j) - \frac{1}{|\mathcal{Y}|} \right|$
end for
Sort nodes by $w(v)$ and select the top θ_1 nodes to form V'
for each node $v' \in V'$ **do**
  Mark the pseudo-label of node v' as the category with the largest value in u_v
end for
for each pair of nodes $(v, v') \in V' \times V'$ **do**
  if the pseudo-labels of v and v' are the same **then**
    Set $E'(v, v') = 1$
  else
    Set $E'(v, v') = 0$
  end if
end for
Retrain GNN with the optimized graph $G' = (V, E')$
Stretch the weight matrix W' as follows:
$W' = W' \times \max\left(\frac{\sigma_{\min}(W)}{\sigma_{\min}(W')}, \frac{\sigma_{\max}(W)}{\sigma_{\max}(W')}\right)$

3 Experiments

In this section, we conduct experiments on several publicly available datasets, including both homogeneous and heterogeneous graphs, to evaluate the effectiveness of our proposed pseudo-label-based adaptive graph structure optimization method in semi-supervised node classification tasks. We apply this algorithmic framework to several classic graph

neural network models and compare it with multiple baseline methods to validate the relative superiority of our approach. The proposed model is named the Separation Coefficient Optimization Graph Neural Network (SCA-GNN), and the framework enhances node classification performance through simple topological structure adjustments. Specifically, when using Graph Convolution Networks (GCN) as the base model, the framework is referred to as SCA-GCN. The experimental design is aimed at addressing the following research questions:

- How much performance improvement can the proposed method achieve in semi-supervised node classification tasks compared to current mainstream baseline methods?
- What are the significant efficiency advantages of the proposed method in terms of computational complexity compared to existing methods?
- How does the proposed method effectively alleviate the over-smoothing problem caused by the increase in network depth, as observed in GNN classification performance across different network depths?
- To what extent can the proposed method improve the classification performance of different types of nodes on heterogeneous graph datasets?

3.1 Experimental Settings

Baselines

We select seven methods as baselines to evaluate the effectiveness of our proposed framework. We combine our framework with four classic base models: GCN (ICLR 2017) [Kipf and Welling, 2017], GAT (ICLR 2018) [Velickovic *et al.*, 2018], GraphCON (ICML 2022) [Rusch *et al.*, 2023], and MaskGAE (KDD 2023) [Li *et al.*, 2022], to verify whether our method can enhance the performance of these base models. Additionally, we compare our approach with three graph structure learning methods that also optimize the graph structure: Pro-GNN (KDD 2020) [Jin *et al.*, 2020], GEN (WWW 2021) [Wang *et al.*, 2021], and PRI-GSL (AAAI 2023) [Sun *et al.*, 2023], to demonstrate the advantages of our method in terms of both computational efficiency and accuracy.

Datasets

We evaluate the performance of our method on four publicly available homogeneous graph datasets (Cora, Citeseer, Pubmed, and OGBN-Arxiv) and four heterogeneous graph datasets (Texas, Chameleon, Actor and Squirrel). Notably, OGBN-Arxiv is a large-scale dataset containing 169,343 nodes and 1,166,243 edges. Detailed information about the datasets can be found in Appendix. To ensure fairness in the experiments, we use the same standard dataset splits as those in [Ji *et al.*, 2023].

Implementation Details

We first train the base graph neural network (GNN) models on the datasets and use the trained models to generate the predicted label y_v, label non-uniformity $w(v)$, and the maximum singular value $\sigma_{\max}(\mathbf{W})$ and minimum singular value $\sigma_{\min}(\mathbf{W})$ of the weight matrix $\mathbf{W}$ for each node v. Next, we sort the nodes based on label non-uniformity and select the largest portion, consisting of $\theta_1 \times |V|$ nodes, and assign their pseudo-labels as their predicted labels. To ensure the accuracy of the pseudo-labels, we fix the hyperparameter θ_1 to 0.02, consistent across all datasets. Then, based on the pseudo-labels of the nodes, we establish edges between same-class nodes and remove edges between different-class nodes to generate the edge set E'. Finally, we retrain the base model on the new graph structure and adjust the weight matrix $\mathbf{W}$ during training as follows: $\mathbf{W}' = \mathbf{W} \times \max\left(\frac{\sigma_{\min}(\mathbf{W})}{\sigma_{\min}(\mathbf{W}')}, \frac{\sigma_{\max}(\mathbf{W})}{\sigma_{\max}(\mathbf{W}')}\right)$ to ensure the stability and effectiveness of the new graph structure during training. Our framework is implemented using PyTorch, and all experiments are conducted on an NVIDIA RTX 3090 GPU.

