Proceedings Paper:
Li, Huan, Wang, Boyuan, Cui, Lixin et al. (2 more authors) (2021) LGL-GNN: Learning Global and Local Information for Graph Neural Networks. In: Torsello, Andrea, Rossi, Luca, Pelillo, Marcello, Biggio, Battista and Robles-Kelly, Antonio, (eds.) Structural, Syntactic, and Statistical Pattern Recognition. Springer , Cham , pp. 129-138.

https://doi.org/10.1007/978-3-030-73973-7_13

LGL-GNN: Learning Global and Local Information for Graph Neural Network

Huan Li[1], Boyuan Wang[1⋆], Lixin Cui[⋆⋆], Lu Bai, Edwin R. Hancock

[1] School of Information, Central University of Finance and Economics, Beijing, China
[2] Department of Computer Science, University of York, York, UK

Abstract. In this article, we have developed a graph convolutional network model LGL that can learn global and local information at the same time for effective graph classification tasks. Our idea is to concatenate the convolution results of the deep graph convolutional network and the motif-based subgraph convolutional network layer by layer, and give attention weights to global features and local features. We hope that this method can alleviate the over-smoothing problem when the depth of the neural networks increases, and the introduction of motif for local convolution can better learn local neighborhood features with strong connectivity. Finally, our experiments on standard graph classification benchmarks prove the effectiveness of the model.

Keywords: Graph Convolutional Networks · Graph Classification

1 Introduction

In recent years, deep learning has achieved outstanding performance in many fields such as computer vision and natural language processing. The existing deep learning models can handle structured data such as images and speech well, but they are difficult to apply to graph data directly. However, in real life, there are a large number of non-European data represented in the form of graphs. For example, graphs can be abstracted from social networks, citation networks, protein-interaction networks and other scenarios. Graph is not only ubiquitous, but also can flexibly describe the complex relationships between real things and has a strong structured expression ability. These advantages have inspired researchers to further expand their research horizons to the field of deep learning and graph. However, unlike image data with a regular grid structure, each node in the graph has a different number of neighbor nodes, so basic convolution and pooling operations cannot be used, which poses a huge challenge to the existing convolutional neural network.

When extending CNN to the irregular grid structure of graph, two main strategies are adopted, a) Spectral-based [1, 2] and b) Spatial-based [3–5] methods. Most existing GCNs are designed under these two strategies. Specifically,

⋆ Co-first Author: These authors have contributed equally to this work.
⋆⋆ Corresponding Author: cuilixin@cufe.edu.cn

Spectral-based GCN defines convolution operations based on spectral graph theory. This method requires graphs to have the same size of structure and is usually used for vertex classification tasks. Spatial-based GCN approximates the spectral convolution operation by defining the layer-by-layer propagation of a node-based one-hop neighborhood. It is not limited to the same size of graph structure, and can be used for graph classification tasks. Although the Spatial-based GCN model can handle graph classification problems, the particularity of the graph structure still brings some difficult problems to GCN. One of the most difficult problems is over-smoothing [6, 7] . As the number of network layers increases and the number of iterations increases, the representation of each node tends to converge to the same value, which means that the global information of the entire graph is synchronized to every node, rather than the local structural features we expect.

To overcome the problem of over-smoothing, there are roughly two ways of thinking at present. On the one hand, the SortPooling layer is used to replace SumPooling. The SumPooling layer directly aggregates the learned local vertex features from graph convolution operations into global features. It is difficult to learn rich local vertex topological information from global features, resulting in poor classification results. M. Zhang et al. [8] proposed a novel Deep Graph Convolutional Neural Network (DGCNN), which uses a novel SortPooling layer to sort the extracted multi-scale vertex features instead of summing. DGCNN pays more attention to local vertex features, but only retains the top specified number of vertices when sorting, which may cause a lot of information to be lost. On the other hand, convolve the local subgraph of the node. Z. Zhang et al. [9] designed a local convolution operation based on a subtree. Since the local subgraph only retains the information of the nodes closer to the root node, the design of the graph convolution operation on the subgraph can limit the information interaction with remote nodes, but also loses global information.

