IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 16, 2021, accepted May 7, 2021, date of publication May 11, 2021, date of current version May 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3079295

Multi-Head Attentional Point Cloud Classification and Segmentation Using Strictly Rotation-Invariant Representations

ZHIYONG TAO[1], YIXIN ZHU[1], TONG WEI[2], AND SEN LIN[3]
[1]School of Electronics and Information Engineering, Liaoning Technical University, Huludao 125105, China
[2]Faculty of Informatics, Eötvös Loránd University, 1117 Budapest, Hungary
[3]School of Automation and Electrical Engineering, Shenyang Ligong University, Shenyang 110159, China

Corresponding author: Zhiyong Tao (xyzmail@126.com)

This work was supported by the National Key Research and Development Program of China under Grant 2018YFB1403303.

ABSTRACT Point cloud processing plays an increasingly essential role in three-dimensional (3D) computer vision target, scene parsing, environmental perception, etc. Compared with using aligned point cloud data for classification and segmentation, the strictly rotation-invariant representations show enough robustness. Inspired by the great success of deep learning, we propose a novel neural network called Multi-head Attentional Point Cloud Classification and Segmentation Using Strictly Rotation-invariant Representations. Our research focuses on processing the point cloud rotated in any direction effectively and precisely. First of all, the strictly rotation-invariant point cloud representations are obtained through point projection. Then we apply a multi-head attentional convolution layer (MACL) using attention coding to develop the performance of point cloud feature extraction. Finally, our network assigns different responses and recognizes the overall geometry well through a key point descriptor, adding to the global feature. Our method can explore more in-depth information for accuracy enhancement with attention pooling and multi-layer perceptron (MLP) based on an advanced DenseNet. Our network enjoys 90.63% and 87.50% classification accuracy testing on ModelNet10 and ModelNet40, and 75.15% intersection over union metric (mIoU) evaluating on ShapeNet Part dataset, remaining under any rotation. Rotating experimental results indicate that our framework realizes better point cloud classification and segmentation performance than most state-of-the-art methods.

INDEX TERMS Point cloud, deep learning, strictly rotation-invariant representations, attention coding, classification and segmentation.

I. INTRODUCTION

Point cloud processing plays an increasingly important role in 3D object recognition technologies. Three-dimensional data representations, like point clouds, can be received conveniently due to recent sensor technology development. Point cloud has sufficient geometric information, widely applying for numerous fields, such as robotics, remote controlling, and self-driving.

According to the great success of deep learning and convolutional neural networks (CNN) [1] in the research of computer vision and graphics with their powerful data processing capabilities. AlexNet [1] first applied CNN in the field of image classification successfully, proving the great potential of CNN in the large-scale data processing. Part of the research has designed residual blocks [2] and dense blocks [3] to optimize the CNN and improve its data processing capabilities. For further study and application, researchers contributed to the analysis of neural networks in various ways. For example, Wang *et al*. [4] utilized graph theory and event-triggered control mechanism and analyzed multiple memristive neural networks (MMNNs), useful for nonlinear systems. A neural network is considered as a typical nonlinear system [5]. As shown in [6], some numerical examples were introduced, then two methods of continuous sampling and discrete sampling were adopted for analysis. They are the mathematical methods to explore neural network models with more complete functions and superior performance theoretically. A norm-based threshold function and signal function controller were proposed for parameter mismatch and

The associate editor coordinating the review of this manuscript and approving it for publication was Varuna De Silva.

theoretical error in nonlinear systems in [7]. Furthermore, deep learning-based models performed unexpectedly in various computer vision tasks, such as hyperspectral image [8], [9] and space remote sensing data processing [10].

More researchers applied and analyzed the neural networks for 3D data processing, As for point cloud preprocessing, the irregular format is a dominant problem. Generally, raw point cloud data is transferred to other regular collections for classification, segmentation, and other tasks. As Qi *et al*. [11] firstly proposed a deep learning framework called PointNet to consume raw point clouds directly, without any transformation, more researchers have explored remarkable achievements, such as DGCNN [12], KPConv [13], Point2sequence [14], and PointConv [15]. Whereas, they can only address the original point cloud coordinates straightway from specific normative directions. When we make some rotation transformation to point clouds, the recognition performance will be sharply decreased. From the aspect of practical applications, intelligent robots [16] and self-driving cars [17] need to recognize general three-dimensional objects with arbitrary rotation accurately. Eliminating the interference of rotation transformation is an essential guarantee for their reliability.

To enhance the robustness of point cloud rotation, partial works [18], [19] [20] applied spherical voxel convolution (SVC) [21] to address point clouds. The discretized sphere had global directions, which could not promise extreme symmetry, or even the rotation-invariant point information. For this reason, RIConv [22] was designed as a convolution operator, which succeeded in rotation-invariant feature learning and robustness improvement. REQNN [23] proposed a rotation equivariant quaternion framework according to the rotation equivariance properties. However, they were exposed to lower accuracy, which prompted the research development. RMGNet [24] constructed a multi-scale graph convolutional neural network for segmentation through the handcrafted rotation-invariant features. In addition, ClusterNet [25] and SRINet [26] were designed for strictly rotation-invariant representations through point projection. Both of them failed to capture detailed point cloud features. The former approach cannot perceive point clouds' overall geometric structure, and the other was not stabilized enough.

Thus, following the previously published samples, there remains a requirement for novel methodologies to fully discover the features of point clouds and global geometric information. Rotated point feature studies make sense for the accuracy enhancement and application for computer sensing in the real world. Here, we propose a novel model called Multi-head Attentional Point Cloud Classification and Segmentation Using Strictly Rotation-invariant Representations.

Compared with state of the art, our method has achieved strictly rotation-invariant classification and segmentation on point sets with high accuracy and robustness. The main contributions of our work are summarized as follows:

- We obtain the strictly rotation-invariant point cloud representations through the coordinate transformation of point projection. For the input points, the key point descriptor is also utilized to deliver different responses, recognize the overall geometry well, and help build a global feature.
- Robustness is essential to consider in 3D point cloud classification algorithms. For this purpose, we design and propose a novel neural network based on the multi-head attentional convolution layer (MACL) for point clouds, strengthening network robustness and developing feature extraction performance.
- Furthermore, we modify the structure of the original DenseNet and combine it with multi-layer perceptron (MLP) [11], achieving deep information and high accuracy. Besides, the attention mechanism is applied to the pooling layer for local 3D point cloud feature acquisition.

The remaining parts of this paper are structured as follows. Section II displays the related works of deep learning on the point cloud and its rotation-invariant representations. Section III shows the methodology of our model proposed in this paper. Next, rotating experimental results, ablation discussions, and remarks on our proposed method are shown in Section IV. Finally, we have drawn some conclusions and given possible feature studies in Section V.

II. RELATED WORK

A. DEEP LEARNING ON 3D POINT CLOUD

With the rapid development of CNN, more and more studies have designed effective neural networks to deal with 3D point clouds. In the beginning, we could not directly input the disordered point sets to CNN. Some previous works adopted voxels [27], [28] or multi-view forms [29], [30] for specific tasks. However, a multi-view point cloud may cause the generated data lack of shape information and loss of all the original geomatical details. Voxels account for excessive memories and increasing time complexity, which causes dramatic damage to computational efficiency. To break through this limit, PointNet [11] has shown the potential of neural networks to process the raw point cloud directly. PointNet aggregated the global feature by symmetric functions and achieved permutation invariance. Based on PointNet, PointNet++ [31] developed multi-scale hierarchical local feature learning through the farthest point sampling (FPS) algorithm. These methods concentrated on the point information, rather than relations between neighbors. Besides, some papers enhanced accuracy through local neighbor relations between neighbor and center nodes. For example, DGCNN [12] designed edge convolution to merge graph features from neighbors. GAPNet [32] is a branch structure consisting of MLP for point information and edge convolution for the global feature.