Dataset	Cora	Citeseer	Pubmed	OGBN-Arxiv
GCN	80.65 ± 0.49	71.23 ± 0.66	79.03 ± 0.38	71.74 ± 0.29
SCA-GCN	**83.53 ± 0.78**	**72.17 ± 0.55**	**80.11 ± 0.91**	**73.16 ± 2.64**
GAT	81.91 ± 0.48	70.21 ± 0.52	78.91 ± 0.42	73.65 ± 0.11
SCA-GAT	**83.54 ± 1.07**	**71.98 ± 0.87**	**80.27 ± 1.42**	**74.41 ± 1.04**
MaskGAE	82.03 ± 0.76	70.10 ± 1.37	80.11 ± 0.51	70.67 ± 0.22
SCA-MaskGAE	**83.59 ± 0.75**	**72.61 ± 0.98**	**81.44 ± 0.95**	**72.48 ± 0.83**
GraphCON	82.36 ± 0.84	70.80 ± 1.40	79.11 ± 1.78	72.52 ± 1.74
SCA-GraphCON	**84.19 ± 1.74**	**72.71 ± 1.34**	**79.88 ± 1.25**	**73.46 ± 1.77**

Table 1: Comparison of node classification accuracy (mean ± standard deviation) between baseline models and their SCA-enhanced versions across four homogeneous datasets: Cora, Citeseer, Pubmed and OGBN-Arxiv. The bold values indicate the best performance for each dataset.

3.2 Experimental Evaluation

Performance Improvement of Our Method in Semi-supervised Node Classification (Q1)

We apply the proposed method to the base models on four homogeneous graph datasets, and the experimental results are presented in Table 1. Across all datasets, our method consistently improves the performance of the base models on the test set. The experimental results demonstrate that our pseudo-label-based adaptive graph structure optimization method significantly and stably enhances the performance of the base models in semi-supervised node classification tasks. Furthermore, our method shows good generalizability and can be easily integrated into most base models, further enhancing their performance.

Computational Complexity Advantage of Our Method (Q2)

We select three commonly used graph structure learning methods and compare their computational complexity with that of our method. The experimental results are shown in Table 2. Since our method adaptively adjusts the graph structure based on interpretable rules, it does not introduce additional training steps or parameters, thus effectively saving computational resources and training time. Compared to other more complex graph structure learning methods, our approach demonstrates significant advantages on large-scale graph datasets. The experimental results indicate that, while maintaining low complexity, our method can significantly improve node classification accuracy.

Dataset	Time Complexity	Cora	Citeseer	Pubmed	OGBN-Arxiv
GEN	$O(\tau_1 nd * (n^2 d + nk \log k) * I)$	82.00 ± 0.35	71.92 ± 0.44	79.08 ± 0.45	OOM
Pro-GCN	$O(\tau_2 I * (n^3 d + n^2))$	80.72 ± 0.75	71.22 ± 0.26	79.18 ± 0.34	OOM
PRI-GSL	$O(\tau_3 nd * (2n^2 d + nk \log k))$	81.70 ± 0.85	70.08 ± 0.90	79.43 ± 0.68	OOM
TR-GCN	$O(nd + n \log n + n^2 + d^3)$	$\mathbf{83.53 \pm 0.78}$	$\mathbf{72.17 \pm 0.55}$	$\mathbf{80.11 \pm 0.91}$	$\mathbf{73.16 \pm 2.64}$

Table 2: Node classification performance and time complexity comparison for our method and graph structure learning methods. OOM indicates Out Of Memory during training with 3090 24G GPU memory.

Dataset	Cora	Citeseer	Pubmed
Layer1	54.6 ± 14.23	39.7 ± 12.48	52.8 ± 19.14
SCA-GCN	$\mathbf{58.74 \pm 13.33}$	$\mathbf{43.58 \pm 11.95}$	$\mathbf{56.87 \pm 14.43}$
Layer2	80.65 ± 0.49	71.23 ± 0.66	79.03 ± 0.38
SCA-GCN	$\mathbf{83.53 \pm 0.78}$	$\mathbf{72.17 \pm 0.55}$	$\mathbf{80.11 \pm 0.91}$
Layer3	79.0 ± 1.28	65.9 ± 1.64	76.25 ± 0.65
SCA-GCN	$\mathbf{80.77 \pm 1.34}$	$\mathbf{67.74 \pm 2.03}$	75.95 ± 1.06
Layer4	70.87 ± 12.47	55.62 ± 11.98	67.34 ± 11.81
SCA-GCN	$\mathbf{72.67 \pm 7.47}$	$\mathbf{61.79 \pm 4.77}$	$\mathbf{69.81 \pm 8.54}$
Layer5	46.92 ± 9.76	35.87 ± 9.94	55.11 ± 9.34
SCA-GCN	$\mathbf{49.36 \pm 8.97}$	$\mathbf{47.58 \pm 7.88}$	$\mathbf{58.75 \pm 9.92}$
Layer6	36.25 ± 9.41	28.9 ± 8.08	49.5 ± 9.13
SCA-GCN	$\mathbf{39.42 \pm 10.54}$	$\mathbf{37.68 \pm 8.94}$	$\mathbf{53.12 \pm 10.23}$

