IEEE/CAA Journal of Automatica Sinica

Modularized Graph Convolutional Network

Tiantian He, Zhixuan Duan, and Xin Luo

Dear Editor,

This letter presents a novel graph neural network, namely Modularized graph convolution network (MGCN), to address the under-explored issue in graph convolution networks (GCNs), wherein the weights for neighbor aggregation are fixed, leading to the limited capability of capturing diverse relationships among nodes for representation learning. Conventional GCNs always learn node representations in the graph according to the weights computed from the graph Laplacian, consequently overlooking the similarity and group cohesiveness of node features. Motivated by this observation, the proposed MGCN is designed to learn expressive node representations by considering the following two aspects. First, we propose a novel Laplacian operator to generate weights for feature aggregation that can adaptively capture the feature similarity among nodes. Second, based on Modularity optimization [1], we design a novel auxiliary loss that enables the weights for aggregating features to capture the cluster structure of the graph. The proposed MGCN, therefore, can learn expressive representations for each node in the graph by leveraging the modularized messages, i.e., predominately aggregating the features of neighbors in the same cluster. The proposed MGCN has been tested with several well-established datasets and compared with strong GCN-based models. The experimental results demonstrate that MGCN is more effective in various downstream tasks.

Recently, graph neural networks (GNNs) have made significant strides in graph deep learning, demonstrating robust tools for tackling complex tasks in graph-structured data [2], [3]. As one of the major GNN workhorses, graph convolution networks (GCNs) have been widely used as the essential building blocks of learning representations for various downstream tasks [4], [5]. Typically, a graph convolution network learns representations for each node by aggregating neighboring features based on the weights that the Laplacian of the adjacency matrix generates. Such aggregation strategy has been widely used in GCN-based models, such as Graphsage [6], Deep Graph Infomax [7], and SGC [8]. Though effective, weights for neighbor aggregation in these GCN-based approaches are always fixed, lacking the comprehensive consideration of heterogeneous relations among nodes in the graph, which consequently hampers existing GCN-based models to learn accurate representations for the downstream task. In contrast to conventional GCNs, the proposed MGCN adopts a novel convolution operator to seamlessly generate weights for feature aggregation that capture group-wise similarity of node features. The proposed MGCN can holistically capture the cluster structure hidden in the graph, thus significantly enhancing the predictive performances in various downstream tasks (See Fig. 1).

Related work: To date, graph neural networks (GNNs) have garnered extensive interest. Most of them lie in the domain of message-passing GNNs, which accomplish the learning of node representations by performing a weighted sum of features from neighbors. For example, GCN [9] learns representations for each node in the graph based on the fixed weights derived from the Laplacian of the adjacency matrix. GraphSAGE [6] is proposed to learn representations in the graph by aggregating the features from randomly sampled neighbors. APPNP [10] is another GCN-based model that further enhances message-passing by incorporating the personalized PageRank [11]. SGC [8] further simplifies the multi-step PageRank as a higher-order transition matrix to speed up the process of representation learning. UGNN [12] leverages an additional denoising loss to discriminate the representations learned by GCNs. Different from these mentioned GCN-based approaches, graph attention networks (GATs) [13] introduce the attention mechanisms for the aggregation of neighbor features, enabling the weights among the neighbors to be adaptive according to the node features. CATs [14] and DCAT [15] improve the performance of attention-based GNNs by introducing structure-aware attention mechanisms for representation learning.

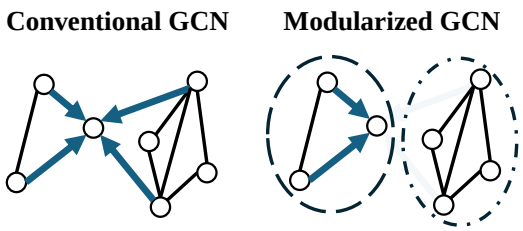


Fig. 1. Comparisons between GCNs (left) and MGCN (right). MGCN aggregates neighbors within the same cluster using adaptive, similarity-based weights, while GCNs sum all connected neighbors with fixed weights.