Point cloud models are so irregular that general convolution cannot perform well, leading to information loss. Then, some people defined convolution in Non-euclidean domain, such as, SO-Net [33], PointCNN [34], and SPLATNet [35]. SO-Net modeled point cloud space distributions

through Self-Organization Mapping (SOM) Network. Also, there are flexible specifications of the lattice structure in SPLATNet [35], enabling hierarchical and spatially-aware feature learning and joint 2D-3D reasoning. PointCNN [34] utilized X-conv to make sure permutation invariance. Others designed novel convolution operators to advance the performance and updated weights through geometric distributions, defined convolution kernel in sequential space. In particular, PointConv [15] defined convolution as relative important sampling sequential 3D convolution Monte Carlo method. ShellConv [36] proposed permutation invariant convolution (Shellconv) based on concentric spherical shell structure. KPConv [13] adopted the rigid and deformable kernel point convolution operators using a set of learnable kernel points. Although these works have made significant progress in this field, they utilized the raw coordinates as the input, which has low robustness to the rotation translation.

B. ROTATION-INVARIANT REPRESENTATIONS FOR POINT CLOUDS

In terms of indefinite directions in the real scenario, general point cloud classification networks cannot realize rotation invariance, limiting the application development. Some prior studies applied spherical voxel convolution (SVC) [21] to learn about point clouds, aiming to enhance the robustness of transformation. For example, Poulenard [19] employed volume function for point representation and spherical harmonics based kernels to improve the convolution computing process. PRIN [18] employed the Density-aware adaptive sampling (DAAS) to convert sparse signals to voxels and SVC to extract approximate rotation-invariant features at point levels instead of global ones. Rao *et al*. [20] made an adaptive projection on the discretized sphere and captured patterns through a hierarchical feature learning model. The discretized sphere had global directions, which failed to reach extreme symmetry, or even rotation-invariant point information.

For this reason, some researchers proposed novel operators to learn rotation-invariant features. Zhang *et al*. [22] designed the rotation-invariant operator IRConv to learn the underlying rotation-invariant geometric features, such as distance and angles. Moreover, several researchers utilized point projection to learn rotation-invariant point cloud representations. Zhang *et al*. [23] applied rotation equivariance to the input point cloud and exhibited higher rotation robustness. The specific rotation transformation they used could cause the same rotation transformation to all intermediate-layer quaternion features. Furuya *et al*. [24] published a rotation-invariant multi-scale framework for any manual 3D graph features with rotation invariance, based on the relations between 3D points and their normal vectors. Chen *et al*. [25] designed ClusterNet [37] for strictly point-wise rotation-invariant representation by rigorously rotation invariant (RRI). ClusterNet adopted graph CNN for feature extraction and unsupervised hierarchical clustering method based on supervised connection standard. It applied EdgeConv [12] for feature acquisition but failed to perceive the overall geometric structure of point sets. In 2019, Sun *et al*. [26] came up with SRINet, which used point projection to represent the rotation-invariant point cloud and PointNet's backbone for global features. SRINet also included graph clustering for local details. Inspired by rotation-invariant, we propose this model to parse geometric information and perform better in 3D point classification and segmentation tasks.

III. METHODOLOGY

This section shows point cloud representations from point projection, multi-head attentional coding method, attention pooling layer construction, and key point detection. The overall framework is demonstrated in Fig. 5.

A. POINT PROJECTION

Point clouds represent the set of measured points [38]. Suppose we place it in a conventional coordinate system, along with rotation transformation, the relative position relations between the axes and points will change significantly.

Generally, deep learning structures work on coordinates. Rotation transformation will affect the recognition and feature extraction performance by neural networks. Our work adopts point projection [26] on the axis, eliminates rotation influence, and achieves the strictly rotation-invariant representations, shown in Figure 1.

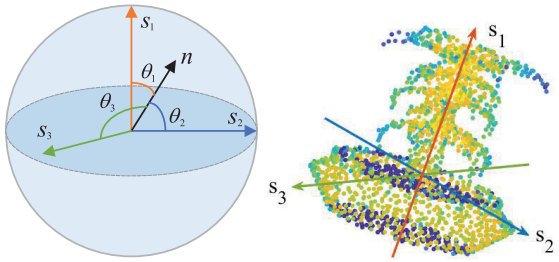


FIGURE 1. We present the abstract mathematical representations (left) of the selected coordinates (right) in point sets. The origin is located at the center of mass. The coordinates are redefined by axes (s_1, s_2, s_3).

In learning algorithms, most testing datasets are in the fixed directions. We need to augment the testing data for robustness testing, in which rotation transformation is a general method. According to Euler's rotation theorem [39], any rotation can be represented by an Euler axis and a rotation angle [25]. The three-dimensional unit vector rotation angle is a scalar. We can use the following formula to solve the rotation matrix R corresponding to the Euler axis s and the rotation angle θ, and I_3 represents the input vector.

$$R = I_3 cos\theta + (1 - cos\theta)ss^T + [s]_x sin\theta,$$
$$[s]_x \triangleq \begin{bmatrix} 0 & -s_3 & s_2 \\ s_3 & 0 & -s_1 \\ -s_2 & s_1 & 0 \end{bmatrix} \quad (1)$$

According to the above, we sample the points as Euler axis, and then uniformly sample the rotation angle in the space

of $[0, 2\pi]$. We choose this method to generate Euler axis and rotation angle, and then solve the rotation matrix by the Equation (1). From this, we can get the points after the rotation transformation for testing.

Algorithm 1 Point Projection Principle

Input: Point vector $n : N \times 3$;

Output: Point projection vector $f : N \times 4$;

1: With the given $(i = 1, 2, 3)$, calculte the 2-norm of every input point x_i,as shown in Equation (2);
2: Within those norm values, choose the input vector n with maximum one as n_1, and that with the minimum as n_2 (usually these two chosen norms are inverse);
3: Normalize the n_1, n_2, then achieve two axes: $s_1 = \frac{n_1}{|n_1|}$ and $s_2 = \frac{n_2}{|n_2|}$;
4: Calculate the cross product of $s_1 \times s_2$ and take its unit vector as $s_3 = \frac{s_1 \times s_2}{|s_1 \times s_2|}$;
5: With the given $(i = 1, 2, 3)$, take the norm of the input vector n as $|x|$, you can get projection $cos(s_i, x) = s_i \times n/x$;
6: Combine the projection $cos(s_i, x)$ with n as a novel vector, $f = (cos(s_1, x), cos(s_2, x), cos(s_3, x), |x|)$;

Assume the input point cloud is a random point set $\{X = x_i \subseteq R^3\}$, then we redefine three linear independent axes and point representatives. The origin is located at the mass point, and each point x_i refers to a vector n. As is shown in [26], choosing the vector of maximum norm as Axis s_1, and minimum norm as Axis s_2. The multiplication cross product of s_1 and s_2 is referred as s_3, with unit norm scaled. The norm is calculated by:

$$n_i = (\sum_k n_k^2)^{\frac{1}{2}} \tag{2}$$

where n_k represents the elements of vector n, and n_i is the vector norm. The angle (s_1, s_2, s_3) between vectors by norm n_i are not calculated further because they will not collide with each other in four-dimensional projection feature space [26]. In this novel coordinate system, no matter how the point cloud data rotates, the relative positions between axes and points will keep consistent. Through the process of point projection, the point cloud is invariant to rotations. As shown in Algorithm 1, we project points on the axes, and construct four-dimensional (4D) projection features $(f(s, x_i))$ for each point, combining with the length of vector x_i, provided by Equation (3):

$$f(s, x_i) = (\cos(s_1, x_i), \cos(s_2, x_i), \cos(s_3, x_i), |x_i|) \tag{3}$$

where $\cos(s_i, x_i)$ represents the point projection feature on axes, and $|x_i|$ refers to the length of vector x_i. Then we encode original point sets as a collection (F) of point projection features, as Equation (4):

$$F = \{f(s, x_1), f(s, x_2), \ldots, f(s, x_n) \in R^{N \times 4}\} \tag{4}$$

where s denotes three axes - (s_1, s_2, s_3), f represents point projection mapping, and $R^{N \times 4}$ stands for 4D projection space. The proof procedure of rotation-invariant the point projection feature is as followed:

The first step is to assume point x and three axes, labeling partial components of 4D features as:

$$(x_n, s_1) = f_1, (x_n, s_2) = f_2, (x_n, s_3) = f_3 \tag{5}$$

where $x_n = \frac{x}{|x|}$. Simply, we define $m_{ij} = (s_i, s_j)$, $m_{ij} = m_{ji}$, because the constant relative relation of s_i and s_j. Next, suppose $C = (x_n, s)$ and matrix M as:

$$\begin{bmatrix} x_n^T \\ s_1^T \\ s_2^T \\ s_3^T \end{bmatrix} \begin{bmatrix} x_n & s_1 & s_2 & s_3 \end{bmatrix} = \begin{bmatrix} 1 & f_1 & f_2 & f_3 \\ f_1 & 1 & m_{12} & m_{13} \\ f_2 & m_{21} & 1 & m_{23} \\ f_3 & m_{31} & m_{32} & 1 \end{bmatrix} \triangleq M \tag{6}$$

Once M is determined, we could find out vector C through Singular Value Decomposition, where $M = USV^T$. Then $C = US^{1/2}V^T$. The value of m_{ij} depends on axes, then elements m_{ij} will remain with the consistent axes. Here we transform the rotation matrix to homogeneous coordinates $([1, X, Y, Z]^T)$ in 4D space for rotation-invariance verification, as

$$R_{4\times4} = \begin{bmatrix} 1 & 0 \\ 0^T & R \end{bmatrix} \tag{7}$$

When using orthogonal rotation matrix $R_{4\times4}$ in 4D space to rotate projected point cloud features, the result matrix M is unchanged, given by:

$$(RC)^T(RC) = C^T R^T RC = C^T C = M \tag{8}$$

In conclusion, projected point presentations are rotation-invariant.

B. MULTI-HEAD ATTENTIONAL ENCODER

This paper combines the attention mechanism [40] and multi-head structure [32] with the convolutional layer to construct a better encoder for 3D point sets. An ideal 3D object recognition model could capture context information in the global space and capture fine-grained local information. First, these two kinds of information balance each other, so we design an attentional convolution layer (ACL). We combine the fine-grained attention features from ACL and the local neighbor information, as in Fig. 2.

To obtain global features based on neighborhood, we project the input point cloud x_i and get the rotation-invariant representation f_i. There is an auto-encoder for each rotation-invariant point, according to the local field choosing mechanism. That means we adopt MLP with F' convolutional kernels to map them to high-dimensional feature space. Equation (9) presents feature u_i' with $N \times F'$ dimensions.

$$u_i' = \mathcal{X}(B(C_{F' \times 1}(f_i, \theta))) \tag{9}$$

where $\mathcal{X}$ denotes parameterized non-linear activation function Relu, θ is a set of learning parameters. B represents normalization. C refers to convolutional computation, with the convolutional kernel of $F' \times 1$ on the subscript.

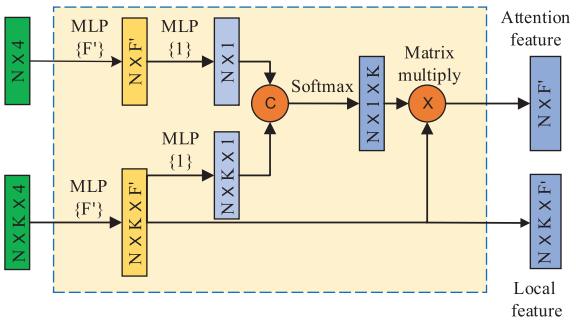


FIGURE 2. Here is the framework of the attentional convolution layer (ACL). We use *N* on behalf of point number, and MLP{ } for multi-layer perceptron, where the content in brace represents the number of convolution kernels. *C* represents concatenation and *X* for matrix multiplication calculation.

Furthermore, to develop the ability of local feature expression, our network searches for the nearest k points using KNN (K-nearest neighbor) algorithm [41]. These points will form a k-neighborhood structure, make a projection, then output local rotation-invariant representation f_{ij}. We apply MLP with F' convolution kernels to get $N \times \mathrm{K} \times F'$ dimensional feature v'_{ij}, processing as Equation (10):

$$v'_{ij} = \mathcal{X}(B(C_{F'\times 1}(f_{ij}, \theta))) \tag{10}$$

where,

$$f_{ij} = x_i - x_{ij} \tag{11}$$

here, x_{ij} is the point near to x_i. Conducting single-layer covolutional computation on u'_i and v'_{ij} can output two 1-dimentional vectors, assigning as the self and neighbor attention coefficients. Subsequently, they are combined in the same feature level without weights to obtain the neighborhood selection coefficient from description x_i to its K-nearest points, given by Equation (12):

$$b_{ij} = LR(\mathcal{X}(B(C_{1\times 1}(u'_i, \theta))) + \mathcal{X}(B(C_{1\times 1}(v'_{ij}, \theta)))) \tag{12}$$

where $LR()$ refers to non-linear activation function - LeakyRelu. Meanwhile, the Softmax function is employed for normalization and increasing convergence rate, as in Equation (13):

$$a_{ij} = \frac{exp(b_{ij})}{\sum_{j=1}^{k} exp(b_{ij})} \tag{13}$$

Next, we multiplied the normalized neighborhood selection coefficient a_{ij} with K neighbor features v'_{ij}, resulting in fine-grained local characteristics with the dimension of $N \times F'$, as following:

$$x'_i = f(\sum_{j\in N_i} a_{ij} v'_{ij}) \tag{14}$$

where f denotes the non-linear activation function, ELU. Different wights will be allocated to neighbor features by the feature selector a_{ij}. Non-meaning neighbor features have lower weights than discriminative ones, which makes efforts to detect fine-grained. For the stable architecture and enriched

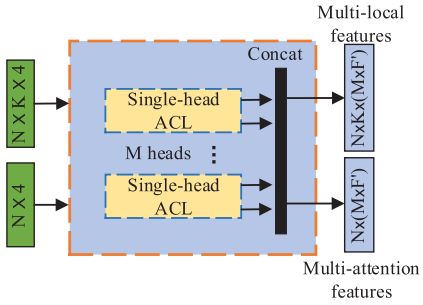


FIGURE 3. Here we present the MACL structure. We input the results of point projection and KNN, then output the multi-local and neighborhood features.

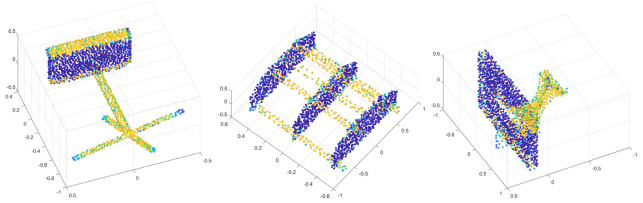


FIGURE 4. Visualization point cloud processed by key point detection.

feature details, we combine the multi-head structure with ACL, called MACL. Independent ACL layers are concatenated together, generating the multi-channel attentional feature $\hat{x}_i'$, with the size of $M \times F'$. Here is the mathematical principle:

$$\hat{x}_i' = \mathop{\Big\|}_{m}^{M} \hat{x}_i^{(m)} \tag{15}$$

In Equation (15), m is the total number of heads, setting to four. $\mathop{\|}_{m}^{M}$ denotes concatenations in feature channels. As shown in Fig. 3, MACL outputs multi-attention neighbor and local features.

