

Point Cloud Classification Network Based on Graph Convolution and Fusion Attention Mechanism

Tengteng Song, Zhao Li*, Zhenguo Liu, Yizhi He

School of Computer Science and Technology, Shandong University of Technology, Zibo, China
Email: s2817407746@163.com, *lizhao_buaa@126.com

How to cite this paper: Song, T.T., Li, Z., Liu, Z.G. and He, Y.Z. (2022) Point Cloud Classification Network Based on Graph Convolution and Fusion Attention Mechanism. *Journal of Computer and Communications*, 10, 81-95.
<https://doi.org/10.4236/jcc.2022.109006>

Received: August 27, 2022

Accepted: September 19, 2022

Published: September 22, 2022

Copyright © 2022 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution-NonCommercial International License (CC BY-NC 4.0).
<http://creativecommons.org/licenses/by-nc/4.0/>



Open Access

Abstract

The classification of point cloud data is the key technology of point cloud data information acquisition and 3D reconstruction, which has a wide range of applications. However, the existing point cloud classification methods have some shortcomings when extracting point cloud features, such as insufficient extraction of local information and overlooking the information in other neighborhood features in the point cloud, and not focusing on the point cloud channel information and spatial information. To solve the above problems, a point cloud classification network based on graph convolution and fusion attention mechanism is proposed to achieve more accurate classification results. Firstly, the point cloud is regarded as a node on the graph, the k-nearest neighbor algorithm is used to compose the graph and the information between points is dynamically captured by stacking multiple graph convolution layers; then, with the assistance of 2D experience of attention mechanism, an attention mechanism which has the capability to integrate more attention to point cloud spatial and channel information is introduced to increase the feature information of point cloud, aggregate local useful features and suppress useless features. Through the classification experiments on ModelNet40 dataset, the experimental results show that compared with PointNet network without considering the local feature information of the point cloud, the average classification accuracy of the proposed model has a 4.4% improvement and the overall classification accuracy has a 4.4% improvement. Compared with other networks, the classification accuracy of the proposed model has also been improved.

Keywords

Graph Convolution Neural Network, Attention Mechanism, Modelnet40, Point Cloud Classification

1. Introduction

In recent years, with the rapid development of 3D laser scanning technology, the acquisition of 3D point cloud data has become more and more convenient. Like image data, lidar point cloud data has gradually become basic data for deep learning. For 3D point cloud data, light, temperature and other external factors will not affect it, with rich geometry, scale, shape and other spatial information. The classification of point cloud data is the key technology for the acquisition of point cloud data information and the reconstruction of 3D model reconstruction. Through the classification of point cloud data, we can divide the disorderly point cloud data into multiple categories. It has a wide range of application prospects in the fields of automatic driving [1], road marking line classification [2], face recognition [3], three-dimensional reconstruction [4], and forest survey [5]. Due to the density distribution nonuniformity and scattered data of 3D point cloud data, some classification methods based on traditional methods cannot be directly applied to the point cloud field, which brings some challenges to the point cloud classification task.

With the development of deep learning technology, the use of deep learning methods to study 3D point clouds has become a major trend [6]. Compared to manual feature extraction [7] [8] [9], deep learning-based methods can automatically extract high-level features from large amounts of data. To be specific, incipient methods were to convert the original point cloud into a specific intermediate representation [10] [11] [12] [13] [14] (such as an image or voxel) based on the corresponding projection relationship, then use conventional 2D or 3D CNN to learn high-dimensional features for subsequent analysis. For example, Voxnet [14] uses the occupied grid algorithm to represent the point cloud data as multiple 3D grids, voxelizes the point cloud in the grid, and then uses 3D CNN to learn features. MVCNN [15] projects the point cloud from multiple perspectives, renders the point cloud into multiple views, then fuses the feature information of multiple views through the convolution layer and the pooling layer, and finally inputs the fused features into the fully connected layer to obtain the classification results.

However, due to the disorder, sparsity and non-structural characteristics of point clouds [16], whether it is multi-view processing or voxelization of point clouds, some point cloud information is lost in the process of point cloud conversion, which causes the limitation of network classification performance. The method of using multilayer perceptron (MLP) to process point cloud data was originally proposed in the PointNet [17] network, which achieved good performance on both classification tasks [18] and segmentation tasks [19]. PointNet network can directly use the point cloud data as input not require transformation. The network learns the characteristics of a single point through MLP and uses a symmetric function to solve the disorder of the point cloud. The three-dimensional Spatial Transformer Networks (STNs) are used to solve the rotation invariance of the point cloud, and the maximum pool aggregation point feature is used to

solve the permutation invariance of the point cloud. Since PointNet only focuses on the information of single point and global point when extracting point cloud features, it cannot obtain local feature information, which causes the poor classification ability of PointNet for fine-grained models, Qi *et al.* [20] proposed the optimization network PointNet++. The PointNet++ network introduces a multi-level structure composed of abstract layers on the basis of the PointNet network, and extracts local features layer by layer. It can effectively divide local point clouds and extract local features of point clouds, and can provide higher-level features for large scenes. However, the PointNet++ network is essentially the same as PointNet. It processes each individual point in the local point cloud set independently, without considering the deep feature information between point pairs.

