IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON DEEP LEARNING FOR INTERNET OF THINGS

Received August 30, 2021, accepted September 30, 2021, date of publication October 13, 2021, date of current version October 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3119622

A Geometry Feature Aggregation Method for Point Cloud Classification and Segmentation

YONG WANG, CHENKE YUE, AND XINTONG TANG
School of Artificial Intelligence, Chongqing University of Technology, Chongqing 400054, China
Corresponding author: Chenke Yue (yueck0928@163.com)

This work was supported by the Natural Science Foundation of China under Grant 61502065 and Grant 61976158.

ABSTRACT In spite of the good performance of convolutional neural network (CNN) and graph neural network (GNN) in 3D point cloud classification and segmentation at present, to aggregate local information of point clouds and improve the robustness of geometric transformation are still challenging problems. In order to tackle the problems, we propose Geometry Feature Aggregation Network (GFA-Net), which can effectively learn the context information of each point to aggregate local information, so as to enhance the robustness of rotation and translation. Compared with the current popular method GNN that convolves on nearby points in Euclidean space, GFA-Net can better aggregate the geometric features around the points. GFA-Net uses the Laplacian feature mapping to reduce dimensions, and aggregates the nearest neighbor features in the space after dimensionality reduction, and fuses them with the nearest neighbor features of Euclidean space, so as to better obtain the geometric features of each point. Then, points are grouped with geometric features, so that nearby points are insensitive to geometric transformations such as rotation and translation. This method allows GFA-Net to better obtain holistic geometry features, such as symmetry. In addition, we use attention mechanism instead of pooling, so that important neighborhood information can be learned automatically and information loss can be reduced. We conduct extensive experiments on public datasets ModelNet40 and ShapeNet Part. The experimental results show that GFA-Net achieves very good performance, which is very close to the current state-of-the-art methods, and GFA-Net has better robustness.

INDEX TERMS Geometry feature, Laplacian feature, Euclidean space, attention mechanism, point cloud.

I. INTRODUCTION

At present, with the development of computer vision and artificial intelligence, point cloud classification and segmentation has become a very challenging and important problem in 3D vision. As robot control [1], automatic driving, remote sensing [2], [3], reconstruction of 3-D buildings [4] and other fields have the need to interact with the real scene. For the past years, tremendous progress in the automatic classification of ALS point clouds has been achieved in the community of remote sensing and photogrammetry [5].

At present, deep learning and convolutional neural network (CNN) show great brilliance in the field of 2D images [6], [7]. 3D point cloud is non-ordered, that is, the change of point order will not affect its meaning, while deep neural networks require input data with regular structure. Moreover, 3D point cloud is irregular, so it is not easy to apply deep learning to 3D point cloud. In this paper, we consider to handle point cloud classification and segmentation. The traditional method to solve such problems is to extract the geometric features of point clouds with hand-made features [8], [9]. In recent years, deep learning has made great progress in LiDAR point cloud analysis, and many deep learning-based point cloud classification and segmentation methods have been proposed [10], [11]. The deep learning method to handle such problems is converting 3D point cloud data into a voxel form with regular shape [12], [13], and then using 3D CNN for feature learning to process point cloud data. However, such conversion consumes expensive computing and memory costs, which increases network overhead and is not conducive to the use of large-scale point cloud scenarios. Then, deep neural network designed to process the raw point cloud data directly is presented. PointNet [14] is the pioneer of neural network that directly processes the raw point cloud data. By learning the information coding of each point and aggregating the features of each point into the global features, PointNet [14] uses the MaxPooling function to realize the permutation invariance of the points, but this

The associate editor coordinating the review of this manuscript and approving it for publication was Ziyan Wu.

lose the local information of each point. In order to tackle the problem that PointNet [14] fails to extract local information, PointNet++ [15] is proposed. PointNet++ [15] adopts the strategy of stratified sampling and uses neighborhood balls to get subregions. Then, PointNet [14] is used to extract local features of different scales of the network.

In order to enhance the learning ability of each point to local features, PointWeb [16] calculates the influence among all points in the local region by considering the interaction among all points in the local region to obtain features. So a Adaptive Feature Adjustment (AFA) module is designed in PointWeb [16], which connects all points in the region and finally forms a local fully connected network. Recent work has focused on designing some unique convolution operations for irregular 3D point cloud. ConvPoint [17] proposed by Boulch in 2020 utilizes the concept of convolution. Multilayer perceptron (MLP) is used to learn the continuous weight of adjacent points in each point, and then the density of each point is calculated to solve the problem of uneven sampling of point cloud. RS-CNN [18], a convolutional network based on geometric relations, proposed a novel convolutional operator RS-Conv that learns from relations. RS-Conv extracts the topological constraint relationship from the point cloud, so the weight of the convolutional network is also constrained by the topological relation in learning. This method extends the convolutional networks based on ordered information to the convolutional networks which adapt to unorganized information, thus improving the shape perception and robustness to a great extent. Similarly, to aggregate information in Euclidean space, DGCNN [19] proposed a new convolution method, EdgeConv. EdgeConv uses K-Nearest Neighbors (KNN) algorithm to construct directed graph of point clouds. Then, the adjacent points in Euclidean space are recalculated with the directed graph of point cloud, and the new graph structure is passed to the next layer for processing, so as to realize the dynamic graph structure. However, in Euclidean space, DGCNN [19] only considers the nearby neighborhood points and fails to consider the points that may have the same geometric structure at the far end. Point2Node [20] proposed a learning network with a high-dimensional node graph model, which can fuse self, local and non-local correlation information between nodes on 3D point cloud, and designed adaptive aggregation features information of Gate Mechanism at channel level.