C. ATTENTION POOLING

As for local features, we also apply the attention mechanism [40] to pooling calculations and present attention pooling layer (APL) [32] for point cloud processing. Based on max pooling, APL could recognize the essential local features and compensate for the global, provided by Equation (16):

$$y_i = \mathop{\Big\|}_{m}^{M} \max v_{ij}^{'(m)} \tag{16}$$

where max is the maximum calculation, and m equals four (the head numbers), defined in MACL. v'_{ij} denotes local features, and $\mathop{\|}_{m}^{M}$ denotes concatenations.

D. KEY POINT DETECTION

In terms of geometric perception, each point's roles from different point cloud datasets are unequal. Moreover, corners and edges are more sensitive to geometric perception than flat areas; and automatic emphasis on these key points is critical for improving the quality of features acquired, as shown in Fig. 4.

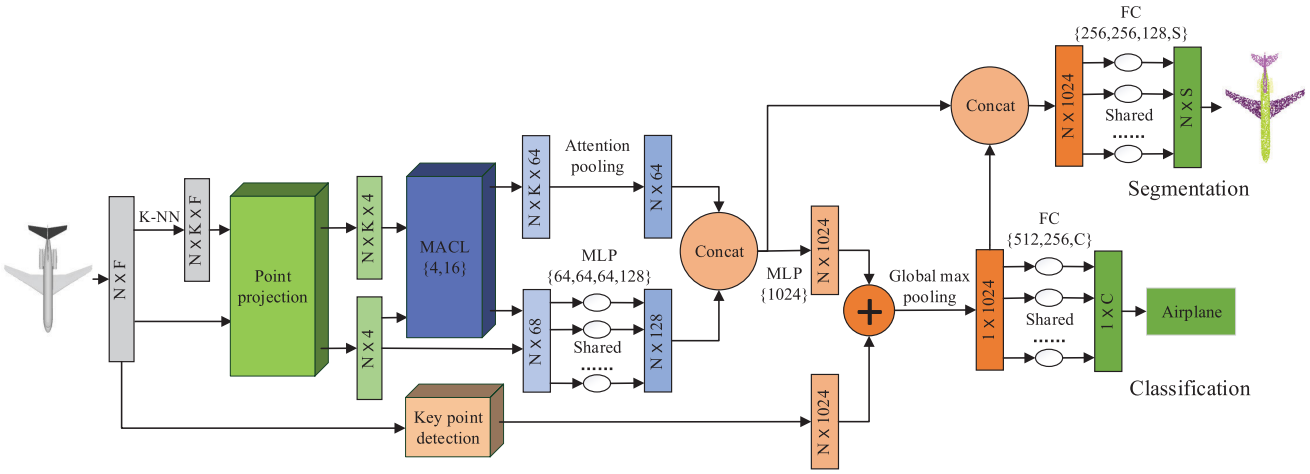


FIGURE 5. Proposed approach architecture for strictly rotation-invariant point cloud classification and segmentation.

In order to find out the corner regions of the point cloud, point normals [26] are chosen in this paper to reflect the shape characteristics. Since the shape information at the corner and edge varies greatly, the normal line has changed. A response can be specified for each point through the normal line transaction of the neighborhood, given by Equation (17):

$$D_r = sum_{i \in N(r)} sin(x_i, x_r) \tag{17}$$

where x_i denotes the general points, x_r denotes the points of the neighborhood, and D_r represents response points. We could utilize this method to detect the key points because the high response points D_r are in the margins, especially the corner areas, with an apparent change of the normal vectors. Along with this key point detection process, the calculated response will be integrated into the global representation of point clouds.

E. MODEL

The overall framework of Multi-head Attentional Point Cloud Classification and Segmentation Using Strictly Rotation-invariant Representations is presented in Fig. 5. Generally, we input the point cloud of $N \times F$ and divide them into two branches. On the one hand, we capture rotation-invariant representation and transfer them and neighborhoods to $N \times K \times 4$ dimensions through KNN and point projection. By applying MACL {4, 16} to point clouds, attentional characteristics are detected from local and neighbors. Then, we construct the attention pooling layer and Advanced-Dense-MLP to mine features, which are concatenated as the global feature. On the other hand, the key point descriptor module is used to find out features of crucial points, forming the final global feature by an addition operation. Finally, fully connected (FC) layers with shared wights classify the point clouds into 40 categories.

In the part segmentation task, the MACL layer obtains the specific local category of each point's semantic label. The attention pooling layers are used for local tag generation and composed the global feature by connecting to intermediate layers. In addition, FC {512, 256, C} and FC {256, 256, 128, S} denotes the fully-connected layers, including 512, 256, 128, C, S for the number of neurons. C and S equal to 40 and 16, respectively. MACL {4, 16} represents the attentional coding layer with four heads and sixteen channels. For the dense blocks [3] adding to MLP, we have made improvements to avoid unnecessarily detailed feature information. As presented in Fig. 6, each dense block applied 2D convolution with 1×1 kernels, using 'concat' operation to concatenate multiple-channel features. We eliminate the connection of $N \times 68$ input data and the last block, decreasing the information reduction and large-scale parameters.

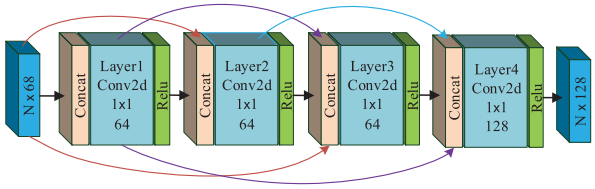


FIGURE 6. Advanced densely connected blocks without the connection between the fourth layer and the beginning.

IV. EXPERIMENTS AND DISCUSSIONS

A. DATASETS AND PARAMETERS

For the classification task, we have conducted several experiments on the opening dataset ModelNet40 and Modelnet10 [27] for three-dimensional pattern recognition, provided by Princeton University. ModelNet40 contains 12311 CAD models from 40 categories, of which 9843 models are used for training and 2468 for testing. Whereas, ModelNet10 consists of 10 classes, 4900 CAD models divided into two parts: 3991 for training our model, and the others for testing.

Part segmentation experiments are based on ShapeNet Part [42] dataset, composed of 16881 point cloud models with 16 categories. Objects are segmented into 50 parts without overlapping in different categories. Every point in the model

has one specific semantic tag, contains no more than five parts. ShapeNet Part is split into two parts: 14006 models for training, 2875 for testing. The experiments sample 2048 points from each model and utilize mIoU for evaluation.

To be specific, our rotated testing datasets come from the multiplication of the Rotation matrix and the original testing points concerning arbitrary rotation angles shown in Section III. All the experiments are conducted in a deep learning environment based on the Ubuntu operating system and Cuda 8.0.61. Table 1 indicates the learning framework, corresponding environments, and partial parameters during training.

TABLE 1. Experiment configurations.

