Pattern Recognition Letters 133 (2020) 327–333

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Feature fusion network based on attention mechanism for 3D semantic segmentation of point clouds

Heng Zhou[a], Zhijun Fang[a,*], Yongbin Gao[a], Bo Huang[a], Cengsi Zhong[a], Ruoxi Shang[b]

[a] Shanghai University of Engineering Science, 333 Longteng Road, Songjiang District, Shanghai, Shanghai 201620, China
[b] UC Berkeley, University of California, Berkeley, CA 94720, USA

ARTICLE INFO

Article history:
Received 17 June 2019
Revised 14 February 2020
Accepted 15 March 2020
Available online 16 March 2020

Keywords:
3D Semantic segmentation
Point clouds
Feature fusion
Attention mechanism

ABSTRACT

3D scene parsing has always been a hot topic and point clouds are efficient data format to represent scenes. The semantic segmentation of point clouds is critical to the 3D scene, which is a challenging problem due to the unordered structure of point clouds. The max-pooling operation is typically used to obtain the order invariant features, while the point-wise features are destroyed after the max-pooling operation. In this paper, we propose a feature fusion network that fuses point-wise features and local features by attention mechanism to compensate for the loss caused by max-pooling operation. By incorporating point-wise features into local features, the point-wise variation is preserved to obtain a refined segmentation accuracy, and the attention mechanism is used to measure the importance of the point-wise features and local features for each 3D point. Extensive experiments show that our method achieves better performances than other prestigious methods.

1. Introduction

The prevalance of depth camera or Laser makes the acquisition of 3D data relatively easy, more researchers have turned their attention to 3D scene parsing. 3D scene parsing includes three tasks:3D object detection [13,15,19], 3D object recognition [14,16,22,24,29] and 3D semantic segmentation [9,11,16,18,24,25]. In these tasks, 3D semantic segmentation is more challenging compared to the other two tasks due to the demand of element-wise classification as well as unordered and unstructured properties of its input data. Reviewing existing literature, many researchers have made great progress, which can be divided into three categories. The first transforms point clouds into voxelized occupancy grids [13,17,27,30], which can be applied to 3D CNNs. However, this method is memory consuming and inefficient. The second uses 2D multi-view images to represent 3D shapes [14,15,17,22], then 2D CNNs are applied to capture feature information. Yet 2D images do not fully show 3D geometry information, so information loss inevitably exists. The last one analyzes the raw point clouds directly [7,9,11,16,18,24,26], which has good performances in terms of speed and accuracy, and one of the most representative networks is PointNet [16]. Later many researchers made improvements based on PointNet architecture [7,8,11,18] and obtained impressive performances on benchmarks.

However, PointNet based methods typically use the max-pooling operation to ensure order invariance but with the price of losing geometry information. The max-pooling operation aggregates a batch of features into one feature, resulting in discarding of the rest features. We take RSNets [7] architecture as an example to give a thorough explanation and our implementation is also modified on it. The brief architecture of RSNets is shown as Fig. 1. The input data of RSNets is raw point clouds, and ***point-wise features*** of each point are extracted by feature extraction module. The slice pooling layer projects *unordered* and *unstructured* point clouds into an *ordered* and *structured* sequence of feature vectors by first grouping points into slices according to their coordinates and then aggregating features of points in each slice into one feature by max-pooling operation and finally generating ordered and structured feature sequences. These feature sequences are fed into RNN layers, making the information from one slice flow to another slice, which achieves information interaction among point clouds. The slice unpooling layer assigns the new feature sequences generated by RNN layers back to each point by reversing the projection. Specifically, each point in the same slice obtains the same features, which are called ***local features*** [21] because of information interaction among points. At last, a classification network is applied to produce semantic results. During the whole process, information loss is both existed in slice pooling layer and slice unpooling layer. Before slice pooling layer, every point in a slice has its unique fea-

* Corresponding author.
E-mail addresses: zjfang@gmail.com (Z. Fang), gaoyongbin@sues.edu.cn (Y. Gao).

https://doi.org/10.1016/j.patrec.2020.03.021
0167-8655/

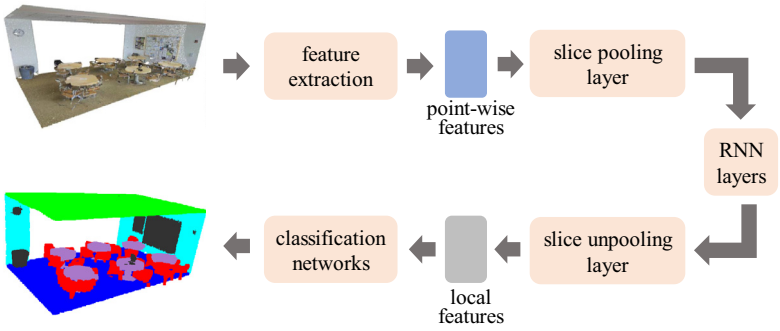

Fig. 1. The brief architecture of RSNets.

tures while after slice unpooling layer, each of them has the same features, which is harmful to the accuracy of classification.