Modularity optimization [1] aims to maximize the probability difference between edges falling within the same cluster and edges randomly distributed in the graph, which is well-known for tackling unsupervised graph learning tasks, e.g., graph clustering [16]. Having investigated prevalent GNNs, we observe that either GCN-based approaches, learning representations by aggregating neighbor features according to the fixed weights, or GAT-based ones, learning representations according to adaptive weights (attention coefficients), do not explicitly capture cluster information during the learning process. Motivated by these observations, in this letter, we hypothesize that leveraging clustering information to generate adaptive weights for aggregating neighbor features can enhance GCNs in semi-supervised learning tasks. To this end, we present MGCN, which utilizes Modularity optimization to construct a new auxiliary loss to seamlessly acquire the group cohesiveness for generating weights for aggregating neighbor features, thus learning expressive representations.

Notation: This letter focuses on learning node representations in the undirected graph. An undirected graph $G = (V, E)$, with V containing containing n nodes in C classes, and E containing m edges, is represented as an adjacency matrix $\mathbf{A} \in \{0, 1\}^{n \times n}$, wherein $a_{ij} \in \mathbf{A}$ describes the relationship between nodes i and j. $a_{ij} = 1$ if $e_{ij} \in E$, and $a_{ij} = 0$ otherwise. The feature matrix $\mathbf{H} \in \mathbb{R}^{n \times f}$ contains f-dimensional row feature vectors $\mathbf{h}_i$ for each node i.

Problem statement: Homophily theory has shown that nodes with similar features often form closer connections [17], e.g., social players connected by shared interests, thus emerging some cohesive properties, e.g., cluster structures in the graph that can be identified via optimizing Modularity. Although significant, the benefit of these properties for representation learning with GCNs has yet to be explored. To address this, we propose MGCN. Specifically, for each layer of MGCN, we derive a new Laplacian operator, wherein the adaptive weights for neighbor aggregation are computed based on the feature correlations between neighbors. To enable previously computed weights to capture the clustering information, we leverage Modularity to derive an auxiliary loss for each MGCN layer, where whether a pair of nodes belong to the same cluster is determined by

Zhixuan Duan and Xin Luo are with the College of Computer and Information Science, Southwest University, Chongqing 400715, China (e-mail: qqww123456@email.swu.edu.cn, luoxin21@gmail.com).

Tiantian He is with the Center for Frontier AI Research, Institute of High Performance Computing, Agency for Science, Technology and Research (A*STAR), 699010, Singapore (e-mail: he_tiantian@cfar.a-star.edu.sg).

Tiantian He and Zhixuan Duan contribute equally to this work. Corresponding author: Xin Luo.

their feature correlation. By optimizing the auxiliary loss regarding the node features, MGCN can learn the hidden cluster structure (i.e., regularizing the features of nodes in the same cluster to be more correlated), thus guiding representation learning with strongly cohesive neighbors. We will present the details of MGCN next.

For a layer in MGCN, let $\delta(\mathbf{h}_i, \mathbf{h}_j)$ be a function computing the feature similarity between node i and j. Then, the weights for aggregating features from neighbors are defined as follows:

$$\begin{aligned}\delta(\mathbf{h}_i, \mathbf{h}_j) &= \frac{1}{1+\exp\{-(\mathbf{h}_i \cdot (\boldsymbol{\alpha} \odot \mathbf{h}_j)^T)\}},\\ \hat{a}_{ij} &= \frac{a_{ij} \cdot \delta(\mathbf{h}_i, \mathbf{h}_j)}{\sqrt{\sum_{k\in\mathcal{N}_i}\delta(\mathbf{h}_i, \mathbf{h}_k)\sum_{k\in\mathcal{N}_j}\delta(\mathbf{h}_j, \mathbf{h}_k)}}, \hat{a}_{ij} \in \hat{\mathbf{A}},\end{aligned} \tag{1}$$

where $\boldsymbol{\alpha}$ is a vector of learnable parameters, $\odot$ denotes element-wise multiplication, and $\mathcal{N}_i$ represents the neighbor set of node i. Unlike GCNs that solely use adjacency matrix to derive the fixed weights among neighbors, Eq. (1) introduces the feature similarity to dynamically re-weight the edges in the graph, thus combining node features and graph structure. Neighbors with similar features and local structures might play more important roles in the subsequent learning of node representations. With $\hat{\mathbf{A}}$, the proposed MGCN can generate representations for each node as follows:

