OVPT: Optimal Viewset Pooling Transformer for 3D Object Recognition

Wenju Wang[1][0000−0002−8549−4710], Gang Chen[1][0000−0002−4771−169X] *, Haoran Zhou[1][0000−0002−3530−0500], and Xiaolin Wang[1][0000−0001−8924−9352]

University of Shanghai for Science and Technology, Shanghai 200093, China
203592861@st.usst.edu.cn

Abstract. The current methods for multi-view-based 3D object recognition have the problem of losing the correlation between views and rendering 3D objects with multi-view redundancy. This makes it difficult to improve recognition performance and unnecessarily increases the computational cost and running time of the network. Especially in the case of limited computing resources, the recognition performance is further affected. Our study developed an optimal viewset pooling transformer (OVPT) method for efficient and accurate 3D object recognition. The OVPT method constructs the optimal viewset based on information entropy to reduce the redundancy of the multi-view scheme. We used convolutional neural network (CNN) to extract the multi-view low-level local features of the optimal viewset. Embedding class token into the headers of multi-view low-level local features and splicing with position encoding generates local-view token sequences. This sequence was trained parallel with a pooling transformer to generate a local view information token sequence. At the same time, the global class token captured the global feature information of the local view token sequence. The two were aggregated next into a single compact 3D global feature descriptor. On two public benchmarks, ModelNet10 and ModelNet40, for each 3D object we only need a smaller number of optimal viewsets, achieving an overall recognition accuracy (OA) of 99.33% and 97.48%, respectively. Compared with other deep learning methods, our method still achieves state-of-the-art performance with limited computational resources. Our source code is available at https://github.com/shepherds001/OVPT.

1 Introduction

With the rapid development of 3D acquisition technology, 3D scanners, depth scanners, and 3D cameras have become popular and inexpensive. The acquisition of 3D data such as point clouds and meshes has become more convenient and accurate [1]. These factors have promoted the widespread application of 3D data–based object recognition techniques in the fields of environment perception for autonomous driving [2], object recognition for robots [3], and scene understanding for augmented reality [4]. Therefore, 3D object recognition has become a hotspot for current research.

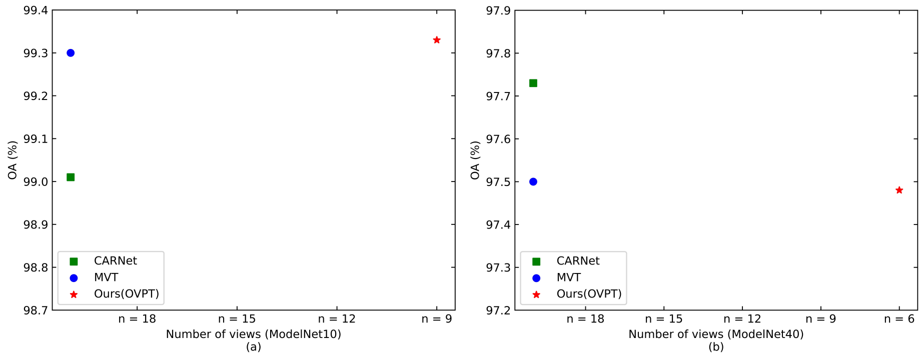


Fig. 1. Comparison with the state-of-the-art approaches in terms of recognition performance, and the number of views required per 3D object.

Deep learning–based methods have become mainstream research techniques for 3D object recognition tasks. In general, these methods can be divided according to the type of data input to the deep neural network: voxel-based methods [5–12], point-cloud-based methods [13–20], and multi-view-based methods [21–30].

Multi-view-based methods render 3D data objects into multiple 2D views, so they no longer need to rely on complex 3D features. Instead, the rendered multi-views are fed into multiple well-established image classification networks to extract multi-view low-level local features. Finally, the multi-view low-level local features are aggregated into global descriptors to complete the 3D object recognition task. Especially when 3D objects are occluded, such methods can complement each other's detailed features of 3D objects according to views from different perspectives. Compared with voxel-based and point cloud-based methods, the multi-view-based method achieves the best 3D object recognition performance.

However, this type of method still has the shortcomings that feature information cannot be extracted for all views simultaneously during training and that related feature information between multiple views cannot be efficiently captured. In addition, there is redundancy when rendering 3D objects into multiple views. The relevant feature information between multiple views is indispensable for aggregating the multi-view local features into a compact global descriptor. The omission of these relevant features is why it is difficult to improve the recognition accuracy of this type of method. The view redundancy problem increases unnecessary network running time and affects final recognition accuracy. This paper researches this and proposes the optimal viewset pooling transformer (OVPT) method. Our main contributions are summarized as follows:

- The method to construct the optimal viewset based on information entropy solves the view redundancy problem of the multi-viewpoint rendering method [24], which reduces computational cost of the network.

- The proposed multi-view low-level local feature token sequence generation method introduces transformers into 3D object recognition tasks. This method combining transformers and CNNs is able to process all views and capture relevant features among views while maintaining strong inductive bias capability.

- The pooling transformer-based global descriptor method can improve the insufficient local feature aggregation ability for insufficient transformer training with small dataset. This method is able to aggregate multi-view low-level features coming from local and global respectively into a compact global descriptor.

- We conducted extensive experiments on ModelNet10 and ModelNet40 datasets to verify the performance our OVPT method. This OVPT method can achieve respectively 99.33% and 97.48% of overall recognition accuracy (OA) in two datasets, only requiring a smaller number of optimal viewsets. Compared with other state-of-the-art methods, our OVPT network achieves the best performance.