The LGL model we proposed is inspired by the simultaneous attention and fusion of global and local information. While using the graph convolutional layer to learn global information, the subgraph convolutional layer is used to learn local node features, and the attention mechanism is introduced to give different weights to them. The framework of the proposed LGL is shown in Fig. 1. Specifically, the main contributions of this paper are summarized as follows:

First, we designed a new local convolution operation based on motif. Motif is a subgraph that appears frequently in graph. Each node in motif has strong connectivity. The using of motif can effectively capture high-quality local neighborhood information.

Second, We have developed a novel hybrid graph convolutional network model for graph classification, which is the LGL model. The LGL model uses the depth graph convolutional network and the subgraph convolutional network to learn global information and local information respectively, and the attention mechanism gives weight to both.

Third, we evaluate the performance of the proposed LGL model on graph classification tasks by means of experience. Experiments on benchmarks demon-

strate the effectiveness of the proposed method, when compared to state-of-the-art methods.

2 Related Works

In this section, we briefly review some important related work of LGL model, including the Deep Graph Convolutional Neural Network (DGCNN) [8], the Subgraph Convolutional Neural Networks (SCN) [9] and motif.

Deep Graph Convolutional Neural Network Given a graph $\mathbf{G}$ with n nodes, $\mathbf{X} \in \mathbb{R}^{n\times c}$ is the node feature vectors and $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the graph adjacency matrix. Spatially-based Deep Graph Convolutional Neural Network (DGCNN) [8] model [8] takes the following graph convolution operation

$$\mathbf{Z} = f(\tilde{\mathbf{D}}^{-1}\tilde{\mathbf{A}}\mathbf{X}\mathbf{W}), \tag{1}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ means that graph is added to the self-loops, $\tilde{\mathbf{D}}$ is its diagonal degree matrix, $\tilde{\mathbf{W}} \in \mathbb{R}^{c\times c'}$ is a matrix of trainable convolution parameters, f is a nonlinear activation function, and $\mathbf{Z}$ is the output after convolution operation.

For the graph convolution operation defined by Eq. 1, $\mathbf{XW}$ maps the features of each node from the c dimension to the c' dimension, $\tilde{\mathbf{A}}\mathbf{Y}(\mathbf{Y} := \mathbf{XW})$ spreads the features information of each node to the neighboring nodes and the node itself, thus realizing the aggregation of nodes information. However, the distance between any two nodes in the graph is relatively close, and it takes only a few steps from one node to another, so when the number of convolutions increases, the problem of over-smoothing appears.

Subgraph Convolutional Neural Network Different from DGCNN, Quantum-based Subgraph Sonvolutional Neural Networks (QS-CNNs) model [9] extract the neighborhood subgraph of each node through quantum walks, and use graph grafting and graph pruning to generate an m-ary tree for each node. The leaf nodes of the m-ary tree are further replaced by their own neighboring m-ary trees, and this process is performed recursively until a K-level and m-ary tree is constructed for each node. Since QS-CNNs generates a K-level extended subtree for each node, it can effectively learn the local connection structure information of the node.

Motif Motif has a long history in network research. The concept of motif was first introduced in 2002 [10] , which represents the frequently repeated patterns in complex networks and is the building block of complex networks. Some work [11–13] proves that motif plays an important role in understanding and capturing higher-order structure information of the biological networks, social networks, academic networks, and so on. Capturing the motif structure and its interaction can improve the quality of network embedding. But the current research basically

