Towards Point Cloud Classification Network Based on Multilayer Feature Fusion and Projected Images

Tengteng Song[1], YiZhi He[2], Muhammad Tahir[3], Jianbo Li[4], Zhao Li[5*], Imran Saeed[6]
School of Computer Science and Technology, Shandong University of Technology, Zibo, 255000, China[1, 2, 5]
Department of Computer Science, Mohammad Ali Jinnah University, P.E.C.H.S, Karachi, 75400, Pakistan[3, 6]
School of Electronic and Electrical Engineering, Zibo Vocational Institute, Zibo, 255000, China[4]

***Abstract*—Deep Learning (DL) based point cloud classification techniques now in use suffer from issues such as disregarding local feature extraction, missing connections between points, and failure to extract two-dimensional information features from point clouds. A point cloud classification network that utilizes multi-layer feature fusion and point cloud projection images is suggested to address the aforementioned problems and produce more accurate classification outcomes. Firstly, the network extracts local characteristics of point clouds through graph convolution to strengthen the connection between points. Then, the fusing attention mechanism is introduced to aggregate the useful characteristics of the point cloud while suppressing the useless characteristics, and the point cloud characteristics are fused by multi-layer characteristic fusion. Finally, a 3D point cloud network plug-in model based on point cloud projection image (3D CLIP) is proposed, which can make up for the defects of other 3D point cloud classification networks that do not extract two-dimensional information characteristics of point clouds, and solve the problem of low accuracy of similar category recognition in datasets. The ModelNet40 dataset was used for classification studies, and the results show that the point cloud classification network, without the addition of a 3D CLIP plug-in model, achieves a classification accuracy of 92.5%. The point cloud classification network with a 3D CLIP plug-in model achieved a classification accuracy of 93.6%, demonstrating that this technique can successfully raise point cloud classification accuracy.**

Keywords*—Point cloud; classification; graph convolution; attention mechanism; CLIP

I. INTRODUCTION

As Artificial Intelligence (AI) has continued to advance, point cloud data has also evolved into a type of fundamental data [1-3]. To gather point cloud data and perform 3D reconstruction, the classification of point cloud data is crucial. As a result of the disorderly and irregular nature of data from point clouds, this poses a challenge to the task of point cloud classification.

Early Deep Learning (DL) based point cloud classification methods transform raw point cloud data into pictures or voxels before extracting point cloud characteristics using traditional classification networks. However, some of the point cloud information disappears during the point cloud transformation procedure, which lowers the network classification accuracy [4 -6]. Researchers have presented point cloud classification methods using original point cloud data, which don't require the transformation of the point cloud data, in response to the drawbacks of the point cloud classification methods. The extraction of local information characteristics from the point cloud is ignored by the present classification methods. Channel information and spatial information in the point cloud are not extracted. It is neglected how points relate to one another. The point cloud two-dimensional information is not taken into consideration. Aiming at the above problems, the primary contributions of this research paper are described below:

- A network GFANet based on fused attention mechanism and graph convolution is proposed for existing point cloud classification networks that do not extract point cloud features well. Using the ModelNet40 dataset, experimental findings demonstrate that the suggested network obtains 92.5% classification accuracy.
- A point cloud classification approach that utilizes a 2D point cloud projection image is proposed because current point cloud classification networks are not focused on the two-dimensional information of the point cloud. According to experimental findings, 3D CLIP can be plugged into a 3D point cloud classification network to increase the network classification accuracy.
- For the proposed two-point cloud classification network models, GFANet and 3D CLIP are combined to produce superior point cloud classification outcomes. On the ModelNet40 dataset, experimental findings show the point cloud classification method utilizing GFANet and 3D CLIP achieves 93.6% classification accuracy.

Based on the above, the focus of this research paper is on ways to improve the extract of the point cloud's local and global features as well as its two-dimensional information features in hopes of improving the accuracy of the point cloud classification network.

The paper is organized as follows: Section-II presents the related works for point cloud categorization. Section-III describes the proposed methodology of GFANet and 3D CLIP. Section-IV discusses the experimental results. Section-V concludes the overall research paper.

II. Related Works

The point cloud is a collection of points that can be represented as a collection of three-dimensional points (x, y, z). In addition to the information on each point location, point clouds also include details about its color, illumination level, category labels, normal vectors, grayscale values, and other characteristics. Applications for classifying point cloud data include automated driving [1], facial recognition [2], 3D reconstruction [3], and many more. The conventional point cloud categorization methods cannot be used directly on point clouds due to their irregularity and disorder.

Considering the disadvantages of conventional point cloud categorization techniques [7-9], deep learning techniques are now widely used in research to categorize point cloud data [10]. Early researchers transformed irregular 3D point cloud data into regular 3D grid data or images [11], [12] and then used 3D CNN for classification. Voxelating a point cloud primarily involves converting the point cloud data fed to the network into a grid, after which 3D CNN is used to extract features. The point cloud classification task is realized after obtaining global features through feature stitching. Other networks that convert point clouds into voxelated representations include FPNN [13], OctNet [14], and KD-NET [15]. The point cloud is projected onto a two-dimensional picture such as MVCNN [16], which projects 3D point cloud data from multiple perspectives to obtain two-dimensional images, uses a convolutional neural network to process and extract features, and then inputs the aggregated features into the convolutional neural network to realize point cloud classification. Other similar networks include GVCNN [17], SnapNet [18], and View-GCN [19].

The above two point cloud classification methods will lose some information during the conversion of point cloud data, resulting in a decline in classification accuracy. The point cloud classification method that utilizes original points may process the original point cloud directly, maximizing the retention of original point cloud data and significantly enhancing classification accuracy and algorithm performance compared to the other two point cloud classification methods mentioned above. Qi et al. suggested applying a model using deep learning on the PointNet [20] of the original point clouds, which performs well in both classifications [21]and segmentation tests [22] for point clouds. The network employs maximum pooling aggregate point features to ensure displacement invariance of point clouds and three-dimensional spatially transformed network STNs [23] to guarantee rotational consistency for point clouds. Although PointNet has several benefits, it simply extracts the point cloud global information properties. Based on the shortcomings of PointNet such as its inability to obtain local feature information and poor classification ability. Qi et al. then proposed an optimized network PointNet++ [24]. This network suggests a multi-level structure based on the PointNet for layer-by-layer extraction of local characteristics from a point cloud. However, PointNet++ also independently handles points in the point cloud, without paying attention to the connection between points. After that, researchers have also proposed some point cloud classification networks, such as ECC [25], DGCNN [26], LDGCNN [27], and GAPNet [28], but the categorization accuracy of point clouds has not been significantly improved.

