Pattern Recognition 112 (2021) 107788

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

MobileGCN applied to low-dimensional node feature learning

Wei Dong[a], Junsheng Wu[a,b,*], Zongwen Bai[a,c,d], Yaoqi Hu[a], Weigang Li[b], Wei Qiao[a,f], Marcin Woźniak[e]

[a] School of Computer Science and Engineering, Northwestern Polytechnical University, China
[b] School of Software, Northwestern Polytechnical University, China
[c] Shaanxi Key Laboratory of Intelligent Processing of Big Energy Data, Yanan University, China
[d] School of Physics and Electronic Information, Yanan University, China
[e] Faculty of Applied Mathematics, Silesian University of Technology, Poland
[f] Xi'an Research Institute of China Coal Technology & Engineering Group Corp, China

ARTICLE INFO

Article history:
Received 23 January 2020
Revised 16 September 2020
Accepted 7 December 2020
Available online 16 December 2020

Keywords:
Graph convolutional networks
Affinity-aware encoding
Updater
Depth-wise separable graph convolution
Low-Dimensional node features

ABSTRACT

The idea of the paper concentrates on an iterative learning process in Graph Convolution Networks (GCNs) involved in two vital steps: one is a message propagation (message passing) step to aggregate neighboring node features via aggregators performed, and another is an encoding output step to encode node feature representations by using updaters. In our model, we propose a novel affinity-aware encoding as an updater in GCNs, which aggregates the neighboring nodes of a node while updating this node's features. By utilizing affinity values of our encoding, we order the neighboring nodes to determine the correspondence between encoding functions and the neighboring nodes. Furthermore, to explicitly reduce the model size, we propose a lightweight variant of our updater that integrates Depth-wise Separable Convolution (DSC) into it, namely Depth-wise Separable Graph Convolution (DSGC). Comprehensive experiments conducted on graph data demonstrate that our models' accuracy improved significantly for graphs of low-dimensional node features. Also, performed in the low-dimensional node feature space we provide state-of-the-art results on two metrics (Macro-f1 and Matthews correlation coefficient (MCC)). Besides, our models are robust when taking different low-dimensional feature selection strategies.

1. Introduction

Graphs are fundamental models used in scientific approaches to describe the relation between objects in the real world. Using graph models to study information propagation [1] or to represent molecular structures [2] is recently an essential research approach. Graph models leveraging deep learning [3] have been receiving more and more attention due to their great expressive ability of graphs across various areas, including social networks [1], physical systems [4], protein-protein interaction networks [2], and knowledge graphs [5]. Deep learning has yielded fruitful results in many areas of machine intelligence, enabling high accuracy in complex tasks of computer vision [6], speech recognition [7], and natural language processing [8]. Among such methods of combining graph with deep learning, Graph Convolution Networks (GCNs) rooted in Convolutional Neural Networks (CNNs) [3] are prevalent. They have also been divided into multiple research orientations according to the type of message passing, such as spectral [9], spatial [1], attention [10], and gate mechanism [11] algorithms.

The essential idea of GCNs is to propagate information between graph nodes that the node features are learned with consideration of the neighboring context [12]. A standard, iterative GCNs' pipeline involves two indispensable steps [13], namely, the message propagation step (message passing) and encoding output step (node feature encoding). The former leverages an aggregator to aggregate each node's features of its neighbors, where this aggregator could be any of a predefined eigendecomposition [9], a graph normalized Laplacian [14], a linear or nonlinear transformation [15], and an attention [10] or gate [11] mechanism. The latter encodes each node's hidden states (representations) independently by using an updater, which is a weight matrix to shrink node feature dimensions uniformly without any aggregating operation for the neighborhood. Briefly, aggregators gather graph-structured information across the whole graph, and updaters learn the information of each node itself. The graph-structured and node-featured information determine the model performance together. However, the accuracy of GCNs is deteriorative when nodes are represented by low-dimensional features (a graph contains only a small num-

* Corresponding author.
E-mail addresses: dw156@mail.nwpu.edu.cn (W. Dong), wujunsheng@nwpu.edu.cn (J. Wu).

https://doi.org/10.1016/j.patcog.2020.107788
0031-3203/

ber of node features). The reason for this deterioration is that low-dimensional features result in the node-featured information learned insufficiently by updaters. Since there is a lack of sufficient node features for encoding, updaters have to learn graph-structured information thoroughly to make up incomplete node-featured information learned for improving the accuracy. Unfortunately, current methods [13] focus on to redesign aggregators to gather graph-structured information, while ignoring updaters can also be used to learn it.

To solve this issue, we propose a novel affinity-aware encoding as an updater, called Local Graph Convolution (LGC). In the encoding output step, compared versus the updaters of previous methods, LGC can not only encode nodes' hidden states to learn node-featured information but also assist aggregators to gather the features of neighbors to capture graph-structured information. This updated process also learns each node's representation jointly via embedding an aggregated architecture, i.e., weight sharing [3]. By introducing the weight sharing architecture from a standard convolution in CNNs [3] to our LGC, its receptive fields of weight sharing are the neighbors of each node, and affinity values order these neighboring nodes. Note that the affinity value can be node ID, uniformly sampling [1], or graph centrality [16] to sample a fixed-size neighboring set. LGC updater can also combine with any aggregators. In our approach, we choose the aggregator of the landmark GCN [14] as a baseline aggregator to collaborate with LGC to build the LGC-GCN model. Moreover, selecting two personalized PageRank (PPR) [17] aggregators, i.e., Personalized Propagation of Neural Predictions (PPNP) [15] and Approximate Personalized Propagation of Neural Predictions (APPNP) [15], are combined with our updater to form LGC-PPNP and LGC-APPNP respectively. Klicpera et al. [15] first designed these two PPR aggregators to learn more graph-structured information via using the personalized PageRank algorithm [17] to provide a desirably larger neighborhood for nodes in the periphery or a sparsely labeled setting. In contrast, other advanced aggregators such as spectral [14] and non-spectral [1] methods only use the information of a minimal neighborhood for each node. The unique capability of these two PPR aggregators is that they can adjust the learning proportion between graph-structured and node-featured information via the teleport (or restart) probability [17]. By lowering the teleport probability, the PPR aggregators gather more graph-structured information to improve model accuracy when nodes are represented by low-dimensional feature space.

Furthermore, to explicitly shrink the model size, we propose a lightweight variant of LGC named as Depth-wise Separable Graph Convolution (DSGC) derived from Depth-wise Separable Convolution (DSC). Sifre and Mallat [18] initially introduced DSC, and then it was used in inception models [19] to reduce the computation in the first few layers of the model. It is also used in MobileNet [20] and MobileNetV2 [21] to tailor the models on mobiles and embedded devices. DSGC also associates GCN, PPNP, and APPNP aggregators to set up MobileGCNs (i.e., MobileGCN, MobilePPNP, and MobileAPPNP). Experiments demonstrate that our models significantly promote the accuracy and provide state-of-the-art results on the other two metrics (Macro-f1 and Matthews Correlation Coefficient(MCC)) compared to benchmark models for graphs involved by low-dimensional node features. Additionally, the experiments of low-dimensional node feature selection show that the metrics of our models is robust when taking different feature selection strategies. The main novelty contributions of our research can be four-fold named.

Modularization. Our LGC and DSGC, as the independent updater modules, can integrate with any of the other aggregators. In this article, we combine our updaters with aggregators of GCN [14], PPNP, and APPNP [15]. In future work, we will design more models, such as attention models [10], to embed our updaters to enhance their robustness in the low-dimensional node feature space.

Expansibility. Extending from a standard convolution of CNNs to LGC implies that the advancement of LGC can take advantage of the achievement of the standard convolution improved. For example, DSC is a lightweight convolution extended by the standard convolution, and, hence, the DSGC is a lightweight version of LGC. Both of LGC and DSGC derived from the standard convolution and its improved version.

Reciprocity. Compared to other GCN updaters [13], the function of LGC and DSGC has cut through the boundary between aggregators and updaters because they have the reciprocity in order to learn the graph-structured information thoroughly across the iterative process involved the message propagation and encoding output steps.

Robustness. By the comparison of different strategies of feature selection for training benchmark GCNs and our models, three metrics (Accuracy, Macro-f1, and MCC) are compared to show robustness of our proposed models in the low-dimensional feature space.

The rest of this paper is organized as follows: we describe the development of GCNs and introduce PPNP and APPNP in Section 2. Based on these, we build an affinity-aware updater, LGC, in Section 3. Further, DSGC is described in Section 4. Section 5 describes experimental setting. Section 6 presents the results and discussion. Section 7 concludes our work and sets the horizon of our future researches.

2. Related works

Several excellent studies have given an impetus to improve aggregators and updaters continuously, and a review was given by Zhou et al. [13]. Bruna et al. [22] pioneered to propose the spectral method of GCN. Their aggregator defined in the Fourier domain by performing the eigendecomposition of the graph Laplacian. Furthermore, the updater is part of this aggregator to parameterize the diagonal matrix in the eigendecomposition. However, the updater results in potentially intense computations and non-spatially localized filters. Henaff et al. [23] hence tried to design the spectral filters spatially localized by introducing a parameterized updater with smooth coefficients. Afterward, ChebNet [9] designed a new K-localized updater which could remove the need to compute the eigenvectors of the Laplacian and be independent of the aggregator. Finally, for alleviating overfitting on local neighborhood structures when graphs involve vast node degree distributions, Kipf et al. [14] proposed the GCN model that limits the layer-wise K-localized updater to $K = 1$. Recently, Liao et al. [24] proposed a novel deep GCN for a multi-scale graph. Wu et al. [25] simplified GCNs by removing nonlinearities and collapsing weight matrices between consecutive layers. Abu-El-Haija et al. [26] designed higher-order message passing architectures by sparsified neighborhood mixing. Klicpera et al. [15] discussed an application of personalized PageRank to improve the normalization of graph Laplacian and to achieve better performance. The PPR aggregators could learn a larger neighboring scale of graph-structured information compared to other spectral GCNs and hence be used suitably to our models performed on the low-dimensional node feature space.

In the non-spectral GCNs, aggregators operated on spatially close neighbors defined convolutions directly on the graph [13]. They have been used to solve a limitation of the aggregators of spectral approaches, that is, a spectral model trained on the Laplacian eigenbasis defined by the specific graph structure, and could not be directly applied to a graph with a different structure, even though, compared to spatial GCNs, the spectral model has the outstanding performance on some datasets. Duvenaud et al. [27] proposed the Neural FPs model to use different weight matrices for nodes with different degrees, but it cannot be applied to large-

scale graphs with more node degrees. Atwood et al. [28] proposed the diffusion-convolutional neural networks (DCNN) to use transition matrices to define the neighborhood for nodes. MoNet proposed by Monti et al. [29] is a spatial-domain model on non-Euclidean domains and could generalize several previous techniques. Bai et al. developed a novel Aligned-Spatial Graph Convolutional Network (ASGCN) model [30] and its backtrackless variant [31] to transform arbitrary-sized graphs into fixed-sized aligned grid structures in order to learn powerful features, which can not only reduce information loss and imprecise information representation arising in existing spatially-based GCNs but also bridge the theoretical gap between traditional CNNs and spatially-based GCNs. Furthermore, they propose a framework of computing the deep depth-based representations [32] for graph structures, applying it to GCN models such as the depth-based [33] and quantum-based [34] subgraph convolutional networks. Martino et al. investigated the graph embedding [35] and kernel [36] based on simplicial complexes. These two simplicial methods can be interpreted as possibly meaningful substructures (i.e., information granules) on the top of which an embedding space can be built employing symbolic histograms. GraphSAGE [1] is a general inductive framework which generates embeddings by sampling and aggregating features from the local neighborhood of a node. There are several works [11] to use gate mechanisms like GRU or LSTM to improve the long-term message propagation of information across the latent graph structure in the sequential data. Recently, non-spectral methods incorporate the attention mechanism into the message propagation step [10].

However, performing these spectral and non-spectral methods on the low-dimensional node feature space results in incomplete node-featured information learned and encounters accuracy deterioration. Although their aggregators can learn graph-structured information well in the message propagation step, their separate updaters, in the encoding output step, only update nodes' hidden states without any extra extracting the neighboring features of message passing. In the low-dimensional node feature space, to learn the graph-structured information only by the message propagation step is not enough. Therefore, we design the LGC and DSGC for aggregating the neighboring features of each node while encoding its hidden states in the encoding output step. Both the two iterative steps in our models have the capability of message passing to learn more graph-structured information when graphs involve the incomplete node representations of the low-dimensional node feature space.

3. Extensible architecture for GCNs

In this section, we will present the LGC used to construct the encoding output step in the building block layer, showing how to extend a standard convolution into our LGC updater. We also directly outline their benefits and limitations compared to prior work in the neural graph processing domain.

3.1. Aggregators and updaters of GCNs

We first present the updaters of GCN, PPNP, and APPNP on which we build LGC updater architecture step by step. For the set nodes V and set of edges E, we denote graph $G = (V, E)$. Let $|V|$ denotes the number of nodes, and $|E|$ is the number of edges. The node features represented by $X \in \mathbb{R}^{|V|\times|F|}$, where F is the features set of each node, and $|F|$ is the number of features. An adjacency matrix $A \in \mathbb{R}^{|V|\times|V|}$ represents graph G, while if extended by self-loops the adjacency matrix is $\hat{A} = A + I_{|V|}$. Hence, the simple GCN [14] is defined

$$Z = softmax(\tilde{A}\ \textbf{\textit{ReLU}}\,(\tilde{A}XW^{(0)})W^{(1)}) \tag{1}$$

where $Z \in \mathbb{R}^{|V|\times|L|}$ is predicted node labels set, L is node labels set, and $|L|$ is the number of node labels. $\tilde{A} = \hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}$ is the normalized graph Laplacian with diagonal degree matrix $\hat{D} = \sum_j \hat{A}_{ij}$. $W^{(0)}$ and $W^{(1)}$ are weight matrices equal to the uniform encoding function adjusted in training. *ReLU* is an activation function. Eq. (1) involves two message passing layers, which can be combined by $H^{(l+1)} = \sigma(\tilde{A}H^{(l)}W^{(l)})$, where $H^{(l)} \in \mathbb{R}^{|V|\times|C^{(l)}|}$ is feature maps in layer l, $C^{(l)}$ represents channels set, and $|C^{(l)}|$ is the number of channels. $H^{(l+1)} \in \mathbb{R}^{|V|\times|C^{(l+1)}|}$ is feature maps in layer $l+1$. $W^{(l)} \in \mathbb{R}^{|C^{(l)}|\times|C^{(l+1)}|}$ is a trainable weight matrix in layer l. The encoding function $W^{(l)}$ is a multichannel unidimensional convolution, and it is used to adjust feature dimension like 1×1 convolution for images. σ is an activation function.

We can further generalize $H^{(l+1)} = \sigma(\tilde{A}H^{(l)}W^{(l)})$ and divide it into two parts. The first part is the normalized graph Laplacian $\tilde{A}H^{(l)}$ as an aggregator, which is generalized as an optional aggregator $S(H^{(l)})$ to perform message passing. The second part is $f_\theta(S(H^{(l)}))$ as an updater, where f_θ can be defined as an alternative updater with a parameter set θ. Therefore, Eq. (1) can be generalized by the composite function $f_\theta(S(H^{(l)}))$ to be formalized as

$$Z = \textbf{\textit{softmax}}(f_\theta^{(1)}(S(\sigma(f_\theta^{(0)}(S(X)))))). \tag{2}$$

By employing the recurrent equation of personalized PageRank [17], $\pi_{ppr}(\vec{i}_x) = (1-\alpha)\tilde{A}\pi_{ppr}(\vec{i}_x) + \alpha\vec{i}_x$, which takes the root node x into account compared with original PageRank, $\pi_{pr} = A_{rw}\pi_{pr}$, Klicpera et al. [15] proposed two PPR aggregators: PPNP and APPNP, where the teleport (or restart) probability $\alpha \in (0, 1]$ controls the proportion of importance of root node x. The teleport vector $\vec{i}_x$ represents the root node x as a one-hot indicator vector. $A_{rw} = AD^{-1}$ with $D_{ij} = \sum_j A_{ij}$. By solving the recurrent equation of personalized PageRank to obtain $\pi_{ppr}(\vec{i}_x) = \alpha(I_{|V|} - (1-\alpha)\tilde{A})^{-1}\vec{i}_x$ and substituting $\vec{i}_x$ with the unit matrix $I_{|V|}$, the equation of PPNP is

$$\begin{aligned} Z &= \textbf{\textit{softmax}}(\alpha(I_{|V|} - (1-\alpha)\tilde{A})^{-1}H) \\ H &= f_\theta(X) \end{aligned} \tag{3}$$

where H is the feature maps in the last layer. The updater $f_\theta(X)$ is defined as an arbitrary neural network. The teleport probability $\alpha \in (0, 1]$ [15]. To solve the directly calculating the inverse matrix of PPNP that is computationally inefficient, a variant of topic-sensitive PageRank [37] used to simplify PPNP to form APPNP

$$\begin{aligned} Z^{(0)} &= H = f_\theta(X) \\ Z^{(k+1)} &= (1-\alpha)\tilde{A}Z^{(k)} + \alpha H \\ Z^{(K)} &= \textbf{\textit{softmax}}((1-\alpha)\tilde{A}Z^{(K-1)} + \alpha H) \end{aligned} \tag{4}$$

where K defines the number of power iteration steps and $k \in [0, K-2]$. Note that if we take the limit $k \to \infty$, Eq. (4) is equivalent to Eq. (3) [15]. Eqs. (3) and (4) can be generalized by the updater f_θ and aggregator S in Eq. (2). The modularized PPNP and APPNP are formalized as Eqs. (5) and (6), respectively

$$\begin{aligned} S(x) &= \alpha(I_{|V|} - (1-\alpha)\tilde{A})^{-1}x \\ Z &= \textbf{\textit{softmax}}(S(f_\theta(X)) \end{aligned} \tag{5}$$

$$\begin{aligned} S(z, x) &= (1-\alpha)\tilde{A}z + \alpha x \\ Z^{(0)} &= f_\theta(X) \\ Z^{(k+1)} &= S(Z^{(k)}, Z^{(0)}) \\ Z^{(K)} &= \textbf{\textit{softmax}}(S(Z^{(K-1)}, Z^{(0)})) \end{aligned} \tag{6}$$

3.2. Expansibility of GCNs

Previous researchers [13] prioritized the improvement and redesign of the aggregator S. We purpose to design a new updater, namely LGC $f_{\theta,lgc}$, to embed the encoding updater function f_θ after modularizing GCNs. Each node in $G(V, E)$ is equal to a unit

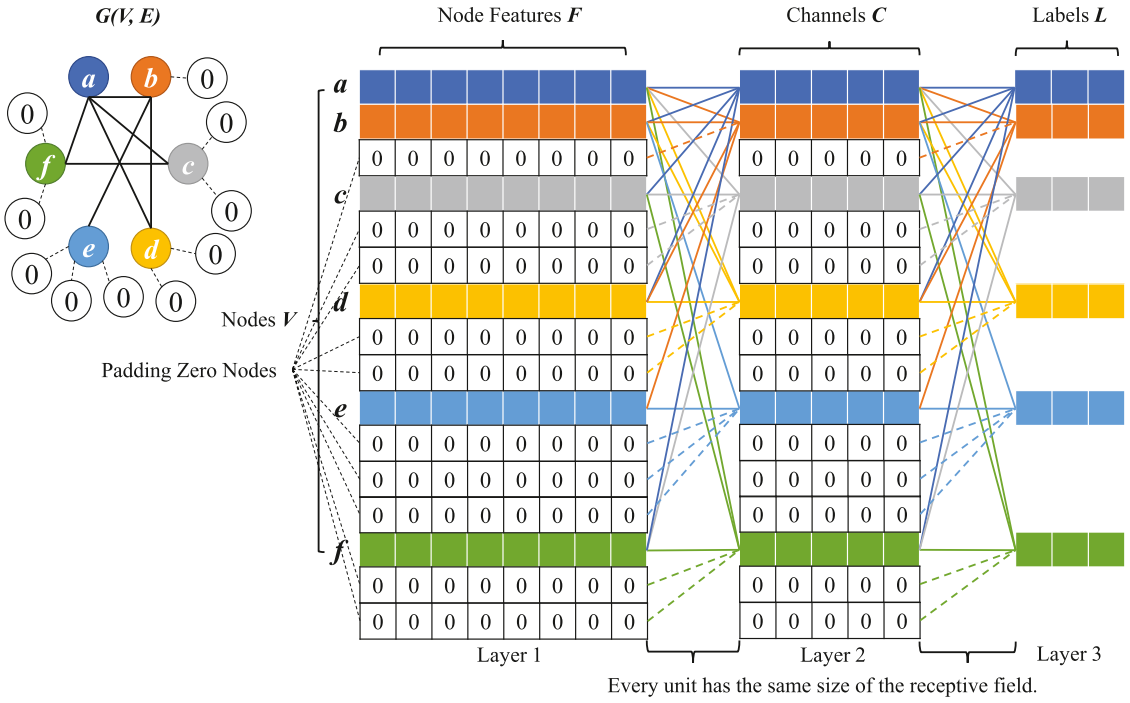


Fig. 1. Padding process from $G(V, E)$ to the architecture of $f_{\theta,lgc}$. In graph G, we pad some zero nodes to connect each node by dotted edges such that they have the same degree, which in turn results in each unit in layer $l+1$ having the same size of the neighborhood. The zero nodes participate in the construction of the architecture in which the dotted connections of zero units complement the size of the neighborhood to achieve a consistent size.

