

## Research Article

# Multiscale Receptive Fields Graph Attention Network for Point Cloud Classification

Xi-An Li <sup>1,2</sup>, Li-Yan Wang <sup>1</sup> and Jian Lu <sup>3</sup>

<sup>1</sup>Department of Stomatology, Foshan Woman and Children's Hospital, Foshan, Guangdong 528000, China

<sup>2</sup>School of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai 200240, China

<sup>3</sup>Guangdong Institute of Aeronautics and Astronautics Equipment and Technology, Zhuhai 519000, China

Correspondence should be addressed to Xi-An Li; [lixa0415@sjtu.edu.cn](mailto:lixa0415@sjtu.edu.cn), Li-Yan Wang; [wangliyankmmc@163.com](mailto:wangliyankmmc@163.com), and Jian Lu; [luj@giaaet.com](mailto:luj@giaaet.com)

Received 12 September 2020; Revised 19 November 2020; Accepted 22 January 2021; Published 24 February 2021

Academic Editor: Rongxin Cui

Copyright © 2021 Xi-An Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Understanding the implication of point cloud is still challenging in the aim of classification or segmentation for point cloud due to its irregular and sparse structure. As we have known, PointNet architecture as a ground-breaking work for point cloud process can learn shape features directly on unordered 3D point cloud and has achieved favorable performance, such as 86% mean accuracy and 89.2% overall accuracy for classification task, respectively. However, this model fails to consider the fine-grained semantic information of local structure for point cloud. Then, a multiscale receptive fields graph attention network (named after MRFGAT) by means of semantic features of local patch for point cloud is proposed in this paper, and the learned feature map for our network can well capture the abundant features information of point cloud. The proposed MRFGAT architecture is tested on ModelNet datasets, and results show it achieves state-of-the-art performance in shape classification tasks, such as it outperforms GAPNet (Chen et al.) model by 0.1% in terms of OA and compete with DGCNN (Wang et al.) model in terms of MA.

## 1. Introduction

Point cloud as a simple and efficient representation for 3D shapes and scenes has become more and more popular in the fields of both academia and industry. For example, autonomous vehicle [1–4], robotic mapping and navigation [5–7], 3D shape representation and modelling [8, 9], and other relevant applications [10–15]. Lots of ways can be used to obtain 3D point cloud data, such as utilizing 3D scanners including physical touch or noncontact measurements with light, sound, LiDAR, etc.

Up to now, a variety of approaches have been developed to handle this kind of data, such as the commonly used traditional handcraft algorithms [16–18]. In terms of these methods, it is significant to classify or segment point cloud by choosing salient features of point cloud, such as normals, curvatures, and colors. Handcrafted features are usually employed to address specific problems but tough to transfer

to new tasks. Then, it is a hot topic, in last decades, that how to overcome the shortcomings for traditional methods.

With the development of deep learning, some existed end-to-end neural networks have overcome many challenges' stem from 3D data and made great breakthrough for point cloud, see Figure 1. In particular, the modificatory works of convolutional neural networks (CNNs) have achieved significant success for point cloud data in computer vision tasks, such as PointNet [19] and its improved version [20], PointCNN [21, 22], and PointSift [23]. Unfortunately, lots of neural networks for point cloud only capture global feature without local information which are also an import semantic feature for point cloud. Hence, exploiting reasonably the local information of point cloud has become a new research hotspot, and some valuable works also have sprung up recently. PointNet++ [20] extends the PointNet model by constructing a hierarchical neural network that recursively applies PointNet with designed sampling and

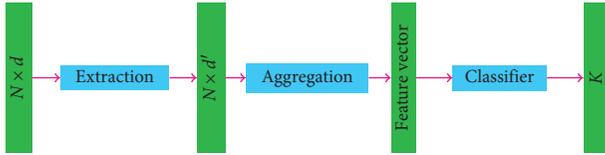


FIGURE 1: A typical framework of a 3D point cloud classification process. The model takes  $N$  points in  $d$ -dimensional space as its input and then extracts point features of dimension  $d'$ , followed by an aggregation module used to form a feature vector which is invariant to point permutation. Finally, a classifier is used to classify the resulting feature vector into  $K$  categories.

grouping layers to extract local features. Graph neural networks [24, 25] can not only directly address a more general class of graphs, e.g., cyclic, directed, and undirected graphs, but also be applied to deal with point cloud data. Recently, DGCNN [26] and its variant [27] well utilized the graph network with respect to the edges' convolution on points and then obtained the local edges' information of point cloud. Other relevant works applying the graph structure of point cloud can be found in [28–30].

Attention mechanism plays a significant role in machine translation task [31], vision-based task [32], and graph-based task [33]. Combining graph structure and attention mechanism, some favorable network architectures are constructed which leverage well the local semantic features of point cloud. Readers can refer to [34–36].

However, the scale of different graphs for the existed graph networks are fixed; then, the semantic expression of the point will not be good. Hence, in this work, inspired by graph attention network [33], graph convolution network [37], and local contextual information networks, we design a multiscale receptive fields' graph attention network for point cloud classification. Unlike previous models that only consider the attribute information such as coordinate of each single point or only exploit local semantic information of point, we pay attention to the spatial context information of both local and global structure for point cloud. Finally, like the standard convolution in grid domain, our model can also be efficiently implemented for the graph representation of a point cloud.