The above methods only consider the local features of each point extracted from the Euclidean space. However, the geometry of many objects may be symmetric, with points moving away from each other. If only the local structure of adjacent points is considered, the features of points with the same geometric structure cannot be obtained, and even the overall geometric structure will be lost. In order to capture points with similar geometrical structure and share the information of points with similar geometrical structure, the features of points cannot be extracted only from Euclidean space.

Inspired by the above work, we propose a Geometric Feature Aggregation Network (GFA-Net). This network not only gathers the eigeninformation of points in Euclidean space, but also the eigeninformation of points in eigenspace after dimensionality reduction of Laplace matrix. The Laplace matrix is calculated by constructing the fully connected point cloud graph. Then, the geometric features of each point can be learned through Laplace eigenmapping, so that points with similar geometric characteristics can exchange information. It makes up for the shortage of remote points with similar characteristics but unable to exchange information. We fuse the feature map calculated in Euclidean space with the feature map obtained in Laplacian eigendimension reduction space. This step takes into account not only nearby points but also points with similar geometrical structures. We also improve the pooling operation. For the information of local points, instead of simple pooling operation, the weight of local information carried by each point is calculated by attention mechanism, that is, the degree of influence on the central point is used for fusion calculation. Different from other methods that mostly deal with point cloud features in Euclidean space, our method deals with point features in Laplace feature space and makes full use of all point information to extract local information. Since we need to consider the relationship between all points, not only will the calculation be complicated, but also the learning rate will sometimes be reduced because of the redundancy of points.

We conduct extensive experiments to explore the effectiveness of GFA-Net for point cloud classification and segmentation. The experiments demonstrate that our method achieves the same performance as the current state-of-the-art methods on both ModelNet40 [36] and ShapeNet [37].

Overall, three key contributions of this paper can be summarized as follows:

- We propose a novel geometric feature aggregation Geometry Feature Aggregation (GFA) module, which effectively extracts the geometric features between points, and integrates with the features obtained from Euclidean space, so as to better extract the local and global geometric features.
- We prove that the neighborhood features obtained from the feature space of Laplacian dimension reduction are invariant in rotation and translation.
- Our GFA-Net achieves the state-of-the-art performances on datasets ModelNet40 and ShapeNet.

II. RELATED WORKS

A. METHOD BASED ON MULTIPLE VIEWS AND VOXELS

MVCNN [21] uses two-dimensional rendering images of multi-view 3D objects as training data, and recognized 3D objects based on CNN. Later, Qi *et al.* [22] propose a voxel-based CNN point cloud identification method, which raster the point cloud data to form the voxel structure, and then process it through 3D convolution operation, and introduce auxiliary training tasks to reduce overfitting. The auxiliary training task is to use partial subvolumes to predict the type of the object. Only partial subvolumes need to be collected

without any additional manipulation. The higher the resolution of the data, the better the effect is achieved by this method, so the effect is not obvious for the data with lower resolution. There are also some approaches to regularize the point cloud data by using the tree structure, for example, OctNet [23] makes use of the sparse input data and uses a group of unbalanced octree to stratify the space, and modifies and realizes the convolution operation, so as to adapt to the mixed grid octree data structure. The KD-Net [24] proposed by Roman Klokov *et al.* uses KD tree to divide the point cloud space, and its hierarchical structure is used as different feature forms.

B. POINT-BASED NETWORKS

In addition to PointNet [15] and PointNet++ [16], much of the new work involves extracting local features of each point by designing complex network modules. Slice pooling layer proposed by RSNet [25], it is to map the features of disordered point cloud data from x, y, z directions to the sequence of ordered feature vectors, and then use bidirectional RNN to update the features, so as to extract local correlation features. The computational complexity of extracting local correlation features is relatively small. SONet [26] simulates the spatial distribution of point cloud by constructing self-organizing map. Based on SOM, SO-Net is used to extract layered features from single point and SOM node. Finally, a feature vector is used to represent the input point cloud. There are also some methods to design a unique convolution for point cloud data [27], [28]. Pointwise [29] proposes a new point-by-point convolution method for 3D point cloud data. Pointwise [29] convolution is very similar to ordinary convolution. The difference is that point clouds are irregular, but they also use fixed-size convolution kernel to convolution. Pointwise convolution is the multiplication of the points in each small square of the convolution kernel with the weight to get the average value of each small square. After that, the average value of each small square is added to get the output of this layer. There are also some very clever convolution methods. FPConv [30] is a flattening convolution method, and maps each point and its neighborhood into a plane by attention mechanism, and uses a convolution method similar to 2D images to carry out sliding convolution on the flat point cloud data. PointCNN [31] uses X-Conv operator to re-encode and weight input points and features to make point cloud data into a canonical order, and then processes the reconstructed points through conventional convolution.

