DeepGCNs-Att for Point Cloud Semantic Segmentation

Xinyu Wang, Bin Jiang, Ziheng Zhang, Cheng Tong, Qing Me, Jian Xiao* and Yi Tong

College of Electronic and Optical Engineer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
Email: xiaoj@njupt.edu.cn; tongyi@njupt.edu.cn

Abstract. Compared with traditional CNNs, Graph Convolutional Networks (GCNs), with the graph’s neural network structure, can better characterize non-Euclidean space. Furthermore, with the number of the network layers increasing, deep GCNs demonstrate outstanding performance in mining the partial relationship between the point cloud’s local features. However, the current deep GCNs algorithm cannot sufficiently exploit the point cloud’s global characteristics for semantic segmentation. This paper proposes a novel network structure called DeepGCNs-Att to efficiently aggregate global context features. Moreover, to speed up the computation, we add an Attention layer after the GCN Backbone Block to mutually enhance the connection between the distant points of the non-Euclidean space. Our model is tested on the standard benchmark S3DIS. By comparing with other deep GCNs algorithms, our DeepGCNs-Att’s mIoU has at least two per cent higher than that of other models and even shows excellent results in space complexity and computational complexity under the same number of GCN layers.

Keywords. Point cloud; semantic segmentation; graph convolutional network; attention module; deep learning.

1. Introduction
In recent years, deep learning has been applied to various image processing applications, such as image classification [1], semantic segmentation [2] and other application methods. However, due to the lack of depth information, two-dimensional image data has certain limitations that cannot fully perceive the surrounding environment, which further facilitates the rapid development of three-dimensional sensors. By transforming the natural scenes into the three-dimensional point clouds, the researchers can utilize every single point to represent pixel’s 3D geometric coordinate, RGB colour, normal vectors, and other information. However, the irregular point cloud format makes a difference between the 2D and 3D datasets, making it difficult for convolutional neural networks [3] to work efficiently in the point cloud data.

To solve this problem, the method that researchers explore can be divided into four mainstream directions as follow: 3D convolution, multi-view projection onto images and 2D convolution, 1D/2D convolution or Multi-layer Perceptron (MLP) on the point cloud, and Graph Convolution Network (GCN). PointNet [4] using MLP and max-pooling, which is concise and effective, is widely used in feature extraction for point cloud detection. However, PointNet’s shortcomings are conspicuous, limiting the accuracy in more complex scenes due to only proposing global features and ignoring local features’ influence. Then, PointNet++ [5] proposed improvements that construct local and global features through the sampling and grouping layers. The Multi-scale grouping and the Multi-resolution grouping are designed to capture different features in densely sampled regions. Simultaneously, too much empirical data, which should be a manual setting, leads to limited results. Besides, because the
graphs can efficiently represent the point cloud, more and more networks are proposed to use GCN for point cloud processing. After the number of GCN layers increased, the problem of vanishing gradient and over-smoothing appeared. DeepGCNs [6] inspired by dilated convolutions and DGCNN [7] construct a dilated graph and solve the problem above. However, DeepGCNs only use one max-pooling layer at the end to aggregate global features without considering the connection between each point and global information.

In this paper, we use ResGCN as GCN Backbone Block and utilize Multilayer Perceptron (MLP) for dimensionality reduction in the output layer of the network, then use the dual attention module, including Spatial-wise Attention and Channel-wise Attention, which adaptively aggregates global features. Our contribution can be summarized as follows:

1. We propose a novel neural network model with higher accuracy and faster computational speed than others under the same number of GCN layers.
2. In the output layer of ResGCNs, dimensionality reduction is performed through MLP. The attention layer is used to directly output the classification of each point instead of the max-pooling layer.
3. The experiments on point cloud segmentation indicate that our model is robust to sampling density variation and has an excellent correct rate, achieving better performance in OA and mIoU than DeepGCNs under the same number of network layers.

2. Methodology
Point cloud segmentation takes the 3D point cloud as input and outputs the classification of each point. Our novel model consists of two parts, namely GCN Backbone Block and Prediction Block, which completes an end-to-end point cloud semantic segmentation network based on GCN. Our model is shown in figure 1.