$$\mathbf{H} = \mathbf{H}^{in}\mathbf{W}, \mathbf{h}_i^{out} = \sigma(\sum_{j\in\mathcal{N}_i}\hat{a}_{ij}\mathbf{h}_j), \tag{2}$$

where $\sigma(\cdot)$ represents the activation function, $\mathbf{W}$ is a matrix for feature projection, and $\mathbf{H}^{in}$ is the input feature matrix of some layer in MGCN. From Eq. (1), we know that $\delta(\mathbf{h}_i, \mathbf{h}_j)$ can only capture the feature similarity of neighbors. To endow the proposed MGCN with the capability of acquiring the group properties carried by the graph, we propose an auxiliary clustering loss for each layer of MGCN. Specifically, the proposed auxiliary loss enables MGCN to optimize the Modularity [1], [16] pertaining to node features at each layer. Following the definition of Modularity, we consider the inner product of features of two nodes, i.e., $\mathbf{h}_i\mathbf{h}_j^T$, to represent whether they belong to the same cluster. Thus, we have the following:

$$\max Q = \frac{1}{2m}\sum_{i,j}\left(a_{ij} - \frac{d_i d_j}{2m}\right)\mathbf{h}_i\mathbf{h}_j^T, \text{ s.t. } d_i\mathbf{h}_i\mathbf{h}_j^T = c, \forall i, \tag{3}$$

where d_i is the degree of node i, m is the number of edges, and c is some bounded constant. Optimizing Modularity (Q) regarding $\mathbf{h}_i$ (Eq. (3)) enables MGCN to generate higher weights for nodes in the same cluster (Eq. (1)) and learn representations by mainly aggregating node features in the same cluster. However, as there are n^2 elements involved in the backpropagation, directly optimizing Eq. (3) is computationally inefficient. Here, we derive an equivalent and efficient form of Eq. (3). Let $1-\beta_i$ be the Lagrangian multiplier of Eq. (3), and we have the following Lagrangian function:

$$L = \sum_{i,j} b_{ij}\mathbf{h}_i\mathbf{h}_j^T - \sum_i(1-\beta_i)(d_i\mathbf{h}_i\mathbf{h}_i^T - c), \tag{4}$$

where $b_{ij} = a_{ij} - \frac{d_i d_j}{2m}$, and we omit $\frac{1}{2m}$ that does not affect the solution. Based on KKT conditions, we derive that the partial derivative regarding $\mathbf{h}_i$ equals zero:

$$\frac{\partial L}{\partial \mathbf{h}_i} = \sum_j b_{ij}\mathbf{h}_j - (1-\beta_i)d_i\mathbf{h}_i = 0. \tag{5}$$

We can obtain the optimal $\mathbf{h}_i$ by solving Eq. (5). By summing over all $\mathbf{h}_i$s, Eq. (5) can be simplified as the following:

$$\sum_{i,j} b_{ij}\mathbf{h}_j = \sum_i(1-\beta_i)d_i\mathbf{h}_i. \tag{6}$$

TABLE I
CHARACTERISTICS OF TEST DATASETS.

	Cora	Citeseer	Pubmed	CoauthorCS	Flickr	BBCsport
n/m	2708/5429	3327/4732	19717/44338	18333/327576	89250/899756	737/17242
f/C	1433/7	3703/6	500/3	6805/15	500/7	966/5
Q	0.807	0.886	0.727	0.665	0.465	0.596

Note that $\sum_{i,j} b_{ij}\mathbf{h}_j = \sum_{i,j}(a_{ij} - \frac{d_i d_j}{2m})\mathbf{h}_j$. Thus, we have:

$$\sum_{i,j} b_{ij}\mathbf{h}_j = \sum_j \mathbf{h}_j(\sum_i a_{ij}) - \sum_j d_j\mathbf{h}_j(\sum_i \frac{d_i}{2m}). \tag{7}$$