Although the categorization of the point cloud method based on original points solves the shortcomings brought by some characteristics of point clouds [29], there are still shortcomings such as insufficient feature extraction and lack of point cloud feature information. To efficiently extract both local as well as global characteristics of point clouds, enhance the network feature extraction capabilities, and make up for the lack of two-dimensional information in the point cloud include an extraction process, a point cloud classification network constructed using multi-layer feature fusion and projected images is presented in this paper.

III. Methodology

There are two main components to the entire network, the network of one part is called GFANet, and the plug-in network of the other part is called 3D CLIP.

The GFANet, mainly includes the input transformation module, Graph Conv module, F-Attention module, and multi-layer feature fusion module. In the input transformation module, the input point cloud data is multiplied with a transformation matrix that the T-Net network has learned in order to ensure the consistency of the input point cloud data sequence and standardize the point cloud. In the Graph Conv module, its input is pointing to cloud features of N×f, N, and f represent the number and dimension of points respectively. The KNN algorithm is used to create a graph out of data from a point cloud. Then the graph of point cloud data is passed through n multilayer perceptions (mlp $\{L_1, L_2, ..., L_n\}$) to extract edge features. And finally, the $N \times L_n$ dimension features are obtained. The spatial and channel information characteristics from the point cloud are extracted using F-Attention to improve the network's feature extraction capabilities. Obtain global and local features of point clouds using multi-layer feature fusion. Three completely connected layers were used to achieve the point cloud final classification outcome.

The existing 3D point cloud classification network mainly extracts 3D point cloud features and then performs classification tasks. The point cloud 2D information properties are not its primary concern, so some single categories with similar features cannot be classified well. The 2D information features can provide more object representations in the network classification task and improve the network classification accuracy. A point cloud categorization approach called 3D CLIP is proposed as a result of this issue and relies upon 3D point cloud projection images. The point cloud projection image features are extracted and categorized using a 2D image classification network to increase the 2D representations available for 3D point cloud classification tasks and boost network accuracy. The key components of 3D CLIP are the text encoder and the image encoder. The network mainly uses the trained text encoder and image encoder in 2D CLIP to obtain the text description features and the projected image features of the point cloud. In the text encoder, using text-transformer to obtain the point cloud's textual description features. In the image encoder, the point cloud projection image features are extracted using ResNet. Then the

correspondence between text features and image features is found from the pre-trained model. Finally, the final classification results are obtained. The network is shown in Fig. 1.

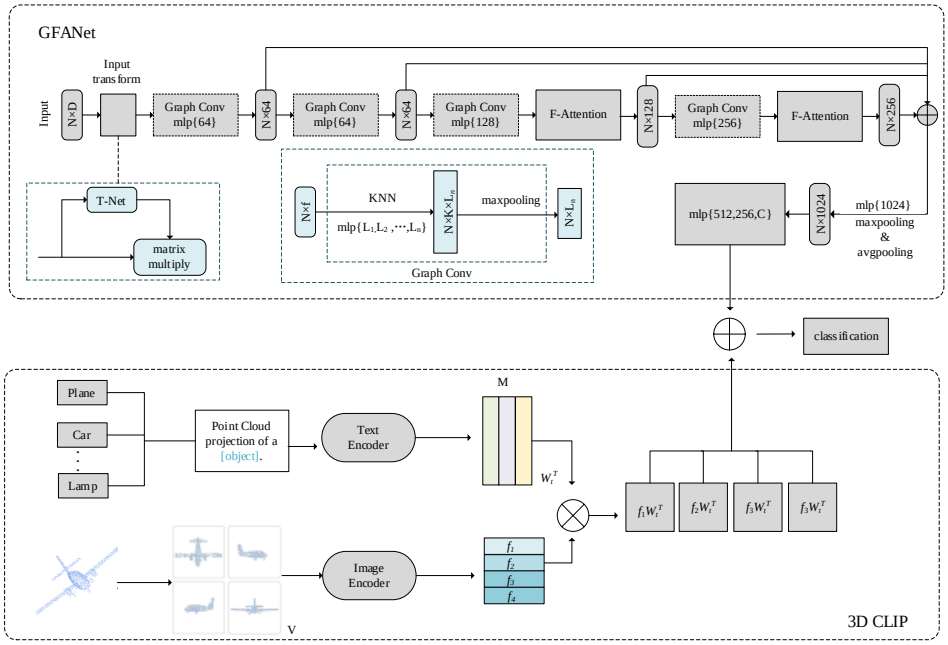

Fig. 1. GFANet and 3D CLIP structure.

A. The GFANet

1) Graph conv module: There are two types of graph convolution: spatial domain convolutions of graphs [30] and spatially domain graphs convolution [26]. Additionally, the information properties of the area of the node can be better obtained using the spatial dimension of graphs convolution. Therefore, the spatial dimension of convolutions of graphs is used to build this model.

For GFANet, model inputs can be expressed as:

$$X = \{x_1, \ldots, x_n\} \subseteq R^D \tag{1}$$

Where X is the point clouds collection, x_i is a point in the collection, and D is each point's distinctive dimension.

A directed graph with the formula $G = (V, E)$ represents the point cloud local arrangement. where V represents a collection of N point locations and E represents the collection of edges connecting nodes.

The directional graph G for GFANet is built using the k-nearest-neighbor classification (KNN) technique. The central node of a point cloud and the K nearest neighbor points which include the central node can be calculated using the KNN algorithm.

In the Graph Conv module, local features of point clouds are extracted using the edge function and the aggregation process. As below:

$$h_\theta\left(x_i, x_j\right) = h_\theta\left(x_i\right) \tag{2}$$

Where x_i and x_j are the attributes of node i and its neighboring nodes j, h_θ is a linear product of parameter x that can be learned, and θ is the collection of weight and other parameters in the network.

