

Lightweight Attention Module for Deep Learning on Classification and Segmentation of 3-D Point Clouds

Yunhao Cui^{ID}, Yi An^{ID}, *Member, IEEE*, Wei Sun^{ID}, Huosheng Hu^{ID}, *Senior Member, IEEE*, and Xueguan Song^{ID}

Abstract—Research on classification and segmentation of 3-D point clouds using deep learning methods has become a hot topic in emerging applications, such as autonomous driving, augmented reality, and indoor navigation. However, as the complexity of the network structures increases, the computational efficiency reduces, which affects the practical applications of these methods. In addition, prior researchers mostly seek to enhance the quality of spatial encodings, while the channel relationships are ignored. It makes the feature learning of point clouds insufficient, which will reduce the accuracy of classification and segmentation. In this article, a lightweight attention module (LAM) is proposed to improve the computational efficiency and accuracy at the same time by adopting a novel convolution mode and introducing a new attention mechanism based on channelwise statistical features. As the submodules of LAM, the lightweight module and the attention module can also be used independently to focus on improving the computational efficiency and accuracy, respectively, according to the actual applications. LAM and its submodules can be easily integrated into state-of-the-art deep learning methods on classification and segmentation of 3-D point clouds. The experimental results show that the proposed modules have a good performance on benchmark data sets.

Index Terms—Attention mechanism, classification, lightweight, point clouds, segmentation.

I. INTRODUCTION

A 3-D point cloud is a set of discrete points on the surfaces of target objects collected by laser rangefinders [1], [2]. With the rapid development of the 3-D laser ranging technology, 3-D point clouds have been widely used in emerging applications, such as autonomous driving, augmented reality, and intelligent robots [3]–[5]. Classification and segmentation are critical techniques for processing of 3-D point clouds. Through these operations, point clouds are classified into different sets. The same set has similar or identical attributes.

Manuscript received March 7, 2020; accepted July 24, 2020. Date of publication July 30, 2020; date of current version November 20, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61673083 and in part by the Science and Technology Major Project of Shanxi Province under Grant 20191102008. The Associate Editor coordinating the review process was Anirban Mukherjee. (*Corresponding author: Yi An.*)

Yunhao Cui, Wei Sun, and Xueguan Song are with the School of Mechanical Engineering, Dalian University of Technology, Dalian 116023, China (e-mail: cyhlovegl@mail.dlut.edu.cn; sunwei@dlut.edu.cn; sxx@dlut.edu.cn).

Yi An is with the School of Control Science and Engineering, Dalian University of Technology, Dalian 116023, China (e-mail: anyi@dlut.edu.cn).

Huosheng Hu is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: hhu@essex.ac.uk).

Digital Object Identifier 10.1109/TIM.2020.3013081

Classification and segmentation of point clouds are the basis of some important processing tasks, such as the detection of 3-D objects and the reconstruction and understanding of 3-D scenes.

In recent years, some researchers have devoted themselves to developing methods for classification and segmentation of 3-D point clouds. Traditional methods for classification and segmentation of 3-D point clouds use handcrafted features to capture geometric attributes [6]. In this way, the performance of feature learning is largely affected by the handcrafted features [7]. More recently, many investigations borrow concepts from the convolutional neural network (CNN) to deal with point clouds motivated by the success of CNN in image processing. This kind of method develops rapidly and outperforms traditional methods in various tasks for classification and segmentation of 3-D point clouds [8]. However, due to the irregularity of point clouds, most of these methods cannot directly deal with them. It needs to convert point clouds into other regular representations since the standard CNN requires input data with a regular structure. One common approach is to project 3-D point clouds onto a 2-D plane [9]–[11], which will lose certain 3-D inherent geometric information. Another common approach is to convert 3-D point clouds into volumetric grids [12]–[14], which will introduce quantization artifacts and reduce computational efficiency. These transformations usually lead to difficulties in extracting fine-grained features.

State-of-the-art architectures are designed to handle the irregular point clouds directly. This kind of method was pioneered by PointNet [15], which implements the permutation invariance of point clouds by independently performing feature learning on each point and then applying max pooling to accumulate features. Although PointNet is more accurate and robust than previous methods, it ignores local features, which limits its fine-grained pattern recognition and perception of complex scenes. To solve this problem, various extensions take measures to make use of local features through considering neighborhoods of each point [7], [16], [17]. Compared with PointNet, these improved algorithms have achieved some improvements in performance.

However, with the increasing complexity of network structures, computational efficiency decreases, which brings obstacles to the practical application of these methods. In addition, these extensions mostly seek to enhance the quality of spatial encodings, the relationships between different feature channels are ignored. It makes the feature learning of point clouds

insufficient, which will affect the accuracy of classification and segmentation. There are a relatively few studies on these problems for 3-D point clouds. However, researchers have made some progress in improving the computational efficiency and accuracy for image processing [18]–[21]. It has certain reference significance for solving related problems in 3-D point clouds.

To deal with these problems, we propose a lightweight attention module (LAM) that is obtained by organically integrating two submodules: lightweight module (LM) and attention module (AM). LM improves computational efficiency by adopting depthwise separable convolution (DSCConv). AM improves the accuracy by automatically rescaling the weights of different feature channels based on channelwise statistical features. LAM can improve the computational efficiency and accuracy at the same time. These modules are easy to be implemented and can be integrated into state-of-the-art deep learning architectures for classification and segmentation of point clouds. In the experiments, we integrate LAM into PointNet and PointNet++ [16]. The improved networks perform more accurately and efficiently on different data sets, such as most notably ModelNet40 [13] for classification, ShapeNet [22] for part segmentation, and the Stanford large-scale 3-D indoor space data set [23] for indoor scene segmentation.

The key contributions of this article are summarized as follows.

1) LAM is proposed to improve the computational efficiency and accuracy simultaneously by adopting a novel convolution mode and introducing a new attention mechanism based on channelwise statistical features.

2) As the submodules of LAM, LM and AM can be used independently to focus on improving the computational efficiency and accuracy, respectively.

3) The proposed modules have good portability and help to improve the performance of the original deep learning methods.

The remainder of this article is organized as follows. Section II reviews the related works on classification and segmentation of 3-D point clouds. Section III describes the proposed module LAM and the improved backbone architectures based on LAM. Section IV shows the experimental results and discussions. The conclusions are drawn in Section V.

II. RELATED WORKS

As critical technologies for processing 3-D point clouds, many classification and segmentation methods have been proposed over recent years [24]. These methods can be classified into three categories: traditional methods based on hand-designed features, deep learning methods based on regular representations of point clouds, and deep learning methods based on irregular point clouds.

Traditional methods based on hand-designed features artificially design feature descriptors according to different problems and then use machine learning methods to complete classification and segmentation of point clouds. Many previous studies have proposed a variety of different local feature descriptors for point clouds to handle different

problems [25], [26]. Common descriptors of point clouds can be divided into two categories: statistical feature descriptors and geometrical feature descriptors. Representative statistical feature descriptors are fast point feature histograms [27], direction histograms [28], normal histograms [29], and inner-distance descriptors [30]. Representative geometrical feature descriptors are spin images [31], local surface patches [32], and intrinsic shapes [33]. Traditional methods based on hand-designed features usually have a high degree of human dependence and are often poorly portable. For some challenging tasks, it is very difficult to find the optimal features with human experience. Some new feature descriptors have been proposed, which play a very important role in extracting features of point clouds. In [34], a new local feature matching method was proposed based on the combination of geometrical and spatial information. In [35], the cross features have been used to combine the correlation between features to avoid overfitting problems.

In recent years, deep learning has made breakthroughs in many fields, including classification and segmentation of 3-D point clouds. Deep learning methods based on regular representations of point clouds focus on converting irregular point clouds to regular representations. Some researchers apply 3-D CNNs on the volumetric grid [13], [14], [36], which makes computational efficiency reduce exponentially with resolution because of the data sparsity generated by this data representation. This is very challenging for the classification and segmentation of point clouds in large scenes. Studies [37], [38] have made some progress in solving the sparsity problem, but it has not been completely solved. Some researchers [11], [39], [40] adopt the multiview method to render 3-D point clouds into multiple 2-D images. This method can make full use of 2-D CNNs' successful experience in image processing. However, point clouds will lose depth information in the process of projection from 3-D to 2-D, which is not conducive to classification and segmentation of 3-D point clouds.

