Improving Distinguishability of Class for Graph Neural Networks

Dongxiao He[1], **Shuwei Liu**[1], **Meng Ge**[2], **Zhizhi Yu**[1*], **Guangquan Xu**[1], **Zhiyong Feng**[1]

[1]College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]Saw Swee Hock School of Public Health, National University of Singapore, Singapore
{hedongxiao, liusw, yuzhizhi, losin, zyfeng}@tju.edu.cn, gemeng@nus.edu.sg

Abstract

Graph Neural Networks (GNNs) have received widespread attention and applications due to their excellent performance in graph representation learning. Most existing GNNs can only aggregate 1-hop neighbors in a GNN layer, so they usually stack multiple GNN layers to obtain more information from larger neighborhoods. However, many studies have shown that model performance experiences a significant degradation with the increase of GNN layers. In this paper, we first introduce the concept of distinguishability of class to indirectly evaluate the learned node representations, and verify the positive correlation between distinguishability of class and model performance. Then, we propose a Graph Neural Network guided by Distinguishability of class (Disc-GNN) to monitor the representation learning, so as to learn better node representations and improve model performance. Specifically, we first perform inter-layer filtering and initial compensation based on Local Distinguishability of Class (LDC) in each layer, so that the learned node representations have the ability to distinguish different classes. Furthermore, we add a regularization term based on Global Distinguishability of Class (GDC) to achieve global optimization of model performance. Extensive experiments on six real-world datasets have shown that the competitive performance of Disc-GNN to the state-of-the-art methods on node classification and node clustering tasks.

Introduction

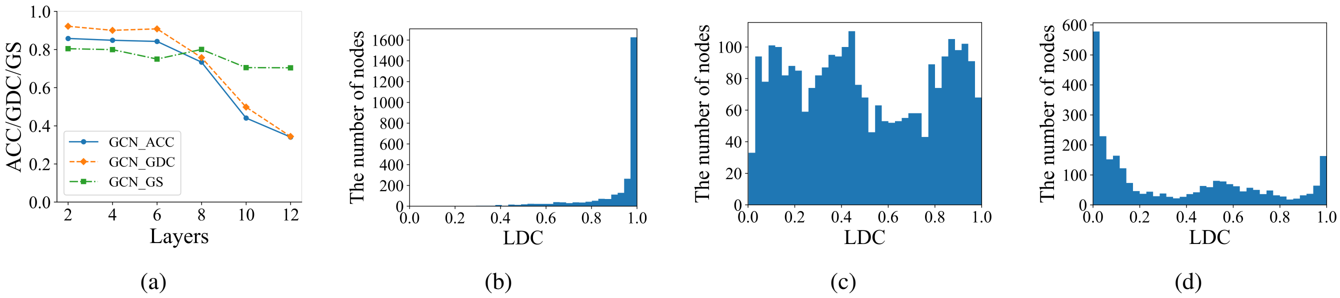
Graphs are ubiquitous in the real-world, and many scenarios such as protein–protein interactions (Yang et al. 2020), financial transactions (Lu et al. 2022), and social relationships (Jin et al. 2023b; Yu et al. 2021) can be modeled as graphs, where nodes denote entities and edges represent relationships between entities. Graph Neural Networks (GNNs), which can make full use of node features and graph topology to learn node representations on graphs, have been widely applied in node-level (Jin et al. 2021, 2022a), edge-level (Jin et al. 2023a; Yu et al. 2023) and graph-level (Zhang et al. 2018) tasks.

Most GNN models learn node representations in each layer by aggregating and transforming information from 1-hop neighbors, leading to localized learning of node representations. In order to learn complex representations at different levels of abstraction from larger neighborhoods, many efforts have been devoted to stacking multiple GNN layers. However, existing studies have shown that as GNN layers stack, the performance of GNN models significantly decline (Li, Han, and Wu 2018; Chen et al. 2020a). This begs the question, how does the node representations change during the stacking of GNN layers, leading to a degradation of the model performance?

Considering that most GNN models (Velickovic et al. 2018; Jin et al. 2018, 2023a) convert the final node representations into a class probability matrix to predict node labels, we indirectly analyze the changes in node representations by observing the distribution change of the class probability matrix. To this end, we define two metrics based on the class probability matrix, that is, Global Distinguishability of Class (GDC) and Local Distinguishability of Class (LDC), to measure the average distinguishability of the whole graph on classes and the distinguishability of each node on classes, respectively. Specifically, taking Graph Convolutional Network (GCN) (Kipf and Welling 2017), a classic GNN model, as an example, we first observe the changes in node classification accuracy (ACC) and GDC as GCN layers stack. As shown in Figure 1a, both ACC and GDC show a decreasing trend with the increase of GCN layers, which indicates a positive correlation between these two metrics. Furthermore, considering many studies attribute the degradation in model performance to over-smoothing issue (Li, Han, and Wu 2018; Li et al. 2019), we also represent the global smoothness (GS) by calculating the average distance among node representations. As shown in Figure 1a, we find that GS suddenly increases in 8-th layer and then drops sharply. This indicates that GCN suffers over-smoothing only after a certain layers are reached. In addition, we calculate the pearson correlation coefficient of 0.9968 between GDC and ACC, while 0.7877 between GS and ACC. This further proves that changes in distinguishability of class can be better used to describe changes in node representations.

For further observing the ability of each node to distinguish different classes in various GCN layers, we analyze the distribution of LDC in the 2-nd, 10-th and 12-th GCN layers on Cora dataset. As shown in Figure 1b-1d, the LDC of most nodes gradually transitions from 1.0 to 0.0 with the increase of GCN layers. This is mainly be-

[*]Corresponding author.