![Figure 1. DeepGCNs-Att model.](image)

2.1. GCN Attention Backbone
The point cloud is difficult to extract features through traditional convolutional neural networks due to its spatial disorder. So ResGCNs utilizes GCNs to represent the point cloud as an undirected graph $g(v, e)$, as shown in figure 2, where $v$ and $e$ are the set of $n$ nodes and $e$ edges of each layer. Each layer of graph convolution performs the following update and aggregation operations,

$$g_{t+1} = F(g_t, w_t) = \text{Update}(\text{Aggregate}(g_t, w^{agg}_t), w^{update}_t)$$

(1)

As shown in equation (1) [6], in each $l-th$ layer, $F()$ is a fixed function that performs convolution operations, $g_t$ and $g_{t+1}$ represent the input and output of each layer, respectively, $w^{agg}_t$ and $w^{update}_t$ are the learnable aggregation and update weights, respectively. $F()$ represents a $1 \times 1$ convolution operation. At the end of each GCN layer, each vertex's information is updated through a $1 \times 1$ convolution. When a graph is used as input, the aggregation function aggregates the feature information of each vertex's neighbourhood and utilizes the update function to renew the representation of each node through a nonlinear transformation to learn new features,
As shown in equation (2) [6], in the \((l+1)-th\) layer, each feature vector \(h_{v_i} \in \mathbb{R}^D\) represents the graph’s vertex \(v_i\), where \(D\) is the dimension of the feature and \(h_{v_i}\) is the feature vector in the \((l+1)-th\) layer. \(N(v_i)\) is the set of neighbor vertices within a certain distance in the \(l-th\) layer. \(w_{\phi}\) parametrizes the vertices \(h_{v_i}\). \(\phi\) and \(\rho\) are the aggregation function and update function, respectively. \(w_{\phi}\) contains the learnable parameters of the aggregation functions, which is essential in GCN. In most GCNs, the algorithm only updates the vertices’ features without updating the edges of each layer. This structure allows the node to learn only the closer neighbours’ features. Nevertheless, when the points departing significantly in the non-Euclidean space have similar characteristics, the traditional GCN is likely to lose this critical information when learning. EdgeConv [7] can update the graph in each layer by searching for the nodes’ neighbouring points. The aggregation function is as follows:

\[
\dot{\phi} = \max(h_{v_i} - h_{v_i} | u_i \in N(v_i))
\]  

As shown in figure 2, ResGCNs subtract the vertex from its neighbours to distinguish different features and use the max-pool for feature aggregation. As the number of network layers increases, each node will have information about its distant neighbours to capture global feature information. Then, ResGCNs utilize MLP as the update function with the batch normalization to speed up gradient descent and use ReLu as the activation function to achieve nonlinear transformation.

Since the algorithm using ResGCN as the GCN Backbone Block, it can be stacked in multiple layers through the residual connection, which solves the vanishing gradient and over smoothing problem of the graph convolutional neural networks.

2.2. Prediction Block
In the prediction block layer, the model's function is to predict the category of all points in the point cloud through a neural network, so our method needs to fuse global features with local features. As shown in figure 3, the prediction block first utilizes MLP to reduce the dimensionality of each vertex' to obtain global features and then uses the attention module to learn global features. Finally, the classification results of each point can be obtained.

![Figure 2. Examples of the ResGCN model [6].](image)

2.3. MLP Feature Aggregation
The structure of MLP and PointNet are very similar. Our model connects each layer’s GCN Backbone
Block features to reduce the high-dimensional vertex features’ dimensionality and uses the $1 \times 1$ convolution kernel to aggregate global and local features.

![Figure 3. The architecture of Prediction Block.](image)

2.4. Attention Prediction

Unlike the traditional method of directly using softmax in the network’s output layer to obtain the classification of each node, our model utilizes the dual attention module [8] in the network’s output layer to strengthen the global context information of each point. We compared the prediction part’s result with DeepGCNs, finding that our model has fewer parameters, faster prediction speed and increased the graph’s output vertex information to get better classification results.

2.5. Spatial-Wise Attention

The role of Spatial-wise Attention is to adaptively aggregate global spatial feature information in the non-Euclidean space. As shown in equation (4) [8] and equation (5) [8], given the feature map $H \in \mathbb{R}^{C \times N}$ of the MLP output layer, the algorithm takes these feature vectors as input. The $C$ and $N$ are the number of channels in the feature map and the number of nodes, respectively.