Deep learning methods based on irregular point clouds, pioneered by PointNet, make it possible to use CNNs directly to handle raw irregular point clouds. Experiments show that PointNet is an effective method to solve the irregularity of point clouds. It is also an effective way to learn features directly from point clouds and retain feature information to the greatest extent. PointNet++ [16], as an extension of PointNet, makes up for the shortcoming of PointNet's inability to extract local features. It leverages proximity to obtain local information on multiple scales for better performance. Dynamic graph convolution neural network (DGCNN) [7] utilizes local feature information by constructing a local neighborhood graph and applying dynamic graph convolution on the edges connecting adjacent point pairs.

Although many alternatives to PointNet have been proposed for higher performance, the simplicity and effectiveness of PointNet and its extension PointNet++ make them popular for classification and segmentation of 3-D point clouds. In this article, PointNet and PointNet++ are used as the backbone architectures for evaluating the effect of LAM and its submodules.

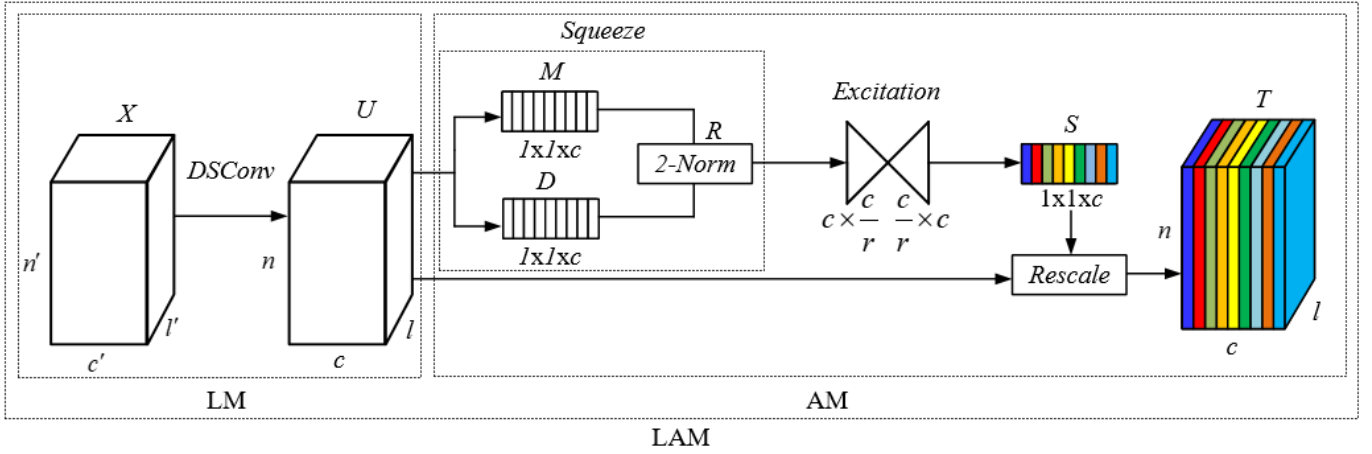


Fig. 1. Schema of LAM: Inside the dotted frame on the left is the submodule of LM, and inside the dotted frame on the right is the submodule of AM.

III. METHODOLOGY

In this section, we show the problem statement in Section III-A, propose LAM in Section III-B, and finally improve PointNet and PointNet++ based on LAM and its submodules in Section III-C.

A. Problem Statement

As a pioneer of deep learning methods based on irregular point clouds, PointNet makes it possible to directly process an irregular 3-D point cloud, which is represented as $P = \{p_i = (x_i, y_i, z_i) | 1 \leq i \leq n\}$. Extra features can also be added, such as normal vectors and colors. Some subsequent improvements, such as PointNet++ and DGCNN, have further enhanced the ability to extract local features. These methods have made significant breakthroughs on classification and segmentation of point clouds. Not only do they avoid reliance on hand-designed features, but they can also take full advantage of the information in point clouds. However, the computational efficiency of these methods decreases with the increasing complexity of network structures. It brings obstacles to the practical application of these methods. In addition, these extensions mainly seek to enhance the quality of spatial encodings while ignoring channel relationships. This makes it impossible to distinguish the importance of different feature channels, which affects the accuracy of classification and segmentation. The specific causes and solutions for these problems are as follows.

- 1) Computational efficiency decreases as network complexity increases. These algorithms use regular convolution to map simultaneously cross-channel correlations and spatial correlations with a single convolution kernel. Through this operation, the cross-channel correlations and the spatial correlations are fused, which are in a decoupled state. This will affect the sufficiency of feature extraction and reduce the computational efficiency. Changing the convolution mode to make the network more lightweight is an effective way to solve this problem.

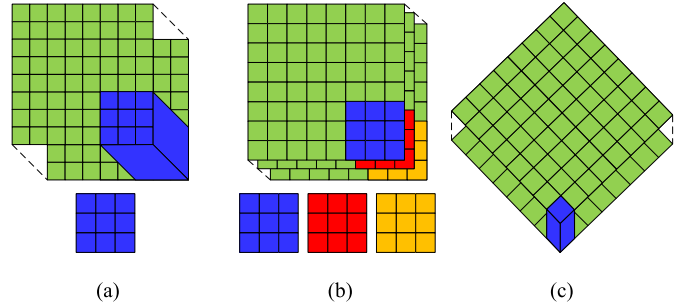


Fig. 2. DSCConv and regular convolution. (a) Regular convolution. (b) Depth-wise convolution of DSCConv. (c) Pointwise convolution of DSCConv.

- 2) Insufficient feature extraction affects the accuracy on classification and segmentation of 3-D point clouds. Traditional CNN cannot explicitly model the relationships between different feature channels. This makes it impossible to distinguish the importance of different feature channels, which will affect the sufficiency of feature extraction and reduce accuracy. Introducing an attention mechanism to automatically rescale the weights of different feature channels is essential to screen out the information that is more critical to the current task and improve the accuracy on classification and segmentation of 3-D point clouds.

B. Lightweight Attention Module

1) *Framework*: LAM is designed to improve the computational efficiency and accuracy on classification and segmentation of 3-D point clouds. As shown in Fig. 1, it is obtained by integrating two submodules: LM and AM. LAM consists of four steps: DSCConv, squeeze, excitation, and rescale.

a) *DSCConv*: In order to improve computational efficiency, the DSCConv is adopted to deal with the cross-channel correlations and spatial correlations. The spatial convolution is performed independently on each channel and followed by a pointwise convolution, as shown in Fig. 2. It conduces to make the network more lightweight by reducing the amount

of parameters. Given an input $X \in \mathbb{R}^{n' \times l' \times c'}$, the intermediate features $U \in \mathbb{R}^{n \times l \times c}$ can be obtained by

$$U = \text{DSConv}(X). \quad (1)$$

Subsequently, batch normalization is performed to increase the stability of the learning process by regulating the distribution of the intermediate features U .

b) Squeeze: In order to make better use of global spatial information, the intermediate features $U = \{U_k | 1 \leq k \leq c\}$ are squeezed into channel descriptors, where k indicates the sequence number of channels. First, two important features of mean and standard deviation are calculated separately on each feature channel. Then, the 2-norm of these two features is calculated to get channelwise statistical features. $M = \{\mu_k | 1 \leq k \leq c\}$ represents the mean, where μ_k is calculated by

$$\mu_k = \frac{1}{h \times w} \sum_{i=1}^h \sum_{j=1}^w U_k(i, j) \quad (2)$$

where $h \times w$ represents the spatial dimensions of the intermediate features. $D = \{\sigma_k | 1 \leq k \leq c\}$ represents the standard deviation, where σ_k is calculated by

$$\sigma_k = \text{sqrt} \left(\frac{1}{h \times w} \sum_{i=1}^h \sum_{j=1}^w (U_k(i, j) - \mu_k)^2 \right). \quad (3)$$

$R = \{r_k | 1 \leq k \leq c\}$ represents the 2-norm of the two kinds of features, where r_k is calculated by

$$r_k = \text{sqrt}(\mu_k^2 + \sigma_k^2). \quad (4)$$

Through the operation of squeeze, the multidimensional features on each feature channel are compressed into a real number. This real number has a global receptive field and can effectively reflect the mean and change range of all features on the feature channel.