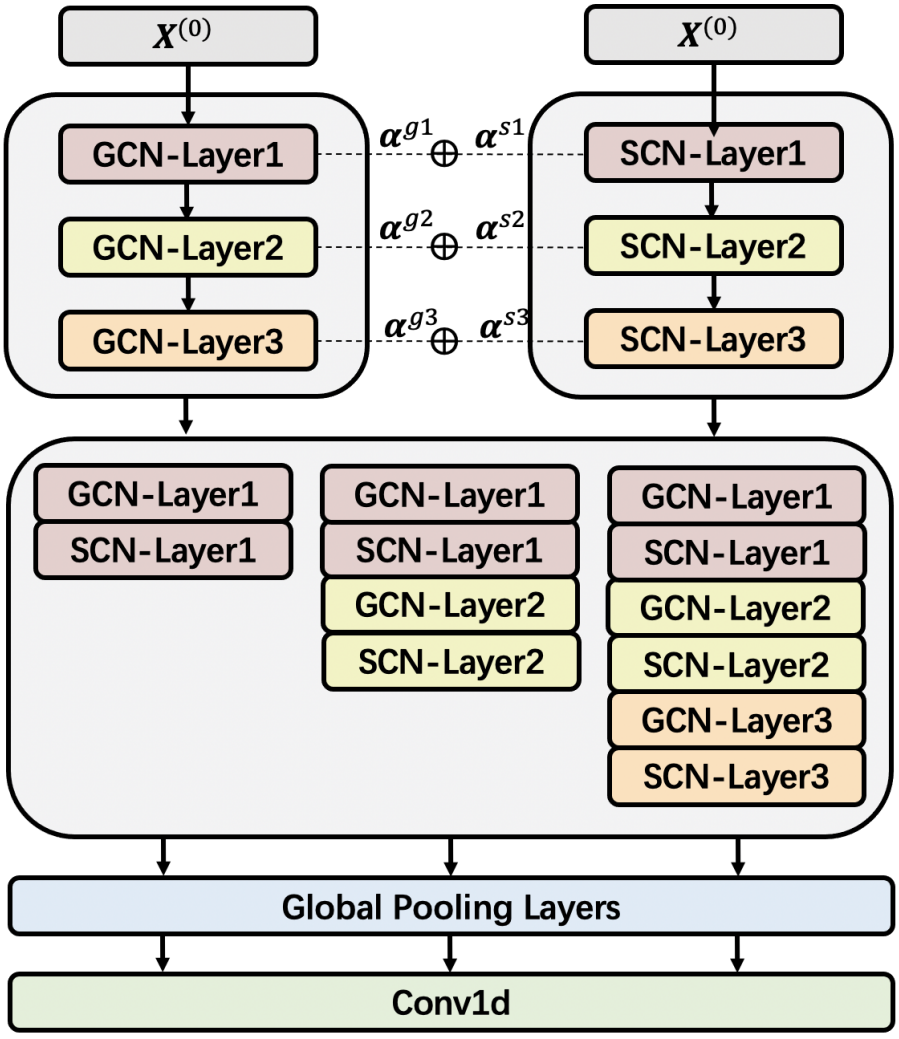

Fig. 1. We use SCN and GCN to perform 3-layer convolution on graph respectively, and splice the convolved features as shown in the figure. Then the spliced features are sorted into a grid structure, which can be directly subjected to 1-dimensional convolution.

ignores the capture and application of Motif. Several common motifs are shown in Fig. 2.

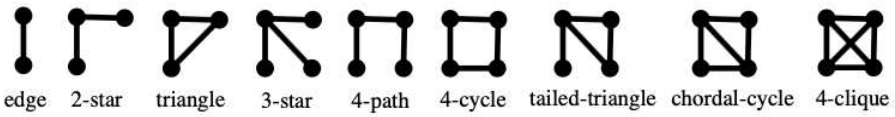

Fig. 2. Several Common Motifs

3 Proposed LGGNN model

In this section, we first give the overall framework of the proposed model LGL. Subsequently, we introduce the subgraph convolutional layer based on motif, and the feature fusion between subgraph convolutional layer and deep convolutional layer based on attention mechanism.

A. Framework We develop a novel hybrid convolution structure based on GCN and SCN. Specifically, both GCN and SCN performed the convolution operation three times, and we spliced the features of the convolutional layer as shown in Fig. 1. Each node in SCN only aggregates adjacent motif neighbors, representing the local node information of the graph, and GCN is based on the one-hop neighbor propagation of the node, including the global topological characteristics of the graph. Hybrid graph convolution operation can be described by the following formula:

$$\mathbf{Z^{(l)}} = f(\alpha^{g(l)}\tilde{\mathbf{D}}^{-1}\tilde{\mathbf{A}}\mathbf{X}\mathbf{W} \oplus \alpha^{s(l)}\tilde{\mathbf{D}_{\mathbf{m}}}^{-1}\mathbf{M}\mathbf{X}\mathbf{W_m}), \tag{2}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ means that graph is added to the self-loops, $\mathbf{M}$ is motif adjacency matrix, $\tilde{\mathbf{W}} \in \mathbb{R}^{c \times c'}$ and $\tilde{\mathbf{W}}_m \in \mathbb{R}^{c \times c'}$ are matrix of trainable convolution parameters, $\alpha^{g(l)}$ and $\alpha^{s(l)}$ are the attention weights of the GCN and SCN layers, f is a nonlinear activation function, and $\mathbf{Z}$ is the output after convolution operation.

B. Motif-based SCN In order to better learn the strong connection relationship between nodes, we introduced motif and designed SCN based on motif. Specifically, SCN includes four key steps: (1) rank nodes according to their degree; (2) find neighbors based on motif for each node; (3) map the subgraph to the tree: construct a m-ary tree for each node. The leaf nodes of the i-level m-ary tree are replaced by the neighboring m-ary trees, and a K-level m-ary tree is recursively constructed for each node; (4) arrange the tree into a regular grid structure.

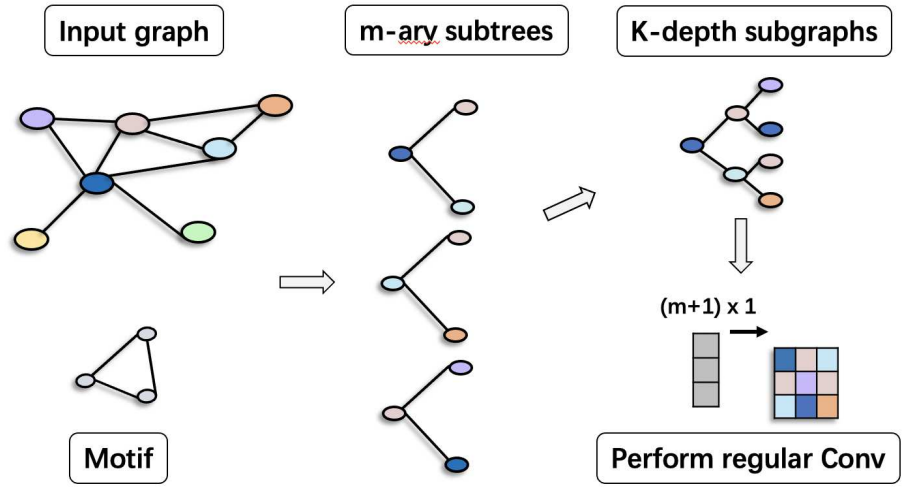

Fig. 3. For the root node (dark blue), find its motif neighbor (light blue, pink) as the leaf node of the root node. Then take the leaf node as the root node, continue to find the leaf node based on the motif pattern, and iterate continuously until it grows into a K-layer m-ary tree. Finally, the trees are arranged into a grid structure.

C. Attention Layer Similar to the attention equation mentioned by Vaswan et al. [14] ,we introduce attention to the splicing process of GCN layer and SCN layer:

$$\text{Attention}^{(l)}(\mathbf{S}^{(l)}, \mathbf{G}^{(l)}) = \text{softmax}(\mathbf{S}^{(l)}\mathbf{W}_s^{(l)} \oplus \mathbf{G}^{(l)}\mathbf{W}_g^{(l)}) \tag{3}$$

where $\mathbf{S}^{(l)} \in \mathbb{R}^{N\times d}$ is the last output of SCN, $D^{(l)} \in \mathbb{R}^{N\times d}$ is the last output of GCN, d denotes the dimensions of each vertex and $\mathbf{W}_s^{(l)} \in \mathbb{R}^{d\times d}$ and $\mathbf{W}_g^{(l)} \in \mathbb{R}^{d\times d_{out}}$ are two learnable matrices.

4 Experimental Results

We set up the experiments on benchmark datasets to evaluate the solid performance of the proposed LGL model against both state-of-the-art graph kernels and other deep learning methods on graph classification problems.

Datasets We conducted the experiments using three bioinformatics datasets: MUTAG, PROTEINS, PTC-MR and one social networks datasets: IMDB-B. The details of the datasets are shown in Table.1.