In this paper, we propose a novel feature fusion method based on attention mechanism to address the problems mentioned above. We add a feature fusion network on the basis of RSNets, where point-wise features and local features are fused by attention mechanism to compensate for the information loss caused by the max-pooling operation. The point-wise features come from the feature extraction module and the local features are generated by the slice unpooling layer. These two kinds of features are the inputs of the fusion network. The local features are first updated by a convolution layer, and next concatenated with point-wise features. Then the local features are normalized to a weighted map after passed through a convolution layer. Finally they are multiplied with point-wise features that are updated by a convolution layer. Point-wise features are added to multiplication result by element-wise operation to produce new features. There are two feature fusions and both use the same framework. In the second fusion, point-wise features are replaced by the result generated from the first fusion process rather than the point-wise features. At the end of our architecture, the result comes from second fusion and local features are added in element-wise then the addition result is fed into a MLP(multilayer perceptron) to produce a label for each point.

The performances of our method are validated on two challenging benchmark datasets. Both the S3DIS dataset [1] and the ScanNet dataset [2] are large-scale real-world datasets. Compared with the performances of RSNets, Our method improves the mean IOU by 1.05%, the mean accuracy by 2.68% on S3DIS dataset. And meanwhile on ScanNet dataset, it improves the mean IOU by 1.35%, the mean accuracy by 3.25%. The key contributions of this paper are as follows:

- We propose to fuse point-wise features and local features for 3D semantic segmentation. For semantic segmentation networks of point clouds, such as PointNet [5,7,8,16,18,19], the problem of unordered point clouds can be solved by using max-pooling operation while discarding the information of non-maximal points. Our proposed feature fusion network can solve the problem by incorporating the point-wise features into local features. Thus, the point-wise variation is preserved to obtain a fine-grained point cloud segmentation.
- This paper is the first one that applies attention mechanism into semantic segmentation on point clouds. The attention mechanism can measure the contributions of features to the segmentation accuracy and can effectively remove redundant features in the fusion process, leaving useful information for classification.

The remainder of this paper is organized as follows: we first review related works in Section 2. Then our proposed architecture is presented in detail in Section 3. Section 4 reports all experimental results and Section 5 makes conclusions.

2. Related works

In this section, we briefly introduce several representative 3D data representations and corresponding methods. At last, the progress in the feature fusion on point clouds is shortly reviewed.

2.1. Voxelized volumes

In [4,13,17,28,30], the authors transformed 3D point clouds into voxelized occupancy grids and used 3D CNNs to perform semantic segmentation. However, there were some flaws using this method. First, quantization artifacts and information loss arose when transforming 3D point clouds into voxelized occupancy grids. Meanwhile, because of the sparsity of data, it is time and memory consuming when applying 3D CNNs. On the other hand, the size of input curb was relatively small, limited by the memory of 3D CNNs. Later, many researchers have been working on solving the problem of computational consuming and data sparseness. In [30], the authors made an attempt to reduce the computation by sampling the data before sending them to the networks. In [20], the authors designed OctNet to use the sparseness of the data to hierarchically partition the space using a set of unbalanced octrees to achieve a deep and high-resolution 3D convolutional network. While many of these works focused on solving the problems of data sparsity and computation, few of them made attempts to solve quantization artifacts and information loss.

2.2. Multi-view renderings

Another form of 3D data representations is multi-view rendering images. In [15,23], the authors projected the 3D shapes into 2D images at different viewpoints, then processed them by applying 2D CNNs and finally combined the 2D segmentation results to obtain 3D semantic segmentation results. However, there was a fatal drawback. It was inevitable for the loss of geometric structure information that was very important in 3D data when 3D shapes were projected to 2D images. Therefore, it requires a more considerate way to use this data format for 3D semantic segmentation.

2.3. Point clouds

Many researchers turned to use point clouds as input [3,5–8,10–12,16,18,25,26]because of the flaws of voxelized volumes and multi-view renderings. In [16], the authors firstly proposed an architecture that analyzed the point clouds, which was referred to as PointNet. In this framework, point-wise feature representations for each point were first produced and aggregated to a global feature by using max-pooling operation, then two features were concatenated and fed to a MLP to get segmentation results. In the subsequent literature, the authors put forward PointNet++ [18]architecture for the defect of PointNet in local features extraction and used

the hierarchical neural networks to obtain the local geometric information of the point clouds. In [11], the authors took place of the common convolutional layer by χ-Conv when dealing with point clouds, in order to overcome the problem of unordered property of point clouds. In [26], the authors proposed EdgeConv module, which was based on graph neural networks. This module incorporated local neighborhood information and could be stacked or recurrently applied to learn global shape properties. In [25], the authors proposed a similar group recommendation network (SGPN) to provide an intuitive learning framework for 3D instance segmentation on point clouds.

2.4. Point clouds feature fusion

In the existing literature, we have not seen the fusion between point clouds features. But the fusion between point clouds and other forms of features was well studied. Such as the fusion of point clouds features and multi-view images features, and the fusion of point clouds features and RGB images. In [14], the authors projected point cloud data into multi-view images, then fused multi-view images and visual data together and fed them into CNN networks. In [29], the author used two networks to extract features from point cloud data and multi-view data. Then the attention mechanism was applied to fuse the two features together to achieve the goal of 3D shape recognition.

