3D Point Cloud Semantic Segmentation Based PAConv and SE_variant

ZHANG Ying, SUN Yue, WU Lin, ZHANG Lulu, MENG Bumin

(*School of automation and electronic information*, *Xiangtan University*, *Xiangtan 411105*, *China.*)

Abstract: With the increasing popularity of 3D sensors (e.g., Kinect) and light field cameras, technologies such as driverless, smart home and virtual reality have become hot spots for engineering applications. As an important part of 3D vision tasks, point cloud semantic segmentation has received a lot of attention from researchers. In this work, we focus on realistically collected indoor point cloud data and propose a point cloud semantic segmentation method based on PAConv and SE_variant. The SE_variant module captures global perception from a broad perspective of feature space by fusing different pooling methods, which fully utilize the channel information of point clouds. The effectiveness of the method is verified by comparing with other methods on S3DIS and ScanNetV2 semantic tagging benchmarks, and achieving 65.3% mIoU in S3DIS, 47.6% mIoU in ScanNetV2. The results of the ablation experiments verify the effectiveness of the key modules and analyze how to use the attention mechanism to improve the 3D semantic segmentation performance.

Keywords: Semantic Segmentation, Point Cloud, SE_variant, Attention Mechanism

1 Introduction

With the development and rise of artificial intelligence technology, point cloud data analysis has aroused widespread concern. Compared with images, point cloud contains richer 3D spatial information, and are not affected by external factors such as illumination and perspective, so they can describe models more accurately and comprehensively. As the key to scene understanding, 3D point cloud segmentation is one of the frontier research directions of artificial intelligence, which has been widely used in the fields of smart cities, robotics, unmanned driving, and laser remote sensing measurement[1].

Point cloud segmentation methods include traditional point cloud segmentation and point cloud semantic segmentation. The traditional segmentation uses the location, shape and other information of the point cloud to segment different region boundaries. There are mainly edge-based, region-based and model-fitting-based segmentation methods. The segmentation results obtained by them do not contain any semantic information, and requires manual semantic annotation of the results, which is extremely inefficient in the case of large data scales. Based on traditional methods, point cloud semantic segmentation automatically labels different types of objects in 3D space with semantic labels of different categories, so that each object has specific category information. At present, it mainly uses deep learning as the implementation method, and the processing of point clouds mainly includes voxel-based, projection-based and point-based method[2].

Voxels are small squares that divide the three-dimensional space, similar to the pixels of images. In order to make the irregular structure orderly,

researchers voxelize the 3D point cloud, and then based on this structured "voxel" data to construct a 3D neural network for segmentation. The projection-based methods project point cloud data into 2D space, then uses the neural network to segment the 2D feature image, and finally maps the segmentation result back to the point cloud. The point-based methods design a new neural network to segment unstructured 3D point cloud directly. The core idea of the voxelization and projection methods is to preprocess disordered point cloud into a regular structure, which can be processed by the traditional Convolutional Neural Network (CNN), but both inevitably bring about the loss of point cloud information. While point-based methods directly take the original point cloud as the input of networks, so in order to make full use of point cloud information, this paper selects PointNet++[3] as the baseline. We find that although the Unit PointNet layer in Encoder uses MLP to extract features better, it ignores the geometric features of points. In addition, when extracting features, shared weights are used for information with different degrees of importance, and no distinction is made. Therefore, this paper improves the above shortcomings and proposes a new network.

The key contributions of this paper are as follows:

1) We propose a 3D point cloud semantic segmentation based PAConv and SE_variant, which achieves 65.3% mIoU in S3DIS and 47.6% mIoU in ScanNetV2.

2) We port the 2D attention module to the 3D segmentation task and propose a SE_variant module to improve the feature extraction performance by paying more attention to the channel information.

3) We evaluate the performance of six attention modules from 2D and 3D on S3DIS dataset. Then we compare the effect and complexity of different attention mechanisms to verify the effectiveness of our SE_variant module, and provide some suggestions on how to effectively use attention mechanism to improve 3D semantic segmentation performance.

2 Related Work

In this section, we introduce previous studies about deep learning algorithms for point clouds.

2.1 Point-based Network

PointNet[4] proposed by Qi is the first one to directly uses the original point cloud as input for semantic segmentation. It uses Multi-Layer Perception (MLP) to calculate the features of point clouds and finally integrates the global features through a max-pooling layer, and proposes a T-Net introduced by PointNet is aim to solve the problem - how to extracted a point cloud feature that invariant to rigid body transformation. Although PointNet can get better segmentation results, it still has the disadvantage of not being able to obtain local features, which makes it difficult to analyze complex scenes. Therefore, the author further proposed PointNet++, drawing on the idea of multi-layer receptive fields in CNN, and proposed a multi-level feature extraction structure to iteratively extract features from the area around each point, so as to solve the problem of ignoring local features. Atzmon et al. proposed PCNN[5] to generalize image CNNs, allowing adjusting the network structure, using expansion operators and constraint operators to generate convolutions adapted to point clouds. PointConv[6] proposed by Wu et al. uses density reweighting to efficiently calculate the weight function, which significantly improves the effect. The RandLA-Net[7] proposed by Hu et al. uses random sampling to solve the scale limitation, and introduces a Local Spatial Encoding (LocSE) module. The LocSE learns complex local structures by increasing the receptive field, which can effectively preserve geometric features. The PCT[8] proposed by Guo et al. is based on the Transformer module, which embeds the coordinates of points into the feature space to generate new features, then feeds them into the attention module to obtain discriminative representations and learn the semantic information of points.

