Knowledge-Based Systems 247 (2022) 108769

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Cross self-attention network for 3D point cloud

Gaihua Wang [a,b], Qianyu Zhai [a,*], Hong Liu [a]

[a] School of Electrical and Electronic Engineering, Hubei University of Technology, Wuhan 430068, China
[b] Hubei Key Laboratory for High-efficiency Utilization of Solar Energy and Operation Control of Energy Storage System, Hubei University of Technology, Wuhan, 430068, China

ARTICLE INFO

Article history:
Received 30 December 2021
Received in revised form 19 March 2022
Accepted 5 April 2022
Available online 13 April 2022

Keywords:
Deep learning
Point cloud
Self-attention
Semantic segmentation
Shape classification
Multi-scale fusion

ABSTRACT

It is a challenge to design a deep neural network for raw point cloud, which is disordered and unstructured data. In this paper, we introduce a cross self-attention network (CSANet) to solve raw point cloud classification and segmentation tasks. It has permutation invariance and can learn the coordinates and features of point cloud at the same time. To better capture features of different scales, a multi-scale fusion (MF) module is proposed, which can adaptively consider the information of different scales and establish a fast descent branch to bring richer gradient information. Extensive experiments on ModelNet40, ShapeNetPart, and S3DIS demonstrate that the proposed method can achieve competitive results.

1. Introduction

The point cloud is a mainstream representation for 3D objects. Unlike 2D images, which are fixed in a grid, 3D point cloud is an unordered, redundant and unstructured data. It is a challenge that design a deep neural network to extract features of the 3D point cloud.

A large number of pioneers have made bold attempts to respond to this challenge. PointNet [1] is a landmark network that first utilizes multilayer perceptron (MLP) and asymmetric functions to process point cloud. It uses neural networks to process point cloud data without any preprocessing operations. Qi et al. [2] designed a hierarchical structure that aggregates similar features based on the metric space distances to better capture local features. Inspired by this idea, some works started to design different deep neural networks to complete the task of point cloud segmentation [3,4]. For instance, Hu et al. [5] tried to use random sampling to process large-scale point cloud and achieved remarkable effects. Qiu et al. [6] proposed BAAFNet to consider the geometric information and semantic information of the point cloud at the same time. Wang et al. [7] proposed a novel EdgeConv based on the dynamic graph to capture the relationship between any two points.

With the deepening of research, self-attention, which has achieved great success in natural language processing and 2D image processing tasks, has also been used to process 3D point cloud. As the core operation of transformer, self-attention can consider the relationship between input vectors and is not affected by input order. In other words, it has permutation invariance. Therefore, self-attention can easily process point cloud. Some researchers have proposed methods based on self-attention for point cloud segmentation and got excellent performance. For instance, Zhao et al. [8] designed the self-attention layer based on the characteristics of point cloud and explored the influence of position coding on segmentation results. Inspired by graph convolutional network, Guo et al. [9] proposed offset-attention to help the transformer realize better segmentation results. Although the self-attention mechanism can assist the network strengthen the relationship between each point, there are two issues that have been overlooked.

First, coordinate information and feature information are not treated properly. A point cloud is a collection of coordinates, features and some other attributes. Previous works usually pay attention to feature information or coordinate information separately. [5,10] only concatenate the 3 dimensions original coordinate information to model the geometric information during the feature extraction process. It is not enough for the model to learn the complete geometric information and may lead to the decline of model generalization ability and robustness. In 2D images, a reasonable solution is to use 3×3 convolution to consider the relative positional relationship in the process of extracting features. However, it is not possible to use a large convolution kernel in a point cloud because it is unordered data. If similar operation is performed in the point cloud, the ability of model feature extraction may be effectively improved.

* Corresponding author.
E-mail address: 20130006@hbut.edu.cn (Q. Zhai).

https://doi.org/10.1016/j.knosys.2022.108769
0950-7051/

Second, multi-scale feature is not considered. The classic segmentation structure requires down-sampling to compress features and then up-sampling to recover features. In the up-sampling process, if only fuse features with the same level, it may cause the feature representation to partially lose the information of different sizes.

To handle the first issue, a simple way is to treat the coordinates and features information of the point cloud equally. Herein, we design the cross self-attention (CSA) module to interactively consider the feature information and coordinate information of the point cloud. In the CSA module, all features and coordinates are selectively enhanced with each other. Then we build the feature extraction network of the point cloud by repeatedly stacking CSA modules, which is called CSANet. The second problem can be solved by low-level and high-level feature fusion. Therefore, we design the multi-scale fusion (MF) module to fuse feature from top to bottom in the segmentation part. Each MF module receive the features of the previous layer, and adaptively merge them together. In general, our work can be summarized as follows:

(1) We design a new CSA module, which not only has permutation invariance, but also can interactively extract coordinates and features information.

(2) A multi-scale fusion (MF) module is designed to adaptively capture features of different scales. It can be easily embedded into different layers of the network, bringing richer scale and gradient information to the network.

(3) Based on CSA and MF module, a cross self-attention network (CSANet) is designed to accomplish the task of point cloud classification and segmentation excellently. And it achieves competitive results in three mainstream datasets (ModelNet40 [11], ShapeNetPart [12] and S3DIS [13]).