(a) (b) (c) (d)

Figure 1: (a) With the stacking of GCN layers, the trends of ACC, GDC and GS on Cora dataset. (b)-(d) The distribution of LDC on Cora dataset in the 2-nd, 10-th and 12-th GCN layers, respectively.

cause stacking more GNN layers not only aggregates too many neighbors from other classes (i.e., heterophilic neighbors) (Fang et al. 2022), but also makes too many interactions between different dimensions of the node representations (Chen et al. 2020b), ultimately leading to the decrease in LDC and model performance. In addition, from Figure 1c and 1d, we can find that there are some nodes with LDC close to 1.0, indicating that these nodes can learn good representations with high distinguishability of class from multi-hop homophilic neighbors. Therefore, we consider how to learn node representations with high distinguishability of class for each node?

To tackle aforementioned question, we propose a Graph Neural Network guided by Distinguishability of class (Disc-GNN) to ensure that all nodes have ability to distinguish different classes and enable learn better representations from larger neighborhoods. Specifically, we first design a gating mechanism based on LDC to filter node representations, so as to ensure node representations with high distinguishability of class in each layer. At the same time, we introduce a certain degree of initial compensation to prevent the nodes from losing their own feature. Finally, to ensure the maximization of distinguishability of class, we add a GDC regularization term to the objective function to globally optimize the training process. The contribution of this paper is summarized as follows:

- We design two new metrics named Global Distinguishability of Class (GDC) and Local Distinguishability of Class (LDC) to evaluate the changes in node representations as GNN layers deepen.
- We propose a novel Graph Neural Network guided by Distinguishability of class (Disc-GNN) to ensure that the learned node representations have a certain ability to distinguish between different classes.
- Experimental results on both node classification and node clustering tasks demonstrate the effectiveness of our proposed Disc-GNN.

Preliminaries

Notations

Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X)$, where $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ is the set of N nodes and $\mathcal{E}$ is the set of M edges. Each node $v \in \mathcal{V}$ is associated with the feature vector $x_v \in R^F$ and $X \in R^{N \times F}$ denotes the node feature matrix. Let $A \in R^{N \times N}$ represents a binary symmetric adjacency matrix such that $a_{ij} \in \{0, 1\}$, where $a_{ij} = 1$ if there is an edge between nodes v_i and v_j, otherwise $a_{ij} = 0$. The corresponding degree matrix is $D = diag(\{d_1, d_2, \dots, d_N\}) \in R^{N \times N}$ and $d_i = \sum_{(v_i, v_j) \in \mathcal{E}} a_{ij}$. Thus, we can also describe the graph as $\mathcal{G} = (A, X)$.

Graph Neural Networks

The core of GNNs is the message passing mechanism (Gilmer et al. 2017) which combines node features and graph topology. More specifically, the information from neighbors is transmitted to the node along topology, and the node aggregates and transforms neighbor information to update its own representation. Without loss of generality, given a node v_i, the message-passing process in l-th GNN layer can be formulated as:

$$\begin{aligned} \tilde{h}_i^{(l)} &= \mathrm{AGGREGATE}(h_j^{(l-1)} | v_j \in \mathcal{N}_i \cup \{v_i\}), \\ h_i^{(l)} &= \mathrm{TRANSFORM}(\tilde{h}_i^{(l)}), \end{aligned} \tag{1}$$

where $\mathcal{N}_i$ represents 1-hop neighbors of node v_i, and $h_i^{(l)}$ denotes the learned representation of node v_i in l-th GNN layer. AGGREGATE and TRANSFORM are individually designed based on specific GNN models. Graph Convolutional Network (GCN) is one of the classic GNN models, which adopts the aggregator with fixed weight and nonlinear transformation to learn node representations. Thus, GCN can rewrite Eq.(1) into the following matrix form:

$$\begin{aligned} \tilde{H}^{(l)} &= \tilde{P} H^{(l-1)}, \\ H^{(l)} &= \sigma(\tilde{H}^{(l)} W^{(l)}), \end{aligned} \tag{2}$$

where $\tilde{P} = (D + I_N)^{-1/2}(A + I_N)(D + I_N)^{-1/2}$ is the symmetrically normalized adjacency matrix with self-loops. $W^{(l)}$ is the trainable weight matrix and σ is a activation function such as $\mathrm{ReLU}(\cdot)$.

Distinguishability of Class

For downstream tasks such as node classification and node clustering, the learned node representations need to have good distinguishability for different classes. To measure

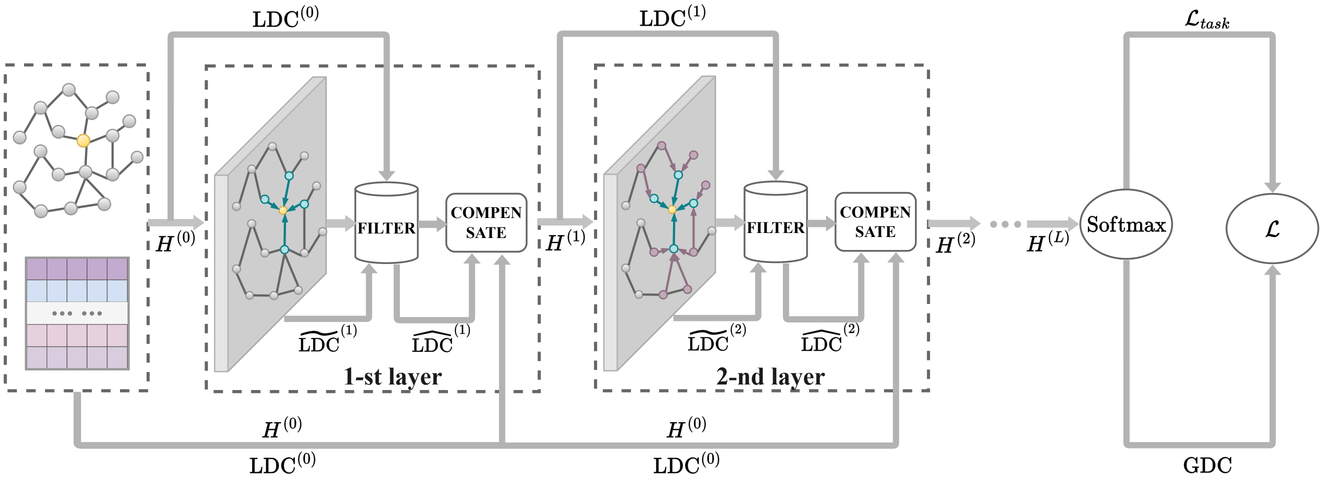


Figure 2: The framework of Disc-GNN.