In summary, the existing network based on deep learning methods to classify point clouds has numerous defects, such as the conversion of point clouds makes point clouds lose certain feature information, does focus on the local feature information of point clouds, and does not take into account the connection between points. These defects limit the performance of the classification network and cannot achieve better classification results in the point cloud classification task.

Since graph convolution has achieved good results in 2D image processing [21], and graph convolution can better handle unstructured data, based on the shortcomings of the above network, with the experience of PointNet network and graph convolution neural network, a point cloud classification network based on graph convolution and fusion attention mechanism is proposed. The proposed model dynamically extracts the local features of the point cloud by adding graph convolution, so that the network can focus on the feature information between points and points and dynamically capture the local features and global features of the point cloud. The fusion attention mechanism is added to emphasize the important features of classification, suppress the unimportant features, improve the expression ability and performance of the network, and improve the classification accuracy of the network model. The proposed method is verified in the ModelNet40 dataset and achieved good classification results.

2. Point Cloud Classification Network Based on Graph Convolution and Attention Fusion Mechanism

2.1. Overall Network Structure

For the PointNet network model, its network structure mainly includes three main parts: point cloud alignment transformation, feature extraction and max pooling to achieved global features. The PointNet network contains two T-Net transform networks and two multi-layer perceptrons with shared weights. T-Net network aligns the point cloud to standardize the point cloud features, the multi-layer perceptron is mainly used to extract the features of the point cloud, and the network fuses multiple features through maximum pooling to achieved a

global feature of 1024 dimensions for final classification. The method of extracting point cloud features based on multi-layer perceptron does not pay attention to the local features of point cloud and ignores the connection between points. Aiming at the above problems, a point cloud classification network based on graph convolution and fusion attention mechanism is proposed. The proposed point cloud classification network structure is mainly composed of dynamic graph convolution module and fusion attention mechanism module.

The overall network structure is shown in **Figure 1**.

The proposed network model contains four Graph Conv modules and two F-Attention fusion attention mechanism modules. The convolution kernels of four Graph Conv modules from left to right are 64, 64, 128 and 256, respectively, and the two F-Attention modules are behind the Graph Conv layer with convolution kernels of 128 and 256, respectively. The point cloud input is $N \times D$ dimension, where N delegate the number of sampling points in the point cloud, and D delegate the data dimension of the point cloud. The most common $D = 3$ delegate that each point only has three-dimensional space coordinate information.

In the Graph Conv module, the input is the feature of $N \times f$ dimension, where N delegate the number of input points and f delegate the input dimension of the point. k means the number of neighbor points of the point cloud center point in the graph after using the KNN algorithm to construct the graph and extracting the edge features of each point through n weight-sharing multi-layer perceptrons ($mlp\{L_1, L_2, \dots, L_n\}$). The maximum pooling function is used to update the features of nodes for the extracted edge features, and finally the features with dimension of $N \times L_n$ are achieved.

The process of the entire network model is that the input $N \times D$ dimensional point cloud data is first extracted by two Graph Conv modules with a convolution kernel of 64 acquired $N \times 64$ dimensional point cloud feature information; after feature extraction by Graph Conv module with convolution kernel of 128, F-Attention (fused attention mechanism module) is used to aggregate local neighborhood features to output $N \times 128$ dimensional point cloud feature information. After feature extraction by Graph Conv module with convolution kernel of 256, F-Attention (fused attention mechanism module) is used to aggregate

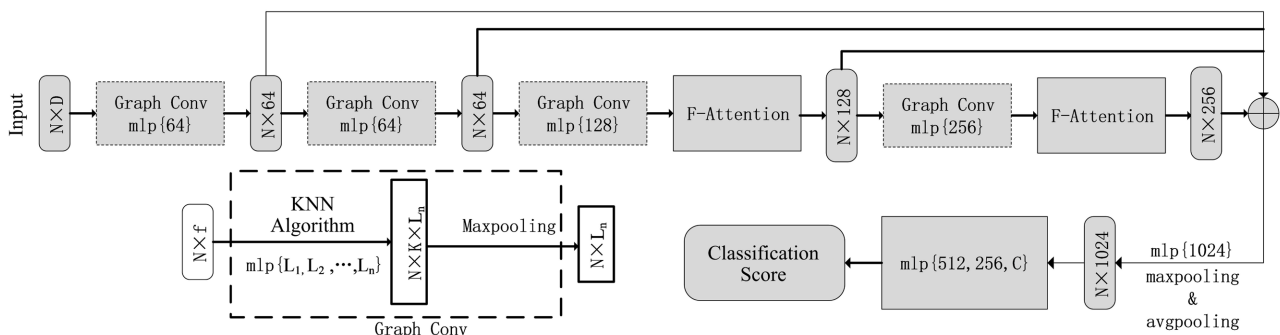


Figure 1. Overall network structure.

