Dual-Attention Network with Aggregate Local Descriptors Vector for 3D Point Cloud Classification

Guodao Zhang  
Zhejiang University of Technology

Jian Zhou  
Zhejiang University of Technology

Hangli Weng  
Zhejiang University of Technology

Yisu Ge  
Wenzhou University

MingTao Ye  
Zhejiang University of Technology

Xiaonan Li  
Zhejiang University of Technology

Ping-Kuo Chen (✉ pgchen@stu.edu.cn)  
Shantou University

Feiyue Qiu  
Zhejiang University of Technology

Research Article

Keywords: point cloud classification, self-attention, graph attention mechanism, VLAD.

Posted Date: April 4th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1286445/v1

License: © This work is licensed under a Creative Commons Attribution 4.0 International License.  
Read Full License
Dual-Attention Network with Aggregate Local Descriptors Vector for 3D Point Cloud Classification

Guodao Zhang, Jian Zhou, Hangli Weng, Yisu Ge, MingTao Ye, Xiaonan Li, Ping-Kuo CHEN* and Feiyue Qiu

Abstract—Contextual fine-grained local features play an essential position in 3D point cloud classification, and have not been utilized effectively in existing deep-learning-based models. Aiming to address this problem, a 3D point cloud classification network based on a dual attention mechanism with vector of locally aggregated descriptors (VLAD) is proposed. First, the local geometric representation is learned by embedding a graph attention mechanism in a multilayer perceptron layer. To make the most of the features, a multithreaded mechanism is applied to aggregate different features from separate headers, and an effective key point descriptor is introduced to help identify the context structural information of the 3D model, enhance the fine-grained detail capture operation is performed on each dimension of the features to apply deep learning to 3D models. However, the application of deep learning to 3D model classification is still a major challenge due to the complex internal structure of 3D models.

Geometric deep learning [1-7] is an algorithm for processing non-Euclidean structure data (such as 3D point clouds) through deep neural networks. In the literature [8], the benefits of graph representations methods for non-Euclidean data processing tasks are demonstrated, and the use of graph convolutional neural networks for the direct processing of irregularly structured data, such as point clouds, is proposed. The PointNet [9] model proposed in the literature is the pioneer of direct processing of 3D point clouds. After feature extraction of each point cloud data by a multilayer perceptron, a max pooling operation is performed on each dimension of the features to obtain the final global features of the classification model. Although each point in the point clouds has location information and additional attribute values, namely, colour, normal, reflectivity, and so on, these express only the physical meaning of the point itself without considering its neighbouring and contextual meaning. Therefore, mining point clouds for contextually fine-grained local features still requires a huge challenge.

To solve the above problems, this paper aims to explore a deep learning based 3D point cloud classification model architecture based on the dual attention mechanism, fully excavate the fine-grained local and global features of the spatial context of the 3D point cloud model, enhance the fine classification ability of the network, and generally improve the classification accuracy of the 3D model. Fine-grained local features are helpful to enhancing the fine-grained detail capture ability of the network, and spatial context features can enhance the integrity of the features as it is more sensitive to the spatial context structural information of the 3D model. The main innovations and contributions of this paper are as follows.

- We have built an end-to-end 3D point cloud classification model, DAMVNet. This network adopts point-by-point feature extraction and aggregation operations to address the problem of disordered point cloud classification accuracy of the 3D model. Fine-grained local features are helpful to enhancing the fine-grained detail capture ability of the network, and spatial context features can enhance the integrity of the features as it is more sensitive to the spatial context structural information of the 3D model. The main innovations and contributions of this paper are as follows.