The key contributions of our work are summarized as follows:

- (i) We construct graph of local patch for point cloud and then enhance the feature representation of point in point cloud by combining edges' information and neighbors' information
- (ii) We introduce a multiscale receptive fields' mechanism to capture the local semantic features in various ranges for point cloud
- (iii) We balance the influence between neighbors and centroid in the local graph by means of attention mechanism
- (iv) We release our code to facilitate reproducibility and future research (<https://github.com/Blue-Giant/MRFGAT-NET>)

The rest parts of this paper are structured as follows. In Section 2, we review the most closely related literatures on point

cloud. In Section 3, we introduce our proposed MRFGAT architecture and provide the details of our framework in terms of shape classification for point cloud. We describe the dataset and design comparison algorithms in Section 4, followed by the experiments' results and discussion. Finally, some concluding remarks are made in Section 5.

## 2. Related Works

**2.1. Pointwise MLP and Point Convolution Networks.** Utilizing the deep learning technique, the classical PointNet [19] was proposed to deal with directly unordered point clouds without using any volumetric or grid-mesh representation. The main idea of this network is as follows. At first, a Spatial Transformer Network (STN) module similar to feature-extracting process is constructed which guarantees the invariance of transformations. Then, a shared pointwise Multilayer-Perceptron (MLP) module is introduced which is used to extract semantic features from point sets. At last, the final semantic information of point cloud is aggregated by means of a max pooling layer. Due to the favorable ability to approximate any continuous function for MLP which is easy to implement by point convolution, some related works were presented according to the PointNet architecture [38, 39].

Similar to convolution operator in 2D space, some convolution kernels for points in 3D space are designed which can capture the abundant information of point cloud. PointCNN [21] used a local  $\mathcal{X}$ -transformation kernel to fulfill the invariance of permutation for points and then generalized this technique to the hierarchical form in analogy to that of image CNNs. The authors in [40–42] extended the convolution operator of 2D space, applied at individual point in local region of point cloud, and then collected the neighbors' information in the hierarchical convolution layer to the center point. Kernel Point Convolution (KPConv) [43] consists of a set of local 3D filters and overcomes stand point convolution limitation. This novel kernel structure is very flexible to learn local geometric patterns without any weights.

**2.2. Learning Local Features.** In order to overcome the shortcoming for PointNet-like networks which fail to exploit local features, some hierarchical architectures have been developed, for example, PointNet [20] and So-Net [38], to aggregate local information with MLP operation by considering local spatial relationships of 3D data. In contrast to the previous type, these methods can avoid sparsity and update dynamically in different feature dimensions. According to a Capsule Networks, 3D Capsule Convolutional Networks were developed which can learn well the local features of point cloud, see [44–46].

**2.3. Graph Convolutional Networks.** Graph Convolutional Neural Networks (GCNNs) have gained more and more attraction to address irregularly structured data, such as citation networks and social networks. In terms of 3D point cloud data, GCNNs have shown its powerful ability on classification and segmentation tasks. Using the convolution

operator with respect to the graph in the spectral domain is an important approach [47–49], but it needs to calculate a lot of parameters on polynomial or rational spectral filters [50]. Recently, many researchers constructed local graph of point cloud by utilizing each point’s neighbors in low-dimensional manifold based on  $N$ -dimensional Euclidean distance and then grouped each point’s neighbors in the form of high-dimensional vectors, such as EdgeConv-like works [26, 27, 51] and graph convolutions [37, 52]. Compared with the spectral methods, its main merit is that it is more consistent with the characteristics of data distribution. Specially, EdgeConv extracts edge features through the relationship between central point and neighbor points by successively constructing graph in the hierarchical model. To sum up, the graph convolution network combines features on local surface patches which are invariant to the deformations of patches for point cloud in Euclidean space.

**2.4. Attention Mechanism.** The idea of attention has been successfully used in natural language processing (NLP) [31] and graph-based work [33, 53]. Attention module can balance the weight relationship of different nodes in graph structure data or different parts in sequence data.

Recently, the attention idea has obtained more and more attraction and made a great contribution to point cloud processes [34, 35]. In these works, it is significant to aggregate point or edge features by means of attention module. Unlike the existing methods, we try to enhance the high-level representation of point cloud by capturing the relation of points and local information along its channel.

### 3. Our Approach

The framework of point cloud classification includes two contents: taking the 3D point cloud as input and assigning one semantic class label for each point. Based on the technique of extracting features from the local directed graph and attention mechanism, a new architecture for shape classification task is proposed to better learn point’s representation for unstructured point cloud. This new architecture is composed of three components which are the point enhancement, the feature representation, and the prediction. These three components are fully coupled together, which leads to an end-to-end training pipeline.

**3.1. Problem Statement.** At first, we let  $P = \{p_i \in \mathbb{R}^d, i = 1, 2, \dots, N\}$  represent a raw set of unordered points as the input for our mode, where  $N$  is the number of the points and  $p_i$  is a feature vector with a dimension  $d$ . In actual applications, the feature vector  $p_i$  might contain 3D space coordinates  $(x, y, z)$ , color, intensity, surface normal, etc. For the sake of simplicity, we set  $d = 3$  in our work and only take 3D coordinates of point as the feature representation for point. A classification or semantic segmentation of a point cloud are map  $\Phi_c$  or  $\Phi_s$ , respectively, which assign individual point semantic labels or point cloud semantic labels, respectively, i.e.,

$$\Phi: P \longrightarrow L^k. \quad (1)$$

Here,  $\Phi$  represents the map  $\Phi_c$  or  $\Phi_s$ . The objective of our model is finding the optimal map that can obtain accurate semantic labels.