C. OTHER METHODS

Most studies focus on the calculation of adjacent regional points from Euclidean space, while GS-Net [32] finds that features of distant points with similar geometric features could not be obtained only in Euclidean space, so the Geometry Similarity Connection (GSC) module is designed. GSC module obtains the similar geometric features of distant points in the feature space, and then fuses them with the features of neighboring points obtained from Euclidean space

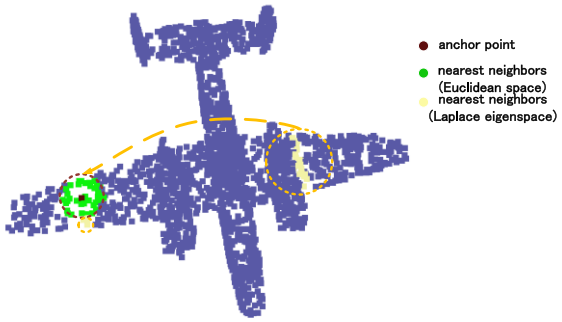

FIGURE 1. Visualization of feature points sets. Red is the anchor point. The selected adjacent points in Euclidean space are green, while the selected points in the dimension reduction feature space of Laplace mapping are yellow. Obviously, the yellow points have higher geometric similarity with the anchor point and are far away from the anchor point, which are not included by the adjacent points in Euclidean space.

to get local feature descriptors. Recently, there has also been some work using transformer for 3D point cloud [33], [34]. PCT [34] proposes a new transformer-based point cloud learning framework PCT, which avoids the disorder of point cloud data by using transformer's inherent order invariance and carries out feature learning through the attention mechanism.

III. METHOD

In this part, we introduce point cloud segmentation and classification network GFA-Net. GFA-Net is in three layer structure. Each layer consists of GFA module and attention mechanism. In each layer, geometric feature aggregation module is used to enrich local geometric information, and then attention mechanism is used to learn and select the extracted geometric features of surrounding points. The output features of attention mechanism are the input of the next GFA module. Finally the whole feature descriptor is obtained, sending it to the corresponding classification and segmentation network to segmentation and classification task.

A. GFA

Considering that in the Euclidean space to aggregate information of nearby points will lose the information of distant points with similar geometric features, while in the feature space, it is unstable to extract the geometric features of distant points, and sometimes feature chaos may occur. So we construct the fully connected graph of the points and compute their Laplace matrix. The Laplacian eigenspace is used to obtain the geometric features of distant points, and then the features of neighboring points in Euclidean space are integrated with them to better collect local features. As shown in Figure 1, the GFA module can not only identify adjacent points from Euclidean space, but also identify distant points with similar geometric features, such as symmetric geometric points.

In this section, we introduce Geometry Feature Aggregation (GFA) module. As we have presented in Figure 2 is its structure. We enter a three-dimensional point cloud