in layer $l+1$. Units are connected to itself and neighbor units in layer l, according to edges E of graph $G(V, E)$. However, each node in $G(V, E)$ has a different degree, which leads to the diverse size of the neighborhood of each unit in feature maps. To solve the neighborhood's inconsistency, we initially define a *mask* function to learn the neighborhood of each unit

$$RF_n^{(l+1)} = \boldsymbol{mask}_n(H^{(l)}) = \rho(H^{(l)} \odot M_n^{(l+1)}) \tag{7}$$

where $H^{(l)} \in \mathbb{R}^{|V|\times|C^{(l)}|}$ is feature maps in layer l, especially $H^{(1)} = X \in \mathbb{R}^{|V|\times|F|}$ when $l=1$ and $C^{(1)} = F$. $RF_n^{(l+1)}$ is a neighborhood of unit n in layer $l+1$. $\odot$ is two matrices element-wise product. $M_n^{(l+1)}$ is a mask matrix of unit n in layer $l+1$, with elements 0 or 1 and dimensions the same as $H^{(l)}$. If unit n in layer $l+1$ and unit k in layer l have an edge in graph G, elements in the dimension $k \in V$ and $n \in V$ of $M_n^{(l+1)}$ will be set to 1. Otherwise, they will be set to 0. The compression function ρ deletes 0 elements. In addition, $RF_n^{(l+1)} \in \mathbb{R}^{|D_n^{(l+1)}|\times|C^{(l)}|}$, where $D_n^{(l+1)}$ represents connections of unit n and $|D_n^{(l+1)}|$ represents degree of node n. The order of connections $D_n^{(l+1)}$ is ranked by the affinity values, which is either node ID, uniformly sampling [1] or graph centrality [16] (degree, eigenvector, closeness, betweenness, harmonic, load, subgraph, etc.). Particularly, connections $D_n^{(l+1)}$ are sorted by the descending order of node centrality. For each node, degree centrality is understood as the number of node incident links. Eigenvector centrality measures how the node affects the network by assigning scores to all nodes related to connections with high-scoring nodes. Closeness centrality is the average length of the shortest path between each node and all other nodes in the graph. Betweenness centrality is the number of bridge positions along the shortest path between nodes. Harmonic centrality reverses the sum reciprocal operations from closeness centrality. Load centrality is a variant of the betweenness centrality. Subgraph centrality of a node is the sum of weighted closed walks of all lengths starting and ending at a node. Furthermore, Eq. (7) can be extended as

$$\begin{aligned} RF_{\text{mask}}^{(l+1)} &= \boldsymbol{mask}(H^{(l)}) = \{RF_n^{(l+1)}\}_{n\in V} \\ &= \rho(H^{(l)} \odot \{M_n^{(l+1)}\}_{n\in V}) \end{aligned} \tag{8}$$

where $RF_{\text{mask}}^{(l+1)}$ is an ordered set $\{RF_n^{(l+1)}\}_{n\in V}$ which contains each $RF_n^{(l+1)}$ ranked by node ID. We extract the inconsistent neighborhood of each unit $RF_n^{(l+1)}$, where we find the maximum degree $|D_n^{(l+1)}|$ as $|D_{\text{max}}^{(l+1)}|$. Then, element 0 is put into $RF_n^{(l+1)}$ along the dimension $|D_n^{(l+1)}|$ until $|D_n^{(l+1)}| = |D_{\text{max}}^{(l+1)}|$. This padding process is shown in Fig. 1. Each node from a to f in $G(V, E)$ is padded by zero nodes so that they have the same degree, and eventually, each unit has the same size of the neighborhood. In the ordered set $RF_{\text{mask}}^{(l+1)}$, every $RF_n^{(l+1)}$ has the same shape, that is, $(|D_{\text{max}}^{(l+1)}|, |C^{(l)}|)$, implying that every $RF_n^{(l+1)} \in \mathbb{R}^{|D_{\text{max}}^{(l+1)}|\times|C^{(l)}|}$ and $RF_{\text{mask}}^{(l+1)} \in \mathbb{R}^{|V|\times|D_{\text{max}}^{(l+1)}|\times|C^{(l)}|}$. The padding process can be defined as the function ***pad***; therefore the updater LGC $f_{\theta,lgc}$ is

$$\begin{aligned} H^{(l+1)} &= f_{\theta,lgc}(H^{(l)}) \\ &= \sigma(\boldsymbol{reshape}(\boldsymbol{pad}(\boldsymbol{mask}(H^{(l)})))W^{(l)}) \\ &= \sigma(\boldsymbol{reshape}(\boldsymbol{pad}(RF_{\text{mask}}^{(l+1)}))W^{(l)}) \end{aligned} \tag{9}$$

where function ***reshape*** is to reshape $RF_{\text{mask}}^{(l+1)}$ from shape $(|V|, |D_{\text{max}}^{(l+1)}|, |C^{(l)}|)$ to $(|V|, |D_{\text{max}}^{(l+1)}| \times |C^{(l)}|)$. Supposing that $W^{(l)}$ is a training weight matrix and its shape is $(|D_{\text{max}}^{(l+1)}| \times |C^{(l)}|, |C^{(l+1)}|)$, then $RF_{\text{mask}}^{(l+1)}$ can be multiplied by $W^{(l)}$. Nevertheless, in exceptional cases, a node in graph G has an excessively maximum degree, which makes $|D_{\text{max}}^{(l+1)}|$ a tremendous value to result in superfluous weights of $W^{(l)}$ and thus to increase the cost of our models. Hence, we define a threshold th to limit the excessively maximum degree, i.e., $|D_{\text{max}}^{(l+1)}| = min(|D_{\text{max}}^{(l+1)}|, th)$. Moreover, similar to the order of connections $D_n^{(l+1)}$, the order of connections $D_{\text{max}}^{(l+1)}$ is ranked by the affinity values. The output $H^{(l+1)} \in \mathbb{R}^{|V|\times|C^{(l+1)}|}$ has the shape $(|V|, |C^{(l+1)}|)$, which is homogeneous with the shape $(|V|, |C^{(l)}|)$

Table 1
The difference in the frameworks of LGC-GCNs.

Module	LGC-GCN	LGC-PPNP	LGC-APPNP
Aggregator	$S(x) = \tilde{A}x$	$S(x) = \alpha(I_{\|V\|} - (1-\alpha)\tilde{A})^{-1}x$	$S(z,x) = (1-\alpha)\tilde{A}z + \alpha x$
LGC Updater	$f^{(l)}_{\theta,lgc}(x) = \boldsymbol{reshape}(\boldsymbol{pad}(\boldsymbol{mask}(x)))W^{(l)}$		
LGC Pipeline	$f^{(l)}_{\theta}(x) = f^{(l+1)}_{\theta,lgc}(S(\boldsymbol{ReLU}(f^{(l)}_{\theta,lgc}(S(x)))))$	$f^{(l)}_{\theta}(x) = f^{(l+1)}_{\theta,lgc}(\boldsymbol{ReLU}(f^{(l)}_{\theta,lgc}(x)))$	
Output	$Z = \boldsymbol{softmax}(f^{(0)}_{\theta}(X))$	$Z = \boldsymbol{softmax}(S(f^{(0)}_{\theta}(X)))$	$Z^{(0)} = f^{(0)}_{\theta}(X)$ $Z^{(k+1)} = S(Z^{(k)}, Z^{(0)})$ $Z^{(K)} = \boldsymbol{softmax}(S(Z^{(K-1)}, Z^{(0)}))$

of $H^{(l)} \in \mathbb{R}^{|V|\times|C^{(l)}|}$. So far, we have built the updater LGC $f_{\theta,lgc}$ to be further combined with the aggregators of GCN [14], PPNP, and APPNP [15], respectively, to form LGC-GCN, LGC-PPNP, and LGC-APPNP. The detail of LGC-GCNs' equations showed in Table 1.

3.3. Comparisons to related work

The updater LGC described in Section 3.2 has several merits that were not present in prior approaches to modeling graph-structured data with neural networks:

(1) Structurally, LGC leverages the improved standard convolutions to aggregate neighbors' features while encoding each node's hidden representations in the encoding output step. Compared with the prior updaters [13], our LGC can assist aggregators in performing message passing in the updating process, which breaks the barrier of aggregators and updaters to learn graph-structured information when the graph involves the low-dimensional node features.
(2) Another merit of leveraging the standard convolution in our LGC is that: we generalize convolutional methods from CNNs to GNNs, building a bridge between them to enrich GNNs' methods via existing or up-to-date methods in CNNs introduced. For instance, in the following section, we transfer the depth-wise separable convolution [21] into our models based on generalizing the standard convolution [3] into our LGC. There are many variants of convolutions in CNNs worth trying on our model in the next work.
(3) GraphSAGE [1] achieved some of its state-of-the-art results when an LSTM-based neighborhood aggregator used on it. Researchers assumed a consistent sequential node ordering across neighborhoods in their GraphSAGE via rectifying this model by consistently feeding randomly-ordered sequences to the LSTM. In our LGC, we can use any graph centrality as the affinity value to order neighborhoods without using extra computations to rectify models. Choosing graph centrality to our model is based on an assumption of the power-law distribution in graphs [38], that is, the category of each node is determined by its most crucial neighbors, not by its all neighborhood. Our model especially costs less to rank neighborhoods when degree centrality is the affinity value, which provides a priori ordered information of the graph structure.
(4) Gate mechanisms used in Tree-LSTM [11] have taken advantage of long-term message propagation of information across the latent graph structure in the sequential data. This model based on LSTM only targets to address many natural language processing (NLP) problems while their text input has an explicit, sequential relationship and a latent graph structure. However, for most other graph-structured data, they have an explicit graph structure without the sequential relation. For learning these data, our LGC has hence more applicable and feasible instead of Tree-LSTM specialized in text applications. Furthermore, compared with Tree-LSTM, in the next section, we extend the lightweight convolution to our model to shrink model costs.

Of course, our LGC is not perfect and has several defects to be improved:

(1) Our model has a fixed-size neighboring set similar to GraphSAGE [1] or Tree-LSTM [11] (a fixed-size children nodes set). Despite assuming power-law distribution existed between most of the nodes and their most crucial neighbors, in LGC, the fixed-size of the neighborhood results in the loss of information contributed by the rest of the neighbors of these nodes. Removing the threshold th in order to get all of the neighborhood can reserve the information of all neighbors for each node, but the calculating cost increases sharply. Therefore, trying to reserve all neighbors' information as far as possible is an improvement direction. ASGCN [30] utilized the transformation from arbitrary-sized graphs into fixed-sized aligned grid structures to reserve the neighboring information, which gives us inspiration for our next work.
(2) At present, our LGC can not be used in the graph classification task. Fortunately, the research of exploring simplicial complexes introduced in the graph embedding [35] and kernel [36] provides an idea to target the graph classification, which will be employed in our LGC in future work.

4. Depth-wise separable graph convolution

In this section, we describe how to integrate DSC into our proposed LGC to form the DSGC in our novel MobileGCNs. DSC is a lightweight variant of the standard convolution used in CNNs. We can use it as a building block in light neural network architectures [21]. In DSC, we replace a standard convolution with a factorized version, which splits the standard convolution into two separate layers. The first is a depth-wise convolution which performs filtering by applying a single convolution filter per input channel. The second one is 1×1 convolution, also called a point-wise convolution. As a result, we receive output channels of the new feature maps through the linear combinations of the input channels. The shape of the input tensor $H^{(l)}$ in images is $(h^{(l)}, w^{(l)}, |C^{(l)}|)$. The convolution kernel $K^{(l)} \in \mathbb{R}^{k\times k\times|C^{(l)}|\times|C^{(l+1)}|}$ is applied to $H^{(l)}$ to produce an output tensor $H^{(l+1)}$ with shape $(h^{(l)}, w^{(l)}, |C^{(l+1)}|)$, where $h^{(l)}$ is the height of an image, $w^{(l)}$ is the width of the image, and k is the size of the convolution kernel $K^{(l)}$. The computational cost of standard convolutional layers is $h^{(l)} \cdot w^{(l)} \cdot |C^{(l)}| \cdot |C^{(l+1)}| \cdot k \cdot k$. DSC works almost as well as standard convolution but costs only $h^{(l)} \cdot w^{(l)} \cdot |C^{(l)}| \cdot (k^2 + |C^{(l+1)}|)$, which is the sum of depth-wise and point-wise convolutions. The cost ratio of DSC to standard convolutions is thereby $\frac{1}{|C^{(l+1)}|} + \frac{1}{k^2}$. The layer of DSGC cost is $|V| \cdot |C^{(l)}| \cdot (|D^{(l+1)}_{\max}| + |C^{(l+1)}|)$, which is $\frac{1}{|C^{(l+1)}|} + \frac{1}{|D^{(l+1)}_{\max}|}$ of the cost of LGC $|V| \cdot |C^{(l)}| \cdot |C^{(l+1)}| \cdot |D^{(l+1)}_{\max}|$.

The proposed DSGC updater has two convolutions, the same as DSC. One is the depth-wise graph convolution, which has the same effect as the depth-wise convolution and is used on graphs. The

Table 2
The difference in the frameworks of MobileGCNs.

Module	MobileGCN	MobilePPNP	MobileAPPNP
Aggregator	$S(x) = \tilde{A}x$	$S(x) = \alpha(I_{\|V\|} - (1-\alpha)\tilde{A})^{-1}x$	$S(z,x) = (1-\alpha)\tilde{A}z + \alpha x$
DSGC Updater	$f_{\theta,d}^{(l)}(x) = \omega(\boldsymbol{TM}((\boldsymbol{pad}(\boldsymbol{mask}(x)))^T, W_d^{(l)}))$		
	$f_{\theta,p}^{(l)}(x) = xW_p^{(l)}$		
DSGC Pipeline	$f_\theta^{(l)}(x) = f_{\theta,p}^{(l+1)}(\boldsymbol{ReLU}(f_{\theta,d}^{(l+1)}(S(\boldsymbol{ReLU}(f_{\theta,p}^{(l)}(S(x)))))))$	$f_\theta^{(l)}(x) = f_{\theta,p}^{(l+1)}(\boldsymbol{ReLU}(f_{\theta,d}^{(l+1)}(\boldsymbol{ReLU}(f_{\theta,p}^{(l)}(x)))))$	
Output	$Z = \boldsymbol{softmax}(f_\theta^{(0)}(X))$	$Z = \boldsymbol{softmax}(S(f_\theta^{(0)}(X)))$	$Z^{(0)} = f_\theta^{(0)}(X)$ $Z^{(k+1)} = S(Z^{(k)}, Z^{(0)})$ $Z^{(K)} = \boldsymbol{softmax}(S(Z^{(K-1)}, Z^{(0)}))$

other one is the point-wise graph convolution, which is also used on graphs. We define the function depth-wise graph convolution $f_{\theta,d}$ and point-wise graph convolution $f_{\theta,p}$ to split the LGC updater $f_{\theta,lgc}$ (Eq. (9)) as

$$\begin{aligned} f_{\theta,d}^{(l)}(x) &= \omega(\boldsymbol{TM}((\boldsymbol{pad}(\boldsymbol{mask}(x)))^T, W_d^{(l)})) \\ &= \omega(\boldsymbol{TM}((\boldsymbol{pad}(RF_{\text{mask}}^{(l+1)}))^T, W_d^{(l)})) \\ f_{\theta,p}^{(l)}(x) &= xW_p^{(l)} \\ H^{(l+1)} &= f_\theta(H^{(l)}) = \sigma(f_{\theta,p}^{(l)}(\sigma(f_{\theta,d}^{(l)}(H^{(l)})))) \end{aligned} \tag{10}$$

where $f_{\theta,d}^{(l)}(x)$ uses the tensor transpose operation T to transform the shape of the padded $RF_{\text{mask}}^{(l+1)}$ from $(|V|, |D_{\max}^{(l+1)}|, |C^{(l)}|)$ to $(|C^{(l)}|, |V|, |D_{\max}^{(l+1)}|)$, a tensor multiplication is defined as $\boldsymbol{TM}$ in which $W_d^{(l)} \in \mathbb{R}^{|C^{(l)}|\times|D_{\max}^{(l+1)}|\times 1}$ is training weights for depth-wise graph convolution. $\boldsymbol{TM}$ performs the matrix multiplication of $|C^{(l)}|$ pairs of matrices $(|V|, |D_{\max}^{(l+1)}|)$ in padded $RF_{\text{mask}}^{(l+1)}$ and $(|D_{\max}^{(l+1)}|, 1)$ in $W_d^{(l)}$. The shape of the output of $\boldsymbol{TM}$ is $(|C^{(l)}|, |V|, 1)$. Hence, we define function ω to transform the output shape from $(|C^{(l)}|, |V|, 1)$ to $(|V|, |C^{(l)}|)$ for the matrix multiplication of $f_{\theta,p}^{(l)}(x)$ whose training weights $W_p^{(l)} \in \mathbb{R}^{|C^{(l)}|\times|C^{(l+1)}|}$ are for point-wise graph convolution. Particularly, $f_{\theta,p}^{(l)}(x) = xW_p^{(l)}$ for multichannel one-dimensional feature maps in Eq. (10) is matrix multiplication. The input of layer $l+1$ is $H^{(l+1)} \in \mathbb{R}^{|V|\times|C^{(l+1)}|}$. σ is an activation function.

In this paper, we design three MobileGCNs: MobileGCN, MobilePPNP, and MobileAPPNP. MobileGCN is a combination of DSGC ($f_{\theta,d}$, $f_{\theta,p}$) and the normalized graph Laplacian S of the GCN [14]. Likewise, MobilePPNP and MobileAPPNP are PPNP [15] and APPNP [15] combined with DSGC, respectively. To explicitly show these three MobileGCNs, we summarize them in Table 2.

5. Experimental setup

5.1. Datasets

For our experiments, we have used open benchmark datasets of four types. Citation Graphs (CITESEER, CORA, and PUBMED) [14] contain high-dimensional node features, with each node as a document containing hundreds or thousands of features described by high-frequency words in the corpus of all these documents. CITESEER and CORA are respectively composed of the citation relation of many machine-learning papers. In CORA, these documents are divided into one of seven classes, while the CITESEER dataset has six class labels. PUBMED dataset is a citation network containing a set of articles related to diabetes from the PUBMED database. We employ the Wikipedia Language Graph as the second type dataset [33]. Real-word webpages as the nodes compose this dataset. All nodes come from nineteen classes, and hyperlinks between them represent the edges which indicate a hyperlink from one webpage to another. Node features of this dataset are webpage text content. Amazon Photo Graph [39] as the third type dataset is segments of Amazon co-purchase graph, where nodes represent goods, node features are bag-of-words encoded product reviews, and the product category gives the classes. The fourth type dataset is Urban Road Graph (ROAD), which includes only one node feature to predict traffic flow [40]. ROAD includes low-dimensional node features because the traffic equipment can only collect a few types of information. Nodes of ROAD graph represent roads, while edges represent intersections between roads. The only node feature of the ROAD graph is traffic density, which derived from statistics of the GPS of cars in the Didi Chuxing GAIA Initiative (https://gaia.didichuxing.com). The ROAD graph has three labels: Rush Traffic (traffic jam), Low Traffic (there is a little traffic which usually happens at night), and Normal Traffic (the traffic with no congestion). We use these six datasets mentioned above for semi-supervised node classification tests over our proposed model. All models in our experiments have been tested exclusively on the same train/validation/test splits [14] of CITESEER, CORA, and PUBMED datasets. For Wikipedia, Photo, and ROAD datasets, we sample randomly 10% nodes from each dataset as the training set, 20% nodes as the validation set, and the rest of nodes as the test set. Table 3 reports the statistics of these datasets.