c) Excitation: In order to make full use of the information gathered in the operation of squeeze, the operation of excitation is adopted to fully capture the weights of different feature channels. It uses the gating mechanism with two fully connected (FC) layers. This helps to better fit the complex relationships between different feature channels by increasing nonlinearity and can greatly reduce the amount of parameters. ReLU δ is used as the activation function after the first FC layer. It contributes to alleviating the vanishing gradient problem and improving the speed of convergence. Sigmoid β is adopted as the activation function after the second FC layer. It is used to get normalized weights between 0 and 1. The recalibration parameters $S \in \mathbb{R}^{1 \times 1 \times c}$ that represent the importance of each feature channel are calculated by

$$S = \beta(W_2 \delta(W_1 R)) \quad (5)$$

where $W_1 \in \mathbb{R}^{c \times r}$ is the weight of the first FC layer, $W_2 \in \mathbb{R}^{r \times c}$ is the weight of the second FC layer, and r represents the reduction ratio of the bottleneck formed by two FC layers.

d) Rescale: The final output of the LAM module $T = \{T_k | 1 \leq k \leq c\}$ is calculated by rescaling the intermediate features with the recalibration parameters $S = \{s_k | 1 \leq k \leq c\}$

$$T_k = s_k U_k. \quad (6)$$

Through this operation, different feature channels are given different weights according to the role of them.

2) Basic Idea: LAM can improve computational efficiency and accuracy on classification and segmentation of point clouds. It mainly includes two core innovations: lightweight and attention mechanism.

a) Lightweight: Regular convolution adopted by the previous methods maps cross-channel correlations and spatial correlations at the same time using a single convolution kernel. This is shown in Fig. 2(a). The cross-channel correlations and the spatial correlations are fused through this operation. This will reduce the computational efficiency. To solve this problem, DSConv is adopted to deal with the cross-channel correlations and spatial correlations in LAM. DSConv divides the convolution operation into two steps: a depthwise convolution and a pointwise convolution. The depthwise convolution carries out spatial convolution on each channel of the input, as shown in Fig. 2(b). The pointwise convolution maps the channels' output by the previous step onto a new channel space using a 1×1 convolution, as shown in Fig. 2(c). The number of parameters is an important indicator of computational efficiency. Assume that the depth of input feature is d_i , the size of convolution kernel is $(c_k, c_{k'})$, and the depth of output feature is d_o . The number n_r of weights for regular convolution is calculated by

$$n_r = c_k \times c_{k'} \times d_i \times d_o. \quad (7)$$

The number n_d of weights for DSConv is calculated by

$$n_d = c_k \times c_{k'} \times 1 \times d_i + 1 \times 1 \times d_i \times d_o. \quad (8)$$

By comparing the two formulas, it can be seen that DSConv can effectively reduce the number of parameters. This plays an important role in improving computational efficiency.

Through adopting DSConv, the mutual interference of feature extraction in two different dimensions of space and channel can be reduced. Different convolution kernels can be selected for the convolution in these two dimensions according to different characteristic properties. What is more, the computational efficiency on classification and segmentation of point clouds can be effectively improved.

b) Attention mechanism: Traditional CNN adopted in previous algorithms on classification and segmentation of point clouds cannot effectively distinguish the importance of different feature channels, which will affect the adequacy of feature extraction and reduce accuracy. To solve this problem, a new attention mechanism is proposed in this article. The attention mechanism is designed to explicitly model the relationship between different feature channels. In the squeeze step, the global spatial information is squeezed into channel descriptors R through calculating channelwise statistical features. These channel descriptors are then excited with two FC layers to get the weights of different feature channels. Finally, the weights S are used to rescale the input features.

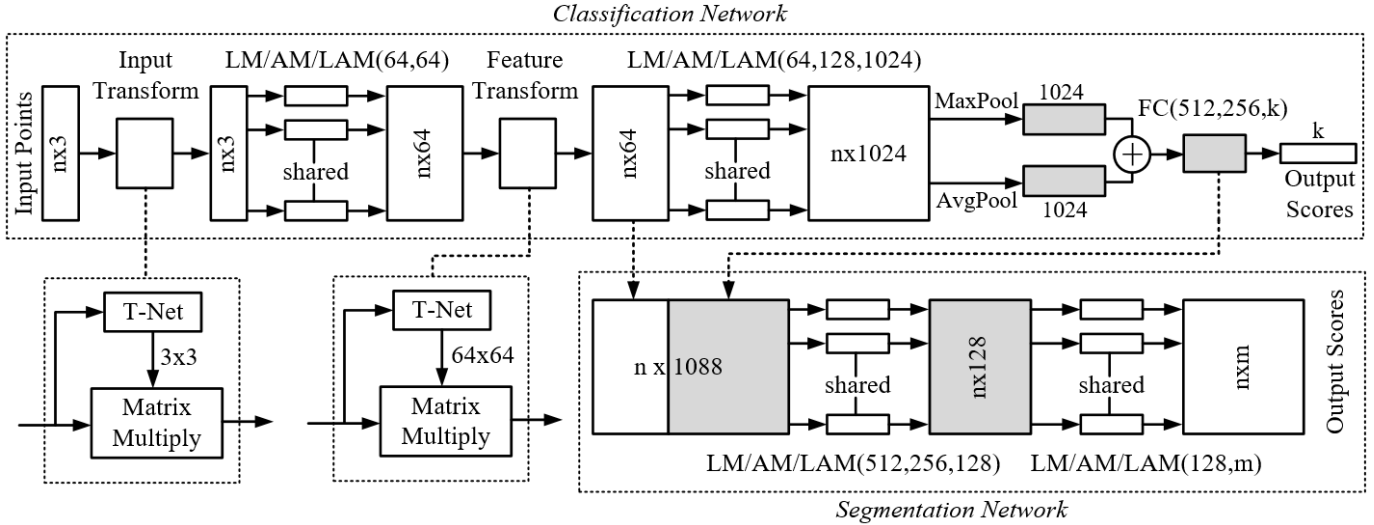


Fig. 3. Improved PointNet architecture on classification and segmentation of point clouds using LAM and its submodules. MaxPool represents global max pooling. AvgPool represents the global average pooling. We modify it on the PointNet framework [15].

By introducing the attention mechanism, the weights of different feature channels are learned adaptively according to the loss. It gives higher weights to valid features and lower weights to invalid ones. The limited computing resources can be used to screen out the information that is more critical to the current task. It helps to improve the adequacy of feature expressions, thereby improving the accuracy on classification and segmentation of point clouds.

C. Improved PointNet and PointNet++

1) *Review of PointNet and PointNet++*: PointNet is a pioneering deep learning network that can directly classify and segment the irregular point clouds. The basic idea of PointNet is to learn and extract the important features of each point through regular convolution and then fuse the information to generate global features using global max pooling. For a raw irregular 3-D point cloud that is represented as $P = \{p_i = (x_i, y_i, z_i) | 1 \leq i \leq n\}$, PointNet can learn a function that maps the point cloud to a vector

$$f(P) = \gamma(\text{MAX}\{h(p_i)\}) \quad (9)$$

where γ represents the FC network, MAX represents the global max pooling, and h represents the regular convolution.

However, PointNet cannot exploit local features that are important for extracting fine features of 3-D point clouds. To solve this problem, PointNet++ constructs a hierarchical neural network, which is composed of many set abstraction levels. At each level, PointNet++ will first conduct sampling and grouping on a point cloud and then extract local features in each small region using PointNet. These local features are grouped into larger units and used as part of the input to extract more advanced features at the higher level. However, these methods have a common problem that they only pay attention to the features expression in spatial dimension while ignoring the channel dimension.