Moreover, different feature maps $A_i$ and $B_j$ are obtained through two identical MLP $\rho_i$ and $\rho_j$ respectively, where $\{A_i, B_j\} \in \mathbb{R}^{C \times N}$ is the number of feature vectors and $C$ is the number of feature channels. The algorithm uses multiplication and softmax to obtain the normalized spatial attention weight $U_{ij}$ as follows:

$$U_{ij} = \text{softmax}(A_i, B_j) = \text{softmax}(\rho_i(H_i), \rho_j(H_j))$$

(4)

After that, our model performs MLP on the feature vectors $H$ to get a new feature map $C_j \in \mathbb{R}^{N \times D}$, multiply $C$ and $U_{ij}$ to update the feature map to aggregate the global spatial features, and merge with the original feature vectors $H_i$ to obtain a new feature map $\hat{H}$ as follows:

$$\hat{H} = \sum_{j=1}^{N} (U_{ij}C_j) + H_i$$

(5)

where $\hat{H}$ represents the weighted sum of the spatial feature dependency between each vertex and the original feature. The Spatial-wise Attention module can adaptively learn the vertices with similar distant features in the space.

2.6. Channel-Wise Attention

Channel-wise Attention is similar to Spatial-wise Attention. In each layer’s output of ResGCN, the channel's high-dimensional feature map can be regarded as a map of classes, making the self-attention module utilizes the interdependence between channels to learn global feature representation. The result can be represented as $\hat{H}$. Finally, the algorithm adds $H$ and $\hat{H}$ to obtain the final classification result.
through nonlinear transformation. So far, our model has made full use of the rich local features in GCN Backbone Block, considering the spatial correlation between features and the interdependence between feature maps. So, our neural network can be more accurate in complex three-dimensional scenes Classification results.

3. Experiment

3.1. Dataset
The S3DIS dataset constructs six different areas by scanning the indoor environment, including 271 rooms and 13 object classes (ceiling, floor, wall, beam, column, window, door, bookcases, board, clutters), 11 Scenes (offices, meeting rooms, corridors, auditoriums, open spaces, lobbies, lounges, pantry, copy rooms, storage, and toilets), which have a rich three-dimensional indoor characteristic structure. Compared with other datasets, S3DIS has a more complex spatial structure, making semantic segmentation more challenging.

3.2. Hardware Configuration
We utilized TensorFlow framework to complete the preparation of the network model. Our computational device equipped Intel(R) Core (TM) i9-9900k CPU @3.60GHz and two NVIDIA RTX 2080 Ti GPUs.

3.3. Implementation
To explain our model more clearly, we marked the models with different layers of DeepGCNs-Att as ResGCN-Att-7 and ResGCN-Att-14. Adam gradient descent optimizer is used for network training, where the initial learning rate is 0.01. The batch size of the ResGCN-Att-7 and ResGCN-Att-7 is set to 8 and 6, respectively. On area 5 and over 6-fold of S3DIS, we compared our model with PointNet [4], SEGCloud [9], RSNet [10], MS+CU [11], G+RCU [11], PointNet++ [5], 3DRNN+CF [12].

Due to hardware performance limitation, the batch size of ResGCN-Att-28 can only be set to 2, which directly cause serious gradient oscillation and the inaccurate final test result. So we did not test ResGCN-Att-28, but through the analysis of ResGCN-Att-7 and ResGCN-Att-14, ResGCN-Att-28 can be proved that it will have a better result.

The overall accuracy (OA) and mean intersection over union (mIoU) are utilized to evaluate the semantic segmentation performance of the network model. The mIoU is an index used to evaluate semantic segmentation. The results of semantic segmentation can be divided in to the following four categories: true positive (TP), false positive (FP), true negative (TN) and false negative (FN). The mIoU measures the ratio of the intersection and union of the two sets of true and predicted values, that is, \[ mIoU = \frac{TP}{TP + (FP + FN + TP)} \].

3.4. Flops and Params
The calculation of a neural network can be regarded as the input and output of the data. The computer’s memory is the Params of the network. The number of Flops is an essential indicator for evaluating the neural network’s overall performance, corresponding to the algorithm’s computational and space complexity. As shown in table 1, after calculating the number of Flops and Params of ResGCN and ResGCN-Att models, we found that our models have fewer Params and Flops under the same network layer, fewer computing resources, and faster computing speed.

3.5. Effect of Network Depth
We tested the DeepGCNs-Att network separately and analyzed the overall performance of the different number of layers. We conducted tests on the ResGCN-Att-7 and ResGCN-Att-14 with the original network due to computing power limitations. Results in table 2 show the effect of different backbone network layers on DeepGCNs-Att. With the number of network layers increasing, ResGCN-Att-7 and ResGCN-Att-14’s mIoU are approximately two per cent higher than the DeepGCN. Although we did
not test ResGCN-Att-28 due to computational limitation, we can make reasonable predictions from mIoU results of ResGCN-Att-7 and ResGCN-Att-14 that the ResGCN-Att-28 will show better performance than ResGCN-28.