5.2. Benchmark models

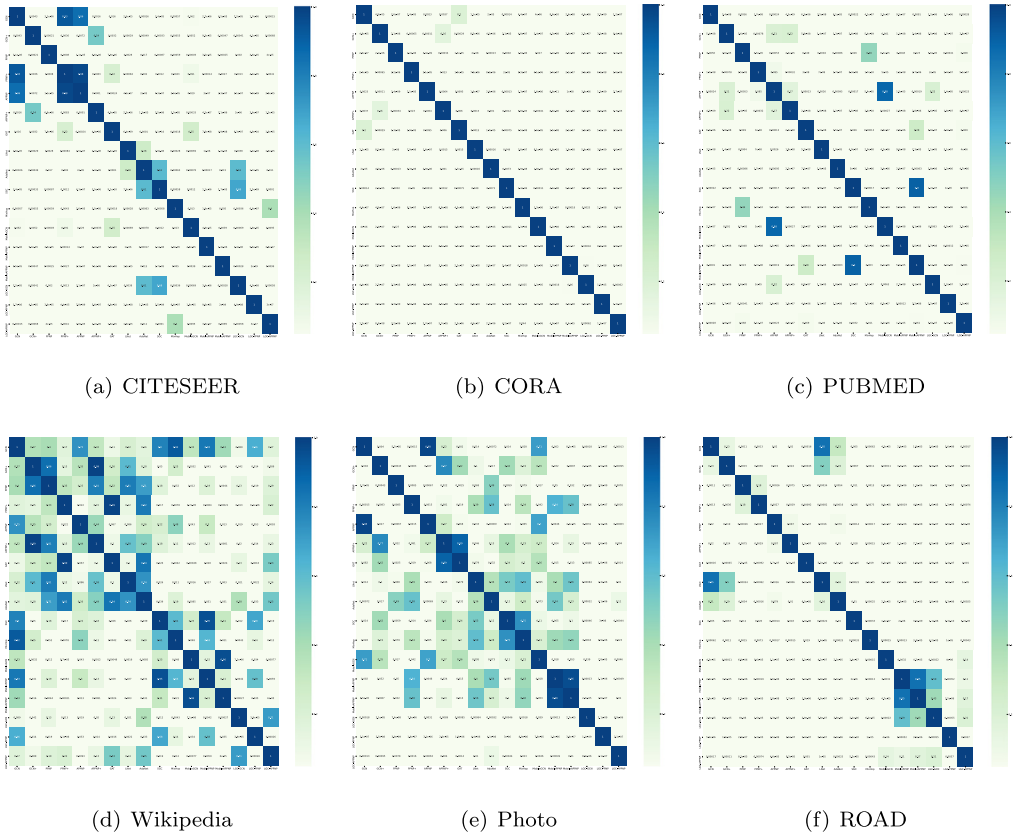
To compare our proposed models with eight state-of-the-art models, we have taken examples from: the GCN [14], GAT [10], LanczosNet (LNet and AdaLNet) [24], simplified GCN (SGC) [25], MixHop GCN [26], PPNP [15], and APPNP [15]. These benchmark models have been introduced in Section 2.

5.3. Hyper-parameters and metrics

For the experimental research, we have used the architectures of LGC-GCNs and MobileGCNs with two layers. Their channels of the hidden layer are $|C^{(1)}| = 32$, which is also the value for all models except GCN†, PPNP†, and APPNP†. These three models, GCN†, PPNP†, and APPNP†, gathered as the control group are $|C^{(1)}| = 1024$, and have the same number of weights as our LGC-GCNs. For PPNP, APPNP, MobilePPNP, MobileAPPNP, LGC-PPNP, and LGC-APPNP, $\alpha = 0.06$ and $K = 2$. We also define the other hyper-parameters of benchmark models according to the experiment setting in their papers because every model has different learning conditions and convergence. For our models, we set the learning rate to 0.0045, and epochs are 1000. To implement feature selection experiments, we select a few features from five datasets (CITESEER, CORA, PUBMED, Wikipedia, and Photo) to form the low-dimensional feature space according to the random selection, Chi^2 test, analysis of variance, and mutual information, respectively. For node classification tasks, we adopt accuracy, Macro-F1, and MCC metrics. Accuracy is one of the most common metrics, but it cannot precisely describe the imbalance of learning data. Hence, Macro-F1 gives equal weight to each category; it tends to be dominated by rare categories. MCC takes true and false positives and negatives into account, and it is generally regarded as a balanced measure which can be used even if the classes are of very different sizes, and the range of its value is –1 to 1 (or –100% to 100%).

Table 3
Research dataset statistics.

Dataset	Type	Classes	Features	Nodes	Edges	Label rate
CITESEER	Citation	6	3703	3327	4732	0.036
CORA	Citation	7	1433	2708	5429	0.052
PUBMED	Citation	3	500	19,717	44,338	0.003
Wikipedia	Language	19	4973	2405	17,981	0.100
Photo	Purchase	8	745	7487	119043	0.100
ROAD	Urban Road	3	1	14,569	27,353	0.100



(a) CITESEER (b) CORA (c) PUBMED

(d) Wikipedia (e) Photo (f) ROAD

Fig. 2. The p-value heatmap of the paired t-test with respect to accuracy (1% features for all datasets).

6. Research results and discussion

6.1. Overall metrics

Metric comparison results are presented in Table 4, 5, 6, 7, 8, 9, and the p-value heatmap of the paired t-tests between these models being compared are shown in Fig. 2, 3, 4. Depending on these datasets, we sample top 1% to 10% node features to constitute the low-dimensional feature space and select 100% node features to form the high-dimensional feature space. We also repeat the learning process five times and report the average performance in terms of accuracy, Macro-F1, and MCC, as many other works do [33]. In each of presented tables the boldfaced values are the best results in each experiment. We have following observations in Table 4, 5, 6, 7, 8, 9 and Fig. 2, 3, 4:

(1) In the low-dimensional feature space, one of our six proposed models always outperforms each of the competing benchmark models with sampling different ratios of top features. Specifically, LGC-PPNP shows the most outstanding accuracy on citation graph datasets (CITESEER, CORA, and PUBMED), following by MobilePPNP. Moreover, the accuracy of LGC-PPNP is more prominent when fewer features describe graph nodes since its p-value less 0.05 shown in Fig. 2. For Wikipedia, Photo, and ROAD datasets represented by the low-dimensional feature space, LGC-PPNP has the dominant accuracy though it has less significance (p-value $>$ 0.05 in Fig. 2 (d) Wikipedia) for differences with other two baseline models (GCN and SGC). It indicates the effectiveness and robustness of LGC updater to assist aggregators to gather graph-structured information on the node classification task. In brief, LGC updater acts as a powerful auxiliary to help aggregators to capture more graph-structured information, while learning node hidden states.

(2) In general, for all datasets except Photo, LGC-APPNP performed on Macro-F1 and MCC has the state-of-the-art results in the low-dimensional feature space, and its difference in results against other models are significant according to p-value $<$ 0.05 in Fig. 3 and 4. A possible explanation is that LGC-APPNP as an approximate formalization of LGC-PPNP has a smaller neighboring size of performing message passing to suppress over-smoothing [15], which gives the fit individuality to node features for building distinct classification boundaries between nodes, rather than the homogeneity to them to blur their boundaries.

(3) Although MobilePPNP and MobileAPPNP are not the most outstanding on each metric, they maintain high performance and robustness from low- to high-dimensional node fea-

Table 4
Evaluation results of node classification on the CITESEER dataset.

Metric	Model	The percentage of the top node features (The number of the top node features)										
		1%(37)	2%(74)	3%(111)	4%(148)	5%(185)	6%(222)	7%(259)	8%(296)	9%(333)	10%(370)	100%(3703)
Accuracy	GCN [14]	31.10±1.20	43.42±0.35	46.21±1.13	47.45±0.51	47.46±0.11	48.60±0.02	52.46±1.18	50.93±0.13	51.80±0.13	51.86±0.24	71.70±0.40
	GCN† [14]	24.50±2.20	37.65±0.15	42.60±0.40	45.95±0.55	46.70±0.20	48.55±0.15	53.50±0.40	54.30±0.20	54.30±0.30	53.85±0.25	71.85±0.63
	PPNP [15]	42.15±0.65	52.92±0.45	55.88±0.42	56.24±1.16	56.40±0.32	58.54±0.65	59.61±0.23	59.11±0.40	60.90±0.29	59.92±0.45	71.95±0.15
	PPNP† [15]	29.40±3.00	39.00±0.50	46.05±0.15	49.90±0.50	52.65±0.45	54.65±0.25	59.00±0.10	59.20±0.20	59.25±0.10	59.05±0.15	72.08±0.22
	APPNP [15]	31.00±0.35	44.12±0.32	45.91±1.27	47.98±0.50	47.90±0.67	48.76±0.35	51.74±0.31	51.85±0.49	52.62±0.60	52.00±1.11	72.00±0.50
	APPNP† [15]	24.90±0.90	37.65±0.15	41.40±0.10	45.45±0.45	46.55±0.45	48.70±0.40	52.65±0.25	53.00±0.30	53.80±0.20	53.30±0.30	72.14±0.10
	GAT [10]	32.62±0.03	47.51±0.25	48.53±1.12	50.10±0.42	50.96±0.57	50.25±0.58	53.22±1.18	53.94±0.43	54.00±0.31	54.00±0.28	**72.15±0.25**
	LNet [24]	40.53±0.24	49.86±1.02	50.20±0.10	52.34±0.44	52.06±0.14	53.56±0.68	56.92±0.25	56.90±0.40	57±0.32	57.00±0.82	70.46±0.10
	AdaNet [24]	40.02±0.56	48.52±0.41	51.78±0.15	52.36±0.70	53.00±0.42	53.45±0.10	54.74±0.45	55.86±0.68	55.29±0.40	56.48±0.60	71.13±0.20
	SGC [25]	39.52±0.11	49.08±0.43	49.86±0.80	50.36±0.65	51.18±0.40	51.86±1.06	52.85±0.36	52.14±0.25	52.30±1.40	51.15±0.83	70.02±0.02
	MixHop [26]	36.48±0.42	47.76±0.15	49.52±0.86	49.87±0.50	50.08±0.46	51.12±0.84	51.20±0.40	50.45±0.28	50.89±0.55	50.72±1.58	69.63±0.44
	MobileGCN	32.90±1.30	46.65±0.45	50.30±0.90	51.10±1.00	50.05±1.25	52.10±0.40	55.95±1.05	55.00±0.70	55.70±1.00	54.40±0.50	70.25±0.65
	MobilePPNP	47.24±0.33	54.41±0.26	56.72±0.15	59.30±1.17	57.22±0.20	**60.00±1.15**	**59.82±1.20**	**61.38±0.76**	**61.35±0.15**	**60.67±0.15**	70.05±0.35
	MobileAPPNP	38.00±0.10	47.61±0.23	49.25±0.05	49.73±0.17	51.20±0.18	51.42±0.20	54.80±0.26	54.12±0.55	55.28±0.37	54.50±0.10	70.20±0.80
	LGC-GCN	39.45±0.25	49.05±1.05	52.25±0.35	52.55±0.95	51.80±0.70	53.05±0.55	56.55±0.75	56.30±0.50	56.60±0.50	56.05±0.85	70.20±0.20
	LGC-PPNP	**50.37±0.16**	**55.42±1.10**	**57.55±0.53**	**60.12±1.06**	**60.00±1.08**	59.12±0.54	59.62±0.35	61.12±0.28	60.72±0.18	58.96±0.20	67.60±0.20
	LGC-APPNP	37.22±0.42	46.50±0.30	49.13±1.15	50.76±0.30	48.55±1.03	50.36±0.23	52.00±0.54	51.73±1.20	52.10±0.63	52.78±0.58	70.75±0.15
Macro-F1	GCN [14]	32.71±0.46	38.64±0.63	40.34±0.79	41.68±0.66	41.41±0.55	42.24±0.32	43.15±0.57	43.33±0.58	43.54±0.34	42.92±0.78	53.04±0.74
	GCN† [14]	33.65±0.16	38.54±0.13	40.56±0.10	41.45±0.05	41.32±0.05	41.82±0.21	43.58±0.15	43.39±0.15	43.54±0.14	43.51±0.10	54.86±0.66
	PPNP [15]	37.15±0.35	41.72±0.56	43.14±0.40	44.22±0.27	44.60±0.46	45.55±0.30	46.65±0.25	47.08±0.54	47.57±0.46	47.21±0.49	54.48±0.72
	PPNP† [15]	36.53±0.14	40.82±0.14	42.09±0.11	42.50±0.06	43.21±0.10	44.01±0.15	45.45±0.17	45.89±0.07	45.89±0.20	45.87±0.14	54.96±0.24
	APPNP [15]	32.79±0.17	38.66±0.49	39.77±0.30	41.87±0.66	41.52±0.23	42.17±0.52	43.70±0.36	43.13±0.48	43.63±0.44	43.97±0.63	53.57±0.63
	APPNP† [15]	33.40±0.26	38.50±0.07	40.71±0.10	41.34±0.07	41.82±0.26	42.18±0.16	43.73±0.18	43.53±0.11	43.65±0.09	43.66±0.07	53.85±0.66
	GAT [10]	33.11±0.23	41.80±0.40	45.57±0.35	47.23±0.30	46.92±0.30	47.78±0.19	50.01±0.33	50.68±0.18	50.32±0.08	50.88±0.34	**68.27±0.41**
	LNet [24]	33.20±0.16	37.22±0.28	40.16±0.25	42.64±0.36	42.86±1.27	42.18±0.09	43.39±0.15	44.08±0.20	44.56±0.10	45.20±0.32	54.40±0.10
	AdaNet [24]	32.18±0.20	39.12±0.70	40.89±0.16	41.28±0.15	41.43±0.11	42.58±0.42	42.60±0.82	43.10±0.05	43.20±0.17	44.02±0.08	53.26±0.45
	SGC [25]	31.46±0.20	38.28±0.06	39.40±0.18	41.25±0.60	41.28±0.15	41.58±0.20	42.85±0.07	42.20±0.76	42.11±0.20	44.42±0.27	54.28±0.16
	MixHop [26]	34.90±0.24	40.86±0.41	41.56±1.00	41.98±0.20	42.47±0.39	42.81±0.40	43.19±0.28	44.56±0.11	44.17±0.10	44.89±0.23	53.80±0.16
	MobileGCN	33.80±0.45	40.37±0.87	42.29±0.66	42.54±0.69	42.69±0.66	42.82±0.68	44.21±0.61	44.37±0.72	44.33±0.55	44.28±0.27	53.91±0.33
	MobilePPNP	40.52±0.71	43.35±0.57	45.04±0.46	46.49±0.34	46.83±0.58	47.34±0.42	48.19±0.81	47.81±0.54	48.76±0.90	48.58±0.57	54.32±0.53
	MobileAPPNP	35.16±0.46	40.75±0.44	42.37±0.36	43.61±0.68	43.89±0.61	43.72±0.43	45.09±0.44	44.61±0.46	45.06±0.37	44.53±0.56	53.78±0.35
	LGC-GCN	36.05±0.26	40.66±0.88	42.33±0.64	43.22±0.27	43.03±0.51	43.50±0.33	44.61±0.91	44.83±1.05	45.05±0.82	45.05±0.77	54.85±0.60
	LGC-PPNP	**42.66±0.68**	**45.12±0.64**	**46.10±0.67**	**47.39±0.48**	**47.52±0.43**	**47.90±0.61**	47.61±0.29	48.20±0.34	48.61±0.27	48.80±0.58	55.78±0.25
	LGC-APPNP	36.24±0.34	42.79±0.14	45.11±0.21	46.78±0.18	46.82±0.18	47.75±0.32	**50.94±0.13**	**50.79±0.22**	**50.40±0.06**	**51.11±0.20**	64.29±0.30
MCC	GCN [14]	15.71±1.22	23.91±1.48	25.80±1.15	27.99±1.13	27.99±0.61	29.36±0.40	30.43±0.55	30.83±0.89	31.25±0.28	30.36±1.08	46.81±0.69
	GCN† [14]	15.78±0.19	23.23±0.16	26.32±0.29	27.58±0.20	27.67±0.36	28.79±0.18	31.40±0.05	31.09±0.24	31.31±0.33	31.21±0.16	46.60±0.20
	PPNP [15]	21.91±0.44	27.75±0.88	29.57±0.44	31.40±0.50	32.09±0.40	33.43±0.38	35.41±0.33	35.57±0.81	36.29±0.87	36.08±0.69	47.61±0.78
	PPNP† [15]	20.84±0.16	26.16±0.28	27.59±0.27	28.25±0.24	29.72±0.15	31.22±0.22	33.14±0.18	33.66±0.08	33.80±0.25	33.78±0.21	48.65±0.88
	APPNP [15]	15.33±0.48	24.18±1.07	24.52±0.13	28.17±1.33	27.86±0.26	28.68±0.80	30.98±0.78	30.41±0.58	31.52±1.06	31.47±0.84	47.16±0.87
	APPNP† [15]	15.31±0.50	23.23±0.18	26.37±0.14	27.18±0.09	28.19±0.52	28.91±0.22	31.28±0.25	30.79±0.14	30.90±0.15	31.10±0.13	48.50±1.25
	GAT [10]	17.39±0.60	29.96±0.88	34.09±0.39	36.11±0.47	36.21±0.43	37.08±0.31	40.14±0.61	40.55±0.20	40.12±0.47	40.88±0.78	**65.03±0.55**
	LNet [24]	21.13±0.28	26.72±0.38	27.42±1.03	29.12±0.58	30.29±0.80	30.26±0.43	31.25±1.78	31.60±1.42	32.06±0.10	32.86±1.89	46.20±0.11
	AdaNet [24]	20.45±0.82	27.10±0.46	28.50±0.40	30.83±0.11	30.17±0.10	31.86±1.21	31.46±0.90	32.82±1.72	32.98±0.40	32.89±1.28	47.22±0.18
	SGC [25]	22.27±0.80	29.38±0.12	30.19±1.28	31.68±0.60	31.20±0.83	32.26±0.11	32.50±1.44	32.75±1.16	32.57±0.46	33.05±0.19	48.28±2.04
	MixHop [26]	23.17±0.33	28.46±1.38	27.65±0.85	31.30±1.46	31.58±0.52	32.74±1.50	32.23±1.70	33.60±0.48	34.50±0.67	34.92±0.44	47.18±0.30
	MobileGCN	16.95±0.36	25.20±1.31	27.69±1.04	29.08±1.02	29.55±0.77	30.24±0.79	32.13±0.62	31.43±0.79	31.85±0.45	32.01±0.53	47.20±0.26
	MobilePPNP	26.17±0.62	30.52±0.64	33.20±1.11	34.79±0.68	35.39±1.36	36.17±0.56	37.23±1.12	36.62±0.88	37.83±1.19	37.24±0.73	47.20±0.70
	MobileAPPNP	18.66±0.91	26.16±0.62	28.33±0.82	30.41±0.96	30.64±0.59	31.20±0.66	32.86±1.11	31.84±0.52	32.49±0.43	32.02±0.99	47.45±0.43
	LGC-GCN	20.01±0.42	27.13±1.24	29.32±0.82	30.24±0.15	30.80±0.24	31.60±0.61	32.56±0.99	32.66±0.94	33.31±1.37	33.12±0.71	47.92±0.49
	LGC-PPNP	**28.97±0.95**	32.37±0.61	33.96±0.74	35.30±0.55	35.68±0.92	37.09±1.58	36.10±0.30	36.91±0.87	37.13±0.67	37.55±0.42	49.31±0.41
	LGC-APPNP	23.65±0.38	**33.23±0.43**	**35.34±0.43**	**37.33±0.61**	**36.59±0.21**	**38.73±0.56**	**41.13±0.24**	**41.31±0.19**	**41.48±0.46**	**41.30±0.24**	59.11±0.28

Table 5
Evaluation results of node classification on the CORA dataset.