local neighborhood features to output $N \times 256$ dimensional point cloud feature information. Finally, the features of 64, 128, 256 dimensions are spliced, and the $N \times 1024$ dimension point cloud features are acquired by the pooling method combining the maximum pooling and the average pooling. This feature contains both the global features of the point cloud and the local features. It also focuses on the spatial information of the point cloud and the channel information of the point cloud, and then passes through three fully connected layers (512, 256, C) acquired the final classification score C .

The proposed model uses graph convolution instead of the method of using multi-layer perceptron to extract point cloud features in the PointNet network, so that the network can not only extract the global features of point cloud but also extract the local features of point cloud. Since in the PointNet++ network, it is verified that the T-Net network does not increase the classification performance of the network, the T-Net network is removed in the proposed model, and the ability of local feature extraction of the network is enhanced by stacking multiple graph convolutions. The attention mechanism module is introduced into the proposed model. To reduce the network complexity, two fusion attention mechanism modules are added, so that the network can focus on both the spatial information characteristics of the point cloud and the channel information characteristics of the point cloud. In the processes of getting global features, the multi-layer feature fusion method is used to make the network model better integrate the high-level features of the point cloud with the low-level features, and focus on the original information features of the point cloud as much as possible. The combination of maximum pooling and average pooling is used to replace the maximum pooling method in the PointNet network model. The global features and local features of the point cloud are better fused, and the fused 1024-dimensional global features are obtained for the final classification, and the final size of the whole model is 44.7 MB. Although it is 3.1 MB larger than the PointNet network model, the proposed model improves classification results than the PointNet network model through experimental analysis.

2.2. Dynamic Graph Convolution

In two-dimensional images, graph convolution is a convolutional neural network that can directly act on the graph and use its structural information. The main operation of graph convolution is to first construct the data into a graph with vertices and edges, and then convolve on the graph data. For graph convolution, it is divided into spatial domain graph convolution [22] and spectral domain graph convolution [23]. Because the spatial domain convolution method can directly define the convolution operation on the connection relationship of each node, it can better focus on the information characteristics of the neighborhood. Therefore, this model uses spatial domain graph convolution.

In the proposed network model, the point cloud input can be expressed as:

$$X = \{x_1, \dots, x_n\} \subseteq R^D \quad (1)$$

Among them, X means a set of point clouds, x_i means each point in the point cloud set. For each point in the point cloud set, they all have D dimensional features. The most common $D = 3$ means that each point only has three-dimensional spatial coordinate information.

For the local point cloud structure, it can be represented by a directed graph: $G = (V, E)$, where V is the set of N local nodes:

$$V = \{x_1, \dots, x_n\} \quad (2)$$

E means the set of edges between nodes:

$$E = \{e_{ij}\}_{i,j=1}^N \quad (3)$$

e_{ij} means the edge between node i and node j . The local directed graph G in the model is constructed using the K -nearest neighbor classification (KNN) algorithm. In a locally directed graph G , assuming that point i is a central node, then K nearest neighbors j including node i can be calculated by KNN algorithm. The edge feature e_{ij} of two adjacent nodes can be described as:

$$e_{ij} = h_\theta(x_i, x_j) \quad (4)$$

In Formula (4), the parameter θ means the set of parameters such as weights in the model, h_θ is a nonlinear function of the learnable parameter θ , x_i and x_j represent the characteristics of node i and its neighbor node j , respectively.

In the Graph Conv module, edge functions and aggregation operations play an important role in the local feature extraction of point clouds. The PointNet network can be regarded as a special form of graph convolutional neural network. There is no edge information between points. Its edge function is:

$$h_\theta(x_i, x_j) = h_\theta(x_i) \quad (5)$$

However, the edge function (5) only considered the global information of the point cloud in the local directed graph, and ignores the local information. In the proposed network model, the following edge function is defined by considering both global and local information of point clouds:

$$h_\theta(x_i, x_j) = h_\theta(x_i, x_j - x_i) \quad (6)$$

In the processes of aggregating features, for the central node x_i of the directed graph, x'_i is defined as the aggregation of the edge features of k points around it:

$$x'_i = \sum_{j:(i,j) \in E} h_\theta(x_j - x_i) \quad (7)$$

Firstly, the KNN algorithm is used to construct the graph structure from the point cloud to the point cloud set and the process of learning the aggregated edge features through the Graph Conv module is shown in **Figure 2**.