P. CHEN is the corresponding author, he is with the Business School, Shantou University, Shantou 515000, China (email: pchen@stu.edu.cn)

Y. Ge with the College of Computer Science and Artificial Intelligence, Wenzhou University Wenzhou 325100, China (email: ysg@wzu.edu.cn)

M. Ye is with the College of Computer Science, Zhejiang University of Technology, Hangzhou 310023, China (email: 18381089570@163.com)

X. Li is with the College of Computer Science, Zhejiang University of Technology, Hangzhou 310023, China (email: 1096093078@qq.com)

F. Qiu is with the College of Computer Science, Zhejiang University of Technology, Hangzhou 310023, China (email: qfyq@zjut.edu.cn)
II. RELATED WORK

The 3D point cloud is one of the most significant representations of 3D data. It can be acquired by data acquisition with a 3D laser scanner. One of the main disadvantages of this representation of 3D data is that it inherently relies on a specific sampling pattern, which makes it sensitive to occlusion, noise and sparse sampling. With rapidly developing deep learning techniques and more datasets [8] becoming available to the public, point cloud processing has become an important area of development in deep learning. Deep neural networks have been used with remarkable success in a variety of visual tasks. However, applying CNNs to irregularly structured data (such as 3D point clouds) remains a 3 challenge. These challenges include: (1) Geometrically, due to the spatial structure of the 3D point cloud, it is hard to represent the points in the point cloud with a matrix, which makes global and local geometric representation quite difficult; (2) Unstructured data (no grid); (3) Invariant alignment, point clouds are essentially a long string of points, which leads to rotation, displacement and translation invariance. However, not all deep neural networks can completely solve these problems.

PointNet, proposed by Qi et al [9], is a pioneering method that directly uses the original point cloud as the input of the network. It directly uses the 3D coordinates of the point as the input, obtains the features of the point with the help of a multilayer perceptron, applies maximum pooling to address the disorder of the point cloud, and finally outputs the classification results of the point cloud through the fully connected layer. However, the learning of local features is not involved. To solve this problem, PointNet + + [2] uses the distance metric of the underlying space to segment the point cloud into overlapping local regions and extracts and captures the local features of fine geometry from the neighbours; it still considers only each point within its local region independently and captures local information using the same semantic level, ignoring the inherent differences between different semantic levels. KCNet [10] adds a graph-based pooling layer to PointNet to enhance the robustness of the network using the local feature structure; however, it does not use the pooling layer to improve the semantic level. These several methods have led many scholars to study the method of direct point-to-point operation one after another.

Successive scholars have introduced k-d trees, octrees, graphs and other structures to capture local relationships between unstructured points [5, 11, 12]; for example, FoldingNet [13] uses graph-based max-pooling to down sample the graph. However, however, these methods do not ensure that the most critical points (key points) are accepted downstream, instead those with less relevant characteristics could be transmitted so that key points could be deleted or depreciated. Some scholars have also proposed a new convolution strategy to collect information from adjacent points. Some have also tried to introduce various local features, such as the distance of adjacent points and the angle of local surface normal, and use them as a representation of the point clouds. Atzmon et al [14] proposed a PCNN network, a framework with extended and restricted operators, using both operators to transform point-based representations into voxel-based representations. The voxels are used as input and volume convolution is performed to extract the features of the points. Li et al [7] introduced PointCNN, a point cloud convolutional network, to apply the element product and sum operations of typical convolutional operators to X-transformed features. Liu et al [15] proposed the RS-CNN framework, which extends the traditional CNN to handle unstructured data and proposes local features of points. Wu et al [16] introduced a density-weighted convolution PointConv that can fully simulate 3D continuous convolution, compensating for the effects of uneven sampling and further improving the quality of captured features.

According to the relationship between local characteristics and global characteristics, we designed a deep neural network based on a dual attention mechanism for 3D point cloud classification. The network uses two parallel encoding mechanisms, self-attention coding and multi-head graph attention coding, to learn fine-grained features in local regions and context geometric features between local regions, and uses the VLAD layer to extract deeper point cloud features. In addition, to ensure the integrity of the geometry, an effective significant point embodiment notation is introduced, setting each point to a different weight. After many experiments, our model obtained a high classification accuracy on the ModelNet40 dataset.