The above map should satisfy some constraints including the following. (1) Permutation invariance: the order of points may vary but does not influence the category of the point or point cloud. (2) Transformation invariance: for the uncertain translation and rotation of point cloud, the results of classification or segmentation should not be changed for point or point cloud.

**3.2. Graph Generation for Point Cloud.** Some works indicate that local features of point cloud can be used to improve the discriminability of point; then, exploring the relationship among points in a whole sets or local patch is a keypoint for our work. Graph neural network is a feasible approach to process point cloud because it propagates on each node for the whole sets or a local patch of point cloud individually, ignores the permutation order of nodes, and then extracts the local information between nodes. To apply the graph neural network on the point cloud, we firstly convert it to a directed graph. Like DGCNN [26, 27] and GAPNet [34], we can obtain the neighbors (including self) of each point in point cloud by means of  $K$ -NN algorithm and then construct a local directed graph in Euclidean space. Figure 2 depicts the directed graph  $G = (V; E)$  of local patch for point cloud,  $V = \{1, 2, \dots, K\}$  is the vertice set of  $G$ , namely, the nodes of local patch,  $E$  stands for the edge set of  $G$ , and each edge is  $e_{ij} = p_i - p_{ij}$  with  $p_i \in P$  and  $p_{ij} \in V$  being centroid and neighbors, respectively.

**3.3. Single Receptive Field Graph Attention Layer (SRFGAT).** In order to aggregate the information of neighbors, we use a neighboring-attention mechanism which is introduced to obtain attention coefficients of neighbors for each point, see Figure 3. Additionally, edge features are important local features which can enhance the semantic expression of point; then, an edge-attention mechanism is also introduced to aggregate information of different edges, see Figure 3. In light of the attention mechanism [33, 34], we firstly transform the neighbors and edges into a high-level feature space to obtain sufficient expressive power. To this end, as an initial step, a parametric nonlinear function  $h(\cdot)$  is applied to every neighbor and edge, and the results are defined by

$$\begin{aligned} e'_{ij} &= h(e_{ij}, \theta) \in \mathbb{R}^{F'}, \\ p'_{ij} &= h(p_{ij}, \theta) \in \mathbb{R}^{F'}, \end{aligned} \quad (2)$$

respectively, where  $\theta$  is a set of learnable parameters of the filter and  $F'$  is output dimension. In our method, function  $h(\cdot)$  is set to a single-layer neural network.

It is worthwhile to noting that edges in Euclidean space not only stand for the local features but also indicate the dependency between centroid and neighbor. We then obtain attentional coefficients of edges and neighbors which are

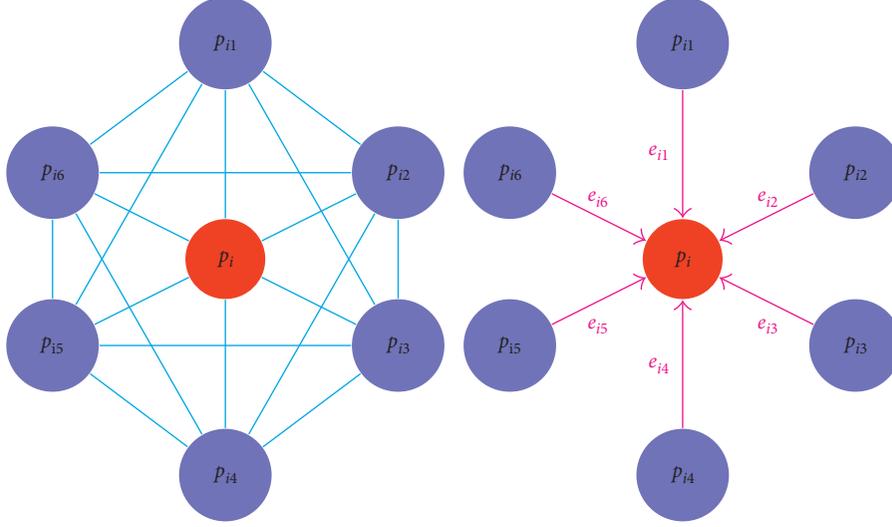


FIGURE 2: Local graph of point cloud.  $p_i$  and  $\{p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}\}$  are a central point and its neighbors, respectively. The directed edges from the neighbors to the central point are denoted by  $\{e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}\}$ .

