Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.067760

ARTICLE

Three-Dimensional Model Classification Based on VIT-GE and Voting Mechanism

Fang Yuan, Xueyao Gao* and Chunxiang Zhang

School of Computer Science and Technology, Harbin University of Science and Technology, Harbin, 150080, China

*Corresponding Author: Xueyao Gao. Email: xueyao_gao@163.com

Received: 12 May 2025; Accepted: 08 August 2025; Published: 23 October 2025

ABSTRACT: 3D model classification has emerged as a significant research focus in computer vision. However, traditional convolutional neural networks (CNNs) often struggle to capture global dependencies across both height and width dimensions simultaneously, leading to limited feature representation capabilities when handling complex visual tasks. To address this challenge, we propose a novel 3D model classification network named ViT-GE (Vision Transformer with Global and Efficient Attention), which integrates Global Grouped Coordinate Attention (GGCA) and Efficient Channel Attention (ECA) mechanisms. Specifically, the Vision Transformer (ViT) is employed to extract comprehensive global features from multi-view inputs using its self-attention mechanism, effectively capturing 3D shape characteristics. To further enhance spatial feature modeling, the GGCA module introduces a grouping strategy and global context interactions. Concurrently, the ECA module strengthens inter-channel information flow, enabling the network to adaptively emphasize key features and improve feature fusion. Finally, a voting mechanism is adopted to enhance classification accuracy, robustness, and stability. Experimental results on the ModelNet10 dataset demonstrate that our method achieves a classification accuracy of 93.50%, validating its effectiveness and superior performance.

KEYWORDS: 3D model; voting algorithm; visual transformer design space

1 Introduction

Recently, with the rapid development of computer vision and sensors, the entry threshold of 3D models has been greatly reduced, making their application in computer fields, autonomous driving, medical imaging, virtual reality, industrial production design, and other fields more and more extensive. In autonomous driving, accurate recognition of surrounding 3D objects is essential for ensuring safe navigation and collision avoidance. In medical imaging, precise classification of 3D anatomical structures assists in disease diagnosis and treatment planning. Industrial inspection relies on 3D model recognition for defect detection and quality control of complex components. Similarly, virtual reality and robotic manipulation systems depend on accurate understanding of object geometries for realistic interaction and reliable operation. As 3D sensing technologies continue to advance, massive volumes of 3D data are being generated across these fields. Developing robust and accurate 3D classification algorithms has therefore become increasingly critical to fully unlock the potential of these intelligent systems.

Biasotti et al. [1] established a dataset of 572 textured 3D models in the SHREC'14 competition. By comparing 22 algorithms from 8 participating teams, they demonstrated that processing textures in CIELab color space outperforms RGB space, advancing the development of multimodal 3D retrieval technology. The data structure of 3D models is richer than that of 2D models and can capture geometric information

such as the depth, contour, and shape of objects, thereby providing the possibility for more accurate analysis and recognition. As the core task of managing and identifying 3D data, 3D model classification can directly affect the application effect of the model in actual scenarios. Traditional 3D model classification methods rely on manually designed features, extracting geometric information through feature descriptors, and then combining them with machine learning methods such as support vector machines (SVM) for classification. However, manual feature design has limitations and is not easy to capture the global information and deep structure of complex models. The rapid development of deep learning can solve this problem, especially convolutional neural networks (CNN) and graph neural networks (GNN), which make it possible to automatically extract features. Li et al. [2] proposed the hierarchical multiview top-k pooling (HMTPool) method with deep-Q-networks, which adaptively selects the optimal pooling ratio and scores nodes based on multiview information, addressing the limitations of existing graph pooling methods. Li et al. [3] presented a systematic survey on graph pooling techniques, comprehensively reviewing the theoretical foundations and research progress of node/subgraph merging methods in graph neural networks. They categorized existing approaches, evaluated performance through benchmark testing, and analyzed current challenges along with future directions. With the introduction of neural networks such as YOLO [4], Efficient Net [5], and Google Net [6], classification tasks have achieved remarkable results.

This indicates the urgent need to develop more advanced architectures capable of capturing global spatial dependencies while preserving fine-grained feature details across diverse views. Based on the diversity of 3D model data representation, 3D model classification algorithms based on deep learning can be divided into four categories: point cloud-based classification methods, voxel-based classification methods, mesh-based classification methods, and view-based classification methods.

Among the point cloud-based methods, Ilamchezhian and Raj [7] proposed an efficient parallel method for segmenting 3D point cloud images using OpenMP. Lagahit et al. [8] proposed a learnable scaling and Laplacian filtering U-Net that can extract and classify sparse point cloud images. Charles et al. [9] proposed Point Net to perform input transformation and feature transformation on unordered point cloud data and then obtain the global features of the point cloud through the Maxol symmetric function to achieve model classification. Subsequently, Song et al. [10] proposed the Kernel Correlation Learning Block (KCB), which can adaptively learn local geometric features and global features of different layers to enhance the perception ability of the network. Hassan et al. [11] introduced a deep hierarchical 3D point cloud classification architecture that consists of multi-layer sampling, concentric ring convolution, a pooling layer, and a residual feature propagation block. They can learn robust geometric features. Although the above methods can directly process point cloud data, the classification effect is still affected by the disorder and limited information of point clouds.

In mesh-based approaches, Biasotti et al. [12] compared six textured 3D model retrieval algorithms and demonstrated that processing textures in CIELab color space yields better results than RGB space. However, existing methods still struggle with complex geometry-texture combinations. This study established the first systematic evaluation benchmark for 3D object recognition.

In voxel-based methods, the 3D model is changed into a standard voxel format, and then a 3D convolutional neural network (3D CNN) is used to identify important details and sort the model. Wu et al. [13] showed the shape of the 3D model as a probability distribution of binary variables on a 3D voxel grid, and then used 3D Shape Nets to learn the model's category features. Xu et al. [14] adopted a framework similar to 3D Shape Nets and formulated the CNN learning process as a directed search. Subsequently, Kim et al. [15] designed a Part Geometry Network (PG-Net), which can be used as a robust feature descriptor for 3D model classification and reconstruction. Ma et al. [16] introduced a network that focuses on two types of input data, using both dual-channel input voxels and 3D Radon feature matrices. Cai et al. [17] suggested

a network that uses both voxel data and three different views, merging them together for better results. He et al. [18] created a two-part system for extracting voxel features by using graph attention networks and 3D sparse convolutions. Vowelized data can directly use 3D convolutional neural networks to extract multi-level and multi-scale local information. Voxel-based methods are limited by voxel resolution. At the same time, 3D convolution consumes more resources in feature extraction than 2D convolution.

Because deep learning is more developed for 2D image processing than for point clouds and voxels, view-based methods create a 3D model's view by projecting it into 2D and use 2D neural networks to gather important features, instead of applying 3D convolution directly to the model. Denisenko et al. [19] proposed using integral spin images as representations of 3D models. Several computational experiments were conducted on constructing spin images of Princeton Shape Benchmark 3D models, and they were used to train deep neural networks. Hegde and Zadeh [20] designed a voxel CNN architecture that represents 3D models by combining views and voxels and extracted model features through CNN. Jin et al. [21] proposed a representative view positioning framework for 3D models using rotation direction prediction and rotation angle prediction. The local information fusion network proposed by Zhu et al. [22] uses the relationship between image regions and feature map channels to perform feature map reorganization and regional feature extraction. Liu et al. [23] created a semantic and contextual fusion network (SCFN) that produces images from various angles of a 3D model and uses CNN to pull out the original features from each image. They then designed a channel attention mechanism (CAM) to enhance view-level semantic information. They then constructed a contextual fusion module (CFM) to achieve the effective fusion of multi-view features.