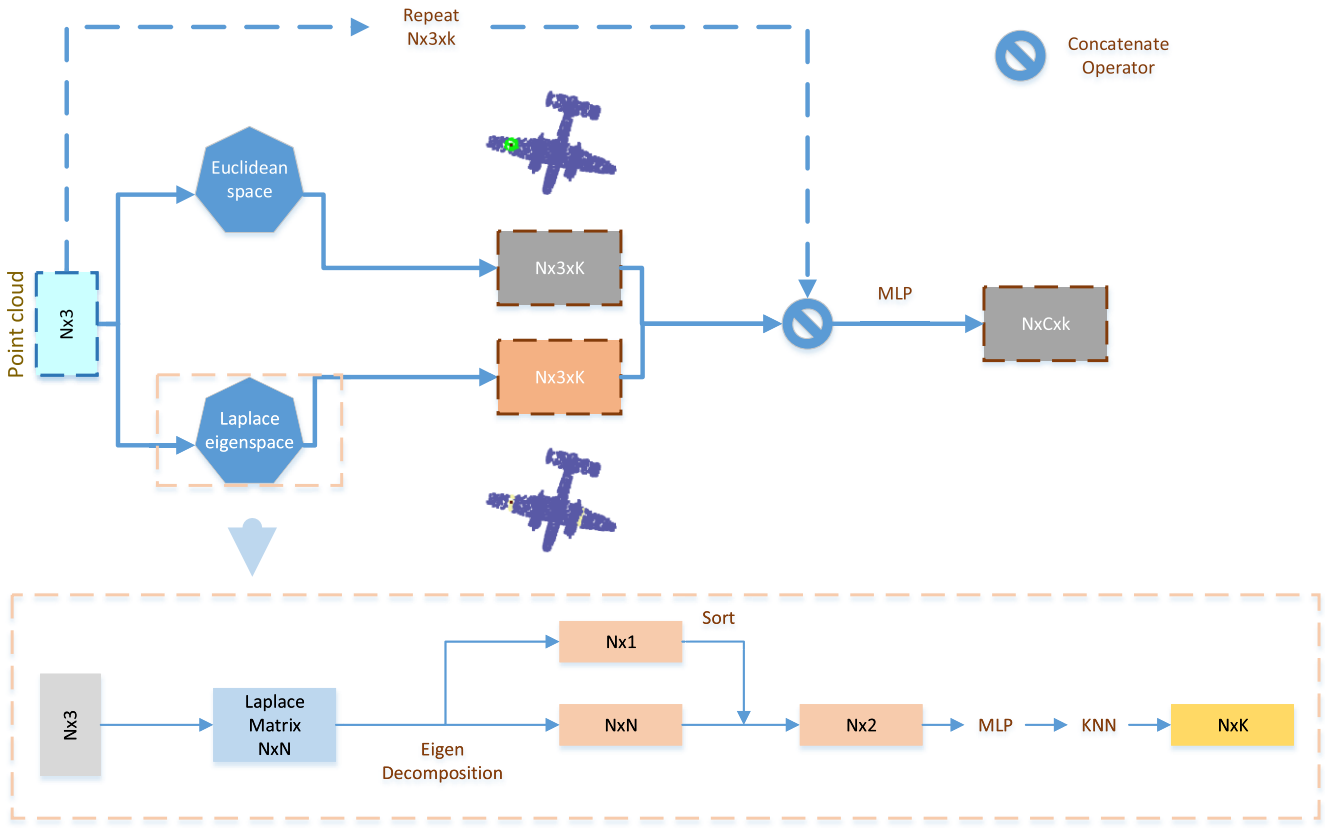

FIGURE 2. Geometry Feature Aggregation (GFA) module. The local features are collected by calculating the adjacent points in the Euclidean space and the Laplacian mapping space, and then the local features are recoded by using the Multi-layer Perceptron (MLP). In the Laplace mapping space, the full connection graph is constructed and the Laplace matrix is calculated. The Laplace matrix is decomposed into eigenvectors, sorted according to the size of eigenvalues, and the eigenvectors corresponding to the first two smallest eigenvalues are obtained. MLP is used to learn the selected eigenvectors.

with N points, denoted by $X = \{x_1, \ldots\ldots, x_N\}$. The three-dimensional features represent the x, y, and z coordinates of each point, $x_i = \{x_i, y_i, z_i\}$. K-Nearest Neighbors (KNN) search algorithm is used to obtain K nearest neighbors $x_{i_1}, \ldots\ldots, x_{i_K}$ of each point x_i in Euclidean space, so as to obtain local features of the point in Euclidean space. We define the local features of the point in Euclidean space as $E = \{x_{i_1} - x_i, \ldots\ldots, x_{i_K} - x_i\}$. Then, Laplace eigenmapping is used to reduce dimensions. The graph structure $G = \{V, E, W\}$ of the point cloud is first constructed, where V is a vertex set composed of N points, E represents the undirected edge between points, and W is a NxN symmetric matrix. We define $a_{i,j}$ as the weight of edge of the connecting vertices i and j, where t is a hyperparameter and we choose its value in experiment.

$$a_{i,j} = e^{-\frac{||x_i - x_j||^2}{t}}. \tag{1}$$

The Laplacian matrix is defined from the adjacency matrix. In different variants of Laplacian matrices, we define the combinatorial graph Laplacian used in [30] as $L_c = D - W$, where D is the degree matrix—a diagonal matrix with $d_{i,i} = \sum_{j=1}^{N} a_{i,j}$. We define normalized graph Laplacian matrix as $L = D^{-\frac{1}{2}} L_c D^{-\frac{1}{2}}$. That matrix is used in the sequel due to its normalization property. Full connection graph, connecting each point to all the other points in the point cloud, gets relationships between all the points. To calculate the eigenvalues and eigenvectors of the obtained Laplace matrix:

$$Ly = \lambda Dy. \tag{2}$$

To sort the eigenvalue and select the feature vectors corresponding to the two smallest feature values as the output after mapping to obtain the feature map $F = \{f_1, \ldots\ldots, f_N\} \in R^2$. We define a basic sort function as $sort(\{\lambda_1, \ldots\ldots, \lambda_N\})$ to get two smallest feature values.

$$\lambda_1, \lambda_2 = sort\left(\{\lambda_1, \ldots\ldots, \lambda_N\}\right), \tag{3}$$

$$Ly_1 = \lambda_1 Dy_1, Ly_2 = \lambda_2 Dy_2. \tag{4}$$

Multi-layer Perceptron (MLP) is used to learn the mapped features and gets a new feature map $\acute{F} \in R^2$. So we define it as $MLP = wX + b$, where w, b are learnable parameters and X is input features.

$$\acute{F} = MLP\left(F\right). \tag{5}$$

In the new feature space, the KNN algorithm is used to obtain K adjacent points $f_{i_1}, \ldots\ldots, f_{i_K}$ for each point x_i. By aggregating the features of adjacent points obtained in Euclidean space and Laplace mapping feature space and the

features of anchor point, the output of anchor point can be expressed as:

$$\acute{x_i} = \sum_{j=0}^{k} h_\theta(x_i, x_j, f_j), \tag{6}$$

where h_θ is for contacting features.

In order to aggregate local geometric features and global geometric features, we choose the following structure for aggregation function $h_\theta\left(x_i, x_j, f_j\right)$:

$$h_\theta\left(x_i, x_j, f_j\right) = h_\theta\left(x_i, x_i - x_j, x_i - f_j\right). \tag{7}$$

B. ATTENTION AGGREGATION MECHANISM

MaxPooling or AvgPooling are usually directly used to integrate the features of all neighborhood points. However, the Pooling operation maintains a global fairness, that is, it regards each neighborhood point as equally important, so a lot of information is lost, and it is unfair to the points carrying more geometric features. To handle this problem, attention mechanism is used to replace pooling for learning. In this way, points carrying more information will be given greater weight, so that they can play a greater role in local feature aggregation.