However, point cloud global information is the sole focus of the edge function. The local information was ignored. In Formula 3, a new edge function is created that takes into account the point cloud local as well as global information.

$$h_\theta\left(x_i, x_j\right) = h_\theta\left(x_i, x_j - x_i\right) \tag{3}$$

For aggregation operation, x_i' is the collection of edge characteristics for the central node x_i at the k points about it.

$$x_i' = \sum_{j:(i,j)\in E} h_\theta\left(x_j - x_i\right) \tag{4}$$

In Fig. 2, to create a graph structure, the KNN method is utilized. And the Graph Conv module is used to learn aggregating edge characteristics from one set of point clouds to another [31].

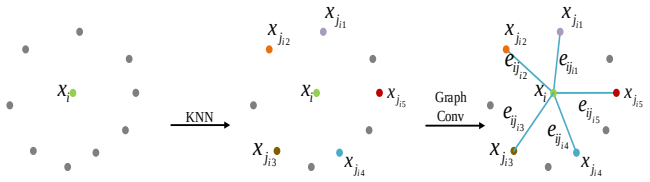

Fig. 2. Graph convolution process.

2) F-attention module: The attention mechanism [32] is divided into the space attention mechanism and the channel attention mechanism. In order to emphasize useful information features for classification tasks while suppressing useless information features, a new fusion attention mechanism was

designed, which incorporate the point cloud channel information characteristics with spatial information characteristics.

The structure of the new fusion attention mechanism (F-Attention) is shown in Fig. 3.

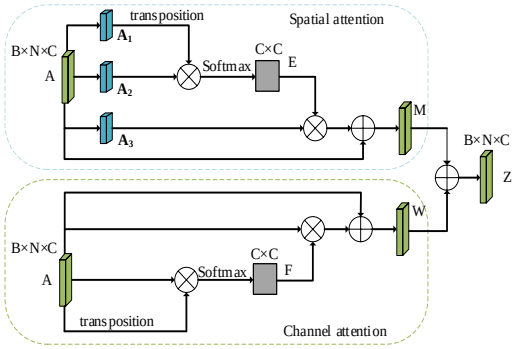

Fig. 3. Fusion attention module (F-Attention).

a) Spatial attention module: In Fig. 3, A is defined as the input point cloud feature matrix and B×N×C is the dimension of A in the Spatial attention module. The new feature matrices A_1 and A_2 can be obtained by linear transformation by A, which contains more spatial features. The two matrices have the dimensions B×N×C. Matrix A_1 is transposed and multiplied with matrix A_2, and then the spatial attention coefficient matrix E(C×C) is got using the SoftMax function, which is calculated as follows:

$$a_{ji} = \frac{exp\left(A_{1_i} \cdot A_{1_j}\right)}{\sum_{i=1}^{N} exp\left(A_{1_i} \cdot A_{1_j}\right)} \tag{5}$$

Where a_{ji} is the outcome of the SoftMax function calculation, which depicts the effect of the location i on j within matrices E.

A_3 is a new feature matrix, which is got by inputting A into the 1×1 convolutional layer. The dimension of A_3 is B×N×C. By multiplying matrices A_3 and E, an outcome feature with a dimension of B×N×C is obtained. In order to adjust weights during training, the output feature is given a linear variable λ. As illustrated in Formula 6, the final output M of feature A is created by adding the elements of the characteristic matrix refreshed on the attention mechanism to those of the initial characteristic matrix A one by one.

$$M_j = \lambda \sum_{i=1}^{N} \left(a_{ji} A_{3_i}\right) + A_j \tag{6}$$

To assign more weights by training the network, λ is initialized to 0. Both the initial point cloud characteristics and the location in space characteristics of the point cloud are included in the final feature M. M more effectively aggregates the information about the global context.

b) Channel attention module: The channel attention module input feature matrix is also defined as A. And the dimension of A is also B×N×C. The matrix A is first inverted. After that, the original matrix is multiplied by the transposed matrix. The SoftMax function is then used to produce a channel attention factor matrix F having a size of C×C.

As shown in Formula 7:

$$b_{ji} = \frac{exp\left(A_i \cdot A_j\right)}{\sum_{i=1}^{N} exp\left(A_i \cdot A_j\right)} \tag{7}$$

Where b_{ji} represents the impact of channel i on channel j.

The characteristic matrix A and the attention factor matrix F are multiplied to obtain a feature output with B×N×C. A parameter χ is introduced to adjust the weights in the network training. The final result W of characteristic A is derived by adding the elements of the original characteristic matrix A and the updated feature matrix produced by the channel mechanism, as illustrated in Formula 8:

$$W_j = \chi \sum_{i=1}^{N} \left(b_{ji} A_i\right) + A_j \tag{8}$$

Similarly, to assign more weights by training the network, χ is initialized to 0.

The final characteristic Z is obtained by fusing the characteristic M with point cloud spatial information and the characteristic W with point cloud channel information.

3) Multi-layer feature fusion module: In 3D point cloud classification tasks, fusing information features of different scales can effectively improve the classification performance of the network. Low-level features contain more location and detail information from point cloud data, but low-level features do not undergo much feature extraction, resulting in more noise and decreased semantic content. A high-level characteristic has more robust semantics, but they have poor feature resolution and poor detail perception. Therefore, before obtaining global features of point clouds through the network, it is necessary to perform feature fusion for features of 64, 128, and 256 dimensions.

In terms of the feature fusion method, select the concat feature fusion method, which essentially combines the number of feature channels, as shown in Fig. 4.

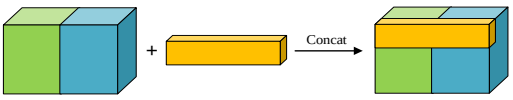

Fig. 4. Concat feature fusion.

For the two input features X and Y, if their feature dimensions are m and n, the output feature dimension after the concat operation is $m+n$. If the channels of input are X_1, X_2, …, X_c, and Y_1, Y_2, …, Y_c, respectively, the result after concat can be written as follows:

$$Z_{concat} = \sum_{i=1}^{c} X_i * K_i + \sum_{i=1}^{c} Y_i * K_{i+c} \tag{9}$$

The 64, 128, and 256-dimensional features obtained from the network are spliced using a multi-level feature fusion method, enabling the final global features to better focus on the global context information of the point cloud, enabling the network to achieve better classification accuracy. The fusion method is shown in Fig. 5.