whether the learned node representations can effectively distinguish different classes, we propose a quantitative metric Local Distinguishability of Class (LDC) as:

$$z_i^{(l)} = \text{Softmax}(h_i^{(l)} \hat{W}^{(l)}), \tag{3}$$

$$\text{LDC}_i^{(l)} = \max(z_i^{(l)}) - \min(z_i^{(l)}), \tag{4}$$

where $\hat{W}^{(l)} \in R^{D \times C}$ is the mapping matrix which maps $h_i^{(l)}$ from D-dimensions to C-classes. For node v_i, the class probability vector $z_i^{(l)}$ is further obtained by the Softmax classifier, and $\sum_{c \in C} z_{ic}^{(l)} = 1$. $\text{LDC}_i^{(l)}$ with the range of $[0.0, 1.0]$ represents the distance between the maximum and minimum class probabilities, which reflects the LDC of node representation $h_i^{(l)}$. Specifically, if $\max(z_i^{(l)})$ is closer to 1.0 and $\min(z_i^{(l)})$ is closer to 0.0, then LDC_i tending to 1.0 to represent that node v_i clearly belongs to the class with the greatest probability. Otherwise, LDC_i tends to 0.0, indicating that the node representation $h_i^{(l)}$ is confused by information from different classes, resulting in a loss of ability to distinguish classes. In this way, the LDC vector of l-th layer can be formulated as: $\text{LDC}^{(l)} = \{\text{LDC}_1^{(l)}, \text{LDC}_2^{(l)}, \ldots, \text{LDC}_N^{(l)}\}$.

By calculating the mean value among $\text{LDC}^{(l)}$, the Global Distinguishability of Class $\text{GDC}^{(l)}$ can be defined as:

$$\text{GDC}^{(l)} = \frac{1}{N} \sum_{v_i \in \mathcal{V}} \text{LDC}_i. \tag{5}$$

Our Proposed Model: Disc-GNN

In this section, we propose a Graph Neural Network guided by Distinguishability of class (Disc-GNN). We start with a brief overview and then introduce the details of components.

Overview

To maintain the ability to distinguish different classes during the learning process of node representations, we propose a new method Disc-GNN, which guides the learning of node representations by combining LDC and GDC. Our method not only locally supervises the representation learning for each node, but also improves model performance through global optimization. Specifically, Disc-GNN can be divided into three components: inter-layer filtering based on LDC, initial compensation based on LDC, as well as global optimization based on GDC. The whole framework of our Disc-GNN is shown in Figure 2. Compared to blindly stacking multiple GNN layers, we first adopt LDC to supervise the representation learning in each layer for each node. That is, if the node representation has high distinguishability for different classes, we need to retain this representation in the inter-layer learning. Otherwise, we would filter out the node representation with low distinguishability of class. At the same time, we introduce initial compensation to ensure that each node retain its original features while aggregating other neighbor information. Finally, we use GDC to optimize the overall node representations, maximizing the ability of all the learned node representations to distinguish between different classes and improving the model performance.

Inter-Layer Filtering Based on LDC

Due to the message passing mechanism in GNNs, node representations can easily be transmitted along the graph topology to all nodes. It is urgent to filter out node representations with low distinguishability of class, aiming to prevent them from spreading to subsequent layers and confusing the representation learning of other nodes. Therefore, we design a LDC-based gating mechanism to filter out node representations that cannot distinguish between different classes:

$$\varphi_i^{(l)} = \begin{cases} 1, & \widetilde{\text{LDC}}_i^{(l)} > \text{LDC}_i^{(l-1)} + \epsilon \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where ϵ is a relaxation factor. $\text{LDC}_i^{(l-1)}$ and $\widetilde{\text{LDC}}_i^{(l)}$ correspond to the final representations generated in (l-1)-th layer and the learned representations generated by the message passing mechanism in l-th layer, respectively. And $\varphi_i^{(l)}$ is the filtering parameter of node v_i in l-th layer. As mentioned in introduction, frequent interactions between different dimensions reduce the distinguishability of class on node representations, leading to the degradation in model performance. Therefore, for node v_i, if it can learn the representation with high distinguishability of class $\widehat{\mathrm{LDC}}_i^{(l)}$, it needs to retain this learned representation, and $\varphi_i^{(l)}$ is set to 1. Otherwise, it is necessary to filter out the learned representations with low distinguishability of class caused by frequent dimensional interactions and set $\varphi_i^{(l)}$ to 0. By synthesizing the filtering parameters of all nodes, we can generate the filtering diagonal matrix $\varPhi^{(l)} = diag(\{\varphi_1^{(l)}, \varphi_2^{(l)}, \dots, \varphi_N^{(l)}\})$ for l-th layer. Through introducing the LDC-based gating mechanism, the representation learning in l-th layer can then be written as:

$$\hat{H}^{(l)} = \sigma((I - \varPhi^{(l)})\tilde{H}^{(l)} + \varPhi^{(l)}\tilde{H}^{(l)}W^{(l)}), \qquad (7)$$

where I is a identity matrix, and the learned node representations in l-th layer are retained with the probability of $\varPhi^{(l)}$ and filtered out with the probability of $(I - \varPhi^{(l)})$. The inter-layer filtering based on LDC effectively prevents these node representations with low distinguishability of class from being transmitted to subsequent layers, alleviating the performance degradation in deep layers.