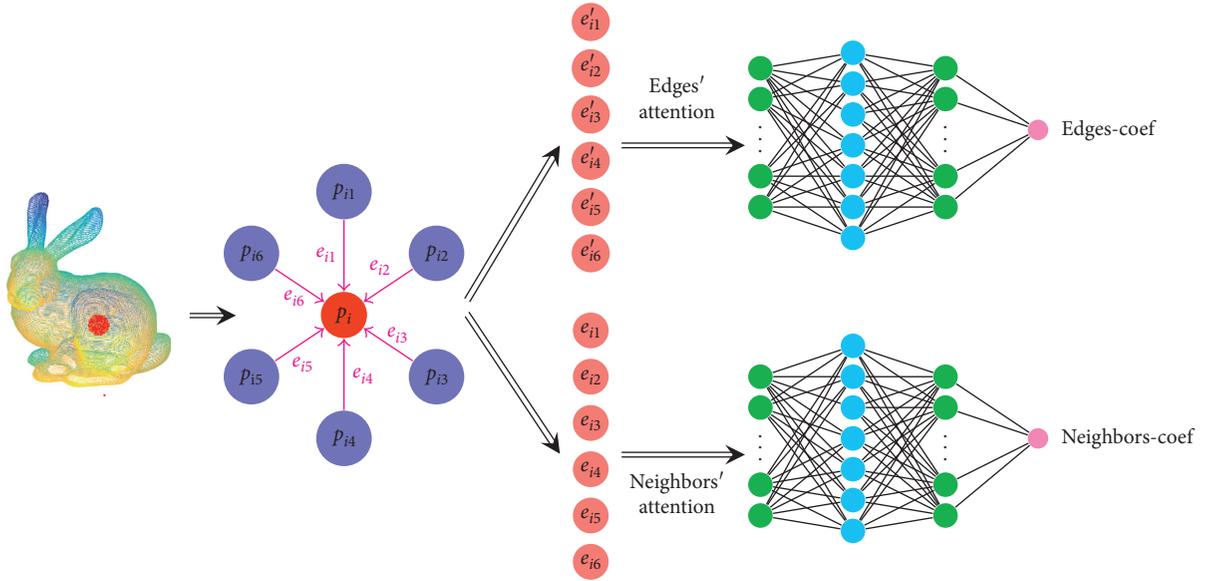


FIGURE 3: An illustration of attention coefficients' generation. The edge feature not only serves as local information to center point but also responds the effect of between neighbors and centroid. Edge attention and neighbor attention reflect the important of edge features and neighbor features to centroid, respectively.

$$a_{ij} = \text{Leakly ReLU}(g(e'_{ij}, \theta)) \text{ and } b_{ij} = \text{Leakly ReLU}(g(e_{ij}, \theta)), \quad (3)$$

respectively, where  $g(e'_{ij}, \theta)$  and  $g(e_{ij}, \theta)$  are single-layer neural network with 1-dimensional output. Leakly ReLU( $\cdot$ ) denotes nonlinear activation function leaky ReLU with  $\text{ReLU} = \max\{0, x\}$ . To make coefficients easily comparable across different neighbors and edges, we use a softmax operation to normalize the above coefficients which are defined as

$$\alpha_{ij} = \frac{\exp(a_{ij})}{\sum_k \exp(a_{ik})}, \quad (4)$$

$$\beta_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})},$$

respectively; then, the normalized coefficients are used to compute contextual feature for every point, and it is

$$\tilde{x}_i = f\left(\left(\sum_j \alpha_{ij} e'_{ij}\right) \parallel \left(\sum_j \beta_{ij} d'_{ij}\right)\right), \quad (5)$$

where  $f(\cdot)$  is a nonlinear activation function and  $\parallel$  is concatenation operation. In our model, we choose ReLU function as  $f(\cdot)$ .

**3.4. Multiscale Receptive Fields Graph Attention Layer (MRFGAT).** In order to obtain sufficient feature information and stabilize the network, the multiscale receptive field strategy analogous to multiheads mechanism is proposed, see Figure 4. Unlike previous works, the sizes of receptive fields in our model are different for various branches. Therefore, we concatenate  $M$  independent SRFGAT module and generate a semantic feature with  $M \times F^l$  channels:

$$\tilde{x}_i = \parallel_{m=1}^M \tilde{x}_i^{(m)}, \quad (6)$$

where  $\tilde{x}_i^{(m)}$  is the receptive field feature of the  $m$ th branch,  $M$  is the total number of branches, and  $\parallel$  is the concatenation operation over feature channels.

**3.5. MRFGAT Architecture.** Our MRFGAT model shown in Figure 5 considers shape classification task for point cloud. The architecture is similar to PointNet [19]. However, there are three main differences between the architectures of MRFGAT and PointNet. Firstly, according to the analyses of LinkDGCNN model, we remove the transformation network which is used in many architectures such as PointNet, DGCNN, and GAPNet. Secondly, instead of only processing individual points of point cloud, we also exploit local features by a SRFGAT-layer before the stacked MLP layers. Thirdly, an attention pooling layer is used to obtain local feature information that is connected to the intermediate layer for forming a global descriptor. In addition, we aggregate individually the original edge feature of every SRFGAT channel and then obtain local features which can enhance the semantic feature of MRFGAT.

## 4. Experiments

In this section, we evaluate our MRFGAT model on 3D point cloud analysis for the classification tasks. To demonstrate effectiveness of our model, we then compare the performance for our model to recent state-of-the-art methods and perform ablation study to investigate different design variations.

### 4.1. Classification

**4.1.1. Dataset.** We demonstrate the feasibility and effectiveness of our model on the ModelNet dataset such as ModelNet40 benchmarks [54] for shape classification. The ModelNet40 dataset contains 12,311 meshed CAD models

that are classified to 40 man-made categories. In this work, we divide the ModelNet40 dataset into two parts: the part one is named as training set which includes 9843 models and the part two is called as testing set includes 2468 models. Then, we normalize the models in the unit sphere and uniformly sample 1,024 points over model surface. Besides, we further augment the training dataset by randomly rotating, scaling the point cloud, and jittering the location of every point by means of Gaussian noise with zero mean and 0.01 standard deviation for all the models.