The remainder of the paper is organized as follows: Section 2 reviews classical methods for point cloud semantic segmentation tasks. Details of the proposed method can be found in Section 3. Section 4 demonstrates the performance of the proposed method through extensive ablation and comparative experiments. Section 5 provides an objective summary of our work and gives an outlook on future research.

2. Related work

Projection-based and voxel-based methods: Early works usually perform special processing on the raw point cloud and then extract features through a neural network. SqueezeSeg [14] firstly projects the 3D point cloud to obtain the front view, then uses the SqueezeNet-based [15] convolutional network to extract and segment the input image, and finally uses the CRF (conditional random field) to further optimizes the segmentation results. Chen et al. [16] first projected 3D images into 2D images from different perspectives. Then 2D CNN (convolutional neural network) was used to extract features. Finally the features of multiple images were fused. Projection-based methods reduce the dimensionality and computational cost of point clouds by projection but inevitably cause a loss of spatial information. Maturana et al. [17] transformed point cloud voxels into 3D grids and then used 3D CNN to extract features. [18] extends U-Net [19] to the field of 3D data segmentation. Because voxelization at low resolution lose information, these methods need to maintain high resolution to retain the detail. It leads to a large memory overhead.

Point-based methods: Inspired by PointNet, a series of excellent methods are proposed to directly process raw point cloud. Li et al. [20] proposed learnable X-Conv to transform the coordinates of input point cloud. Inverse density was proposed by Wu et al. [10] to solve the problem of uneven distribution of point cloud. Liu et al. [21] proposed RSConv to segment point cloud. And the relationship between coordinates was used to assist the network learn more shape information. Liu et al. [22] fairly compared current local aggregation operations and proposed a simple, learning-free local aggregation approach. Jiang et al. [23] proposed an orientation-encoding unit to describe eight key orientation information. These methods focus on setting up a mechanism to aggregate local features and directly extract them layer by layer through a hierarchical structure. Point-based methods can direct process of point cloud data without much GPU overhead.

Multi-scale feature fusion: Multi-scale feature fusion is widely used in 2D image task. For example, in the task of object detection, Lin et al. [24] proposed a FPN (feature pyramid) structure to improve the prediction results of small targets. Li et al. [25] proposed self-interaction modules to adaptively extract multi-scale features and enhance the robustness of the network. In semantic segmentation, Gu et al. [26] used two parallel encoders to extract information of different scales and merge them. Zhao et al. [27] used global average pooling of different sizes to construct a spatial pyramid to fuse multi-scale features. Chen et al. [28] used dilated convolutions of different sizes to capture multi-scale information. Kamnitsas et al. [29] proposed a dual CNN architecture to process high/low resolution images, and achieved good score in medical image segmentation. The prediction accuracy and generalization ability of the model can be improved by reasonably integrating different scale features. In the point cloud, Huang et al. [4] proposed a feature-based multi-scale network to complete the point cloud task. Li et al. [30] proposed a multi-scale domain feature and aggregation model to enhance the feature extraction ability of the network. Geng et al. [31] proposed a multi-scale attention aggregation network to capture global features from the encoder and decoder. In this work, a novel multi-scale feature fusion is proposed to adaptively connect different layers of the network.

Self-attention: Early self-attention has been applied to the field of natural language processing. Vaswani et al. [32] built the earliest transformer using self-attention mechanism, which is used for machine translation. With the deepening of research, the self-attention mechanism has been applied to the field of computer vision. Dosovitskiy et al. [33] proved that the transformer can be applied to 2D images. Han et al. [34] proposed a novel TNT (transformer in transformer) module to model the relationship between patch and pixel. Swin transformer was proposed by Liu et al. [35] to efficiently complete the object detection task. Fu et al. [36] used the self-attention mechanism to capture the contextual information of channels and locations, and achieved excellent performance in semantic segmentation tasks. Wang et al. [37,38] used self-attention mechanism to capture contextual information. Zhu et al. [39] compressed feature dimensions through pooling operation to reduce the computational amount of self-attention. In 3D point cloud, Engel et al. [40] designed point transformer to extract local and global features. Zhao et al. [8] and Guo et al. [9] modified the self-attention mechanism to better complete the point cloud segmentation. Different from the previous work, our network uses the self-attention mechanism to interactively capture coordinate and feature information.

3. Methodology

In this section, we first show the network structure roughly and describe how they complete classification and segmentation tasks. Then, we introduce how to build a CSA module using the self-attention mechanism. Finally, we describe how to build the multi-scale fusion (MF) module in detail.