Initial Compensation Based on LDC

With the stacking of GNN layers, the initial features of the nodes suffer severe losses (Miao et al. 2023). This is not only because a large amount of heterophilic information introduced by multiple aggregations confuses the initial features of the nodes, but also because the repeated multiplication with weight matrices brought about by multiple transformations results in diminishing feature reuse. In order to enable nodes to learn better representations from larger neighborhoods while also retaining their initial features to a certain extent, we further add initial compensation in the l-th layer. We denote initial compensation based on LDC for l-th layer as:

$$H^{(l)} = (I - \varLambda^{(l)})H^{(0)} + \varLambda^{(l)}\hat{H}^{(l)}, \qquad (8)$$

where $H^{(0)}$ with the same dimension as $H^{(l)}$ is the initial representation matrix converted from the initial feature X. And $\varLambda^{(l)}$ is defined as the compensating diagonal matrix:

$$\varLambda^{(l)} = diag\left(\frac{\widehat{\mathrm{LDC}}^{(l)}}{\mathrm{LDC}^{(0)} + \widehat{\mathrm{LDC}}^{(l)} + \mu}\right), \qquad (9)$$

where $\mathrm{LDC}^{(0)}$ and $\widehat{\mathrm{LDC}}^{(l)}$ are generated based on $H^{(0)}$ and $\hat{H}^{(l)}$, respectively. And $\mu > 0$ is a constant that prevents the denominator from being 0. By comparing l-th layer's $\widehat{\mathrm{LDC}}^{(l)}$ and the initial layer's $\mathrm{LDC}^{(0)}$, we can evaluate the degree to which initial compensation needs to be introduced without additional hyper-parameters.

Global Optimization Based on GDC

As mentioned in introduction, we find a significant positive correlation between GDC and ACC which is used to measure model performance in node classification task. In order to further optimize the model performance globally, we add a GDC regularization term to the training objective function:

$$\arg\min_{\Theta} \mathcal{L} = \arg\min_{\Theta}(\mathcal{L}_{task} - \eta \mathrm{GDC}^{(L)}), \qquad (10)$$

where Θ is the parameters that the model needs to learn and L represents the depth of the model. $\mathcal{L}_{task}$ is the loss function related to specific downstream tasks. $\mathrm{GDC}^{(L)}$ denotes the average ability of the final learned representations to distinguish different classes and η is the regularization coefficient that controls the influence of GDC. By adding the GDC regularization term to the objective function, the model can maximize distinguishability of class during the training process, thereby improving model performance.

Datasets	#Nodes	#Edges	#Features	#Classes
Cora	2,708	5,429	1,433	7
Citeseer	3,327	4,732	3,703	6
Pubmed	19,717	44,338	500	3
Wisconsin	251	499	1,703	5
Texas	183	309	1,703	5
Cornell	183	295	1,703	5

Table 1: Statistics of datasets.

Experiments

To demonstrate the effectiveness of our proposed Disc-GNN, we first compare it with nine state-of-the-art GNN models in the downstream tasks of node classification and node clustering, respectively. Then, we analyse the performance of Disc-GNN under different layers to prove that our method can not only prevent a sharp decline in model performance in deep layers, but also learn better node representations from multi-hop homophilic neighbors. Finally, the ablation study is given to explain the contribution of individual components in Disc-GNN.

Experimental Setup

Datasets. Six real-world datasets with varying sizes and features are used to comprehensively evaluate the performance of our proposed Disc-GNN. The statistical information of datasets is summarized in Table 1. Specifically, the datasets can be divided into two categories:

- Homophilic datasets: We choose three common citation graphs, i.e., Cora, Citeseer, and Pubmed (McCallum et al. 2000; Sen et al. 2008), as homophilic datasets. These citation graphs represent papers as nodes which are characterized by the bag-of-words vectors of the papers. Labels are the research field and edges are used to denote the citation relationship between two papers.
- Heterophilic datasets: We also select three web graphs, i.e., Wisconsin, Texas, and Cornell (Pei et al. 2019), as heterophilic graphs. These web graphs represent web pages and hyperlinks as nodes and edges, respectively. Nodes are associated with the bag-of-words representation of the corresponding web page, and labels denotes page classes (student, project, course, staff, and faculty).

Metric	Models	Cora	Citeseer	Pubmed	Wisconsin	Texas	Cornell
ACC (%)	GCN	85.79 ±1.32	72.12 ±0.94	87.30 ±0.64	51.18 ±6.04	47.84 ±8.38	50.54 ±8.97
	GAT	86.68 ±1.62	73.42 ±0.97	85.08 ±0.70	53.73 ±6.15	54.05 ±5.67	56.22 ±6.82
	SGC	86.10 ±1.08	72.92 ±1.36	77.21 ±0.94	50.59 ±6.25	54.59 ±3.97	55.68 ±8.04
	JK-Net	86.91 ±1.48	73.53 ±1.04	<u>88.60 ±0.59</u>	50.78 ±6.59	58.65 ±5.93	52.16 ±4.84
	APPNP	87.30 ±1.43	73.56 ±1.01	87.55 ±0.60	53.14 ±6.35	49.46 ±9.90	59.46 ±8.29
	DAGNN	86.64 ±1.54	72.78 ±1.72	88.26 ±0.42	58.63 ±7.82	54.32 ±6.22	60.54 ±6.53
	DeCorr	87.31 ±1.19	74.42 ±1.31	88.02 ±3.64	<u>82.23 ±6.15</u>	<u>79.00 ±6.02</u>	<u>81.43 ±6.65</u>
	FAGCN	86.82 ±1.79	74.10 ±1.67	88.12 ±0.70	78.24 ±5.71	73.10 ±5.40	72.38 ±5.93
	GCNII	<u>88.09 ±1.56</u>	<u>75.97 ±1.86</u>	88.01 ±0.69	79.53 ±3.42	77.16 ±3.86	79.73 ±4.72
	Disc-GNN	**88.73 ±1.48**	**76.65 ±1.11**	**88.84 ±0.55**	**85.45 ±2.51**	**81.45 ±4.59**	**83.89 ±4.26**