2) *Improved PointNet and PointNet++*: As shown in Fig. 3, PointNet is improved by integrating LAM and its submodules: LM and AM, known as LAM-PointNet, LM-PointNet, and AM-PointNet. For an irregular 3-D point cloud $P = \{p_i = (x_i, y_i, z_i) | 1 \leq i \leq n\}$, the improved PointNet can learn a function that maps the point cloud to a vector

$$\tilde{f}(P) = \gamma((\text{MAX} + \text{AVG})\{\text{LM}(p_i)\}) \quad (10)$$

$$\tilde{f}(P) = \gamma((\text{MAX} + \text{AVG})\{\text{AM}(p_i)\}) \quad (11)$$

$$\tilde{f}(P) = \gamma((\text{MAX} + \text{AVG})\{\text{LAM}(p_i)\}) \quad (12)$$

where γ refers to the FC network and MAX + AVG refers to the sum of output features using global max pooling and average pooling.

Compared to PointNet, the improved methods have the following two main improvements.

- 1) LAM and its submodules are adopted for feature extraction of point clouds. LM can improve the computational efficiency by adopting DSConv. AM can improve the accuracy by introducing a new attention mechanism with channelwise statistical features. LAM improves the computational efficiency and accuracy simultaneously by organically integrating LM and AM.
- 2) Global features are generated by aggregating information of all points through global max pooling and average pooling. Through this operation, the influence of salient features and the retention of feature ranges can be considered at the same time.

To improve PointNet++, LM-PointNet, AM-PointNet, and LAM-PointNet are adopted in place of PointNet to extract local features in each small region which is obtained by sampling and grouping at each set abstraction level.

IV. EXPERIMENTS AND ANALYSIS

In order to verify the effectiveness of LAM and its submodules, we use PointNet and PointNet++ as the backbone architectures and conduct comparative experiments before and

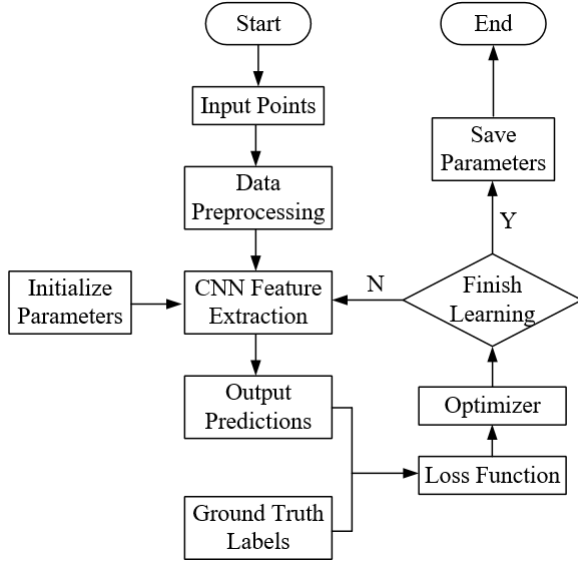


Fig. 4. Training algorithm of classification and segmentation networks.

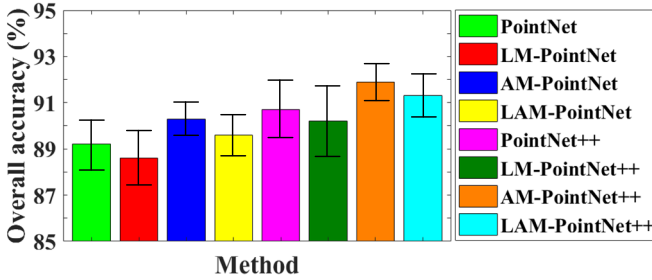


Fig. 5. Error bars of the overall accuracy for different architectures.

after using the proposed modules for improvement. Experiments on classification and segmentation of point clouds are carried out for each network. We choose the ModelNet40 data set in classification experiments, the ShapeNet data set in part segmentation experiments, and the Stanford Large-Scale 3-D Indoor Space data set in scene segmentation. These data sets are very typical in the field of point cloud processing. They are widely used by many classic algorithms of point cloud classification and segmentation. They can be used to fully verify the performance of point cloud classification and segmentation algorithms. In addition, these data sets are open source, easy to obtain, and easy to use. In the experiments, the cross-entropy loss function is adopted as

$$H_{y'}(y) = - \sum_i y'_i \log(y_i) \quad (13)$$

where $H_{y'}(y)$ represents the loss, y'_i represents the i th value in the labels, and y_i is the corresponding component in the predicted values output after normalization by softmax. The training algorithm is shown in Fig. 4. The experimental results are shown and analyzed. The key details are visualized. Experiments are conducted in Python3.5. The deep learning framework is TensorFlow1.8, and the GPU is Tesla P100.

A. Classification

1) *Data Set*: The ModelNet40 data set is used to evaluate our module on classification of point clouds. ModelNet40

TABLE I
ABLATION ANALYSIS

Method	Overall accuracy (%)	Time (s)	Parameters (M)	FLOPs (G)
PointNet	89.2	13.0	3.52	0.93
LM-PointNet	88.6	9.6	3.14	0.80
AM-PointNet	90.3	13.7	3.82	1.05
LAM-PointNet	89.6	10.5	3.45	0.85
PointNet++	90.7	15.3	1.75	2.97
LM-PointNet++	90.2	11.0	1.40	2.69
AM-PointNet++	91.9	16.5	1.97	3.06
LAM-PointNet++	91.3	13.2	1.68	2.87

contains 12 311 CAD models from 40 categories of artificial objects. These models are split into two parts: 9843 for training and 2468 for testing. The point cloud of each model is formed by uniformly sampling 1024 points from the grid surface of the model and normalizing them into a unit sphere. In the preprocessing stage, the same method as PointNet is adopted for data enhancement, including the random rotation of point clouds and jitter of point positions.

2) *Training*: The momentum optimizer is used with an initial learning rate of 0.001. The epoch number is 250. The learning rate decreases with the increase in training epoch number until it reaches 0.00001. The decay step is 200 000 and the decay rate is 0.8. The batch size is 32 and the momentum is 0.9. Other parameter settings are the same as PointNet [15] in the experiments of classification.

3) *Results*: In order to better verify the experimental results of classification, we select reasonable performance metrics. Overall accuracy represents the ratio between the number of correctly classified point clouds and the total number of point clouds on all the test sets. It can effectively reflect the accuracy of the models. Time represents the forward pass time of all the test sets and it can well reflect the computational efficiency of the models. Parameters represent the number of weights and bias of the models. FLOPs represent the floating-point operations. Parameters and FLOPs can well reflect the complexity of the models.

a) *Ablation analysis*: In order to show the performance of LAM and illustrate the contributions of each submodule, the ablation analysis is carried out in Table I. As can be observed, the proposed LM can effectively improve the computational efficiency and reduce the model complexity, but it reduces the accuracy. The proposed AM can effectively improve the accuracy, but it will increase the model complexity and reduce the computational efficiency. After the organic integration of the two modules, it can be seen that LAM not only helps to improve the computational accuracy but also reduces the complexity of the model and improves the computational efficiency.

In general, LM is suitable for applications that focus on efficiency, and AM is suitable for applications that focus on accuracy. LAM has achieved the best tradeoff among the computational efficiency, the accuracy, and the model complexity. It is suitable for a wide range of applications.

TABLE II
LIGHTWEIGHT COMPARATIVE EXPERIMENTS WITH AMs

Method	Parameters(M)	FLOPs(G)
SENet-PointNet	4.02	1.04
CBAM-PointNet	4.05	1.06
LAM-PointNet	3.45	0.85
SENet-PointNet++	1.98	3.06
CBAM-PointNet++	2.03	3.07
LAM-PointNet++	1.68	2.87

b) *Standard deviation analysis*: In order to analyze the standard deviation of the overall accuracy, the error bars are shown in Fig. 5. As can be seen, LM makes the fluctuation range of the overall accuracy slightly larger. AM can effectively reduce the fluctuation range of the overall accuracy. LAM obtained by the organic integration of the two sub-modules can effectively suppress the fluctuation of the overall accuracy. The standard deviation of the overall accuracy is controlled at about 1%.

c) *Comparative experiments with AMs*: In order to verify the lightweight effect of LAM, comparative experiments are conducted with other AMs in Table II. As can be observed, LAM has lower model complexity compared with SENet [20] and CBAM [41] in experiments with PointNet and PointNet++ as the backbone architectures. This proves that LAM is more lightweight.

d) *Robustness tests*: In practical applications, the input point clouds are often affected by changes in sampling densities, outliers, and Gaussian noises. In order to show the robustness of the proposed method, robust experiments are conducted on these issues. We obtain the point clouds of each CAD model with different sampling densities by randomly dropping out some points. In the experiment, each model samples 1024, 768, 512, 256, and 128 points. What is more, outliers are evenly added to the point cloud, and Gaussian noise is added independently for each point.