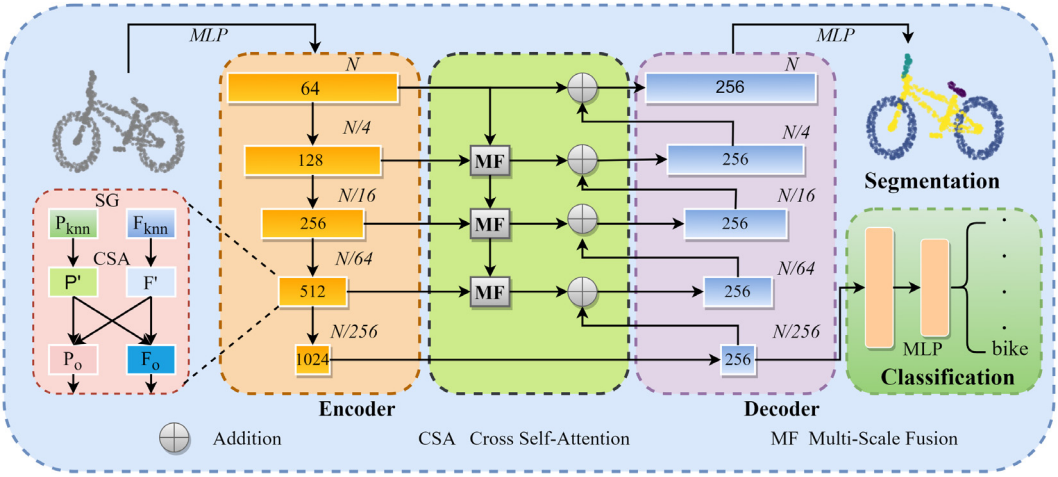

Fig. 1. Network architecture diagram. Where N is the number of points, MLP represents multilayer perceptron, P_o and F_o are the outputs to the CSA module.

3.1. Overview

Given a point cloud collection that contains N points. Each point has 3 coordinates and d-dimensional features (for example, color, normal, etc.), denoted as P and F respectively. Our goal is to reasonably use P and F to extract features from point cloud collections. As illustrated in Fig. 1, a Cross self-attention Network (CSANet) is proposed for 3D point cloud classification and semantic segmentation. CSANet adopts an encoder–decoder structure, which is conducive to full feature fusion. In the encoding part, there are three parts: Projection, SG (sampling and grouping), and CSA (cross self-attention). **Projection:** we send the input features and coordinates to the MLP for projection, and increase the dimension to 64. **SG:** The FPS [2] (farthest point sampling) algorithm is used to down-sample the coordinates and features to determine the center point. And we use the Euclidean distance-based KNN (K-Nearest Neighbor) algorithm to find the neighbors of the center point. **CSA:** The relationship between coordinates and features is captured by the CSA module. The details of CSA module can be found in 3.2. Please note that in the SG process, we add some geometric information to P and F to obtain P' and F', which can be used as the position encoding of the CSA module. They can be described as:

$$P' = cat(P_{knn}, xyz_k - xyz_c, xyz_c) \tag{1}$$

$$F' = cat(F_{knn}, xyz_k - xyz_c, xyz_c) \tag{2}$$

where P_{knn} is the k neighbors found by KNN from the projected coordinates, F_{knn} is the k neighbors found by KNN from the projected features. Herein, xyz_c is the center point calculated by the FPS algorithm through the original coordinates (without projection), and xyz_k is the k neighbors found by KNN from the original coordinates. We concatenate them in the channel dimension and find the largest feature of the neighbor after passing through the convolutional layer. Then send it to the CSA module.

In the decoding part, we up-sample the final features of the encoder and merge them with the output of the MF module (the MF module will be described in Section 3.3). We repeat the above process to obtain the final segmentation result. Please note that the up-sampling operation we used is distance-based interpolation [2]. In classification task, the output of the encoder can also be serially connected to the MLP to get the result of the classification task (e.g. bike, table, chair).

3.2. Cross self-attention

The function of the CSA module is to use the self-attention mechanism to adaptively enhance the coordinates and features information. It receives two inputs, which are the coordinates and features after SG. The concrete structure of CSA module is shown as in Fig. 2.

Given a set of intermediate features (F_{mid}) and coordinates (P_{mid}), which are sampled and grouped. As shown in Fig. 2, we first project P_{mid} and F_{mid} to obtain F_q, F_k, F_v, F_R and P_k, P_q, P_v, P_R. Herein, q, k, v, r, and o represent query, key, value, residual, and output respectively. Then P_o and F_o are obtained by a cross self-attention operation, and this process is defined as follows:

$$P_o = P_v \odot F_A + P_R \tag{3}$$

$$F_o = F_v \odot P_A + F_R \tag{4}$$

where P_R and F_R denote the original residual branches, $\odot$ denotes the element-wise multiplication operation. It can be inferred from Eq. (3) that the missing coordinate information in the feature is explicitly enhanced. F_A and P_A are the weighted sum of the medium features, and they are defined as follows:

$$P_A = Softmax(Sum(P_q \otimes P_k)) \tag{5}$$

$$F_A = Softmax(Sum(F_q \otimes F_k)) \tag{6}$$

where $\otimes$ denotes the matrix multiplication operation and Sum denotes adding each row of the result of matrix multiplication to the first row. Compared to the self-attention mechanism, CSA not only reassigns weights to each element but also reduces the memory overhead of the computer.

3.3. Multi-scale fusion

Different from the previous methods that directly adopt the fusion of the same level layer, MF adopts a multi-scale fusion strategy to fuse features at different levels. The architecture of MF is shown in Fig. 3.