III. METHOD

To better obtain the fine-grained features of 3D point clouds and to better combine local and global features, this work proposes a deep neural network based on a dual-attention mechanism and VLAD for point cloud classification. Unlike PointNet [9], which extracts features from isolated points, this network uses self-attention mechanism coding to make each point perform feature interaction with the remaining points of the point clouds. Moreover, this network constructs neighbouring geometric graphs and then performs convolution on the features of the local graphs to gather powerful features for point cloud classification.
A. Dual attention mechanism network

An ideal 3D model recognition algorithm should not only capture its global spatial context information but also fully mine local fine-grained information; these two tasks are complementary to each other. For this reason, in this paper, a feature extractor DAMN layer based on a self-attention mechanism and a graphical attention encoding mechanism is constructed to learn global spatial contextual features and local detail features of the point cloud, respectively. Figure 1 shows the network structure. The points’ number is recorded as N, MLP is the multilayer perceptron machine, and the number in {} is the number of convolution cores.

![Dual attention mechanism network](image)

**FIGURE 1.** Dual attention mechanism network.

1) Self-attention mechanism

In today’s few years, self-attention networks have attracted much attention because they can obtain discriminant features in the area of interest. To date, models based on self-attention have been widely used in machine translation, subtitle generation [17], speech recognition [18], antagonistic networks [19] and other tasks. Attention gives the model the ability to distinguish between different discerning points and finds the focus of what should be focused on from the information. Attention mechanism models applied to the field of natural language usually consist of a set of encoders and decoders. The encoder is responsible for building context relationships, mapping words and sentences, and forming features. The decoder is responsible for interpreting the features and restoring the semantic information. The attention mechanism is applied in this paper to aggregate local and global information through encoding, to avoid using convolution stacking to increase the field of receptivity layer by layer, and to bypass the point cloud hierarchy problem. Therefore, we use the coding part of the attention mechanism to obtain the feature vectors that can be used to classify point clouds. The general definition of the attention mechanism [20] is shown in equation 1:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$$  \hspace{1cm} (1)

Where $Q \in \mathbb{R}^{n \times d_k}$, $K \in \mathbb{R}^{m \times d_k}$, $V \in \mathbb{R}^{n \times d_f}$. $q_t \in Q$. The resulting code obtained for a single input vector $q_t$ can be expressed as in equation 2:

$$\text{Attention}(Q, K, V) = \sum_{s=1}^{m} \frac{1}{Z} \exp\left(\frac{<q_t, k_s>}{\sqrt{d_k}}\right) v_s$$  \hspace{1cm} (2)

2) Graph attention coding

In practical applications, such as autonomous driving, there are very large point clouds of data. The local structure of a point cloud represented by the directed k-nearest neighbour graph $G = (V, E)$, where each point is represented by $V_i$, $i \in \{1,2, ..., N\}$, the set of neighbours of point $i$ is denoted $N_i$ and the edges connecting pairs of neighbouring points are denoted $y_{ij} = (x_i - x_j)$. We define the adjacency feature as $y_{ij} = (x_i - x_j)$, where $i \in V, j \in N_i$ and $x_i$ denotes the neighbouring point $i$ to Point $x_i$, $x_i, x_j$ denotes a point and its corresponding neighbours, and $y_{ij}$ is the corresponding edge. The attention coefficients of adjacent pairs are first fused with the self-attention coefficient and the local attention coefficient via a leaky ReLU activation function, and then further normalized by a softmax function.

![Attention coefficient generation](image)

**FIGURE 2:** An illustration of attention coefficient generation.

We introduce graph attention coding to obtain the attention coefficients of each point’s neighbour so as to give different attention to different neighbourhoods, as shown in Figure 2. Here, $x_i$ and $x_{ij}$ respectively represents a certain point and its corresponding neighbour, and $y_{ij}$ denotes the corresponding edges.