At present, view-based classification methods are widely studied, and they can achieve excellent classification results when combined with neural networks. However, traditional neural networks still have some problems. First, in an image, many useful features may be distributed between distant pixel regions. Second, traditional neural networks gradually accumulate global information through local convolution; it is not easy to capture the overall layout and global semantic information of the image at a shallow level. Finally, traditional neural networks are not very efficient when working with large image datasets, and because the convolution operations happen one after another, they can't easily take advantage of parallel processing. To address the above problems, this paper proposes a 3D model classification method based on the ViT-GE feature extraction network and voting mechanism. First, the 3D model data is changed into a 2D view by rotating it 60° up and down and 60° side to side, which shows the structure of the 3D model from different angles. This view setting can ensure that the key features of the model are captured in different orientations, providing a more comprehensive input for subsequent feature extraction. Secondly, based on the ViT-GE feature extraction network, deep feature extraction of the view is performed. GGCA enhances how the model understands spatial information by using global grouping coordinate interaction, and the ECA mechanism helps improve the flow of channel information, allowing the network to smartly pick important features, which boosts its ability to express those features. After that, the modified patch features go into the Transformer encoder to gather overall features, and the multilayer perceptron (MLP) is used to further refine the features after each self-attention layer to improve how well they are represented. At the same time, ViT introduces a special classification token in the input to aggregate all patch information to form the final global image feature representation. Finally, the features taken by the Transformer encoder are sent to the classification layer, where the results from several separate classifiers are combined using the Softmax function and a voting system to enhance accuracy, stability, and reliability in classification.

This study targets the core challenge of achieving robust and accurate multi-view 3D object classification, which plays a crucial role in real-world applications such as autonomous driving, medical imaging, and industrial inspection. These tasks often require reliable understanding of geometric structures from

diverse viewpoints, even under occlusions, noise, or limited data availability. To this end, the proposed ViT-GE framework incorporates global self-attention with lightweight attention modules to effectively capture both local and long-range spatial dependencies. Compared with existing CNN-based and transformer-based approaches, ViT-GE introduces a modular attention design and a probabilistic multi-view fusion strategy, enabling a favorable trade-off between performance and computational cost. These contributions underscore both the practical applicability and theoretical advancement of our method, making it a compelling solution for complex 3D perception tasks.

2 Methods

The 3D model classification framework of this paper is shown in Fig. 1. The 3D model classification framework mainly consists of three key parts: multi-view projection, the ViT-GE feature extraction network, and the voting mechanism.

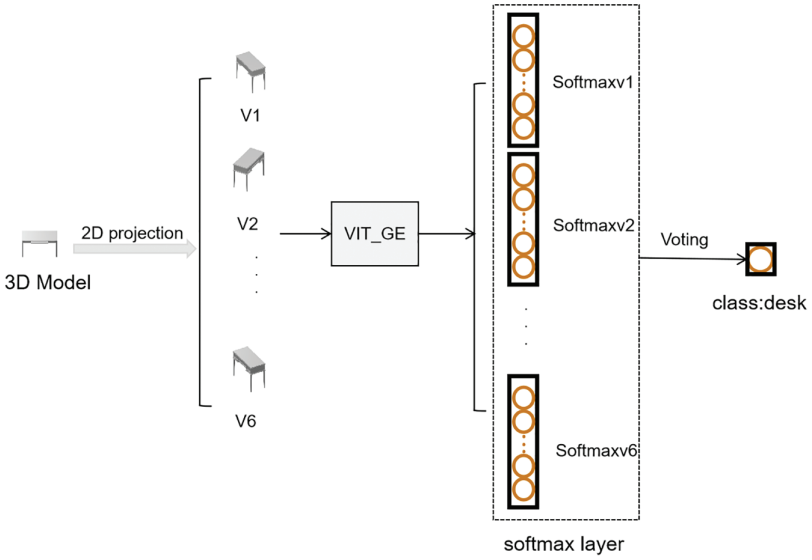

Figure 1: 3D model classification framework

First, the 3D model is converted into a 2D image by multi-view projection. Specifically, the 3D model is rendered from different angles (such as front view, top view, side view, etc.) to generate grayscale images of multiple views. This process can effectively map the 3D information to the 2D space while retaining the geometric features of the object, ensuring that the model can learn the key information from different angles.

Then, the generated multi-view image is inserted into the ViT-GE feature extraction network. The network builds on the traditional Vision Transformer (ViT) by adding global grouped coordinate attention (GGCA) and efficient channel attention (ECA) to improve how it understands features. GGCA helps recognize the overall layout of the image, especially linking different views, while ECA improves how channel information is organized, allowing the model to better focus on important features. GGCA can capture the global spatial relationship in the image, especially to establish connections between different views, while ECA is used to optimize the distribution of channel information so that the model can focus on key features more effectively. In the ViT-GE network, each view image is processed through several Transformer encoder blocks, which include multi-head self-attention (MHA) and feed-forward networks, to pull out important details. Ultimately, the features of each view are mapped to a high-dimensional feature space for subsequent classification tasks. In the classification decision stage, a voting mechanism is used for

final classification. Since multi-view images extract features separately and predict independently, a fusion strategy is required to determine the final classification result. The framework integrates the classification results of all views through a voting mechanism, using hard voting or soft voting methods to improve the robustness and accuracy of classification. Compared with the single-view classification method, the voting mechanism makes full use of the information of different viewpoints, reduces the classification errors caused by occlusion or information loss of a single viewpoint, and thus improves the overall performance. The overall classification framework, including GGCA, ECA, and the voting mechanism, is summarized in Algorithm 1 for clarity.

Algorithm 1: 3D model classification algorithm based on ViT-GE and voting mechanism

Inputs: Multi-view image set $X = \{x_1, x_2, \ldots, x_n\}$ of a 3D model
Outputs: Final predicted class label V^*
1: Each view x_i is input to the GGCA module to enhance spatial geometric representation
2: The output of GGCA is input to the ViT encoder for feature extraction
3: The resulting feature f_i is passed through the ECA module to recalibrate channel-wise attention
4: The refined feature is forwarded through a classification head to produce a class probability vector p_i
5: Repeat steps 1–4 for all N views
6: If soft voting is used:
7: Average all probability vectors: $S_k = \frac{1}{N}\sum_{i=1}^{N} p_i$
8: Determine final class label: $V^* = \arg\max(s)$
9: Else if hard voting is used:
10: For each p_i, determine predicted class $C_i = \arg\max(p_i)$
11: Determine final label V^* via majority vote from $\{C_1, \ldots, C_n\}$
12: Return predicted class label V^*

2.1 Projection Method

The main dataset of this experiment uses the ModelNet10 dataset, which is a classic benchmark dataset for 3D model classification. To classify it more efficiently using image processing methods, this paper image-processes the 3D model. Specifically, each 3D model in the dataset is first rotated at a 45° vertical angle and then photographed in turn at intervals of 60° on the horizontal plane to generate images. Show Fig. 2.