Metric	Model	The percentage of the top node features (The number of the top node features)										
		1%(14)	2%(29)	3%(43)	4%(57)	5%(72)	6%(86)	7%(100)	8%(115)	9%(129)	10%(143)	100%(1433)
Accuracy	GCN [14]	31.65±0.25	52.21±0.13	53.19±0.12	58.34±0.51	57.20±0.22	57.95±0.34	60.80±0.14	61.72±0.11	64.16±0.41	66.36±0.37	80.67±0.10
	GCN† [14]	25.90±0.70	46.00±0.10	45.85±0.15	53.80±0.30	56.25±0.15	65.65±0.85	59.90±0.30	60.35±0.25	63.25±0.55	66.40±0.60	81.54±0.23
	PPNP [15]	38.92±0.20	61.96±0.65	60.53±0.22	70.30±0.08	69.92±0.40	70.28±0.14	70.89±0.47	71.91±0.33	74.50±0.20	75.74±0.61	81.42±0.24
	PPNP† [15]	18.90±1.20	40.70±0.10	46.40±0.20	56.60±0.40	58.60±0.10	59.90±0.30	68.30±0.30	68.60±0.20	71.10±0.30	74.80±0.20	82.45±0.46
	APPNP [15]	30.72±0.12	51.90±0.16	51.76±0.18	57.83±0.46	56.46±1.10	58.48±0.65	59.90±0.22	61.53±0.03	63.60±0.35	66.28±0.40	80.65±0.30
	APPNP† [15]	25.30±2.20	45.40±0.30	45.70±0.30	52.70±0.30	55.55±0.25	55.35±0.25	58.50±0.40	58.90±0.30	61.80±0.30	65.75±0.55	81.98±0.32
	GAT [10]	32.00±0.70	54.00±0.52	53.81±0.46	60.12±0.64	61.60±0.34	63.03±0.30	64.82±0.50	65.52±0.47	69.51±0.42	71.26±0.28	82.61±0.40
	LNet [24]	34.20±0.52	47.03±0.20	50.25±0.40	58.24±0.35	60.05±0.63	60.89±0.36	64.20±0.10	66.30±0.35	68.52±0.66	69.62±0.40	81.23±0.64
	AdaNet [24]	36.12±0.42	50.20±0.15	56.58±0.65	61.04±0.12	63.10±0.40	65.28±0.40	66.55±0.68	68.16±0.28	69.70±0.24	70.51±0.36	82.30±0.20
	SGC [25]	35.08±0.50	48.23±0.26	58.61±0.48	63.25±0.05	65.05±0.04	67.52±0.20	68.76±0.26	70.10±0.15	69.27±0.18	72.00±0.06	81.52±0.17
	MixHop [26]	48.21±0.36	57.82±0.10	60.46±0.26	65.42±0.18	66.56±0.64	67.77±0.35	67.35±0.17	68.96±0.55	67.30±0.20	71.40±0.36	81.36±0.10
	MobileGCN	39.35±0.45	58.35±0.35	59.70±0.50	66.05±0.65	64.20±0.60	65.10±1.00	66.00±0.60	67.25±0.75	69.90±0.50	71.75±0.65	82.30±0.40
	MobilePPNP	62.32±0.10	68.27±0.84	69.57±0.69	76.13±0.30	73.98±0.42	74.34±0.63	74.85±0.46	76.44±0.10	**79.26±0.80**	**78.14±0.21**	82.39±0.26
	MobileAPPNP	51.23±0.23	64.54±0.33	62.81±0.61	68.13±0.58	66.56±0.74	67.92±0.15	68.41±0.66	67.90±0.20	72.30±0.63	72.34±0.45	**82.62±0.51**
	LGC-GCN	53.35±0.75	64.30±0.60	64.30±0.30	70.05±0.55	70.25±0.95	69.10±0.50	68.90±0.90	69.65±0.55	72.40±0.40	73.30±0.70	80.15±0.35
	LGC-PPNP	**71.80±0.22**	**76.46±0.50**	**76.62±0.21**	**77.46±0.12**	**77.13±0.30**	**77.28±0.50**	**77.57±0.37**	**76.80±0.44**	77.30±0.32	76.81±0.40	80.00±0.30
	LGC-APPNP	47.56±0.45	64.23±0.60	63.52±0.16	66.98±0.73	66.12±0.40	68.10±0.35	67.84±0.26	69.37±0.49	70.39±0.20	71.95±0.11	79.00±0.10
Macro-F1	GCN [14]	33.68±0.27	37.57±0.66	36.84±0.22	39.28±0.32	39.58±0.50	40.72±0.63	41.52±0.65	41.71±0.75	42.69±0.72	44.55±0.24	52.47±0.78
	GCN† [14]	30.18±0.04	37.36±0.96	35.49±0.16	38.16±0.45	39.68±0.62	41.34±0.86	41.92±1.00	42.19±0.63	44.32±0.19	45.07±0.53	54.63±0.40
	PPNP [15]	38.06±0.43	39.78±0.40	40.28±0.16	41.34±0.57	43.72±1.05	43.73±0.53	44.62±0.62	45.78±1.74	45.53±0.56	47.26±0.58	51.48±0.38
	PPNP† [15]	23.42±0.25	34.90±0.35	35.78±0.05	38.60±1.20	39.96±1.37	40.50±1.16	42.76±1.22	42.94±1.54	43.90±0.80	44.64±0.29	56.25±1.65
	APPNP [15]	33.07±0.28	36.70±0.64	36.19±0.55	39.06±0.88	39.60±0.53	41.10±0.63	41.26±0.16	41.76±0.72	42.69±0.45	44.79±0.77	52.38±0.68
	APPNP† [15]	30.13±0.07	37.09±0.39	35.97±0.64	37.28±0.13	39.59±1.02	40.14±0.33	41.79±0.50	42.87±0.40	44.12±0.44	44.57±0.28	55.56±0.48
	GAT [10]	36.17±1.03	46.42±0.70	46.30±0.81	51.61±0.30	52.26±0.17	55.07±0.35	56.61±0.79	58.57±0.45	60.53±0.61	63.92±0.45	74.65±0.21
	LNet [24]	39.56±0.50	42.44±1.52	42.05±0.56	43.23±0.20	46.60±0.50	48.69±1.05	48.68±0.07	50.25±0.46	52.30±0.56	52.48±0.80	56.25±0.65
	AdaNet [24]	38.56±0.64	42.48±0.63	43.30±0.84	44.00±0.08	46.50±0.05	49.60±0.63	50.69±1.20	52.23±0.34	53.42±0.82	56.25±0.61	57.04±0.46
	SGC [25]	39.67±0.33	44.30±0.11	44.69±0.06	46.14±0.87	48.11±0.24	50.99±0.04	52.19±0.60	52.06±1.86	52.26±0.44	51.49±0.58	57.70±0.43
	MixHop [26]	37.80±0.25	39.68±0.22	40.00±0.50	42.20±0.14	42.50±0.58	45.58±1.25	46.10±0.42	48.70±1.45	49.52±1.09	48.65±0.40	52.86±1.05
	MobileGCN	36.59±0.65	38.42±0.64	38.40±0.55	40.41±0.83	41.09±0.23	42.36±0.69	42.32±0.44	44.07±1.39	43.70±0.34	45.60±0.76	54.43±1.15
	MobilePPNP	**46.81±1.47**	44.24±2.60	45.30±1.17	46.29±0.72	46.57±1.22	46.47±1.16	45.80±1.19	47.04±0.52	48.25±0.64	49.46±1.35	55.10±0.98
	MobileAPPNP	41.42±0.90	41.53±1.19	40.93±0.80	42.52±0.57	42.34±0.66	43.96±0.92	43.08±0.51	44.51±1.23	45.30±0.32	46.17±0.95	54.50±0.59
	LGC-GCN	40.12±0.22	42.26±0.42	42.95±0.66	44.36±0.34	45.79±1.15	45.44±0.88	46.04±0.42	46.80±0.56	48.28±0.97	48.74±0.95	54.58±0.65
	LGC-PPNP	46.43±0.33	47.37±0.68	48.23±0.56	48.83±0.99	49.45±0.66	49.67±0.64	50.59±1.27	49.70±0.88	50.75±0.94	51.28±0.42	55.72±1.14
	LGC-APPNP	45.61±0.25	**53.22±0.92**	**53.45±0.41**	**56.03±0.37**	**57.76±0.11**	**59.98±0.16**	**62.15±0.44**	**62.12±0.26**	**63.10±0.07**	**65.58±0.08**	**74.98±0.09**
MCC	GCN [14]	20.21±0.46	27.95±1.30	27.17±0.33	31.14±0.41	31.60±0.40	33.19±0.57	34.73±0.89	35.29±1.13	36.79±1.20	39.66±0.17	51.06±0.91
	GCN† [14]	15.20±0.62	28.76±0.94	26.99±0.73	29.06±0.46	31.63±0.48	32.68±1.04	34.44±1.56	35.00±0.90	37.57±0.34	38.52±1.03	52.12±1.82
	PPNP [15]	27.30±0.53	30.98±0.47	31.87±0.32	34.37±0.64	36.28±0.55	36.46±0.49	37.02±0.63	37.76±0.45	40.07±0.63	42.05±0.44	49.09±0.55
	PPNP† [15]	6.79±0.50	23.24±1.00	25.07±1.31	28.74±1.48	30.41±1.28	30.24±0.65	33.87±1.87	34.10±1.62	34.84±0.89	36.99±0.45	49.86±1.72
	APPNP [15]	19.39±0.52	27.02±1.39	25.89±1.15	30.33±1.19	30.82±0.48	33.10±0.51	33.51±0.67	33.75±0.79	36.03±0.61	38.69±0.63	50.85±0.63
	APPNP† [15]	14.91±0.51	28.32±0.43	26.82±0.86	28.31±0.40	31.78±0.93	32.03±0.40	34.67±0.33	35.69±0.42	37.50±0.67	38.58±0.88	51.46±0.38
	GAT [10]	24.09±1.23	38.63±1.07	38.63±0.87	45.73±0.40	46.07±0.27	49.42±0.41	51.31±0.80	53.51±0.61	56.27±0.59	60.08±0.36	71.02±0.24
	LNet [24]	20.65±1.54	25.61±0.85	26.36±0.46	28.74±1.40	28.49±0.80	30.08±1.83	30.12±0.40	31.43±1.68	32.84±0.45	34.51±1.46	46.35±0.85
	AdaNet [24]	22.68±1.06	26.85±0.84	28.44±0.69	30.21±0.22	31.21±0.70	33.40±0.86	35.58±0.66	35.40±1.26	35.68±1.08	38.68±1.47	48.82±0.63
	SGC [25]	21.25±0.45	28.54±0.68	30.06±1.42	30.88±1.95	33.00±0.85	33.20±0.40	36.98±0.23	38.86±2.40	40.19±0.55	41.87±0.54	40.07±1.24
	MixHop [26]	22.04±0.78	25.80±0.40	27.47±1.40	30.58±0.50	33.74±1.06	37.46±0.82	38.93±1.50	39.25±0.40	40.28±0.60	40.86±0.40	50.10±0.52
	MobileGCN	26.84±0.64	29.62±0.76	30.77±0.92	33.78±0.74	34.60±0.86	36.15±0.34	35.56±0.85	37.75±0.57	39.32±0.46	40.57±0.50	52.63±0.68
	MobilePPNP	39.41±1.83	37.37±2.52	39.00±1.82	40.37±1.58	41.17±1.69	40.54±1.12	40.53±1.58	41.51±0.85	43.70±0.80	44.79±1.12	53.10±0.82
	MobileAPPNP	32.47±1.76	33.67±1.17	33.49±1.52	35.56±1.59	36.30±1.22	37.80±0.91	36.92±1.14	38.11±1.21	40.36±0.32	41.29±0.76	53.00±0.87
	LGC-GCN	31.80±0.41	35.76±0.49	37.13±0.85	39.70±0.42	41.10±1.22	40.82±1.04	41.44±0.92	42.12±1.17	44.54±0.89	44.70±0.84	52.85±0.74
	LGC-PPNP	**40.66±0.64**	41.78±0.92	43.45±0.70	44.35±1.22	45.07±0.90	45.43±0.47	46.15±1.41	44.89±0.70	46.47±1.09	46.93±0.69	54.17±1.05
	LGC-APPNP	35.78±0.22	**46.81±0.97**	**47.14±0.86**	**49.63±0.22**	**52.38±0.30**	**54.52±0.21**	**56.73±0.73**	**56.81±0.48**	**58.09±0.33**	**61.46±0.13**	**71.32±0.21**

Table 6
Evaluation results of node classification on the PUBMED dataset.

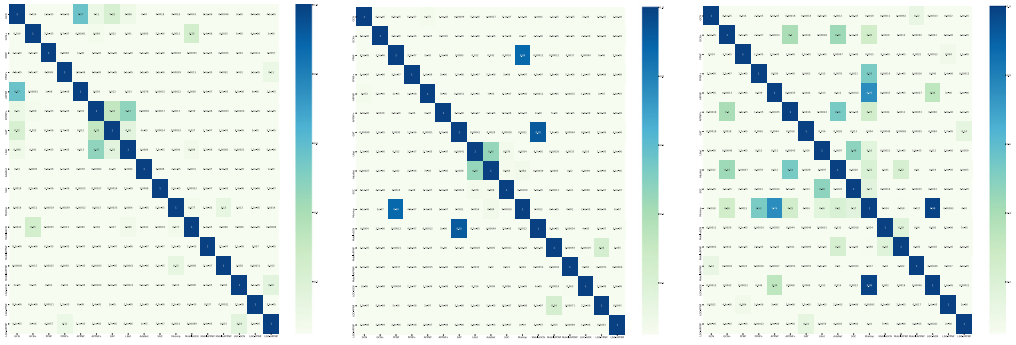
Metric	Model	The percentage of the top node features (The number of the top node features)										
		1%(5)	2%(10)	3%(15)	4%(20)	5%(25)	6%(30)	7%(35)	8%(40)	9%(45)	10%(50)	100%(500)
Accuracy	GCN [14]	59.95±0.35	63.55±0.75	64.20±0.90	68.85±0.95	69.70±0.90	71.50±1.00	70.90±0.30	69.65±0.25	71.15±0.45	70.50±0.20	78.56±0.54
	GCN† [14]	50.60±0.10	52.40±0.40	63.10±0.20	69.30±0.10	70.35±0.25	71.40±0.26	71.15±0.15	70.45±0.35	71.50±0.10	72.00±0.10	78.89±0.36
	PPNP [15]	64.30±1.20	71.15±0.25	71.25±0.35	72.90±0.32	73.00±0.80	73.20±0.61	71.90±0.55	72.10±0.50	72.60±0.42	72.45±0.05	78.89±0.46
	PPNP† [15]	50.15±0.05	52.50±0.20	63.30±0.50	75.25±0.05	76.55±0.25	75.50±0.04	75.40±0.10	74.95±0.15	75.05±0.05	75.15±0.15	79.98±0.30
	APPNP [15]	51.10±1.30	52.00±2.10	56.05±0.55	66.30±0.60	66.40±0.30	69.15±0.25	66.50±0.10	66.05±0.25	67.55±0.25	68.00±0.10	78.25±0.54
	APPNP† [15]	50.55±0.15	53.30±0.60	61.70±0.30	69.10±0.10	70.75±0.15	72.05±0.05	71.25±0.15	70.05±0.05	71.30±0.20	71.50±0.40	78.80±0.42
	GAT [10]	61.92±0.23	60.00±0.50	61.43±0.22	66.83±0.20	67.30±0.20	68.15±0.13	69.52±0.82	69.40±1.20	71.66±0.68	71.52±0.18	79.69±0.28
	LNet [24]	56.20±0.10	57.20±0.30	61.52±0.28	68.25±0.12	69.00±0.40	70.08±0.55	70.86±0.30	71.25±0.15	71.68±0.10	71.16±0.25	78.50±0.39
	AdaNet [24]	58.12±0.24	60.14±0.20	63.18±0.20	67.26±0.17	69.15±0.24	70.25±0.30	70.54±0.14	71.52±0.40	72.42±0.11	72.20±0.45	78.16±0.43
	SGC [25]	61.50±0.20	65.02±0.10	66.76±0.42	68.25±0.44	71.78±0.67	72.43±0.10	72.40±0.50	72.48±0.55	73.84±0.20	73.06±0.50	77.12±0.18
	MixHop [26]	63.42±0.20	65.86±0.25	68.62±0.26	70.16±0.30	71.48±0.16	71.50±0.04	72.70±0.51	73.15±0.06	74.59±0.52	74.12±0.48	78.86±0.04
	MobileGCN	51.65±0.15	55.25±0.45	63.40±0.10	69.85±0.35	70.55±0.15	71.00±0.50	72.25±0.45	70.90±0.20	70.90±0.50	72.05±0.15	77.63±0.56
	MobilePPNP	67.82±0.27	70.42±0.56	70.68±0.35	76.54±0.21	**77.70±0.86**	76.56±0.50	77.16±0.10	76.52±0.46	76.68±0.40	77.00±0.16	**80.25±0.78**
	MobileAPPNP	61.52±0.62	62.10±0.25	64.96±1.23	71.45±0.34	72.65±0.24	72.38±0.60	73.25±0.10	72.76±0.86	72.54±0.40	72.98±0.80	78.65±0.40
	LGC-GCN	52.85±0.50	57.45±0.55	63.15±0.45	69.25±0.55	69.95±0.35	71.25±0.65	72.60±0.50	71.95±0.25	71.75±0.52	70.75±0.15	78.00±0.64
	LGC-PPNP	**72.56±0.68**	**74.83±0.15**	**74.40±0.83**	**78.00±0.36**	77.40±0.52	**78.22±0.45**	**77.90±0.20**	**77.48±0.60**	**77.35±0.82**	**77.38±0.64**	79.96±0.54
	LGC-APPNP	62.15±0.46	62.75±0.60	66.35±0.28	70.16±0.86	69.72±0.46	71.78±1.45	70.96±0.41	70.62±0.28	71.25±0.68	71.10±0.48	79.00±0.40
Macro-F1	GCN [14]	66.84±0.26	67.82±0.56	67.86±0.23	69.32±0.64	69.31±0.52	68.43±0.05	67.90±0.54	67.21±0.33	67.40±0.16	67.44±0.18	71.56±0.23
	GCN† [14]	64.70±0.02	66.00±0.03	67.08±0.13	68.58±0.23	68.84±0.02	68.58±0.19	68.09±0.28	67.72±0.15	67.76±0.18	67.96±0.02	71.95±0.06
	PPNP [15]	67.39±0.28	68.43±0.30	68.33±0.16	68.44±0.09	68.51±0.18	68.60±0.13	66.82±0.09	66.86±0.11	66.71±0.09	67.51±0.03	71.98±0.20
	PPNP† [15]	63.93±0.08	65.71±0.01	67.26±0.06	69.45±0.06	69.48±0.10	69.48±0.14	69.48±0.20	69.19±0.14	69.43±0.18	69.49±0.07	72.08±0.14
	APPNP [15]	64.14±0.07	64.51±0.04	65.58±0.22	68.02±0.21	68.07±0.35	67.17±0.11	66.62±0.30	65.96±0.12	66.21±0.35	66.63±0.34	70.45±0.21
	APPNP† [15]	64.74±0.13	66.07±0.04	66.74±0.09	68.54±0.14	69.11±0.11	68.29±0.09	68.56±0.08	67.86±0.13	68.00±0.27	68.32±0.19	71.08±0.62
	GAT [10]	62.30±0.40	67.56±0.69	69.23±0.80	69.66±0.58	69.32±0.30	69.25±0.52	69.56±0.20	69.86±0.82	69.35±0.69	68.99±0.63	71.36±0.65
	LNet [24]	63.50±0.25	65.10±0.40	67.25±0.10	68.20±0.14	68.24±0.10	68.20±0.14	69.80±0.40	68.18±0.05	68.45±0.28	67.56±0.40	70.04±0.08
	AdaNet [24]	64.85±0.40	66.74±0.15	68.42±0.25	69.21±0.26	69.28±0.10	68.86±0.25	68.48±0.08	69.46±0.18	68.89±0.45	69.14±0.60	70.62±0.54
	SGC [25]	63.64±0.07	66.28±0.14	68.42±0.84	69.10±0.25	69.12±0.08	68.40±0.12	67.26±0.17	69.85±0.16	69.25±0.08	69.24±0.12	71.42±0.10
	MixHop [26]	64.09±0.42	65.68±0.11	68.25±0.08	68.27±0.10	68.74±0.20	68.18±0.26	69.11±0.20	68.54±0.15	68.10±0.14	69.02±0.20	71.50±0.10
	MobileGCN	65.34±0.20	65.86±0.15	66.72±0.34	67.07±0.14	67.10±0.22	66.74±0.38	66.40±0.28	65.91±0.16	65.93±0.23	66.61±0.07	71.43±0.85
	MobilePPNP	65.26±0.12	67.50±0.23	68.23±0.72	69.75±0.20	69.82±0.30	69.60±0.11	69.63±0.50	69.75±0.30	69.82±0.16	69.33±0.18	**72.30±0.15**
	MobileAPPNP	65.86±0.20	67.65±0.26	68.30±0.12	68.88±0.36	69.42±0.32	69.13±0.16	69.78±0.25	69.65±0.28	69.13±0.68	69.05±0.20	70.18±0.56
	LGC-GCN	64.38±0.21	66.37±0.05	66.65±0.50	66.94±0.11	66.58±0.20	66.79±0.23	66.92±0.52	66.32±0.24	66.33±0.25	66.57±0.32	71.50±0.36
	LGC-PPNP	**67.63±0.25**	68.52±0.50	69.36±0.20	69.27±0.23	69.86±0.26	69.33±0.06	70.02±0.16	70.07±0.25	69.92±0.80	**70.16±0.26**	72.06±0.60
	LGC-APPNP	62.85±0.30	**68.68±0.21**	**69.73±0.52**	**70.36±0.50**	**70.62±0.22**	**70.78±0.52**	**70.55±0.24**	**70.63±0.30**	**70.46±0.20**	70.06±0.34	72.14±0.50
MCC	GCN [14]	44.28±0.49	46.26±0.40	46.61±0.37	50.12±0.12	50.65±1.34	49.01±0.15	48.22±0.37	47.65±0.42	48.19±0.21	48.62±0.40	50.32±0.65
	GCN† [14]	41.28±0.04	44.02±0.02	45.85±0.21	49.38±0.46	50.06±0.11	50.00±0.26	49.44±0.48	48.85±0.29	49.04±0.12	49.31±0.05	50.68±0.35
	PPNP [15]	44.41±0.37	47.62±0.62	47.40±0.19	47.47±0.27	47.43±0.52	47.93±0.32	45.95±0.26	46.16±0.25	46.11±0.27	47.33±0.16	51.68±0.13
	PPNP† [15]	40.15±0.13	43.02±0.03	45.51±0.09	50.18±0.06	**50.80±0.18**	50.03±0.13	50.14±0.16	50.81±0.16	50.99±0.01	50.35±0.16	52.08±0.21
	APPNP [15]	40.52±0.34	41.16±0.13	42.48±0.25	47.14±0.35	47.18±0.25	46.48±0.55	46.11±0.11	45.26±0.30	45.56±0.26	46.00±0.23	48.04±1.25
	APPNP† [15]	41.43±0.23	43.96±0.09	44.84±0.10	48.65±0.18	49.46±0.14	48.91±0.15	49.58±0.04	48.64±0.17	48.74±0.19	49.18±0.47	49.56±0.26
	GAT [10]	42.56±0.65	48.65±0.42	**49.63±0.30**	50.03±0.12	50.02±0.14	49.87±0.46	50.13±0.25	50.36±0.14	49.76±0.80	49.24±0.30	51.90±0.40
	LNet [24]	42.21±0.43	44.26±0.72	45.27±0.42	47.29±0.32	47.62±0.15	47.65±0.23	48.52±0.15	48.62±1.24	48.26±0.43	47.50±1.40	50.06±0.20
	AdaNet [24]	43.25±0.40	46.23±0.15	46.10±0.86	46.65±0.15	46.28±0.40	47.25±1.12	47.21±0.55	47.32±0.24	48.32±0.04	48.43±1.50	50.46±0.06
	SGC [25]	44.72±0.25	47.20±1.22	47.30±1.63	47.31±0.25	47.20±0.31	48.35±0.26	48.28±1.20	48.57±0.12	47.75±0.40	48.15±0.25	51.09±0.14
	MixHop [26]	43.26±0.25	46.28±0.15	47.26±0.25	47.15±0.06	48.23±0.32	48.74±0.52	48.38±1.10	49.20±0.24	48.26±0.14	48.12±0.20	51.12±0.25
	MobileGCN	42.13±0.25	43.69±0.17	45.95±0.06	46.56±0.21	47.21±0.04	46.62±0.11	47.11±0.13	46.18±0.82	46.52±0.14	47.01±0.02	50.21±0.04
	MobilePPNP	41.94±0.20	47.32±0.23	48.82±0.24	49.80±1.45	49.56±0.42	49.25±0.41	50.50±0.14	50.42±1.58	50.25±0.24	49.43±0.10	51.42±0.20
	MobileAPPNP	40.56±0.30	43.54±0.75	45.56±0.14	46.52±0.12	47.89±0.32	47.85±1.20	48.56±0.21	49.23±0.22	49.52±0.04	49.20±0.20	50.01±0.20
	LGC-GCN	40.69±0.29	46.14±0.19	46.55±0.26	46.94±0.14	47.34±0.22	48.44±0.24	48.97±0.40	47.77±1.56	48.09±0.32	47.91±0.85	50.06±0.42
	LGC-PPNP	45.56±0.12	48.25±0.54	49.52±0.20	49.90±1.45	50.02±0.14	50.15±0.24	50.24±0.82	50.45±0.10	50.00±0.04	49.52±0.36	**52.42±0.60**
	LGC-APPNP	**46.58±0.36**	**48.76±0.25**	49.56±0.20	**50.21±0.30**	50.36±0.43	**50.23±0.14**	**50.68±0.25**	**50.98±0.13**	**51.24±0.20**	**51.03±0.45**	51.56±0.28