The above process is the graph convolution process. The main step is to select a point as the center point in the point cloud set, find its neighbor points by the K -nearest neighbor algorithm (five neighbor points shown on the graph), and

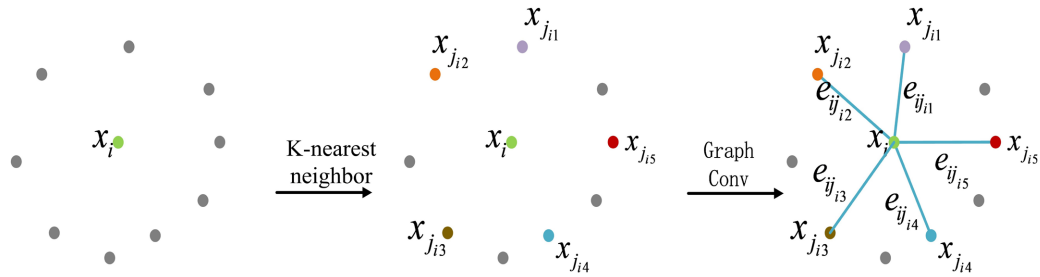


Figure 2. Graph convolution process.

learn the edge features between the center point and its neighbor points through the Graph Conv module. Finally, the edge features are aggregated. To increase the feature extraction ability of the network and expand the receptive field of the model, the proposed model realizes the dynamic update of the graph structure by stacking multiple Graph Conv modules in the network, thus forming a dynamic graph convolution. The input and directed graph structure of Layer l can be expressed as:

$$X^l = \{x_1^{(l)}, x_2^{(l)}, \dots, x_n^{(l)}\} \subseteq R^{D_l} \tag{8}$$

$$G = (V^{(l)}, E^{(l)}) \tag{9}$$

The output of the $l + 1$ graph convolution is updated to:

$$x_i^{(l+1)} = \sum_{j:(x_i, x_j) \in E^{(l)}} h_{\theta}^{(l)}(x_i^{(l)}, x_j^{(l)} - x_i^{(l)}) \tag{10}$$

2.3. Integrated Attention Mechanism

The attention mechanism [24] has important application scenarios in 2D images domain, which are divided into attention mechanisms that focus on space and attention mechanisms that pay more attention to channels. Based on the experience of the previous 2D attention mechanism, an attention mechanism that combines the spatial information features of the point cloud and the information features of the point cloud channel is designed in the model to emphasize the information features that are useful for classification and suppress the useless information features. The spatial information features and channel information features of the point cloud were better concerned, the feature extraction ability of the network model is strengthened, and the accuracy of network classification is improved.

The entire fusion attention module, shown in **Figure 3**, is called F-Attention in the network structure.

In the spatial attention mechanism module (Spatial attention), the input point cloud feature matrix is defined as A , and its dimension can be expressed as $B \times N \times C$. For matrix A , new feature matrices A_1 and A_2 containing more spatial information can be achieved by corresponding linear transformation, and the dimension of the matrix is $B \times N \times C$. The matrix A_1 is transposed, and the

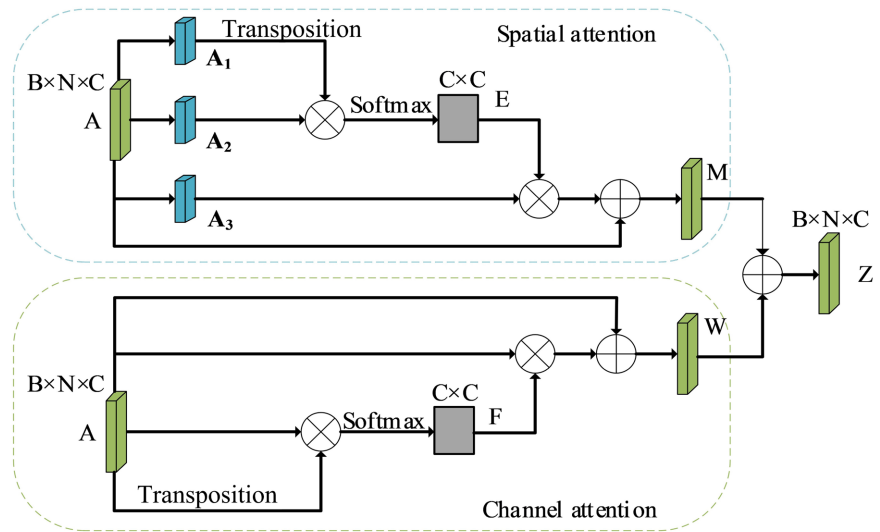


Figure 3. Fusion attention mechanism module (F-Attention).