For convenience, we define F_{EL} and P_{EL} as the features and coordinates from the Lth layer encoder, respectively. N_L and D_L are the number of points and the number of dimensions in the Lth layer, respectively. MF_{EL-1} is the output of the previous MF module. F_{DL} is the features from the Lth layer decoder. The whole process can be described as:

$$MF_{out} = MLP(cat(FPS(MF_{EL-1}), F_{EL}, P_{EL})) \odot FA \tag{7}$$

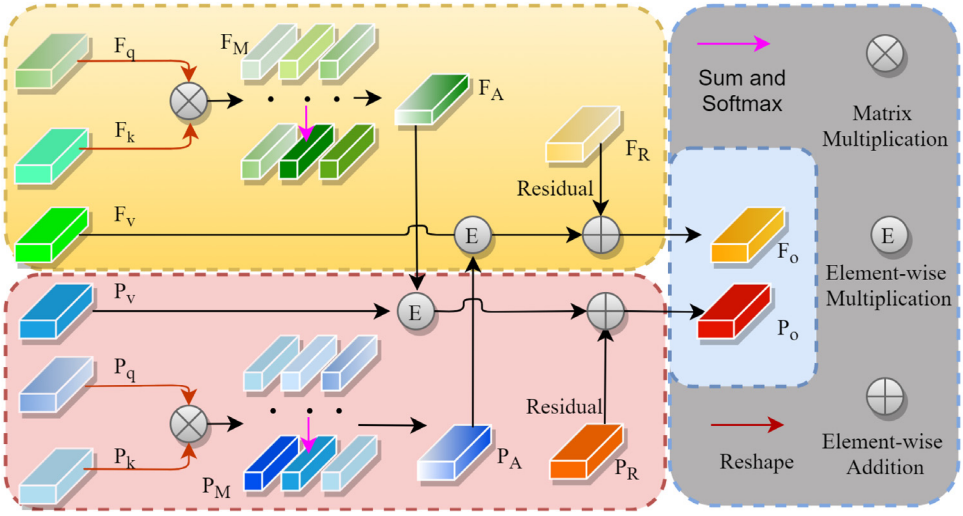

Fig. 2. CSA module architecture diagram.

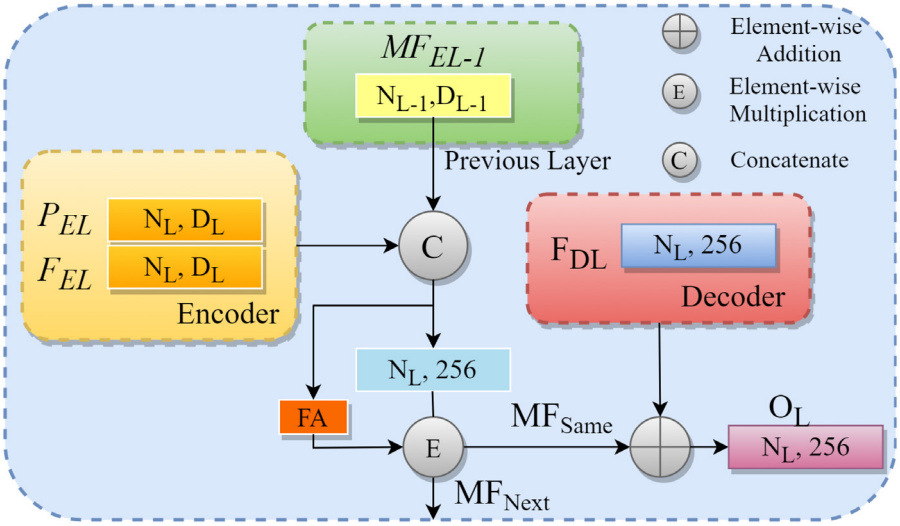

Fig. 3. The details of MF module. MF_{Next} and MF_{Same} have the same value. MF_{Next} indicates that the result of MF in this layer is passed to the next MF module, and MF_{Same} indicates the add operation with the decoder in the same level.

where MLP represents multilayer perceptron, FPS represents farthest point sampling, cat represents concatenate in channel dimension, $\odot$ represents element-wise multiplication and FA represents feature attention. FA is a simple feature attention that can assist networks better select features. It is defined as:

$$FA = \delta(MLP(Sum(MF_{out} \otimes MF^{T}{}_{out}))) \tag{8}$$

where δ represents the sigmoid activation function, T represents the transpose, $\otimes$ represents matrix multiplication and Sum represents the sum of all rows to the first row. It is not difficult to find that FA is a simplified version of the CSA module. O_L represents the fusion result of the current MF module and F_{DL}. It is expressed by:

$$O_L = MLP(MF_{Same} + F_{DL}) \tag{9}$$

The MF module implicitly receive the results of all previous layers as input and deliver the output to the decoder and the next layer at the same time. With the deepening of the networks, the MF module gradually capture features of different sizes and merge them with the decoder.

4. Experiments

In this section, we evaluate the proposed network on three mainstream datasets (ModelNet40, ShapeNetPart, and S3DIS). Besides, a series of ablation experiments are reported to verify the effectiveness of the proposed module, including the choice of the number of neighbors k and the impact of different structures.

4.1. Datasets and experimental settings

Point cloud data is a disordered and redundant collection. The shape of each point cloud data is $N \times d$, where N denotes the number of points and d denotes the feature dimension. Different datasets have different feature dimensions. Next, we introduce each dataset in detail.

ModelNet is a basic point cloud classification dataset containing 40 categories (for example, aircraft, bed, car, etc.) with a total of 12,311 CAD (computer aided design) models. Each point cloud data has 6 features, namely color information R, G, B, and coordinate information x, y, z. In order to make a fair comparison, we follow the official split method, that is, 9843 models are

Table 1
The influence of different k values of the KNN algorithm on ModelNet validation set.

k	mAcc	Accuracy
(8,8,8,8)	89.1	92.2
(16,16,16,16)	**89.9**	**92.8**
(24,24,24,24)	89.5	92.2
(32,32,32,32)	89.8	92.3
(8,16,24,32)	89.7	92

used for training and 2468 models are used for validation. Please note that both coordinates and features are used for training. We use mean accuracy within each category (mAcc) and the overall accuracy (OA) as the evaluation indicators of the model.