Table 7
Evaluation results of node classification on the Wikipedia dataset.

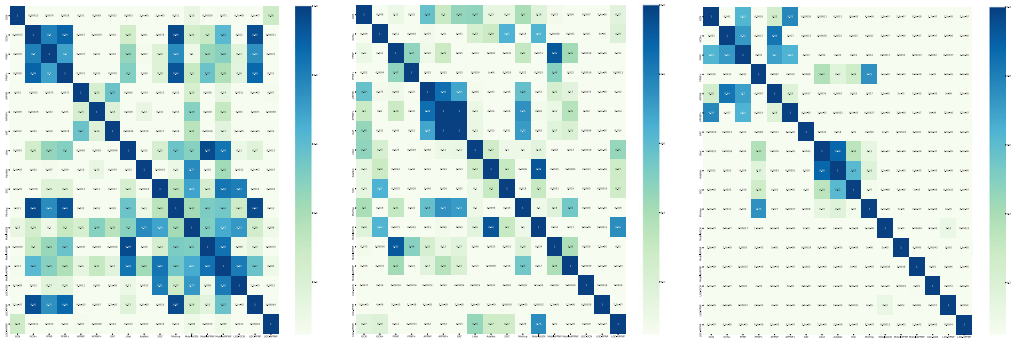
Metric	Model	The percentage of the top node features (The number of the top node features)										
		1%(50)	2%(100)	3%(149)	4%(199)	5%(249)	6%(298)	7%(348)	8%(398)	9%(448)	10%(497)	100%(4973)
Accuracy	GCN [14]	18.26±6.05	12.87±6.11	15.30±6.02	16.74±9.49	20.18±8.86	20.72±5.39	24.61±4.67	37.37±5.81	32.63±6.23	37.63±6.08	68.20±0.30
	GCN† [14]	24.79±2.29	12.16±4.13	18.41±5.42	23.95±7.13	20.48±3.29	22.69±2.87	29.34±5.45	37.63±2.78	46.92±2.07	48.71±5.78	66.85±0.42
	PPNP [15]	21.47±4.34	8.68±4.97	13.56±4.70	13.32±6.26	13.26±7.34	11.02±3.95	16.38±5.30	24.76±6.50	27.75±4.94	19.25±7.63	41.95±0.33
	PPNP† [15]	26.26±0.45	8.59±3.38	14.62±5.02	14.20±5.80	15.65±4.38	11.56±8.96	15.27±6.20	24.05±8.09	28.50±3.65	21.56±9.03	43.22±0.14
	APPNP [15]	21.59±3.08	15.99±9.40	14.70±3.14	25.33±7.43	20.78±4.49	20.45±5.60	29.73±3.56	33.44±6.62	43.26±4.70	44.46±4.76	70.69±0.39
	APPNP† [15]	20.63±5.89	14.20±6.30	14.30±5.27	24.12±6.43	21.36±3.75	20.56±5.56	30.68±2.40	35.21±4.40	46.37±5.28	46.52±4.21	**71.25±0.82**
	GAT [10]	26.68±3.57	13.08±5.30	18.47±6.32	25.99±3.65	22.07±5.18	20.09±9.19	28.38±7.90	41.44±7.25	39.85±4.76	47.63±5.24	66.56±0.75
	LNet [24]	25.08±2.20	16.15±8.24	15.22±6.19	28.10±4.20	23.06±5.18	22.18±6.21	29.80±1.09	36.27±4.20	42.68±5.00	44.18±1.78	61.20±0.59
	AdaNet [24]	24.21±4.42	15.41±5.02	16.81±7.56	32.21±4.28	32.18±6.14	33.78±6.46	36.45±4.00	42.25±5.28	43.26±3.29	45.02±4.18	62.61±0.15
	SGC [25]	20.08±5.20	14.12±2.28	15.76±2.08	13.68±4.25	20.50±6.85	22.42±5.28	38.41±1.27	40.84±3.68	42.40±4.23	45.10±4.52	58.26±0.50
	MixHop [26]	21.24±4.28	13.08±6.04	13.75±2.35	16.61±4.20	22.36±4.12	25.05±2.44	34.72±4.12	40.51±3.12	39.29±3.26	43.28±5.28	59.47±0.64
	MobileGCN	16.56±2.49	11.32±4.37	24.43±12.69	28.29±13.26	31.74±15.63	38.11±10.93	**61.47±1.83**	**63.02±3.20**	**67.54±0.72**	**67.57±0.69**	64.55±0.42
	MobilePPNP	19.94±2.46	16.59±3.59	15.06±3.38	18.86±5.03	16.80±5.48	18.35±2.60	15.15±5.33	17.46±5.00	24.43±11.68	17.01±5.39	66.11±0.48
	MobileAPPNP	16.14±1.29	18.50±6.89	20.00±5.33	27.81±12.07	20.15±7.63	27.54±17.72	36.77±23.29	41.17±20.69	61.86±2.46	51.11±14.70	66.44±0.33
	LGC-GCN	27.72±0.84	31.26±0.36	35.36±0.81	35.90±0.39	36.71±0.48	38.20±0.54	43.56±0.63	48.71±0.33	52.75±0.18	51.98±0.24	55.54±0.09
	LGC-PPNP	18.02±0.18	**48.77±0.69**	**50.00±0.72**	**51.80±0.24**	**52.69±0.60**	**53.23±0.78**	56.14±0.51	57.31±0.36	60.00±0.60	60.24±0.24	60.81±0.21
	LGC-APPNP	**29.10±4.07**	25.57±2.51	29.55±1.35	30.42±1.68	28.26±0.36	27.46±1.23	33.53±0.48	39.52±0.12	36.29±0.24	37.34±0.15	58.95±0.21
Macro-F1	GCN [14]	15.89±0.17	13.67±2.41	15.25±2.23	14.78±2.02	17.15±2.78	18.04±2.21	16.27±1.16	20.19±1.76	18.31±1.67	18.41±1.24	20.02±0.66
	GCN† [14]	12.19±0.63	16.60±1.59	16.54±1.02	16.28±1.64	17.89±0.31	18.76±0.56	18.14±0.43	17.70±0.27	17.43±1.52	18.42±1.36	21.06±0.25
	PPNP [15]	12.09±1.69	11.07±0.99	11.47±1.48	13.15±3.29	13.95±2.40	13.92±3.71	14.27±2.67	14.70±2.05	15.79±3.97	18.11±1.75	16.69±0.89
	PPNP† [15]	12.91±1.70	13.75±2.17	14.50±1.69	14.89±2.40	15.30±1.42	15.45±0.87	16.22±1.55	15.91±0.29	17.24±0.22	18.01±0.21	18.54±0.22
	APPNP [15]	15.23±0.64	14.60±1.43	14.53±1.70	16.92±2.88	18.85±2.77	19.19±2.05	19.12±0.79	18.51±1.31	18.72±1.61	18.91±2.43	20.29±0.61
	APPNP† [15]	14.48±0.56	15.60±2.02	16.84±1.22	17.23±1.58	19.70±0.85	19.20±0.42	19.80±1.45	19.90±0.25	19.58±1.65	19.88±0.55	22.08±0.26
	GAT [10]	15.21±0.51	15.48±2.44	17.24±1.10	18.13±1.03	19.76±1.14	19.22±1.55	16.41±1.73	17.69±2.12	19.39±1.92	18.68±1.80	27.33±0.43
	LNet [24]	12.25±1.40	15.28±1.13	14.15±1.15	14.25±1.24	14.34±0.12	16.68±0.20	16.69±0.20	16.30±0.10	17.50±0.23	18.20±1.42	21.10±0.18
	AdaNet [24]	13.96±0.25	14.24±0.20	14.26±1.03	15.01±0.20	15.10±1.12	16.52±0.60	16.27±1.04	16.40±0.27	15.29±0.08	17.28±0.46	20.20±0.61
	SGC [25]	13.06±0.28	13.18±1.05	14.26±0.55	14.12±0.11	13.54±1.07	15.25±0.87	16.68±1.17	16.20±0.20	17.08±1.08	18.44±1.10	22.28±1.17
	MixHop [26]	14.25±2.83	15.27±0.56	16.63±1.08	16.29±0.15	16.70±0.26	16.28±0.40	18.12±1.20	17.04±0.43	18.12±1.20	19.22±0.44	21.58±0.16
	MobileGCN	12.20±3.04	11.98±2.42	12.48±2.29	14.06±2.60	12.91±2.83	14.79±0.50	15.84±0.29	16.33±0.37	16.71±0.34	16.66±0.12	18.87±0.57
	MobilePPNP	12.04±2.08	9.63±1.36	10.58±1.41	11.08±1.85	10.88±1.35	10.43±1.17	11.85±3.13	11.52±1.32	12.72±2.29	13.37±1.33	19.58±0.57
	MobileAPPNP	12.27±1.80	12.74±1.85	15.40±0.66	13.63±2.13	14.60±0.44	12.78±2.05	14.56±0.68	15.11±1.40	16.49±1.77	15.59±0.48	18.96±0.50
	LGC-GCN	13.04±0.08	14.00±0.22	14.99±0.20	15.29±0.33	15.70±0.21	16.06±0.28	15.96±0.14	16.73±0.11	17.27±0.22	17.30±0.21	20.01±0.22
	LGC-PPNP	11.86±0.09	13.38±0.14	14.52±0.25	14.99±0.44	15.17±0.14	15.63±0.27	15.85±0.18	16.87±0.41	16.83±0.24	16.46±0.26	19.70±0.77
	LGC-APPNP	**16.19±0.72**	**17.94±0.46**	**19.24±0.62**	**19.26±0.80**	**20.15±0.86**	**21.30±0.91**	**21.91±1.06**	**22.30±0.24**	**23.00±0.52**	**23.37±1.03**	**32.35±0.27**
MCC	GCN [14]	14.11±5.26	13.44±4.02	14.60±2.90	14.59±5.18	17.57±2.33	18.14±2.05	20.89±0.99	22.41±1.41	23.04±0.22	23.01±0.21	26.18±0.38
	GCN† [14]	18.84±1.36	18.19±1.50	19.88±1.44	19.89±1.30	20.73±0.68	20.64±0.37	21.53±0.39	22.54±0.10	23.80±0.02	23.65±0.06	27.45±0.24
	PPNP [15]	14.76±3.99	15.37±1.65	16.61±2.00	16.58±1.35	16.17±1.58	15.21±2.01	16.94±2.19	17.56±1.74	18.39±3.79	19.53±2.23	22.53±0.24
	PPNP† [15]	19.58±1.06	17.71±0.79	17.86±2.50	18.45±1.40	19.40±2.14	20.65±0.58	20.06±0.48	21.35±0.38	22.85±1.52	23.24±1.30	25.86±0.20
	APPNP [15]	14.20±2.88	14.94±1.83	14.61±3.95	18.95±1.72	18.99±0.51	17.89±1.51	21.20±0.92	22.59±0.18	23.83±0.06	23.49±0.06	26.46±0.45
	APPNP† [15]	17.85±1.45	17.56±0.48	18.63±0.58	19.42±1.46	19.47±0.25	20.52±0.17	22.87±0.86	23.45±0.60	24.35±0.40	24.87±1.85	28.63±0.41
	GAT [10]	20.60±0.47	14.27±3.41	16.76±2.87	17.74±1.83	18.05±2.94	17.57±2.41	21.51±1.57	22.17±0.47	24.15±0.78	24.49±0.43	35.92±0.47
	LNet [24]	16.00±3.64	14.58±2.12	15.10±1.20	16.10±2.52	17.20±1.38	17.28±1.03	18.16±1.27	20.45±1.24	22.75±2.06	22.51±2.81	24.18±0.66
	AdaNet [24]	17.10±2.26	16.20±1.27	16.15±1.54	17.10±2.06	18.18±1.46	18.00±1.26	19.25±1.36	21.36±2.04	22.38±1.44	22.41±1.68	25.63±0.42
	SGC [25]	15.68±2.10	13.22±2.40	14.36±2.13	16.54±1.00	17.42±1.68	18.50±2.15	18.68±1.47	18.24±1.28	20.15±2.06	21.20±1.20	24.91±0.27
	MixHop [26]	16.20±2.20	15.20±1.20	15.68±2.40	16.27±1.68	17.20±1.30	17.70±1.50	19.38±2.06	21.22±2.00	22.25±1.14	22.86±1.07	24.69±0.34
	MobileGCN	12.88±4.21	13.88±4.90	13.40±6.03	14.58±5.56	17.14±2.78	19.18±0.67	21.59±0.64	21.96±1.10	22.61±0.51	22.68±0.24	24.48±0.19
	MobilePPNP	13.53±5.53	11.26±3.95	11.19±1.55	13.82±1.01	15.46±1.54	12.31±2.41	14.47±2.45	13.68±2.72	16.81±1.20	14.70±2.79	25.50±0.50
	MobileAPPNP	12.08±4.08	13.79±3.33	15.53±3.27	15.74±3.91	16.93±2.79	15.83±3.05	18.82±2.13	16.98±4.31	20.25±1.68	20.70±1.23	25.75±0.20
	LGC-GCN	18.79±0.14	20.40±0.13	21.51±0.09	22.27±0.28	22.72±0.20	23.31±0.28	23.65±0.47	24.71±0.36	25.37±0.27	25.22±0.33	25.22±0.40
	LGC-PPNP	18.30±0.03	19.34±0.17	18.58±0.36	18.80±0.32	18.97±0.19	19.12±0.33	19.31±0.24	19.09±0.21	19.65±0.23	19.71±0.42	23.17±0.67
	LGC-APPNP	**21.05±0.29**	**23.01±0.10**	**23.99±0.66**	**24.00±0.18**	**24.63±0.16**	**25.15±0.33**	**26.05±0.60**	**26.44±0.31**	**27.55±0.16**	**27.51±0.34**	**36.31±0.36**

Table 8
Evaluation results of node classification on the Photo dataset.