Table 2: Comparison on node classification in terms of ACC (%). Bold and underline are adopted to display the best and the second best results.

Baselines. To verify the effectiveness of our proposed Disc-GNN, nine methods are employed as the baselines with default hyper-parameters. They are divided into 3 categories:

- classic GNN models: GCN (Kipf and Welling 2017) is the most classic GNN models, which adopts the aggregator with fixed weight and nonlinear transformation to update node representations in each layer. GAT (Velickovic et al. 2018) further expands the aggregation method, and assigns the aggregation weight to each neighbor by the self-attention mechanism. SGC (Wu et al. 2019) decouples the message passing mechanism, and removes nonlinear transformation between consecutive layers to reduce the excessive complexity.
- GNN models that only alleviate over-smoothing issue: JK-Net (Xu et al. 2018), APPNP (Klicpera, Bojchevski, and Günnemann 2019) and DAGNN (Liu, Gao, and Ji 2020) extend the depth of GNN architecture. Specifically, APPNP adds initial features to the learning of each interlayer representation to alleviate the loss of node feature, while JK-Net and DAGNN retain all learned inter-layer representations in the output layer through skip connections and adaptively combine these hidden representations to generate the final representation.
- GNN models that alleviate both over-smoothing and heterophily issues: DeCorr (Jin et al. 2022b), FAGCN (Bo et al. 2021) and GCNII (Chen et al. 2020b) further extend deep architecture to heterophilic graphs. FAGCN enables the node to distinguish between homophilic information and heterophilic information; GCNII adopts identity mapping and initial residual connection, which can aggregate multi-hop homophilic neighbors; DeCorr prevents node-wise over-smoothing and feature-wise overcorrelation in node representations, thereby alleviating both over-smoothing and heterophily issues.

Parameter setting. All methods are implemented in Pytorch with Adam optimizer (Kingma and Ba 2015). We run 10 times and report the mean values with standard deviation. To ensure fair comparisons, we follow the default setting proposed in the original papers for all these state-of-the-art methods, including the number of hidden units, activation functions, learning rate, L2 regularization, etc. In order to explore the performance of the model in deep layers, we vary the number of layers L from the set $\{2, 5, 10, 15, 20\}$. For Disc-GNN, the hyper-parameter settings are as follow: learning rate is 0.01, dropout in $[0.4, 0.6]$, weight decay in $[1e-2, 5e-4]$, regularization coefficient η in $[0, 0.5]$, relaxation factor ϵ in $[-0.5, 0.5]$ and μ is $1e-4$.

Node Classification

In node classification task, each node is assigned a unique label. We adopt the semi-supervised learning scenario, where a portion of node labels are used for training, while the remaining node labels are masked for testing. We compare the classification accuracy (ACC) between predicted labels and ground truth labels to evaluate the model performance. Table 2 reports the mean ACC with the standard deviation.

As shown, our proposed Disc-GNN exhibits competitive performance over the other nine baseline models, especially on the heterophilic datasets, demonstrating Disc-GNN's ability to distinguish between different classes for the nodes. And it also reflects our designed two metrics LDC and GDC can effectively guide representation learning on both homophilic and heterophilic datasets. In addition, compared with GAT and FAGCN, which calculate the contribution score of each neighbor to the node from a microscopic perspective, Disc-GNN outperforms them in most datasets as it can evaluate the contribution of the entire l-order neighborhood to the node from a macroscopic perspective, thereby learning a more generalized model.

Currently, most existing methods for analyzing node representations mainly focus on node-wise smoothness, such as calculating the distance between node representations. However, DeCorr pays attention to the feature dimensions of the representations and supervises the representation learning by preventing over-correlation between the different dimensions. Our Disc-GNN indirectly analyzes the relationship between the various dimensions by calculating the distance between the maximum and minimum class probabilities. These two methods can effectively prevent confusion between various feature dimensions and allow each feature to fully leverage its contribution to representation learning. From Table 2, it can be seen that the performance of DeCorr and Disc-GNN exceed most baselines, especially on