In order to test the robustness of LAM, robustness tests are carried out for sampling density, outliers, and Gaussian noise. Fig. 6 shows the overall accuracy with the increasing of sampling density, outlier ratio, and perturbation noise standard. Fig. 6(a) shows the results with different sampling densities of point clouds. As can be observed, with the increase in the sampling densities, the classification accuracy has shown an overall upward trend. This is basically consistent with the actual situation. We also find that the performance of our methods at each sampling density is always close to the best compared with the backbone architectures. Fig. 6(b) shows the results with different outlier ratios. As can be observed, with the increase in outlier ratios, the classification accuracy has shown an overall downward trend. We also find that the performance of our methods at each outlier ratio is always close to the best compared with the backbone architectures. Fig. 6(c) shows the results with different perturbation noise standards. As can be observed, with the increase in perturbation noise standards, the classification accuracy has shown an overall downward trend. We also find that the performance of our methods at

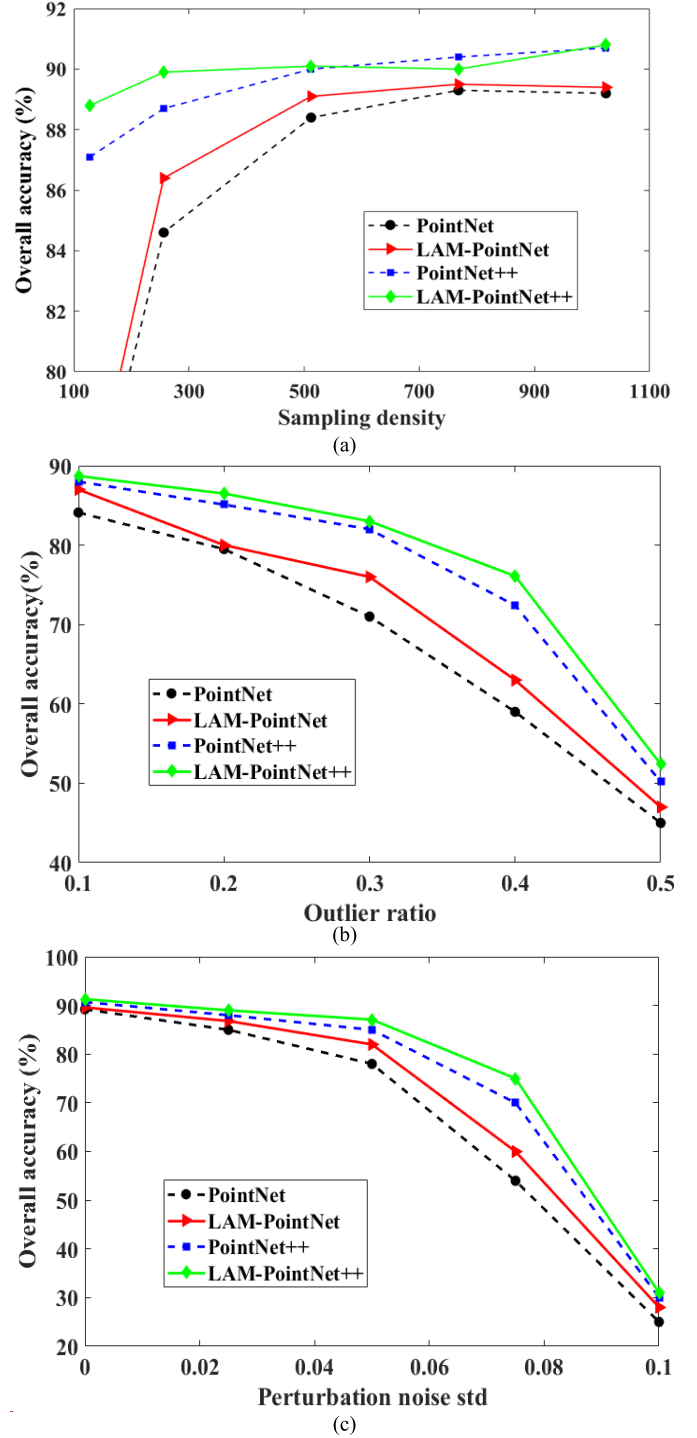


Fig. 6. Classification of point clouds: overall accuracy with the increasing of (a) sampling density, (b) outlier ratio, and (c) perturbation noise standard.

each perturbation noise standard is always close to the best compared with the backbone architectures. This proves that the proposed LAM improves the robustness of classification.

B. Part Segmentation

1) *Data Set*: Compared with classification, part segmentation is a more precise 3-D recognition task. Its main task is to assign part category labels to each point in the point

TABLE III
SEGMENTATION RESULTS ON THE SHAPENET DATA SET

Method	mIoU (%)	Time (s)	Parameters (M)	FL-OPs (G)	Air-craft	Bag	Cap	Car	Chair	Ph-one	Gui-tar	Kn-ife	Lamp	Lap-top	Mo-tor	Mug	Pistol	Roc-ket	Skate-board	Table
PointNet	83.7	329	8.63	0.95	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
LAM-PointNet	84.1	290	7.66	0.87	83.6	82.1	81.7	77.1	89.8	71.0	91.3	85.9	81.6	95.7	64.9	93.4	83.6	58.4	74.3	80.7
PointNet++	85.1	428	1.41	2.65	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
LAM-PointNet++	85.3	397	1.06	2.54	83.0	79.2	87.5	78.4	90.9	70.7	91.3	88.1	84.0	95.3	71.9	94.3	81.9	58.9	76.7	82.8

cloud of a 3-D model. The ShapeNet segmentation data set is used to evaluate our module. It contains 16 881 3-D shapes of 16 object categories. The points in the data set are annotated as a total of 50 parts. Most of the training shapes are labeled as 2–5 parts; 2048 points are sampled from each shape.

2) *Training*: The momentum optimizer is used with an initial learning rate of 0.001. The epoch number is 200. The learning rate decreases with the increase in training epoch number until it reaches 0.00001. The decay step is 200 000 and the decay rate is 0.5. The batch size is 32 and the momentum is 0.9. Other parameter settings are the same as PointNet [15] in the experiment of part segmentation.

3) *Results*: In order to better verify the experimental results of part segmentation, we select reasonable performance metrics. Intersection-over-Union (IoU) represents the ratio between the intersection and union between the actual segmentation point cloud and the ground truth. The IoUs of all the parts that belong to a certain shape are calculated, and then, these IoUs are averaged to calculate the IoU of the shape. The IoU of a category is obtained by averaging the IoUs of all the shapes in it. The mean IoU (mIoU) is finally obtained by averaging the IoUs of all the testing shapes. Time represents the forward pass time of all the test sets and it can well reflect the computational efficiency of the models. Parameters represent the number of weights and bias of the models. FLOPs represent the floating-point operations. Parameters and FLOPs can well reflect the complexity of the models.

In the experiments of part segmentation, we compare LAM-PointNet with PointNet, and LAM-PointNet++ with PointNet++ in terms of the segmentation accuracy, model complexity, and computational efficiency. Table III shows the results of part segmentation on the ShapeNet data set. As can be observed, the model complexity, computational efficiency, and accuracy of both LAM-PointNet and LAM-PointNet++ are always close to the best compared with the backbone architectures. Although our methods have good performance in most of the 16 categories, they do not achieve the expected results in the remaining few. This is mainly due to the fact that small changes in parameters can cause large changes in results.