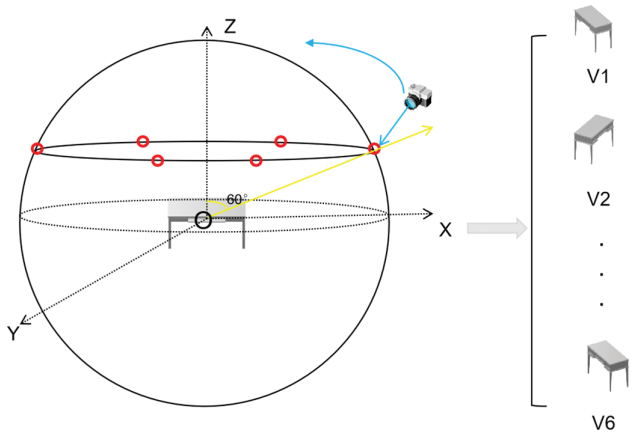

Figure 2: Representation by projection methods

In this way, each 3D model can generate 2D images from multiple perspectives, thus forming a view-based dataset. This multi-perspective image processing method can not only retain the important structural information of the 3D model but also make full use of the existing technology for processing 2D images for feature extraction and classification tasks.

2.2 View Feature Extraction

Nowadays, convolutional neural networks are developing rapidly, and using them to classify images has become the preferred method, but they still have some shortcomings and problems. ViT [24] has significant advantages over traditional convolutional neural networks in capturing global features and adapting to complex task requirements, so this paper chooses the ViT neural network. The structure of ViT is shown in Fig. 3. ViT extracts image features through the self-attention mechanism, which is different from the CNN. The whole process starts with the input image. First, the image is divided into several patches of fixed size, and then each patch is linearly projected, flattened, and converted into a series of feature vectors. To retain the position information, ViT also adds positional encoding to help Transformer identify the spatial relationship between image patches. The processed patch features are fed into the Transformer Encoder as tokens, which is made up of several multi-head self-attention and feedforward neural networks (MLP). In the Transformer structure, each patch token interacts with all other tokens through the self-attention mechanism to capture global features. Each self-attention layer is followed by a normalization operation, and MLP is used to further enhance the feature expression capability.

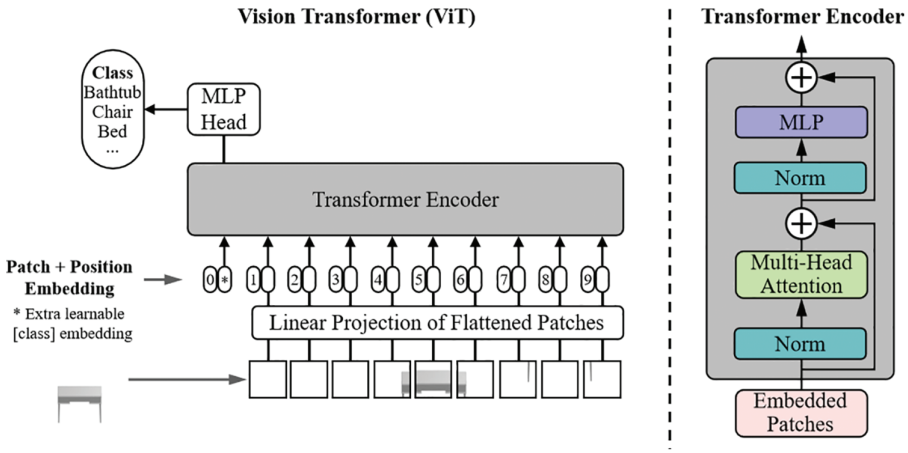

Figure 3: Vision Transformer Architecture (adapted from Dosovitskiy et al., "An image is worth 16 × 16 words: Transformers for image recognition at scale" [24])

In addition, ViT introduces a special class token into the input token sequence. This token aggregates the information of all patches after multiple layers of processing by the Transformer encoder and is ultimately used for classification tasks. The classification head (MLP head) uses SoftMax to make the final classification prediction based on the output of this classification token, for example, identifying that the input image belongs to the category of "chair", "bed", or "bathtub".

Vision Transformer (ViT) is a model based on the self-attention mechanism that can effectively extract global features. However, due to the lack of modeling capabilities for local information, some fine-grained key information may be ignored when expressing features. The CNN have advantages in processing local features. Therefore, to make up for the shortcomings of ViT in local modeling, the channel attention mechanism can be combined to enhance the feature expression capability. Among them, Efficient Channel Attention

(ECA) can boost how well features are represented without making things more complicated by learning how different channels relate to each other. When ECA is used with ViT, it helps improve how channel information works together, allowing ViT to better capture overall features and also improve the details of local features, which leads to better classification results for the model. ECA (Efficient Channel Attention) is a lightweight attention module based on the channel domain. It uses 1D convolution without dimensionality reduction to achieve better cross-channel information interaction. The ECA structure is shown in Fig. 4.

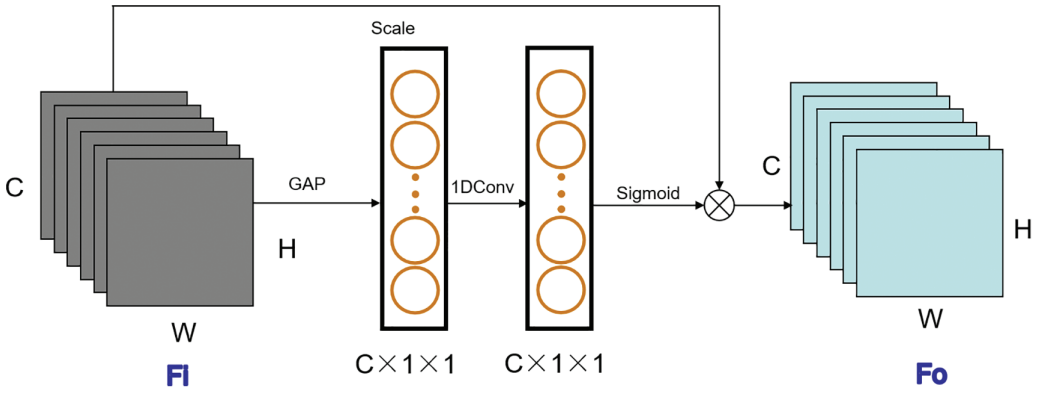

Figure 4: Efficient channel attention architecture

As shown in Fig. 4, for the input feature matrix Fi that has the size (C, H, W), we first use GAP to get the features that are still the same size in two dimensions (C, 1, 1); next, we apply 1D convolution (1DConv) to gather information across the channels; finally, we use the Sigmoid function to adjust the feature weights to a range between 0 and 1, and then combine these with Fi to create the output feature matrix Fo. The convolution kernel K represents the local cross-channel interaction coverage, which is obtained by Formula (1), where |t| odd represents the odd number closest to t.

$$K = \left| \frac{\log_2\left(\frac{w_i}{b}\right)}{2} + \frac{1}{2} \right|_{odd} \tag{1}$$

Although ViT can capture global features, the ECA mechanism makes up for its shortcomings in modeling local features to a certain extent; ECA mainly focuses on information interaction in the channel dimension and lacks in-depth modeling of spatial relationships. Therefore, to further enhance the model's ability to understand three-dimensional shape information, the Global Grouped Coordinate Attention (GGCA) mechanism is introduced. GGCA is a new attention method that seeks to greatly boost how well features are extracted and combined by using grouping and global context interaction. It is particularly suitable for application in computer vision tasks such as image classification, object detection, and image segmentation. Instead of using traditional attention methods, GGCA focuses on organizing input features by their channel dimension (group-wise), which makes it less complicated to compute while keeping the features in the group independent and expressive. For the input feature tensor $X \in R^{H\times W\times C}$, its channel dimension is first divided into G groups; each group contains $\frac{C}{G}$ channels, forming a tensor of shape $R^{H\times W\times \frac{C}{G}\times G}$. Through this grouping operation, the computational cost of the model is significantly reduced while providing a relatively independent subspace for each group of feature learning, thereby enhancing the flexibility of feature expression.