Metrics	Models	Cora	Citeseer	Pubmed	Wisconsin	Texas	Cornell
NMI (%)	GCN	75.73 ±1.31	55.12 ±1.27	55.62 ±0.19	28.47 ±2.14	26.35 ±4.04	27.89 ±5.22
	GAT	74.16 ±0.73	53.48 ±0.66	49.84 ±0.62	20.15 ±3.08	19.63 ±2.49	16.49 ±4.09
	SGC	75.51 ±0.66	55.14 ±0.85	35.65 ±0.50	27.46 ±3.19	25.96 ±2.87	27.25 ±3.64
	JK-Net	78.55 ±1.05	54.57 ±0.96	62.50 ±0.33	32.29 ±3.19	34.60 ±4.17	31.06 ±3.99
	APPNP	78.89 ±0.84	56.26 ±1.20	55.00 ±0.34	32.44 ±2.61	27.36 ±3.36	32.37 ±3.42
	DAGNN	78.61 ±0.98	55.97 ±0.71	57.98 ±0.99	40.95 ±2.44	36.74 ±3.25	41.43 ±3.46
	DeCorr	77.91 ±1.23	55.40 ±1.11	57.84 ±0.32	43.29 ±3.15	55.38 ±2.95	47.54 ±4.50
	FAGCN	76.38 ±1.15	55.38 ±0.79	54.26 ±0.78	46.96 ±3.39	42.00 ±6.26	40.48 ±4.04
	GCNII	77.22 ±0.83	56.37 ±0.80	58.08 ±0.90	63.35 ±4.81	65.49 ±3.45	63.63 ±2.95
	Disc-GNN	**80.66 ±1.08**	**58.56 ±0.90**	**62.70 ±0.45**	**70.87 ±2.34**	**69.43 ±2.07**	**69.59 ±4.89**
ARI (%)	GCN	79.18 ±1.10	59.20 ±1.42	62.18 ±0.28	22.34 ±5.38	26.85 ±4.09	29.05 ±8.29
	GAT	76.38 ±1.48	56.61 ±0.63	54.69 ±1.12	16.71 ±3.46	21.83 ±3.23	15.35 ±6.28
	SGC	78.95 ±0.72	59.10 ±1.09	35.02 ±0.72	23.15 ±6.06	24.58 ±4.66	27.52 ±7.22
	JK-Net	82.04 ±1.05	58.76 ±1.12	**64.08 ±0.73**	27.99 ±4.20	33.07 ±5.02	33.88 ±5.00
	APPNP	82.24 ±0.79	60.42 ±1.33	59.74 ±0.55	27.51 ±4.43	27.11 ±5.34	32.70 ±5.43
	DAGNN	81.71 ±1.20	60.15 ±0.79	54.94 ±1.86	37.30 ±4.26	36.54 ±5.60	44.44 ±6.47
	DeCorr	81.21 ±1.20	56.21 ±1.36	44.12 ±1.32	26.18 ±6.21	26.99 ±4.95	27.72 ±7.40
	FAGCN	81.57 ±1.27	58.49 ±1.03	56.84 ±1.40	51.04 ±4.05	46.23 ±4.79	38.53 ±6.01
	GCNII	81.12 ±1.42	60.60 ±0.83	61.70 ±1.96	67.06 ±4.57	70.17 ±6.93	70.57 ±3.37
	Disc-GNN	**84.10 ±0.97**	**62.15 ±1.03**	63.96 ±0.70	**71.49 ±6.32**	**72.26 ±5.50**	**74.72 ±6.45**

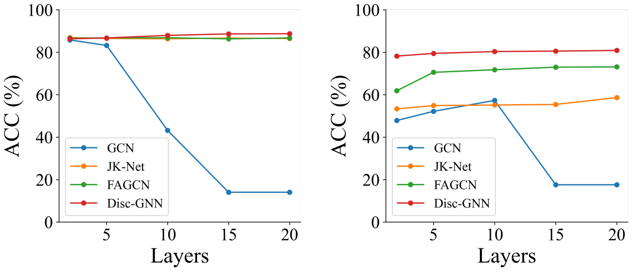
Table 3: Comparison on node clustering in terms of NMI (%) and ARI (%). Bold and underline are adopted to display the best and the second best results.

heterophilic datasets. These results further provide a new supplementary perspective for future research, which is to improve model performance though enhancing the model's learning ability for different feature dimensions.

Node Clustering

For node clustering task, the learned node representations are used as the input to a clustering model. Here we employ the k-means algorithm (Hartigan and Wong 1979) to cluster the data and evaluate the clustering performance in terms of normalized mutual information (NMI) and adjusted rand index (ARI). The experimental results are shown in Table 3.

As shown, our Disc-GNN can outperforms most of the baselines both on homophilic and heterophilic datasets. This is due to the fact that Disc-GNN adpots distinguishability of class to guide the learning of node representations, making the learned representations as capable of distinguishing different classes. In addition, compared with shallow classic GNN models, i.e., GCN, GAT, deep GNN models such as APPNP, GCNII, and our Disc-GNN can obtain richer neighbor information to assist in representation learning. Especially on heterophilic datasets, neighbors with the same class often exist in higher-order neighborhoods, and deep GNN models can effectively aggregate homophilic neighbors within higher-order neighborhoods. Therefore, most deep GNN models have better node clustering performance than shallow GNN models. For example, compared to GCN, our Disc-GNN improves NMI by 4.93% and ARI by 4.92% on homophilic Cora dataset, while it improves NMI by 43.08% improvement in NMI and ARI by 45.41% on heterophilic Texas dataset.



(a) Cora

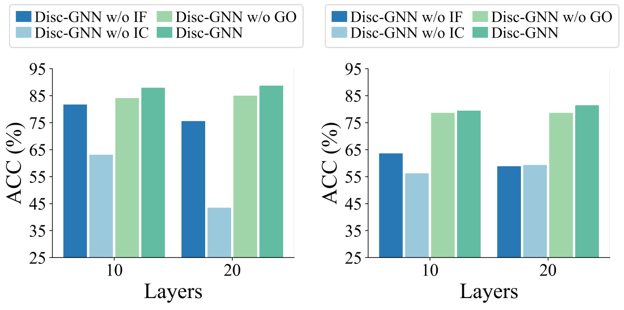
(b) Texas

Figure 3: Node classification accuracy (%) of GNN models with different depth on Cora and Texas datasets.

Depth Analysis

Compare to the classic model GCN, we further explore the performance of JK-Net, FAGCN and our proposed Disc-GNN under different model depth.