2.2 Point Convolution Method

Compared with the 2D convolution on images, the convolution method for 3D point clouds is difficult to design due to the irregularity. In DensePoint[9], convolution is decomposed into two core steps: feature transformation and feature aggregation, where feature

transformation is realized through a shared Single-Layer Perceptron (SLP), and feature aggregation is a symmetric function. The input of KPConv[10] is the position difference between the centre point and the adjacent point, and the definition field is a sphere with radius r, which has different weights in different areas. The correlation in the kernel function adopts a simpler linear correlation. In SpiderCNN[11], SpiderConv defines convolution as the product of a simple step function and a Taylor polynomial. The step function obtains rough geometry by encoding local geodesic distances, and the Taylor polynomial obtains local geometric changes by inserting arbitrary values on cube vertices. PointCNN[12] designed an MLP-based χ- Transformation, which also uses local regions to make full use of spatial local correlation, and the specific input is domain points and related features. To improve segmentation performance, we replace the MLP in each Encoder with PAConv[13], which allows the network to take full advantage of position information.

2.3 Attention Mechanism

The attention mechanism focuses on the information that is more critical to the current task in the input information, reduces the attention to other information, and even filters out irrelevant information, which can solve the problem of information overload and improve the efficiency and accuracy of the task. It is widely used in deep neural networks to obtain new feature maps by reweighting original features with estimated attention maps. In image-related tasks, attention maps can be generated from spatial or channel information, while some methods combine both to integrate information better. Furthermore, point cloud neural networks tend to use the self-attention module, which can calculate the long-term dependence without considering the order of elements. In practice, we can exploit the basic form of self-attention to compute point relationships or channel associations in point cloud analysis problems. However, related experiments prove that the self-attention mechanism requires expensive computation, especially on large-scale point cloud data, and spatial or channel attention modules can be embedded into point cloud feature representation to achieve the same or even better results. At the same time, it is found that for point cloud features, channel information is more important with the least increase in memory and number of parameters, so we decide to introduce SE attention mechanism into the network and propose a SE_variant module to capture more channel information.

3 Methodology

In this section, we will introduce the details of our overall network structure. As shown in the top half of Fig.1, the network consists of three major components: Encoder, SE_variant module and Decoder.

3.1 Encoder

The Encoder is composed of Sampling layer, Grouping layer and PAConv. The Sampling layer chooses a group of points from the input to determine the centers of local areas. The Grouping layer then creates sets of local regions by finding points that are "nearby" the centers. Finally, PAConv is used to encode each region to obtain feature vectors, which constructs the convolution kernel through dynamic data, making full use of the location information of points.

1) Sampling layer: Like PointNet++, the Sampling layer in Encoder as shown in the left bottom of Fig.1 uses FPS (farthest point sampling) to sample inputs. Given input points $\{x_1, x_2, ..., x_n\}$, it select a subset of m center points $\{x_{i_1}, x_{i_2}, ..., x_{i_m}\}$, so that x_{i_1} is the farthest point relative to other points in the set $\{x_{i_1}, x_{i_2}, ..., x_{i_{j-1}}\}$. Given the same number of centers, it covers the entire point set better than random sampling.

2) Grouping layer: The Grouping layer divides the point sets obtained in the Sampling layer into several regions. The input is a set of points $N\times(d+C)$ and a set of coordinates $N_0 \times d$ of centers. The result is a group of point sets $N_0\times K\times(d+C)$, where each group represents a local region and K is the number of points near the center point. It adopts the Boolean query method to select K points within a given radius, where the query distance is the metric distance, and K is different in different local areas. Compared with K-Nearest Neighbor (KNN) search, Boolean query keeps

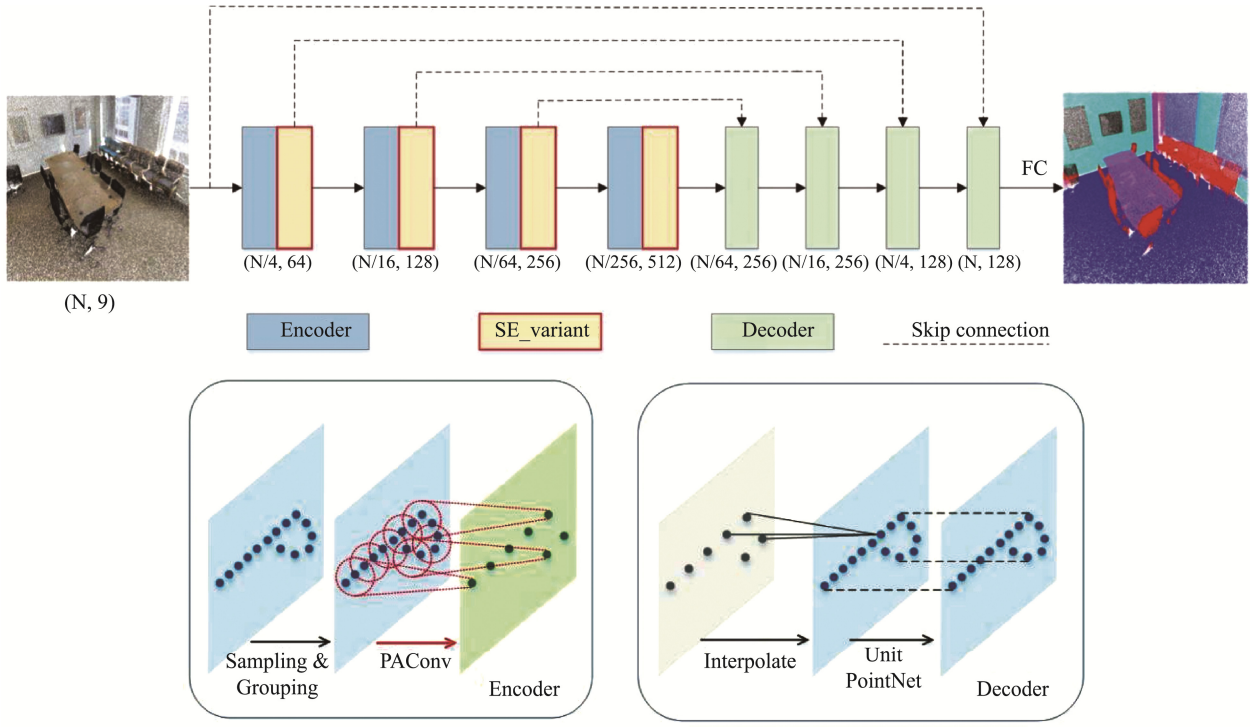

Fig.1 The Details of Our Semantic Segmentation Network
(The upper part is the overall framework, the lower part is the detailed structure of encoder and decoder.)

local neighborhoods within a fixed region, making local region features more generalizable in space.