Table 3: Comparison of node classification accuracy (mean $\pm$ standard deviation) for different layers of the GCN model and its SCA-enhanced version across three datasets: Cora, Citeseer and Pubmed. The bold values indicate the best performance for each layer and dataset.

Performance of Our Method in Multi-layer GNNs (Q3)

Through pseudo-label-based adaptive graph structure optimization, our method maintains intra-class compactness while enhancing the separability of heterogeneous representations, effectively alleviating the over-smoothing phenomenon that occurs in graph neural networks (GNNs) during multiple message passing iterations. To validate the effectiveness of our method in mitigating the over-smoothing problem in GNN models with different depths, we conducted semi-supervised node classification experiments with GCN models containing 1 to 6 layers on the Cora, Citeseer and PubMed datasets, comparing the performance before and after applying our method. The experimental results, shown in Table 3, indicate that although the accuracy of our method decreases as the number of GNN layers increases, it still significantly alleviates the over-smoothing problem compared to the GCN models without our method.

Performance of Our Method on Heterogeneous Graph Datasets (Q4)

Semi-supervised node classification on heterogeneous graph datasets has long been a challenging problem in graph neural networks (GNNs). Studies have shown that due to the presence of many heterogeneous connections in these datasets, models tend to experience significant over-smoothing even at shallow layers. To verify the effectiveness of our method on heterogeneous graph datasets, we applied it to base models on four heterogeneous graph datasets, and the results are shown in Table 4. The experimental results demonstrate that our method significantly improves node classification accuracy on all baseline models across heterogeneous graph datasets.

Dataset	Texas	Chameleon	Actor	Squirrel
GCN	54.60 ± 6.47	63.62 ± 3.14	38.64 ± 0.97	44.23 ± 0.31
SCA-GCN	$\mathbf{61.83 \pm 7.21}$	$\mathbf{65.88 \pm 4.64}$	$\mathbf{39.87 \pm 2.71}$	$\mathbf{46.77 \pm 3.68}$
GAT	52.70 ± 8.38	64.25 ± 3.41	35.98 ± 0.23	42.73 ± 0.33
SCA-GAT	$\mathbf{60.82 \pm 6.45}$	$\mathbf{66.44 \pm 3.48}$	$\mathbf{39.24 \pm 1.75}$	$\mathbf{43.54 \pm 1.44}$
MaskGAE	63.51 ± 5.02	57.85 ± 2.43	36.71 ± 2.47	43.29 ± 1.77
SCA-MaskGAE	$\mathbf{64.11 \pm 4.56}$	$\mathbf{58.92 \pm 3.68}$	$\mathbf{38.68 \pm 1.94}$	$\mathbf{44.17 \pm 2.57}$
GraphCON	80.81 ± 4.12	50.77 ± 2.35	39.33 ± 0.88	43.64 ± 0.52
SCA-GraphCON	$\mathbf{81.51 \pm 5.64}$	$\mathbf{55.84 \pm 3.67}$	$\mathbf{40.36 \pm 5.09}$	$\mathbf{44.91 \pm 3.64}$

Table 4: Comparison of node classification accuracy (mean $\pm$ standard deviation) between baseline models and their SCA-enhanced versions across four heterogeneous datasets: Texas, Chameleon, Actor and Squirrel. The bold values indicate the best performance for each dataset.

4 Conclusion

In this paper, we propose a pseudo-label-based adaptive graph structure optimization method aimed at addressing the over-smoothing problem in graph neural networks (GNNs). By analyzing the changes in inter-class representation distances, we introduce the concept of the separation coefficient to quantify the feature contraction process and design a low-complexity graph structure adjustment method. This method enhances the separability of inter-class features while maintaining the compactness of intra-class features. Experimental results demonstrate that our approach significantly improves node classification accuracy across multiple benchmark datasets and outperforms existing methods in terms of computational efficiency. Our research provides an effective solution to alleviate the over-smoothing problem in GNNs and offers strong support for graph structure optimization and the application of GNNs on large-scale datasets.