After grouping, GGCA uses average pooling (Avg Pooling) and maximum pooling (Max Pooling) to extract features in the height direction (Height) and width direction (Width), respectively. These two pooling

operations capture global feature distribution and local saliency information, respectively. Among them, the pooling operation in the height direction outputs a shape of $R^{\frac{C}{G}\times 1\times G}$, while the pooling operation in the width direction outputs $R^{\frac{C}{G}\times 1\times G}$. This design allows the GGCA module to gather detailed context in space while keeping local details, striking a balance between global information and local features.

The results of the pooling operation are further processed by the cross-dimensional interaction module, which is the core component of GGCA. The cross-dimensional interaction module first reduces the pooling results using a shared 1 × 1 convolutional layer, then adds batch normalization and ReLU activation functions to allow for more complex expressions. The processed features are re-expanded through a second 1 × 1 convolutional layer to form context-enhanced features. In this process, all operations share weights within each group, thereby achieving cross-group feature interaction and further reducing computational complexity. Through this module, the information between pooled features can be effectively fused to generate highly expressive contextual features.

After generating cross-dimensional interactive features, the GGCA module generates two attention weight vectors through weighted fusion, corresponding to the height and width directions, respectively. Subsequently, the input feature tensor is weighted element by element according to these attention weight vectors to complete the fusion of global context and local features and output an enhanced feature tensor consistent with the input dimension. Through this mechanism, the GGCA module can focus on important feature areas and dynamically adjust the attention distribution, thereby improving the performance of the model in complex tasks. The specific structure is shown in Fig. 5.

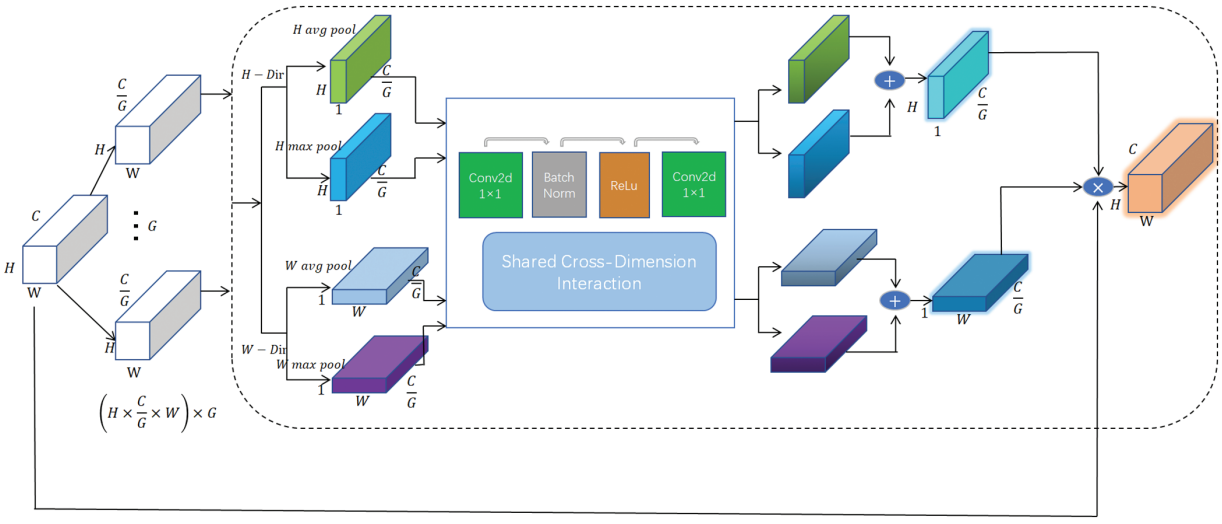

Figure 5: GGCA structure

The GGCA module has significant advantages in many aspects. First, its grouping design and shared interaction mechanism greatly reduce the computational overhead, enabling it to efficiently process high-resolution feature maps. Second, by combining the average pooling and maximum pooling methods, GGCA can capture both global and local contextual information, thereby improving the model's feature expression capabilities. In addition, the module is designed to be highly versatile and scalable and can be seamlessly integrated into existing networks such as ViT, Reset, and NextViT. By adjusting the number of groups G, GGCA can flexibly adapt to different hardware conditions and model requirements, providing optimization for applications of different scales.

GGCA uses a grouping method to understand how features relate to each other in space and across channels, which helps the model better represent 3D structures while still considering the overall picture. When ViT, ECA, and GGCA work together, ViT gathers global information, ECA boosts attention on different channels, and GGCA handles the interaction of spatial information globally, which makes the features more complete and ultimately leads to better accuracy and stability in classifying 3D models.

The VIT-GE feature extraction network combines three methods: Vision Transformer (ViT), global grouped coordinate attention (GGCA), and efficient channel attention (ECA), to enhance the ability to extract features for classifying 3D models. The feature extraction network is shown in Fig. 6. The main process of the entire network includes three core stages: global spatial feature modeling, channel attention enhancement, and global feature extraction and classification. First, the network starts with the input view image and uses the GGCA module to model the global spatial features. The GGCA module uses a method called grouped coordinate attention to analyze the feature information in both the horizontal (H_Attn) and vertical (W_Attn) directions. Specifically, the input features are subjected to two parallel pooling operations (average pooling, AvgPool, and maximum pooling, MaxPool) to extract information of different scales, and then the features are transformed through shared convolution (Shared Conv), batch normalization (BatchNorm), and ReLU activation functions, and weights are generated through Sigmoid activation. Finally, H_Attn and W_Attn are weighted and fused with the original features to enhance the expression of global spatial information. Next, the features processed by GGCA enter the ECA module to further improve the feature expression ability in the channel dimension. The ECA module uses adaptive one-dimensional convolution (Conv1D) to calculate the correlation between channels and generates channel attention weights through Sigmoid activation, thereby enhancing important feature channels and suppressing irrelevant channels. Unlike the traditional SE attention mechanism, this method does not reduce dimensions using a fully connected layer, allowing it to fully utilize channel information while keeping computations efficient, which enhances the ability to express features. The features optimized by GGCA and ECA are then sent to the ViT network for deep feature extraction. First, ViT divides the input image into several fixed-size patches through linear projection and flattens them into feature vectors. Subsequently, these feature vectors enter the core module of ViT, the Transformer encoder, which consists of multiple Transformer Blocks, each of which contains multi-head self-attention, layer normalization, and multi-layer perceptron (MLP). The multi-head self-attention mechanism can effectively model the long-distance dependencies between different patches, allowing the model to focus on the overall structural information of the 3D shape, while the MLP layer enhances the expressiveness of features through nonlinear transformations. In ViT's self-attention mechanism, the features of each patch are first processed using query (Q), key (K), and value (V); then, an attention score is calculated, and the features are adjusted using Softmax normalization to create the final attention representation. After all transformer blocks complete feature extraction, the final global features are transformed by normalization (Norm) and linear layers (Linear), and the feature representation for 3D model classification is output. The ViT-GE network enhances the spatial feature modeling capability through GGCA, ECA optimizes channel attention allocation, and ViT provides powerful global information modeling capabilities, enabling the entire network to achieve higher classification accuracy and stronger feature expression capabilities in 3D model classification tasks.