We encode point cloud’s nodes and edges with an output dimension $F_0$ defined by equations 3 and 4. Where $h()$ represents a parametrized nonlinear function that acts as a single-layer neural network on our overall network architecture, a set of learnable filter parameters noted as $\theta$.

$$x_i' = h(x_i, \theta)$$  \hspace{1cm} (3)

$$y_{ij}' = h(y_{ij}, \theta)$$  \hspace{1cm} (4)

The equation 5 defines the attention coefficients, which are generated by fusing self-coefficients $h(x_i', \theta)$ and local-coefficients $h(y_{ij}', \theta)$, where a single-layer neural network
with a 1-dimensional output is used to represent the self-coefficients and local-coefficients, the nonlinear activation function Leaky ReLU is noted as $\text{LeakyReLU}(\cdot)$.

$$c_{ij} = \text{LeakyReLU}\left(h(x'_i, \theta) + h(y'_{ij}, \theta)\right)$$  \hspace{1cm} (5)

We normalized the attention coefficients of all neighbours of each point using the softmax function, which is known as equation 6, to keep the comparison of attention coefficients between neighbours of different points consistent.

$$a_{ij} = \frac{\exp(c_{ij})}{\sum_{k \in N_i} \exp(c_{ik})}$$  \hspace{1cm} (6)

The goal of each single-headed attention convolution is to calculate the fine-grained local characteristics of each point. For this purpose, as shown in equation 1, we calculate a linear combination using the resulting normalization coefficients. The output of the single-headed graph attention mechanism is the attention feature $\hat{x}_i \in \mathbb{R}^F$ and the graph features encoded from the edge of the graph.

$$\hat{x}_i = f\left(\sum_{j \in N_i} a_{ij} y'_{ij}\right)$$  \hspace{1cm} (7)

We connect $M$ independent single-headed graph attention mechanisms in series to produce multiple attention features with $M \times F_0$ channels to obtain point cloud’s sufficient structural features and stabilize the deep neural network. The equation is defined as equation 7. The outputs of the multihead graph attention mechanism is multiple attention features and multiple graph features, fusing the attention and graph features of each head, as shown in Figure 3.

$$\hat{x}'_i = \big|\big| \hat{x}'_i^{(m)} \big|\big|_m$$  \hspace{1cm} (8)

where $\big|\big|$ is the join operation on the feature channels, and the attentional feature of the $M$th head is noted as $\hat{x}'_i^{(m)}$, and the heads’ total number is noted as $M$.

**Figure 3:** Multi-head Graph Attention Mechanism

### B. VLAD layer deep feature extraction

In recent years, vector of locally aggregated descriptors (VLAD) and product quantization techniques in large-scale image retrieval have been extensively researched, with a focus on improving aggregate representations and optimising indexing schemes using more discriminative local features. With the significant effectiveness of deep neural networks in various fields, there has been widespread interest in how VLAD can be integrated with deep neural networks. In [21], combining the existing PointNet [9] with NetVLAD [22], PointNetVLAD is proposed to retrieve large-scale location identification. PointNetVLAD quickly identifies locations in large scenes by combining the existing PointNet [9] and NetVLAD [22]. Inspired by PointNetVLAD [21] and NetVLAD [22], we can use the relationship between the low-level geometric descriptors of each point and some visual vocabulary to describe high-level semantic features, as shown in Figure 4.

As shown in the picture above, given 6 2-dimensional points $\{p_i|i \in [1,6]\}$ as input and the VLAD parameters are 4 visual words (“cluster centres”) $\{c_k|k \in [1,4]\}$, a $4 \times 2$ matrix represents the image $V$ as the output of the VLAD. The following equation computes the components $(j, k)$ of $V$.

$$V(j, k) = \sum_{i=1}^{6} a_k(p_i)(p_i(d) - c_k(d)) \hspace{1cm} d \in [1,2]$$  \hspace{1cm} (9)

where the $d$-th dimension of the $k$th cluster centre is denoted $c_k(d)$, and the $d$-th dimension of the $i$-th point $p_i(d)$. The relationship of the $K$-th visual word to the descriptor $p_i$ is noted as $a_k(p_i)$, with 1 representing the cluster $c_k$ being the cluster closest to the descriptor $p_i$ and the rest being 0. Thus, PointNetVLAD is designed to aggregate local features into visual words.