3) Position Adaptive Convolution: Position Adaptive Convolution (PAConv) first defines a weight bank consisting of weight matrices. ScoreNet then learns a vector of coefficients based on point locations to combine weight matrices. Finally, the dynamic kernel is generated by combining the weight matrix and its associated position adaptation coefficients. The resulting convolution kernel is applied to the input features, and then the output features are obtained by max pooling. The details are shown in Fig.2 and elaborated below.

The weight bank $B=\{B_m|m=1, ..., M\}$ is generated by random initialization, where each $B_m \in R^{C_{in} \times C_{out}}$ represents a weight matrix, and M represents the number of matrices. Intuitively, a larger M represents a more diverse weight matrix, but at the same time, it will bring more memory usage and even cause redundancy.

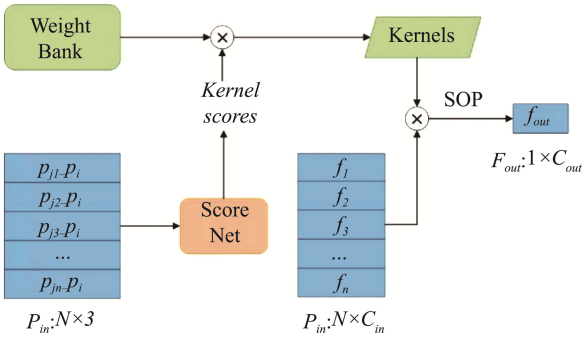

Fig.2 The Structure of Position Adaptive Convolution Where SOP Means Symmetric Operations, like MAX, p_{jk} Represents the Different Neighbors of p_i.

ScoreNet is responsible for correlating the relative positions of points with the weight matrix. Given the positional relationship $(p_i, p_j) \in R^3$ between the centre point p_i and its adjacent point p_j, ScoreNet predicts the positional adaptation coefficient S_{ij}^m of B_m:

$$S_{ij} = \alpha\left(\theta(p_i, p_j)\right) \quad (1)$$

where θ denotes MLP and α is the normalization op-

eration implemented using the softmax function. The output is a normalized vector $S_{ij}=\{ S_{ij}^m \mid m=1, ..., M\}$, where S_{ij}^m represents the coefficients of B_m when building the kernel $K(p_i, p_j)$. The softmax function guarantees that the value range of the coefficient is between 0 and 1. This normalization makes sure that each weight matrix is selected with a specific probability. The larger the value, the stronger the relationship between the position input and the weight matrix.

The final kernel is derived by combining the weight matrix from the weight bank with the positional adaptation coefficients predicted by ScoreNet:

$$K(p_i, p_j) = \sum_{m=1}^{M} (S_{ij}^m B_m) \tag{2}$$

PAConv constructs the convolution kernel through dynamic data, generates adaptive coefficients S_{ij}^m based on the position information of points, and flexibly utilizes the irregular geometric structure of 3D point clouds.

3.2 SE_variant Module

SE_variant module is improved on Squeeze and Excitation Network (SENet)[14]. SENet is a plug-and-play module proposed in 2017, which is widely used in the field of computer vision. It makes full use of the channel information of the features and adaptively complete the recalibration of channel features by explicitly modeling the interdependence between channels.

SE_variant consists of Squeeze, Excitation and Reweight. Squeeze compresses the features along the spatial dimension of point clouds, and converts the feature information into a real number, which has the global receptive field of the channel. The output dimension is the same as the input feature. Adopting different pooling methods means collecting feature information from different angles. Therefore, in order to effectively improve the performance of our network, we decided to use average pooling and max pooling in parallel to aggregate the input feature P_{in}, then generate feature descriptors P_{avg} and P_{max} for different angles.

$$P_{avg}=AvgPool(P_{in}) \tag{3}$$

$$P_{max}=MaxPool(P_{in}) \tag{4}$$

Excitation generates the required weight information through the weight W, which is obtained through learning and is used to model feature correlation. Since it is not possible to use CNN directly on the point cloud, we use the double hidden layer MLP with shared parameters to train the aggregated features. Finally, the activation function σ is used to activate weights.

$$P_s = \sigma(L(\delta(L(P_{avg}))) + L(\delta(L(P_{max})))) \tag{5}$$

where σ represents the sigmoid function, L represents the Linear function, and δ represents the Leaky_ReLU activation function.

During backpropagation, the Leaky_ReLU function performs the gradient calculation for the part of the input that is less than 0, instead of taking the value of 0 as in ReLU, which solves the problem of "death" of neurons, as shown in Fig.3.

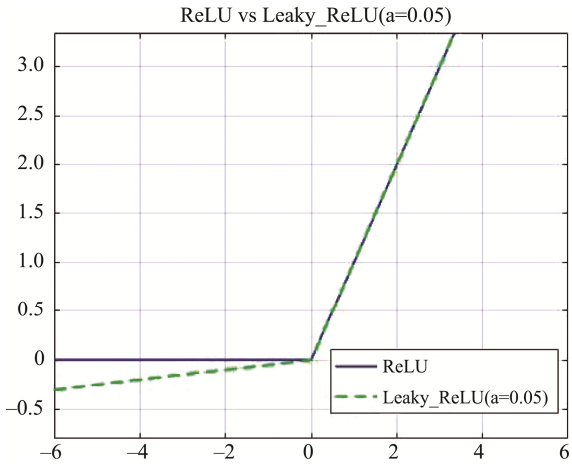

Fig.3 The Comparison of ReLU and Leaky_ReLU Function

In order to reduce complexity and improve adaptability to different data, the first Linear function reduces the input channel dimension to C/16, then passes through the Leaky_ReLU function, and the data dimension is expanded to the same as the original input by the second Linear function. Finally, the weight values are normalized in the range (0, 1) using the sigmoid function.