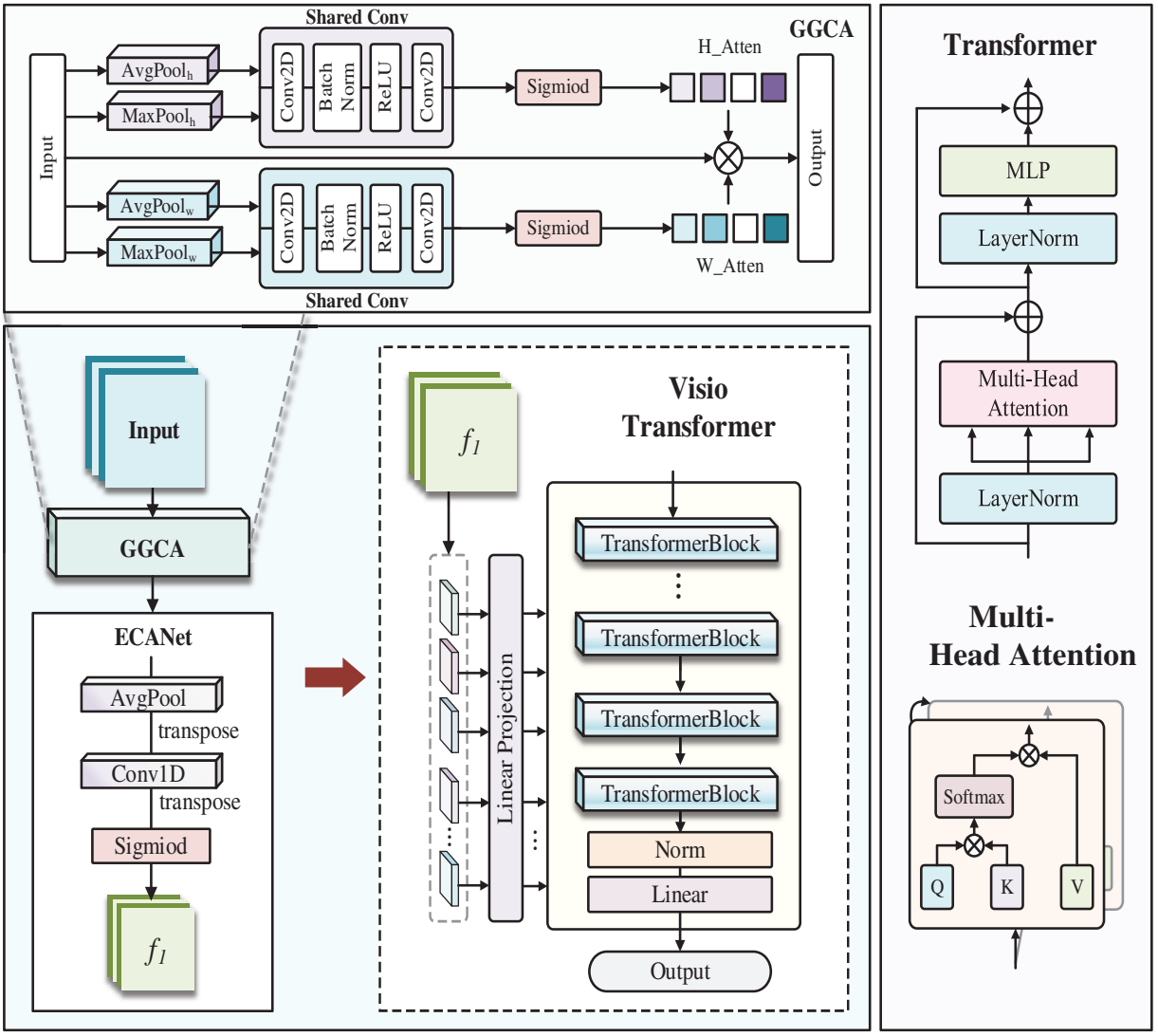

Figure 6: VIT-GE feature extraction network

2.3 Voting Mechanism

In multi-view 3D model classification, each model is projected into multiple 2D views from different angles. Since each view provides partial information, individual predictions from multiple views may vary. To aggregate multi-view information and improve classification robustness, we adopt a voting-based ensemble strategy. Specifically, two fusion strategies are utilized: hard voting and soft voting.

2.3.1 Hard Voting

In hard voting, each view independently predicts a class label. The final prediction is determined by counting the number of votes received for each class and selecting the class with the highest count.

Let:

N denote the number of views,

$Vi \in L$ be the predicted class label for the ith view,

L be the set of all possible class labels.

The final predicted class V* is shown in Eq. (2), where $\delta(\cdot)$ is the indicator function, equal to 1 if the condition holds, otherwise 0. Hard voting considers each view equally, regardless of the model's confidence in individual predictions. The method is simple, but may ignore useful confidence information when some views are more informative than others.

$$V^* = \underset{l \in L}{arg\max} \left(\sum_{i=1}^{N} \delta \left(V_i = l \right) \right) \tag{2}$$

2.3.2 Soft Voting

In soft voting, instead of directly using class labels, we aggregate the predicted probability distributions across views, allowing the confidence of each prediction to contribute to the final decision.

Let:

$P_i = [p_{i,1}, p_{i,2}, \ldots, p_{i,K}]$ be the predicted probability distribution (after softmax) from the ith view,

K be the total number of classes.

We compute the aggregated probability score for each class k is shown in Eq. (3). The final predicted class $V^*$is shown in Eq. (4). Soft voting leverages the full probability distribution output from each view, enabling a more nuanced integration of multi-view information. This often leads to improved performance, especially when some views are ambiguous or less informative.

$$S_k = \frac{1}{N} \sum_{i=1}^{N} p_{i,k} \tag{3}$$

$$V^* = \arg \max_{k \in \{1,2,\ldots,K\}} S_k \tag{4}$$

In the 3D model classification task, the voting mechanism is usually applied to multi-view learning and multi-model fusion. Specifically, 3D objects can often be classified from multiple perspectives, and each perspective provides different information. Through the voting mechanism, the classification results from different perspectives can be fused, thereby improving the accuracy and stability of classification

3 Equations and Mathematical Expressions

The experiment is based on the PyTorch framework, the PC is Ubuntu 18.04.2, and the GPU is RTX4080Ti. The classification experiment uses the rigid dataset ModelNet10. The dataset is divided into a dataset and a test set in a ratio of 4:1. Among them, the training set in ModelNet10 has 10 categories and a total of 3991 models, and the test set has 10 categories and a total of 908 models. The ViT-Base (ViT-B/16) architecture serves as the backbone of the classification framework. Each input image is resized to 224 × 224 and divided into non-overlapping 16 × 16 patches, resulting in 196 tokens per view. The encoder consists of 12 Transformer layers, each with 12 self-attention heads. Token embeddings have a dimensionality of 768, and each MLP block contains a hidden layer with 3072 dimensions, corresponding to an expansion ratio of 4.0. GELU is used as the activation function, and Layer Normalization is applied before both the attention and MLP modules. A learnable positional encoding is added to each token, and a class token is prepended to the input sequence to serve as the final aggregated representation for classification.