**Figure 4:** The relationships between several visual words and the low-level geometric descriptors of each point

To clearly describe the VLAD module of this manuscript, we then illustrate the problem in the case of two dimensions. As shown in Figure 4(b), we take 6 2-dimensional points $\{p_i|i \in [1,6]\}$ as input for the VLAD module. Meanwhile, initializing 4 visual words (clustering centres), which are learnable parameters by back propagation, denoted as $\{c_k|k \in [1,4]\}$. Each visual word $c_k$ is assigned multiple points $p_i$ and the residual vector $p_i - c_k$ represents the difference between point $p_i$ and the visual word $c_k$ represents the relationship between the 4 visual words and the $i$-th point $p_i$ . The components $(j, k)$ of $r$ is calculated as follows:

$$r_{n,d} = \sum_{m=1}^{M} a_k(c_k)(p_i(d) - c_k(d)) \hspace{1cm} d \in [1,2]$$  \hspace{1cm} (10)

The $d$-th dimension of the $i$-th point is denoted as $p_i(d)$ and the $d$-th dimension of the $k$-th visual word is denoted as $c_k(d)$ and $a_k(c_k)$ represents the attention coefficients. The attention coefficients $a_k(c_k)$ are utilized to weight the relationship between the $k$-th visual word and the $i$-th point.

To control the selection of different dimensional features for the VALD layer, we designed a $top-k$ for the VLAD module to control dimensionality (note that $top-k=2$ was set as the default value in the overall architecture of our network) to consider the effect of different relationships between each point and visual words on high-level semantic features, since each point is only strongly related to a few visual words. Therefore, the components $(j, k)$ of $r$ is computed as follows:
\[ r_{id} = \sum_{k=1}^{2} a_i(c_k(p_i(d)) - c_k(d)) \quad d \in [1, 2] \quad (11) \]

Top-\( k \) completes the overlap between different visual words on the one hand, and realises that the output of the VLAD module has the same dimension as the input (vectors have different dimensions) on the other hand, a shared FC layer is used, which eventually aggregates the top-\( k \) transform features to increase the non-linear transform of the network.

C. Deep neural network for point cloud classification based on dual attention mechanism and VLAD

The structure of DAMVnet is shown in the following figure. The DAMLayer, is a dual-attention mechanism module, which consists of self-attention mechanism coding and graphic attention coding. The model input is \( N \times 3 \) dimensional point cloud matrix passes through the DAMLayer layer first. The self-attention mechanism mainly completes the linear mapping of the input vectors \( Q, K, \) and \( V \). The values of these vectors are the neighbour vector matrix \( X \) of the 3-dimensional point cloud. The graphic attention mechanism mainly extracts fine-grained features of multiheaded context from point cloud models. Then the extracted features are fused, and the weight matrix of the calculated key points is added to make it contain both global features and local area information. Then the max pooling symmetric function fused the features, and the VLAD layer further extracted the high-level 8 feature. MLP \{512, 256, C\} denotes the fully connected layers which have 512, 256 and \( C \) neurons. Among them, \( C \) means the categories’ number. In this experiment, \( C \) is set to 40. Finally, three layers of MLP are used as classifiers to obtain classification results.

![DAMVnet network framework](image)

**FIGURE 5:** DAMVnet network framework

### IV. RESULTS AND ANALYSIS

To evaluate the deep level point cloud feature learning network based on the attention mechanism, the ModelNet40 dataset consists of 40 categories and 12311 CAD models, is used for training and testing our proposed model. According to the official training and testing standards, there are 2468 shapes for testing and 9843 shapes for training. The implementation details of our proposed method will be briefly discussed in the Section B. In Section C, we have done a considerable assessment about the superiority among our method and many advanced 3D point cloud classification method on the ModelNet40 benchmark. In Section D, We performed ablation studies and analyses that broadly demonstrated the effectiveness of our method for each individual component. As shown in the table below, the details of the experiments are listed.