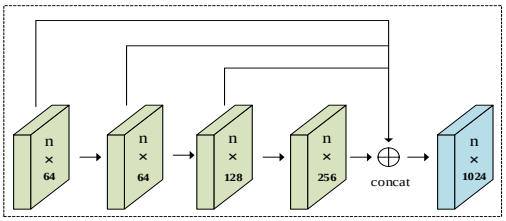

Fig. 5. Multi-layer feature fusion.

B. The 3D CLIP

The 3D CLIP plug-in network can directly perform classification tasks without pre-training. In order to obtain the text description characteristics of point clouds as well as the image features for point cloud projection images, this model primarily uses the trained text encoder and image encoder in the two-dimensional CLIP [33] and finds the corresponding relationship between text features and image features from the pre-trained model. Then, each image feature is weighted and summed with all text features, and the cosine similarity is calculated. The category corresponding to the maximum similarity text is the final classification result.

1) Text encoder: First, construct an appropriate descriptive text for each object class in the dataset. Then input these description texts into the text encoder to extract text features. M text features will be obtained after extracting the features through the text encoder. The text features extracted by the text encoder can be represented as $W_t \in R^{M \times C}$. The model text encoder employs text-transform, and the primary method of extracting textual characteristics is depicted in Fig. 6:

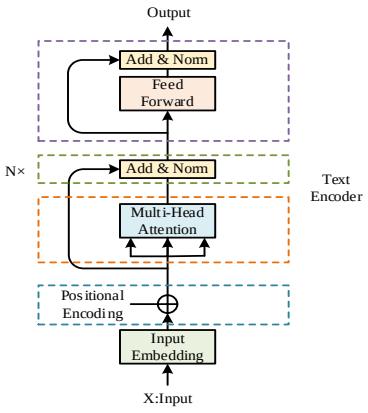

Fig. 6. Text feature extraction.

In this structure, the input of the text encoder is *X*, which represents a sentence. The final text features of the sentence are obtained after feature extraction from several modules of the encoder.

The input embedding module, the main purpose of this module is to transform the characters in a sentence into a vector. *X* can be converted into a $X_{embedding}$ vector after this module. This vector's three dimensions stand for the total number of sentences, the number of words in a sentence, and the size of each individual word.

In the positional encoding module, the position of each word in the input sentence is encoded and marked. The encoding calculation process can use the sine and cosine function, which is calculated as:

$$PE(pos, 2i) = sin\left(pos / 10000^{2i/d_{model}}\right) \quad (10)$$

$$PE(pos, 2i+1) = cos\left(pos / 10000^{2i/d_{model}}\right) \quad (11)$$

where *i* indicates the size of the word vector and *pos* the position of each word within the phrase.

After the position encoding module, an encoding array X_{pos} with the same dimension as the input sentence can be obtained. And the new word vector can be obtained by superimposing X_{pos} with the original vector:

$$X_{embedding} = X_{embedding} + X_{pos} \quad (12)$$

In the multi-head attention module, this module enables the model to learn the expression of multiple meanings. The module uses the self-attention attention mechanism to linearly map the inputs to obtain Q, K, V:

$$\begin{aligned} Q &= X_{embedding*} W_Q \\ K &= X_{embedding*} W_k \\ V &= X_{embedding*} W_v \end{aligned} \quad (13)$$

where the dimensions of Q, K, and V are the same as the $X_{embedding}$ dimensions.

In the add and norm module, the main operations are residual concatenation and normalization. The preceding layer input *X* is added to the output via the residual join. The normalization operation is to subtract the mean value of each row and divide it by the standard deviation of the row to obtain the normalized value.

The feedforward module contains two layers of linear mapping and activation using the activation function. The final output is obtained after the same add and norm operation.

2) Image encoder: Because the images input by the CLIP model when using the image encoder to extract image features are all two-dimensional, it is necessary to perform two-dimensional processing of three-dimensional point cloud data. The specific operation is to project the three-dimensional point cloud data in the dataset from multiple perspectives into a two-dimensional depth image.

The spatial coordinates of a point cloud for 3D data in a dataset can be represented as (x, y, z). When projecting in the z-direction, the point can be transformed into ([x/z], [y/z]). The advantage of this projection is that it can make the image closer to a natural image. Because the image encoder of the CLIP

model processes three-channel RGB images, to obtain point cloud-related features from the projected image, the projected image is copied twice to become a three-channel image before being input to the image encoder.

The mapping formula for projecting 3D point cloud data point *A* to 2D coordinate system point *B* is as follows:

$$\vec{A}=\begin{bmatrix}x\\y\\z\end{bmatrix}\rightarrow\vec{B}=\begin{bmatrix}\alpha\cdot\frac{x}{z}+C_x\\\beta\cdot\frac{y}{z}+C_y\end{bmatrix} \tag{14}$$

In the selection of the image encoder, since the ResNet [34] is used in the 2D CLIP to achieve better results in classification tasks, the ResNet will also be used for feature extraction in the 3D CLIP selection of the image encoder.

ResNet is a residual network, and a residual network is composed of a series of residual blocks. For ResNet, it contains two basic modules, identity block, and conv block, and the module structure is shown in Fig. 7:

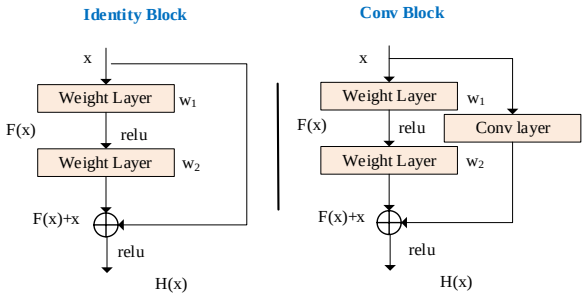

Fig. 7. ResNet main module.

In the identity block module, *x* is the input and *H(x)* is the output:

$$H(x)=F(x,\{w_i\})+x \tag{15}$$

where $F(x, \{w_i\})$ denotes the residual, which is the target to be learned. It represents the operator relationship between the weights and the input, $F(x)=H(x)-x$.

Unlike the identity block module, the conv block module adds the conv layer convolution operation on top of it. The shape of the input matrix can be adjusted so that the residual edges and the convolution in the module can be summed.

For depth, images projected from *V* different angles of view, use an image encoder to extract image features. The extracted image features have a total of f_i, where $i=1, ..., V$.