Table 1. Information of the Benchmark Datasets

Datasets	Graphs	Classes	Avg.Nodes	Avg.Edges	Labels	Description
MUTAG	188	2	17.93	19.79	7	Bioinformatics
PROTEINS	1113	2	39.06	72.82	3	Bioinformatics
PTC-MR	344	2	14.29	14.69	19	Bioinformatics
IMDB-B	1000	2	19.77	96.53	-	Social

Experimental Setting We compare the performance of the proposed LGL model on graph classification tasks with a) four alternative state-of-the-art graph kernels and b) five alternative state-of-the-art deep learning approaches for graphs. To be specific, the graph kernels include 1) the Weisfeiler-Lehman subtree kernel (WLSK) [15], 2) the shortest path graph kernel (SPGK) [16], 3) the random walk graph kernel (RWGK) [17], and 4) the graphlet count kernel (GK) [18]. The deep learning methods include 1) the deep graph convolutional neural network (DGCNN) [8], 2) the quantum-based subgraph convolutional neural networks (Qs-CNNs) [9], 3) the backtrackless aligned-spatial graph convolutional networks [19], 4) the deep graphlet kernel(DGK) [20], and 5) the diffusion convolutional neural network((DCNN) [21].

Table 2. Classification Accuracy (In%± Standard Error) for Comparisons

Datasets	MUTAG	PROTEINS	PTC-MR	IMDB-B
WLSK	82.88±0.57	73.52±0.43	58.26±0.47	71.88±0.77
SPGK	83.38±0.81	75.10±0.50	55.52±0.46	71.26±1.04
RWGK	80.77±0.72	74.20±0.40	55.91±0.37	67.94±0.77
GK	81.66±2.11	71.67±0.55	52.26±1.41	65.87±0.98
DGCNN	85.83±1.66	75.54±0.94	58.59±2.47	70.03±0.86
Qs-CNNs	93.13±4.67	78.80±4.63	65.99±4.43	-
BASGCN	90.05±0.82	76.05±0.57	61.51±0.77	74.00±0.87
DGK	82.66±1.45	71.68±0.50	57.32±1.13	66.96±0.56
DCNN	66.98	61.29±1.60	58.09±0.53	49.06±1.37
LGL	**90.16**±1.39	**78.41**±0.82	**65.74**±1.80	**66.51**±1.51

For the evaluation, we adjust a number of hyperparameters to get the best performance of each dataset, as shown in Table.3 . In SCN, we set up two motifs, triangle and 4-cycle, to capture the neighbors of nodes, and construct 2-ary and 3-ary trees respectively. For our model, we perform 10-fold cross-validation to compute the classification accuracy, with nine training folds and one validating fold. For each dataset, we repeat the experiment 10 times and report the average classification accuracy and standard errors in Table.2.

For the alternative graph kernels and deep learning methods except Qs-CNNs, we report the best results collected and experimented by Bai et al. [22]. We report the best results for Qs-CNNs from the original paper [9]. Classifica-

tion accuracy and standard error of each competing approach are also shown in Table.2.

Table 3. Hyperparameters settings for each dataset.

Parameters	K	m	Motif	conv	conv1d	fc	fc_num	batch	lr	L2norm	dropout
MUTAG	5	2	triangle	256	32	64	2	64	0.01	0	0.1
PROTEINS	4	3	4-cycle	32	32	256	3	16	0.0003	0	0
PTC-MR	4	3	4-cycle	256	32	128	2	32	0.01	0	0
IMDB-B	4	2	triangle	256	32	64	2	64	0.01	0	0

Experimental Results and Discussions Table.2 indicates that the proposed LGL model can significantly outperform either the competing graph kernel methods or the deep learning methods for graph classification.