Since $\sum_i a_{ij} = d_j$ and $\sum_i \frac{d_i}{2m} = 1$, we have $\sum_i(1-\beta_i)d_i\mathbf{h}_i = 0$. The solution to this equation is $d_i\mathbf{h}_i = 0$ as Lagrangian multipliers (β_is) can not be all zeros. Thus, solving Eq. (5) is equivalent to solving the following simplified equation:

$$\sum_j a_{ij}\mathbf{h}_j = (1-\beta_i)d_i\mathbf{h}_i \tag{8}$$

Let $\mathbf{x}_i = \sqrt{d_i}\mathbf{h}_i$, we have:

$$\sum_j a_{ij}\frac{\mathbf{x}_j}{\sqrt{d_j}} = (1-\beta_i)\sqrt{d_i}\mathbf{x}_i \Rightarrow \mathbf{L}\mathbf{x}_i = \beta_i\mathbf{x}_i, \forall i, \tag{9}$$

where $\mathbf{L}$ is the normalized Laplacian matrix of $\mathbf{A}$. Eq. (9) is an eigenvalue decomposition problem. Its solution is given by identifying the first k smallest eigenvectors of $\mathbf{L}$. Alternatively, Eq. (9) can be solved in the following form:

$$\min tr(\mathbf{X}^T\mathbf{L}\mathbf{X}), \text{subject to } \mathbf{X} = \mathbf{D}\mathbf{H}, \mathbf{X}^T\mathbf{X} = \mathbf{I}, \tag{10}$$

where $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii} = \sqrt{d_i}$ and $\mathbf{I}$ is the identity matrix. Here, we do not force each node to belong to only one cluster. Thus, the Modularity optimization problem (Eq. (3)) at l layer can finally be simplified to the following auxiliary clustering loss:

$$\mathcal{L}_Q^l = \frac{1}{n}tr(\mathbf{X}^T\mathbf{L}\mathbf{X}), \text{subject to } \mathbf{X} = \mathbf{D}\mathbf{H}, \tag{11}$$

where l represents layer l in MGCN. Based on the derivations above, we show that Modularity optimization (Eq. (3)) is approximately equivalent to efficiently solve Eq. (9), where $\mathbf{L}$ is very sparse. Now, we are able to formulate the loss function of the proposed MGCN:

$$\mathcal{L} = \lambda\sum_l \mathcal{L}_Q^l + \mathcal{L}_T, \tag{12}$$

where λ is a hyperparameter controlling the significance of the clustering loss, and $\mathcal{L}_T$ is the task-specific loss. In this paper, $\mathcal{L}_T$ is the binary cross-entropy loss that is widely adopted by GNNs for semi-supervised learning. Previous studies have shown considering adaptive graph Laplacian [18] or implicit group structure [19] can enhance the learning performance of GCNs. With Eq. (12), MGCN can adaptively learn $\delta(\mathbf{h}_i, \mathbf{h}_j)$ that can optimize the Modularity, thus constructing the Laplacian operator to explicitly capture the clustering information for learning expressive representations.

Datasets, baselines, and experimental settings: In this letter, we mainly consider semi-supervised node classification and clustering tasks to assess the efficacy of the proposed MGCN. We select six widely recognized graph datasets, including Cora, Citeseer, Pubmed, CoauthorCS, BBCsport, and Flickr (Table I). For classification tasks in Cora, Citeseer, and Pubmed datasets, we adopt the training, validation, and test split established in previous works [9]. For CoauthorCS, Flickr, and BBCsport datasets, the random split of training-validation-testing sets is conducted with the ratio of 4:3:3 (CoauthorCS) and 6:2:2 (Flickr and BBCsport). As for clustering tasks on all test datasets, we adopt the same split of training and validation as classification tasks and use all nodes for the test.

We select eight strong GCN-based approaches as baselines, including GCN [9], GCNII [20], DGI [7], GraphSage [6], JKNet

TABLE II
ACCURACY (%) AND F1-SCORE (%) ON NODE CLASSIFICATION TASKS.