During the classification process, since the 3D CLIP has already obtained w_t text features and f_i image features, it is only necessary to calculate the classification $logits_i$ of each projection view separately. Finally, weighted summation can be used to get the point cloud final classification $logits_h$, and the classification outcome. The calculation formula is as follows:

$$log\ its_i = f_i W_t^T, i=1,\ldots,V \tag{16}$$

$$log\ its_h = \sum_{i=1}^{v} log\ its_i \tag{17}$$

IV. Experiments and Result

A. Datasets

For accurately assessing the network categorization performance for this article, the open dataset ModelNet40 proposed by Princeton University was selected for training and testing the network. There are a total of 12311 CAD models in the dataset, with 9843 models used for training and 2468 models used for testing. Each model has its corresponding category and is divided into 40 categories of artificial objects. Select four categories from the ModelNet40 dataset: airplane, plant, chair, and person for visualization. The results are shown in Fig. 8:

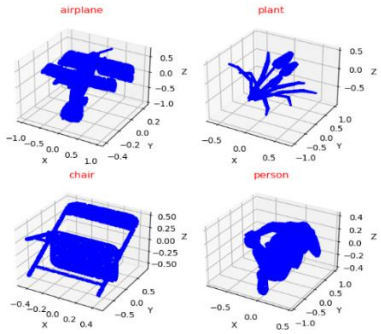

Fig. 8. Partial category visualization.

Because the point cloud set contains a sizable number of useless and noise points. The point cloud categorization network capacity to extract features will decline. In addition, when the number of points used to input the network is too large, it can generate many parameters during training. It will affect the training speed of the network. The subsampling algorithm can remove noise points and ensure the same number of points input to the model.

Fig. 9 displays the visualization of point cloud data following sampling. The original point cloud contains 10000 points. After sampling, 1024 points can be obtained. These points can represent the object well and contain rich object details.

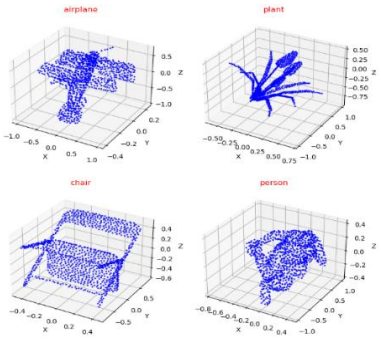

Fig. 9. Point cloud data sampling.

B. Experimental Setting

The hardware environment required for this model is Intel core i5-9400, the software environment consists of Python 3.7,

CUDA10.1, PyTorch 1.6, and Ubuntu 20.04.2 LTS. The learning rate for the experimental parameters has been set to 0.001. There are 250 iterations in total. 32 is the set batch size. The Adam optimizer is used.

The evaluation metrics of the network are the overall classification accuracy (OA) and the average classification accuracy (mAcc). As follows:

$$OA=\frac{TP+TN}{TP+TN+FP+FN} \tag{18}$$

$$Precision=\frac{TP}{TP+FP} \tag{19}$$

$$mAcc=\frac{\sum_{m=1}^{M} Precision_m}{M} \tag{20}$$

where *TP* is the number of samples with accurate predictions. *TN* represents the number of samples with incorrect predictions. The sample quantity of false positives is denoted by *FP*. The sample quantity of false negatives is denoted by *FN*. *Precision* is the accuracy rate. *M* is the classification number.

C. Experimental Results Analysis

1) Pooling method selection: To study the effects of various pooling methods on the classification precision of network models, max pooling, average pooling, and a combination of the two pooling methods were compared in the process of getting the global feature. Assume that method A uses only average pooling, method B uses only max pooling, and method C uses both average pooling and max pooling. Where √ indicates using this method, × indicates that this method is not used, and the classification accuracy is shown in Table I.

TABLE I. GFANET CLASSIFICATION ACCURACY UNDER DIFFERENT POOLING MODES

Pooling method	Avg. Pooling	Max Pooling	mAcc/%	OA/%
A	√	×	89.3	91.5
B	×	√	89.2	91.6
C	√	√	**90.2**	**92.5**

According to the test results of AvgPooling and MaxPooling in Table I, the combined use of max pooling and average pooling improves classification accuracy compared to utilizing either pooling approach alone. The average classification accuracy of using method C is 0.09% and 0.1% higher than that of method A and method B, respectively. And the overall classification accuracy of method C is 0.1% and 0.09% higher than that of method A and method B, respectively. This demonstrates that the information lost during the global feature selection process can be reduced by combining average pooling and max pooling. As a result, for feature extraction during the construction of GFANet, average pooling, and max pooling are combined.

2) Analysis of network classification accuracy: To compare with the GFANet, many traditional point cloud classification networks are used. The ModelNet40 dataset is selected as the testing dataset. The classification accuracy of different networks on the ModelNet40 dataset is shown in Fig. 10 and Fig. 11.

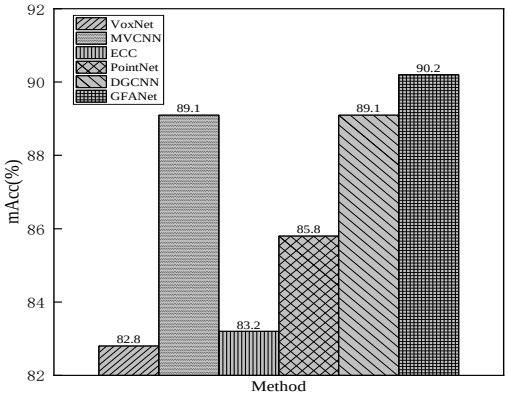

Fig. 10. Average classification accuracy of different networks.

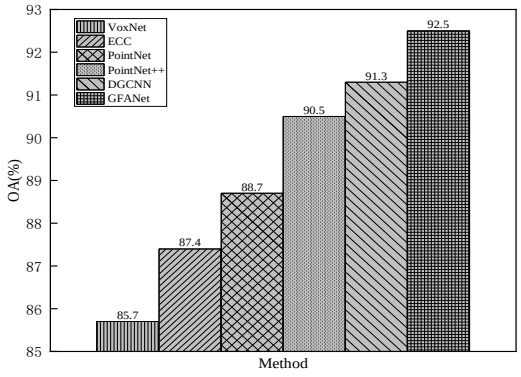

Fig. 11. Overall classification accuracy of different networks.