<table>
<thead>
<tr>
<th>Table I</th>
<th>STYLES EXPERIMENTAL EQUIPMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>ModelNet40</td>
</tr>
<tr>
<td>Num point</td>
<td>512</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adam</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.001</td>
</tr>
</tbody>
</table>

#### A. Network architecture

The classification model that we proposed can be viewed in the Figure 1. The network uses 512 points as input and uses a DAM layer to fully mine its fine-grained features. The GAM part adopts 4 multi-head mechanisms and uses the DAM layer to obtain self-attention, multi-attention and multigraph features. The multi-head GAM generates the multi-attention feature of \( M \times F_0 \) and sets the number of headers to \( M = 4 \), we set up the number of encoding channels for 16, and the number of neighbours for 20. Then, we use the shared MLP layer to aggregate the features. The aggregated features are input into the VLAD layer through the MLP layer and pooling layer. There are 40 categories needs to be distinguished. So finally we use the MLP layers\{512, 256, 40\} to transform the global features to resolve the classification.

#### B. Comparison with other 3D point cloud classification methods

For the purpose of demonstrating our proposed algorithm’s superiority, current mainstream algorithms, under the same experimental conditions, are compared with the novel algorithm we proposed on the ModelNet40 dataset, and the evaluation criterion is the classification accuracy of 3D model recognition. The experimental results are shown in Table II.
The classification accuracy of other algorithms is obviously lower than this algorithm. Compared with PointNet [9], the classification accuracy is improved by 3.25 percentage points, and compared with the most advanced DGCNN [3], the recognition accuracy is improved by 0.25 percentage points. The reason is that DAMVNet can not only mine the fine-grained features in local regions but also capture the global structure information by combining the correlation between local regions, which enhances the network’s ability to distinguish classes with similar geometry.

Table II
NETWORK PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS ON MODELNET40 (%)

<table>
<thead>
<tr>
<th>Network</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>avg.class</td>
</tr>
<tr>
<td>3DshapeNets[23]</td>
<td>77.3</td>
</tr>
<tr>
<td>OctNet[12]</td>
<td>83.8</td>
</tr>
<tr>
<td>VoxNet[24]</td>
<td>83.0</td>
</tr>
<tr>
<td>O-CNN[25]</td>
<td>-</td>
</tr>
<tr>
<td>ECC[26]</td>
<td>83.2</td>
</tr>
<tr>
<td>So-Net[27]</td>
<td>87.3</td>
</tr>
<tr>
<td>PointNet[9]</td>
<td>86.2</td>
</tr>
<tr>
<td>PointNet++[2]</td>
<td>-</td>
</tr>
<tr>
<td>Kd-Net(depth 10) [11]</td>
<td>86.3</td>
</tr>
<tr>
<td>A-SCN[28]</td>
<td>87.4</td>
</tr>
<tr>
<td>DGCNN[3]</td>
<td>90.2</td>
</tr>
<tr>
<td>DAMVNet (Ours)</td>
<td>89.41</td>
</tr>
</tbody>
</table>

Figure 6: Accuracy comparison chart with pointNet and KP-PointNetVLAD

C. Module ablation analysis and multi-head mechanism analysis

This work explores the impact of key points, DAM layers and VLAD module to the original network can effectively improve the classification accuracy. From Table III, there has a conclusion that the highest classification accuracy is achieved when the key points, the DAM module and the VLAD module are present at the same time.

Table III
IMPACT OF THE KEY POINT, DAM LAYER AND VLAD MODULE ON THE CLASSIFICATION OF THE 3D POINT CLOUDS

<table>
<thead>
<tr>
<th>Key point</th>
<th>DAM module</th>
<th>VLAD module</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>√</td>
<td>-</td>
<td>87.86</td>
</tr>
<tr>
<td></td>
<td>√</td>
<td>√</td>
<td>89.41</td>
</tr>
<tr>
<td></td>
<td>√</td>
<td>√</td>
<td>91.15</td>
</tr>
<tr>
<td></td>
<td>√</td>
<td>√</td>
<td>92.45</td>
</tr>
</tbody>
</table>