Method	Cora		Citeseer		Pubmed		CoauthorCS		BBCsport		Flickr	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
DGI	80.76 ± 0.5	79.32 ± 0.6	69.26 ± 0.7	59.52 ± 0.7	74.56 ± 0.7	70.44 ± 1.2	88.30 ± 0.3	80.72 ± 0.4	91.72 ± 0.6	92.22 ± 1.2	51.17 ± 0.3	20.20 ± 0.2
GCN	80.56 ± 0.8	78.54 ± 0.8	71.20 ± 0.8	62.60 ± 0.9	79.78 ± 0.3	77.58 ± 0.4	88.51 ± 0.2	84.89 ± 1.0	93.98 ± 1.2	94.20 ± 1.2	50.71 ± 1.4	18.35 ± 1.8
GCNII	82.45 ± 0.6	81.57 ± 0.2	71.43 ± 0.8	64.60 ± 0.7	76.80 ± 0.5	74.25 ± 1.3	87.98 ± 0.3	69.20 ± 0.7	94.92 ± 0.8	95.55 ± 0.4	46.90 ± 0.4	14.50 ± 0.4
GraphSage	81.94 ± 0.4	81.94 ± 0.4	71.70 ± 0.5	64.60 ± 0.6	80.96 ± 0.4	79.53 ± 0.5	87.85 ± 0.3	82.64 ± 1.0	96.11 ± 1.3	95.94 ± 1.0	43.50 ± 1.9	8.87 ± 0.9
APPNP	82.56 ± 0.5	82.56 ± 0.5	71.36 ± 1.0	64.55 ± 1.3	81.04 ± 0.4	79.44 ± 0.5	89.47 ± 1.5	86.12 ± 1.0	96.49 ± 1.4	96.65 ± 1.7	48.01 ± 2.1	13.37 ± 3.7
SGC	79.95 ± 1.4	78.84 ± 1.2	69.38 ± 2.1	62.68 ± 2.1	80.70 ± 2.1	78.88 ± 3.2	83.43 ± 2.8	80.13 ± 1.5	96.62 ± 0.9	96.69 ± 0.9	48.50 ± 0.6	19.87 ± 1.0
JKnet	78.52 ± 1.0	75.36 ± 1.2	66.76 ± 1.0	61.47 ± 1.0	78.90 ± 0.5	75.98 ± 1.5	86.38 ± 2.4	79.28 ± 1.1	95.81 ± 1.3	95.32 ± 1.2	48.03 ± 0.1	14.10 ± 0.1
UGNN	82.64 ± 0.6	82.22 ± 0.8	71.06 ± 0.6	62.37 ± 1.3	82.11 ± 0.6	81.59 ± 0.5	89.61 ± 0.1	74.33 ± 0.4	96.60 ± 0.7	97.00 ± 0.7	50.89 ± 2.3	19.74 ± 4.7
MGCN	**83.54 ± 0.2**	**82.97 ± 0.2**	**72.97 ± 0.4**	**66.93 ± 0.3**	**84.42 ± 0.1**	**82.25 ± 0.3**	**93.63 ± 0.1**	**91.94 ± 0.2**	**97.84 ± 0.3**	**98.60 ± 0.3**	**52.15 ± 1.4**	**24.85 ± 3.0**

TABLE III
ACCURACY (%) AND F1-SCORE (%) ON NODE CLUSTERING TASKS.