3. Method

3.1. Problem statement

The unordered point clouds are represented as $X = \{x_1, x_2, \ldots, x_i, \ldots, x_n\}$ with $x_i \in \mathbb{R}^d$, and the corresponding labels are $L = \{l_1, l_2, \ldots, l_i, \ldots, l_K\}$. The task of semantic segmentation is to assign each point x_i with one of the K semantic labels. In our architecture, the input is raw point clouds and the output is $Y = \{y_1, y_2, \ldots, y_i, \ldots, y_n\}$ where $y_i \in L$ is the label of x_i. The max-pooling operation is usually used in networks to ensure order invariance. In mathematics, suppose the input of max-pooling operation is $F^{in} = \{f_1, f_2, \ldots, f_i, \ldots, f_n\}$, and the output is F^{out}, the operation can be described as follows:

$$F^{out} = \max_{f_i \in F^{in}} \{f_i\} \tag{1}$$

This formula indicates that max-pooling operation selects one element while the other elements are discarded directly. Back to RSNets, slice pooling layer aggregates features of points within one slice into one feature to represent the information of this slice. Meanwhile, slice unpooling layer assigns the features yielded by RNN layers to points in the slice, where each point has same features. Information loss exists obviously. This paper aims to address this problem. Details are illustrated below.

3.2. A brief introduction to RSNets architecture

The overall framework is shown as Fig. 2. The blue part is RSNets. The input feature extraction module performs convolution operation on input data with a series of multiple 1×1 convolution layers to extract features of each point, which are called point-wise features. The slice pooling layer takes the point-wise feature as input and groups points into slices according to their coordinates. Three slicing directions, including x, y and z axis. In each slice, the slice pooling layer aggregates the point-wise features of all points in this slice into one feature by max-pooling operation separately in three directions. In each direction, the features of each slice constitutes a sequence of feature vectors. These feature vectors are sent to RNN layers to make information of each slice flow to another. After this process, each slice has interaction with other slices. And in each slice, the slice unpooling layer assigns the same feature information to the points within this slice. Each point has features called local features [21] because of the interaction with other points.

In RSNets, the authors ignored the information loss in slice pooling layer. After slice unpooling layer, the points in the same slice have the same feature information while the unique features of each point should be considered for point-wise classification. Regarding this issue, we propose an idea with feature fusion that fuses point-wise features and local features to improve the accuracy of the classification.

3.3. Point clouds feature fusion network

Our detailed feature fusion network is shown in Fig. 3, the green part. The inputs are point-wise features and local features. Local features F^l with shape $n \times c_1$ (n points with features of c_1 dimensions) are first convoluted by a convolution layer and next concatenated with point-wise features F^p with shape $n \times c_2$ (n points with features of c_2 dimensions) and then sent to a convolution layer to extract feature information. Finally, a normalized function is applied to it to generate a weighted map. The concatenation function is defined as $\varphi(\cdot) = \mathbb{R}^{n \times c_1} \times \mathbb{R}^{n \times c_2} \rightarrow \mathbb{R}^{n \times (c_1 + c_2)}$. The normalized function is $\phi(\cdot) = sigmoid(\cdot)$. The weighted map $M(F^l, F^p)$ can be described as follows:

$$M(F^l, F^p) = \phi\left(C^*\left(\varphi(C^*(F^l), F^p)\right)\right) \tag{2}$$

where C^* denotes the convolution operation. The values of weighted map are in the range [0,1], which represents the significance of different features. Point-wise features are first updated by a convolution layer and then multiplied with the weighted map and finally added to the multiplication results to produce new features $F^*(F^l, F^p)$. The whole process is shown in the expression blow:

$$F^*(F^l, F^p) = \left(C^*(F^p) \otimes M(F^l, F^p)\right) \oplus C^*(F^p) \tag{3}$$

Where the symbols C^*, $\otimes$ and $\oplus$ denote the convolution operation, element-wise multiplication and element-wise addition respectively. There are two feature fusions in total and both share the same architecture. The first one is the fusion of the point-wise features and local features, where the point-wise features are generated by the input feature extraction module and the local features are obtained from the RNN layers. The second one is the fusion of the result from the first fusion and the local features. It is necessary to perform two fusions. The first fusion is to compensate for the loss caused by max-pooling operation and the fusion results are too low-level to perform point-wise classification, while the second fusion aims to produce fine-grained semantic features for semantic segmentation in order to improve the accuracy of classification. The experiment results also show that the performances of two feature fusions are better than the one of single feature fusion, which verifies our idea.

We have tried other fusion methods, such as concatenating the point-wise features and the local features directly or adding them in element-wise, while we found that the fusion method based on attention mechanism performed best. The reason is that concatenation just increases the dimension of features, which is too low-level to classify. Element-wise addition only changes the value of features, which damages the property of local features to classify. More detailed discussions are presented in ablation studies.