The part segmentation results of some testing shapes are visualized for showing the details in Figs. 7 and 8. As shown in Fig. 7, compared with the ground truth, there are obvious errors in the results of part segmentation using PointNet. For

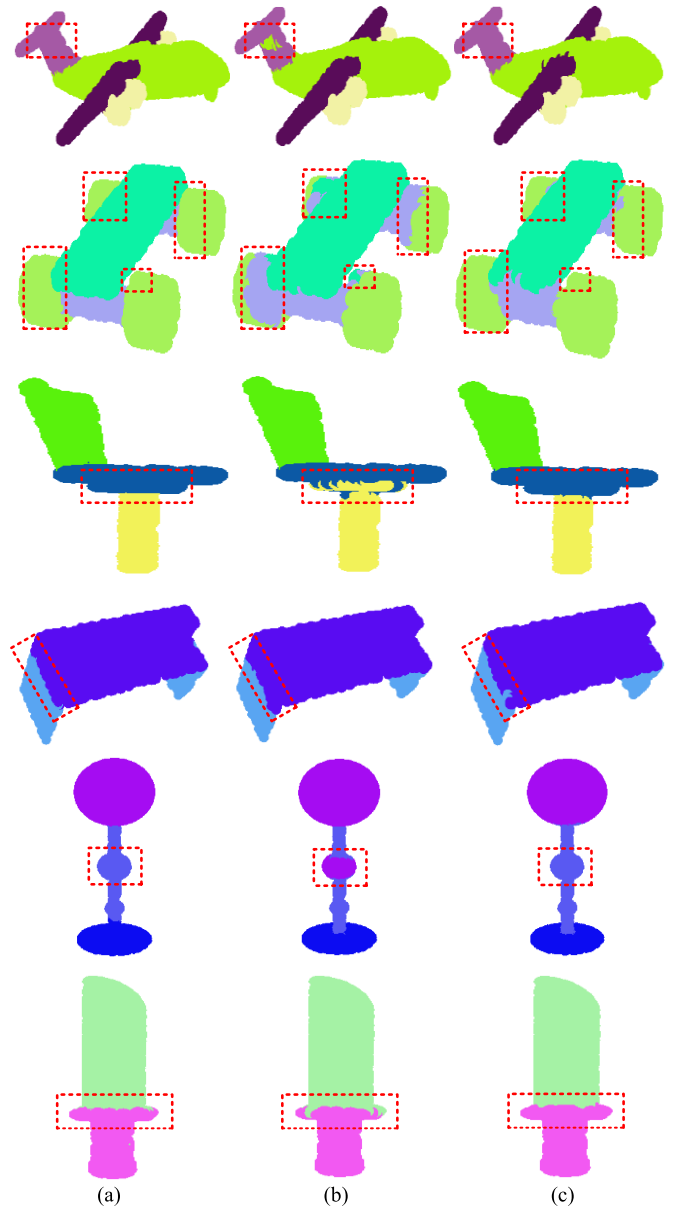


Fig. 7. Part segmentation II on the ShapeNet data set. (a) Ground truth. (b) Results of PointNet. (c) Results of LAM-PointNet.

an aircraft, a part of the tail is incorrectly segmented as the fuselage. For a car, tires are incorrectly segmented as the car body or the axle. For a chair, part of the cushion is incorrectly

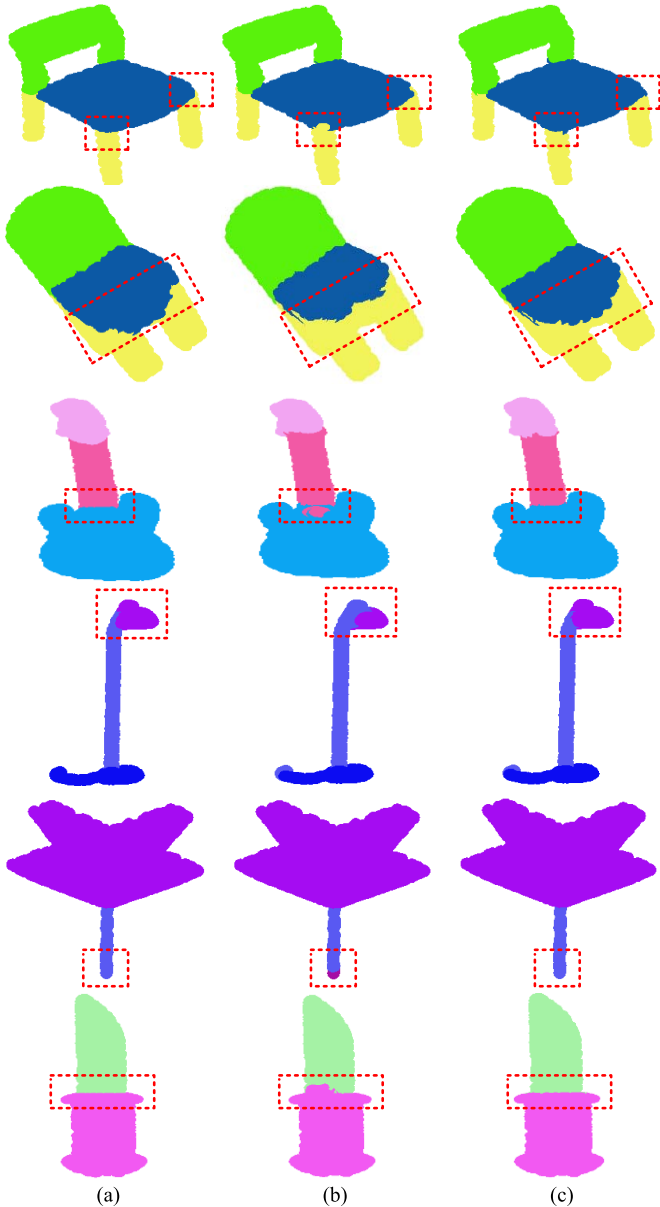


Fig. 8. Part segmentation II on the ShapeNet data set. (a) Ground truth. (b) Results of PointNet++. (c) Results of LAM-PointNet++.

segmented as the leg. These errors have been significantly improved after integrating LAM. The part segmentation results of LAM-PointNet are very close to the ground truth.

As shown in Fig. 8, compared with the ground truth, there are obvious errors in the results of part segmentation using PointNet++. For a guitar, part of the body is incorrectly segmented as the neck. For a lamp, part of the lampshade is incorrectly segmented as the lamppost. For a knife, part of the blade is incorrectly segmented as the handle. These problems are basically resolved after integrating LAM to improve the backbone architectures. The results of LAM-PointNet++ are very close to the ground truth.

It can be seen from Figs. 7 and 8 that segmentation errors are prone to occur at the positions of the connecting surfaces, corner points, edges, and so on. At these locations, the segment results of the methods integrating LAM are much closer to the ground truth than that of the backbone architectures. This can

intuitively show that LAM is helpful to improve the accuracy on part segmentation of point clouds, especially at the key locations that are difficult to segment.

C. Scene Segmentation

1) *Data Set*: The Stanford Large-Scale 3D Indoor Space data set (S3DIS) is used to evaluate our module for the segmentation of indoor scenes. The data set includes the 3-D scan point clouds for six indoor areas, covering a total of 272 rooms. These areas are split into two parts: areas 1–5 for training and area 6 for testing. Each point in the point clouds corresponds to one of the 13 semantic categories; 4096 points are sampled from each block. Each point is represented as a 9-D vector (XYZ, RGB, and normalized location as to the room).

2) *Training*: The momentum optimizer is used with an initial learning rate of 0.001. The epoch number is 50. The learning rate decreases with the increase in training epoch number until it reaches 0.00001. The decay step is 300 000 and the decay rate is 0.5. The batch size is 24 and the momentum is 0.9. Other parameter settings are the same as PointNet [15] in the experiment of scene segmentation.

3) *Results*: In order to better verify the experimental results of scene segmentation, we select reasonable performance metrics in the experiments of scene segmentation. IoU represents the ratio between the intersection and union between the actual segmentation point cloud and the ground truth. The mIoU is finally obtained by averaging the IoUs of all the semantic categories in the testing rooms. Time represents the forward pass time of all the test sets and it can well reflect the computational efficiency of the models. Parameters represent the number of weights and bias of the models. FLOPs represent the floating-point operations. Parameters and FLOPs can well reflect the complexity of the models.

Table IV shows the results of scene segmentation on the S3DIS data set. As can be observed, the computational efficiency, model complexity, and accuracy of LAM-PointNet are always close to the best compared with PointNet. Although our methods have good performance in most of the 13 semantic categories, they do not achieve the expected results in the remaining few. This is mainly due to the fact that small changes in parameters can cause large changes in results.