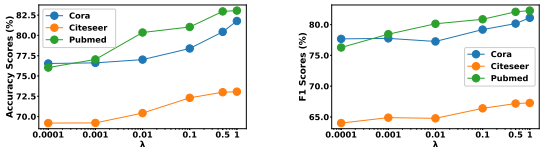
Method	Cora		Citeseer		Pubmed		CoauthorCS		BBCsport		Flickr	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
DGI	80.97 ± 0.2	79.90 ± 0.4	68.88 ± 0.3	65.45 ± 0.2	78.41 ± 0.8	78.36 ± 0.7	86.93 ± 0.3	75.01 ± 1.2	92.30 ± 0.3	92.79 ± 0.2	51.76 ± 0.5	20.36 ± 0.8
GCN	74.70 ± 0.7	69.76 ± 1.4	65.04 ± 1.3	58.96 ± 1.3	74.38 ± 2.2	73.83 ± 2.2	93.90 ± 0.2	92.04 ± 0.3	97.09 ± 0.2	98.17 ± 0.2	50.61 ± 0.5	16.95 ± 0.2
GCNII	75.50 ± 0.4	68.10 ± 0.7	69.23 ± 0.3	61.02 ± 0.3	77.56 ± 0.7	76.60 ± 1.1	87.70 ± 0.3	70.02 ± 0.2	97.55 ± 0.4	97.88 ± 0.2	47.08 ± 0.7	15.30 ± 0.5
GraphSage	79.12 ± 1.7	78.19 ± 1.7	67.29 ± 1.0	65.01 ± 0.6	75.96 ± 0.6	76.02 ± 0.5	89.02 ± 0.2	73.33 ± 1.3	95.93 ± 0.8	96.10 ± 0.7	47.77 ± 1.6	8.49 ± 1.1
APPNP	77.94 ± 2.7	77.53 ± 2.2	63.48 ± 1.9	61.70 ± 1.6	74.35 ± 3.2	74.31 ± 3.0	93.52 ± 0.2	91.32 ± 0.4	98.10 ± 0.6	98.03 ± 0.5	49.48 ± 0.2	18.95 ± 0.2
SGC	80.24 ± 2.1	78.27 ± 3.3	68.04 ± 0.9	62.22 ± 0.8	75.26 ± 1.6	75.32 ± 1.2	92.47 ± 0.1	89.61 ± 0.2	95.79 ± 0.1	95.96 ± 0.1	43.62 ± 0.3	16.67 ± 1.3
JKnet	77.05 ± 1.1	77.37 ± 0.9	64.47 ± 0.9	58.96 ± 2.7	76.40 ± 0.5	75.98 ± 0.7	94.54 ± 0.2	91.50 ± 0.1	97.31 ± 1.3	98.07 ± 0.7	46.78 ± 0.2	14.49 ± 0.7
UGNN	81.75 ± 0.5	80.60 ± 1.4	69.16 ± 1.2	64.13 ± 0.1	78.07 ± 1.0	77.48 ± 0.9	94.43 ± 0.1	**93.07 ± 0.2**	97.74 ± 0.5	97.79 ± 0.5	52.96 ± 1.6	19.83 ± 2.0
MGCN	**82.95 ± 0.2**	**82.78 ± 0.3**	**70.94 ± 0.1**	**65.56 ± 0.2**	**82.93 ± 0.1**	**82.34 ± 0.1**	**94.74 ± 0.2**	92.58 ± 1.7	**98.64 ± 0.1**	**98.67 ± 0.1**	**54.04 ± 3.7**	**23.95 ± 2.3**

[11], APPNP [10], SGC [8], and UGNN [12]. We use the source codes released by the authors with their recommended settings in the experiment. For fair comparisons, we configure the proposed MGCN according to other GCN baselines. Specifically, MGCN leverages a two-layer network structure. The dimension of the hidden layer, dropout ratio, and λ are set to 32, 0.6, and 1.0. We use the Adam optimizer to train MGCN, and the learning rate and weight decay are set to 0.01 and 0.0005. All experiments are conducted on a workstation with NVIDIA GeForce RTX2080Ti 11GB. All GNNs are run ten times on each test dataset to obtain the average performance.

Results and analysis: Semi-supervised node classification and clustering are closely related to several important downstream tasks, such as document classification, social community detection, and collaboration analysis. In this letter, we let all GNNs conduct these two learning tasks on the previously mentioned datasets. The performance of each GNN is evaluated by two different metrics, i.e., $Accuracy$ and $F1-score$. The comparative performances have been summarized in Tables II and III, wherein the best and second-best performances are highlighted in bold and underlined. We observe from both tables that MGCN significantly outperforms all the baselines built upon GCN, except for the clustering task on CoauthorCS evaluated by $F1-score$, where MGCN obtains the second-best performance. For example, the performance gains of MGCN on CoauthorCS are %4.49 (Acc) and %6.76 ($F1-score$) when compared to the second-best baseline in classification tasks. On the Flickr dataset, MGCN is better than the second-best baseline (DGI) by %23.02 and %17.63 when classification and clustering performances are evaluated by $F1-score$. The remarkable results indicate the proposed auxiliary clustering loss leads to better performances in those graphs with high Modularity (see Table I). By regularizing the neighbor aggregation with the proposed auxiliary clustering loss, MGCN can mainly consider neighbors within the same clusters for generating output representations, thus exhibiting significant performance improvement compared to other GCN-based approaches.