4. Experiments

To verify the performances of our proposed method and compare them with the state-of-the-art algorithms, we have evaluated

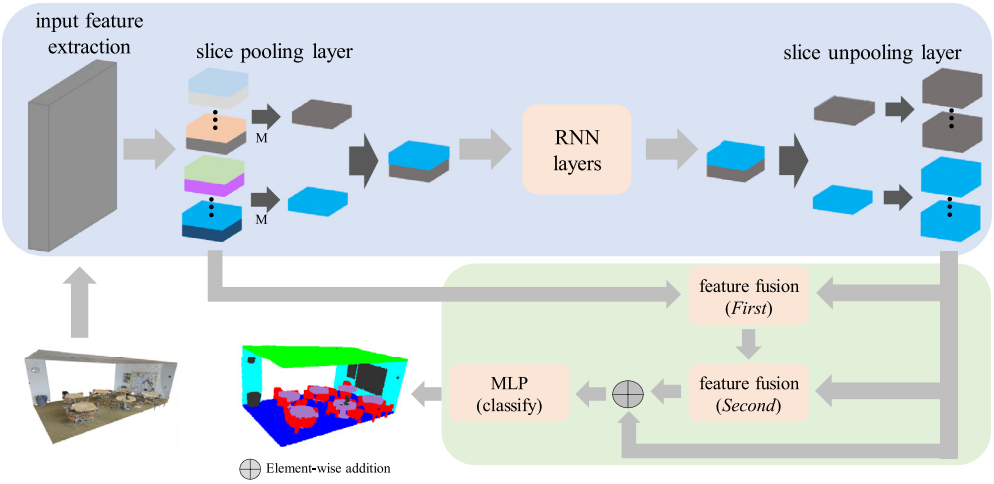

Fig. 2. Diagram of our proposed network. The blue part corresponds to RSNets and the green part is our feature fusion network. In slice pooling layer, M denotes the max-pooing operation. S_1 and S_2 indicate two slices respectively(just take two slices as an example to explain). In each slice, different color blocks indicate different features. The channel of all features is 64. The features in a slice are aggregated into one feature by max-pooling operation. These aggregated features form a feature sequence and flow to RNN layers. In slice unpooling layer, the same color blocks mean the same features and the channel of all features is 64. In the feature fusion network, there are two fusions and both use the same network. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

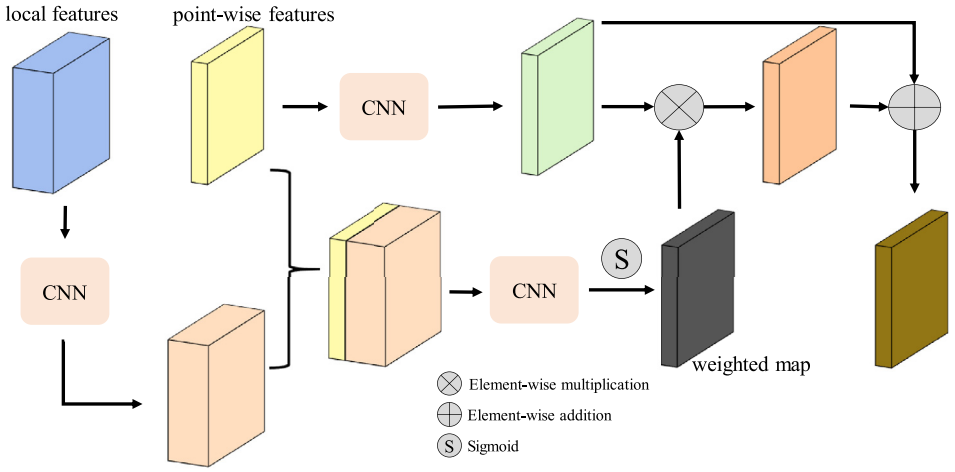

Fig. 3. Architecture of feature fusion networks. All CNNs are with one layer. The size of all convolution kernels are 1×1. The channels of local features and point-wise features are 64 and 64. The numbers of output channel of convolution on local features, point-wise features and concatenated features are 512, 64 and 64 respectively.

our method on two datasets, Stanford 3D dataset (S3DIS) [1] and ScanNet dataset [2]. These two datasets are both large-scale realistic 3D segmentation datasets.

We follow the strategies in RSNets [7] to process all datasets. For the S3DIS and ScanNet datasets, the data are divided into smaller cubes with fixed size. A fixed number of points are sampled from the cubes as the inputs of RSNets. The number is fixed as 4096 for these two datasets. In terms of RSNets, there are three 1×1 convolution layers with output channel number of 64, 64, and 64 respectively in the input feature extraction module. The implementation of RNN layers is a stack of 6 bidirectional RNN layers where the numbers of channels are 256, 128, 64, 64, 128, and 256. The channel of features output by slice unpooling layer is 64. In terms of proposed feature fusion networks, the kernel sizes of all convolutions with one layer are 1×1. The numbers of output channels of the convolution on local features, point-wise features and concatenating result are 512, 64 and 64, respectively. Three 1×1 convolution layers with output channel number of 512, 256 and K are used in the MLP, where K is the number of semantic categories. The last convolution layer in the MLP produces a predictable label for each point. And the cross entropy function is employed to compute errors. Our proposed architecture takes point clouds as the input data and generates labels for points after all modules, which makes it an end-to-end trainable network.