Metric	Model	The percentage of the top node features (The number of the top node features)										
		1%(7)	2%(15)	3%(22)	4%(30)	5%(37)	6%(45)	7%(52)	8%(60)	9%(67)	10%(75)	100%(745)
Accuracy	GCN [14]	40.01±3.70	32.26±0.88	31.82±2.24	28.20±0.02	37.67±6.69	23.94±0.77	32.95±8.42	36.77±3.46	37.87±5.67	29.98±6.19	86.40±0.11
	GCN† [14]	37.06±1.12	38.14±2.06	34.62±0.39	35.74±2.23	37.54±2.34	34.45±3.93	41.51±2.35	39.15±3.41	39.91±4.15	40.96±3.16	87.24±0.08
	PPNP [15]	31.81±0.32	35.61±5.24	33.33±0.71	31.28±1.79	29.66±4.10	32.55±1.21	35.60±3.04	37.63±5.24	32.42±0.94	31.19±1.43	80.54±0.22
	PPNP† [15]	32.99±0.96	34.71±0.67	34.02±0.26	34.49±0.18	34.69±0.23	34.23±0.18	34.36±0.76	34.09±0.39	33.34±0.18	33.71±0.08	83.87±0.56
	APPNP [15]	40.82±2.08	33.20±4.42	32.86±2.91	24.14±1.78	26.99±5.14	29.35±6.99	36.99±3.58	30.98±3.06	39.12±2.94	38.05±2.57	87.60±0.05
	APPNP† [15]	38.36±4.90	35.80±3.21	34.52±2.49	31.59±5.85	32.98±0.70	30.76±7.99	36.61±5.18	38.31±1.81	39.24±2.89	38.56±0.99	89.25±0.54
	GAT [10]	39.29±2.64	38.07±3.36	40.54±6.83	27.86±0.90	34.73±4.24	41.78±6.36	44.21±7.75	43.78±18.98	47.91±5.51	42.06±9.36	88.94±1.07
	LNet [24]	35.12±2.56	36.86±2.50	34.08±1.20	34.56±2.65	33.28±2.05	36.22±5.10	37.05±2.16	38.19±3.28	39.15±4.05	39.02±2.25	88.03±0.22
	AdaNet [24]	34.34±3.16	36.85±2.56	36.87±5.50	35.25±4.26	35.17±4.20	37.76±3.42	38.12±2.15	38.20±2.26	39.52±3.19	41.12±3.08	89.27±0.05
	SGC [25]	35.28±2.25	33.26±4.21	34.08±2.10	36.54±5.08	37.12±2.18	39.51±4.04	40.21±2.56	40.84±2.66	40.14±2.27	41.12±5.42	88.20±0.25
	MixHop [26]	36.45±4.16	38.12±4.25	37.25±1.40	37.26±4.25	38.16±4.54	39.25±2.57	38.73±2.16	39.25±2.16	39.21±1.20	40.00±2.20	88.26±0.10
	MobileGCN	38.86±2.65	69.59±0.14	71.01±0.59	72.69±0.43	75.34±0.34	77.33±0.30	79.39±0.24	82.25±0.25	82.38±0.35	82.61±0.18	88.16±0.16
	MobilePPNP	33.72±2.19	34.44±1.17	41.26±5.97	72.25±3.29	77.49±2.21	76.01±4.56	81.26±1.01	81.11±0.91	82.21±1.12	84.04±0.18	89.34±0.71
	MobileAPPNP	34.50±1.36	52.06±9.66	51.84±12.32	78.18±1.28	79.98±1.02	81.85±0.30	**83.16±0.61**	**85.97±0.13**	**86.13±0.13**	**86.86±0.17**	**91.94±0.06**
	LGC-GCN	45.93±0.11	55.98±0.76	57.59±0.56	58.88±0.72	61.61±0.08	63.30±0.31	64.03±0.30	66.48±0.84	66.24±0.55	66.82±0.19	73.91±0.84
	LGC-PPNP	**74.11±0.23**	**82.44±0.03**	**82.97±0.05**	**82.06±0.04**	**82.65±0.42**	**82.80±0.12**	82.54±0.34	84.82±0.42	84.62±0.26	84.22±0.13	85.81±0.30
	LGC-APPNP	29.45±1.80	35.84±0.02	35.77±0.10	35.55±0.15	36.17±0.07	36.08±0.01	36.41±0.22	35.80±0.31	35.70±0.33	35.23±0.10	73.64±0.02
Macro-F1	GCN [14]	27.83±3.38	23.49±1.40	28.23±4.54	27.50±2.29	27.32±1.11	31.84±1.45	33.25±1.74	31.81±0.88	35.43±4.40	36.69±2.88	43.81±0.13
	GCN† [14]	28.58±2.25	28.89±0.11	28.34±1.46	23.82±1.27	26.36±2.24	34.09±0.52	30.61±0.31	34.00±3.72	30.52±0.12	35.79±0.97	45.60±0.20
	PPNP [15]	20.31±2.66	22.51±1.65	21.69±2.28	22.76±3.85	18.85±0.91	21.42±2.60	17.58±1.29	22.18±5.33	20.29±3.09	23.28±3.56	43.37±0.27
	PPNP† [15]	20.49±0.48	21.02±1.59	20.67±1.95	18.20±0.38	18.13±0.58	18.14±0.51	15.49±1.87	15.39±1.56	20.67±5.50	19.82±6.76	44.98±0.28
	APPNP [15]	24.23±2.17	26.32±2.41	25.95±1.47	26.84±1.18	27.56±1.04	31.01±0.90	31.64±1.09	32.53±1.84	31.97±0.50	35.02±1.72	43.87±0.23
	APPNP† [15]	23.97±1.70	27.81±0.78	28.26±1.11	28.04±3.31	27.89±1.00	34.26±0.33	30.85±0.10	31.21±0.33	35.61±4.62	34.60±0.22	45.50±0.38
	GAT [10]	26.57±2.10	34.38±2.52	33.52±0.86	31.97±2.74	36.26±2.63	41.37±2.71	44.13±3.13	48.38±5.43	42.07±1.92	39.74±2.69	67.47±0.74
	LNet [24]	26.52±2.03	29.84±1.04	30.57±0.28	28.55±0.50	29.36±1.80	32.40±0.23	31.76±0.48	31.46±0.05	30.44±0.28	32.29±0.33	45.12±0.08
	AdaNet [24]	28.86±0.18	30.48±1.26	31.54±0.31	31.11±0.25	31.15±0.18	32.58±1.65	32.25±1.48	31.46±1.60	32.25±1.42	35.82±0.68	47.21±0.22
	SGC [25]	29.60±2.80	32.16±0.25	33.68±0.29	32.22±1.00	33.58±2.80	34.68±0.50	34.10±0.48	33.68±1.28	34.20±0.06	36.62±1.82	48.73±0.06
	MixHop [26]	26.52±3.08	33.21±1.80	34.20±1.10	34.60±1.41	34.07±0.40	35.46±1.45	36.25±0.28	35.42±0.20	36.26±0.07	38.25±1.52	49.46±0.80
	MobileGCN	25.92±4.46	39.45±0.63	41.47±0.27	41.62±0.61	41.94±0.37	40.51±0.70	40.65±0.22	39.92±0.23	40.39±0.29	42.22±0.78	47.97±0.59
	MobilePPNP	21.43±2.44	26.92±6.89	28.92±6.49	41.60±1.60	41.97±5.31	43.30±2.65	43.54±2.42	40.98±0.12	46.02±2.60	42.19±0.51	55.93±5.57
	MobileAPPNP	26.14±6.06	35.56±1.29	24.59±9.79	38.96±1.34	38.32±0.35	39.30±0.59	41.12±0.25	39.63±0.85	39.97±0.48	43.31±0.55	48.55±2.16
	LGC-GCN	37.67±1.57	45.23±0.66	47.92±0.81	48.19±0.52	50.44±0.42	**51.34±1.18**	**52.33±1.16**	**52.12±0.36**	**51.20±0.92**	**53.71±1.55**	54.81±0.67
	LGC-PPNP	**50.73±0.20**	**48.45±0.08**	**48.72±0.04**	**51.53±0.09**	**51.90±0.10**	50.97±1.14	48.37±0.72	48.92±2.19	48.78±1.25	49.08±1.38	55.99±1.65
	LGC-APPNP	28.02±0.79	29.81±1.56	24.71±1.93	22.41±0.34	29.60±0.95	31.67±0.51	36.34±0.61	33.94±0.54	35.27±0.59	33.71±0.85	**69.30±0.04**
MCC	GCN [14]	20.82±0.65	14.28±0.36	17.26±1.16	18.10±1.89	18.36±1.69	21.81±0.87	23.63±0.38	23.63±0.55	26.36±3.52	26.86±0.27	41.55±0.29
	GCN† [14]	17.80±1.52	19.52±0.94	19.15±1.47	13.83±1.95	17.07±3.91	21.73±0.25	20.84±0.40	24.84±2.67	22.24±0.14	28.53±1.06	42.56±0.11
	PPNP [15]	13.95±1.94	15.61±1.73	12.66±0.50	12.18±4.97	11.98±3.03	13.51±5.10	5.57±3.49	9.95±7.26	15.06±1.73	10.31±6.98	37.25±0.20
	PPNP† [15]	16.62±1.11	20.40±1.65	20.00±0.79	17.69±1.29	11.93±1.63	11.91±3.49	4.52±3.93	4.73±1.15	10.36±5.46	8.78±5.09	43.20±0.45
	APPNP [15]	16.13±3.02	16.43±2.02	17.41±0.05	18.21±1.12	19.74±3.09	20.59±0.50	21.83±0.19	23.39±1.39	23.62±1.36	28.26±2.59	42.15±0.23
	APPNP† [15]	20.22±4.05	21.48±1.48	18.95±0.80	15.45±2.72	19.30±0.59	23.36±0.72	23.21±1.20	22.54±0.18	26.87±4.67	28.94±0.18	44.08±0.42
	GAT [10]	25.60±3.17	30.76±4.76	26.19±2.29	25.53±2.32	23.13±1.50	33.40±10.54	32.66±2.57	36.53±2.21	34.40±2.61	34.82±1.82	62.12±1.88
	LNet [24]	18.26±4.75	19.06±3.28	20.18±3.08	20.35±2.68	21.68±2.86	21.69±1.13	20.26±1.50	22.80±1.40	23.62±2.18	23.58±2.60	45.68±0.12
	AdaNet [24]	19.12±2.66	20.25±1.09	19.18±1.36	20.12±3.40	22.49±3.58	22.75±3.25	23.48±2.80	23.62±2.41	23.82±2.60	22.65±1.24	44.68±0.25
	SGC [25]	19.98±2.52	21.48±4.83	20.54±2.40	21.44±1.67	21.46±1.58	22.52±2.62	22.60±2.65	23.28±2.40	24.64±2.20	24.68±1.52	46.14±0.09
	MixHop [26]	20.64±2.62	22.28±1.49	22.63±2.62	23.25±1.75	23.48±1.35	22.18±1.64	23.28±2.40	24.58±2.18	24.85±1.28	24.57±1.18	48.25±0.13
	MobileGCN	18.08±4.91	34.93±0.25	37.44±0.33	37.76±0.69	39.61±0.23	38.16±0.70	38.66±0.15	37.71±0.15	38.20±0.27	39.92±1.14	44.21±0.54
	MobilePPNP	12.38±2.70	18.34±4.73	18.50±8.35	32.95±0.55	35.72±6.38	37.05±2.09	37.98±1.71	36.84±0.08	**41.25±2.52**	39.74±0.66	48.51±4.69
	MobileAPPNP	13.34±4.66	27.00±2.06	15.13±12.19	34.55±1.15	35.17±0.64	36.16±0.38	39.34±0.01	37.30±0.71	37.58±0.47	41.15±0.56	45.28±1.81
	LGC-GCN	24.51±0.23	33.50±0.78	36.11±0.18	35.16±0.69	38.42±0.33	40.04±1.18	**40.95±1.13**	**40.71±0.42**	40.74±0.60	**43.89±0.39**	43.17±0.53
	LGC-PPNP	**35.93±0.14**	**38.82±0.25**	**39.53±0.04**	**41.73±0.50**	**41.79±0.63**	**41.06±0.36**	39.99±0.24	39.94±0.15	40.12±0.14	41.19±0.56	45.25±2.10
	LGC-APPNP	14.06±1.41	17.33±0.36	11.22±0.10	9.90±1.10	14.39±0.58	18.19±1.03	20.35±0.88	20.00±1.21	20.27±0.24	20.26±0.75	**66.14±0.05**

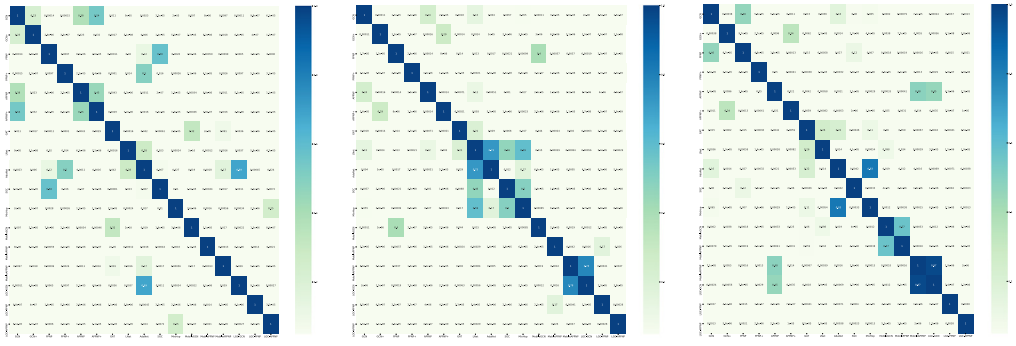


(a) CITESEER (b) CORA (c) PUBMED

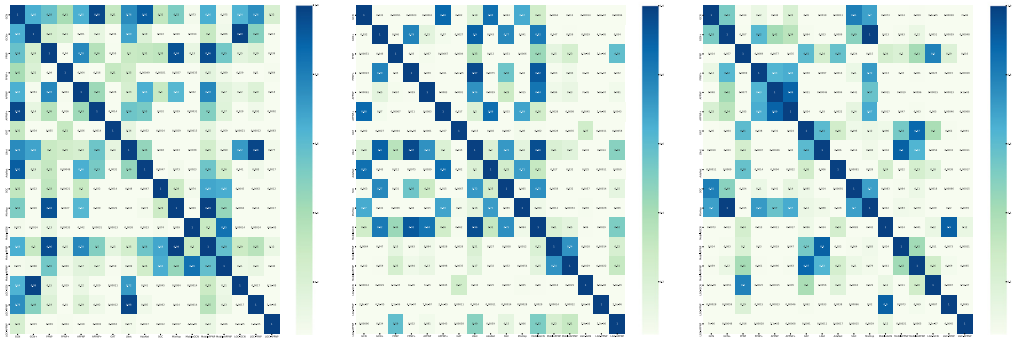


(d) Wikipedia (e) Photo (f) ROAD

Fig. 3. The p-value heatmap of the paired t-test with respect to Macro-F1 (1% features for all datasets).



(a) CITESEER (b) CORA (c) PUBMED



(d) Wikipedia (e) Photo (f) ROAD

Fig. 4. The p-value heatmap of the paired t-test with respect to MCC (1% features for all datasets).

Table 9
Evaluation results of node classification on the ROAD dataset.

Model	Accuracy	Macro-F1	MCC
GCN [14]	54.35±1.56	38.82±0.48	31.15±1.24
GCN† [14]	54.86±2.85	39.47±0.62	31.78±2.52
PPNP [15]	59.76±0.58	41.34±1.83	33.56±3.40
PPNP† [15]	61.42±1.05	41.48±0.69	32.28±2.46
APPNP [15]	57.72±0.46	40.06±0.84	31.17±0.88
APPNP† [15]	58.52±0.68	38.84±0.24	30.58±1.46
GAT [10]	64.65±0.45	44.35±0.86	35.54±2.50
LNet [24]	55.21±0.74	42.21±0.40	34.82±1.26
AdaNet [24]	55.65±1.48	42.25±0.82	36.78±0.68
SGC [25]	58.48±0.84	42.25±0.41	33.24±1.82
MixHop [26]	62.75±0.84	41.58±0.28	31.40±3.05
MobileGCN	70.42±0.45	51.20±0.24	38.48±2.56
MobilePPNP	73.54±1.40	48.85±0.20	34.48±1.50
MobileAPPNP	72.50±0.22	49.92±0.40	34.82±2.50
LGC-GCN	72.69±0.40	50.04±0.22	36.41±0.14
LGC-PPNP	**75.43±0.50**	52.28±1.24	38.89±0.86
LGC-APPNP	71.48±1.45	**56.45±0.60**	**40.03±0.14**

ture space. The proposed lightweight DSGC updater can reduce weights to offset over-fitting partly on these datasets and, simultaneously, it can aggregate features of neighboring nodes. Therefore, DSGC updater with the aggregation of neighboring node features, has applicability in a wide range of dimensions of the node feature space. However, in the high-dimensional feature space, GAT [10] also has excellent performance on the CITESEER, CORA, and PUBMED, and its attention mechanism can learn more graph-structured information between nodes. Our updaters, combined with attention mechanisms, will be a direction worthy in our future studies.

(4) Table 4 - 9 also show the comparison between GCNs' updater and LGC updater with the same number of weights. When the channels $|C^{(1)}|$ of GCN† [14], PPNP†, and APPNP† [15] are all set to 1024, they have the same weights as LGC-GCNs, whose channels $|C^{(1)}|$ is 32. Although we add several weights of GCNs (GCN†, PPNP†, and APPNP†) to prevent under-fitting, their metrics are still lower than that of our models. This experiment highlights the importance of LGC and DSGC updaters to learn graph-structured information in order to enhance metrics.

To summarize our comparisons on research datasets, we assume that the updaters, LGC and DSGC, introduce the standard convolution and its lightweight variant to aggregate the features of neighboring nodes in the encoding output step, which helps the message passing step to propagate graph-structured information between graph nodes. Our models hence provide the state-of-the-art results in the low-dimensional feature space.

6.2. Feature selection

Table 10 - 14 shows the influence of the feature selection on models' metrics. By employing the strategies of random selection, Chi^2 test, analysis of variance, and mutual information, respectively, we select 1% features from all datasets except ROAD that has only one feature. Experimental results demonstrate that:

(1) In the strategy of random selection, our LGC-PPNP performed on CITESEER, CORA, PUBMED, and Photo have the state-of-the-art results on all metrics. It indicates that our LGC updater collaborating with the PPNP aggregator is robust when sampling node features randomly.

(2) For the methods of Chi^2 test and analysis of variance, our six models provide respectively outstanding results of metrics on CITESEER, CORA, PUBMED, Wikipedia, and Photo. Along with the two feature selection methods, although the met-

Table 10
Evaluation results of feature selection on the CITESEER dataset (1%(37) features).