In contrast to traditional point cloud categorization networks, the GFANet has higher classification accuracy. Compared with PointNet, the GFANet has an overall classification accuracy improvement of 3.8% and an average improvement in classification accuracy of 4.4%. The reason is that GFANet concentrates on the point clouds local and global information characteristics. The GFANet exhibits an overall classification accuracy improvement of 2.0% when compared to PointNet++. The reason is that the connection between points is strengthened and the information feature between point pairs is focused in GFANet. While PointNet++ just processes points separately. Compared with DGCNN, the GFANet has an overall classification accuracy improvement of 1.2% and an average improvement in classification accuracy of 1.1%. The reason is that the information of point pairs is focused on GFANet. And a fusion attention mechanism is added in GFANet. In addition, the spatial and channel information properties of point clouds are extracted by GFANet.

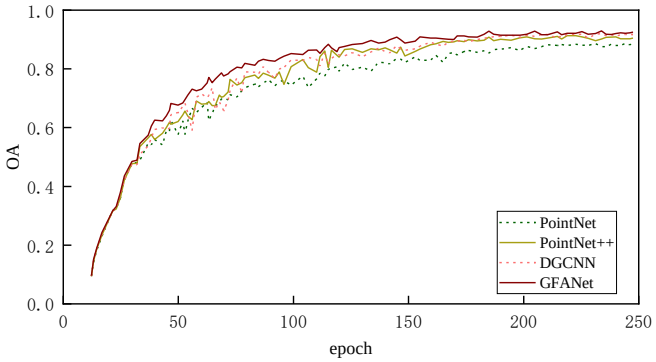

Fig. 12. Network classification accuracy.

For the ModelNet40 dataset, the classification accuracy curve obtained through 250 iterations for PointNet, PointNet++, DGCNN, and GFANet is shown in Fig. 12.

The GFANet has significantly better classification accuracy than the other three networks in most training cycles, especially in the middle and late stages of training. It has been demonstrated that GFANet can increase the classification accuracy of point clouds.

3) Comparative experiments of different categories: Like the PointNet, the GFANet is a classification net that accepts data from point clouds directly. And the construction of the GFANet also refers to the PointNet. The comparison results of GFANet with PointNet and PointNet++ on ModelNet40 data set for individual classification of each category are shown in Table II.

Compared with PointNet and PointNet++, the GFANet has higher accuracy for most categories. For the categories with obvious features, such as the Bench, Guitar, and Lamp, the classification accuracy of GFANet is 5%, 2%, and 1.3% higher than that of PointNet, and is 3%, 0.5% and 0.7% higher than that of PointNet++. For the categories with no obvious features, such as the Bathtub, Door, and Wardrobe, the classification accuracy of GFANet is 5%, 1.2%, and 4% higher than that of PointNet, and is 3%, 1%, and 3% higher than that of PointNet++.

The reason is that both the local information characteristics and the global information characteristics for data points are considered in GFANet. Additionally, GFANet takes into account the channel information features as well as the spatial information features for data points.

TABLE II. COMPARISON RESULTS FOR 40 CATEGORIES

Category	PointNet	PointNet++	GFANet	Category	PointNet	PointNet++	GFANet
Airplane	1.000	1.000	1.000	Laptop	1.000	1.000	1.000
Bathtub	0.870	0.890	0.920	Mantel	0.930	0.940	0.950
Bed	0.960	0.964	0.970	Monitor	0.950	0.960	0.980
Bench	0.700	0.720	0.750	Night_stand	0.742	0.753	0.776
Bookshelf	0.910	0.918	0.930	Person	0.920	0.930	0.950
Bottle	0.940	0.952	0.960	Piano	0.900	0.910	0.930
Bowl	0.900	0.920	0.940	Range_hood	0.920	0.930	0.952
Car	0.960	0.971	0.980	Sink	0.780	0.800	0.850
Chair	0.970	0.974	0.980	Sofa	0.960	0.963	0.970
Cone	0.950	0.960	1.000	Stairs	0.800	0.840	0.900
Cup	0.780	0.790	0.800	Stool	0.850	0.860	0.800
Curtain	0.900	0.910	0.920	Table	0.800	0.830	0.870
Desk	0.800	0.880	0.900	Tent	0.950	0.951	0.953
Door	0.800	0.860	0.920	Toilet	0.980	0.982	0.970
Dresser	0.696	0.700	0.726	Tv_stand	0.800	0.830	0.860
Flower	0.220	0.230	0.250	Vase	0.820	0.825	0.830
Glass	0.950	0.960	0.970	Wardrobe	0.750	0.760	0.790
Guitar	0.980	0.985	1.000	Xbox	0.650	0.680	0.750
Keyboard	1.000	1.000	1.000	Plant	0.760	0.770	0.780
Lamp	0.950	0.956	0.963	Radio	0.750	0.770	0.800

4) Analysis of adding 3D CLIP classification accuracy: The effectiveness of the 3D CLIP network is demonstrated by comparing the accuracy of point cloud classification with and without adding 3D CLIP in PointNet, PointNet++, DGCNN, and GFANet. The classification accuracy is shown in Table III.

TABLE III. CLASSIFICATION ACCURACY OF DIFFERENT NETWORKS

Method	3D CLIP	mAcc/%	OA/%
PointNet [20]	×	85.8	88.7
	√	87.3	90.1
PointNet++ [24]	×	—	90.5
	√	—	91.4
DGCNN [26]	×	89.1	91.3
	√	90.4	92.7
GFANet	×	90.2	92.5
	√	**91.1**	**93.6**

The overall accuracy of classification of PointNet is 1.4% higher and the average accuracy of classification is 1.5% higher when the 3D CLIP is used. The overall accuracy of the classification of PointNet++ is 0.9% higher when the 3D CLIP is used. The overall accuracy of classification of DGCNN is 1.4% higher and the average accuracy of classification is 1.3% higher when the 3D CLIP is used. The overall accuracy of classification of GFANet is 1.1% higher and the average accuracy of classification is 0.9% higher when the 3D CLIP is used. The experiment results indicate that point cloud classification networks with 3D CLIP have a certain improvement in classification accuracy compared to networks without 3D CLIP. The reason is that 3D CLIP can extract two-dimensional feature information of point clouds. The GFANet with 3D CLIP has the highest classification accuracy compared to other networks with 3D CLIP. It proves the effectiveness of the GFANet and 3D CLIP. It also demonstrates the potential of 3D CLIP to enhance the classification accuracy of point cloud categorization networks.