Reweight performs feature recalibration by multiplying the original features by the output of Excitation, which represents the importance of feature channels.

$$P_{out} = P_s \otimes P_{in} \tag{6}$$

where P_{out} is the new feature output by the SE_variant module. The calculation process is shown in Fig.4.

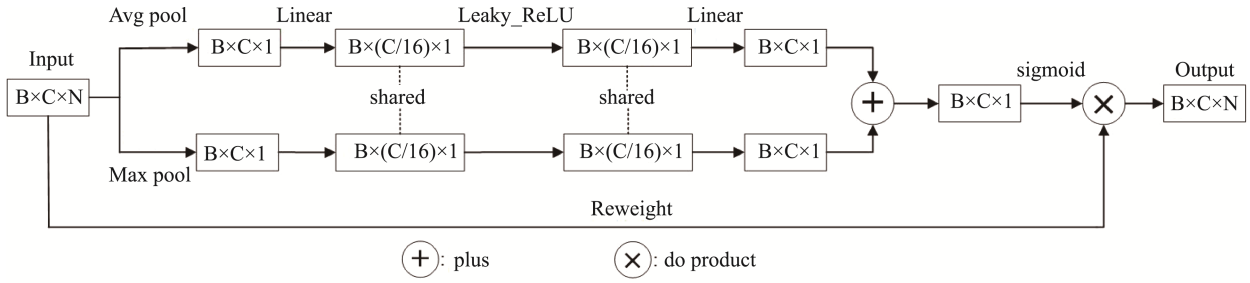

Fig.4 The Calculation Process of SE_variant

3.3 Decoder

As shown in the lower right of Fig.1, Decoder first uses the interpolation method to restore the original point cloud scale for Up sampling. According to the coordinates of the center point, use the KNN with K=3:

$$P^j(x)=\frac{\sum_{i=1}^{K} w_i(x)P_i^j}{\sum_{i-1}^{K} w_i(x)},\ w_i=\frac{1}{d(x,\ x_i)},\ 1\leqslant j\leqslant C_3 \quad (7)$$

The Unit PointNet network is mainly composed of a Transformation Network (T-Net) and a Multi-Layer Perceptron (MLP). The transformation matrix generated by T-Net is directly applied to the coordinates of points. Specifically, two-dimensional regularization is used, and in order to maintain the rotation invariance of point clouds, an orthogonal matrix is used as much as possible:

$$P_{reg}=\left\|I-AA_T\right\|^2 \quad (8)$$

where P_{reg} is the transformed feature matrix, I is the identity matrix with the same dimension as the input matrix, and A is the feature matrix to be transformed. The role of T-Net is to align features to make them easier to extract.

3.4 Loss Function

The loss function in the network is calculated by adding up the loss values of two parts:

$$L=L_{pred}+L_{regu} \quad (9)$$

where L_{pred} is the standard cross entropy function used in PointNet++, and L_{regu} is a weight regularization proposed in PAConv.

The weight matrix in PAConv is randomly initialized, so it may cause similar weight matrices, and L_{regu} is used to punish the correlation between different matrices, defined as:

$$L_{regu}=\sum_{B_i,\ B_j\in B,\ i\neq j}\frac{\left|\sum B_iB_j\right|}{\left\|B_i\right\|_2\left\|B_j\right\|_2} \quad (10)$$

This ensures that the weight matrix is distributed differently, further guaranteeing the diversity of generated kernels.

4 Experiments

4.1 Experimental Settings

1) Datasets: To prove the efficiency of our approach, we run experiments on two popular 3D point cloud datasets, which represent real point clouds in different scenes.

S3DIS: Stanford large-scale 3D Indoor Spaces (S3DIS)[15] dataset is collected from indoor environment and contains 271 rooms in 6 areas of three different buildings. It has 695878620 point clouds, each with corresponding coordinates and colour information, and semantic labels such as chair, table, floor, wall, etc. in a total of 13 categories. We choose areas 1, 2, 3, 4, and 6 for training and area 5 for testing. According to a common protocol, the input points are sampled into 4096 uniform points during training and all points are used for testing.

ScanNetV2: ScanNetV2[16] dataset consists of 3D scenes of real indoor rooms, resulting in 2.5 million views across over 1500 scans, annotated with 3D camera pose, surface reconstruction and instance-level semantic segmentation. It samples point

cloud data from reconstructed grid vertices, labels each point from 20 semantic categories, and divides them into 1201 training, 312 validation, and 100 test scan scenes.

2) Training Settings & Evaluation Metrics: We train the model for 150 epochs on a GeForce RTX 3090 GPU with a batch size of either 16 or 8. The optimization algorithm is SGD with an initial learning rate of 0.02 and divided by 10 at epoch 60 and 80. The momentum is set to 0.9 and the weight decay is 10^{-4}.

To evaluate the semantic segmentation performance of our network, we use mean intersection-over-union (mIoU) as the evaluation metric, following the official guidance[17]. It is the average value of IoUs for all semantic classes across the entire dataset. IoU_i is formulated as $\frac{TP_i}{TP_i + FP_i + FN_i}$, where TP_i, FP_i, FN_i represent true positive, false positive, and false negative predictions for class i.