Environment Configurations		Model Parameters	
Name	Configuration	Name	Value
CPU	Core i7-6850k	Batch size	32
GPU	TITAN Xp	Number point	1024
RAM	12G	Max epoch	250
Operation system	Ubuntu 14.04	Optimizer	Adam [43]
Language	Python 3.6	Learning rate	0.001
Learning framework	Tensorflow-gpu 1.9.0	Momentum	0.9

B. CLASSIFICATION RESULTS

To verify our work's performance, we have conducted some comparison experiments with other state-of-the-art models, under the same conditions, which are shown in Table 2. It can prove that the study in this paper significantly outperforms other methods in the case of random rotation testing. All experiments are applied on ModelNet40 [27], regarding 3D model recognition accuracy as evaluation criterion. We train the model using the original data set with a fixed direction. The test process is divided into two groups: one group uses the original data set with a fixed direction, and the other group tests the data set after arbitrary rotation transformation. NR/NR means we input unrotated point clouds for training and testing. NR/AR represents no rotation reinforcement training but random rotation testing. Besides, drop by indicates the decay rate of rotation experiments. Except for the precision shown in Table 2, our classification training experiment costs 8.65 hours and 8.39 Gb memory to compute and save a 2.2 MB model.

The experimental comparison covers the traditional point cloud classification methods, spherical harmonic convolution, and other strict rotation-invariance methods. The accuracy of the traditional methods have a great drop in the arbitrary rotation test, showing poor robustness to point cloud rotation, such as PointNet [11], PointNet++ [31], and DGCNN [12]. Networks based on spherical convolutions have good robustness to rotation, with the accuracy dropped by less than 10%, such as PRIN [18], Spherical CNN [44], SFCNN [20], SPHNet [19], and RICNN [22]. However, the discretized sphere cannot guarantee complete symmetry or rotation-invariance due to global directionality.

TABLE 2. Result Comparison on ModelNet40 [27].

Method	NR/NR (%)	NR/AR (%)	drop by (%)
PointNet [11]	88.45	12.47	75.98
PointNet++ [31]	89.42	21.35	68.70
DGCNN [12]	91.77	20.73	71.04
Spherical CNN [44]	88.90	78.50	10.40
PRIN [18]	80.13	70.35	9.78
SFCNN [20]	**92.31**	85.34	6.97
SPHNet [19]	87.70	86.62	1.08
RICNN [22]	86.50	86.35	0.50
REQNN [23]	83.02	83.02	0
SRINet [26]	87.01	87.01	0
ClusterNet [25]	87.10	87.10	0
Ours	87.50	**87.50**	0

TABLE 3. Result comparison on ModelNet10 [27].

Method	NR/NR (%)	NR/AR (%)	drop by (%)
PointNet [11]	**91.47**	27.15	74.32
Spherical CNN [44]	90.25	82.13	7.12
PRIN [18]	83.26	75.42	7.84
RICNN [22]	88.92	88.61	0.31
SRINet [26]	90.08	90.08	0
Ours	90.63	**90.63**	0

Similar to us, REQNN [23], SRINet [26] and ClusterNet [25] can realize strictly rotation-invariant classification, with 0 drop rate. However, the accuracy of our method has surpassed them by 4.48%, 0.49%, and 0.40%, respectively. The point projection module can reconstruct the points' coordinates and obtain the strictly rotation-invariant representations for various tasks. Our method can effectively discover the local and global features through the proposed multi-head attentional convolution structure. In addition, a key point detection is added to extract the key feature points of the point cloud, which makes up for the perception of the overall geometric structure.

To prove our excellent classification performance, we analyze all the recognition accuracy in every category with SRINet [26], and RICNN [22], as demonstrated by Fig. 7. Here, we arrange 40 different classes on the horizontal axis and each class's accuracy on the vertical axis. In general, we enhance the accuracy rate of every class compared with SRINet [26], and RICNN [22], especially the categories where the global geometric structure are similar to each other, but the difference in detailed local information, such as a tent, stairs, flowerpot, etc. We have employed the attention coding method and emphasized the key points to enhance the recognition of vital local features. Therefore, our framework can distinguish the classification information of confusing 3D models effectively.

Based on the above experiments, more comparative tests on ModelNet 10 are conducted to verify the reliability. In Table 3, our research approach is proved higher classification accuracy at 90.63% on rotated testing datasets than other frameworks.

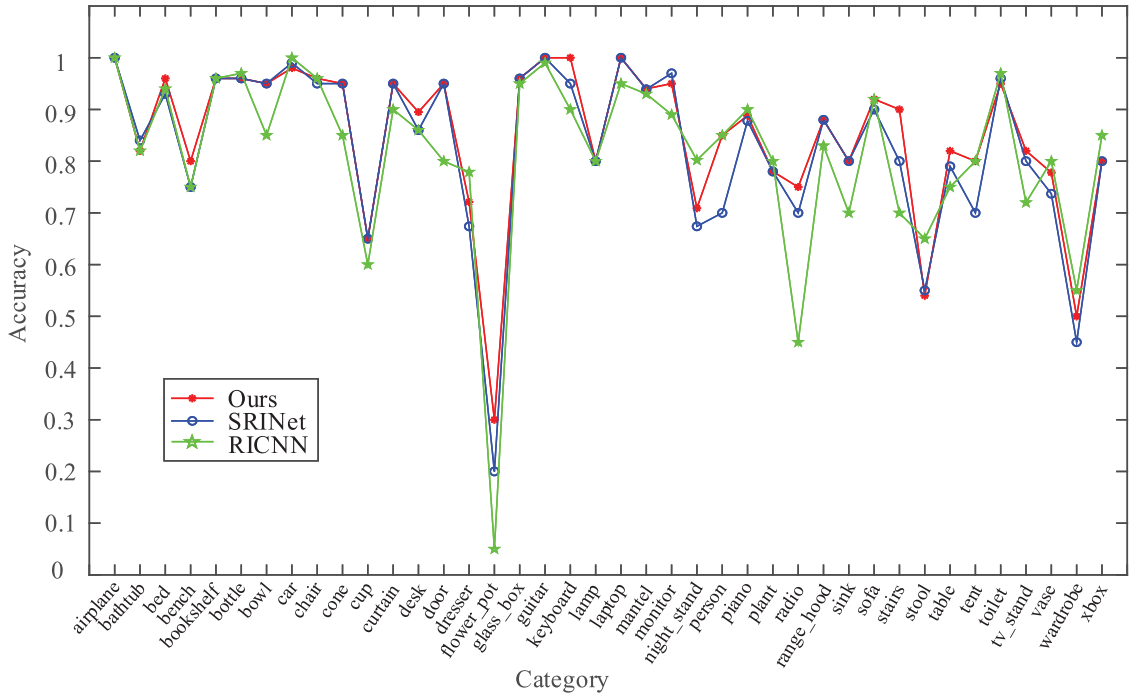


FIGURE 7. The classification performance of each category on ModelNet40.

TABLE 4. Part segmentation NR/NR testing results on ShapeNet Part [42] dataset.

Method	mIoU(%)	air plane	bag	cap	car	chair	ear phone	guitar	knife	lamp	laptop	motor bike	mug	pistol	rocket	skate board	table
PointNet [11]	83.37	82.81	76.31	77.85	74.50	89.50	73.38	91.16	85.96	79.45	95.19	65.56	93.05	81.21	57.89	74.16	80.14
PointNet++ [31]	85.15	82.41	79.08	87.65	77.26	90.76	71.83	91.04	85.95	83.72	95.37	71.63	94.16	81.34	58.76	76.41	82.67
DGCNN [12]	85.21	84.07	83.43	86.68	77.76	90.62	74.53	90.08	86.48	82.86	95.62	66.37	94.91	81.16	63.57	74.53	82.62
PRIN [18]	71.35	70.19	71.05	71.74	45.42	77.17	52.23	88.62	78.83	70.25	73.99	48.54	83.93	59.03	48.94	53.19	70.00
RICNN [22]	73.64	78.92	72.59	72.34	68.85	87.21	68.55	89.58	79.10	76.49	73.65	57.61	89.55	71.19	48.23	66.20	77.13
SRINet [26]	73.46	72.54	73.89	67.97	48.64	79.41	61.10	87.98	78.45	75.43	74.59	49.63	82.30	65.38	46.65	52.98	72.40
Ours	75.15	75.05	74.50	73.97	54.46	81.28	66.45	88.35	84.42	77.63	78.16	51.33	87.27	66.60	55.03	62.12	71.72

TABLE 5. Part segmentation NR/AR testing results on the ShapeNet Part [42] dataset.