2 Related Work

Voxel-based Methods. VoxNet [5] uses 3D CNN to extract voxelized 3D object features and processes non-overlapping voxels through max pooling. However, it cannot compactly represent the structure of 3D objects. Therefore, Kd-network [9] was proposed. It creates a structural graph of 3D objects based on a Kd-tree structure, and computes a sequence of hierarchical representations in a feed-forward bottom-up fashion. Its network structure occupies less memory and is more computationally efficient, but loses information about local geometry. These voxel-based methods solve the problems of large memory footprint and long training time of point cloud voxelization, but still suffer from the problems of lost information and high computational cost.

Point Cloud-based Methods. Point cloud voxelization inevitably loses information that may be essential. Some methods consider processing point clouds directly. PointNet [13] was the earliest method to process point clouds directly. It uses T-Net to perform an affine transformation on the input point matrix, and extract each point feature through a multi-layer perceptron. But it could not capture the local neighborhood information between points. Thus, PointNet++ [14] was developed, which constructs local neighborhood subsets by introducing a hierarchical neural network, and then extracted local neighborhood features based on PointNet. PointNet++ solved the local neighborhood information extraction of PointNet to a certain extent. Dynamic Graph CNN (DGCNN) [19] uses EdgeConv to build a dynamic graph convolutional neural network for object recognition. EdgeConv could extract local domain feature information, and the local shape features of the extracted point cloud could keep the arrangement invariance. Although the point cloud-based method can directly process the point cloud to reduce the loss of information, its network is often complex, the training time is long, and the final recognition accuracy is not high enough.

Multi-view-based Methods. Multi-view-based approaches render 3D data objects into multiple 2D views, so they no longer need to rely on complex 3D features. This class of methods achieves the best 3D object recognition performance. Multi-view Convolutional Neural Networks (MVCNN) [21] employs a 2D CNN network to process rendered multiple views individually. MVCNN then uses view pooling to combine information from multiple views into a compact shape descriptor. However, it lost the position information of the views when pooling multiple views. Multi-view convolutional neural network (GVCNN) [23] mines intra-group similarity and inter-group distinguishability between views by grouping multi-view features to enhance the capture of location information. Hierarchical Multi-View Context Modeling (HMVCM) [30] adopted adaptive calculation of feature weights to aggregate features into compact 3D object descriptors. This type of hierarchical multi-view context modeling used a module that combined a CNN and a Bidirectional Long Short-Term Memory (Bi-LSTM) network to learn the visual context features of a single view and its neighborhood. This network had an overall recognition accuracy (OA) on the ModelNet40 dataset of 94.57%. However, it did not consider the local features of all views in parallel during training, and lost relevant information between views, so the aggregated global descriptors were not sufficiently compact. Its 3D object recognition accuracy has room for improvement.

3 Methods

The OVPT network proposed in this paper is shown in Fig. 2. OVPT has three parts: (a) Optimal viewset construction based on information entropy; (b) Multi-view low-level local feature token sequence generation; (c) Global descriptor generation based on the pooling transformer.

3.1 Optimal Viewset Construction based on Information Entropy

The input stage of the OVPT network renders 3D objects (represented by point cloud or meshes) into multiple 2D views. This study selected a mesh representation with higher accuracy for 3D object recognition. Of course, 3D objects in the form of point clouds can also be reconstructed into mesh forms [31–33]. The specific process is as follows:

Multi-view Acquisition. For a 3D object O, different 2D rendering views $V = \{v_1, ...v_i..., v_N\}$ can be obtained by setting the camera in different positions, where v_i represents the view taken from the i-th viewpoint, $v_i \in \mathbb{R}^{C\times H\times W}$.

$$[v_1, ...v_i..., v_N] = Render(O) \tag{1}$$

We use the dodecahedron camera viewpoint setting [24]. It places the 3D object in the center of the dodecahedron then sets the camera viewpoint at the

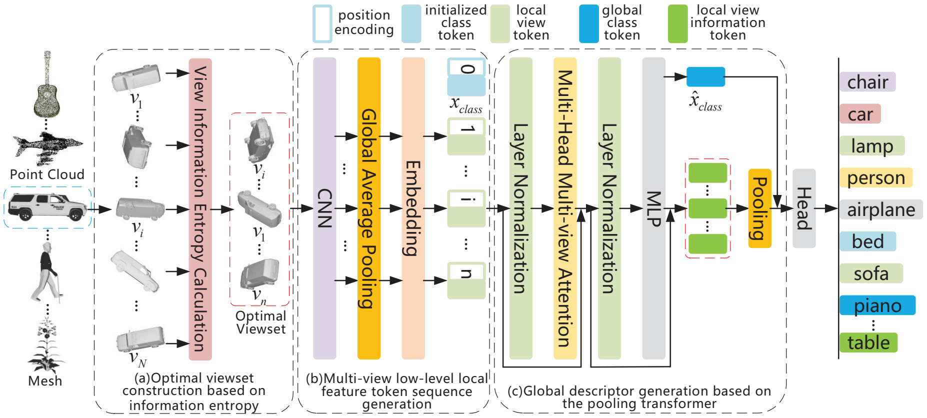


Fig. 2. The architecture of the proposed optimal viewset pooling transformer (OVPT).

vertices of the dodecahedron. This setup evenly distributes the camera viewpoints in 3D space, capturing as much global spatial information as possible and reducing information loss. However, among the 20 views rendered in this way, there are always duplicates. This may lead to redundant features being extracted by the deep neural network and increase the running time of the network, eventually decreasing the accuracy of the recognition.