The core objective of this study is to evaluate the effectiveness of the proposed ViT-GE architecture for multi-view 3D object classification. We aim to assess whether the combination of grouped spatial attention,

channel attention, and transformer-based encoding improves recognition accuracy while maintaining computational efficiency. Experiments are conducted on two standard benchmark datasets, ModelNet10 and ModelNet40, which represent widely adopted evaluation protocols in the 3D shape classification field. Both quantitative performance and component-level ablation analyses are presented to validate the proposed contributions.

The experimental procedure includes data preprocessing, model training, and evaluation. Multi-view images are generated by rendering each 3D model from six viewpoints with fixed angular intervals. All images are resized to 224 × 224 and normalized. During training, the ViT-GE model is initialized with ViT-Base parameters and trained using the Adam optimizer with a cosine learning rate scheduler. Cross-entropy loss is used as the objective function. We evaluate classification performance using top-1 accuracy on the official test splits of ModelNet10 and ModelNet40. In addition to overall accuracy, ablation studies are conducted to analyze the effects of attention modules (GGCA, ECA), view grouping (G), and voting strategies (soft vs. hard voting). The classification results are shown in Table 1.

Table 1: Classification accuracy comparison of ViT-GE and plain ViT on ModelNet10 and ModelNet40 datasets

Network	Dataset	Accuracy
VIT (baseline)	Modelnet10	88.71
	Modelnet40	86.42
VIT-GE+SV	Modelnet10	93.50
	Modelnet40	91.96

The tabular data in Table 1 compares the classification accuracy of the baseline Vision Transformer (ViT) and the proposed ViT-GE framework on the ModelNet10 and ModelNet40 datasets. The experimental results show that the ViT-GE model, which incorporates Global Grouped Coordinate Attention (GGCA), Efficient Channel Attention (ECA), and a soft voting (SV) mechanism, achieves an accuracy of 93.50% on ModelNet10 and 91.96% on ModelNet40. In contrast, the plain ViT model achieves 88.71% and 86.42% on the respective datasets. This corresponds to relative improvements of 4.79% on ModelNet10 and 5.54% on ModelNet40. These improvements demonstrate that the introduced attention mechanisms effectively enhance the model's capability to capture global spatial dependencies and fine-grained feature representations. Moreover, the soft voting mechanism enables more refined aggregation of multi-view predictions by leveraging probabilistic outputs rather than discrete decisions, thus further improving overall classification performance.

The tabular data in Table 2 shows the classification accuracy performance of the four network models under different voting algorithms (no-voting, HV, SV). Compared to the original ViT model, GGCA and ECA introduce only a negligible increase in FLOPs and parameter count (from 17.58 G/85.83 M to 17.60 G/85.90 M), yet yield substantial gains in classification accuracy. It can be seen that the voting algorithm has a significant improvement on the model performance as a whole, among which SV has the most significant improvement effect. The accuracy of the basic ViT model is only 88.71% when voting is not used. After the introduction of HV and SV, it increased to 89.98% and 91.74%, respectively, indicating a significant improvement. ViT+GGCA's best feature is its combination with the voting algorithm (up to 92.95%), even though its accuracy is slightly higher than ViT's without voting. ViT+ECA is better than the first two in all three cases and reaches 92.29% under the SV strategy. The most outstanding performance is the ViT-GE model, whose accuracy has reached 91.81% without voting, and it is even increased to 93.50% with the SV strategy, the highest in the table. In general, the combination of model structure optimization and a voting

mechanism is an effective way to improve the accuracy of multi-view image classification, especially the combination of ViT-GE and SV voting, which has the best effect.

Table 2: The impact of GGCA and ECA embedding methods on ViT classification

Network model	Flops (G)	Params (M)	Voting algorithm	Accuracy
ViT (baseline)	17.58	85.83	×	88.71
			HV	89.98
			SV	91.74
ViT+GGCA	17.60	85.90	×	86.71
			HV	91.96
			SV	92.95
ViT+ECA	17.58	85.83	×	88.40
			HV	90.64
			SV	92.29
VIT-GE	17.60	85.90	×	91.81
			HV	92.84
			SV	93.50

Table 3 analyzes the impact of GGCA and ECA on classification performance at different embedding positions. This paper attempts to embed GGCA and ECA modules at different stages of ViT and regards ECA and GGCA as a whole, represented by a model. The insertion module positions include before input (Model 1), between multi-head attention and MLP (Model 2), after output and before classification head (Model 3), and fused into the self-attention module (Model 4), as shown in Fig. 7.

Table 3: The impact of different embedding positions of GGCA and ECA on ViT classification

Voting algorithm	Model1	Model2	Model3	Model4
HV	92.84	90.85	89.53	89.42
SV	93.50	91.62	90.63	89.31

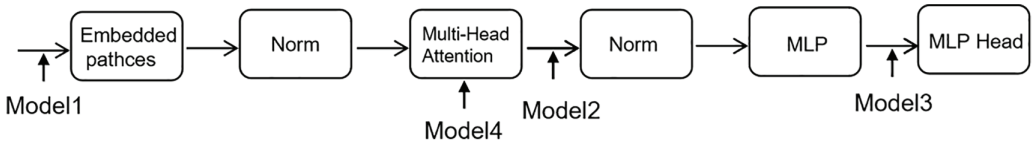

Figure 7: Insertion position of the module

The experimental results show that when the model is embedded before the input (Model 1), the classification accuracy is the highest, with the SV algorithm reaching 93.50% and the HV algorithm reaching 92.84%. When the model is embedded in other positions (such as 89.42% of Model 4), the classification performance is significantly worse. This experimental result indicates that the embedding position of the attention mechanism has a significant impact on the performance of the model. Model 1 is better at understanding the overall context of multi-view data and improving the importance of key features, which leads to the best classification results

The tabular data in Table 4 are compared with other classic methods, such as DeepPano (88.66%) and Geometry Image (88.40%). It can be seen that the classification performance of this method is better than these traditional methods and slightly higher than CNN-Voting (92.85%). From Table 4, we can draw the following conclusions: The voting mechanism plays a crucial role in the fusion effect of multi-view data, particularly the SV algorithm. This algorithm is more suitable for processing complex 3D model classification tasks, effectively reducing the error of a single view and improving classification stability.

Table 4: Accuracy comparison based on view-based methods

Algorithm	Accuracy
DeepPano [25]	88.66
Geometry Image [26]	88.40
CNN-Voting [27]	92.85
PVR [28]	92.70
Efficient Net	90.60
VIT-GE-HV	92.84
VIT-GE-SV	93.50

The tabular data in Table 5 show the impact of different group numbers (G) on the performance of the GGCA module. Among all configurations, the highest accuracy is achieved when G is set to 4, where Soft Voting (SV) reaches 92.95% and Hard Voting (HV) also performs well at 91.96%. Compared with other settings, G = 4 demonstrates superior fusion capability, indicating that a proper grouping strategy enhances the representational capacity of the GGCA module. Moreover, SV consistently outperforms HV in most cases, validating its advantage in multi-view information integration. It effectively mitigates single-view errors and improves the overall classification stability.