4.2 Comparison about Different Attention Module

To prove the effectiveness of our SE_variant module, we selected three 3D attention module and three 2D attention module shown in Table 1 to replace the SE_variant, then conduct experiments of different attention modules on S3DIS dataset. As Table I indicates, SE_variant method achieves the best mIoU (65.3%) in all attention modules and exceeds the method without any attention by 1%. For each category of IoU, SE_variant method gets the highest values in two of the ten categories. Point-attn[18], A-SCN[19] and Offset-attn are all composed of self-attention modules. Compared with them, the simple structure of SE_variant or SE can help improve network performance more effectively. In addition, we find that channel information is more important than spatial information for point cloud features, as SE_variant and SE are more effective than Non-local[20] and Convolutional Block Attention Module (CBAM)[21].

Fig.5 compares the two evaluation metrics (mIoU and oAcc) after adding different attention modules. It is easy to find that although SE obtains the best value under the metric of oAcc, the results of SE_variant and CBAM are also good and not far behind. It is worth noting that this metric is not a good measure of semantic segmentation ability due to the unbalanced nature of the classes[22].

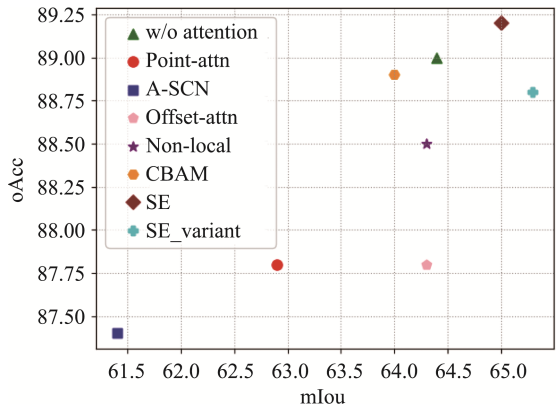

Fig.5 Overall Results (%) Testing on Area 5, S3DIS dataset. (oAcc: overall accuracy)

Table 2 shows several metrics that can explain the complexity. Because the parameter quantity of each attention module is different, adding different modules after each Encoder will also increase the whole network size more or less. We can find SE_variant not only achieves the best results, but also increases the number of parameters and FLOPs less. However, some attentions, like Point-Attention, A-SCN and Offset-Attention require more computational resources such as larger FLOPs. Moreover, the testing time of all modules is at the same level.

From the experimental results, we find some insights about using attention mechanism to improve the semantic segmentation performance of 3D point clouds. The self-attention module is not preferable for 3D point cloud data because of long-range dependencies and high computational resources. While the compact modules like SE and our SE_variant can improve the effectiveness of 3D feature refinement. Moreover, we find that when designing the attention module for point cloud feature representation, the performance of channel information is better than spatial information.

Table 1 Semantic Segmentation Results Testing on Area 5, S3DIS Dataset (w/o: without)

	Method	Ceiling	Floor	Wall	Beam	Column	Window	Door	Table	Chair	Sofa	Bookcase	Board	Clutter	mIoU
	w/o Attention	94.4	98.3	81.3	0.0	6.6	58.3	58.7	78.6	88.6	67.5	75.0	70.7	58.3	64.3
3D Attention	Point-attn[18]	93.5	97.8	80.0	0.0	13.4	54.5	56.1	75.5	85.1	64.5	71.9	68.4	55.7	62.8
3D Attention	A-SCN[19]	93.8	97.9	79.1	0.0	14.6	**63.3**	43.2	72.0	86.9	57.7	68.0	66.7	55.1	61.4
3D Attention	Offset-attn[8]	94.5	98.3	80.1	0.0	17.8	60.7	64.6	76.1	87.7	57.4	73.5	64.7	59.9	64.3
2D Attention	Non-local[20]	94.9	**98.5**	80.2	0.0	20.4	57.0	**65.2**	76.9	**88.7**	58.8	70.4	65.7	59.4	64.3
2D Attention	CBAM[21]	**95.3**	98.2	80.8	0.0	19.2	53.6	57.7	**77.7**	87.5	57.7	72.5	70.6	**61.0**	64.0
2D Attention	SE[14]	95.1	**98.5**	**82.1**	0.0	**20.9**	53.5	60.3	77.5	88.3	59.7	73.7	**75.8**	59.8	65.0
2D Attention	SE_variant	94.0	98.4	81.0	0.0	16.3	59.2	61.5	77.4	88.0	**67.9**	**74.1**	71.6	59.7	**65.3**

Table 2 Model Complexity of Different Attention Modules, Evaluated on S3DIS Dataset (Counted by The First Attention Module in The Encoder)

	Method	Parameters ($\times 10^3$ / attention)	Parameters' Proportion	FLOPs ($\times 10^3$ /attention)	FLOPs' Proportion	Testing Time (s/task)
3D Attention	Point-attn[18]	5.440	0.044%	5652.480	0.180%	4.7
3D Attention	A-SCN[19]	5.440	0.044%	5652.480	0.180%	4.6
3D Attention	Offset-attn[8]	9.632	0.077%	10010.624	0.317%	4.8
2D Attention	Non-local[20]	2.400	0.020%	2547.712	0.081%	4.6
2D Attention	CBAM[21]	1.028	0.009%	138.256	0.004%	4.6
2D Attention	SE[14]	1.024	0.009%	66.568	0.002%	4.7
2D Attention	SE_variant	1.024	0.009%	133.136	0.004%	4.8

4.3 Ablation Studies about Pooling Method in SE_variant

To verify the effectiveness of the proposed pooling method, we design several variants of the SE_variant module to investigate the effect of max-pooling and avg-pooling methods as shown in Table 3.