Optimal Viewset Construction. Information entropy can highlight the grayscale information for the pixel position in the view and the comprehensive characteristics of the grayscale distribution in the pixel neighborhood. Assuming a given amount of information contained in the view, information entropy can be an effective means to evaluate view quality. Aiming at the problem of repetitive in the viewpoint settings of the dodecahedron camera, we introduce the information entropy [34] of 2D views as an evaluation criterion to construct the optimal viewset to reduce redundant views. Different from the previous viewpoint selection methods based on information entropy [35, 36], our method does not require human intervention and only considers the quality of the view itself without calculating the projected area of the viewpoint, which is more reliable. Dodecahedron camera viewpoint settings and optimal viewset are shown in Fig. 3. The optimal viewset is constructed as follows:

(1) Information entropy calculation of N views ($N = 20$): H_i represents the information entropy of the i-th view v_i. The specific calculation is shown in formulas (2) and (3):

$$P_{a,b} = f(a,b)/W \cdot H \tag{2}$$

$$H_i = -\sum\nolimits_{a=0}^{255} P_{a,b} log P_{a,b} \tag{3}$$

where (a, b) is a binary group, a represents the gray value of the center in a sliding window, and b is the average gray value of the center pixel in the window. $P_{a,b}$ is the probability that (a, b) appears in the full view v_i. $f(a, b)$ is the number of

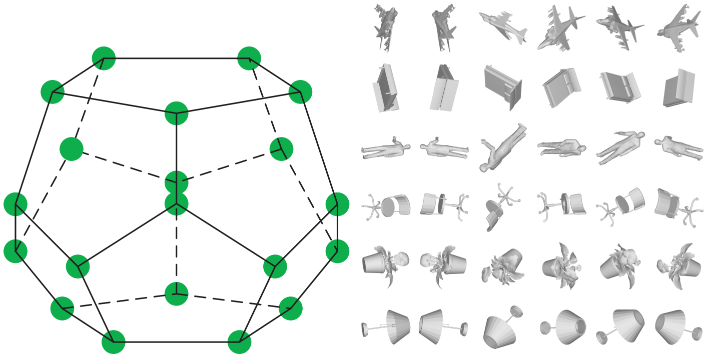


Fig. 3. Dodecahedron camera viewpoint settings [24] and optimal viewset.

times the binary group (a, b) appears in the full view v_i. W and H represent the width and height of the view.

(2) Rank the information entropy values $H_i(i = 1, ..., N, N = 20)$. (3) Construct the optimal viewset: The views with the best information entropy ranking n $(n < N, n = 6, 9$ in this paper) are regarded as the optimal viewset. When $n = 1$ in the optimal viewset, the single view with the highest information entropy value is selected, which is called the optimal view.

3.2 Multi-View Low-level Local Feature Token Sequence Generation

Because the transformer was first proposed for natural language processing, its input requirements are two-dimensional matrix sequences. For the optimal viewset $V = \{v_1, ...v_i..., v_n\}$, $v_i \in \mathbb{R}^{C \times H \times W}$, where n is the number of views, C is the number of channels, H is the height of the image, and W is the width of the image. Therefore, the obtained view cannot be directly input to the transformer, and it needs to be flattened into a local view token sequence $X = \{x_1, ...x_i..., x_n\}$. x_i represents the local view token generated by the i-th view, $x_i \in \mathbb{R}^{1 \times D}$, where D is the constant latent vector size used in all transformer layers. To this end, this paper proposes a multi-view low-level local feature token sequence generation method. The specific process is as follows:

Low-level Local Feature Extraction. We used multiple CNNs pretrained on ImageNet [37] to extract low-level local features of multi-view $V = \{v_1, ...v_i, ...v_n\}$. Any well-established 2D image classification network can be used as a multi-view low-level feature extractor.

Local View Token Sequence Generation. After the multi-view low-level local features are extracted, a local view token sequence $X = \{x_1, ...x_i..., x_n\}$ is generated by embedding.

Addition of Initialized Class Token and Position Encoding. After obtaining the local view token sequence $X = \{x_1, ...x_i..., x_n\}$, it is also necessary to add an initialized class token x_{class} [38] to the header of the local view token sequence, and concatenate them with position encoding E_{pos} [38]. where x_{class}

is a random initial value that matches the dimension of the local view token, $x_{class} \in \mathbb{R}^{1\times D}$, E_{pos} is used to save location information from different viewpoints x_i, $E_{pos} \in \mathbb{R}^{(n+1)\times D}$. Finally, the multi-view low-level local feature token sequence X_0 can be generated.

$$X_0 = [x_{class}; x_1, ...x_i..., x_n] \oplus E_{pos}, X_0 \in \mathbb{R}^{(n+1)\times D} \tag{4}$$

3.3 Global Descriptor Generation based on the Pooling Transformer

The method uses a pooling transformer to aggregate X_0 into one compact 3D global descriptor in two steps:

Global Feature Information Generation based on Transformer. The generation of the global feature information has three steps: layer normalization [39] processing; multi-head multi-view attention calculation; and residual connection and the use of multi-layer perceptron:

(1) Layer Normalization Processing: X_0 is input to the pooling transformer as a sequence of multi-view low-level local feature tokens. Before the calculating Multi-Head Multi-View Attention (MHMVA), X_0 undergoes Layer Normalization (LN), see formula (5):

$$\hat{X}_0 = LN(X_0), \hat{X}_0 \in \mathbb{R}^{(n+1)\times D} \tag{5}$$

(2) Multi-Head Multi-View Attention Calculation: We use the normalized $\hat{X}_0$ to generate $Query$, Key , and $Value$ through linear transformations. MHMVA performs multiple parallel Multi-View Attention (MVA). The inputs q_i, k_i and v_i of each MVA can be obtained by equally dividing the $Query$, Key and $Value$ vectors, where $q_i \in \mathbb{R}^{(n+1)\times \frac{D_Q}{N}}$, $k_i \in \mathbb{R}^{(n+1)\times \frac{D_K}{N}}$, $v_i \in \mathbb{R}^{(n+1)\times \frac{D_V}{N}}$, $D_Q = D_K = D_V$ represents the vector dimension of Query, Key and Value respectively. We obtained multiple MVA according to the number of heads N, and multiple subspaces can be formed. Therefore, MHMVA can pay to take the information of various parts of the input features into account. The calculation result X_i^{MVA} of each MVA is obtained from formula (6):

$$X_i^{MVA} = Softmax(\frac{q_i k_i^T}{\sqrt{D_K/N}})v_i, X_i^{MVA} \in \mathbb{R}^{(n+1)\times \frac{D_K}{N}} \tag{6}$$

Concat is performed on each X_i^{MVA} after calculation, and the MHMVA calculation is finally completed after a linear transformation, as shown in formula (7):

$$X_{MHMVA} = h_{\Theta}(\sum_{i=1}^{N} X_i^{MVA}), X_{MHMVA} \in \mathbb{R}^{(n+1)\times D} \tag{7}$$

where h_{Θ} represents a linear function with dropout.

(3) Residual Connection and the Use of Multi-Layer Perceptrons: The X_{MHMVA} obtained after the MHMVA calculation uses residual connections [40] to avoid vanishing gradients.

$$X_1 = X_{MHMVA} + X_0, X_1 \in \mathbb{R}^{(n+1)\times D} \tag{8}$$

After X_1 is obtained, it is also processed by layer normalization and input to a multi-layer perceptron (MLP). Because MHMVA does not fit the complex process sufficiently, MLP is added after them to enhance the model's generalization. The MLP consists of linear layers that use the GELU [41] activation function shown in formula (9):

$$MLP(x) = GELU(W_1 x + b_1)W_2 + b_2 \tag{9}$$

where W_1 and b_1 are the weights of the first fully connected layer, W_2 and b_2 are the weights of the second fully connected layer, and x represents the input feature information.

There is also a residual connection between the output of MLP and X_1, and the calculation is given by formula (10):

$$\hat{X_1} = MLP(LN(X_1)) + X_1 \tag{10}$$

The final $\hat{X_1}$ is the output of global feature information generation based on the transformer method, where $\hat{X_1} \in \mathbb{R}^{(n+1)\times D}$. It consists of a global class token $\hat{x}_{class}$ and the local view information token sequence $\{\hat{x}_1, ...\hat{x}_i..., \hat{x}_n\}$, where $\hat{x}_{class} \in \mathbb{R}^{1\times D}, \hat{x}_i \in \mathbb{R}^{1\times D}$. The global class token $\hat{x}_{class}$ stores the global feature information of the local view token sequence.

Local View Information Token Sequence Aggregation based on Pooling. After parallel transformer training, the global class token $\hat{x}_{class}$ saves the global feature information of the local view token sequence, but the single best local view information token may be lost. It is very efficient to aggregate this part of the information into a 3D global descriptor. Our local view information token sequence aggregation based on the pooling method can solve this problem. It can simultaneously capture the single best local view information token while preserving the global feature information of the local view token sequence. This method pools the local view information token sequence $\{\hat{x}_1, ...\hat{x}_i..., \hat{x}_n\}$ to obtain the best local view information token then splices the best local view information token and the global class token $\hat{x}_{class}$. After these processes, we can aggregate multi-view low-level local feature token sequences locally and globally, then generate a more compact 3D global descriptor Y, $Y \in \mathbb{R}^{1\times D}$. The 3D global descriptor is input to the head layer to complete the object recognition task.

$$Y = max[\hat{x}_1, ...\hat{x}_i..., \hat{x}_n] + \hat{x}_{class} \tag{11}$$

4 Experimental Results and Discussion

4.1 Dataset

ModelNet [6] is a widely used 3D object recognition dataset, popular for its diverse categories, clear shapes, and well-built advantages. The two benchmark datasets ModelNet40 and ModelNet10 are its subsets. Among them, ModelNet40 consists of 40 categories (such as airplanes, cars, plants, and lights), with 12,311 CAD models, including 9,843 training samples and 2,468 test samples. ModelNet10 consists of 10 categories, with 4,899 CAD models, including 3,991 training samples and 908 test samples.

4.2 Implementation Details

We conducted extensive comparative experiments using PyCharm on a computer with the Windows10 operating system. The relevant configuration of this computer is as follows: (1) Central Processing Unit (CPU) was an Intel(R) Xeon CPU @2.80 GHz, (2) Graphic Processing Unit (GPU) was RTX2080, (3) Random Access Memory (RAM) was 64.0 GB, and (4) Pytorch 1.6 was used. For all our experiments, the learning rate was initialized to 0.0001, the epoch was set to 20, the batch size is set to 8 by default. On ModelNet10 and ModelNet40, the CNNs are resnet34 [40] and densenet121 [42], respectively. We used the Adam [43] algorithm to optimize the network structure based on the learning rate decay and the L2 regularization weight decay strategy to avoid overfitting in our network.

4.3 The Influence of the Number of Views

When 3D objects are rendered into multiple 2D views, the number of views has different effects on the object recognition performance of the network. We selected eight different view numbers to quantitatively analyze the recognition accuracy of the OVPT method on ModelNet40 (including 1, 3, 6, 9, 12 and 15 views selected using the optimal viewset construction method based on information entropy, and the batch size is uniformly set to 8).