Overall, the reasons for the effectiveness of our method are threefold. First of all, as mentioned earlier, most deep learning methods for graph classification cannot well avoid the problems of oversmoothing and retention of rich global and local information. On the contrary, the proposed LGL can alleviate these problems and get better representation learning. Second, the graph kernels with C-SVM classifier are shallow learning methods, while the proposed LGL model can provide an end-to-end deep learning architecture. Thus LGL model can learn better graph characteristics. Third, the use of motif to extract strongly connected neighbor information for nodes simplifies the steps of quantum walk in Qs-CNNs. The splicing of the results of local subgraph convolution and deep graph convolution has achieved a performance exceeding DGCNN. This empirically proves the effectiveness of the proposed LGL model.

5 Conclusions

In this paper, we have shown how to construct motif-based subgraph convolution network for a graph and how to make use of both the global topological arrangement information and local connectivity structures within a graph. Experimental results on graph classification show our LGL model is superior to a number of baseline methods.

It is interesting to notice that different practical problems have different requirements for global and local information. For example, social networks may rely more on the near-end neighbors of nodes, but the properties of chemical molecules may depend on some remote nodes. In addition, the choice of motif also greatly affects the effect of graph classification. Our future plan is to explore the impact of more types of motifs on the experimental results, and a better way to gather global and local information.

References

1. Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.
2. Bruna J, Zaremba W, Szlam A, et al. Spectral networks and locally connected networks on graphs[J]. arXiv preprint arXiv:1312.6203, 2013.
3. Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering[C]//Advances in neural information processing systems. 2016: 3844-3852.
4. Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs[C]//Advances in neural information processing systems. 2017: 1024-1034.
5. Wang Y, Sun Y, Liu Z, et al. Dynamic graph cnn for learning on point clouds[J]. Acm Transactions On Graphics (tog), 2019, 38(5): 1-12.
6. Li Q, Han Z, Wu X M. Deeper insights into graph convolutional networks for semi-supervised learning[J]. arXiv preprint arXiv:1801.07606, 2018.
7. Oono K, Suzuki T. Graph neural networks exponentially lose expressive power for node classification[J]. arXiv preprint arXiv:1905.10947, 2019.
8. Zhang M, Cui Z, Neumann M, et al. An end-to-end deep learning architecture for graph classification[C]//Thirty-Second AAAI Conference on Artificial Intelligence. 2018.
9. Zhang Z, Chen D, Wang J, et al. Quantum-based subgraph convolutional neural networks[J]. Pattern Recognition, 2019, 88: 38-49.
10. Milo R, Shen-Orr S, Itzkovitz S, et al. Network motifs: simple building blocks of complex networks[J]. Science, 2002, 298(5594): 824-827.
11. Benson A R, Gleich D F, Leskovec J. Higher-order organization of complex networks[J]. Science, 2016, 353(6295): 163-166.
12. Yin H, Benson A R, Leskovec J, et al. Local higher-order graph clustering[C]//Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017: 555-564.
13. Lee J B, Rossi R A, Kong X, et al. Graph convolutional networks with motif-based attention[C]//Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019: 499-508.
14. Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in neural information processing systems. 2017: 5998-6008.
15. Shervashidze N, Schweitzer P, Van Leeuwen E J, et al. Weisfeiler-lehman graph kernels[J]. Journal of Machine Learning Research, 2011, 12(9).
16. Borgwardt K M, Kriegel H P. Shortest-path kernels on graphs[C]//Fifth IEEE international conference on data mining (ICDM'05). IEEE, 2005: 8 pp.
17. Kashima H, Tsuda K, Inokuchi A. Marginalized kernels between labeled graphs[C]//Proceedings of the 20th international conference on machine learning (ICML-03). 2003: 321-328.
18. Shervashidze N, Vishwanathan S V N, Petri T, et al. Efficient graphlet kernels for large graph comparison[C]//Artificial Intelligence and Statistics. 2009: 488-495.
19. Bai L, Cui L, Jiao Y, et al. Learning backtrackless aligned-spatial graph convolutional networks for graph classification[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020.
20. Yanardag P, Vishwanathan S V N. Deep graph kernels[C]//Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2015: 1365-1374.

21. Atwood J, Towsley D. Diffusion-convolutional neural networks[C]//Advances in neural information processing systems. 2016: 1993-2001.
22. Bai L, Jiao Y, Cui L, et al. Learning Aligned-Spatial Graph Convolutional Networks for Graph Classification[C]//Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Cham, 2019: 464-482.