transposed matrix is multiplied by the matrix A_2 . Then the softmax function is used to achieve the spatial attention coefficient matrix E with the size of $C \times C$. The calculation method is shown in Formula (11):

$$a_{ji} = \frac{\exp(A_i \cdot A_j)}{\sum_{i=1}^N \exp(A_i \cdot A_j)} \quad (11)$$

A_{ji} is the value calculated by the function Softmax, which means the influence of spatial position i on j in matrix E . The feature matrix A is input into a 1×1 convolution layer to achieve a new feature matrix A_3 with a dimension of $B \times N \times C$. Then, the output feature with a dimension of $B \times N \times C$ is achieved by matrix multiplication of the feature matrix A_3 and the attention coefficient matrix E . A learnable linear parameter λ is introduced for this output feature. The main purpose is to adjust the weight during the training process. After the above steps, a feature matrix updated by the attention mechanism can be achieved. Finally, the feature matrix and the elements in the original feature matrix A are summed one by one to achieve the final output M of feature A , as shown in Formula (12):

$$M_j = \lambda \sum_{i=1}^N (a_{ji} A_{3_i}) + A_j \quad (12)$$

The parameter λ in formula (12) is initialized to 0 to gradually assign more weights through network training. The final feature M achieved in this module not only contains the relevant features of the original point cloud, but also contains the spatial location features of the point cloud, and the updated feature M better aggregates the global context information.

In the more concerned point cloud channel information attention mechanism (Channel attention) module, which is similar to the above spatial attention module, the input point cloud feature matrix is defined as A , the feature size is $B \times N \times C$, the matrix A is transposed, the transposed matrix is multiplied by the

original matrix, and then the softmax function is used to achieve the channel attention coefficient matrix F with a size of $C \times C$. The calculation method is as shown in formula (13):

$$b_{ji} = \frac{\exp(A_i \cdot A_j)}{\sum_{i=1}^N \exp(A_i \cdot A_j)} \quad (13)$$

where b_{ji} measures the effect of channel i on channel j . The feature matrix A and the attention coefficient matrix F are multiplied by the matrix to achieve the output feature with a dimension of $B \times N \times C$. A learnable linear parameter χ is introduced for this output feature, and the weight can be adjusted during the training process. After the above steps, a feature matrix updated by the channel attention mechanism can be achieved. Finally, the feature matrix and the elements in the original feature matrix A are summed one by one to make up for the information of the input feature to achieve the final output W of the feature A , as shown in Formula (14):

$$W_j = \chi \sum_{i=1}^N (b_{ji} A_i) + A_j \quad (14)$$

Similarly, the parameter χ is initialized to 0 and trained to assign weights. As shown in **Figure 2**, the final fused local feature Z is achieved by adding the feature M obtained by spatial attention and the feature W achieved by channel attention.

3. Experiment and Analysis

3.1. Experimental Data Set

The data set used for the point cloud classification task uses the Princeton University's standard public data set ModelNet40, including 40 artificial object categories, with a totality of 12311 CAD models, of which 9843 models are used for training, and 2468 models are used for testing. For the input point cloud data, 1024 points are sampled from it, and the dimension D of the sampling points is 3, and the sampling points only contain 3D coordinate information.

3.2. Experimental Environment and Evaluation Indicators

The software environment required to run this model is Ubuntu 20.04.2 LTS + CUDA10.1 + PyTorch 1.6 + python3.7, the experimental parameter learning rate is 0.001, the number of iterations is 250, the batch size is 32, and the Adam optimizer is used.

The experiment mainly reflects the classification accuracy of the model by comparing the average classification accuracy (mAcc) and overall classification accuracy (OA) of different network models. The calculation formula of average classification accuracy and overall classification accuracy is:

$$OA = \frac{TP + TN}{TP + TN + FP + FN} \quad (15)$$

$$\text{mAcc} = \frac{\text{OA}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (16)$$

TP means the number of samples predicted to be positive, TN means the number of samples predicted to be negative, FP and FN represent the number of false negative and false positive samples, respectively.

3.3. Result Analysis

To get better experimental comparison results, different classical point cloud classification network models are selected on the ModelNet40 dataset. The selected network models are VoxNet, MVCNN, ECC, PointNet, PointNet++, LDGCNN, DGCNN. The data from different network model experiments are shown in **Table 1**.