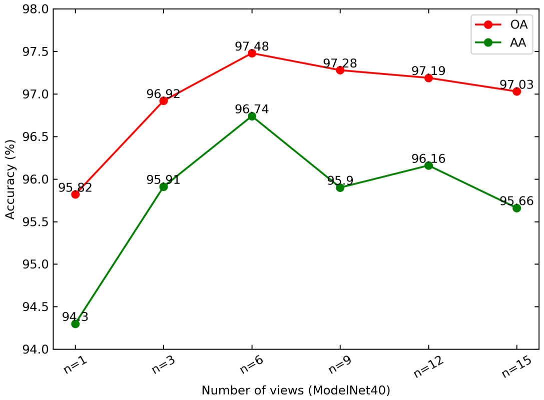
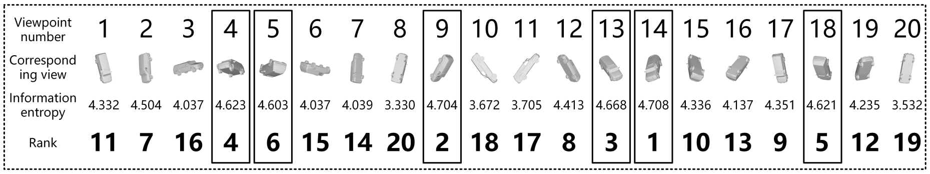


Fig. 4. Recognition performance with different numbers of views.

The object recognition performance of the OVPT method with different number of views is shown in Figure 4. We can find that more views are not necessarily conducive to 3D object recognition. That is, the object recognition performance can't always improve with the increase of the number of views. On the ModelNet40 dataset, when the number of views $n <= 6$, the recognition accuracy is proportional to the number of views. When $6 < n <= 15$, the recognition performance generally appears a downward trend with the increase of the number of views. This experimental result verifies our proposal: the 20 views of the 3D object obtained by the dodecahedron viewpoint method have redundancy, that is, the features of these views have repeated parts. Redundant view features will lead to the degradation of the object recognition performance of the network.



Viewpoint number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Corresponding view																				
Information entropy	4.332	4.504	4.037	4.623	4.603	4.037	4.039	3.330	4.704	3.672	3.705	4.413	4.668	4.708	4.336	4.137	4.351	4.621	4.235	3.532
Rank	11	7	16	4	6	15	14	20	2	18	17	8	3	1	10	13	9	5	12	19

Fig. 5. Optimal viewset construction method based on information entropy.

The proposed optimal viewset construction method solves this problem (see Fig. 5). On the ModelNet40 dataset, selecting the top six views according to the information entropy value as the best viewset, the OVPT method achieves the best OA of 97.48% and AA of 96.74%. Compared with 97.03% OA and 95.66% AA obtained by 15 views, this result is improved by 0.45% and 1.08%.

4.4 The Influence of Viewset Information Entropy Ranking

We further evaluate the effectiveness about the proposed method of constructing the optimal viewset based on information entropy. While ensuring that the number of views n (n=1, 3, 6, 9) is consistent, We construct viewsets with random, uniform viewpoint selection and top-n and bottom-n viewsets with information entropy ranking, respectively. Table 1 is our experimental result.

Table 1. Comparison of different viewpoint selection methods (ModelNet40).

Number of views (n)	1 view	3 views	6 views	9 views
Bottom-n	91.97	95.62	97.16	96.67
Random-n	93.31	96.83	97.12	96.63
Uniform-n	94.04	96.88	97.04	97.11
Top-n	**95.82**	**96.92**	**97.48**	**97.28**

It can be found that the object recognition performance of top-n viewsets is always better than random, uniform and bottom-n viewsets. Especially when the number of views n is fixed to 1, the OVPT method performance for bottom-n viewset and the top-n viewset has a large gap. For example, the OVPT method achieves 91.97% OA with the bottom-n viewset ($n = 1$), which is much lower than the top-n viewset ($n = 1$). Their difference is 3.85%. This is because when the number of views n is set to 1, the bottom n viewsets and the top n viewsets select the view with the 20th and 1st information entropy ranking, respectively. Obviously their information entropy difference is larger, which means that the top-n viewset contains more visual features to improve the recognition performance.

4.5 The Influence of the Pooling Transformer block

We tried 3 different model settings on ModelNet40 to evaluate the impact of pooling transformers on recognition performance. As shown in table 2, we do not need a larger pooling transformer model due to the optimal viewset we build. We achieve the best performance under the tiny model, which means less computational cost. The size of the tiny model is only 29.7MB, which is more lightweight than other model settings.

Table 2. The Influence of the Pooling Transformer block.

Model	Hidden Size	Heads	OA (%)	AA (%)	Model Size
tiny	192	3	**97.48**	**96.74**	**29.7MB**
small	384	6	97.20	96.16	35.6MB
base	768	12	97.28	96.13	57.4MB

Table 3. Ablation study.

Optimal Viewset	Transformer	Pooling Transformer	OA (%)	AA (%)
	✓		98.45	98.26
		✓	98.78	98.66
✓		✓	**99.33**	**99.21**

4.6 Ablation Study

We performed an ablation study on the OVPT network on ModelNet10. The results experiment are shown in Table 3. It can be found that the best recognition performance (99.33% for OA and 99.21% for AA) is achieved with our optimal viewset and pooling transformer method. The main reason is that the optimal viewset solves the redundancy problem of current viewpoint rendering methods. At the same time, the pooling transformer method solves the problem of insufficient local feature aggregation ability of the transformer, which can obtain the feature information of all local view token sequences from local and global aggregation respectively.