From Figure 3, it can be seen that on homophilic Cora dataset, GCN only achieves the best performance in the 2-nd layer. And on heterophilic Texas dataset, GCN aggregates more homophilic neighbors in higher order neighborhoods, alleviating the impact of lower order heterophilic neighbors and performing better in the 10-th layer. However, due to frequent interactions between dimensions and loss of initial information, GCN experiences a sharp decline in performance both on Cora and Texas datasets. JK-Net, FAGCN and our proposed Disc-GNN have alleviated the sharp decline in model performance to some extent. Among them, JK-Net introduces inter-layer skip connections to prevent the



(a) Cora

(b) Texas

Figure 4: Ablation study on Cora and Texas datasets.

loss of initial information, FAGCN decouples aggregations and transformations to reduce dimension interaction and information loss caused by multiple transformations, and our Disc-GNN employs inter-layer filtering and initial compensation to alleviate the effects of dimension interaction and information loss on node representations, respectively. Specifically, through inter-layer filtering, our Disc-GNN filters out node representations that have lost the ability to distinguish between different classes due to frequent feature interactions. In addition, through initial compensation, nodes retain their own features while aggregating neighbor information, alleviating the performance degradation caused by information loss.

Ablation Study

We further take a deeper look at Disc-GNN to understand how each component affects its performance. We choose 10-th and 20-th layers for ablation study on Cora and Texas datasets. Specifically, we create the following variants:

- Disc-GNN w/o IF: we remove the inter-layer filtering.
- Disc-GNN w/o IC: we remove the initial compensation.
- Disc-GNN w/o GO: we remove the global optimization.

As shown in Figure 4, Disc-GNN w/o IC has poor performance in both 10-th and 20-th layers. This is due to the fact that as the layers increase, heterophilic information introduced by multiple aggregations and repeated multiplication brought about by multiple transformations lead to a significant loss of initial information. The initial compensation in our Disc-GNN passes the initial information into the learning process of inter-layer representations to alleviate information loss. And our Disc-GNN can adaptively provide more initial compensation for node representations with severe information loss to stabilize the model performance in deep layers. At the same time, global optimization based on GDC can also improve the model performance to a certain extent. Moreover, we can find that inter-layer filtering plays a more important role on heterophilic Texas dataset than on homophilic Cora dataset. This is because frequent interactions between dimensions can exacerbate the impact of heterophilic information, leading to node-wise similar and dimension-wise indistinguishable on node representations. The inter-layer filtering in our Disc-GNN filters out node representations with low distinguishability of class, alleviating the impact of excessive interactions between dimensions on the representation learning.

Related Work

Most GNN models assume that neighbors are homophilic, and learn node representations by aggregating and transforming 1-hop neighbor information in a GNN layer. To obtain larger neighborhoods, stacking multiple GNN layers is usually used to learn node representations. However, classic GNN models, such as GCN (Kipf and Welling 2017), GAT (Velickovic et al. 2018), exhibit a sharp decline in performance at deeper layers. Many studies attribute performance degradation to the node-wise over-smoothing problem (Li, Han, and Wu 2018), where all node representations become too similar to distinguish each other. To alleviate this problem, GCNII (Chen et al. 2020b) and APPNP (Klicpera, Bojchevski, and Günnemann 2019) introduce initial information of nodes into inter-layer representation learning, aiming to maintain the uniqueness of each representation while aggregating other neighbors. DAGNN (Liu, Gao, and Ji 2020) and JK-Net (Xu et al. 2018) connect the representations of all intermediate layers to the output layer, preserving the information of each layer. In addition, GNNs also have the heterophily problem. Recent studies suggest that the over-smoothing and heterophily problems can be seen as two sides of the same coin (Yan et al. 2022), both of which are influenced by the interference of heterophilic neighbors. FAGCN (Bo et al. 2021) reduces the influence of heterophilic neighbors by controlling the aggregated weights of neighbors, while Fang et al. (Fang et al. 2022) imitates the mechanism of attitude polarization, and performs polarized aggregation on a hyper-sphere to cluster similar neighbors and separate dissimilar ones. Our work introduces dimension-wise LDC and GDC to guide the representation learning, generating node representations with high distinguishability of class. Each node representation can distinguish different classes and identify its own class, indirectly alleviating the problems of over-smoothing and heterophily.

Conclusion

In this work, we first propose two quantitative metrics LDC and GDC to indirectly analyze the changes in node representations from the perspective of distinguishability of class. Then, we design a novel graph neural network guided by distinguishability of class, which learns the inter-layer representation for each node under the supervision of LDC and globally optimizes node representations based on GDC, so that the learned representations are able to distinguish between different classes. Extensive experiments demonstrate the effectiveness of our proposed model in various node classification and clustering tasks.

Acknowledgments

This work is supported by National Key Research and Development Program of China (2023YFC3304503), National Natural Science Foundation of China (62276187, U22B2027, 62172297).

References

Bo, D.; Wang, X.; Shi, C.; and Shen, H. 2021. Beyond Low-frequency Information in Graph Convolutional Networks. In *Proceedings of the AAAI conference on artificial intelligence*, 3950–3957.

Chen, D.; Lin, Y.; Li, W.; Li, P.; Zhou, J.; and Sun, X. 2020a. Measuring and Relieving the Over-Smoothing Problem for Graph Neural Networks from the Topological View. In *Proceedings of the AAAI conference on artificial intelligence*, 3438–3445.

Chen, M.; Wei, Z.; Huang, Z.; Ding, B.; and Li, Y. 2020b. Simple and Deep Graph Convolutional Networks. In *Proceedings of the International Conference on Machine Learning*, 1725–1735.

Fang, Z.; Xu, L.; Song, G.; Long, Q.; and Zhang, Y. 2022. Polarized Graph Neural Networks. In *Proceedings of the ACM Web Conference*, 1404–1413.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural Message Passing for Quantum Chemistry. In *Proceedings of the International Conference on Machine Learning*, 1263–1272.