We take two widely used metrics: mean intersection over union(mIOU) and mean accuracy(mACC) as our principles to measure the segmentation performances. We first give the performances of our method on the S3DIS dataset and then comprehensive studies are conducted to validate various architecture choices in our method. At last, we report the performances on the ScanNet dataset.

4.1. Segmentation on the S3DIS dataset

We first report the performances of our method on the S3DIS dataset. The S3DIS dataset captures RGB-D point clouds from three buildings, including 271 rooms. The number of categories of point clouds tags is 13. We follow the division of the training set and the testing set in [24]. All the parameters in data processing are set as in RSNets. The initial learning rate is 0.001. The performances of our method are reported in Table 1. Besides the overall mean IOU and mean accuracy, the IOU of each category is also

Table 1
Results on Large-Scale 3D Indoor Space Dataset(S3DIS). Superscripts ***A*** and ***C*** denote data augmentation and post-processing (CRF) are used. IOU of each category is also reported here.

Method	mIOU	mACC	ceiling	floor	wall	beam	column	window	door	chair	table	bookcase	sofa	board	clutter
PointNet^{A}	41.90	48.98	88.80	97.33	69.80	0.05	3.92	46.26	10.76	52.61	58.93	40.28	5.85	26.38	33.22
3D CNN	43.67	–	–	–	–	–	–	–	–	–	–	–	–	–	–
3D CNN^{A}	47.67	54.91	90.17	96.48	70.16	0.00	11.40	33.36	21.12	**76.12**	70.07	57.89	37.46	11.16	41.61
3D CNN^{AC}	48.92	57.35	90.06	96.05	69.86	0.00	**18.37**	38.35	23.12	75.89	**70.40**	**58.42**	40.88	12.96	41.60
RSNet	51.93	59.42	93.34	98.36	79.18	0.00	15.75	45.37	50.10	65.52	67.87	22.45	**52.45**	**41.02**	43.64
Ours	**52.98**	**62.10**	**93.40**	**98.39**	**79.43**	**2.15**	17.03	**48.17**	**55.24**	66.09	66.62	52.65	24.93	39.99	**44.68**

Table 2
The varying number of fusion times on S3DIS dataset. *1* and *2* denotes one fusion and two fusions respectively.

Number of times	mIOU	mACC
1	51.02	59.40
2	**52.98**	**62.10**

Table 3
The varying styles of fusion on S3DIS dataset.

Style of fusion	mIOU	mACC
Element-wise addition	49.08	59.39
Concatenation	50.73	59.03
Based on attention mechanism	**52.98**	**62.10**

presented. Some segmentation results are visualized in Fig. 5. The performances of RSNets and previous state-of-the-art methods are reported in Table 1 as well. The results show that our method performs better than RSNets owing to the additional feature fusion network. In particular, compared to RSNets, our method improves the mean IOU by 1.05% and mean accuracy by 2.68%. The detailed per-category IOU results show that our method performs better than RSNets in more than half of all categories(7 out of 13).