**4.1.2. Implementation Details.** According to the analysis of the LinkDGCNN model [27], we omit the spatial transformation network to align the point cloud to a canonical space. The network employs four SRFGAP layer modules with (8, 16, 16, 24) channels to capture attention features, respectively. Then, four shared MLP layers with sizes (128, 64, 64, 64), respectively, followed by it are used to aggregate the feature information. Next, the output features are fed into an aggregation operation followed by the MLP layer with 1024 neurons. In the end of network, a max pooling operation and two full-connected layers (512, 256) are used to finally obtain the classification score. The training is carried out using Adam optimizer with minibatch training (batch size of 16) and an initial learning rate of 0.001. The ReLU activate function and Batch Normalization (BN) are also used in both the SRFGAP module and MLP layer. At last, the network was implemented using TensorFlow and executed on the server equipped with four NVIDIA GTX2080Ti.

**4.1.3. Results.** Figures 6–8 depict the process for training and testing. From the figures, we see that our model will quickly attain the stage of high accuracy, which means our model is highly efficient. Table 1 lists the results of our method and several recent state-of-the-art works. The methods listed in Table 1 have one thing in common. The input is only raw point cloud with 3D coordinates  $(x_i, y_i, z_i)$ . Based on these results, we can conclude that our model performs better than other methods and obtains wonderful performance on both the ModelNet40 benchmark. Compared to other point-based methods, the performance for our model is only a little weaker than that of DGCNN in terms of MA on ModelNet 40. However, it outperforms the previous state-of-the-art model GAPNet by 0.1% accuracy in terms of OA. These phenomena show that the strategy employing local and global features in different receptive fields is efficient, and it will help us to capture the prominent semantic feature for point cloud. And, in our model, since we introduce the structure of the data by providing the local interconnection between points and explore graph features from different scale field levels by the localized graph convolutional layers, it guarantees the exploration of more distinctive latent representations for each object class.

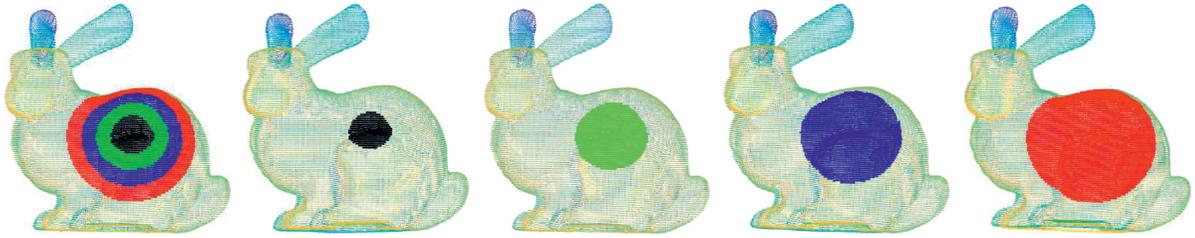


FIGURE 4: Multiscale receptive fields.

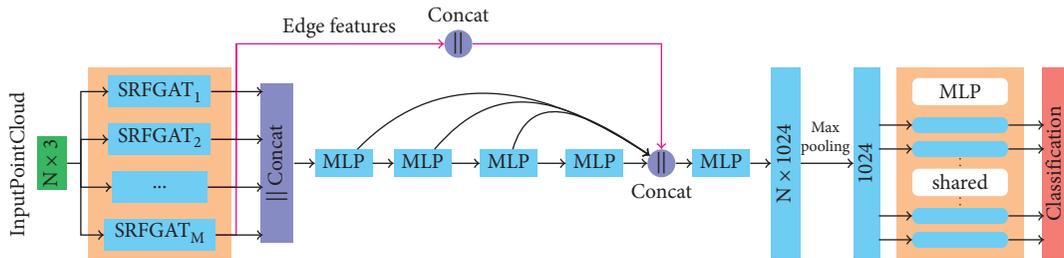


FIGURE 5: The architecture of classification. In this framework, it takes  $N$  points as input and applies  $M$  individual SRFGAT modules to obtain multiattention features on multilocal graphs; then, the output features are recast by means of five shared MLP layers and attention pooling layer, respectively. Finally, a shared full-connected layer is employed to form a global feature and then classification scores for  $c$  categories are obtained.

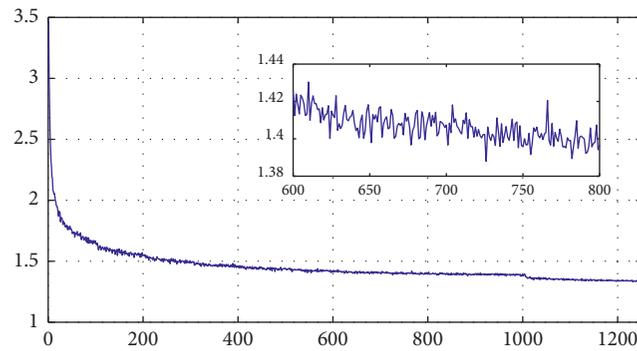


FIGURE 6: The curve of loss varying with epochs for ModelNet40, and it includes five training in one epoch.

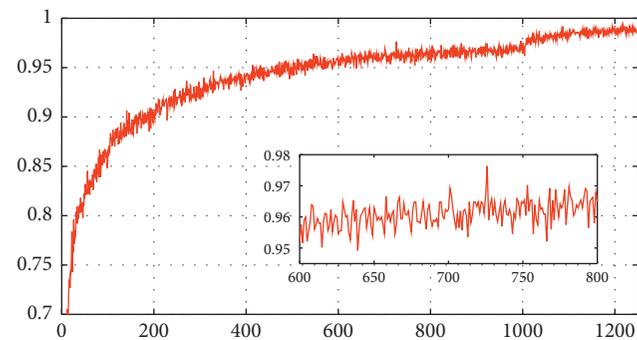


FIGURE 7: The curve of training accuracy varying with epochs for ModelNet40, and it includes five training in one epoch.