4.7 Model Complexity

The results of our experiments comparing model complexity with other methods are shown in Table 4. It can be found that our OVPT method outperforms these enumerated methods in both time and space complexity. In terms of space complexity, the model size of our OVPT method is 29.7MB, and the model sizes of MVCNN and CARNet are 44.8MB and 45.7MB, respectively. It can be seen that the model size of OVPT is reduced by 35% compared to the previous SOTA method CARNet. Even though CARNet uses ResNet18 as the multi-view feature extractor, we use DenseNet121 which is more complex than ResNet18. This is because in the subsequent part of OVPT, which requires only one Tiny Pooling Transformer block (see Fig. 2) to aggregate multi-view features, CARNet also contains more smaller hand-designed components. In terms of time complexity, the running time of our OVPT method is 0.10 seconds, and the running times of MVCNN and CARNet are 0.12 seconds and 0.20 seconds, respectively. Obviously, the running time of our OVPT method is the shortest among these methods, and only needs half of the running time of the previous SOTA method CARNet.

Table 4. Model size and running time comparison (ModelNet40).

Model	Model Size	Running Time	Relative Time Cost
MVCNN [21]	44.8 MB	0.12 seconds	0.6 ×
CARNet [44]	45.7 MB	0.20 seconds	1 ×
OVPT (Ours)	**29.7 MB**	**0.10 seconds**	**0.5 ×**

4.8 Visual Analysis of Confusion Matrix

We use confusion matrix to visually analyze the recognition performance of the OVPT method on the ModelNet40 dataset. It can help us understand which categories are easier or harder to identify. The values on the diagonal of the confusion matrix represent the number of correct identifications, and the values outside the diagonal indicate the number of incorrect identifications. When the value outside the diagonal of the category is lower, it means that the OVPT method is more accurate in identifying the category. As shown in Figure 6, OVPT achieves 100% recognition accuracy on categories such as airplane, bed, and car. Of course, there are also some mis-judgments like cup, night_stand and vase. For the vase object containing 100 samples, 4 samples were misjudged as lamps. We attribute this to the fact that some objects have similar visual features, leading to in model recognition errors. Notably, OVPT has excellent recognition performance in most categories with complex visual features.

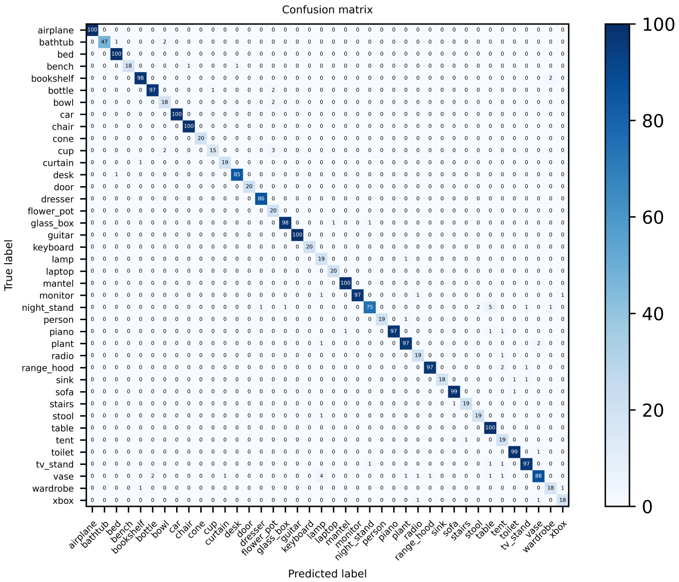


Fig. 6. Confusion Matrix.

4.9 Comparsions with State-of-the-Art Methods

As seen in the Table 5, the OVPT method achieves the state-of-the-art recognition accuracy on the ModelNet10 dataset with OA of 99.33% and AA of 99.21%. Compared with the CARNet [44] method, OVPT improves OA by 0.32%. However, we only need nine views of each 3D object to accomplish the object recognition task. Compared with other multi-view-based methods, the view number is also minimal, which helps reduce computational cost and running time. We can also find that our OVPT method outperforms other point cloud-based and voxel-based methods.

On ModelNet40, the Transformer architecture of MVT [45] follows DeiT's setting [46] to solve the problem of insufficient local feature aggregation capability of the transformer on smaller-scale datasets, and requires stacking multiple local-global Transformer blocks (12 blocks in total) for 3D objects identify. When 20 views of each 3D object are input, the OA reaches 97.50%. Our OVPT method also can achieve close to its recognition accuracy (OA reaches 97.48%). However, one our proposed pooling transformer block and 6 views of each 3D object are only employed, which are much less than MVT. CARNet [44] combines the dodecahedron view rendering method and the knn search method to exploit the latent correspondence of views and viewpoints to aggregate shape features in communication. In contrast, our OVPT method is simpler and easier to use.

Table 5. Recognition performance comparison with present state-of-the-art methods.