ShapeNetPart is a 3D object segmentation dataset, which contains 16 shape categories, 50 component categories, and a total of 16 880 annotated 3D models. Each point cloud data has 6 features, namely color information R, G, B, and coordinate information x, y, z. 14 006 models are used for training and 2874 models are used for test. For evaluation metrics, we report category mIoU (cat.mIoU) and instance mIoU (ins.mIoU).

S3DIS is a large-scale indoor scene segmentation dataset, which contains 13 categories and a total of 271 rooms. Each point cloud data has 9 features, namely color information R, G, B, coordinate information x, y, z and 3 normal vectors. 271 rooms are divided into 6 areas, each room is divided into 1 m $\times$ 1 m blocks. We take 6-fold cross-validation for six areas and use ins.mIoU for evaluation metrics to fully validate the proposed model.

Experimental settings: The implementation of our network is based on the Pytorch framework. Its version is 1.1.0, and the CUDA version is 10.0. We only use a Nvidia GTX 1080TI (11 GB) to complete the experiment. Our code is based on Qi et al. [2] and adopts the same data preprocessing. SGD is used as an optimizer for training, where weight decay and momentum are 0.0001 and 0.9, respectively. The batch sizes of the three datasets are set to 32, 32, and 16 respectively. We adopt the "Poly" strategy to dynamically adjust the learning rate and set the initial learning rate to 0.1, 0.1, and 0.3 respectively.

About the model structure, we set the number of output channels of each layer in the encoder to: [N/256, N/64, N/16, N/4. And the number of output channels of the decoder is all set to 256. where N denotes the number of input points.

4.2. Ablation experiments

The number of neighbors is very important for the prediction of the network. Therefore, we take the original classification network (the encoder part of the proposed model) as the baseline and use the number of neighbors k = 16 as the initial value to explore the impact of different neighbors on the ModelNet validation set. The classification results are shown in Table 1. Where k is a list, each element of k is the number of neighbors in the CSA module. It is not difficult to find that the best classification result can be obtained when k = (16,16,16,16). When k is increased in transition (for example, k = (32,32,32,32)), it cause a decrease in network performance (from 92.8% to 92.3%). A reasonable explanation is that 32 neighbors bring some noise to the network. When the number of neighbors is smaller, such as k = (8,8,8,8), the model cannot obtain sufficient context information, resulting in performance degradation. When k is increased layer by layer (for example, k = (8,16,24,32)), an accuracy rate of 92% is obtained, which is 0.8% lower than k = (16,16,16,16).

We reduce the number of input points in the ModelNet dataset to test out the robustness of the proposed model. For a fair comparison, we set the number of neighbors to 16 for all models, and

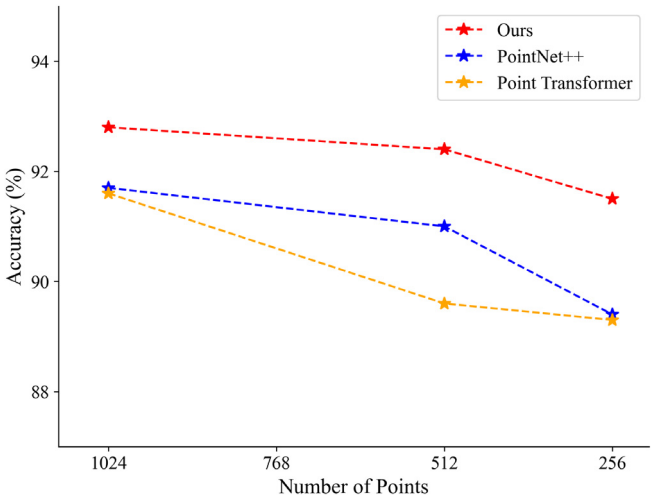

Fig. 4. Classification results for different input points on the ModelNet dataset.

Table 2
The effect of the proposed module on part segmentation.

Model	+Geometric	+MF	cat.mIoU	ins.mIoU
A	×	×	79.8	84
B	✓	×	81.3	84.2
C	✓	✓	82.7	85.2
D	✓	✓	**83.1**	**85.3**

the number of input points is initially set to 1024. Then decreased to 512 and 256 sequentially. Fig. 4 shows the experimental results of different models in the ModelNet validation set with different input points. As the number of input points decreases, the performance of all models decreases to some extent. PointNet++ and PT (Point Transformer) only use coordinate information, and the performance of the model decreases substantially as the number of points decreases. Our model comprehensively considers the coordinate and feature information, so it shows better robustness in the face of decreasing number of points.

As shown in Table 2, the proposed modules are successively added to verify their effectiveness in the ShapeNetPart dataset. Model A is a baseline segmentation network that only uses CSA modules. Model B represents the segmentation network after concatenating the geometric information. Model D means adding the geometric information and the proposed MF module at the same time. Please note that Model C represents that the concatenating operation replace the addition operation when the MF module and the decoder are fused. It is not difficult to find that the CSA module can achieve 84% ins.mIoU and 79.8% cat.mIoU respectively. And simply adding the geometric information can bring an improvement of 0.2% ins.mIoU (from 84% to 84.2%). Adding the proposed MF module can bring 1.1% (from 84.2% to 85.3%) ins.mIoU improvement. In the MF module, we found that the element-wise addition operation obtains better performance than the concatenating operation.