Model	Methods of feature selection for accuracy				Methods of feature selection for Macro-F1				Methods of feature selection for MCC			
	Random	Chi^2	Variance	Mutual	Random	Chi^2	Variance	Mutual	Random	Chi^2	Variance	Mutual
GCN [14]	23.00±0.70	68.20±0.10	67.65±0.35	41.65±14.45	29.02±0.17	51.37±1.02	50.44±0.20	39.20±5.98	10.32±1.72	42.83±2.16	40.91±0.34	26.31±9.49
GCN† [14]	23.80±2.50	67.95±0.15	67.90±0.30	43.25±7.15	29.73±0.29	51.59±0.93	51.10±0.56	41.64±5.52	10.12±0.60	43.38±1.13	42.96±1.77	30.25±5.91
PPNP [15]	28.20±2.30	67.75±0.45	68.40±0.30	34.90±4.30	28.80±0.92	51.73±0.27	52.12±0.55	36.34±4.62	10.07±2.48	43.44±1.38	43.50±1.57	20.58±5.18
PPNP† [15]	32.50±1.30	67.90±0.20	68.05±0.35	45.00±11.70	29.83±0.66	52.47±0.44	52.03±0.09	38.32±9.00	11.47±1.50	43.63±0.85	42.77±0.41	24.58±9.56
APPNP [15]	20.15±1.95	67.85±0.25	67.95±0.15	46.00±14.40	29.07±0.32	51.26±0.72	50.89±0.24	37.00±7.81	9.71±1.65	42.29±1.74	41.58±0.58	25.78±9.87
APPNP† [15]	20.15±1.65	67.90±0.40	67.95±0.35	47.85±9.55	28.86±0.43	51.38±0.38	50.99±0.28	41.58±3.46	8.64±0.89	43.21±0.59	41.70±0.61	30.38±8.02
GAT [10]	25.40±1.60	68.35±0.35	69.60±0.20	49.75±5.65	29.43±0.65	56.29±0.17	55.54±0.25	43.06±4.31	11.46±0.45	50.90±0.38	**51.14±0.53**	32.14±5.42
LNet [24]	26.42±1.87	66.25±0.08	67.04±0.26	40.26±4.20	28.62±1.54	50.52±0.40	50.12±0.38	40.21±2.34	9.25±2.30	40.08±0.12	40.25±0.22	26.45±3.25
AdaNet [24]	27.20±1.52	66.78±0.15	66.43±0.56	42.27±6.43	29.24±1.22	51.20±0.33	52.35±0.27	41.17±2.00	12.20±2.90	41.28±0.08	42.20±0.40	28.26±2.01
SGC [25]	24.51±2.58	65.52±0.20	66.78±0.33	41.40±3.41	29.25±2.42	50.08±0.36	52.27±0.10	42.62±1.01	13.24±1.28	42.08±0.14	43.14±0.20	29.51±0.42
MixHop [26]	28.85±2.50	66.23±0.25	67.82±0.02	42.25±1.20	28.40±1.68	52.25±0.06	53.10±0.11	41.07±0.48	11.28±3.08	43.25±0.24	43.41±0.20	30.13±0.40
MobileGCN	25.15±3.05	67.50±0.60	68.40±0.60	42.80±12.00	26.23±2.46	49.89±0.73	50.17±0.77	38.34±5.00	6.99±4.31	40.00±0.97	40.00±0.62	22.80±7.85
MobilePPNP	28.40±2.90	68.70±0.10	**69.80±0.20**	48.70±4.10	27.38±2.50	50.81±0.45	51.31±0.50	42.42±2.57	8.59±4.48	40.72±0.68	41.40±0.68	25.29±1.77
MobileAPPNP	25.00±1.90	67.85±0.35	69.10±0.40	55.45±6.25	26.72±2.09	51.19±1.76	49.36±0.70	44.37±3.04	7.26±2.09	41.35±2.54	39.01±0.83	29.95±4.10
LGC-GCN	31.60±0.50	67.25±0.85	67.50±0.30	49.40±3.40	31.07±0.90	49.18±0.59	48.87±0.39	40.03±2.18	14.47±1.15	39.75±0.56	39.32±0.18	27.73±1.42
LGC-PPNP	**45.35±0.25**	**68.90±0.40**	69.25±0.65	**58.15±6.95**	**38.99±0.35**	51.63±0.36	51.38±0.10	**45.89±2.90**	**22.28±0.34**	41.94±0.10	41.97±0.83	**32.46±4.25**
LGC-APPNP	31.85±0.95	66.15±0.25	68.60±0.20	41.50±4.80	30.47±0.29	**57.56±0.33**	**56.12±0.88**	37.94±4.92	14.84±1.51	**51.09±0.40**	49.64±1.06	29.61±5.54

Table 11
Evaluation results of feature selection on the CORA dataset (1%(14) features).

Model	Methods of feature selection for accuracy				Methods of feature selection for Macro-F1				Methods of feature selection for MCC			
	Random	Chi^2	Variance	Mutual	Random	Chi^2	Variance	Mutual	Random	Chi^2	Variance	Mutual
GCN [14]	22.50±0.50	68.60±0.30	68.70±0.10	37.85±7.35	25.96±0.16	46.09±1.48	46.05±1.54	37.59±4.31	9.73±0.18	39.72±0.28	36.30±0.90	25.98±4.25
GCN† [14]	23.65±1.75	68.45±0.35	69.40±0.50	34.45±12.95	26.61±0.51	48.13±0.86	47.76±0.47	34.88±7.21	11.93±0.97	38.36±0.40	36.17±0.38	22.08±11.03
PPNP [15]	16.90±3.30	73.90±0.40	72.55±0.35	36.50±5.00	22.10±4.82	47.37±1.82	47.26±2.77	32.08±5.11	6.84±6.08	39.58±0.43	35.86±0.27	24.02±4.35
PPNP† [15]	17.50±3.80	74.45±0.15	73.60±0.20	47.75±19.55	18.35±0.81	47.44±0.81	46.80±0.97	38.14±7.68	1.76±0.71	39.81±0.13	36.02±0.20	25.53±10.85
APPNP [15]	23.95±1.35	68.35±0.25	67.80±0.50	46.05±9.75	26.38±0.88	45.96±1.41	46.41±1.74	39.76±2.84	12.59±3.05	39.26±0.46	35.80±0.43	29.19±5.61
APPNP† [15]	24.30±0.80	68.85±0.25	69.40±0.30	34.05±17.75	26.24±0.61	46.69±0.66	46.89±0.68	31.22±8.05	10.63±1.24	38.02±0.24	36.24±0.53	17.23±11.40
GAT [10]	25.20±0.70	70.30±0.70	68.45±0.35	38.80±17.60	27.13±0.12	**61.20±0.72**	**58.68±0.37**	41.77±10.91	12.14±0.30	**56.99±0.72**	**52.42±0.66**	31.78±14.01
LNet [24]	23.58±2.48	67.24±0.16	68.12±0.40	40.27±3.16	25.53±0.64	46.00±0.35	44.96±0.68	38.25±2.04	10.06±2.58	37.75±0.42	35.15±0.18	24.62±4.25
AdaNet [24]	22.14±1.55	68.10±0.48	70.04±0.24	42.35±4.56	29.36±0.28	46.28±0.56	45.24±0.45	38.68±4.16	11.56±3.48	36.28±0.69	37.10±0.27	22.46±2.15
SGC [25]	24.45±3.58	68.24±0.10	69.92±0.46	41.45±5.25	28.80±0.16	49.68±0.15	47.10±0.54	39.96±2.80	10.58±2.15	38.58±0.98	39.06±0.28	23.31±3.30
MixHop [26]	25.68±2.20	66.54±1.28	70.52±0.40	41.16±2.10	28.05±0.42	49.76±0.68	46.28±0.18	40.24±2.08	13.25±3.24	39.16±0.14	40.21±0.30	24.46±3.68
MobileGCN	28.25±0.45	72.80±0.40	73.75±0.25	51.90±13.50	30.82±0.36	44.09±1.05	44.66±1.28	41.21±5.87	17.87±0.53	38.59±1.02	36.39±0.64	30.63±9.31
MobilePPNP	26.10±3.50	**77.30±0.60**	**78.20±0.50**	56.15±17.45	35.51±0.65	44.83±0.43	41.58±0.64	40.49±7.48	22.27±1.07	39.35±0.85	36.57±0.75	30.92±7.53
MobileAPPNP	28.10±2.10	73.55±0.55	74.75±0.35	50.30±12.20	32.69±0.36	45.06±0.81	45.13±3.28	42.58±4.88	19.15±0.33	38.81±0.57	37.17±1.24	31.34±5.40
LGC-GCN	35.55±0.35	72.30±0.20	74.35±0.15	45.30±11.20	33.45±0.13	46.14±1.27	45.68±0.47	39.79±4.79	20.61±0.23	41.06±1.19	39.10±0.34	29.20±6.75
LGC-PPNP	**62.25±0.75**	76.95±0.35	78.15±0.25	**68.90±3.70**	**41.62±0.23**	46.61±0.52	45.57±1.63	**45.31±1.34**	**35.05±0.26**	41.52±0.43	39.14±0.36	38.53±3.35
LGC-APPNP	35.40±3.00	59.70±0.40	57.10±0.30	51.00±7.50	35.83±1.99	49.25±0.17	50.35±0.28	44.24±3.20	25.23±1.75	39.13±0.08	37.58±0.19	**39.31±4.74**

Table 12
Evaluation results of feature selection on the PUBMED dataset (1%(5) features).

Model	Methods of feature selection for accuracy				Methods of feature selection for Macro-F1				Methods of feature selection for MCC			
	Random	Chi^2	Variance	Mutual	Random	Chi^2	Variance	Mutual	Random	Chi^2	Variance	Mutual
GCN [14]	49.70±0.80	51.70±2.90	49.85±2.95	49.50±0.90	44.07±0.91	51.33±0.21	52.91±2.08	51.31±4.14	18.00±3.49	24.41±0.77	21.33±0.77	22.08±5.68
GCN† [14]	50.10±0.30	55.10±0.08	64.50±1.00	52.50±2.10	46.75±1.32	55.95±0.05	62.14±0.11	62.79±4.66	17.45±0.74	34.57±0.22	43.17±0.16	39.27±5.05
PPNP [15]	57.70±0.70	48.10±1.30	47.55±3.95	39.70±0.90	51.48±0.75	47.68±1.04	52.21±2.85	33.74±10.67	27.01±0.56	18.47±2.71	21.35±4.46	7.57±3.24
PPNP† [15]	60.75±0.65	68.50±0.10	70.80±0.10	54.20±3.50	54.49±0.35	64.96±0.20	68.05±0.33	54.40±14.02	29.18±0.66	48.67±0.36	52.78±0.41	31.40±14.75
APPNP [15]	49.45±0.55	52.55±3.55	52.45±3.15	50.40±2.30	47.34±2.79	52.09±1.00	53.12±1.96	56.15±4.15	18.42±1.89	25.01±0.31	24.28±3.74	28.56±6.22
APPNP† [15]	49.70±0.50	55.25±0.15	66.50±0.40	67.15±1.65	47.11±1.82	59.71±0.26	64.58±0.50	67.35±2.06	21.71±5.90	38.56±0.46	47.60±0.85	48.01±2.99
GAT [10]	46.55±3.45	52.15±1.85	40.40±14.70	39.45±7.15	42.77±5.89	51.68±3.87	37.28±12.73	40.82±4.50	17.70±0.46	26.59±4.86	14.33±14.33	12.18±9.06
LNet [24]	48.26±0.28	65.52±0.26	67.76±0.21	50.01±0.41	49.16±2.18	51.08±0.45	58.34±1.24	57.20±3.14	22.08±2.48	30.06±2.15	38.52±2.50	28.62±4.72
AdaNet [24]	48.86±1.05	66.24±0.38	68.25±0.13	52.26±3.28	49.88±1.67	51.42±0.16	56.24±3.06	57.15±2.45	23.25±1.48	32.06±4.05	39.14±2.60	30.17±3.48
SGC [25]	46.52±2.58	65.48±0.64	67.25±0.26	52.68±2.19	48.80±2.53	50.18±0.46	57.12±2.51	56.45±3.00	24.45±2.06	33.14±2.58	38.45±1.69	31.41±5.40
MixHop [26]	50.26±0.09	67.25±0.15	68.12±0.28	51.36±4.25	50.06±2.17	53.36±0.12	58.28±1.52	59.42±2.80	26.45±2.18	35.02±1.45	40.08±2.16	32.28±1.42
MobileGCN	59.30±1.70	63.60±1.30	70.20±0.10	58.35±4.45	51.22±1.38	63.23±0.71	68.59±0.10	63.41±1.58	32.88±0.79	44.99±1.15	52.17±0.37	41.10±4.20
MobilePPNP	**62.90±0.40**	70.35±0.05	71.45±0.25	60.80±7.10	53.37±1.22	**67.03±0.45**	**68.65±1.20**	64.70±1.84	**36.13±3.79**	**51.89±0.63**	**53.06±1.57**	41.66±1.88
MobileAPPNP	52.75±0.85	64.60±0.50	70.05±0.25	53.80±8.00	49.25±0.21	62.95±1.05	66.52±0.47	58.42±8.23	19.46±1.33	45.38±1.33	49.90±0.71	34.22±9.66
LGC-GCN	59.00±0.30	67.15±0.15	69.20±0.20	60.85±4.35	52.94±0.87	65.55±0.25	68.29±0.52	64.39±0.91	29.03±1.97	49.39±0.42	52.80±0.69	43.34±2.25
LGC-PPNP	62.05±0.05	**71.05±0.15**	**73.15±0.15**	**70.90±6.10**	**57.92±0.17**	66.06±0.25	66.66±0.09	**69.06±2.12**	34.00±0.10	49.24±0.46	50.47±0.10	**51.19±5.66**
LGC-APPNP	46.70±2.40	40.05±1.35	38.20±3.80	45.50±2.80	44.12±1.74	44.38±2.36	42.93±5.61	41.64±4.77	15.06±2.75	10.39±1.25	8.50±5.04	17.25±4.35

Table 13
Evaluation results of feature selection on the Wikipedia dataset (1%(50) features).

Model	Methods of feature selection for accuracy				Methods of feature selection for Macro-F1				Methods of feature selection for MCC			
	Random	Chi^2	Variance	Mutual	Random	Chi^2	Variance	Mutual	Random	Chi^2	Variance	Mutual
GCN [14]	7.75±3.74	30.57±0.15	22.99±1.86	26.38±2.19	12.27±1.24	15.25±0.30	14.61±0.83	13.80±1.10	10.29±1.46	17.86±0.57	14.69±0.49	14.77±1.69
GCN† [14]	28.83±2.25	33.32±0.51	24.46±1.89	30.15±3.68	19.88±1.31	15.56±0.32	18.39±0.48	16.42±2.79	12.90±2.16	19.10±0.26	17.33±1.12	18.65±2.45
PPNP [15]	7.13±4.85	25.45±2.69	15.39±4.07	17.81±5.00	8.25±3.21	13.56±1.36	12.28±1.09	13.40±1.69	5.10±4.01	15.22±0.62	10.87±2.33	11.80±2.76
PPNP† [15]	7.10±3.62	26.86±0.33	16.14±1.41	20.93±3.62	13.42±1.79	13.35±0.13	11.78±2.28	15.24±1.59	13.92±4.21	16.16±0.73	12.50±3.58	15.46±6.01
APPNP [15]	18.44±0.66	30.33±0.39	23.56±1.65	25.21±4.91	14.30±0.78	14.97±0.10	13.97±0.14	15.39±0.82	15.78±3.06	18.54±0.28	13.88±0.12	18.22±3.55
APPNP† [15]	29.94±2.87	33.80±0.87	26.32±1.71	26.89±6.29	18.47±0.92	15.36±0.09	16.10±0.98	16.96±1.67	20.33±0.73	19.11±0.27	15.47±1.49	19.46±1.94
GAT [10]	24.16±3.80	30.81±1.11	26.89±4.01	27.10±2.07	17.60±1.25	18.41±0.33	17.23±1.51	15.85±1.88	19.01±0.60	23.03±0.66	17.27±1.91	17.33±1.45
LNet [24]	20.87±2.15	31.28±0.58	26.46±1.27	25.10±3.41	15.98±1.53	19.85±0.24	18.96±0.40	18.54±1.40	16.68±0.45	21.16±0.47	17.78±1.48	18.82±2.50
AdaNet [24]	21.62±3.45	32.18±0.34	26.15±2.58	26.45±3.54	15.42±1.28	20.04±0.84	20.15±0.47	16.54±2.45	17.75±0.41	22.25±0.41	18.85±2.48	18.46±2.10
SGC [25]	18.45±2.58	30.18±0.45	24.86±1.46	25.63±3.50	16.54±0.82	19.57±0.47	19.40±0.47	18.62±1.06	15.45±1.24	18.56±0.24	19.42±1.42	19.25±2.85
MixHop [26]	22.25±2.86	33.48±0.18	27.72±1.50	26.66±3.30	16.48±2.10	20.05±0.18	20.25±0.16	19.90±1.46	16.48±1.40	19.85±1.58	21.12±2.58	18.68±1.68
MobileGCN	13.05±3.53	54.73±0.72	53.98±0.45	**57.13±1.14**	11.50±1.72	16.48±0.63	16.17±0.13	16.13±0.33	9.35±2.90	22.23±0.40	22.53±0.15	22.19±0.77
MobilePPNP	18.05±1.35	52.31±0.51	44.13±1.08	32.51±16.11	10.17±0.78	16.46±0.29	15.08±0.40	15.14±3.03	9.26±4.13	21.17±0.64	19.92±0.63	15.07±5.08
MobileAPPNP	17.49±4.19	**56.26±0.75**	**54.07±1.44**	43.74±14.46	11.57±0.69	17.06±0.34	16.27±0.18	16.35±0.80	9.89±3.68	22.49±0.35	22.57±0.18	20.87±2.33
LGC-GCN	24.37±0.36	42.63±0.54	35.78±0.33	41.02±2.69	12.51±0.35	18.44±0.39	16.29±0.44	17.26±0.60	18.29±0.07	22.74±0.76	20.42±0.26	22.23±0.69
LGC-PPNP	12.93±0.30	56.05±0.54	35.87±0.36	41.11±1.77	11.28±0.55	17.65±0.43	16.06±0.14	16.99±0.51	15.55±0.35	19.58±0.38	19.58±0.29	19.11±0.29
LGC-APPNP	**38.41±0.87**	35.00±3.26	32.16±3.95	41.89±3.56	**20.76±0.59**	**24.52±0.56**	**23.87±1.24**	**23.55±1.62**	**21.85±0.54**	**29.72±1.70**	**27.51±1.67**	**30.57±2.63**

Table 14
Evaluation results of feature selection on the Photo dataset (1%(7) features).

Model	Methods of feature selection for accuracy				Methods of feature selection for Macro-F1				Methods of feature selection for MCC			
	Random	Chi^2	Variance	Mutual	Random	Chi^2	Variance	Mutual	Random	Chi^2	Variance	Mutual
GCN [14]	40.88±5.73	20.14±6.52	31.43±8.07	27.05±7.84	25.29±0.53	27.15±0.14	37.38±0.39	26.33±4.93	20.10±4.24	16.41±2.26	31.43±0.55	14.01±4.31
GCN† [14]	44.97±1.59	14.21±0.59	28.20±3.62	29.17±4.61	27.26±1.35	27.29±0.51	37.78±0.28	23.72±1.63	27.17±4.00	14.61±0.26	32.86±0.11	12.81±7.31
PPNP [15]	40.64±0.86	25.45±11.64	16.25±1.45	23.94±5.49	21.95±3.59	31.80±0.25	20.41±3.20	31.89±6.62	17.81±3.17	20.81±0.59	5.13±3.71	13.81±7.43
PPNP† [15]	40.28±4.87	17.36±3.55	22.82±8.39	33.09±10.12	20.19±0.64	32.44±0.01	23.87±1.80	33.47±6.20	14.99±1.79	22.42±0.04	13.15±5.74	17.22±3.84
APPNP [15]	43.80±2.07	19.24±2.68	25.95±1.84	30.14±4.59	27.55±3.81	27.84±1.00	37.39±0.09	26.57±6.96	23.38±3.11	17.09±0.98	32.53±0.08	19.27±3.09
APPNP† [15]	40.76±6.62	16.00±1.64	27.30±3.02	28.76±1.14	26.99±0.56	28.22±0.17	37.53±0.04	28.74±6.86	24.00±0.97	16.41±0.45	32.87±0.13	18.77±7.42
GAT [10]	42.74±3.41	24.11±8.26	24.54±0.03	23.75±7.46	31.33±1.92	20.37±1.26	37.94±0.13	31.03±8.16	25.84±5.24	8.97±2.93	32.49±0.19	19.84±4.56
LNet [24]	42.56±5.26	26.65±5.68	28.45±2.58	25.46±3.48	26.24±1.50	21.48±1.40	33.45±0.21	24.54±4.50	23.15±1.33	15.45±0.15	30.45±0.14	17.49±3.08
AdaNet [24]	43.56±2.18	27.15±4.52	29.42±1.48	26.48±3.45	25.15±2.40	20.18±1.50	32.12±1.46	26.14±3.28	22.15±2.48	16.58±0.75	31.05±0.45	18.82±2.48
SGC [25]	45.52±2.48	28.89±4.78	30.05±1.85	27.54±2.86	30.08±1.48	21.48±1.47	32.54±0.48	25.58±4.50	26.14±2.48	18.84±0.14	30.08±0.25	19.45±3.68
MixHop [26]	44.15±3.14	27.74±1.47	29.48±2.57	27.74±1.45	30.04±2.84	24.36±0.47	34.84±0.17	28.81±4.61	27.75±1.04	20.06±1.64	31.12±0.04	21.18±4.20
MobileGCN	28.76±1.09	72.16±0.30	67.03±0.67	74.44±3.26	22.79±1.29	36.96±0.10	42.03±0.09	40.18±0.64	13.61±5.78	**34.06±0.19**	**37.49±0.27**	32.83±1.57
MobilePPNP	32.90±3.34	67.17±0.26	72.89±0.22	78.48±0.80	18.54±5.12	40.76±2.14	40.69±0.89	41.12±3.65	16.16±3.77	28.42±1.73	33.96±0.94	31.36±1.57
MobileAPPNP	41.12±3.75	72.93±0.73	69.71±0.32	78.84±6.44	24.16±1.92	37.98±0.27	40.81±0.61	38.32±0.58	19.37±3.33	32.57±0.20	33.75±0.85	32.54±1.16
LGC-GCN	39.10±0.29	51.74±0.14	52.83±0.11	60.81±4.77	32.60±0.17	40.35±0.25	42.62±0.20	**43.78±1.27**	22.67±0.23	29.35±0.20	33.29±0.18	34.01±0.58
LGC-PPNP	**70.96±2.04**	**81.76±0.52**	**79.25±0.04**	**83.97±1.07**	**43.88±0.53**	**42.13±0.28**	**43.14±0.24**	42.56±1.57	**32.98±0.17**	31.57±0.65	36.07±0.10	**34.92±1.43**
LGC-APPNP	30.49±0.52	20.19±3.18	30.23±5.66	29.56±17.62	26.85±1.32	18.93±3.81	28.69±1.62	34.25±8.84	15.60±1.07	10.91±5.00	22.86±2.54	18.56±5.82