Computing Attention Scores: Given a set of local features $F = \{f_{i_1}, \ldots\ldots, f_{i_K}\}$, a shared function $g()$, consisting of a shared MLP followed by softmax, is used to learn a unique attention score for each feature. Shared function $g()$ is formally defined as follows:

$$d_{i_j} = g\left(f_{i_j}, W\right), \tag{8}$$

where W is the learnable weights of a shared MLP.

Finally, the weighted sum operation is carried out for the features of all selected neighborhood points:

$$f_i = \sum_{j=1}^{K} (f_{i_j} \cdot d_{i_j}). \tag{9}$$

To sum up, given the input point cloud p, for the i^{th} point p_i, the geometric patterns and features of its K nearest points are aggregated by Attentive Pooling units, and finally get an informative feature vector f_i.

C. NETWORK STRUCTURE

The GFA-Net network composed of GFA module and attention mechanism for point cloud classification and segmentation is shown in Figure 3. This model is in three layer network. The GFA module at each layer can effectively collect local geometric feature information for each point, and then attention mechanism learns and selects the extracted geometric features of surrounding points, so that neighboring points carrying more geometric features can play a greater role.

1) CLASSIFICATION MODEL

With n points as input, the local geometric features of each point are collected by GFA + attention in three layer. Then the MaxPooling operation is carried out for all points information to form a global descriptor. Finally, the global descriptor is input into the classification network model to get a classification score.

2) SEGMENTATION MODEL

The 1D global descriptor for each point and the output of GFA+ attention for each layer (used as a local descriptor) are cascading to extend, the resulting features are copied to each point through repeat operation, and then the classification score for each point is obtained through MLP.

IV. EXPERIMENTS

A. DATASETS

We studied and evaluated our proposed approach in the public datasets ModelNet40 [36] and ShapeNet [37]. ModelNet40 [36] was used to evaluate and test the point cloud classification task. ModelNet40 [36] contains 12311 models of mesh CAD from 40 categories, including 9843 models for training and 2468 models for testing. The objects in this data set are complete, without any occlusion or background. ShapeNet [37] was used to evaluate and test the point cloud segmentation task. ShapeNet [37] contains 16,881 3D shapes from 16 object categories, with a total of 50 parts are labeled. Each object has 2 to 6 parts, among which the training sample number is 12,137 and the test sample number is 2,874.

B. CLASSIFICATION

1) EXPERIMENT SETTING

We followed the experimental scheme of the model such as PointNet [14]. 1024 points were uniformly sampled from the mesh surface, and the 3D coordinates of each sampling point were used as the input data. Three layer GFA + attention extracts the local geometric features, in which the output of each GFA module was weighted by attention mechanism learning as the input to the next GFA module. For the selection of the hyperparameter, we chose the neighborhood value $K = 25$, and the calculation of the weight of the Laplace matrix, the hyperparameter $t = 20$. The feature dimensions output by each layer of GFA + attention were respectively 64, 128 and 256, which were then concatenated with the 3D coordinate features of the initial input point cloud to obtain the complete local features, then the MaxPooling was used to obtain the final global features, and finally three fully connected layers (512, 256, C) were used to classify the global features and obtain the classification score C. We used Dropout and L2 regularization to prevent overfitting and leakyRelu as the activation function. Pytorch framework was used to build the network model, set epoch = 200, batchSize = 8, and the training was carried out on the NVIDIA Tesla V100 16GB GPU. Using the SGD optimizer, the initial learning rate was 0.01, and the learning rate decreased by 0.75 decay rate for every 40 epochs.

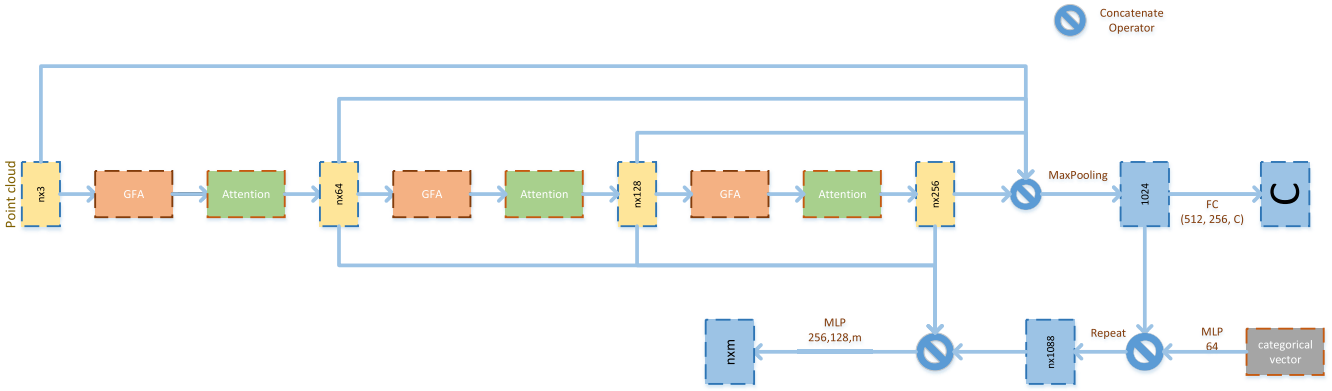

FIGURE 3. Geometry Feature Aggregation Network structure for 3D point cloud classification and segmentation. The network inputs N points, each of which has only three dimensional coordinates as its feature as input. In each layer, geometric feature aggregation module is used to enrich local geometric information, and then attention mechanism is used to learn and select the extracted geometric features of surrounding points. In addition, the output of each layer is the input of the next layer, and finally the output of the three layers is integrated into a global feature descriptor. For segmentation model, the feature descriptor of each point is concatenated with the global feature descriptor and the classification score of each point is *M* semantic labels. For classification model, the Pooling operation of the global feature descriptor is drawn into one-dimensional vector, and then input into the fully connected neural network to get a classification score.