Table 5: Performance analysis of GGCA under multiple group numbers

G	HV	SV
2	91.96	92.84
3	90.63	91.40
4	91.96	92.95
6	91.07	91.51
8	91.74	92.73

The tabular data in Table 6 compared the classification performance and computational complexity of Vision Transformer (ViT) models enhanced with various attention modules, including SE, SA, CBAM, ECA, and the proposed GGCA. The baseline ViT without attention achieves 88.71% accuracy. Introducing attention modules significantly improves performance, with the GGCA module achieving the highest accuracy of 92.95%. Among all modules, GGCA not only delivers the best accuracy but also maintains a relatively low computational cost (17.60 GFLOPs and 85.90 M parameters), which is comparable to other lightweight modules like SE and ECA. Although CBAM and SA slightly increase the number of parameters and FLOPs, their performance gains are limited compared to GGCA. These results demonstrate that GGCA provides an optimal balance between performance and efficiency in multi-view 3D classification tasks.

Table 6: Classification accuracy under different attention modules

Class	No attention	SE	SA	CBAM	ECA	GGCA
ACC (%)	88.71	90.12	90.14	90.47	92.29	92.95
Flops (G)	17.58	17.58	17.59	17.60	17.58	17.60
Params (M)	85.83	85.83	85.85	85.88	85.83	85.90

Fig. 8 looks at how well the two ensemble learning methods, soft voting and hard voting, perform in classifying objects into 10 categories. Experimental data show that Soft Voting does much better than Hard Voting's 92.84% and the baseline method's 91.81%, achieving an overall accuracy of 93.50%. Experimental data show that Soft Voting significantly outperforms Hard Voting's 92.84% and the baseline method's 91.81% with an overall accuracy of 93.50%. From the perspective of subdivided categories, the four categories of bed, chair, monitor, and toilet all achieved a perfect recognition rate of 100% under Soft Voting, and bathtub and sofa also performed well (96%). However, categories such as dresser (80.23%), table (82%), and nightstand (89.53%) still have great recognition difficulties, which may be due to the morphological similarities between furniture items. According to the confusion matrix in Fig. 9, desk and table, as well as dresser and night, in the dataset are misclassified. By comparing the instances of nightstand and dresser as well as desk and table, it can be found that the overall shapes of the two models are extremely similar, and they differ only in local details, which are easy to cause confusion in the classification experiment and affect the classification effect of the model.

It is worth noting that Soft Voting has advantages in 7 categories, especially in difficult categories such as dresser (improved by 5.45%) and nightstand (improved by 4.65%), which proves that its probability weighting mechanism can more effectively deal with the problem of blurred category boundaries. This result not only shows that the soft voting strategy is better, but it also highlights areas for future research, such as improving features in tough categories, combining different strategies, and other ways to make it better.

Figure 8: Accuracy comparison across different categories

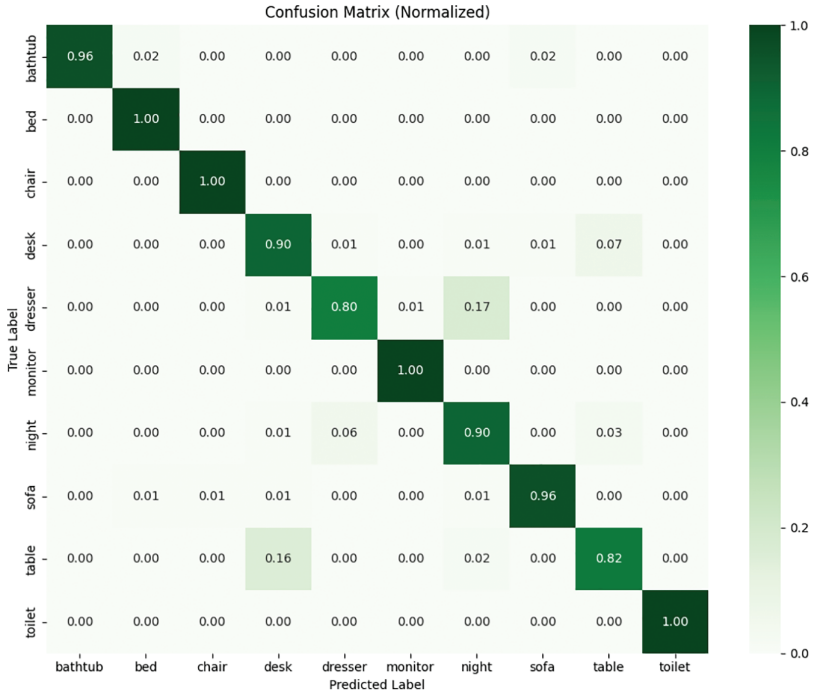

Figure 9: SV confusion matrix

The analysis of Tables 1–6, and Figs. 7–10 shows that the method suggested in this paper has clear benefits in the voting mechanism, improving the attention mechanism, and extracting features. The SV voting algorithm works better for multi-view classification tasks compared to the HV algorithm. The SV voting algorithm is more suitable for multi-view classification tasks than the HV algorithm. The embedding position of the GGCA module has a significant impact on the model performance, and the classification accuracy of the model is further improved after the introduction of GGCA. These experimental results verify the superiority and practical application value of the method in this paper on the ModelNet10 dataset. Although this work focuses on the effectiveness of ViT-GE as a whole, future research will further investigate the placement and interaction of attention modules within the network.

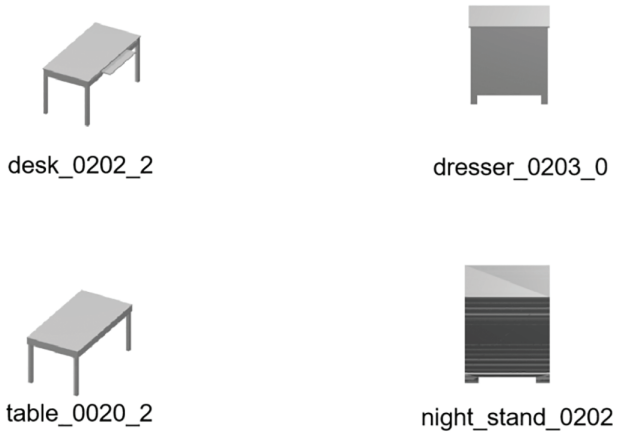

Figure 10: Misclassified model instances

4 Conclusion and Future Research

To further improve the accuracy of 3D model classification, this paper proposes a 3D model classification method that integrates ViT architecture, the ECA channel attention mechanism, the GGCA global spatial attention mechanism, and a voting strategy. This method first projects the 3D model from multiple perspectives to generate 2D views in six directions and uses ViT as the feature extraction backbone network. The ECA mechanism is introduced to enhance the inter-channel feature modeling capability, and the GGCA mechanism is combined to strengthen the key area response in the spatial dimension, thereby comprehensively improving the feature expression capability. Each view group shares network parameters to ensure a compact model structure and stable training. Subsequently, a voting mechanism is used to fuse the discriminative features under multiple view groups, effectively alleviating the problem of multi-view information redundancy and conflict. Compared with the traditional multi-view method of convolutional neural networks, this method achieves more efficient collaborative expression in local channel modeling and global spatial perception. Experimental results further verify the effectiveness of the proposed method. In terms of voting strategy, the VIT-GE network achieved an accuracy of 92.84% on the ModelNet10 dataset when using horizontal voting (HV), and the accuracy was increased to 93.50% after introducing the soft voting (SV) strategy. Overall, VIT-GE showed significant advantages in accuracy, robustness, and feature expression ability, proving the broad prospects and practical value of the three-dimensional model classification method that integrates the Transformer structure and dual attention mechanism and combines the voting strategy in practical applications.