4.3. Comparative experiments and results analysis

ModelNet40: The results of the classification experiment are shown in Table 3. The proposed method achieves the highest mAcc and accuracy. Compared with PT (point transformer) and PCT (point could transformer), which are composed of multiple self-attention layers, the proposed method achieves higher accuracy and F1-score. This phenomenon shows that the CSA module has stronger feature capture ability and robustness than the ordinary SA (self-attention) module.

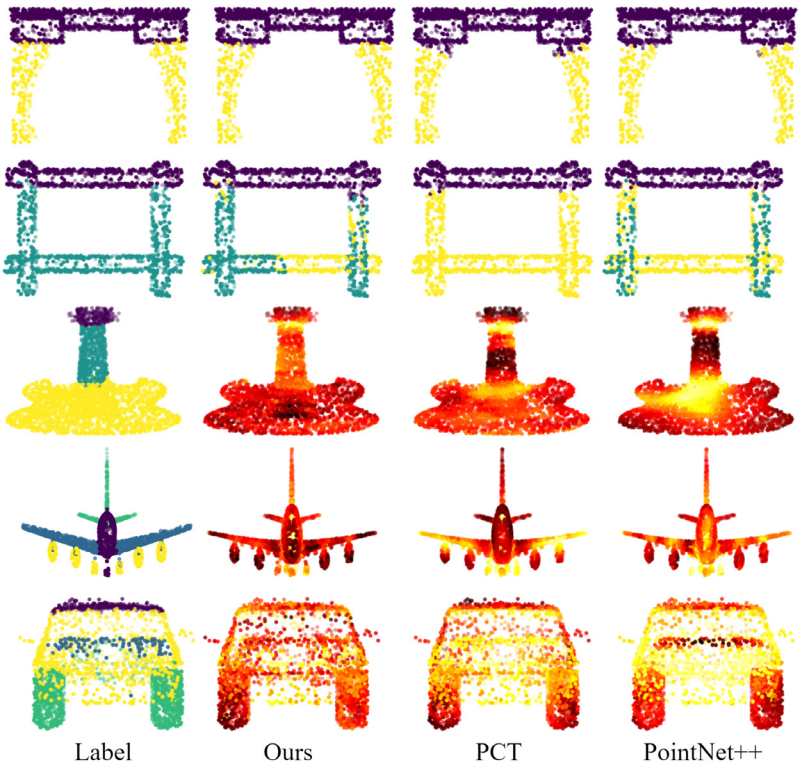

Fig. 5. Partial segmentation results and heat maps of ShapeNetPart. The first and second rows are the segmentation results of categories 'table' and 'chair' by different models. The third, fourth, and fifth rows show the heat maps of the different models for the categories 'guitar', 'aircraft', and 'car'. Each column refers to ground truth and comparison models.

Table 3
Classification results of ModelNet dataset.

Model	Points	mAcc	Accuracy	F1-score
PointNet [1] (2017)	1024	87.7	90.1	85.5
PointNet++ [2] (2017)	1024	89.7	91.7	88.4
PointConv [10] (2017)	1024	89.8	92	88
KPConv [41] (2019)	1024	–	92	88.2
DGCNN [7] (2019)	1024	89.4	92.6	89.5
PPNet [22] (2020)	1024	–	90.8	87.3
PT [8] (2021)	1024	89.4	91.6	88
PCT [9] (2021)	1024	89.4	92.6	88.2
Ours	2048	89.4	92.7	89.8
Ours	1024	**89.9**	**92.8**	**90**

Table 4
ShapeNetPart validation and test set segmentation results.

Model	Point	Weight	$ins.mIoU_{val}$	$ins.mIoU_{test}$	Speed/s
PointNet [1]	2048	95.5 MB	83.3	83.3	0.02
PointNet++ (SSG) [2]	2048	16.2 MB	84.8	85.1	0.11
PointNet++ (MSG) [2]	2048	20 MB	85	85.5	0.79
DGCNN [7]	2048	**11.2 MB**	85.1	85.3	**0.03**
PPNet [22]	2048	211 MB	84.9	84.7	0.93
PCT [9]	2048	18.7 MB	85.1	84.9	0.02
Ours	2048	148 MB	**85.3**	**85.7**	0.11

ShapeNetPart: In order to quantify the effectiveness of the proposed method, we conducted a series of comparative experiments on ShapeNetPart validation set and test set. The experimental results are shown in Table 4. The proposed method achieves the highest validation and test ins.mIoU (85.3%, 85.7%). Compared with PCT, which uses a self-attention mechanism, the proposed method has 0.8% improvement in the ins.mIoU. In terms of model complexity, the proposed method needs to consider coordinates and features at the same time, more learnable parameters are needed. When training 16 images on the same device, the proposed method consumes about the same amount of time as PointNet++ (SSG), but ins.mIoU is boosted by 0.5%. Compared to [7], the proposed method requires more parameters and consumes more extra time.