In the data achieved in **Table 1**, it can be found that compared with some classical point cloud classification networks, the proposed model has a certain improvement in average classification accuracy and overall classification accuracy, and improved performance. Compared with the PointNet network, the average classification accuracy of the proposed model is increased by 4.4%, and the overall classification accuracy is increased by 3.8 %. This is because the proposed model not only extracts the global information features of the point cloud, but also extracts the local information features of the point cloud, which solves the defect that PointNet does not focus on the local information features of the point cloud. Compared with PointNet++ network, the overall classification accuracy of the proposed model is improved by 2.0%. This is because the proposed model strengthens the connection between points, pays attention to the information characteristics between point pairs, and solves the defect of PointNet++ processing points alone. Compared with the DGCNN network, the average classification accuracy of the proposed model is improved by 1.1%, and the overall classification accuracy is improved by 1.2%. This is because the proposed model not only focus on the point-to-point information, but also adds the fusion attention mechanism, focus on the channel information characteristics of the point cloud and

Table 1. Classification accuracy of different network models on ModelNet40.

Method	Model Input (size)	Model Size/MB	Average Classification Accuracy/%	Overall Classification Accuracy/%
VoxNet [14]	Voxels (12)	—	82.8	85.7
MVCNN [15]	Views (80)	—	89.1	—
ECC [25]	Points (1024)	—	83.2	87.4
PointNet [17]	Points (1024)	41.6	85.8	88.7
PointNet++ [20]	Points + normal (1024)	24.2	—	90.5
DGCNN [26]	Points (1024)	21.0	89.1	91.3
Ours	Points (1024)	44.7	90.2	92.5

the spatial information characteristics of the point cloud, so its classification accuracy is higher.

Like the PointNet network model, this model uses a classification network that directly inputs point cloud data without any changes to the point cloud data, and the network model is constructed with reference to the PointNet network model. The proposed model compares the PointNet network on the ModelNet40 dataset. The results achieved by classifying each category separately are shown in **Table 2**.

It can be seen from **Table 2** that compared with the PointNet network without considering the local information of the point cloud, the proposed model has higher classification accuracy than the PointNet network in most categories. For Bench, Guitar and Lamp categories with obvious features in the table, the proposed model improved the classification accuracy of these categories by 5%, 2% and 1.3% respectively compared to with the PointNet network. For categories such as Bathtub, Door and Wardrobe that do not have obvious features, the proposed model improved the classification accuracy of these categories by 5%, 1.2% and 4% respectively compared with the PointNet network. This is because

Table 2. Comparison of classification results of 40 object categories on the proposed model and pointnet model.

Category	PointNet	Ours	Category	PointNet	Ours
Airplane	1.000	1.000	Laptop	1.000	1.000
Bathtub	0.870	0.920	Mantel	0.930	0.950
Bed	0.960	0.970	Monitor	0.950	0.980
Bench	0.700	0.750	Night stand	0.742	0.776
Bookshelf	0.910	0.930	Person	0.920	0.950
Bottle	0.940	0.960	Piano	0.900	0.930
Bow	0.900	0.940	Range hood	0.920	0.952
Car	0.960	0.980	Sink	0.780	0.850
Chair	0.970	0.980	Sofa	0.960	0.970
Cone	0.950	1.000	Stairs	0.800	0.900
Cup	0.780	0.800	Stool	0.850	0.800
Curtain	0.900	0.920	Table	0.800	0.870
Desk	0.800	0.900	Tent	0.950	0.953
Door	0.800	0.920	Toilet	0.980	0.970
Dresser	0.696	0.726	Television stand	0.800	0.860
Flower pot	0.220	0.250	Vase	0.820	0.830
Glass box	0.950	0.970	Wardrobe	0.750	0.790
Guitar	1.000	0.980	Xbox	0.650	0.750
Keyboard	1.000	1.000	Plant	0.760	0.780
Lamp	0.950	0.963	Radio	0.750	0.800

the proposed model not only considered the local information characteristics of the point cloud but also considered the global information characteristics of the point cloud, and focus on the spatial information characteristics and channel information characteristics of the point cloud. Therefore, the recognition rate of the proposed model is improved in the categories with obvious and unobvious features.

For the proposed network model, in the processes of constructing graph structure by the K -nearest neighbor algorithm, different K values represent different local geometric information, which will affect the final classification results. Comparing different K values, the classification accuracy achieved by experimental analysis is the best when the K value is 20, as shown in **Table 3**.

Table 3 shows that the classification accuracy cannot be achieved when the K value is too small or too large. If the K value is too small, the local directed graph will be too small, resulting in insufficient local feature extraction. K value is too large will cause redundancy of information, both will lead to poor classification results. When the K value is 20, the average classification accuracy is 90.2%, the overall classification accuracy is 92.5%, and the achieved classification accuracy is the best. Therefore, the K value is set to 20 in the experiment.

The proposed network model contains four Graph Conv layers and two F-Attention modules, and the two F-Attention modules are after the Graph Conv layer with convolution kernels of 128 and 256, respectively. The effectiveness of the fusion attention mechanism for classification tasks is verified by reducing the F-Attention module.