Table 3 Ablation Studies about Pooling Methods Testing on Area 5, S3DIS Dataset

Baseline	PAConv	SE_variant		mIoU
		Max	Avg	
√	√			64.3
√	√	√		64.4
√	√		√	65.0
√	√	√	√	**65.3**

We found that only using max-pooling or avg-pooling method can achieve 0.1% or 0.7% gain in term of mIoU, but using both pooling methods further boosts the performance with 0.3%. Based on the advantages of local max and mean features, we conclude that the best form of SE_variant module is using mixed local aggregation.

4.4 Ablation Studies about Effects of Network Components

In this section, we make several variants of our model to verify the contributions of different components as shown in Table 4. Baseline method represents the PointNet++ network. It can be observed that PAConv based on position information of points provides effective global context features, improving by 7.7%. SE_variant module promotes performance based on channel information, about 1.1%. In the end, the best value of mIoU is achieved by introducing both parts at the same time. These demonstrates that both PAConv and SE_variant module is crucial in the proposed method.

Table 4 Ablation Studies about Network Components Testing on Area 5, S3dis Dataset

Baseline	PAConv	SE_variant	mIoU
√			56.6
√		√	57.7
√	√		64.3
√	√	√	**65.3**

4.5 Semantic Segmentation Results and Visualization

1) S3DIS: Table 5 shows the performance comparison of our network with other state-of-the-art methods on the S3DIS dataset under the same experimental environment configuration. Notably, we significantly outperform the competitors regarding mIoU (65.3%), improved by 8.7% over the baseline network (PointNet++), and reach the best IoU in categories ceiling, chair, bookcase and board. Moreover, we visualize the segmentation results of baseline and our network for three scenes in S3DIS dataset as shown in Fig.6, including conference room, lobby and office. It is obvious that the segmentation effect of our network in red box has been significantly improved. For example, in the first row, our ceiling segmentation result is much better than baseline, as are the board in the second row and the bookcase in the third row.

2) ScanNetV2: We also conduct contrast experiments on ScanNetV2 dataset with other method as shown in Table 6. We surpass other methods in five out of the twenty categories. In particular, we perform well on small objects such as chair, table and shower, the visualization of semantic segmentation results is shown in Fig.7. We can

Table 5 Comparison with the Typical Streams of Methods Testing on Erea5, S3DIS Dataset

Method	Ceiling	Floor	Wall	Beam	Column	Window	Door	Table	Chair	Sofa	Bookcase	Board	Clutter	mIoU
PointNet[4]	88.2	97.7	72.6	0.0	7.1	51.5	20.2	57.7	43.1	9.9	46.3	33.8	33.2	43.8
PointNet++[3]	92.1	97.9	78.4	0.0	19.8	56.1	29.8	71.7	79.7	34.4	67.3	59.8	48.9	56.6
PointCNN[12]	92.3	98.2	79.4	0.0	17.6	22.8	62.1	74.4	80.6	31.7	66.7	62.1	56.7	57.3
SCF-Net[24]	91.5	95.7	80.8	0.0	16.6	**61.5**	35.4	75.2	**88.0**	67.1	70.6	65.8	51.9	61.5
BAAF-Net[22]	91.9	96.9	**83.0**	0.0	**26.4**	61.0	60.4	79.3	87.4	60.0	70.1	65.4	53.5	64.3
KPConv[10]	93.9	**98.5**	81.9	0.0	16.7	51.8	**71.7**	**90.9**	81.0	**75.6**	59.3	62.0	**60.8**	65.1
Ours	**94.0**	98.4	81.0	0.0	16.3	59.2	61.5	77.4	**88.0**	67.9	**74.1**	**71.6**	59.7	**65.3**

Table 6 Comparison with the Typical Streams of Methods on ScanNetV2 Dataset

Method	Wall	Floor	Cabinet	Bed	Chair	Sofa	Table	Door	Window	Book-shelf	Picture	Counter	Desk	Curtain	Refri-Gerator	Shower	Toilet	Sink	Bath	Other furniture	mIoU
PointNet++[3]	52.3	67.7	25.6	47.8	36.0	34.6	23.2	26.1	25.2	45.8	11.7	25.0	27.8	24.7	21.1	14.5	54.8	36.4	**58.4**	18.3	33.9
SPLATNet[25]	**69.9**	92.7	31.1	51.1	65.6	51.0	38.3	19.7	26.7	60.6	0.0	24.5	32.8	40.5	0.1	24.9	59.3	27.1	47.2	22.7	39.3
TangentConv[26]	63.3	91.8	36.9	**64.6**	64.5	**56.2**	42.7	27.9	35.2	47.4	14.7	35.3	28.2	25.8	28.3	29.4	61.9	**48.7**	43.7	29.8	43.8
3DMV[23]	60.2	79.6	**42.4**	53.8	60.6	50.7	41.3	**37.8**	**53.9**	**64.3**	**21.4**	31.0	**43.3**	**57.4**	**53.7**	20.8	**69.3**	47.2	48.4	**30.1**	**48.4**
Ours	69.8	**93.0**	42.2	59.5	**72.4**	53.3	**55.4**	32.5	39.5	63.5	11.3	**44.7**	39.5	39.3	22.5	**34.9**	63.5	33.5	54.6	27.6	47.6