As shown in Fig. 5, we visualized some of the prediction results and heat maps. The proposed method can better locate through the coordinate information and obtain an excellent segmentation effect. Compared with PointNet++, which focuses on the handle of the guitar, the proposed network can focus on the overall information. In the third row, the proposed network pays more attention to the wing of the aircraft than the PCT, which is an important feature of the aircraft. Car is a relatively complex category in ShapeNetPart. The proposed network can use coordinates and features to focus on the entire car instead of only on the wheels. These can prove that our method can better capture the relationship between pixels and make more reasonable segmentation according to coordinates and features.

S3DIS: Previous experiments have proved the effectiveness of the proposed method on simple tasks. Next, we conduct a 6-fold cross-validation experiment on the S3DIS dataset to evaluate the model more comprehensively. The experimental results are shown in Table 5. where the 'Area 1' column indicates the experimental results using areas 2,3,4,5,6 as the training samples and area 1 as the test sample. The last column of Table 5 indicates the mean ins.mIoU values for the 6-fold experiment. It is easy to see from the table that the proposed method achieved the highest mean ins.mIoU in the 6-fold cross-validation experiment. Compared with PointNet++ (MSG), which aggregates features of different scales, the proposed method gets higher scores (an increase of 1.8 percentage points). The graph neural networks DGCNN and GACNet achieve 55.5% and 44.1%, which are 1.7% and 13.1% lower than the proposed method.

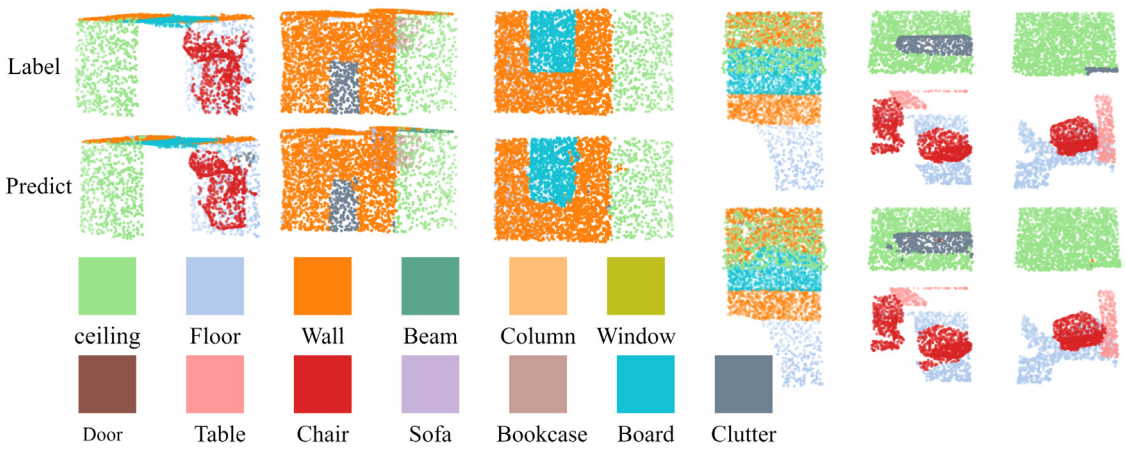

Fig. 6. Partial segmentation results in S3DIS dataset. The first row is the ground truth and the second row is the segmentation result of the proposed model.

Table 5
Results of 6-fold cross-validation experiment on the S3DIS dataset.

Model	Area 1	Area 2	Area 3	Area 4	Area 5	Area 6	Mean
PointNet [1]	50	35.3	52.9	44.5	46	61.6	48.4
GACNet [42]	43.6	33.4	49.2	40.1	48.1	50.4	44.1
PointNet++ (SSG) [2]	59.8	37	68.3	44.7	52.3	68.3	55
PointNet++ (MSG) [2]	61.2	37.5	68	45	52.5	68	55.4
DGCNN [7]	64.1	38	63.8	**47.5**	51.1	68.6	55.5
Ours	**68.7**	**40.4**	**63.9**	46.3	**53.3**	**70.3**	**57.2**

As shown in Fig. 6, partial prediction results of area 5 are visualized by us. Like training, the prediction results are also divided into 1 m $\times$ 1 m blocks. The proposed network can perceive contextual information and make accurate segmentation in each block. These experiments prove that the proposed method is very competitive on large point cloud datasets.

5. Conclusion

In this paper, we proposed a novel CSANet for 3D point cloud classification and segmentation, which adaptively integrates coordinates and features information through an improved self-attentive mechanism. Specifically, we introduce a CSA (cross self-attention) module and a MF (multi-scale fusion) module. The CSA module can assist the network to fully extract features during the encoding process, and greatly improves model robustness. In the process of decoding, the MF module helps the model to fully integrate the semantic information between different layers and scales. Compare with other self-attention-based methods, CSANet achieves better performance on shape classification, part segmentation and large scale semantic segmentation tasks.

In future research, we will study more efficient fusion methods of features and coordinates information, and dig deeply into the potential of self-attention mechanism on the point cloud.