Acknowledgements

This work is supported by the Young Elite Scientists Sponsorship Program by CAST (No. 2023QNRC001) and the National Natural Science Foundation of China (No. 62106082). We thank Prof. Zijiang James Yang and Rui Ding for helpful discussions.

References

[Bohde *et al.*, 2024] Montgomery Bohde, Meng Liu, Alexandra Saxton, and Shuiwang Ji. On the markov property of neural algorithmic reasoning: Analyses and methods. In *The Twelfth International Conference on Learning Representations*, 2024.

[Chen *et al.*, 2020] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep

graph convolutional networks. In *International conference on machine learning*, pages 1725–1735. PMLR, 2020.

[Chen *et al.*, 2023] Yuhan Chen, Yihong Luo, Jing Tang, Liang Yang, Siya Qiu, Chuan Wang, and Xiaochun Cao. Lsgnn: towards general graph neural network in node classification by local similarity. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 3550–3558, 2023.

[Ding *et al.*, 2024] Rui Ding, Jielong Yang, Ji Feng, Xionghu Zhong, and Linbo Xie. Fr-gnn: Mitigating the impact of distribution shift on graph neural networks via test-time feature reconstruction. *IEEE Internet of Things Journal*, 2024.

[Ji *et al.*, 2023] Feng Ji, See Hian Lee, Hanyang Meng, Kai Zhao, Jielong Yang, and Wee Peng Tay. Leveraging label non-uniformity for node classification in graph neural networks. In *International Conference on Machine Learning*, pages 14869–14885. PMLR, 2023.

[Jin *et al.*, 2020] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, pages 66–74, 2020.

[Kipf and Welling, 2017] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proc. Int. Conf. Learn. Representations*, pages 1–14, 2017.

[Li *et al.*, 2022] Jintang Li, Ruofan Wu, Wangbin Sun, Liang Chen, Sheng Tian, Liang Zhu, Changhua Meng, Zibin Zheng, and Weiqiang Wang. Maskgae: Masked graph modeling meets graph autoencoders. *arXiv:2205.10053*, 2022.

[Liao *et al.*, 2021] Renjie Liao, Raquel Urtasun, and Richard Zemel. A pac-bayesian approach to generalization bounds for graph neural networks. In *International Conference on Learning Representations*, 2021.

[Rong *et al.*, 2020] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *Proc. Int. Conf. Learn. Representations*, 2020.

[Rusch *et al.*, 2023] T Konstantin Rusch, Ben Chamberlain, James Rowbottom, Siddhartha Mishra, and Michael Bronstein. Graph-coupled oscillator networks. In *Proc. Int. Conf. Mach. Learn.*, pages 18888–18909, 2023.

[Sun *et al.*, 2023] Qingyun Sun, Jianxin Li, Beining Yang, Xingcheng Fu, Hao Peng, and S Yu Philip. Self-organization preserved graph structure learning with principle of relevant information. In *Proc. AAAI Conf. Artif. Intell.*, pages 4643–4651, 2023.

[Velickovic *et al.*, 2018] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *Proc. Int. Conf. Learn. Representations*, 2018.

[Wang *et al.*, 2019] Zihan Wang, Zhaochun Ren, Chunyu He, Peng Zhang, and Yue Hu. Robust embedding with multi-level structures for link prediction. In *IJCAI*, pages 5240–5246, 2019.

[Wang *et al.*, 2021] Ruijia Wang, Shuai Mou, Xiao Wang, Wanpeng Xiao, Qi Ju, Chuan Shi, and Xing Xie. Graph structure estimation neural networks. In *Proc. Web Conf.*, pages 342–353, 2021.

[Wang *et al.*, 2022] Yuelin Wang, Kai Yi, Xinliang Liu, Yu Guang Wang, and Shi Jin. Acmp: Allen-cahn message passing for graph neural networks with particle phase transition. *arXiv preprint arXiv:2206.05437*, 2022.

[Wang *et al.*, 2025] Jinlu Wang, Yanfeng Sun, Jiapu Wang, Junbin Gao, Shaofan Wang, and Jipeng Guo. Contrastive learning meets pseudo-label-assisted mixup augmentation: A comprehensive graph representation framework from local to global. *arXiv:2501.18357*, 2025.

[Xu *et al.*, 2019] Nuo Xu, Pinghui Wang, Long Chen, Jing Tao, and Junzhou Zhao. Mr-gnn: multi-resolution and dual graph neural network for predicting structured entity interactions. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 3968–3974, 2019.