Method	mIoU(%)	air plane	bag	cap	car	chair	ear phone	guitar	knife	lamp	laptop	motor bike	mug	pistol	rocket	skate board	table
PointNet [11]	32.73	20.28	48.27	44.33	21.58	27.04	15.63	34.72	34.64	42.10	35.74	22.35	48.93	29.72	27.63	32.27	29.82
PointNet++ [31]	37.21	22.19	49.03	40.12	23.13	43.03	12.57	38.51	40.73	46.37	41.75	21.35	53.67	44.23	25.73	38.91	36.53
DGCNN [12]	42.45	26.75	50.28	38.63	24.15	30.12	28.15	38.06	47.92	42.29	34.86	20.51	49.23	25.85	26.88	26.95	29.67
PRIN [18]	58.82	51.13	56.07	63.26	40.78	57.32	54.39	57.85	72.07	42.29	34.86	24.93	70.43	45.89	49.68	56.92	64.47
RICNN [22]	73.51	78.91	72.61	72.34	68.85	86.89	68.55	89.54	79.06	76.50	73.65	57.61	89.55	71.19	48.16	65.69	77.13
SRINet [26]	73.46	72.54	73.89	67.97	48.64	79.41	61.10	87.98	78.45	75.43	74.59	49.63	82.30	65.38	46.65	52.98	72.40
Ours	**75.15**	75.05	**74.50**	**73.97**	54.46	81.28	66.45	88.35	**84.42**	**77.63**	**78.16**	51.33	87.27	66.60	**55.03**	62.12	71.72

C. SEGMENTATION PERFORMANCE

Like the classification task, we conduct unrotated training and testing (NR/NR) and normal training but rotated testing (NR/AR) experiments, shown in Table 4 and Table 5. By training the network with original data and two sets of non-rotated and rotated datasets for testing, it is proved that the network proposed in this paper has excellent part segmentation ability and robustness to the arbitrary rotation. The segmentation task (with the model size of 4.3 MB) costs 16.28 hours, 8.36 Gb memory in the training process.

From Table 4 we present, the existing methodologies can achieve high accuracy in conventional segmentation testing, such as PointNet [11], PointNet++ [31], and DGCNN [12]. However, as shown in Table 5, their performances sharply decline when processing rotated point sets. Although PRIN [18] is robust to rotation, it cannot achieve

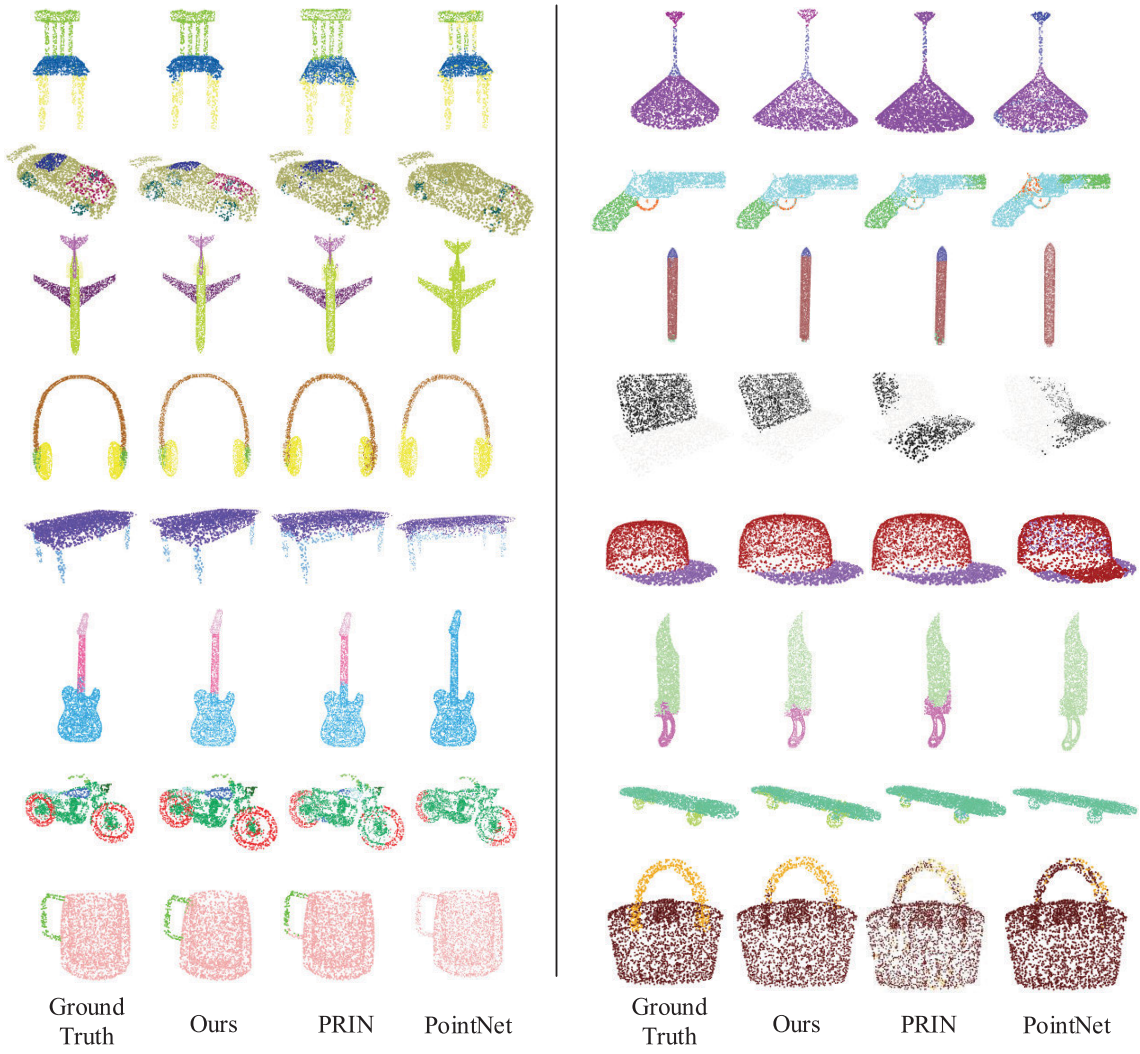


FIGURE 8. Part segmentation NR/AR testing result visualization for 16 categories in the ShapeNet Part [42] dataset.

strict rotation invariance, and the segmentation accuracy is low, indicating that spherical convolution will be disturbed by rotation transformation. In the experiment of segmenting the rotating point cloud, this method achieves strict rotation invariance and obtains a good segmentation result. SRINet [26] and RICNN [22] are also rotation-invariant networks. In the comparison between our method and them, 6 types of point clouds achieve the best accuracy, and our mIoU value is the largest.

We visualize the segmentation results in Fig. 8 to verify the superiority of our work. In PointNet results, partial rotated point cloud could not be recognized and segmented because of its low robustness to rotation transformation. PRIN [18] shows a little robustness to rotation but low segmentation precision and non-rotation invariance, indicating that spherical convolution will be disturbed by rotation transformation. Part of the segmentation disorder indicates that it has a poor perception of the overall structure of the point sets. Our proposed method achieves effective segmentation for rotation tests, showing a good perception of the geometric structure of the point cloud.