It can be seen from **Table 4** that when the proposed model does not contain the F-Attention module, the average classification accuracy and the overall classification accuracy respectively are 89.1% and 91.3%. The average classification accuracy and overall classification accuracy respectively improved by 0.6% and 0.7%, when the F-Attention module was introduced after the Graph Conv layer with 128 convolutional kernels; the average classification accuracy and overall

Table 3. Classification accuracy of different K values on ModelNet40.

	$K=10$	$K=20$	$K=30$	$K=40$
Average Classification Accuracy/%	89.3	90.2	89.7	88.7
Overall Classification Accuracy/%	91.2	92.5	91.8	90.5

Table 4. Ablation experiment.

Lead-in F-Attention	Average Classification Accuracy/%	Overall Classification Accuracy/%
Nil	89.1	91.3
F-Attention (128)	89.7	92.0
F-Attention (256)	89.3	91.8
F-Attention (128) + (256)	90.2	92.5

classification accuracy respectively improved by 0.2% and 0.5%, when the F-Attention module was introduced after the Graph Conv layer with 256 convolutional kernels. After introducing two F-Attention modules, the average classification accuracy and the overall classification accuracy are respectively improved by 1.1% and 1.2% compared to the original model. The addition of the fusion attention mechanism to the model can not only focus on the spatial information features of the point cloud but also focus on the channel information features of the point cloud, and can emphasize the useful information features in the classification task to suppress the useless information features, and strengthen the ability of the network to extract features, to getting better classification results. The experimental results show that the F-Attention module has a significant improvement in the performance of the proposed model classification.

4. Conclusion

A point cloud classification network model based on dynamic graph convolution and fusion attention mechanism is proposed to address the shortcomings of the existing point cloud classification network, such as inadequate extraction of point cloud local information, ignoring the information in other neighborhood features in point cloud and not focusing on point cloud channel information and spatial information. By introducing graph convolution to achieve the extraction of point cloud local features, the local features and global features are better fused, and the attention mechanism of fusing more attention point cloud spatial information and point cloud channel information is introduced to enhance the feature extraction ability of the network. The experimental results show that compared with some existing classical point cloud classification networks, the proposed model has a certain improvement in classification accuracy and better classification performance. However, since this model does not pay attention to the direction feature information of the point cloud, it has certain deficiencies. Subsequent work will consider how to integrate the direction information features of point clouds into the model to improve the classification results.

Acknowledgements

This work is partially supported by the Development Program of Youth Innovation Teams in Colleges and Universities of Shandong Province (2019KJN048).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Jia, Z., Yuan, H.J., Zhao, X.F., *et al.* (2021) Single-Cell Genetic Analysis of Lung Tumor Cells Based on Self-Driving Micro-Cavity Array Chip. *Talanta*, **226**, Article ID: 122172. <https://doi.org/10.1016/j.talanta.2021.122172>
- [2] Liu, S., Liu, R.F. and Chai, Y.N. (2021) A Road Marking Line Classification Method