Hartigan, J. A.; and Wong, M. A. 1979. Algorithm As 136: A K-Means Clustering Algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1): 100–108.

Jin, D.; Ge, M.; Yang, L.; He, D.; Wang, L.; and Zhang, W. 2018. Integrative Network Embedding via Deep Joint Reconstruction. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 3407–3413.

Jin, D.; Wang, R.; Ge, M.; He, D.; Li, X.; Lin, W.; and Zhang, W. 2022a. RAW-GNN: RAndom Walk Aggregation based Graph Neural Network. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2108–2114.

Jin, D.; Wang, R.; Wang, T.; He, D.; Ding, W.; Huang, Y.; Wang, L.; and Pedrycz, W. 2023a. Amer: A New Attribute-Missing Network Embedding Approach. *IEEE Transactions on Cybernetics*, 53(7): 4306–4319.

Jin, D.; Yu, Z.; Huo, C.; Wang, R.; Wang, X.; He, D.; and Han, J. 2021. Universal Graph Convolutional Networks. In *Proceedings of the Advances in Neural Information Processing Systems*, 10654–10664.

Jin, D.; Yu, Z.; Jiao, P.; Pan, S.; He, D.; Wu, J.; Yu, P.; and Zhang, W. 2023b. A Survey of Community Detection Approaches: From Statistical Modeling to Deep Learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(2): 1149–1170.

Jin, W.; Liu, X.; Ma, Y.; Aggarwal, C. C.; and Tang, J. 2022b. Feature Overcorrelation in Deep Graph Neural Networks: A New Perspective. In *Proceedings of the Knowledge Discovery and Data Mining*, 709–719.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the International Conference on Learning Representations*.

Klicpera, J.; Bojchevski, A.; and Günnemann, S. 2019. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *Proceedings of the International Conference on Learning Representations*.

Li, G.; Muller, M.; Thabet, A.; and Ghanem, B. 2019. DeepGCNs: Can GCNs Go As Deep As CNNs? In *Proceedings of the IEEE/CVF international conference on computer vision*, 9267–9276.

Li, Q.; Han, Z.; and Wu, X.-M. 2018. Deeper Insights Into Graph Convolutional Networks for Semi-Supervised Learning. In *Proceedings of the AAAI conference on artificial intelligence*, 3538–3545.

Liu, M.; Gao, H.; and Ji, S. 2020. Towards Deeper Graph Neural Networks. In *Proceedings of the Knowledge Discovery and Data Mining*, 338–348.

Lu, M.; Han, Z.; Rao, S. X.; Zhang, Z.; Zhao, Y.; Shan, Y.; Raghunathan, R.; Zhang, C.; and Jiang, J. 2022. BRIGHT - Graph Neural Networks in Real-time Fraud Detection. In *Proceedings of the ACM International Conference on Information & Knowledge Management*, 3342–3351.

McCallum, A. K.; Nigam, K.; Rennie, J.; and Seymore, K. 2000. Automating the Construction of Internet Portals with Machine Learning. *Information Retrieval*, 3(2): 127–163.

Miao, X.; Zhang, W.; Shao, Y.; Cui, B.; Chen, L.; Zhang, C.; and Jiang, J. 2023. Lasagne: A Multi-Layer Graph Convolutional Network Framework via Node-Aware Deep Architecture. *IEEE Transactions on Knowledge and Data Engineering*, 35(2): 1721–1733.

Pei, H.; Wei, B.; Chang, K. C.-C.; Lei, Y.; and Yang, B. 2019. Geom-GCN: Geometric Graph Convolutional Networks. In *Proceedings of the International Conference on Learning Representations*.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective Classification in Network Data. *AI magazine*, 29(3): 93–93.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *Proceedings of the International Conference on Learning Representations*.

Wu, F.; Jr., A. H. S.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. Q. 2019. Simplifying Graph Convolutional Networks. In *Proceedings of the International Conference on Machine Learning*, 6861–6871.

Xu, K.; Li, C.; Tian, Y.; Sonobe, T.; Kawarabayashi, K.; and Jegelka, S. 2018. Representation Learning on Graphs with Jumping Knowledge Networks. In *Proceedings of the International Conference on Machine Learning*, 5449–5458.

Yan, Y.; Hashemi, M.; Swersky, K.; Yang, Y.; and Koutra, D. 2022. Two Sides of the Same Coin: Heterophily and Oversmoothing in Graph Convolutional Neural Networks. In *Proceedings of the International Conference on Data Mining*, 1287–1292.

Yang, F.; Fan, K.; Song, D.; and Lin, H. 2020. Graph-based Prediction of Protein-Protein Interactions with Attributed Signed Graph Embedding. *BMC bioinformatics*, 21(1): 1–16.

Yu, Z.; Jin, D.; Huo, C.; Wang, Z.; Liu, X.; Qi, H.; Wu, J.; and Wu, L. 2023. KGTrust: Evaluating Trustworthiness of SIoT via Knowledge Enhanced Graph Neural Networks. In *Proceedings of the ACM Web Conference*, 727–736.

Yu, Z.; Jin, D.; Liu, Z.; He, D.; Wang, X.; Tong, H.; and Han, J. 2021. AS-GCN: Adaptive Semantic Architecture of Graph Convolutional Networks for Text-Rich Networks. In *IEEE International Conference on Data Mining, ICDM*, 837–846.

Zhang, M.; Cui, Z.; Neumann, M.; and Chen, Y. 2018. An End-to-End Deep Learning Architecture for Graph Classification. In *Proceedings of the AAAI conference on artificial intelligence*, 4438–4445.