Table 1. Params and flops of the two network models under the same number of layers.

| Method           | FLOPs      | Params    |
|------------------|------------|-----------|
| ResGCN-Att-7     | 405670670811 | 1239503   |
| ResGCN-Att-14    | 688299896242 | 1756943   |
| ResGCN-7         | 436747801923 | 1403853   |
| ResGCN-14        | 764491157786 | 2150669   |

Table 2. Results of S3DIS dataset on “area 5” in different layers of GCN Backbone Block.

| Test area | Method          | mIoU  |
|-----------|-----------------|-------|
|           | ResGCN-Att-7    | 50.74 |
|           | ResGCN-Att-14   | 52.38 |
|           | ResGCN-7 [6]    | 48.95 |
| Area 5    | ResGCN-14 [6]   | 49.90 |
|           | PointNet [4]    | 41.09 |
|           | SEGCloud [9]    | 48.92 |
|           | RSNet [10]      | 51.93 |

The results of OA and mIoU are shown in table 3. In area 5, our model’s mIoU is very similar to the results of ResGCN-28’s and exceeds the existing network model. In terms of over 6 fold, our model achieves accuracy close to the original network model and surpasses the current model on mIoU, but has fewer parameters and faster computational speed.

Table 3. IoU segmentation results of each category of the S3DIS data set in area 5 and 6-fold.

| Test area | Method          | OA   | mIoU  |
|-----------|-----------------|------|-------|
|           | PointNet [4]    | -    | 41.09 |
|           | SEGCloud [9]    | -    | 48.92 |
|           | RSNet [10]      | -    | 51.93 |
| Area 5    | ResGCN-28 [6]   | -    | 52.49 |
|           | ResGCN-Att-14 (Ours) | 83.96 | 52.38 |
|           | PointNet [4]    | 78.5 | 47.6  |
|           | MS+CU [11]      | 79.2 | 47.8  |
|           | G+RCU [11]      | 81.1 | 49.7  |
| 6-fold    | DGCNN [7]       | 84.3 | 56.1  |
|           | PointNet++ [5]  | -    | 53.2  |
|           | ResGCN-28 [6]   | 85.9 | 60.0  |
|           | ResGCN-Att-14 (Ours) | 84.7  | 57.7  |

To test the overall model performance on the whole dataset, we operated 6-fold cross validation. In
repeated training, each area can be used as a test set, and finally, the IoU is averaged to get the final mIoU. Therefore, the 6-fold crossover verification proves the accuracy of our model’s prediction. Our models are compared with the state-of-art models on area 5 and over 6-fold, respectively.

We summarized each category’s IoU in table 4. Our proposed method has a similar IoU result to ResGCN-28 in most categories, with only a slight decline due to the decrease in the number of GCN layers. We replaced the fusion module of the original network layer with the dual attention module, which made the prediction module have fewer parameters.

With the reduction of the GCN Backbone Block’s layers, the network model can still achieve relatively good results in semantic segmentation. Besides, in terms of area 5 and over 6-fold, the semantic segmentation performance of our model is better than the state-of-art model in most categories.