Methods	Input Modality	ModelNet40		ModelNet10	
		OA (%)	AA (%)	OA (%)	AA (%)
3D ShapeNets [6]	Voxel	-	77.32	-	83.54
VoxNet [5]	Voxel	-	83.00	-	92.00
O-CNN [8]	Voxel	90.60	-	-	-
PointNet [13]	Point Cloud	89.20	86.20	-	-
PointNet++ [14]	Point Cloud	91.90	-	-	-
DGCNN [19]	Point Cloud	93.50	90.70	-	-
MVCNN [21]	12 Views	92.10	89.90	-	-
HMVCM [30]	12 Views	94.57	-	95.70	-
MHBN [47]	6 Views	94.70	93.10	95.00	95.00
RotationNet [24]	20 Views	97.37	95.84	98.46	95.99
MVT [45]	20 Views	97.50	-	99.30	-
CARNet [44]	20 Views	**97.73**	-	99.01	-
OVPT (Ours)	**1 View**	95.82	94.30	98.45	97.98
OVPT (Ours)	6 Views	97.48	**96.74**	98.89	98.88
OVPT (Ours)	9 Views	97.28	95.90	**99.33**	**99.21**

5 Conclusion

This paper proposes an OVPT network for efficient, accurate 3D object recognition tasks. Compared with other deep learning methods, OVPT introduces information entropy to solve the problem of redundancy when rendering 3D objects into multiple views. The pooling transformer can efficiently capture the relevant feature information between multiple views and realize the global and local aggregation of multi-view low-level local feature token sequences into compact 3D global descriptors. We conducted a series of experiments on two popular ModelNet datasets, and the results show that OVPT achieves state-of-the-art performance while using the least number of views. Our method significantly improves the accuracy and efficiency of 3D object recognition tasks and reduces the computational cost, and is especially suitable for computationally resource-constrained environments. This method can be widely used in areas such as autonomous driving, augmented reality, interior design, and robotics.

Acknowledgments. The fnancial support for this work was sponsored by Natural Science Foundation of Shanghai under Grant No. 19ZR1435900.

References

1. Liang, Z., Guo, Y., Feng, Y., Chen, W., Qiao, L., Zhou, L., Zhang, J., Liu, H.: Stereo matching using multi-level cost volume and multi-scale feature constancy. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(1), 300–315 (2021). https://doi.org/10.1109/TPAMI.2019.2928550
2. Li, Y., Ma, L., Zhong, Z., Liu, F., Chapman, M.A., Cao, D., Li, J.: Deep learning for lidar point clouds in autonomous driving: A review. IEEE Transactions on Neural Networks and Learning Systems **32**(8), 3412–3432 (2021). https://doi.org/10.1109/TNNLS.2020.3015992
3. Kästner, L., Frasineanu, V.C., Lambrecht, J.: A 3d-deep-learning-based augmented reality calibration method for robotic environments using depth sensor data. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 1135–1141 (2020). https://doi.org/10.1109/ICRA40945.2020.9197155
4. Lee, D., Ryu, S., Yeon, S., Lee, Y., Kim, D., Han, C., Cabon, Y., Weinzaepfel, P., Guérin, N., Csurka, G., Humenberger, M.: Large-scale localization datasets in crowded indoor spaces. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3226–3235 (2021). https://doi.org/10.1109/CVPR46437.2021.00324
5. Maturana, D., Scherer, S.: Voxnet: A 3d convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 922–928 (2015). https://doi.org/10.1109/IROS.2015.7353481
6. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1912–1920 (2015). https://doi.org/10.1109/CVPR.2015.7298801
7. Riegler, G., Ulusoy, A.O., Geiger, A.: Octnet: Learning deep 3d representations at high resolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6620–6629 (2017). https://doi.org/10.1109/CVPR.2017.701
8. Wang, P.S., Liu, Y., Guo, Y.X., Sun, C.Y., Tong, X.: O-cnn: Octree-based convolutional neural networks for 3d shape analysis. ACM Trans. Graph. **36**(4) (jul 2017). https://doi.org/10.1145/3072959.3073608, https://doi.org/10.1145/3072959.3073608
9. Klokov, R., Lempitsky, V.: Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 863–872 (2017). https://doi.org/10.1109/ICCV.2017.99
10. Zeng, W., Gevers, T.: 3dcontextnet: K-d tree guided hierarchical learning of point clouds using local and global contextual cues. In: Leal-Taixé, L., Roth, S. (eds.) Computer Vision – ECCV 2018 Workshops. pp. 314–330. Springer International Publishing, Cham (2019)
11. Le, T., Duan, Y.: Pointgrid: A deep network for 3d shape understanding. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9204–9214 (2018). https://doi.org/10.1109/CVPR.2018.00959
12. Meng, H.Y., Gao, L., Lai, Y.K., Manocha, D.: Vv-net: Voxel vae net with group convolutions for point cloud segmentation. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 8499–8507 (2019). https://doi.org/10.1109/ICCV.2019.00859

13. Charles, R.Q., Su, H., Kaichun, M., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 77–85 (2017). https://doi.org/10.1109/CVPR.2017.16
14. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf
15. Jiang, M., Wu, Y., Zhao, T., Zhao, Z., Lu, C.: Pointsift: A sift-like network module for 3d point cloud semantic segmentation (2018). https://doi.org/10.48550/ARXIV.1807.00652, https://arxiv.org/abs/1807.00652
16. Zhao, H., Jiang, L., Fu, C.W., Jia, J.: Pointweb: Enhancing local neighborhood features for point cloud processing. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5560–5568 (2019). https://doi.org/10.1109/CVPR.2019.00571
17. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3141–3149 (2019). https://doi.org/10.1109/CVPR.2019.00326
18. Feng, M., Zhang, L., Lin, X., Gilani, S.Z., Mian, A.: Point attention network for semantic segmentation of 3d point clouds. Pattern Recognition **107**, 107446 (2020). https://doi.org/https://doi.org/10.1016/j.patcog.2020.107446, https://www.sciencedirect.com/science/article/pii/S0031320320302491
19. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Trans. Graph. **38**(5) (oct 2019). https://doi.org/10.1145/3326362, https://doi.org/10.1145/3326362
20. Zhang, K., Hao, M., Wang, J., de Silva, C.W., Fu, C.: Linked dynamic graph cnn: Learning on point cloud via linking hierarchical features (2019). https://doi.org/10.48550/ARXIV.1904.10014, https://arxiv.org/abs/1904.10014
21. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 945–953 (2015). https://doi.org/10.1109/ICCV.2015.114
22. Wang, C., Pelillo, M., Siddiqi, K.: Dominant set clustering and pooling for multi-view 3d object recognition (2019). https://doi.org/10.48550/ARXIV.1906.01592, https://arxiv.org/abs/1906.01592
23. Feng, Y., Zhang, Z., Zhao, X., Ji, R., Gao, Y.: Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 264–272 (2018). https://doi.org/10.1109/CVPR.2018.00035
24. Kanezaki, A., Matsushita, Y., Nishida, Y.: Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5010–5019 (2018). https://doi.org/10.1109/CVPR.2018.00526
25. Esteves, C., Xu, Y., Allec-Blanchette, C., Daniilidis, K.: Equivariant multi-view networks. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1568–1577 (2019). https://doi.org/10.1109/ICCV.2019.00165