TABLE 1. Classification results on ModelNet40.

Method	Mean Class Accuracy	Overall Accuracy
3DShapeNet[37]	77.3	84.7
MVCNN[21]	79.5	90.1
PointNet[14]	86.0	89.2
PointNet++[15]	-	90.7
PointCNN[31]	88.1	92.2
Pointweb[16]	89.4	92.3
DGCNN[19]	89.3	92.7
KPConv[28]	-	92.9
FPConv[30]	-	92.5
Ours	**90.4**	**93.2**

TABLE 2. Comparison of different *K* nearest neighbors and hyperparameter *t*.

K	t	Mean Class Accuracy	Overall Accuracy
15	15	87.0	89.2
20	15	89.1	90.3
20	20	89.3	90.5
25	15	89.5	91.6
25	20	**90.4**	**93.2**
30	20	90.0	92.7

TABLE 3. Ablation study of architecture design (%).

$K-nn$ space	Mean Class Accuracy	Overall Accuracy
$x_i - x_j$	89.7	92.0
$x_i - f_j$	89.9	92.1
$x_i - x_j, x_i - f_j$	89.4	92.4
$x_i, x_j, x_i - f_j$	**90.4**	**93.2**
$x_i, x_i - x_j, f_j$	90.0	92.7

2) EXPERIMENT ANALYSIS

We tested the mean classification accuracy (MA) and overall accuracy (OA) on ModelNet40. The results are shown in Table 1. It can be seen that our proposed method has achieved very competitive results, with the OA accuracy reaching 93.2%, 0.5% higher than DGCNN [19]. It can be seen that GFA-Net is the mainstream point cloud identification method in the classification task at present.

In addition, we tried to use different neighborhood numbers K and different hyperparameters t to carry out comparative tests to find the influence of different neighborhood numbers K and hyperparameters on the experiment. As shown in Table 2, when the neighborhood number K is 25, the performance is better. When K is less than 25, there is insufficient extraction of local points. If K is greater than 25, there may be too many neighborhood points, leading to local feature redundancy. We tested the hyperparameter t with the fixed value of K, and found that the performance is obviously better when t gets 20 than other values.

3) ABLATION STUDY

We performed ablation analysis on the components in the classification task on ModelNet40. All experiments in the ablation analysis used 1024 points, the neighborhood value $K = 25$, and the hyperparameter $t = 20$ for testing. The selection of features is a factor that affects the local geometric features information and the relationship between each point, so how to select features is important. In order to extract the most appropriate eigenvalues, we tried four settings, as shown in Table 3. It can be seen that only using the three-dimensional coordinates of Euclidean space, the performance is greater than 92.0%, while fusing the Euclidean space and the Laplacian eigenmapping space reaches 92.4%, and adding the original three-dimensional features into it reaches 93.2%. In summary, the ablation analysis prove that the dimension reduction using Laplace eigenspace is effective.

4) COMPLEXITY ANALYSIS

We evaluated the complexity of the model in terms of model size and running time in Table 4. The experiment