**Table 4.** IoU segmentation results of each category of the S3DIS data set in area 5.

| Test area | Method | Ceiling | Floor | Wall | Beam | Column | Window | Door | Table | Chair | Sofa | Bookcase | Board | Clutter |
|-----------|--------|---------|-------|------|------|--------|--------|------|-------|-------|------|-----------|-------|---------|
| Area 5    | PointNet [4] | 88.8    | 97.33 | 69.8 | 0.05 | 3.92   | 46.26  | 10.76 | 52.61 | 58.93 | 40.28 | 5.85      | 26.38 | 33.22   |
|           | SEGCloud [9]  | 90.06   | 96.05 | 69.86 | 0.00 | 18.37  | 45.37  | 23.12 | 75.89 | 70.40 | 58.42 | 40.88     | 12.96 | 41.06   |
|           | RSNet [10]    | 93.34   | 98.36 | 79.18 |      | 15.75  | 42.48  | 65.52 | 67.87 | 22.45 | 52.45 |           | 41.02 | 43.64   |
|           | ResGCN-Att-14 (Ours) | 91.44   | 97.62 | 75.85 | 0.00 | 15.38  | 50.47  | 42.48 | 67.64 | 75.39 | 14.46 | 56.66     | 51.32 | 42.23   |
| 6-fold    | PointNet [4]  | 88.0    | 97.3  | 69.3  | 42.4 | 23.1   | 47.5   | 51.6  | 42.0  | 54.1  | 9.6   | 38.2      | 29.4  | 35.2    |
|           | MS+CU [11]    | 88.6    | 95.8  | 67.3  | 36.9 | 24.9   | 48.6   | 52.3  | 51.9  | 45.1  | 10.6  | 36.8      | 24.7  | 37.5    |
|           | G+RCU [11]    | 90.3    | 92.1  | 67.9  | 44.7 | 24.2   | 52.3   | 51.2  | 58.1  | 47.4  | 6.9   | 39.0      | 30.0  | 41.9    |
|           | PointNet++ [5] | 90.2    | 91.7  | 73.1  | 42.7 | 21.2   | 49.7   | 42.3  | 62.7  | 59.0  | 19.6  | 45.8      | 36.7  | 51.6    |
|           | 3DRNN+CF [12] | 92.9    | 93.8  | 73.1  | 42.5 | 25.9   | 47.6   | 59.2  | 60.4  | 66.7  | 24.8  | 57.0      | 36.7  | 51.6    |
|           | ResGCN-28 [6] | 93.1    | 95.3  | 78.2  | 33.9 | 37.4   | 56.1   | 68.2  | 64.9  | 61.0  | 34.6  | 51.5      | 51.1  | 54.4    |
|           | ResGCN-Att-14 (Ours) | 93.0    | 92.3  | 76.9  | 32.7 | 33.1   | 56.6   | 67.0  | 61.6  | 63.5  | 21.9  | 51.3      | 47.0  | 53.3    |

As shown in figure 4, the classification effect of ResGCN-Att-14 is almost identical to the result of the point cloud classification with ResGCN-28. Only a few parts of the point cloud are detected since the algorithm adopted the dual attention module, enabling the model to learn global context information autonomously in the prediction part. So, we utilized fewer GCN layers to obtain classification results similar to 28 layers.
4. Conclusion
In this paper, we proposed a novel neural network model to improve DeepGCNs performance. After the GCN Backbone Block, we utilized MLP for feature fusion and the dual attention module to actively learn global features in the point cloud to replace the max-pooling layer. Under different indicators and the same number of layers, our network shows excellent results in terms of space complexity and computational complexity.

Acknowledgements
Our project was supported by the National Natural Science Foundation of China (Grant No.61974073).

Reference
[1] Lu D and Weng Q 2007 A survey of image classification methods and techniques for improving classification performance International Journal of Remote Sensing 28 (5) 823-870.
[2] Lateef F and Ruichek Y 2019 Survey on semantic segmentation using deep learning techniques Neurocomputing 338 321-48.
[3] Li Y, Hao Z B and Lei H 2016 Survey of convolutional neural network Journal of Computer Applications Journal of Computer Applications 36 (9) 2508-2515.
[4] 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 the IEEE Conference on Computer Vision and Pattern Recognition pp 652-660.
[5] 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 pp 5105-5114.
[6] Li G, Muller M, Thabet A and Ghanem B 2019 Deepgcns: Can GCNs go as deep as CNNs? *Proceedings of the IEEE/CVF International Conference on Computer Vision* pp 9267-9276.

[7] Wang Y, Sun Y, Liu Z, Sarma S E, Bronstein M M and Solomon J M 2019 Dynamic graph CNN for learning on point clouds *ACM Transactions on Graphics (TOG)* 38 (5) 1-12.

[8] Fu J, Liu J, Tian H, Li Y, Bao Y, Fang Z and Lu H 2019 Dual attention network for scene segmentation *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* pp 3146-3154.

[9] Tchapmi L, Choy C, Armeni I, Gwak J and Savarese S 2017 Segcloud: Semantic segmentation of 3D point clouds *2017 International Conference on 3D Vision (3DV)* pp 537-547.

[10] Huang Q, Wang W and Neumann U 2018 Recurrent slice networks for 3d segmentation of point clouds *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* pp 2626-2635.

[11] Engelmann F, Kontogianni T, Hermans A and Leibe B 2017 Exploring spatial context for 3D semantic segmentation of point clouds *Proceedings of the IEEE International Conference on Computer Vision Workshops* pp 716-724.

[12] Ye X, Li J, Huang H, Du L and Zhang X 2018 3D recurrent neural networks with context fusion for point cloud semantic segmentation *Proceedings of the European Conference on Computer Vision (ECCV)* pp 403-417.