26. Han, Z., Lu, H., Liu, Z., Vong, C.M., Liu, Y.S., Zwicker, M., Han, J., Chen, C.L.P.: 3d2seqviews: Aggregating sequential views for 3d global feature learning by cnn with hierarchical attention aggregation. IEEE Transactions on Image Processing **28**(8), 3986–3999 (2019). https://doi.org/10.1109/TIP.2019.2904460
27. He, X., Huang, T., Bai, S., Bai, X.: View n-gram network for 3d object retrieval. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7514–7523 (2019). https://doi.org/10.1109/ICCV.2019.00761
28. Yang, Z., Wang, L.: Learning relationships for multi-view 3d object recognition. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7504–7513 (2019). https://doi.org/10.1109/ICCV.2019.00760
29. Chen, S., Zheng, L., Zhang, Y., Sun, Z., Xu, K.: Veram: View-enhanced recurrent attention model for 3d shape classification. IEEE Transactions on Visualization and Computer Graphics **25**(12), 3244–3257 (2019). https://doi.org/10.1109/TVCG.2018.2866793
30. Liu, A.A., Zhou, H., Nie, W., Liu, Z., Liu, W., Xie, H., Mao, Z., Li, X., Song, D.: Hierarchical multi-view context modelling for 3d object classification and retrieval. Information Sciences **547**, 984–995 (2021). https://doi.org/https://doi.org/10.1016/j.ins.2020.09.057, https://www.sciencedirect.com/science/article/pii/S0020025520309671
31. Lin, C., Li, C., Liu, Y., Chen, N., Choi, Y.K., Wang, W.: Point2skeleton: Learning skeletal representations from point clouds. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4275–4284 (2021). https://doi.org/10.1109/CVPR46437.2021.00426
32. Liu, M., Zhang, X., Su, H.: Meshing point clouds with predicted intrinsic-extrinsic ratio guidance. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 68–84. Springer International Publishing, Cham (2020)
33. Rakotosaona, M.J., Guerrero, P., Aigerman, N., Mitra, N., Ovsjanikov, M.: Learning delaunay surface elements for mesh reconstruction. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 22–31 (2021). https://doi.org/10.1109/CVPR46437.2021.00009
34. Shannon, C.E.: A mathematical theory of communication. The Bell System Technical Journal **27**(3), 379–423 (1948). https://doi.org/10.1002/j.1538-7305.1948.tb01338.x
35. Vázquez, P.P., Feixas, M., Sbert, M., Heidrich, W.: Viewpoint selection using viewpoint entropy. In: VMV. vol. 1, pp. 273–280. Citeseer (2001)
36. Vázquez, P.P., Feixas, M., Sbert, M., Heidrich, W.: Automatic view selection using viewpoint entropy and its application to image-based modelling. Computer Graphics Forum **22**(4), 689–700. https://doi.org/https://doi.org/10.1111/j.1467-8659.2003.00717.x, https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2003.00717.x
37. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
38. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale (2020). https://doi.org/10.48550/ARXIV.2010.11929, https://arxiv.org/abs/2010.11929
39. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization (2016). https://doi.org/10.48550/ARXIV.1607.06450, https://arxiv.org/abs/1607.06450

40. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
41. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus) (2016). https://doi.org/10.48550/ARXIV.1606.08415, https://arxiv.org/abs/1606.08415
42. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2261–2269 (2017). https://doi.org/10.1109/CVPR.2017.243
43. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2014). https://doi.org/10.48550/ARXIV.1412.6980, https://arxiv.org/abs/1412.6980
44. Xu, Y., Zheng, C., Xu, R., Quan, Y., Ling, H.: Multi-view 3d shape recognition via correspondence-aware deep learning. IEEE Transactions on Image Processing **30**, 5299–5312 (2021). https://doi.org/10.1109/TIP.2021.3082310
45. Chen, S., Yu, T., Li, P.: Mvt: Multi-view vision transformer for 3d object recognition (2021). https://doi.org/10.48550/ARXIV.2110.13083, https://arxiv.org/abs/2110.13083
46. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jegou, H.: Training data-efficient image transformers amp; distillation through attention. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 10347–10357. PMLR (18–24 Jul 2021), https://proceedings.mlr.press/v139/touvron21a.html
47. Yu, T., Meng, J., Yuan, J.: Multi-view harmonized bilinear network for 3d object recognition. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 186–194 (2018). https://doi.org/10.1109/CVPR.2018.00027