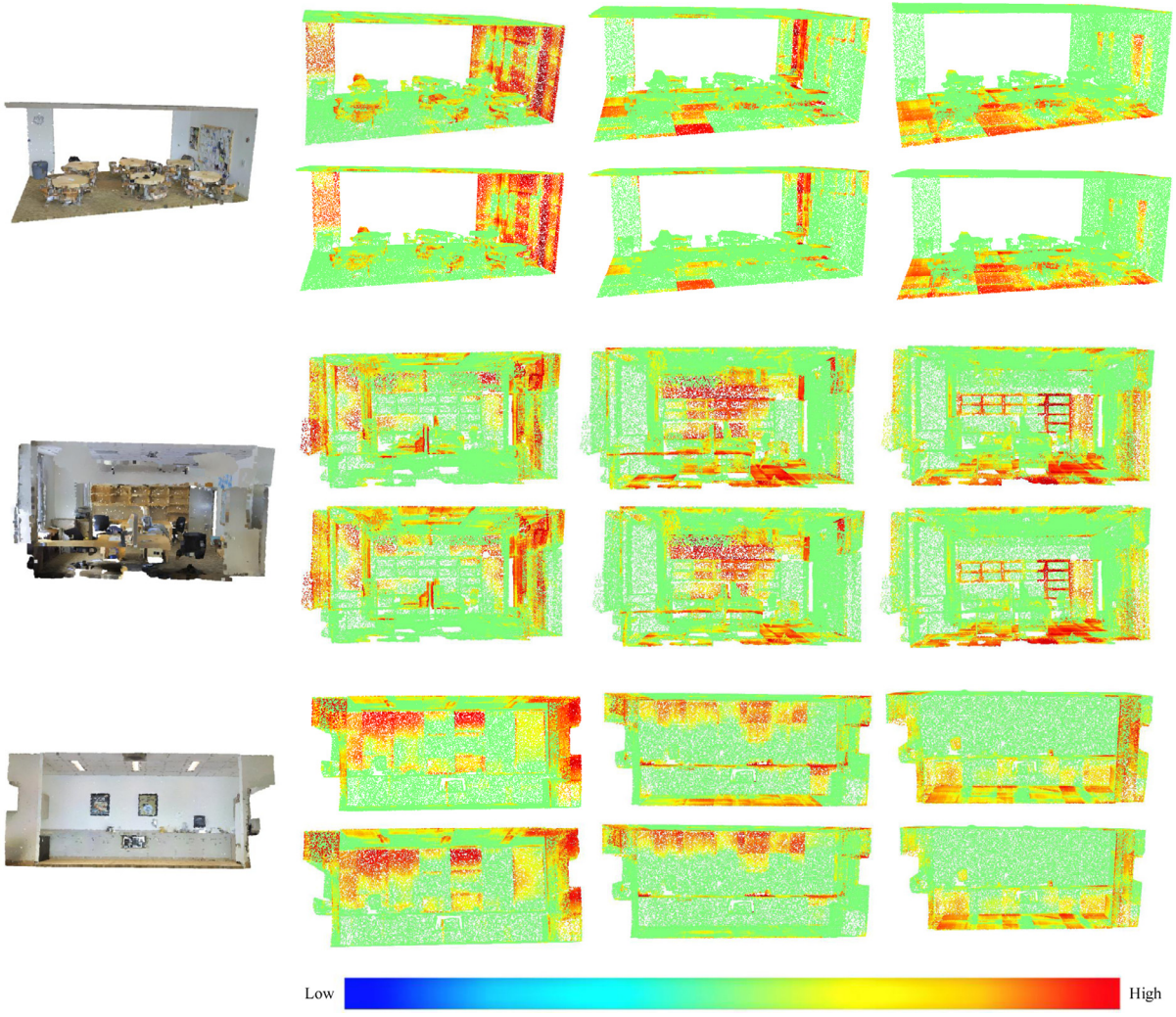

Fig. 4. There are three scenes in the figure. The attention maps of first fusion are listed on the top and the maps of second fusion are in the below. From left to right are maps with different feature channel. The color from blue to red indicates different weight values for each point feature. The high values tend to be more crucial for segmentation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
Results on the ScanNet dataset. IOU of each category is also reported here.

Method	mIOU	mACC	wall	floor	chair	table	desk	bed	book-shelf	sofa	sink
PointNet	14.69	19.90	69.44	88.59	35.93	32.78	2.63	17.96	3.18	32.79	0.00
PointNet+	34.26	43.77	77.48	92.50	64.55	46.60	12.69	51.32	52.93	52.27	30.23
RSNet	39.35	48.37	79.23	94.10	64.99	51.04	34.53	**55.95**	53.02	55.41	**34.84**
Ours	**40.67**	**51.62**	**82.36**	**95.24**	**85.80**	**51.88**	**36.29**	55.89	**53.42**	**55.96**	33.46

Method	bathtub	toilet	curtain	counter	door	window	shower curtain	refrigerator	picture	cabinet	other furniture
PointNet	0.17	0.00	0.00	5.09	0.00	0.00	0.00	0.00	0.00	4.99	0.13
PointNet+	42.72	31.37	**32.97**	20.04	2.02	3.56	27.43	18.51	0.00	23.81	2.20
RSNet	49.38	54.16	6.78	22.72	3.00	8.75	29.92	37.90	**0.95**	31.29	18.98
Ours	**51.89**	**56.79**	7.59	**24.98**	**6.84**	**10.66**	**32.10**	**39.25**	0.00	**32.45**	**20.59**