The scene segmentation results of some testing rooms are visualized for showing the details. As shown in Fig. 8, compared with the ground truth, there are obvious errors in the results of scene segmentation using PointNet. For a conference room, the board has a severe segmentation error and cannot be effectively distinguished from the wall. For a lounge, sofas are incorrectly segmented as chairs. For an office, part of the board is incorrectly segmented as the wall. These problems have been greatly improved after integrating LAM. The indoor scene segmentation results of LAM-PointNet are always closer to the ground truth than that of PointNet in most cases. For some chairs, the segmentation results do not meet expectations.

It can be seen from Fig. 9 that the errors of indoor scene segmentation are prone to occur when adjacent different kinds of objects have similar shapes or very close depth range. At these difficult-to-segment locations, the segment results

TABLE IV
SEGMENTATION RESULTS ON THE S3DIS INDOOR SCENE DATA SET

Method	mIoU (%)	Time (s)	Para- meters (M)	FLOPs (G)	Ceiling	Floor	Wall	Beam	Col- umn	Win- dow	Door	Table	Chair	Sofa	Book- case	Board	Clutter
PointNet	57.0	174	1.41	7.22	91.2	96.5	70.5	49.2	25.9	40.7	68.0	67.0	65.8	11.3	57.7	48.0	49.5
LAM-PointNet	59.1	160	1.14	7.08	91.7	97.2	70.5	50.5	29.3	38.7	69.3	66.7	61.2	38.4	55.7	48.9	50.4

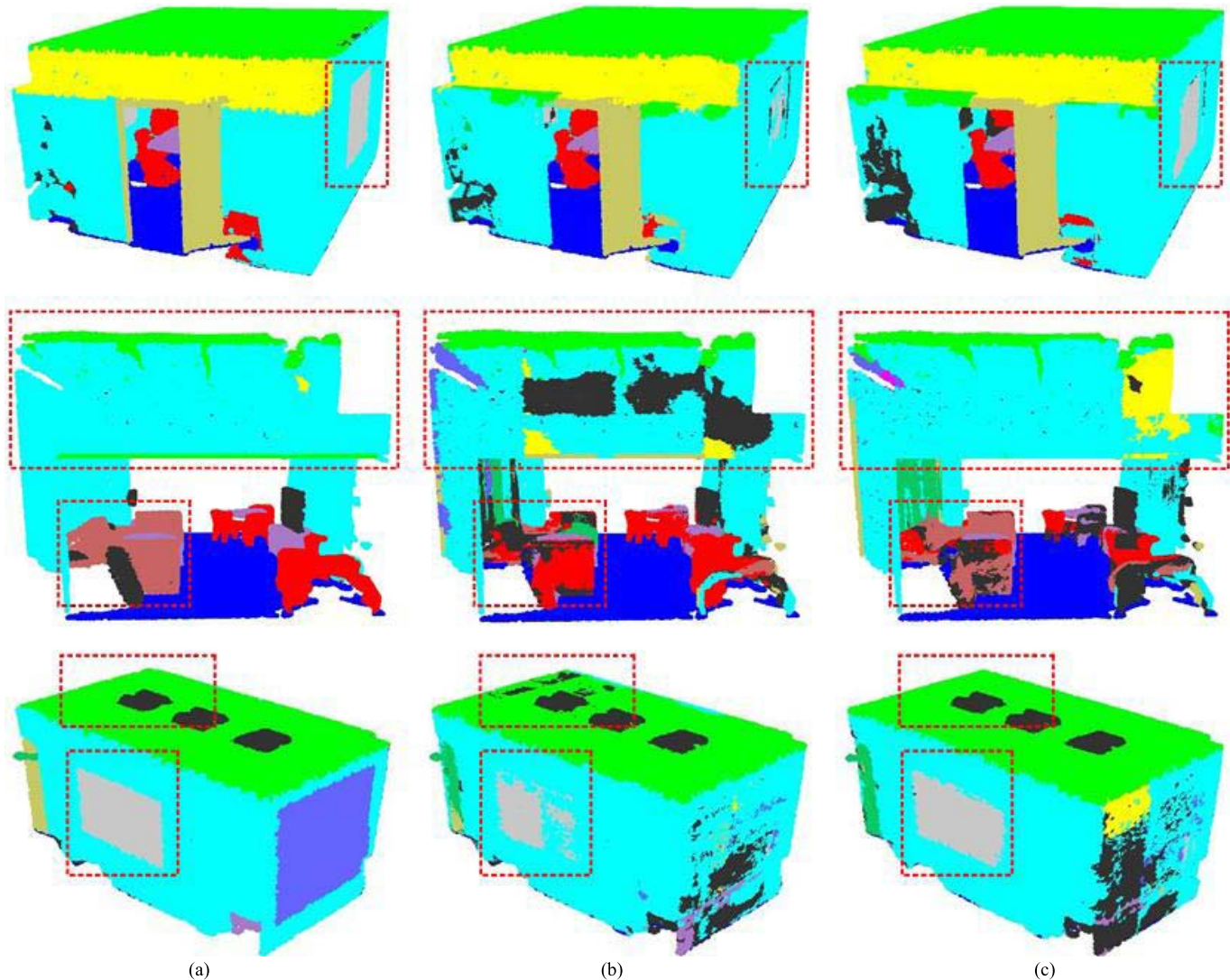


Fig. 9. Scene segmentation on the Stanford Large-Scale 3-D Indoor Space Data Set. (a) Ground truth. (b) Results of PointNet. (c) Results of LAM-PointNet.

of the methods integrating LAM are always closer to the ground truth than that of the backbone architectures. This can intuitively show that LAM is helpful to improve the accuracy on scene segmentation of point clouds, especially at the key locations that are difficult to segment.

D. Discussion

To show the performance of our method, the results are explained in more detail in comparison with state-of-the-art methods. For the computational accuracy, Kd-Net [42] is 1.4% higher than PointNet [15], PointNet++ [16] is 1.5% higher than PointNet, and PCNN [43] is 1.7% higher than Kd-Net. Based on a comprehensive comparison, the improvement of our methods on the accuracy of the original algorithms is

competitive. For computational efficiency, LAM helps PointNet to process 45 more classification models per second and helps PointNet++ to process 26 more classification models per second. The improvement of 2–10 s is very important for practical applications, which can effectively improve the real-time performance of the algorithm. What is more, as the amount of point clouds increases, the more time our model can save. With the wide application of point cloud in large scenes, our proposed modules can save more time.

The reason that our method can achieve good performance is mainly due to the following aspects.

1) By adopting DSConv, the backbone architectures become more lightweight to improve the computational efficiency on classification and segmentation of point clouds.

2) By integrating a new attention mechanism based on channelwise statistical features, the weights of different feature channels are adaptively rescaled to improve the accuracy on classification and segmentation of point clouds.

3) We integrate DSConv and the attention mechanism to propose a new module LAM, which is easy to be implemented and has good portability.

V. CONCLUSION

In this article, we propose a novel module LAM for deep learning on classification and segmentation of point clouds. The main technical contributions of our module include a new convolution mode, a new attention mechanism based on channelwise statistical features, and a modular structure obtained by organic combination of different submodules. The adoption of DSConv makes the backbone architectures more lightweight. The definition and integration of the new attention mechanism enable the weights of different feature channels to be adaptively rescaled. As the submodules of LAM, LM and AM can also be used independently to focus on improving the computational efficiency and accuracy, respectively, according to the practical applications. The modular structure makes LAM and its two submodules have good portability. These technical contributions make our module easily be integrated into state-of-the-art deep learning methods on classification and segmentation of point clouds and contribute to improving the computational efficiency and accuracy of the backbone architectures. Experimental results show that the improved methods obtained by integrating LAM have better performance and are easy to be implemented in practical applications.