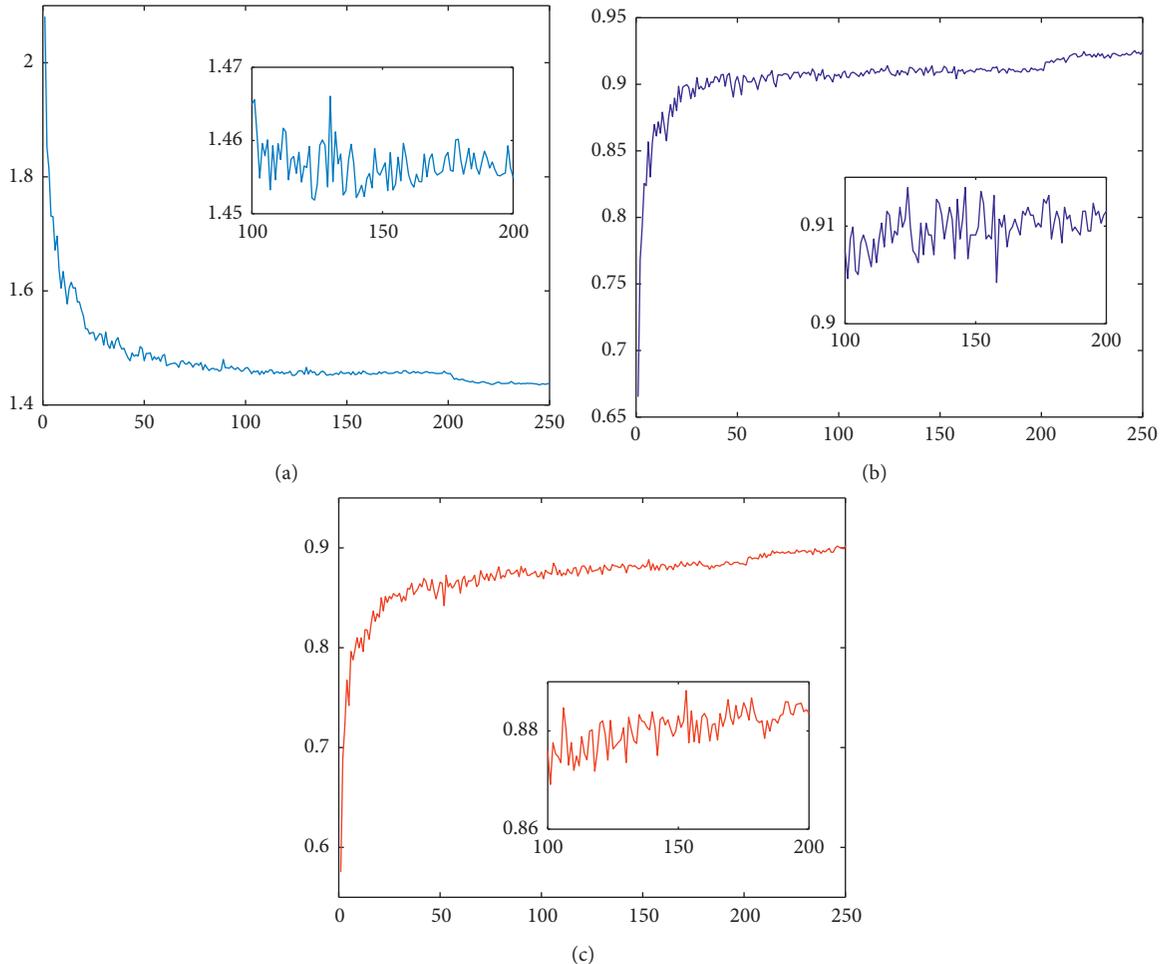


FIGURE 8: The curves of testing loss and accuracy varying with epochs for ModelNet40. (a) Loss. (b) Accuracy. (c) Mean accuracy.

TABLE 1: Classification results on ModelNet40. MA represents mean per-class accuracy, and the per-class accuracy is the ratio of the number of correct classifications to that of the objects in a class. OA denotes the overall accuracy, which is the ratio of the number of overall correct classifications to that of overall objects.

Method	Points	MA (%)	OA (%)
SO-Net [38]	$2048 \times 3$	87.3	90.9
KD-Net [40]	$1024 \times 3$	88.5	91.8
PointNet [19]	$1024 \times 3$	86.0	89.2
PointNet++ [20]	$1024 \times 3$	—	90.7
PointCNN [21]	$1024 \times 3$	88.1	92.2
DGCNN [26]	$1024 \times 3$	90.2	92.2
PCNN [22]	$1024 \times 3$	—	92.3
GAPNet [34]	$1024 \times 3$	89.7	92.4
Ours	$1024 \times 3$	90.1	92.5

## 5. Conclusion

Enlightening by graph convolutional networks for the task of classification in 3D computer vision, we design a novel MRFGAT-based modules for point feature and context aggregation. Utilizing different receptive fields and attention strategies, the pipeline MRFGAT can capture

more fine features of point clouds for classification task. In addition, we list some comparable results with recent works which show that our model can achieve the state-of-the-art performance on the dataset ModelNet for classification task of point clouds; it outperforms the GAPNet model by 0.1 % in terms of OA and competes with the DGCNN model in terms of MA. It is necessary to point out that our model will have some burden for constructing varying scale graphs. Based on the state-of-the-art Graph Convolution Networks (GCN) for semantic segmentation in point cloud, it would be interesting to introduce our model to address this problem for unstructured data in the future.