Support information: Our model can be found at the following link:
https://github.com/Qianyu1998/CSANet.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] C.R. Qi, H. Su, K. Mo, L.J. Guibas, 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.
[2] 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. 30 (2017).
[3] Z. Du, H. Ye, F. Cao, 3D mixed CNNs with edge-point feature learning, Knowl.-Based Syst. 221 (2021) 106985.
[4] Z. Huang, Y. Yu, J. Xu, F. Ni, X. Le, Pf-net: Point fractal network for 3d point cloud completion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 7662–7670.
[5] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, A. Markham, Randla-net: Efficient semantic segmentation of large-scale point clouds, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11108–11117.
[6] S. Qiu, S. Anwar, N. Barnes, Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1757–1767.
[7] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, J.M. Solomon, Dynamic graph cnn for learning on point clouds, Acm Trans. Graph. (Tog) 38 (5) (2019) 1–12.
[8] H. Zhao, L. Jiang, J. Jia, P.H. Torr, V. Koltun, Point transformer, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 16259–16268.
[9] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R.R. Martin, S.-M. Hu, PCT: Point cloud transformer, Comput. Visual Media 7 (2) (2021) 187–199.
[10] W. Wu, Z. Qi, L. Fuxin, Pointconv: Deep convolutional networks on 3d point clouds, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9621–9630.
[11] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3d shapenets: A deep representation for volumetric shapes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1912–1920.
[12] L. Yi, V.G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, L. Guibas, A scalable active framework for region annotation in 3d shape collections, ACM Trans. Graph. (ToG) 35 (6) (2016) 1–12.
[13] I. Armeni, O. Sener, A.R. Zamir, H. Jiang, I. Brilakis, M. Fischer, S. Savarese, 3d semantic parsing of large-scale indoor spaces, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1534–1543.
[14] B. Wu, A. Wan, X. Yue, K. Keutzer, Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 1887–1893.
[15] F.N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally, K. Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size, 2016, arXiv preprint arXiv:1602.07360.
[16] X. Chen, H. Ma, J. Wan, B. Li, T. Xia, Multi-view 3d object detection network for autonomous driving, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1907–1915.
[17] 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), IEEE, 2015, pp. 922–928.

[18] O. Çiçek, A. Abdulkadir, S.S. Lienkamp, T. Brox, O. Ronneberger, 3D U-Net: learning dense volumetric segmentation from sparse annotation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2016, pp. 424–432.
[19] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241.
[20] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, B. Chen, Pointcnn: Convolution on x-transformed points, Adv. Neural Inf. Process. Syst. 31 (2018) 820–830.
[21] Y. Liu, B. Fan, S. Xiang, C. Pan, Relation-shape convolutional neural network for point cloud analysis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8895–8904.
[22] Z. Liu, H. Hu, Y. Cao, Z. Zhang, X. Tong, A closer look at local aggregation operators in point cloud analysis, in: European Conference on Computer Vision, Springer, 2020, pp. 326–342.
[23] M. Jiang, Y. Wu, T. Zhao, Z. Zhao, C. Lu, Pointsift: A sift-like network module for 3d point cloud semantic segmentation, 2018, arXiv preprint arXiv:1807.00652.
[24] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.
[25] S. Li, X. Chen, Y. Liu, D. Dai, C. Stachniss, J. Gall, Multi-scale interaction for real-time lidar data segmentation on an embedded platform, IEEE Robot. Autom. Lett. (2021).
[26] F. Gu, N. Burlutskiy, M. Andersson, L.K. Wilén, Multi-resolution networks for semantic segmentation in whole slide images, in: Computational Pathology and Ophthalmic Medical Image Analysis, Springer, 2018, pp. 11–18.
[27] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2881–2890.
[28] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 801–818.
[29] K. Kamnitsas, C. Ledig, V.F. Newcombe, J.P. Simpson, A.D. Kane, D.K. Menon, D. Rueckert, B. Glocker, Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation, Med. Image Anal. 36 (2017) 61–78.
[30] D. Li, G. Shi, Y. Wu, Y. Yang, M. Zhao, Multi-scale neighborhood feature extraction and aggregation for point cloud segmentation, IEEE Trans. Circuits Syst. Video Technol. 31 (6) (2020) 2175–2191.
[31] X. Geng, S. Ji, M. Lu, L. Zhao, Multi-scale attentive aggregation for LiDAR point cloud segmentation, Remote Sens. 13 (4) (2021) 691.
[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
[33] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, 2020, arXiv preprint arXiv:2010.11929.
[34] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, Y. Wang, Transformer in transformer, 2021, arXiv preprint arXiv:2103.00112.
[35] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, 2021, arXiv preprint arXiv:2103.14030.
[36] J. Fu, J. Liu, J. Jiang, Y. Li, Y. Bao, H. Lu, Scene segmentation with dual relation-aware attention network, IEEE Trans. Neural Netw. Learn. Syst. (2020).
[37] G. Wang, Q. Zhai, Feature fusion network based on strip pooling, Sci. Rep. 11 (1) (2021) 1–8.
[38] G. Wang, Q. Zhai, J. Lin, Multi-scale network for remote sensing segmentation, IET Image Process (2022) 1–10.
[39] Z. Zhu, M. Xu, S. Bai, T. Huang, X. Bai, Asymmetric non-local neural networks for semantic segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 593–602.
[40] N. Engel, V. Belagiannis, K. Dietmayer, Point transformer, IEEE Access 9 (2021) 134826–134840.
[41] H. Thomas, C.R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, L.J. Guibas, Kpconv: Flexible and deformable convolution for point clouds, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6411–6420.
[42] L. Wang, Y. Huang, Y. Hou, S. Zhang, J. Shan, Graph attention convolution for point cloud semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10296–10305.