5) Analysis of 40 categories classification results of GFANet adding 3D CLIP: According to Table II, for the categories with similar characteristics in the ModelNet40 dataset, such as cup and vase, flower_pot and plant, nightstand, and wardrobe. GFANet and the existing classical point cloud classification network cannot be well classified, and these categories are shown in Fig. 13. The primary cause is that the network only concentrates on the three-dimensional information feature information of the point cloud and does not extract the two-dimensional information feature of these categories, making it difficult for the network to distinguish and identify, and resulting in a relatively small improvement in the classification accuracy of these single categories. In this experiment, the 3D CLIP is added to the GFANet to prove that the 3D CLIP can help the GFANet to better distinguish different categories with similar features and improve the classification performance of the network. The findings of the experiment are displayed in Table IV.

Table IV shows that in comparison to GFANet alone, the network comprising GFANet and 3D CLIP has somewhat increased the classification accuracy of the 40 categories of the ModelNet40 data set. For the cup and vase categories with similar features, the accuracy of classification is increased by 2% and 1.8%, respectively. For the flower_pot and plant categories with similar features, the accuracy of classification is increased by 3% and 2%, respectively. For the nightstand and wardrobe categories with similar features, the accuracy of classification is increased by 3% and 2%, respectively. The classification accuracy of the network is improved by 2.6% and 3%, respectively. This is so that the network can both extract the three-dimensional information features of the point cloud and learn the two-dimensional information representation of the point cloud. The 3D CLIP is an addition to the GFANet that can provide more two-dimensional information about the point cloud for the network. Thereby the network can better distinguish between different categories with similar features and raise the classification accuracy of various categories. Improve the classification accuracy of the network.

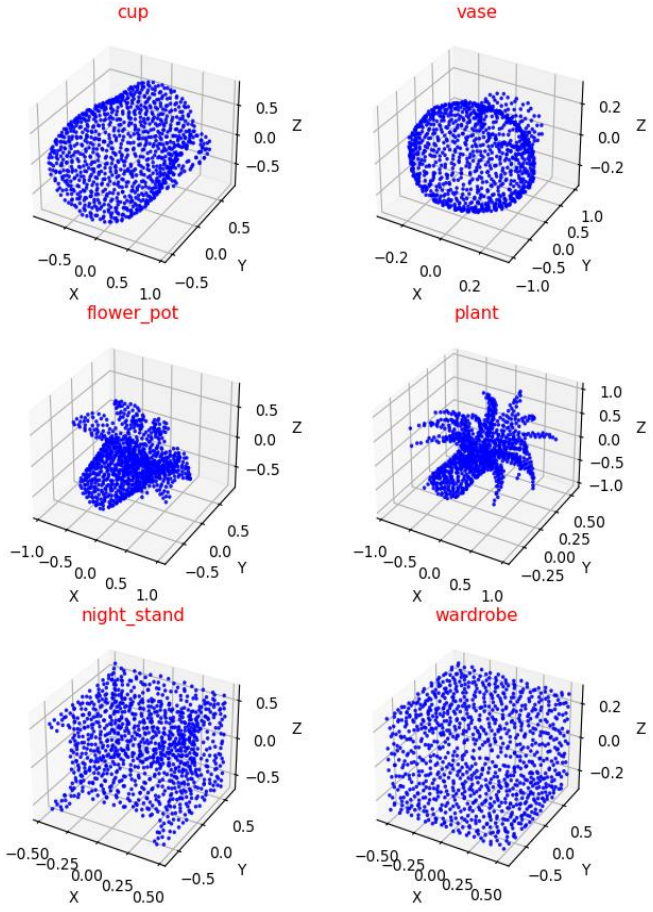

Fig. 13. Different categories with similar features in the ModelNet40 dataset.

TABLE IV. ANALYSIS OF 40 CATEGORY CLASSIFICATION RESULTS

Category	GFANet	GFANet+3D CLIP	Category	GFANet	GFANet+3D CLIP
Airplane	1.000	1.000	Laptop	1.000	1.000
Bathtub	0.920	0.930	Mantel	0.950	0.960
Bed	0.970	0.978	Monitor	0.980	0.983
Bench	0.750	0.800	Night_stand	0.770	0.802
Bookshelf	0.930	0.940	Person	0.950	0.960
Bottle	0.960	0.970	Piano	0.930	0.940
Bowl	0.940	0.950	Range_hood	0.952	0.962
Car	0.980	0.987	Sink	0.850	0.860
Chair	0.980	0.983	Sofa	0.970	0.980
Cone	1.000	1.000	Stairs	0.900	0.920
Cup	0.800	0.820	Stool	0.800	0.850
Curtain	0.920	0.930	Table	0.870	0.900
Desk	0.900	0.910	Tent	0.953	0.961
Door	0.920	0.926	Toilet	0.970	0.980
Dresser	0.726	0.862	Tv_stand	0.860	0.880
Flower	0.250	0.280	Vase	0.830	0.848
Glass	0.970	0.975	Wardrobe	0.790	0.820
Guitar	1.000	1.000	Xbox	0.750	0.800
Keyboard	1.000	1.000	Plant	0.780	0.800
Lamp	0.963	0.986	Radio	0.800	0.880

V. CONCLUSION

Targeting the issues that the current point cloud classification methods disregard the point cloud's local feature extract, lack the connection between points and points, and do not extract the two-dimensional information features of the point cloud when obtaining the point cloud features. To obtain a more precise classification result, a point cloud categorization network using a multi-layer fusion of features and point cloud projection image was proposed. The network employs dynamic graph convolution to enhance the association between points by extracting local characteristics from the point cloud. The point cloud features were fused via multi-layer feature fusion, and the fusion attention method was devised to collect the useful characteristics of the point cloud while suppressing the useless features. Finally, a 3D point cloud network plug-in model based on a point cloud projection image, 3D CLIP, is used to make up for the lack of extracting two-dimensional information features of the point cloud, to increase the network accuracy at classifying objects.

FUNDING STATEMENT

This Research was funded by the National Key R&D Program of P.R. China under project number: 2022YFE0107300.

ACKNOWLEDGMENT

The authors are grateful to the anonymous referees for their insightful comments and suggestions.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest to report regarding the present research paper.