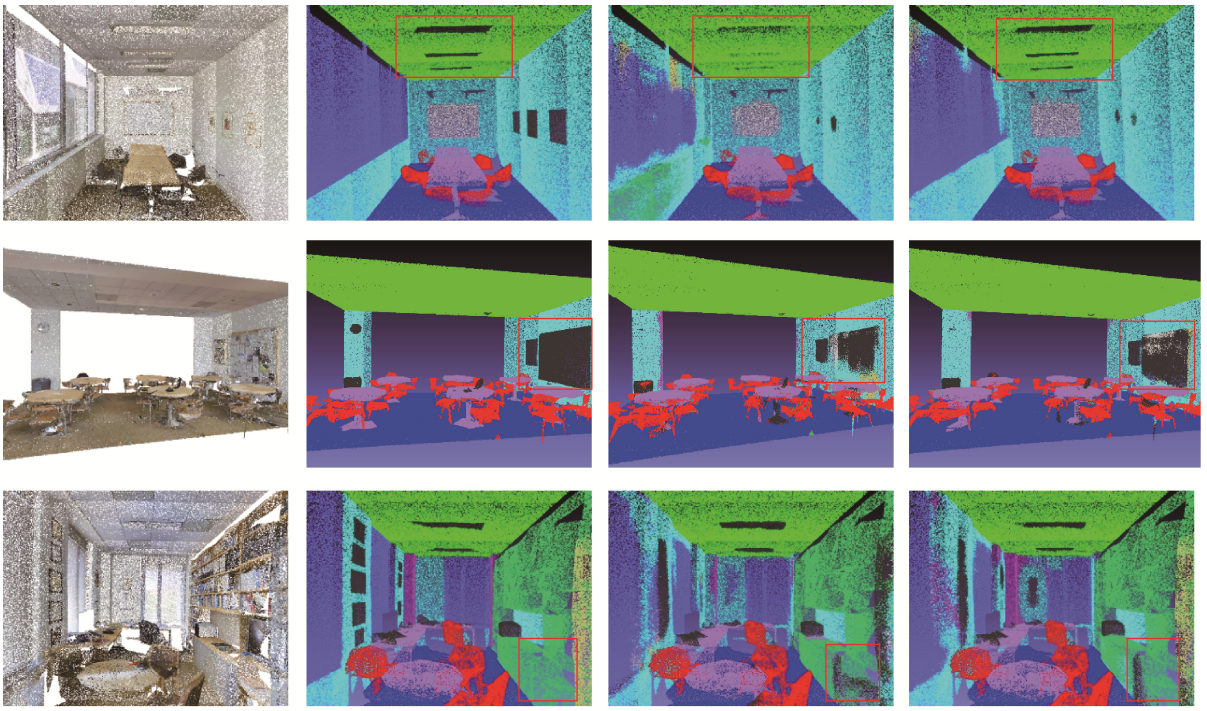

Fig.6 Visualization of Semantic Segmentation Results for Three Scenes in S3DIS Dataset

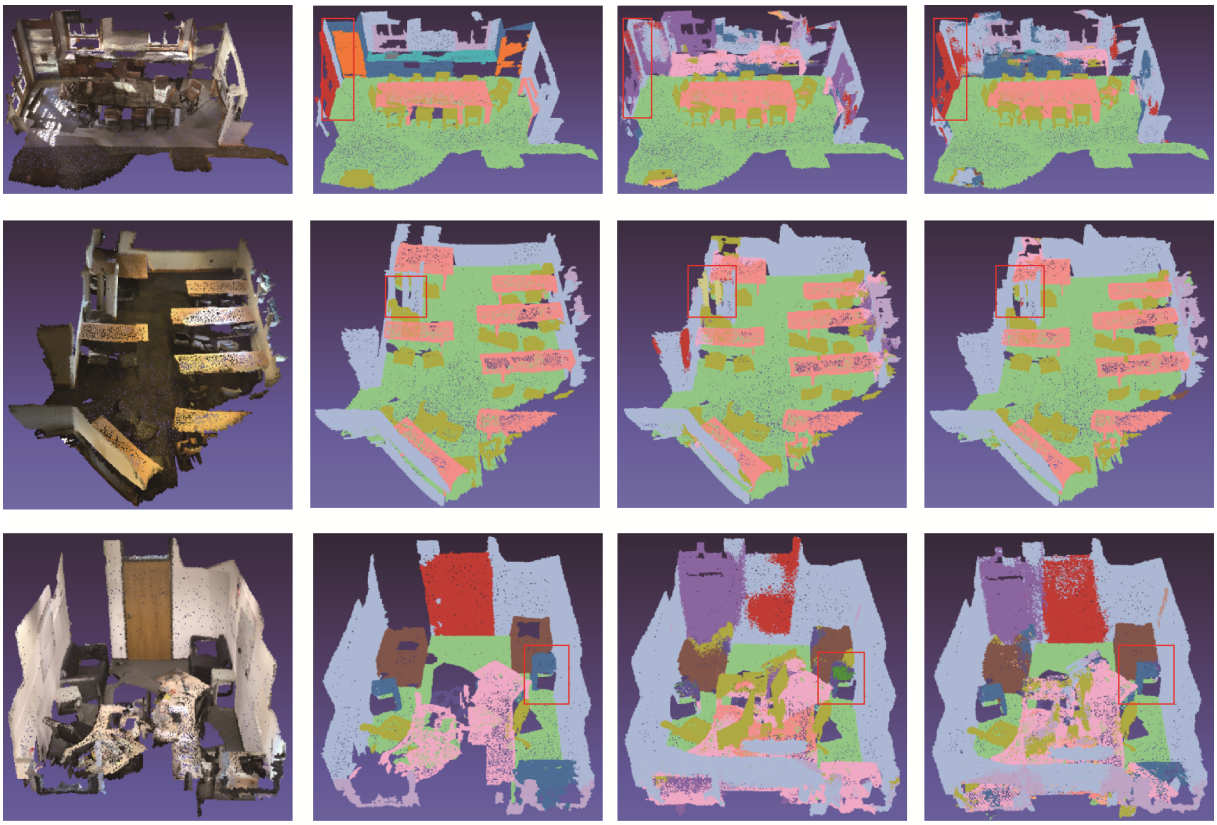

Fig.7 Visualization of Semantic Segmentation Results for Three Scenes in ScanNetV2 Dataset

see that our door segmentation result in the first row is better than baseline, as are the wall in the second row and the chair in the third row. The excellent results can be attributed to our SE_variant module, which thoroughly considers relevant information between channels. In general, compared with the baseline, our network has improved 13.7% mIoU, but it is slightly behind 3DMV[23].