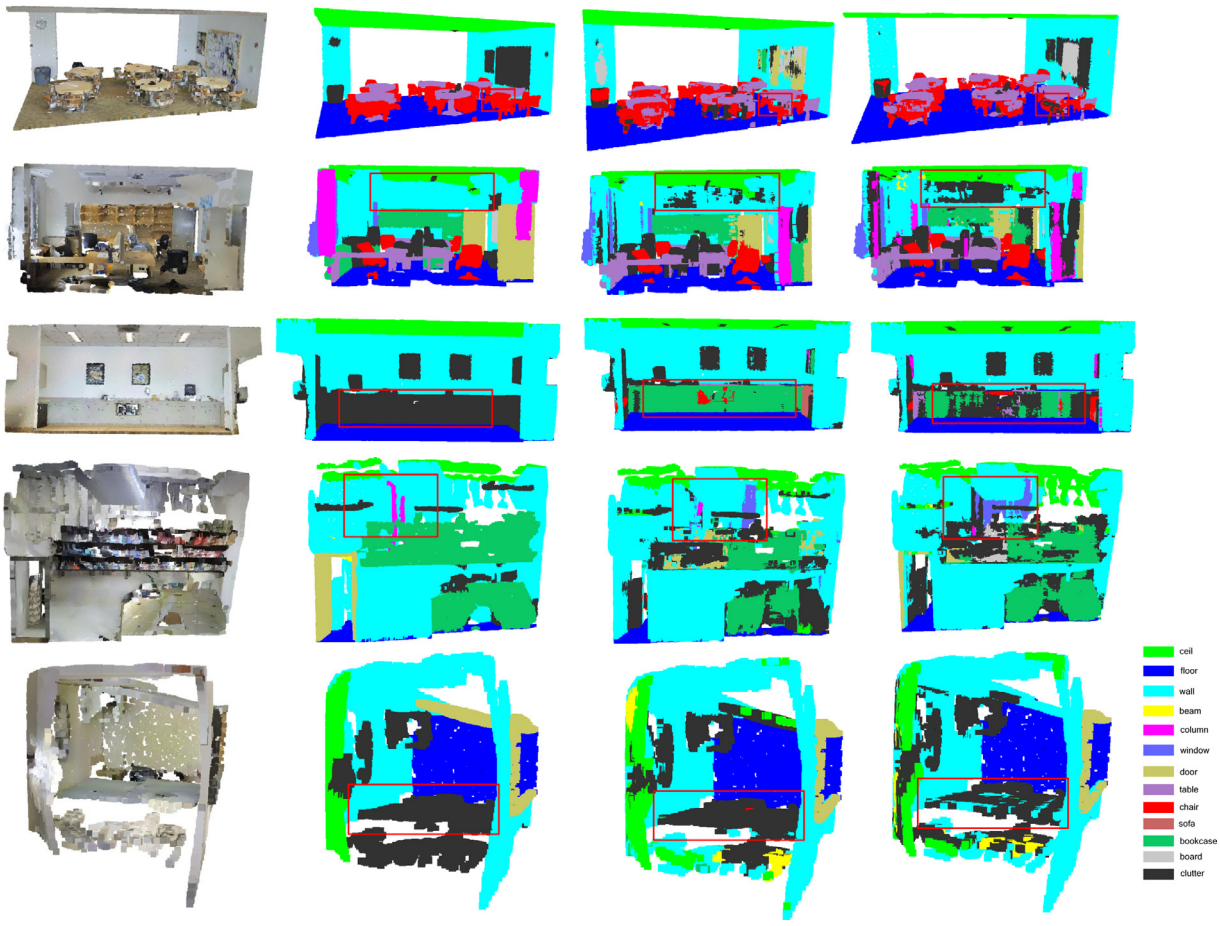

Fig. 5. Sample segmentation results on the S3DIS dataset. From left to right are the input scenes, ground truth, results produced by our method and results of RSNet.

From the segmentation results, we find that our method outperforms than RSNets at the junction of several objects(labeled by red bounding box in Fig. 5). We argue that it benefits from feature fusion. In RSNets, the points within a slice are assigned the same features by slice unpooling layer while in the junction areas, it can not ensure that the points belong to the same category. But the feature fusion makes sure that the points in one slice have both local features and the unique features of themselves. Therefore, our method can correctly classify the points even in junction areas.

4.2. Ablation studies

In this subsection, We discuss the choices of the number of fusions and the way to fuse. For the number of fusions, we have two candidate values: one and two. The results are reported in Table 2. It is clear that the performances of two fusions are better than the one of just one fusion. We argue the reason is that one fusion is just able to compensate for the information loss but does not produce fine-grained features for classification. And the second fusion can solve this issue perfectly.

For the design of the fusion network, we have explored several fusion methods, such as adding local features and point-wise features in element-wise style or concatenating them directly, which are inferior to the attention mechanism fusion method. The results are reported in Table 3. It is clear that the method based on attention mechanism outperforms than others which attribute to its function that pay more attention to key features and less to the

otherwise. The attention map is presented in Fig. 4. The points with high weight values in second fusion attention map are less than the one in first fusion, which means that the second fusion captures more fine-grained features.

4.3. Segmentation on the ScanNet dataset

We also have evaluated the performance on ScanNet dataset [2]. ScanNet dataset is a scene semantic labeling task with a total of 1513 scanned scenes. We use 1201 scenes for training and the rest for testing as in [7,18]. In our method, we only use *xyz* information and take mIOU and mACC as our metrics. As shown in Table 4, our method achieves better performances compared with RSNets. The results demonstrate the feature fusion network is useful to solve the problem caused by max-pooling operation in 3D point clouds semantic segmentation again.

5. Conclusion

In this paper, we introduce a novel method based on attention mechanism that fuses point-wise features and local features in semantic segmentation on point clouds, which can compensate for the information loss caused by max-pooling operation. There are two fusions in our method where the first is the fusion of point-wise features and local features and the second is the fusion of the result from the first fusion and local features. Meanwhile, the feature fusion network based on the attention mechanism effectively removes redundant features in fusion process and obtains new features containing more useful information to improve the accuracy of point clouds segmentation. The results of experiments have shown that our method performs better than RSNets, particularly in the junction area of several objects, which verifies the effectiveness of feature fusion based on attention mechanism.

Declaration of Competing Interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant 61772328, Grant 61802253, and Grant 61831018.