Acknowledgement: The authors declare that there are no non-author contributors to acknowledge for this study.

Funding Statement: This research was funded by the project supported by the Heilongjiang Provincial Natural Science Foundation of China (Grant Number LH2022F030).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Xueyao Gao, Fang Yuan; data collection: Chunxiang Zhang; analysis and interpretation of results: Xueyao Gao, Fang Yuan, Chunxiang Zhang; draft manuscript preparation: Xueyao Gao, Fang Yuan. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data used in this paper can be requested from the corresponding author.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Biasotti S, Cerri A, Abdelrahman M, Aono M, Hamza AB, El-Melegy M, et al. SHREC'14 track: retrieval and classification on textured 3D models. In: Proceedings of the 7th Eurographics Workshop on 3D Object Retrieval; 2014 Apr 6; Strasbourg, France. p. 111–20.
2. Li ZP, Su HL, Wu Y, Zhang QH, Yuan CA, Gribova V, et al. Hierarchical multiview top-k pooling with deep-Q-networks. IEEE Trans Artif Intell. 2024;5(6):2985–96. doi:10.1109/TAI.2023.3334261.
3. Li ZP, Wang SG, Zhang QH, Pan YJ, Xiao NA, Guo JY, et al. Graph pooling for graph-level representation learning: a survey. Artif Intell Rev. 2024;58(2):45. doi:10.1007/s10462-024-10949-2.
4. Wang H, Liu C, Cai Y, Chen L, Li Y. YOLOv8-QSD: an improved small object detection algorithm for autonomous vehicles based on YOLOv8. IEEE Trans Instrum Meas. 2024;73:2513916. doi:10.1109/TIM.2024.3379090.
5. Tan M, Le QV. EfficientNet: rethinking model scaling for convolutional neural networks. In: Proceedings of the 36th International Conference on Machine Learning; 2019 Jun 9–15; Long Beach, CA, USA. p. 6105–14.

6. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015 Jun 7–12; Boston, MA, USA. p. 1–9. doi:10.1109/CVPR.2015.7298594.
7. Ilamchezhian J, Raj VC. An efficient parallel approach for 3D point cloud image segmentation using OpenMP. Digit Image Process. 2013;5:229–38.
8. Lagahit MLR, Liu X, Xiu H, Kim T, Kim KS, Matsuoka M. Learnable resized and Laplacian-filtered U-Net: better road marking extraction and classification on sparse-point-cloud-derived imagery. Remote Sens. 2024;16(23):4592. doi:10.3390/rs16234592.
9. Charles RQ, Hao S, Mo K, Guibas LJ. PointNet: deep learning on point sets for 3D classification and segmentation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 Jul 21–26; Honolulu, HI, USA. p. 77–85. doi:10.1109/CVPR.2017.16.
10. Song Y, He F, Duan Y, Liang Y, Yan X. A kernel correlation-based approach to adaptively acquire local features for learning 3D point clouds. Comput Aided Des. 2022;146(2):103196. doi:10.1016/j.cad.2022.103196.
11. Hassan R, Fraz MM, Rajput A, Shahzad M. Residual learning with annularly convolutional neural networks for classification and segmentation of 3D point clouds. Neurocomputing. 2023;526(4):96–108. doi:10.1016/j.neucom.2023.01.026.
12. Biasotti S, Cerri A, Aono M, Ben Hamza A, Garro V, Giachetti A, et al. Retrieval and classification methods for textured 3D models: a comparative study. Vis Comput. 2016;32(2):217–41. doi:10.1007/s00371-015-1146-3.
13. Wu Z, Song S, Khosla A, Yu F, Zhang L, Tang X, et al. 3D ShapeNets: a deep representation for volumetric shapes. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015 Jun 7–12; Boston, MA, USA. p. 1912–20. doi:10.1109/CVPR.2015.7298801.
14. Xu X, Todorovic S. Beam search for learning a deep Convolutional Neural Network of 3D shapes. In: 2016 23rd International Conference on Pattern Recognition (ICPR); 2016 Dec 4–8; Cancun, Mexico. p. 3506–11. doi:10.1109/ICPR.2016.7900177.
15. Kim S, Chi HG, Ramani K. Object synthesis by learning part geometry with surface and volumetric representations. Comput Aided Des. 2021;130(1–4):102932. doi:10.1016/j.cad.2020.102932.
16. Ma Z, Zhou J, Ma J, Li T. A novel 3D shape recognition method based on double-channel attention residual network. Multimed Tools Appl. 2022;81(22):32519–48. doi:10.1007/s11042-022-12041-9.
17. Cai W, Liu D, Ning X, Wang C, Xie G. Voxel-based three-view hybrid parallel network for 3D object classification. Displays. 2021;69(1):102076. doi:10.1016/j.displa.2021.102076.
18. He Y, Xia G, Luo Y, Su L, Zhang Z, Li W, et al. DVFENet: dual-branch voxel feature extraction network for 3D object detection. Neurocomputing. 2021;459(10):201–11. doi:10.1016/j.neucom.2021.06.046.
19. Denisenko AI, Krylovetsky AA, Chernikov IS. Integral spin images usage in deep learning algorithms for 3D model classification. J Phys Conf Ser. 2021;1902(1):012114. doi:10.1088/1742-6596/1902/1/012114.
20. Hegde V, Zadeh R. Fusionnet: 3D object classification using multiple data representations. arXiv:1607.05695. 2016.
21. Jin X, Li D. Rotation prediction based representative view locating framework for 3D object recognition. Comput Aided Des. 2022;150:103279. doi:10.1016/j.cad.2022.103279.
22. Zhu F, Xu J, Yao C. Local information fusion network for 3D shape classification and retrieval. Image Vis Comput. 2022;121(8):104405. doi:10.1016/j.imavis.2022.104405.
23. Liu AA, Guo FB, Zhou HY, Li WH, Song D. Semantic and context information fusion network for view-based 3D model classification and retrieval. IEEE Access. 2020;8:155939–50. doi:10.1109/access.2020.3018875.
24. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, et al. An image is worth 16 × 16 words: transformers for image recognition at scale. arXiv:2010.11929. 2020.
25. Shi B, Bai S, Zhou Z, Bai X. DeepPano: deep panoramic representation for 3-D shape recognition. IEEE Signal Process Lett. 2015;22(12):2339–43. doi:10.1109/LSP.2015.2480802.
26. Sinha A, Bai J, Ramani K. Deep learning 3D shape surfaces using geometry images. In: Computer Vision—ECCV 2016. Cham, Swizterland: Springer International Publishing; 2016. p. 223–40. doi:10.1007/978-3-319-46466-4_14.

27. Si Q, Qin F. 3D model classification and retrieval based on CNN and voting scheme. J Comput-Aided Des Comput Graph. 2019;31(2):303–14. doi:10.3724/SP.J.1089.2019.17160.
28. Zhou Y, Zeng F, Qian J, Han X. 3D shape classification and retrieval based on polar view. Inf Sci. 2019;474(11):205–20. doi:10.1016/j.ins.2018.09.051.