D. ABLATION STUDY

Inspired by [45], dynamical behavior, especially stability, plays an essential role in learning algorithms. We have conducted ablation experiments to prove each module's availability and compare different parameter combinations in our proposed neural network.

1) OPTIMIZATION OF MLP

To enhance scientificity and reasonability, first, we test different combinations of the advanced MLP with MACL, as presented in Table 6. From the mathematical point of view, [47] proposed an optimization methodology through parallel computing and swarm intelligence. Regrading to the numerical optimization method proposed in [48], we analyze the optimization performance to MLP. From the results, we could find the enhanced dense connected convolutional blocks

TABLE 6. Analysis of different optimizations to MLP.

Combination	Mean Accuracy(%)	Overall Accuracy(%)
MACL-MLP	83.21	87.11
MACL-Res [46] -MLP	83.85	87.33
MACL-Dense [3] -MLP	83.42	87.25
MACL-Advanced -Dense-MLP (ours)	**84.00**	**87.50**

perform better than other combinations. Because adding connected blocks can decrease the possibility of over-fitting and gradient vanishing and reinforce feature propagation. Our combination with dense blocks has another advantage of low information redundancy and learning efficiency enhancement.

2) INFLUENCE OF MULTI-HEAD CHANNELS

Referring to [45], robust performance analysis from various parameters is essential to neural networks. Especially for the multi-head structure in this paper, we explore some experiments to detect the effect of different head numbers and encoding channels on the recognition performance. Fig. 9 indicates the results, where the horizontal axis denotes the number of channels, the vertical axis for the accuracy rate, and $1, 2, 3, 4$ for the head of numbers. It can be found that proper increasing of head numbers and channels cause a positive effect on accuracy. We utilize more heads and channels to detect enough features for classification. According to the chart, there is an optimal portfolio of four heads and sixteen channels. Not adding more is always better than previous; redundant information and high complicity will reduce the property.

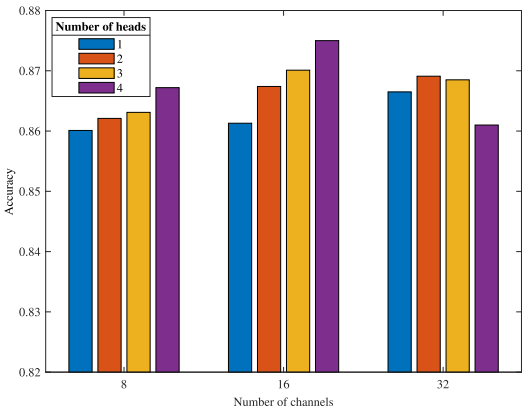


FIGURE 9. Analysis of different channels and head numbers.

3) ANALYSIS OF K VALUES

We preprocess the KNN sampling method before point projection, where K represents the receptive field in local regions. The proper size of receptive fields is beneficial to local feature detection. We present other experiments to prove the influence of different local sampling point numbers (the value of K), shown in Table 7. The number of K will cause an effect on local feature extraction, detailed information correct detection, then affect the overall performance. Although we cannot test all the possibilities, the results indicate that $K = 25$ enjoys the best local receptive field and accuracy, decreasing when K is greater.

TABLE 7. Performance on different K values.

KNN point number	Mean Accuracy(%)	Overall Accuracy(%)
16	83.08	86.56
20	83.14	86.64
25	**84.00**	**87.50**
30	82.89	86.81
32	82.05	86.48

4) EVALUATION OF ATTENTION CODING AND KEY POINT DETECTION

This section is aiming to verify the effects of the attention encoder and key point detection. They are related to extracting geometric features and structures. Just as its name implies, the attention encoder mines the attentional point cloud features, deeper than general. The key point detection method gives the response to the features of crucial points we selected. These responses will be combined with the global feature to the final information. We enhance information effectiveness through this kind of selection and attention extraction, verified in Table 8. Without them, the overall accuracy equals 84.31%, which could be increased by 3.19% when adding them to the model.

TABLE 8. Influence of using key point detection method and attention coding.

Attention coding	Key point detection	Mean Accuracy(%)	Overall Accuracy(%)
×	×	79.67	84.31
×	√	81.29	85.47
√	×	82.06	85.55
√	√	**84.00**	**87.50**

E. REMARKS ON OUR PROPOSED METHOD

1) ADVANTAGES

Our framework can effectively classify and segment the point cloud without being disturbed by the rotation transformation, potentially applying in the unseen direction objects of the real world. From the experimental results, our work can perceive the point clouds' overall geometric structure and achieve better accuracy than most state-of-the-art methods. From the ablation study, the multi-head structure we employed contributes to model robustness. The rotation-invariant features through point projection enjoy simple structure, easy transformation, and strong embeddedness.

2) LIMITATIONS

This method has not been applied to large-scale engineering problems. Except for the rotated 3D point cloud study,

high-dimensional problems would be an interesting extension to this work. We can combine other dimension reduction method with our work to cope with this kind of complicated scenarios. However, the complicity and high-dimensional point structure should be taken into consideration. In future studies, more practical, accurate, and less-complicity deep learning models will be explored for 3D point processing and computer perception.

V. CONCLUSION AND FUTURE WORK

1) The neural network proposed in this paper realizes effective recognition of the rotated point clouds. First of all, we reconstruct the point cloud coordinators from the origin and get the strictly rotation-invariant representations through point projection. Next, our proposed MACL module comprises the attentional convolution layer (ACL) and a multi-head structure, which can detect in-depth features of the point sets. The overall framework includes the model of attention pooling and Advanced-Dense-MLP to enhance local information relations. At last, we distribute different responses to every point according to the key point descriptor, emphasizing the geometric structure. From the classification experiment results on ModelNet10 and ModelNet40, the accuracy of our work is better than most mainstream algorithms, with strong robustness.
2) From the rotation testing on the ShapeNet Part dataset, our algorithm can also achieve strictly rotation-invariant partial segmentation, better precision than the state-of-the-art. Future studies will focus on improving 3D point cloud classification, segmentation, and other complicated tasks in real scenarios, such as reconstruction and complement. Making an application of our work and strictly rotation-invariant features to other data types or fields is a meaningful topic.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1097–1105.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[3] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4700–4708.

[4] Y. Cao, Y. Cao, Z. Guo, T. Huang, and S. Wen, "Global exponential synchronization of delayed memristive neural networks with reaction–diffusion terms," *Neural Netw.*, vol. 123, pp. 70–81, Mar. 2020.

[5] S. Wang, Y. Cao, T. Huang, Y. Chen, and S. Wen, "Event-triggered distributed control for synchronization of multiple memristive neural networks under cyber-physical attacks," *Inf. Sci.*, vol. 518, pp. 361–375, May 2020.

[6] Y. Wang, Y. Cao, Z. Guo, T. Huang, and S. Wen, "Event-based sliding-mode synchronization of delayed memristive neural networks via continuous/periodic sampling algorithm," *Appl. Math. Comput.*, vol. 383, Oct. 2020, Art. no. 125379.

[7] S. Wang, Y. Cao, Z. Guo, Z. Yan, and T. Huang, "Periodic event-triggered synchronization of multiple memristive neural networks with switching topologies and parameter mismatch," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 427–437, Jan. 2021.

[8] D. Hong, X. Wu, P. Ghamisi, J. Chanussot, N. Yokoya, and X. X. Zhu, "Invariant attribute profiles: A spatial-frequency joint feature extractor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 3791–3808, Jun. 2020.