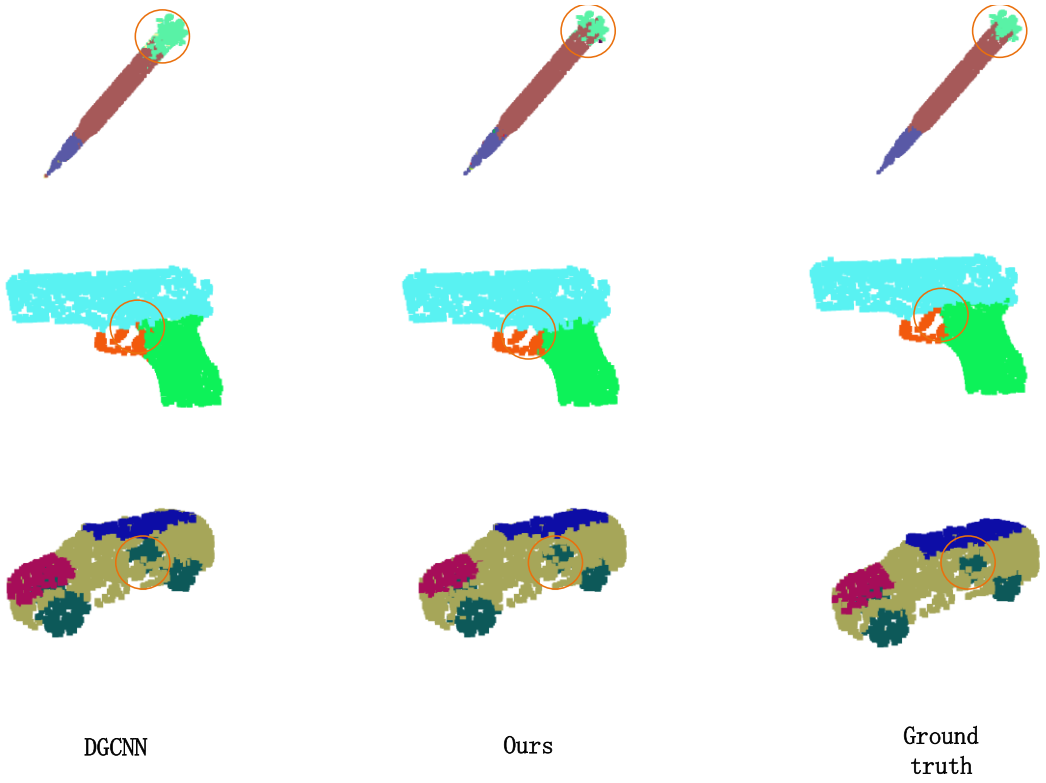

FIGURE 4. Compare part segmentation results. We visualize the results. For each set, from left to right: DGCNN [19], ours and ground truth. The differences in the figure have been circled in red.

TABLE 4. Complexity analysis of GFA in classification.

Method	Model Size(MB)	Forward Time(ms)
PointNet[14]	13.4	**30**
PointNet++[15]	**7.0**	603
DGCNN[19]	7.2	73
Ours	11.2	329

was performed on an NVIDIA Tesla V100 16GB GPU with 8 batches to calculate Forward Time (ms). Other conditions were the same as the hardware environment, and the model was implemented by PyTorch. The results show that the size of our model is second only to DGCNN [19], but in Forward Time(MS), the Time efficiency is not high because the Laplacian characteristic graph needs to be constructed.

C. SEGMENTATION

1) EXPERIMENT SETTING

The segmentation experiment was carried out on ShapeNet [37]. GFA + attention were used to extract the local geometric features in each layer, then the 1D global descriptor was connected in series with the respective outputs of the three layers, and finally the classification output of each point was calculated by the shared MLP (256,128). We also used Dropout and L2 regularization to prevent overfitting and leakyRelu as the activation function. The same training settings as the classification task were used to train on

TABLE 5. The results of part segmentation on ShapeNet.

Model	MEAN	AREO	BAG	CAP	CAR	CHAIR	EAR PHONE	GUITAR	KNIFE
#SHAPES		2690	76	55	898	3758	69	787	392
PointNet[14]	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9
PointNet++[15]	85.1	82.4	79.0	**87.7**	77.3	**90.8**	71.8	91.0	85.9
Kd-Net[24]	82.3	80.1	74.6	74.3	70.3	88.6	73.5	90.2	87.2
PointCNN[31]	86.1	84.1	**86.45**	86.0	80.8	90.6	**79.7**	92.3	**88.4**
DGCNN[19]	85.2	84.0	83.4	86.7	77.8	90.6	74.7	91.2	87.5
Ours	**86.3**	**84.1**	81.8	84.9	**81.6**	89.8	76.9	**93.3**	80.6

Model	MEAN	LAMP	LAPTOP	MOTOR	MUG	PISTOL	ROCKET	SKATE BOARD	TABLE
#SHAPES		1547	451	202	184	283	66	152	5271
PointNet[14]	83.7	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
PointNet++[15]	85.1	83.7	95.3	71.6	94.1	81.3	**68.7**	76.4	82.6
Kd-Net[24]	82.3	81.0	94.9	57.4	86.7	78.1	51.8	69.9	80.3
PointCNN[31]	86.1	85.3	**96.1**	**77.2**	95.3	**84.2**	64.2	**80.0**	83.0
DGCNN[19]	85.2	82.8	95.7	66.3	94.9	81.1	63.5	74.5	82.6
Ours	**86.3**	**85.5**	95.7	75.6	**95.6**	84.0	61.1	76.3	**83.1**

the NVIDIA Tesla V100 16GB GPU. The same evaluation scheme as DGCNN [19] was adopted. The IoUs of a shape was calculated by averaging the IoU of the different parts that appeared in the shape, and the IoUs of that shape was obtained by averaging the IoUs of all shapes that belonged to that category. At last, the mean IOU (mIOU) was calculated by averaging the IOUs of all test shapes.

2) EXPERIMENT ANALYSIS

We compared the results with the network models of PointNet [14], PointNet++ [15], PointCNN [31], DGCNN [19], and KD-Net [24], and the results are shown in Table 5. It can be seen that our method's result reaches the state-of-the-art performance in some objects' segmentation. However,

in some categories such as Cap, Bag, Rocket and other objects with small training samples, they may be inferior to other methods. The main reason is that due to the small number of samples, our method may not learn enough point features, leading to segmentation errors. In order to show the performance of our method more intuitively, Figure 4 shows the direct comparison between our method and DGCNN [19] and the ground truth.

V. CONCLUSION

We propose a novel network structure GFA-Net for point cloud feature aggregation. It consists of GFA module and attention mechanism. GFA-Net gathers the information of the same geometric feature points, which makes up for the deficiency that only the features of nearby points can be considered in Euclidean space, thus improving the robustness of rotation and translation of point clouds.