References

[1] I. Armeni, O. Sener, A.R. Zamir, H. Jiang, I. Brilakis, M. Fischer, S. Savarese, 3D semantic parsing of large-scale indoor spaces, in: The IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2016, pp. 1534–1543.
[2] A. Dai, A.X. Chang, M. Savva, M. Halber, T. Funkhouser, M. Niener, Scannet: richly-annotated 3D reconstructions of indoor scenes, in: The IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2017, pp. 5828–5829.
[3] A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, M. Niener, Scancomplete: large-scale scene completion and semantic segmentation for 3D scans, in: The IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2018, pp. 4578–4587.
[4] Y. Gao, F. Villecco, M. Li, W. Song, Multi-scale permutation entropy based on improved LMD and HMM for rolling bearing, Entropy 19 (2017) 176.
[5] L. Ge, Y. Cai, J. Weng, J. Yuan, Hand pointnet: 3D hand pose estimation using point sets, in: The IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2018, pp. 8417–8426.
[6] B. Graham, M. Engelcke, L. van der Maaten, 3D semantic segmentation with submanifold sparse convolutional networks, in: The IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2018, pp. 9224–9232.
[7] Q. Huang, W. Wang, U. Neumann, Recurrent slice networks for 3D segmentation of point clouds, in: The IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2018, pp. 2626–2635.
[8] M. Jiang, Y. Wu, C. Lu, Pointsift: a sift-like network module for 3D point cloud semantic segmentation, arXiv:1807.00652 (2018).
[9] L. Landrieu, M. Simonovsky, Large-scale point cloud semantic segmentation with superpoint graphs, in: The IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2018, pp. 4558–4567.
[10] T. Le, Y. Duan, Pointgrid: a deep network for 3D shape understanding, in: The IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2018, pp. 9204–9214.
[11] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, B. Chen, Pointcnn: convolution on χ-transformed points, Adv. Neural Inf. Process. Syst. (2018) 828–838.
[12] Y. Li, S. Pirk, H. Su, C.R. Qi, L.J. Guibas., Fpnn: field probing neural networks for 3D data, Adv. Neural Inf. Process. Syst. (2016) 307–315.
[13] D. Maturana, S. Scherer, Voxnet: a 3D convolutional neural network for real–time object recognition, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 922–928.
[14] G. Pang, U. Neumann, Fast and robust multi-view 3D object recognition in point clouds, in: 2015 International Conference on 3D Vision, 2015, pp. 171–179.
[15] G. Pang, U. Neumann., 3D point cloud object detection with multi-view convolutional neural network, in: 2016 23rd International Conference on Pattern Recognition (ICPR), 2016, pp. 585–590.
[16] C.R. Qi, H. Su, K. Mo, L.J. Guibas, Pointnet: deep learning on point sets for 3Dclassification and segmentation, in: The IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2017, pp. 652–660.
[17] C.R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, L.J. Guibas, Volumetric and multi-view CNNS for object classification on 3D data, in: The IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2016, pp. 5648–5656.
[18] C.R. Qi, L. Yi, H. Su, L.J. Guibas, Pointnet++: deep hierarchical feature learning on point sets in a metric space, Adv. Neural Inf. Process. Syst. (2017) 5099–5108.
[19] C.R.W.L. Qi, C. Wu, H. Su, L.J. Guibas, Frustum pointnets for 3D object detection from RGB-D data, in: The IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2018, pp. 918–927.
[20] G. Riegler, O.U. Ali, A. Geiger, Octnet: learning deep 3D representations at high resolutions, in: The IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2017, pp. 3577–3586.
[21] Y. Shen, C. Feng, Y. Yang, D. Tian, Mining point cloud local structures by kernel correlation and graph pooling, in: The IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2018, pp. 4548–4557.
[22] H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller, Multi-view convolutional neural networks for 3D shape recognition, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 945–953.
[23] H. Su, F. Wang, E. Yi, L.J. Guibas, 3D-assisted feature synthesis for novel views of an object, in: The IEEE International Conference on Computer Vision(ICCV), 2015, pp. 2677–2685.
[24] L. Tchapmi, C. Christopher, A. Iro, G. JunYoung, S. Savarese, Segcloud: semantic segmentation of 3D point clouds, in: 2017 International Conference on 3D Vision (3DV), 2017, pp. 537–547.
[25] W. Wang, R. Yu, Q. Huang, U. Neumann, Sgpn: similarity group proposal network for 3D point cloud instance segmentation, in: The IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2018, pp. 2569–2578.
[26] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M. M, Dynamic graph CNN for learning on point clouds, arXiv:1801.07829 (2018).
[27] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3D shapenets: a deep representation for volumetric shapes, in: The IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2015, pp. 1912–1920.
[28] D. Xu, D. Anguelov, A. Jain, Pointfusion: deep sensor fusion for 3D bounding box estimation, in: The IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2018, pp. 244–253.
[29] H. You, Y. Feng, R. Ji, Y. Gao, Pvnet: a joint convolutional network of point cloud and multi-view for 3D shape recognition, in: 2018 ACM Multimedia Conference on Multimedia Conference, 2018, pp. 1310–1318.
[30] Y. Zhou, O. Tuzel, Voxelnet: end-to-end learning for point cloud based 3D object detection, in: The IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2018, pp. 4490–4499.