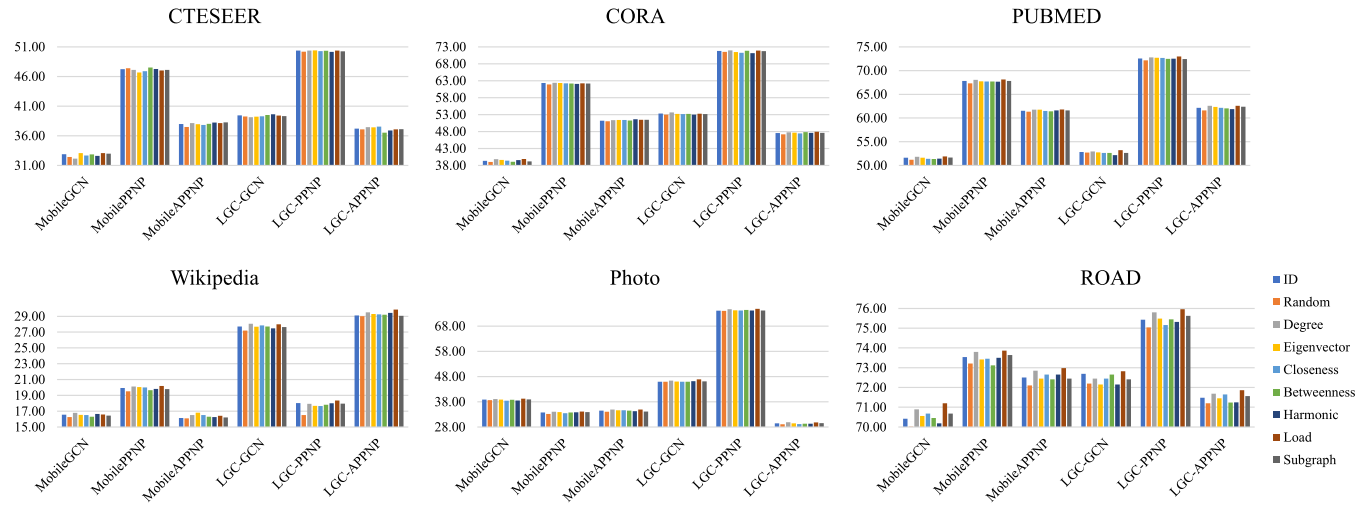


Fig. 5. Our models' accuracy with different affinity values.

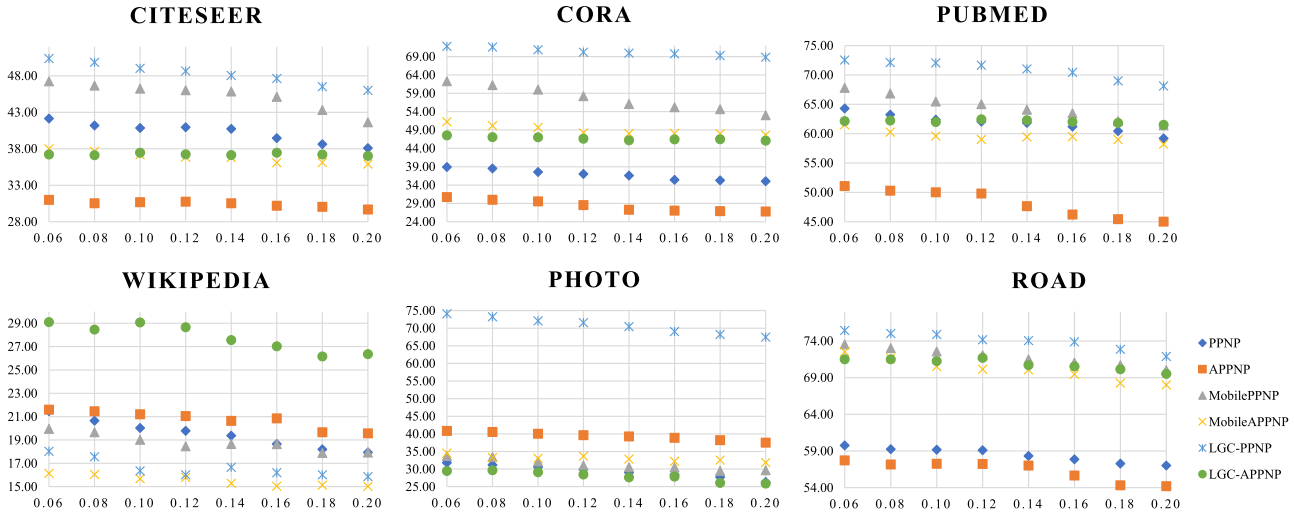


Fig. 6. The teleport probability α of PPNP, APPNP, and our models controls the proportion of learning graph-structured information to enhance the models' accuracy.

ric results of benchmark models have been improved immensely, the results of MobilePPNP and LGC-PPNP are still prominent. Moreover, LGC-PPNP and LGC-APPNP dominate the accuracy of Macro-F1 and MCC on the method of mutual information. Therefore, the experiments further prove that our models are robust.

(3) However, GAT performed on CORA has outstanding results of Macro-F1 and MCC, and on CITESEER, it has a state-of-the-art result on MCC by using the method of the analysis of variance. Therefore, our models are not outstanding on all datasets and need to be verified on more datasets in future work.

6.3. Floating-point of operations (FLOPs) and weights

Table 15 summarizes the FLOPs and weights of MobileGCNs and LGC-GCNs, where FLOPs can be used to measure one model's complexity, and weights describe this model's memory cost. According to the values of FLOPs, we conclude that the complexity of LGC-GCN is the highest because its GCN aggregator needs to calculate the cost of the graph normalized Laplacian matrix at each layer while the heavyweight LGC updater is used in it. On the contrary, the complexity of MobilePPNP is the lowest since the matrix of its PPNP aggregator is calculated once in the last layer, and it also uses lightweight DSGC as its updater, synchronously. Furthermore, the experiment indicates that DSGC in MobileGCNs, as a lightweight updater, occupies less dramatically memory than the heavyweight LGC updater in LGC-GCNs on all datasets.

6.4. Affinity values

Fig. 5 summarizes the effect of various affinity values on the accuracy of our models performed on all datasets with sampling 1% features. We can see that the accuracy of all our models does not significantly differ from diverse affinity values. This experiment may be explained by the fact that node features in each aggregated node's neighborhood are homogeneous when diverse affinity values are adopted to arrange the order of connections in this neighborhood. Another possible explanation for this is that as long as each neighborhood maintains the same connection order, irrespective of the affinity value used to determine this connection, all our models' accuracy is insignificantly different. Therefore, further intensive research is required to investigate why our models with the various connection orders yield a similar result.

6.5. Teleport probability α

As an essential hyper-parameter of PPNP and APPNP [15], teleport probability α controls the proportion of learning graph-structured information via using the personalized PageRank algo-

Table 15
The statistics of FLOPs (**M** indicates a million) and weights.

Model	CITESEER		CORA		PUBMED		Wikipedia		Photo		ROAD	
	FLOPs	weights	FLOPs	weights	FLOPs	weights	FLOPs	weights	FLOPs	weights	FLOPs	weights
MobileGCN	2975**M**	2400	1626**M**	1696	78569**M**	1280	1725**M**	3232	12084**M**	1504	1856**M**	2506
MobilePPNP	429**M**	2400	321**M**	1696	7038**M**	1280	695**M**	3232	2837**M**	1504	526**M**	2506
MobileAPPNP	828**M**	2400	629**M**	1696	14037**M**	1280	1355**M**	3232	5647**M**	1504	968**M**	2506
LGC-GCN	3826**M**	44,032	1965**M**	21,504	79509**M**	8192	2712**M**	70,656	12767**M**	15,360	2468**M**	26,530
LGC-PPNP	1281**M**	44,032	660**M**	21,504	7977**M**	8192	1683**M**	70,656	3520**M**	15,360	852**M**	26,530
LGC-APPNP	1679**M**	44,032	968**M**	21,504	14976**M**	8192	2343**M**	70,656	6330**M**	15,360	1645**M**	26,530

rithm [17]. Fig. 6 demonstrates that: the accuracy of MobilePPNP, MobileAPPNP, LGC-PPNP, and LGC-APPNP decreases gradually on the low-dimensional datasets (1% features) with rising α. We observe that our models learn more graph-structured information to enhance their accuracy by decreasing α in the PPR aggregators. Inversely, when increasing α, low-dimensional features have to provide incomplete node representations to learn inadequate node-featured information while ignoring the importance of graph-structured information. In future works, this probability α should be adjusted for the dataset under investigation because the neighborhood structures of different graphs vary [15].

7. Conclusions and future works

In this paper, we introduced LGC-GCNs constructions (i.e., LGC-GCN, LGC-PPNP, and LGC-APPNP) based on the LGC updater; and the lightweight versions of LGC-GCNs (MobileGCN, MobilePPNP, and MobileAPPNP) created by using the DSGC updater. Both LGC and DSGC updaters encode node hidden states during the encoding output step to capture node-featured information, and they can also cooperate with aggregators in the message propagation step to gather the features of neighboring nodes to learn graph-structured information. Consequently, our models can improve the model metrics in low-dimensional node feature space, highlighting three advantages:

(1) Our LGC and DSGC, designed in a modular way, can integrate with any of the other aggregators. In this article, we have combined our updaters with GCN [14], PPNP, and APPNP [15] aggregators to enhance the three metrics effectively in the low-dimensional feature space.
(2) LGC updater can be extended and improved by taking advantage of the up-to-date achievement of the standard convolution, since its infrastructure roots in the framework of the standard convolution.
(3) Proposed two updaters can not only assist any aggregators to learn graph-structured information but also encode node hidden representations to extract node-featured information in the encoding output step, which observably powers the ability to obtain graph-structured information when graphs represented by the low-dimensional feature space.

For future work, there are two improvement directions for the defects of our updaters:

(1) To solve the information loss in our incomplete neighborhood, it is worth utilizing the ASGCN's transformation [30] from arbitrary-sized graphs into fixed-sized aligned grid structures.
(2) To perform our updaters on the graph classification task, we will try to introduce graph embedding [35] and kernel [36] into the framework of our updaters in the next work.

Furthermore, in order to further exploit our models' potential capacity, we try to combine our updaters with more aggregators such as GAT [10] and Tree-LSTM [11] to improve metrics on more datasets. Moreover, more achievements in standard convolution can be embedded into our models to deal with a variety of tasks like the graph, node, or link classification.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Wei Dong, Junsheng Wu, Zongwen Bai, Yaoqi Hu, Weigang Li, Wei Qiao, Marcin Wozniak.

Acknowledgments

Our work has been supported by these following projects:(1) Grant No. 2017YFC0804103, Chinese 13th Five-Year Key Research and Development Program. (2) Grant No. 61761042, National Natural Science Foundation of China. (3) Grant No. 2017KG-01, Yanan Research Project. (4) Grant No. 2019GY-020, Key R & D projects of Shaanxi Province.

References

[1] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Advances in Neural Information Processing Systems, 2017, pp. 1024–1034.
[2] M. Ahmed, J. Islam, M.R. Samee, R.E. Mercer, Identifying protein-protein interaction using tree lstm and structured attention, in: 2019 IEEE 13th International Conference on Semantic Computing, IEEE, 2019, pp. 224–231.
[3] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al., Recent advances in convolutional neural networks, Pattern Recognit 77 (2018) 354–377.
[4] M. Fey, J.E. Lenssen, C. Morris, J. Masci, N.M. Kriege, Deep graph matching consensus, in: International Conference on Learning Representations, 2020.
[5] D. Garg, S. Ikbal, S.K. Srivastava, H. Vishwakarma, H. Karanam, L.V. Subramaniam, Quantum embedding of knowledge for reasoning, in: Advances in Neural Information Processing Systems, 2019, pp. 5595–5605.
[6] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[7] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, et al., Deep speech 2: End-to-end speech recognition in english and mandarin, in: International Conference on Machine Learning, 2016, pp. 173–182.
[8] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 4171–4186.
[9] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Advances in Neural Information Processing Systems, 2016, pp. 3844–3852.
[10] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention tetworks, in: International Conference on Learning Representations, 2018.
[11] V. Zayats, M. Ostendorf, Conversation modeling on reddit using a graph-structured lstm, Transactions of the Association for Computational Linguistics 6 (2018) 121–132.
[12] Q. Li, Z. Han, X.-M. Wu, Deeper insights into graph convolutional networks for semi-supervised learning, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[13] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, M. Sun, Graph Neural Networks: A Review of Methods and Applications, 2018. arXiv preprint arXiv:1812.08434
[14] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations, 2017.
[15] J. Klicpera, A. Bojchevski, S. Günnemann, Predict then propagate: Graph neural networks meet personalized pagerank, in: International Conference on Learning Representations, 2018.
[16] S.P. Borgatti, Centrality and network flow, Soc Networks 27 (1) (2005) 55–71.
[17] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank citation ranking: Bringing order to the web., Technical Report, Stanford InfoLab, 1999.
[18] L. Sifre, S. Mallat, Rigid-motion scattering for image classification, PhD thesis, Ph. D. thesis 1 (2014) 3.
[19] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, 2015, pp. 448–456.
[20] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861 (2017).
[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520.
[22] J. Bruna, W. Zaremba, A. Szlam, Y. Lecun, Spectral networks and locally connected networks on graphs, in: International Conference on Learning Representations, 2014, pp. http–openreview.
[23] M. Henaff, J. Bruna, Y. LeCun, Deep convolutional networks on graph-structured data, Advances in Neural Information Processing Systems, 2015.
[24] R. Liao, Z. Zhao, R. Urtasun, R.S. Zemel, Lanczosnet: Multi-scale deep graph convolutional networks, in: International Conference on Learning Representations, 2019.
[25] F. Wu, T. Zhang, A. Holanda de Souza, C. Fifty, T. Yu, K.Q. Weinberger, Simplifying graph convolutional networks, in: Proceedings of Machine Learning Research, 2019.
[26] S. Abu-El-Haija, B. Perozzi, A. Kapoor, N. Alipourfard, K. Lerman, H. Harutyunyan, G. Ver Steeg, A. Galstyan, Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing, in: International Conference on Machine Learning, 2019, pp. 21–29.
[27] D.K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, R.P. Adams, Convolutional networks on graphs for learning molecular fingerprints, in: Advances in Neural Information Processing Systems, 2015, pp. 2224–2232.
[28] J. Atwood, D. Towsley, Diffusion-convolutional neural networks, Advances in Neural Information Processing Systems, 2016.
[29] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, M.M. Bronstein, Geometric deep learning on graphs and manifolds using mixture model cnns, in: IEEE Conference on Computer Vision and Pattern Recognition, 1, 2017, p. 3.
[30] L. Bai, Y. Jiao, L. Cui, E. Hancock, Learning aligned-spatial graph convolutional networks for graph classification, in: The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, 2019.
[31] L. Bai, L. Cui, Y. Jiao, L. Rossi, E. Hancock, Learning backtrackless aligned-spatial graph convolutional networks for graph classification, IEEE Trans Pattern Anal Mach Intell (2020). 1–1
[32] L. Bai, L. Cui, X. Bai, E. Hancock, Deep depth-based representations of graphs through deep learning networks, Neurocomputing, 336, 2018.
[33] Z. Zhang, C. Dongdong, Z. Wang, H. Li, L. Bai, E. Hancock, Depth-based subgraph convolutional auto-encoder for network representation learning, Pattern Recognition, 90, 2019.
[34] Z. Zhang, C. Dongdong, J. Wang, L. Bai, E. Hancock, Quantum-based subgraph convolutional neural networks, Pattern Recognit 88 (2019) 38–49.
[35] A. Martino, A. Giuliani, A. Rizzi, (hyper)graph embedding and classification via simplicial complexes, Algorithms, 11: 223, 2019.
[36] A. Martino, A. Rizzi, (hyper)graph kernels over simplicial complexes, Pattern Recognition, 2020.
[37] T.H. Haveliwala, Topic-sensitive pagerank: a context-sensitive ranking algorithm for web search, IEEE Trans Knowl Data Eng 15 (4) (2003) 784–796.
[38] M.E. Newman, Power laws, pareto distributions and zipf's law, Contemp Phys 46 (5) (2005) 323–351.
[39] J. McAuley, C. Targett, Q. Shi, A. Hengel, Image-based recommendations on styles and substitutes, Special Interest Group on Information Retrieval, 2015.
[40] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, Y. Wang, Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction, Sensors 17 (4) (2017) 818.

Wei Dong is a Ph.D. student with the School of Computer Science and Engineering, Northwestern Polytechnical University, China. He received the MS degree in Xi'an University of Architecture and Technology 2014. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an. His research interests cover computer vision, graph neural network and deep learning.

Junsheng Wu is currently a doctoral supervisor in the School of Computer Science and Engineering, Northwest Polytechnical University and a professor in the School of Software, Northwest Polytechnical University. His research interests cover domain-oriented software engineering technology and application. He is engaged in research and development, testing and validation of manufacturing enterprise application software system, airborne data acquisition and integrated processing system, computing visualization technology and its application system.

Zongwen Bai is with the Shaanxi Provincial Key Lab of bigdata of energy and intelligence processing, School of physics and electronic information, Yanan University, Yanan 716000, China. He received the MS degree in Yanan university 2008. He is currently pursuing the Ph.D. degree with the School of Computer Science, Northwestern Polytechnical University, Xi'an. He is a joint PhD student with the School of Computer Science, Northwestern Polytechnical University. His research interests cover computer vision, nature language processing and deep learning. This work was supported by the National Natural Science Foundation of China (Grant No.61761042), the Key Research and Development Program of Yanan (Grant No. 2017KG-01,2017WZZ-04-01).

Yaoqi Hu received the B.S. and M.S. degrees from Northwestern Polytechnical University, Xin, China, in 2014 and 2017, respectively, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Northwestern Polytechnical University. His current research interests include target tracking, and pattern recognition.

Weigang Li received the Ph.D. degrees in Manufacturing Engineering from Northwestern Polytechnical University, Xi'an, China in 2003. In 2007, he joined the faculty of Software Engineering of the Northwestern Polytechnical University (NPU). He is working as an associate professor of software engineering in the School of Software at the NPU. His research interests include cloud computing, big data processing for enterprise applications. He has successfully delivered many IT projects in China since 2004 and published over 30 peer-reviewed papers.

Wei Qiao works in Xi'an Research Institute Co., Ltd., China Coal Technology and Engineering Group, Shaanxi Key Laboratory of Coal Mine Water Hazard Prevention and Control Technology, Shaanxi Xi'an 710077. He is a joint Ph.D. student with the School of Computer Science and Engineering, Northwestern Polytechnical University, China. He received the MS degree in Xi'an University of Architecture and Technology 2014. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an. His research interests cover big data processing, cloud computing and deep learning.

Marcin Woźniak received diplomas in applied mathematics and computational intelligence. He is an Assoc. Professor at the Faculty of Applied Mathematics of the Silesian University of Technology in Gliwice, Poland. In his scientific career, he was visiting University of Wurzburg, Germany, University of Lund, Sweden and University of Catania, Italy. His main scientific interests are neural networks with their applications together with various aspects of applied computational intelligence. He is a scientific supervisor in editions of "the Diamond Grant" and "The Best of the Best" programs for highly gifted students from the Polish Ministry of Science and Higher Education. Marcin Woźniak served as an editor for various special issues in IEEE ACCESS, Sensors, Frontiers in Human Neuroscience, CIN, etc., and as an organizer or a session chair at various international conferences and symposiums, including IEEE SSCI, FedCSIS, ICAISC, APCASE, ICIST, etc.