To reveal how λ influences the performance of MGCN, we conduct sensitivity tests on the datasets with fixed data splits, including Cora, Citeseer, and Pubmed. Specifically, λ is set to $\{0.0001, 0.001, 0.01, 0.1, 0.5, 1\}$ when MGCN conducts semi-supervised classification tasks. The performances evaluated by Acc and $F1-score$ have been plotted in Fig. 2. As the figure shows, MGCN performs better as λ becomes larger within the interval $(0, 1]$. In general, λ can be set to a higher value (e.g., 1.0 in our experiments) if the Modularity of the graph is high and vice versa.

Conclusion: This letter has presented a novel graph neural network, namely Modularized graph convolution network (MGCN). In contrast to existing GCNs, MGCN adopts a novel strategy to capture the structure-induced feature similarity among neighbors, which is then optimized by a clustering loss to acquire the group-level properties in the graph. With the convolution weights derived from the previously



(a) Acc (b) $F1-score$

Fig. 2. Sensitivity test on λ.

mentioned similarity, the proposed MGCN can adaptively aggregate the features of neighbors within the same clusters to generate node representations. Empirical studies conducted on several real graph datasets have demonstrated the superiority of MGCN in semi-supervised learning tasks against strong GCN-based approaches. In the future, we will further improve the proposed MGCN by designing novel loss functions to enhance its generalizability and developing new network architectures to adapt MGCN to diverse learning tasks. Besides, we will concentrate on designing new strategies that can better handle high-dimensional features carried by the graph data.

REFERENCES

[1] M. E. Newman, "Modularity and community structure in networks," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 103, no. 23, pp. 8577–8582, 2006.

[2] X. Hong, T. Zhang, Z. Cui, and J. Yang, "Variational gridded graph convolution network for node classification," *IEEE/CAA J. Autom. Sin.*, vol. 8, no. 10, pp. 1697–1708, 2021.

[3] Q. Zhu, Q. Xiong, Z. Yang, and Y. Yu, "Rgcnu: Recurrent graph convolutional network with uncertainty estimation for remaining useful life prediction," *IEEE/CAA J. Autom. Sin.*, vol. 10, no. 7, pp. 1640–1642, 2023.

[4] F. Bi, T. He, and X. Luo, "A two-stream light graph convolution network-based latent factor model for accurate cloud service qos estimation," in *ICDM*, 2022, pp. 855–860.

[5] T. He, Y. Liu, Y. S. Ong, X. Wu, and X. Luo, "Polarized message-passing in graph neural networks," *Artificial Intelligence*, vol. 331, p. 104129, 2024.

[6] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *NeurIPS*, vol. 30, 2017.

[7] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," in *ICLR*, 2019.

[8] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *ICML*, 2019, pp. 6861–6871.

[9] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2016.

[10] J. Gasteiger, A. Bojchevski, and S. Günnemann, "Predict then propagate: Graph neural networks meet personalized pagerank," in *ICLR*, 2019.

[11] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *ICML*, 2018, pp. 5453–5462.

[12] Y. Ma, X. Liu, T. Zhao, Y. Liu, J. Tang, and N. Shah, "A unified view on graph neural networks as graph signal denoising," in *Proc. ACM Int. Conf. Inf. Knowl. Manag.*, 2021, pp. 1202–1211.

[13] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.

[14] T. He, Y. S. Ong, and L. Bai, "Learning conjoint attentions for graph neural nets," *NeurIPS*, vol. 34, pp. 2641–2653, 2021.

[15] H. Zhou, T. He, Y. S. Ong, G. Cong, and Q. Chen, "Differentiable clustering for graph attention," *IEEE Trans. Knowl. Data Eng.*, 2024.

[16] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E*, vol. 70, no. 6, p. 066111, 2004.

[17] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annu. Rev. Sociol.*, vol. 27, no. 1, pp. 415–444, 2001.

[18] K. Huang, Y. G. Wang, M. Li, and P. Lio, "How universal polynomial bases enhance spectral graph neural networks: Heterophily, over-smoothing, and over-squashing," in *ICML*, 2024.

[19] C. Huang, M. Li, F. Cao, H. Fujita, Z. Li, and X. Wu, "Are graph convolutional networks with random weights feasible?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 2751–2768, 2023.

[20] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *ICML*, 2020, pp. 1725–1735.