Experiments show that GFA-Net has state-of-the-art performance, can better collect local geometric features, and has strong robustness. In future work, we hope to apply our method to processing large point cloud data with more feature information.

REFERENCES

[1] A. Ioannidou, E. Chatzilari, S. Nikolopoulos, and I. Kompatsiaris, "Deep learning advances in computer vision with 3D data: A survey," *ACM Comput. Surv.*, vol. 50, no. 2, pp. 1–38, Jun. 2017.

[2] Z. Ye, J. Xue, J. Fang, J. Dou, and Y. Pan, "A decision fusion model for 3D detection of autonomous driving," in *Proc. Chin. Automat. Congr. (CAC)*, Xi'an, China, 2018, pp. 3773–3777, doi: 10.1109/CAC.2018.8623699.

[3] X. Huang, K. Shi, Y. Cui, and Y. Li, "A saturated light correction method for DMSP-OLS nighttime stable light data by remote and social sensing data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 1885–1894, 2021.

[4] S. Kunwar, H. Chen, M. Lin, H. Zhang, P. D'Angelo, D. Cerra, S. M. Azimi, M. Brown, G. Hager, N. Yokoya, R. Hänsch, and B. L. Saux, "Large-scale semantic 3-D reconstruction: Outcome of the 2019 IEEE GRSS data fusion contest—Part A," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 922–935, 2021.

[5] T. Kogut and A. Slowik, "Classification of airborne laser bathymetry data using artificial neural networks," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 1959–1966, 2021.

[6] D. Zoran, M. Chrzanowski, P.-S. Huang, S. Gowal, A. Mott, and P. Kohli, "Towards robust image classification using sequential attention models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9483–9492.

[7] W. Chen, D. Xie, Y. Zhang, and S. Pu, "All you need is a few shifts: Designing efficient convolutional neural networks for image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7241–7250.

[8] J. Koh, M. Suk, and S. M. Bhandarkar, "A multilayer self-organizing feature map for range image segmentation," *Neural Netw.*, vol. 8, no. 1, pp. 67–86, Jan. 1995.

[9] A. Golovinskiy, V. G. Kim, and T. Funkhouser, "Shape-based recognition of 3D point clouds in urban environments," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Kyoto, Japan, Sep. 2009, pp. 2154–2161, doi: 10.1109/ICCV.2009.5459471.

[10] C.-C. Feng and Z. Guo, "A hierarchical approach for point cloud classification with 3D contextual features," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 5036–5048, 2021.

[11] N. Li, O. Kahler, and N. Pfeifer, "A comparison of deep learning methods for airborne lidar point clouds classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 6467–6486, 2021.

[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] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-CNN: Octree-based convolutional neural networks for 3D shape analysis," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–11, Jul. 2017.

[14] 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)*, Honolulu, HI, USA, Jul. 2017, pp. 652–660.

[15] C. 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. systems*, Long Beach, CA, USA, Dec. 2017, pp. 5099–5108.

[16] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia, "PointWeb: Enhancing local neighborhood features for point cloud processing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5565–5573.

[17] A. Boulch, "ConvPoint: Continuous convolutions for point cloud processing," *Comput. Graph.*, vol. 88, pp. 24–34, May 2020.

[18] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8895–8904.

[19] 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.

[20] W. Han, C. Wen, C. Wang, X. Li, and Q. Li, "Point2Node: Correlation learning of dynamic-node for point cloud feature modeling," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 10925–10932.

[21] 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.

[22] 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.

[23] G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3577–3586.

[24] R. Klokov and V. Lempitsky, "Escape from cells: Deep Kd-networks for the recognition of 3D point cloud models," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 863–872.

[25] Q. Huang, W. Wang, and U. Neumann, "Recurrent slice networks for 3D segmentation of point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2626–2635.

[26] J. Li, B. M. Chen, and G. H. Lee, "SO-Net: Self-organizing network for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9397–9406.

[27] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "SpiderCNN: Deep learning on point sets with parameterized convolutional filters," *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 87–102.

[28] 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. (ICCV)*, Oct. 2019, pp. 6411–6420.

[29] B.-S. Hua, M.-K. Tran, and S.-K. Yeung, "Pointwise convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 984–993.

[30] Y. Lin, Z. Yan, H. Huang, D. Du, L. Liu, S. Cui, and X. Han, "FPConv: Learning local flattening for point convolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4293–4302.

[31] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on Xtransformed points," in *Proc. NeurIPS*, 2018, pp. 820–830

[32] M. Xu, Z. Zhou, and Y. Qiao, "Geometry sharing network for 3D point cloud classification and segmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 12500–12507.

[33] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point transformer," 2020, *arXiv:2012.09164*. [Online]. Available: http://arxiv.org/abs/2012.09164

[34] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: Point cloud transformer," 2020, *arXiv:2012.09688*. [Online]. Available: http://arxiv.org/abs/2012.09688

[35] G. Shen, W.-S. Kim, S. K. Narang, A. Ortega, J. Lee, and H. Wey, "Edge-adaptive transforms for efficient depth map coding," in *Proc. 28th Picture Coding Symp.*, Dec. 2010, pp. 566–569.

[36] 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. (CVPR)*, Jun. 2015.

[37] L. Yi, V. G. Kim, D. Ceylan, I. Shen, M. Yan, H. Su, A. 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, p. 210, 2016.