Figure 7: Comparison of the accuracy in the ablation experiments

At the same time, this work continues to explore the impact of different numbers of headers and coding signature channels on the classification accuracy, as shown in Table IV. Due to the experimental results, we can obtain that the classification accuracy of the model can be improved by increasing the number of header and coded signature channels, which means the network’s ability of extracting more adequate features is promoted by increasing the number of header and coded signature channels. When the number of headers is 4 and the number of coding signature channels is 16, the classification accuracy is the best. The network, which is already saturated with extracted features, will decrease its performance on account of the number of header or code-signature channels. Increasing the number of headers or coded signature channels will not improve the learning capability of the network, but increase a quantity of the redundant parameters and reduces the network’s performance.

Table IV
DIFFERENT COMBINATIONS OF SEVERAL HEADERS AND SOME CODING SIGNATURE CHANNELS(%)
<table>
<thead>
<tr>
<th>Heads</th>
<th>Channels</th>
<th>Accuracy (overall)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>91.51</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>92.00</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>92.20</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>92.45</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>91.11</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>92.28</td>
</tr>
</tbody>
</table>

IV. CONCLUSION

This paper builds a novel model, based on dual attention mechanism and VLAD, for the classification of point cloud. By building a DAM layer to capture the fine-grained spatial context of point clouds, the self-attention mechanism can capture the local and global connections of 3D point clouds in one step, which accelerate the classification capacity of the network. At the same time, the multi-head mechanism is introduced to better promote the accuracy of the network classification. Extensive experimental data imply that on the ModelNet40 dataset, the algorithm achieves a classification accuracy of 92.45% for 3D point clouds, which is superior to the current mainstream methods in classification accuracy. Compared with the current mainstream methods, it has a strong advantage in classification accuracy. In the experiments, we always input 512 points due to the limitation of the equipment. At a later stage, if available, we will adjust the number of input points such as input 1024, to further validate the experimental results.

ACKNOWLEDGMENT

The authors feel grateful for Mr. Zhang’s assistance in the experiment.

FUNDING

This study was supported by grant from National Natural Science Foundation of China No. 71872131. In addition, is also supported by grant from Shantou University SRIG No. 2007 and Science and technology project of Zhejiang Province No.2018C01080.

ETHICAL APPROVAL

The data set used in this paper does not involve ethical aspects.

INFORMED CONSENT

The data set used in this paper does not involve informed consent aspects.

CONFLICT OF INTEREST

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

AUTHOR CONTRIBUTIONS


REFERENCES

Y. Cui, X. Liu, H. Liu, J. Zhang, A. Zare, and B. Fan, "Point

M. Atzmon, H. Maron, and Y. Lipman, "Point
cloud auto-encoder via deep grid deformation," in Proceedings of the IEEE Conference on

M. A. Uy and G. H. Lee, "Pointnetvlad: Deep point
cloud based retrieval for large-scale place

Guodao Zhang, received the B.S degree from Wenzhou University, Wenzhou, China, in 2010, the M.S. degree from Nanjing University of Science & Technology, Nanjing, China, in 2016. And now pursuing the Ph.D. degree from Zhejiang University of Technology, Hangzhou, China. He is a CCF student member and No.97716G. His main research direction is computational intelligence, process modeling, data mining, virtual reality.

Jian Zhou, received the B.S degree from Zhejiang University of Technology, Hangzhou, China, in 2021. And now pursuing the M.S degree from Zhejiang University of Technology, Hangzhou, China. His main research direction is data mining, virtual reality, deep learning.

Chen, Ping-Kuo received the Ph.D degree in Industrial Engineering and Management from National Chiao Tung University in Hsinchu, Taiwan in 2008. He is currently the Professor at the Business School of Shantou University. Dr. Chen’s research interests include manufacturing strategy, production and operations management, supply chain management, and green supply chain.


14] M. Atzmon, H. Maron, and Y. Lipman, "Point


17] K. Xu et al., "Show, attend and tell: Neural image caption generation with visual attention," in International conference on machine learning, 2015, pp. 2048-2057: PMLR.