REFERENCES

- [1] F. Wang, Y. Zhuang, H. Gu, and H. Hu, "Automatic generation of synthetic LiDAR point clouds for 3-D data analysis," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 7, pp. 2671–2673, Jul. 2019.
- [2] Y. An, B. Li, H. Hu, and X. Zhou, "Building an omnidirectional 3-D color laser ranging system through a novel calibration method," *IEEE Trans. Ind. Electron.*, vol. 66, no. 11, pp. 8821–8831, Nov. 2019.
- [3] C. R. Qi, "Deep learning on point clouds for 3D scene understanding," Ph.D. dissertation, Dept. Elect. Eng., Stanford Univ., Stanford, CA, USA, 2018.
- [4] Z. Qiu, Y. Zhuang, F. Yan, H. Hu, and W. Wang, "RGB-DI images and full convolution neural network-based outdoor scene understanding for mobile robots," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 1, pp. 27–37, Jan. 2019.
- [5] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan, "An integrated framework for 3-D modeling, object detection, and pose estimation from point-clouds," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 3, pp. 683–693, Mar. 2015.
- [6] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan, "3D object recognition in cluttered scenes with local surface features: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2270–2287, Nov. 2014.
- [7] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," 2018, *arXiv:1801.07829*. [Online]. Available: <http://arxiv.org/abs/1801.07829>
- [8] A. X. Chang *et al.*, "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*. [Online]. Available: <http://arxiv.org/abs/1512.03012>
- [9] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 945–953.

- [10] Y. Feng, Z. Zhang, X. Zhao, R. Ji, and Y. Gao, "GVCNN: Group-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 264–272.
- [11] H. Guo, J. Wang, Y. Gao, J. Li, and H. Lu, "Multi-view 3D object retrieval with deep embedding network," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5526–5537, Dec. 2016.
- [12] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 922–928.
- [13] Z. Wu *et al.*, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.
- [14] M. Gadelha, R. Wang, and S. Maji, "Multiresolution tree networks for 3D point cloud processing," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 105–122.
- [15] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.
- [16] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.
- [17] G. Li, M. Müller, A. Thabet, and B. Ghanem, "DeepGCNs: Can GCNs go as deep as CNNs?" 2019, *arXiv:1904.03751*. [Online]. Available: <http://arxiv.org/abs/1904.03751>
- [18] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," 2016, *arXiv:1610.02357*. [Online]. Available: <http://arxiv.org/abs/1610.02357>
- [19] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: <http://arxiv.org/abs/1704.04861>
- [20] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation networks," 2017, *arXiv:1709.01507*. [Online]. Available: <http://arxiv.org/abs/1709.01507>
- [21] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
- [22] L. Yi *et al.*, "A scalable active framework for region annotation in 3D shape collections," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–12, Nov. 2016.
- [23] I. Armeni *et al.*, "3D semantic parsing of large-scale indoor spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1534–1543.
- [24] D. Kong, L. Xu, X. Li, and S. Li, "K-Plane-based classification of airborne LiDAR data for accurate building roof measurement," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 5, pp. 1200–1214, May 2014.
- [25] O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or, "A survey on shape correspondence," *Comput. Graph. Forum*, vol. 30, no. 6, pp. 1681–1707, Sep. 2011.
- [26] Y. An, L. Wang, R. Ma, and J. Y. Wang, "Geometric properties estimation from line point clouds using Gaussian-weighted discrete derivatives," *IEEE Trans. Ind. Electron.*, early access, Jan. 15, 2019, doi: [10.1109/TIE.2020.2965456](https://doi.org/10.1109/TIE.2020.2965456).
- [27] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3212–3217.
- [28] F. Tombari, S. Salti, and L. D. Stefano, "Unique signatures of histograms for local surface description," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2010, pp. 356–369.
- [29] F. Tombari, S. Salti, and L. Di Stefano, "A combined texture-shape descriptor for enhanced 3D feature matching," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 809–812.
- [30] H. Ling and D. W. Jacobs, "Shape classification using the inner-distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 286–299, Feb. 2007.
- [31] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, May 1999.
- [32] H. Chen and B. Bhanu, "3D free-form object recognition in range images using local surface patches," *Pattern Recognit. Lett.*, vol. 28, no. 10, pp. 1252–1262, Jul. 2007.
- [33] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3D object recognition," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops, ICCV Workshops*, Sep. 2009, pp. 689–696.

- [34] W. Zhou, C. Ma, T. Yao, P. Chang, Q. Zhang, and A. Kuijper, "Histograms of Gaussian normal distribution for 3D feature matching in cluttered scenes," *Vis. Comput.*, vol. 35, no. 4, pp. 489–505, Apr. 2019.
- [35] Z. Geng, Y. Zhang, C. Li, Y. Han, Y. Cui, and B. Yu, "Energy optimization and prediction modeling of petrochemical industries: An improved convolutional neural network based on cross-feature," *Energy*, vol. 194, Mar. 2020, Art. no. 116851.
- [36] C. R. Qi, H. Su, M. NieBner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view CNNs for object classification on 3D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5648–5656.
- [37] L. Minto, P. Zanuttigh, and G. Pagnutti, "Deep learning for 3D shape classification based on volumetric density and surface approximation clues," in *Proc. 13th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2018, pp. 317–324.
- [38] H.-C. Shin *et al.*, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1285–1298, May 2016.
- [39] C. Wang, M. Pelillo, and K. Siddiqi, "Dominant set clustering and pooling for multi-view 3D object recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1–12.
- [40] Y. Li, S. Pirk, H. Su, C. R. Qi, and L. J. Guibas, "FPNN: Field probing neural networks for 3D data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 307–315.
- [41] S. Woo, J. Park, J. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Computer Vision*, V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss, Eds. Cham, Switzerland: Springer, 2018, pp. 3–19.
- [42] R. Kulkov and V. Lempitsky, "Escape from cells: Deep kd-networks for the recognition of 3D point cloud models," 2017, *arXiv:1704.01222*. [Online]. Available: <http://arxiv.org/abs/1704.01222>
- [43] M. Atzmon, H. Maron, and Y. Lipman, "Point convolutional neural networks by extension operators," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–12, Aug. 2018.



Wei Sun received the B.S. degree in mechanical engineering and the M.S. and Ph.D. degrees from the Dalian University of Technology, Dalian, China, in 1988, 1993, and 2000, respectively.

He is currently a Professor and a Doctoral Supervisor with the School of Mechanical Engineering, Dalian University of Technology. His current research interests include intelligent mechanical equipment, 3-D environment perception, knowledge-based product digital design, and design and optimization of complex mechanical equipment.



Huosheng Hu (Senior Member, IEEE) received the M.Sc. degree in industrial automation from Central South University, Changsha, China, in 1982, and the Ph.D. degree in robotics from the University of Oxford, Oxford, U.K., in 1993.

He is currently a Professor with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K., where he is leading the Robotics Research Group. He has authored over 420 articles. His current research interests include robotics, human-robot interaction, embedded systems, mechatronics, and pervasive computing.

Dr. Hu is also a Founding Member of the IEEE Robotics and Automation Society Technical Committee on Networked Robots, a fellow of the Institution of Engineering and Technology, and a Senior Member of the Association for Computing Machinery. He also serves as the Editor-in-Chief for the International Journal of Automation and Computing and the online *Robotics journal* and an Executive Editor of the *International Journal of Mechatronics and Automation*.



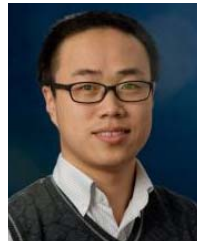
Yunhao Cui received the M.S. degree in mechanical design and theory from Northeastern University, Shenyang, China, in 2014. He is currently pursuing the Ph.D. degree with the School of Mechanical Engineering, Dalian University of Technology, Dalian, China.

His research interests include point cloud data processing, intelligent mechanical equipment, and 3-D environment perception.



Yi An (Member, IEEE) received the B.S. degree in automation and the M.S. and Ph.D. degrees in control theory and control engineering from the Dalian University of Technology, Dalian, China, in 2001, 2004, and 2011, respectively.

From 2007 to 2011, he was a Lecturer with the School of Control Science and Engineering, Dalian University of Technology, where he has been an Associate Professor since 2012. His research interests include point cloud data processing, sensing and perception, information fusion, robot vision, and intelligent robot.



Xueguan Song received the B.S. degree in mechanical engineering from the Dalian University of Technology, Dalian, China, in 2004, and the M.S. and Ph.D. degrees in mechanical engineering from Dong-A University, Busan, South Korea, in 2007 and 2010, respectively.

He is currently a Professor with the School of Mechanical Engineering, Dalian University of Technology. His current research interests include intelligent mechanical equipment, point cloud data processing, multidisciplinary design optimization, and simulation-based engineering design (finite element method and computational fluid dynamics).