[9] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, early access, Aug. 18, 2020, doi: 10.1109/TGRS.2020.3015157.

[10] D. Hong, L. Gao, N. Yokoya, J. Yao, J. Chanussot, Q. Du, and B. Zhang, "More diverse means better: Multimodal deep learning meets remote-sensing imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 4340–4354, May 2021.

[11] 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.*, Jul. 2017, pp. 652–660.

[12] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Nov. 2019.

[13] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 6411–6420.

[14] X. Liu, Z. Han, Y.-S. Liu, and M. Zwicker, "Point2sequence: Learning the shape representation of 3D point clouds with an attention-based sequence to sequence network," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8778–8785.

[15] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 9621–9630.

[16] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 108–117, Sep. 2006.

[17] G. P. Meyer, A. Laddha, E. Kee, C. Vallespi-Gonzalez, and C. K. Wellington, "LaserNet: An efficient probabilistic 3D object detector for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 12677–12686.

[18] Y. You, Y. Lou, Q. Liu, Y.-W. Tai, L. Ma, C. Lu, and W. Wang, "Pointwise rotation-invariant network with adaptive sampling and 3D spherical voxel convolution," 2018, *arXiv:1811.09361*. [Online]. Available: http://arxiv.org/abs/1811.09361

[19] A. Poulenard, M.-J. Rakotosaona, Y. Ponty, and M. Ovsjanikov, "Effective rotation-invariant point CNN with spherical harmonics kernels," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2019, pp. 47–56.

[20] Y. Rao, J. Lu, and J. Zhou, "Spherical fractal convolutional neural networks for point cloud recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 452–460.

[21] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling, "Spherical CNNs," 2018, *arXiv:1801.10130*. [Online]. Available: http://arxiv.org/abs/1801.10130

[22] Z. Zhang, B.-S. Hua, D. W. Rosen, and S.-K. Yeung, "Rotation invariant convolutions for 3D point clouds deep learning," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2019, pp. 204–213.

[23] W. Shen, B. Zhang, S. Huang, Z. Wei, and Q. Zhang, "3D-rotation-equivariant quaternion neural networks," 2019, *arXiv:1911.09040*. [Online]. Available: http://arxiv.org/abs/1911.09040

[24] T. Furuya, X. Hang, R. Ohbuchi, and J. Yao, "Convolution on rotation-invariant and multi-scale feature graph for 3D point set segmentation," *IEEE Access*, vol. 8, pp. 140250–140260, 2020.

[25] C. Chen, G. Li, R. Xu, T. Chen, M. Wang, and L. Lin, "ClusterNet: Deep hierarchical cluster network with rigorously rotation-invariant representation for point cloud analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 4994–5002.

[26] X. Sun, Z. Lian, and J. Xiao, "SRINet: Learning strictly rotation-invariant representations for point cloud classification and segmentation," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 980–988.

[27] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1912–1920.

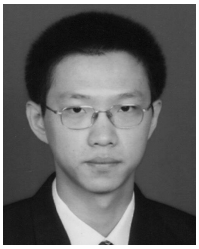
[28] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 922–928.

[29] H. Huang, E. Kalogerakis, S. Chaudhuri, D. Ceylan, V. G. Kim, and E. Yumer, "Learning local shape descriptors from part correspondences with multiview convolutional networks," *ACM Trans. Graph.*, vol. 37, no. 1, pp. 1–14, Jan. 2018.
[30] 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.*, Dec. 2015, pp. 945–953.
[31] 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.
[32] C. Chen, L. Z. Fragonara, and A. Tsourdos, "GAPNet: Graph attention based point neural network for exploiting local feature of point cloud," 2019, *arXiv:1905.08705*. [Online]. Available: http://arxiv.org/abs/1905.08705
[33] J. Li, B. M. Chen, and G. H. Lee, "SO-Net: Self-organizing network for point cloud analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9397–9406.
[34] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on *X*-transformed points," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 820–830.
[35] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz, "SPLATNet: Sparse lattice networks for point cloud processing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2530–2539.
[36] Z. Zhang, B.-S. Hua, and S.-K. Yeung, "ShellNet: Efficient point cloud convolutional neural networks using concentric shells statistics," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 1607–1616.
[37] D. Müllner, "Modern hierarchical, agglomerative clustering algorithms," 2011, *arXiv:1109.2378*. [Online]. Available: http://arxiv.org/abs/1109.2378
[38] P. Chmelar, L. Beran, and N. Kudriavtseva, "Projection of point cloud for basic object detection," in *Proc. ELMAR*, Sep. 2014, pp. 1–4.
[39] L. Euler, "General formulas for the translation of arbitrary rigid bodies," *Novi Commentarii Academiae Scientiarum Petropolitanae*, vol. 20, no. 1776, pp. 189–207, 1776.
[40] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*. [Online]. Available: http://arxiv.org/abs/1710.10903
[41] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification and regression," in *Proc. Adv. Neural Inf. Process. Syst.*, 1996, pp. 409–415.
[42] L. Yi, V. G. Kim, D. Ceylan, I. C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas, "A scalable active framework for region annotation in 3D shape collections," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 210.1–210.12, 2016.
[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[44] C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis, "Learning so (3) equivariant representations with spherical CNNs," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 52–68.
[45] H. Wang, G. Wei, S. Wen, and T. Huang, "Impulsive disturbance on stability analysis of delayed quaternion-valued neural networks," *Appl. Math. Comput.*, vol. 390, Feb. 2021, Art. no. 125680.
[46] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-V4, inception-ResNet and the impact of residual connections on learning," 2016, *arXiv:1602.07261*. [Online]. Available: http://arxiv.org/abs/1602.07261
[47] W.-J. Niu, Z.-K. Feng, B.-F. Feng, Y.-S. Xu, and Y.-W. Min, "Parallel computing and swarm intelligence based artificial intelligence model for multi-step-ahead hydrological time series prediction," *Sustain. Cities Soc.*, vol. 66, Mar. 2021, Art. no. 102686.
[48] Z.-K. Feng, W.-J. Niu, and S. Liu, "Cooperation search algorithm: A novel Metaheuristic evolutionary intelligence algorithm for numerical optimization and engineering optimization problems," *Appl. Soft Comput.*, vol. 98, Jan. 2021, Art. no. 106734.

ZHIYONG TAO received the Ph.D. degree from Liaoning Technical University. He worked as a Visiting Scholar with the Clausthal University of Technology, Germany, in 2006. Since 2008, he has been working as an Associate Professor and a Master Tutor with Liaoning Technical University. He designed the proposed model, analyzed the result of experiments, and wrote the original article. His main research interests include the Internet of Things, machine learning, and biometric recognition, with rich project and engineering experience.

YIXIN ZHU received the bachelor's degree in electronic information engineering from Liaoning Technical University, where he is currently pursuing the master's degree. He designed the proposed model, conducted experiments, and wrote the original article. His main research interests include machine learning, pattern recognition, and 3D point cloud processing.

TONG WEI received the bachelor's degree in communication engineering from Liaoning Technical University. She is currently pursuing the master's degree with the Faculty of Informatics, Eötvös Loránd University. She helped to complete the extension experiments and the format of the article. Her main research interests include machine learning, computer science, and 3D computer vision.



SEN LIN received the bachelor's and master's degrees from Liaoning Technical University, in 2003 and 2006, respectively, and the Ph.D. degree from the Shenyang University of Technology, in 2013. He is currently an Associate Professor and a Master Supervisor with Shenyang Ligong University. He is also a Postdoctoral Researcher with the Shenyang Institute of Automation, Chinese Academy of Sciences. He helped the revision of the original article. His main research interests include image processing, machine vision, and biometric recognition.