REFERENCES

[1] M. Klinger, K. Muller, M. Mirzaie, J. Breitenstein, J. Termohlen, and T. Fingscheidt, "On the Choice of Data for Efficient Training and Validation of End-to-End Driving Models," *IEEE/CVF Conf. Comput. Vis. Pattern Recognit Workshop. (CVPRW)*, Jun, 2022, pp. 4802-4811.

[2] G. Gao, H. Yang, and H. Liu. "3D point cloud face recognition based on deep learning," *Journal of Computer Applications*, May, 2021, pp. 2736-2740.

[3] B. Ma, Y. S. Liu and Z. Han, "Reconstructing Surfaces for Sparse Point Clouds with On-Surface Priors," *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, June, 2022, pp. 6305-6315.

[4] Z. Deng and L. J. Latecki, "Amodal detection of 3d objects: Inferring 3d bounding boxes from 2d ones in rgb-depth images," *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, July, 2017: pp. 398-406.

[5] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong and I. Posner, "Vote3 deep: Fast object detection in 3d point clouds using efficient convolutional neural networks," *IEEE Int. Conf. on Robotics and Automation. (ICRA)*, May, 2017, pp. 1355-1361.

[6] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," I*EEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, June, 2017, pp. 1137-1149.

[7] R. B. Rusu, Z. C. Marton, N. Blodow and M. Beetz, "Learning informative point classes for the acquisition of object model maps," I*EEE Int. Conf. on Robotics and Automation. (ICARCV)*, Dec, 2008, pp. 643-650.

[8] R. B. Rusu, G. Bradski, R. Thibaux and J. Hsu, "Fast 3d recognition and pose using the viewpoint feature histogram," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. (IROS)*, 2010, pp. 2155-2162.

[9] J. Sun, M. Ovsjanikov and L. Guibas, "A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion," *Computer Graphics Forum*, Aug, 2009, pp. 1383-1392.

[10] W. B. Jie, N. L. Ping and Z. W. Hui, "3D point cloud classification and segmentation network based on Spider convolution," *Journal of Computer Applications*, 2020, pp. 1607-1612.

[11] J. Lahoud and B. Ghanem, "2D-Driven 3D Object Detection in RGB-D Images," *IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct, 2017, pp. 4622–4630.

[12] D. Maturana and S. Scherer, "VoxNet: A 3D Convolutional Neural Network for real-time object recognition," *IEEE Int. Conf. Intell. Rob. Syst. (IROS)*, Dec, 2015, pp. 922-928.

[13] Y. Li, S. Pirk, H. Su, C. R. Qi and L. J. Guibas, "Fpnn: Field probing neural networks for 3d data," *Advances in Neural Information Processing Systems. (NIPS)*, Dec, 2016, pp. 307-315.

[14] G. Riegler, A. Ulusoy and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, Nov, 2017, pp. 3577-3586.

[15] R. Klokov and V. Lempitsky, "Deep kd-networks for the recognition of 3d point cloud models," *IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec, 2017, pp. 863-872.

[16] H. Su, S. Maji, E. Kalogerakis and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," *IEEE Int. Conf. Comput. Vis. (ICCV)*, Feb, 2015, pp. 945-953.

[17] Y. Feng, Z. Zhang, X. Zhao, R. Ji and Y. Gao, "GVCNN: Group-view convolutional neural networks for 3D shape recognition," *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, June, 2018, pp. 264-272.

[18] A. Boulch, J. Guerry, B. L. Saux and N. Audebert, "SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks," *Computers and Graphics*, April, 2018, pp. 189-198.

[19] X. Wei, R. Yu and J. Sun, "View-GCN: View-Based Graph Convolutional Network for 3D Shape Analysis," *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, Aug, 2020, pp. 1847-1856.

[20] R. Q. Charles, H. Su, M. Kaichun and L. J. Guibas, "PointNet: Deep learning on point sets for 3d classification and segmentation," *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, July, 2017, pp. 652-660.

[21] X. Ma, C. Qin, H. You, H. Ran and Y. Fu, "Rethinking Network Design and Local Geometry in Point Cloud: A Simple Residual MLP Framework," *IEEE. Conf. Learn. Represent.* (*ICLR*), Jan, 2022.

[22] X. Lai, J. Liu, L. Jiang, L. Wang, H. Zhao, S. Liu and et al, "Stratified Transformer for 3D Point Cloud Segmentation," *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, June, 2022, pp. 8490-8499.

[23] M. Jaderberg, K. Simonyan, A. Zisserman and K. Kavukcuoglu, "Spatial Transformer Networks," *Advances in Neural Information Processing Systems. (NIPS)*, Dec, 2015, pp. 2017-2025.

[24] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in Neural Information Processing Systems. (NIPS)*, Dec, 2017, pp. 5105-5114.

[25] M. Simonovsky and N. Komodakis, "Dynamic edge-conditioned filters in convolutional neural networks on graphs," *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, July, 2017, pp. 29-28.

[26] 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 Transactions on Graphics*, Oct, 2019, pp. 1-12.

[27] K. Zhang, M. Hao, J. Wang, X. Chen, Y. Leng, C. W. Silva and et al, "Linked dynamic graph CNN: Learning on point cloud via linking hierarchical features," *IEEE Conference on M2VIP*, Nov, 2021, pp. 7-12.

[28] C. Chen, L. Z. Fragonara and A. Tsourdos, "GAPoint-Net: Graph attention based point neural network for exploiting local feature of point cloud," *Neurocomputing*, May, 2021, pp. 122-132.

[29] W. W. Xi and L. L, Lin, "A review of deep learing in point cloud classification," *Computer Engineering and Applications*, Jan, 2022, pp. 26-40.

[30] Y. Zhang and M. Rabbat, "A graph-CNN for 3D point cloud classification," *IEEE Intl. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, April, 2018, pp. 6279-6283.

[31] T. Song, Z. Li, Z. Liu and Y. He, "Point Cloud Classification Network Based on Graph Convolution and Fusion Attention Mechanism," *Journal of Computer and Communications*, Oct, 2022, pp. 81-95.

[32] Z. C, Z. Lei and Y. Lu, "Review of Attention Mechanism in Convolutional Neural Networks," *Computer Engineering and Applications*, Oct, 2021, pp. 64-72.

[33] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal and et al, "Learning transferable visual models from natural language supervison," *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, Feb, 2021, pp. 1-48.

[34] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, June, 2016, pp. 770-778.