## Data Availability

This dataset used in this manuscript is available at [https://shapenet.cs.stanford.edu/media/modelnet40\\_ply\\_hdf5\\_2048.zip](https://shapenet.cs.stanford.edu/media/modelnet40_ply_hdf5_2048.zip).

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by GDAS' Project of Science and Technology Development (no. 2018GDASCX-0804) and Project of Guangdong Engineering Technology Research Center (no. 810115228131).

## References

- [1] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018*, pp. 918–927, Salt Lake City, UT, USA, June 2018.
- [2] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3D proposal generation and object detection from view aggregation," in *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–8, Madrid, Spain, October 2018.
- [3] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3D object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 663–678, Munich, Germany, September 2018.
- [4] C. Yang, C. Chen, W. He, R. Cui, and Z. Li, "Robot learning system based on adaptive neural control and dynamic movement primitives," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 777–787, 2019.
- [5] J. Biswas and M. Veloso, "Depth camera based indoor mobile robot localization and navigation," in *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*, pp. 1697–1702, St Paul, MIN, USA, May 2012.
- [6] Y. Zhu, R. Mottaghi, E. Kolve et al., "Target-driven visual navigation in indoor scenes using deep reinforcement learning," 2017, <http://arxiv.org/abs/1609.05143>.
- [7] H. Wang, S. Wang, J. Yao, R. Pan, and J. Yang, Effective Anti-collision Algorithms for RFID Robots System, Assembly Automation Ahead-Of-Print (2019).
- [8] A. Golovinskiy, V. G. Kim, and T. Funkhouser, "Shape-based recognition of 3D point clouds in urban environments," in *Proceedings of the 2009 IEEE 12th International Conference on Computer Vision*, Kyoto, Japan, October 2009.
- [9] P. Gu, F. Zhou, D. Yu, F. Wan, W. Wang, and B. Yu, "A 3D reconstruction method using multisensor fusion in large-scale indoor scenes," *Complexity*, vol. 2020, Article ID 6973790, 14 pages, 2020.
- [10] H. Qiao, Y. Li, T. Tang, and P. Wang, "Introducing memory and association mechanism into a biologically inspired visual model," *IEEE Transactions on Cybernetics*, vol. 44, pp. 1485–1496, 2014.
- [11] H. Qiao, M. Wang, J. Su, S. Jia, and R. Li, "The concept of "attractive region in environment" and its application in high-precision tasks with low-precision systems," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 5, pp. 2311–2327, 2015.
- [12] C. Yang, H. Wu, Z. Li, W. He, N. Wang, and C.-Y. Su, "Mind control of A robotic arm with visual fusion Technology," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 3822–3830, 2018.
- [13] C. Yang, C. Zeng, C. Fang, W. He, and Z. Li, "A DMPs-based framework for robot learning and generalization of humanlike variable impedance skills," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, pp. 1193–1203, 2018.
- [14] Z. Zhao, C. K. Ahn, and H.-X. Li, "Boundary antidisturbance control of a spatially nonlinear flexible string system," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 6, pp. 4846–4856, 2020.
- [15] Z. Zhao, C. K. Ahn, and H.-X. Li, "Dead zone compensation and adaptive vibration control of uncertain spatial flexible riser systems," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 3, pp. 1398–1408, 2020.
- [16] G. Vosselman and S. Dijkman, "3D building model reconstruction from point clouds and ground plans," in *Proceedings of the ISPRS Workshop: Land Surface Mapping and Characterization Using Laser Altimetry*, pp. 37–43, Annapolis, MD, USA, October 2001.
- [17] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 2009.
- [18] F. Tombari, S. Salti, and L. D. Stefano, "Unique signatures of histograms for local surface description," in *Proceedings of the 11th European Conference on Computer Vision ECCV 2010*, Heraklion, Crete, Greece, September 2010.
- [19] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: deep learning on point sets for 3D classification and segmentation," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85, Honolulu, HI, USA, July 2017.
- [20] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: deep hierarchical feature learning on point sets in a metric space," 2017, <http://arxiv.org/abs/1706.02413>.
- [21] M. S. W. W. X. D. Yangyan Li, R. Bu, and B. Chen, "PointCNN: convolution on  $\mathcal{X}$ -transformed points," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 828–838, Denver, CO, USA, December 2018.
- [22] M. Atzmon, H. Maron, and Y. Lipman, "Point convolutional neural networks by extension operators," *International Conference on Computer Graphics and Interactive Techniques*, vol. 37, p. 71, 2018.
- [23] M. Jiang, Y. Wu, T. Zhao, Z. Zhao, and C. Lu, "PointSIFT: a SIFT-like network module for 3D point cloud semantic segmentation," 2018, <http://arxiv.org/abs/1807.00652>.
- [24] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings of the 2005 IEEE International Joint Conference on Neural Networks*, pp. 729–734, Montreal, Que, Canada, December 2005.
- [25] F. Scarselli, M. Gori, A. C. Ah Chung Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.
- [26] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Transactions on Graphics*, vol. 38, p. 146, 2019.
- [27] K. Zhang, M. Hao, J. Wang, C. W. de Silva, and C. Fu, "Linked dynamic graph CNN: learning on point cloud via linking hierarchical features," 2019, <http://arxiv.org/abs/1904.10014>.
- [28] G. Te, W. Hu, Z. Guo, and A. Zheng, "RGCNN: regularized graph CNN for point cloud segmentation," 2018, <http://arxiv.org/abs/1806.02952>.
- [29] X. Gao, W. Hu, and Z. Guo, "Exploring structure-adaptive graph learning for robust semi-supervised classification," 2020, <http://arxiv.org/abs/1904.10146>.
- [30] Q. Lu, C. Chen, W. Xie, and Y. Luo, "PointNGCNN: deep convolutional networks on 3D point clouds with neighborhood graph filters," *Computers & Graphics*, vol. 86, pp. 42–51, 2020.