- Based on Structural Feature Matching. *Geospatial Information*, **19**, 14-17. (In Chinese) <https://doi.org/10.3969/j.issn.1672-4623.2021.03.004>
- [3] Gao, G., Yang, H.Y. and Liu, H. (2021) 3D Point Cloud Face Recognition Based on Deep Learning. *Journal of Computer Applications*, **41**, 2736-2740.
- [4] Wei, Y., Xiao, Y. and Yan, C. (2020) Data Optimization of Pavement 3D Reconstruction Based on Point Pre-Processing. *Journal of Jilin University*, **50**, 987-997. (In Chinese) <https://doi.org/10.13229/j.cnki.jdxbgxb20190121>
- [5] Gu, Z.X. and Pei, F.R. (2022) Single Tree Recognition Algorithm Based on Multi-Layer K-Means in Forest Point Cloud. *Forest Resources Management*, **1**, 124-131. (In Chinese) <https://doi.org/10.13466/j.cnki.lyzygl.2022.01.015>
- [6] Wang, B.J., Nong, L.P. and Zhang, W.H. (2020) 3D Point Cloud Classification and Segmentation Network Based on Spider Convolution. *Journal of Computer Applications*, **40**, 1607-1612. (In Chinese) <https://doi.org/10.11772/j.issn.1001-9081.2019101879>
- [7] Rusu, R.B., Marton, Z.C., Blodow, N. and Beetz, M. (2008) Learning Informative Point Classes for the Acquisition of Object Model Maps. *Proceedings of 10th International Conference on Control, Automation, Robotics and Vision*, Hanoi, 17-20 December 2008, 643-650. <https://doi.org/10.1109/ICARCV.2008.4795593>
- [8] Rusu, R.B., Bradski, G., Thibaux, R. and Hsu, J. (2010) Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram. *Proceedings of 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, 18-22 October 2010, 2155-2162. <https://doi.org/10.1109/IROS.2010.5651280>
- [9] Sun, J., Ovsjanikov, M. and Guibas, L. (2009) A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion. *Computer Graphics Forum*, **28**, 1383-1392. <https://doi.org/10.1111/j.1467-8659.2009.01515.x>
- [10] Deng, Z. and Latecki, L.J. (2017) Amodal Detection of 3D Objects: Inferring 3D Bounding Boxes from 2D Ones in RGB-Depth Images. *Proceedings of Conference on Computer Vision and Pattern Recognition*, Honolulu, 21-26 July 2017, 398-406. <https://doi.org/10.1109/CVPR.2017.50>
- [11] Engelcke, M., Rao, D., Wang, D.Z., Tong, C.H. and Posner, I. (2017) Vote3Deep: Fast Object Detection in 3D Point Clouds Using Efficient Convolutional Neural Networks. *Proceedings of IEEE International Conference on Robotics and Automation*, Singapore City, 29 May-3 June 2017, 1355-1361. <https://doi.org/10.1109/ICRA.2017.7989161>
- [12] Ren, S., He, K., Girshick, R. and Sun, J. (2017) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**, 1137-1149. <https://doi.org/10.1109/TPAMI.2016.2577031>
- [13] Lahoud, J. and Ghanem, B. (2017) 2D-Driven 3D Object Detection in RGB-D Images. *Proceedings of 2017 IEEE International Conference on Computer Vision*, Venice, 22-29 October 2017, 4622-4630. <https://doi.org/10.1109/ICCV.2017.495>
- [14] Maturana, D. and Scherer, S. (2015) VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. *Proceedings of 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hamburg, 28 September-2 October 2015, 922-928. <https://doi.org/10.1109/IROS.2015.7353481>
- [15] Su, H., Maji, S., Kalogerakis, E. and Learned-Miller, E. (2015) Multi-View Convolutional Neural Networks for 3D Shape Recognition. *Proceedings of 2015 IEEE International Conference on Computer Vision*, Santiago, 7-13 December 2015, 945-953. <https://doi.org/10.1109/ICCV.2015.114>

- [16] Wang, W.X. and Li, L.L. (2022) A Review of Deep Learning in Point Cloud Classification. *Computer Engineering and Applications*, **58**, 26-40. (In Chinese)
<https://doi.org/10.3778/j.issn.1002-8331.2105-0200>
- [17] Qi, C.R., Su, H., Mo, K. and Guibas, L.J. (2017) Pointnet: Deep Learning on Point Sets for 3D Classification and Segmentation. *Proceedings of Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 21-26 July 2017, 652-660.
- [18] Shi, Q., Anwar, S. and Barnes, N. (2020) Dense-Resolution Network for Point Cloud Classification and Segmentation. *Proceedings of Winter Conference on Applications of Computer Vision*, Snowmass Village, 1-5 May 2020, 3813-3822.
- [19] Shi, Q., Anwar, S. and Nick B. (2021) Semantic Segmentation for Real Point Cloud Scenes via Bilateral Augmentation and Adaptive Fusion. *Proceedings of Conference on Computer Vision and Pattern Recognition*, Nashville, 20-25 June 2021, 1757-1767.
- [20] Qi, C.R., Yi, L., Su, H. and Guibas, L.J. (2017) PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, 4-9 December 2017, 5105-5114.
- [21] Xu, B.B., Cen, K.Y., Huang, J.J., Shen, H.W. and Cheng, X.Q. (2020) A Survey on Graph Convolutional Neural Network. *Chinese Journal of Computers*, **43**, 754-780.
- [22] Wang, Y., Sun, Y.B., Liu, Z.W., et al. (2019) Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics*, **38**, 1-12.
<https://doi.org/10.1145/3326362>
- [23] Zhang, Y.X. and Rabbat, M. (2018) A Graph-CNN for 3D Point Cloud Classification. *Proceedings of 2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, Calgary, 15-20 April 2018, 6279-6283.
<https://doi.org/10.1109/ICASSP.2018.8462291>
- [24] Zhang, J.C., Zhu, L. and Yu, L. (2021) Review of Attention Mechanism in Convolutional Neural Networks. *Computer Engineering and Applications*, **57**, 64-72.
- [25] Simonovsky, M. and Komodakis, N. (2017) Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs. *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 21-26 July 2017, 9-38.
<https://doi.org/10.1109/CVPR.2017.11>
- [26] Wang, Y., Sun, Y.B., Liu, Z.W., et al. (2019) Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions Graphics*, **38**, 146.
<https://doi.org/10.1145/3326362>