5 Conclusion

In this paper, we propose a 3D point cloud semantic segmentation based PAConv and SE_variant. PAConv constructs the convolution kernel through dynamic data, generates adaptive coefficients based on the position information of points, and flexibly utilizes the irregular geometric structure of 3D point clouds. SE_variant module we proposed makes full use of the channel information of point clouds by fusing different pooling methods, capturing the global perception from a broad perspective in feature space. We conduct extensive experiments and ablation studies to validate the effectiveness of the method, achieve 65.3% mIoU in S3DIS and 47.6% mIoU in ScanNetV2. In addition, we provide some suggestions for understanding the attention mechanisms of 3D point cloud semantic segmentation by comparing the complexity and effect of different attention modules.

References

[1] Chu S R, Review of Segmentation of Point Clouds in Complex Scenes, in: *Modern Computer*, China, 2021, pp. 111-116.

[2] Fang Y R, A Review of Three-dimensional Point Cloud Segmention, in: *Metrology & Measurement Technique*, China, 2021, pp. 52-55.

[3] R.Q. Charles, L. Yi, H. Su, and L.J. Guibas, PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, in: *Proceedings of the conference and workshop on neural information processing systems, Long Beach, CA*, 2017, pp. 5099-5108.

[4] R.Q. Charles, Su H, Mo K, et al, Pointnet: Deep learning on point sets for 3d classification and segmentation, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652-660.

[5] Atzmon M, Maron H, Lipman Y, Point convolutional neural networks by extension operators, in: *ACM Transactions on Graphics*, 2018, pp. 1-12.

[6] Wu W, Qi Z, Fuxin L, Pointconv: Deep convolutional networks on 3d point clouds, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 9621-9630.

[7] Q. Hu, B. Yang, L. Xie, et al, RandLA-Net: Efficient semantic segmentation of large-scale point clouds, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2020, pp. 11108-11117.

[8] Guo M H, Cai J X, Liu Z N, et al, Pct: Point cloud transformer, in: *Computational Visual Media*, 2021, pp. 187-199.

[9] Liu Y, Fan B, Meng G, et al, Densepoint: Learning densely contextual representation for efficient point cloud processing, in: *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 5239-5248.

[10] Thomas H, Qi C R, Deschaud J E, et al, Kpconv: Flexible and deformable convolution for point clouds, in: *Proceedings of the IEEE international Conference on computer vision*, 2019, pp. 6411-6420.

[11] Xu Y, Fan T, Xu M, et al, Spidercnn: Deep learning on point sets with parameterized convolutional filters, in: *Proceedings of the European Conference on Computer Vision*, 2018, pp. 87-102.

[12] Li Y, Bu R, Sun M, et al, PointCNN: Convolution On X-Transformed Points, in: *Proceedings of the conference and workshop on neural information processing systems*, Montreal, Canada, 2018, pp. 820-830.

[13] Xu M, Ding R, Zhao H, et al, Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2021, pp. 3173-3182.

[14] Hu J, Shen L, Sun G, Squeeze-and-excitation networks, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7132-7141.

[15] I. Armeni, O. Sener, A.R. Zamir, et al, 3d semantic parsing of large-scale indoor spaces, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1534-1543.

[16] Dai A, Chang A X, Savva M, et al, Scannet: Richly-annotated 3d reconstructions of indoor scenes, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5828-5839.

[17] Zhu X, Zhou H, Wang T, et al, Cylindrical and asymmetrical 3d convolution networks for lidar segmentation, in: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 9939-9948.

[18] Feng M, Zhang L, Lin X, et al, Point attention network for semantic segmentation of 3D point clouds, in: *Pattern Recognition*, 2020, 107, 107446.

[19] Xie S, Liu S, Chen Z, et al, Attentional shapecontextnet for point cloud recognition, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4606-4615.

[20] Wang X, Girshick R, Gupta A, et al. Non-local neural networks, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794-7803.

[21] S. Qiu, S. Anwar, N. Barnes, Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2021, pp. 1757-1767.

[22] Qiu S, Anwar S, Barnes N, Pnp-3d: A plug-and-play for 3d point clouds, in: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021, 3137794

[23] Dai A, Nießner M, 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation, in: *Proceedings of the European Conference on Computer Vision*, 2018, pp. 452-468.

[24] S. Fan, Q. Dong, F. Zhu, et al., SCF-Net: Learning spatial contextual features for large-scale point cloud segmentation, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2021, pp. 14504-14513.

[25] Su H, Jampani V, Sun D, et al, Splatnet: Sparse lattice networks for point cloud processing, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2530-2539.

[26] Tatarchenko M, Park J, Koltun V, et al. Tangent convolutions for dense prediction in 3d, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3887-3896.

Author Biographies

ZHANG Ying received the Ph.D. degree from Hunan University in 2010. He is now a lecturer in Xiangtan University. His research interests include the development of electric vehicles controller and automotive Internet applications.

E-mail: zhangying@xtu.edu.cn

SUN Yue received the B.E. degree from Hunan University of Technology and Business in 2020. She is now a M.E. candidate in Xiangtan University. Her main research interests include pattern recognition and image processing.

E-mail: sunyue1049@163.com

WU Lin received the B.E. degree from Xiangtan University in 2021. She is now a M.E. candidate in Xiangtan University. Her main research interests include pattern recognition and image processing.

E-mail: wl119060@163.com

ZHANG Lulu received the B.Sc. degree from University of Science and Technology Liaoning in 2020. She is now a M.E. candidate in Xiangtan University. Her main research interests include pattern recognition and image processing.

E-mail: 208366092@qq.com

MENG Bumin received the Ph.D. from Hunan University in 2018. He is now a lecturer in Xiangtan University. His research interests include the development of electric vehicles controller and automotive Internet applications.

E-mail: mengbm@163.com