- [31] A. Vaswani, N. Shazeer, N. Parmar et al., “Attention is all you need,” 2017, <http://arxiv.org/abs/1706.03762>.
- [32] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, “Recurrent models of visual attention,” 2014, <http://arxiv.org/abs/1406.6247>.
- [33] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” 2018, <http://arxiv.org/abs/1710.10903>.
- [34] C. Chen, L. Z. Fragonara, and A. Tsourdos, “GAPNet: graph attention based point neural network for exploiting local feature of point cloud,” 2019, <http://arxiv.org/abs/1905.08705>.
- [35] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, “Graph attention convolution for point cloud semantic segmentation,” in *Proceedings of the Computer Vision and Pattern Recognition*, pp. 10296–10305, Long Beach, CA, USA, January 2019.
- [36] M. Feng, L. Zhang, X. Lin, S. Z. Gilani, and A. Mian, “Point Attention network for semantic segmentation of 3D point clouds,” 2019, <http://arxiv.org/abs/1909.12663>.
- [37] M. Niepert, M. Ahmed, and K. Kutzkov, “Learning convolutional neural networks for graphs,” 2016, <http://arxiv.org/abs/1605.05273>.
- [38] J. Li, B. M. Chen, and G. H. Lee, “SO-Net: self-organizing network for point cloud analysis,” in *Proceedings of the Computer Vision and Pattern Recognition*, pp. 9397–9406, Salt Lake City, UT, USA, June 2018.
- [39] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, P.-A. Heng, and PU-Net, “Point cloud upsampling network,” in *Proceedings of the Computer Vision and Pattern Recognition*, pp. 2790–2799, Salt Lake City, UT, USA, June 2018.
- [40] R. Klokov and V. Lempitsky, “Escape from cells: deep kd-networks for the recognition of 3D point cloud models,” in *Proceedings of the Computer Vision and Pattern Recognition*, pp. 863–872, Honolulu, HI, USA, July 2017.
- [41] Y. You, Y. Lou, Q. Liu et al., “Pointwise rotation-invariant network with adaptive sampling and 3D spherical voxel convolution,” 2018, <http://arxiv.org/abs/1811.09361>.
- [42] W. Wu, Z. Qi, and L. Fuxin, “PointConv: deep convolutional networks on 3D point clouds,” in *Proceedings of the Computer Vision and Pattern Recognition*, pp. 9621–9630, Long Beach, CA, USA, June 2019.
- [43] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, “KPConv: flexible and deformable convolution for point clouds,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 6411–6420, Long Beach, CA, USA, June 2019.
- [44] Y. Zhao, T. Birdal, H. Deng, and F. Tombari, “3D point Capsule networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1009–1018, Long Beach, CA, USA, June 2019.
- [45] N. Srivastava, H. Goh, and R. Salakhutdinov, “Geometric capsule autoencoders for 3D point clouds,” 2019, <http://arxiv.org/abs/1912.03310>.
- [46] A. Cheraghian and L. Petersson, “3DCapsule: extending the capsule architecture to classify 3D point clouds,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1194–1202, Long Beach, CA, USA, June 2019.
- [47] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, “SpiderCNN: deep learning on point sets with parameterized convolutional filters,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 87–102, Munich, Germany, September 2018.
- [48] D. Boscaini, J. Masci, S. Melzi, M. M. Bronstein, U. Castellani, and P. Vandergheynst, “Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks,” *Proceedings of the Eurographics Symposium on Geometry Processing*, vol. 34, pp. 13–23, 2015.
- [49] L. Yi, H. Su, X. Guo, and L. Guibas, “SyncSpecCNN: synchronized spectral CNN for 3D shape segmentation,” in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6584–6592, Honolulu, HI, USA, July 2017.
- [50] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Proceedings of the 30th International Conference on Neural Information Processing Systems NIPS’16*, pp. 3844–3852, Barcelona, Spain, December 2016.
- [51] H. Xiu, T. Shinohara, and M. Matsuoka, “Dynamic-scale graph convolutional network for semantic segmentation of 3d point cloud,” in *Proceedings of the 2019 IEEE International Symposium on Multimedia (ISM)*, pp. 271–2717, Diego, CA, USA, December 2019.
- [52] N. Verma, E. Boyer, and J. Verbeek, “FeaStNet: feature-steered graph convolutions for 3D shape analysis,” 2018, <http://arxiv.org/abs/1706.05206>.
- [53] J. B. Lee, R. A. Rossi, S. Kim, N. K. Ahmed, and E. Koh, “Attention models in graphs,” *ACM Transactions on Knowledge Discovery From Data*, vol. 13, no. 6, pp. 1–25, 2019.
- [54] Z. Wu, S. Song, A. Khosla et al., “3D ShapeNets: a deep representation for volumetric shapes,” in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1912–1920, Boston, MA, USA, June 2015.