VGNet: Multimodal Feature Extraction and Fusion Network for 3D CAD Model Retrieval

Feiwei Qin, Gaoyang Zhan, Meie Fang, C. L. Philip Chen, *Fellow, IEEE*, and Ping Li, *Member, IEEE*

***Abstract*—The reuse of 3D CAD models is crucial for industrial manufacturing because it shortens development cycles and reduces costs. Significant progress has been made in deep learning-based 3D model retrievals. There are many representations for 3D models, among which the multi-view representation has demonstrated a superior retrieval performance. However, directly applying these 3D model retrieval approaches to 3D CAD model retrievals may result in issues such as the loss of the engineering semantic and structural information. In this paper, we find that multiple views and B-rep can complement each other. Therefore, we propose the view graph neural network (VGNet), which effectively combines multiple views and B-rep to accomplish 3D CAD model retrieval. More specifically, based on the characteristics of the regular shape of 3D CAD models, and the richness of the attribute information in the B-rep attribute graph, we separately design two feature extraction networks for each modality. Moreover, to explore the latent relationships between the multiple views and B-rep attribute graphs, a multi-head attention enhancement module is designed. Furthermore, the multimodal fusion module is adopted to make the joint representation of the 3D CAD models more discriminative by using a correlation loss function. Experiments are carried out on a real manufacturing 3D CAD dataset and a public dataset to validate the effectiveness of the proposed approach. Our source code and dataset are available at: https://github.com/xzzz011/VGNet.**

***Index Terms*—3D CAD model retrieval, graph neural network, multi-view, multimodal fusion, attention mechanism.**

Manuscript received 8 July 2023; revised 3 June 2024; accepted 13 July 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62072126, in part by the Fundamental Research Projects Jointly Funded by Guangzhou Council and Municipal Universities under Grant 2024A03J0394, in part by the Key Laboratory of Philosophy and Social Sciences in Guangdong Province of Maritime Silk Road of Guangzhou University under Grant GD22TWCXGC15, in part by the Aeronautical Science Foundation of China under Grant 2022Z0710T5001, in part by the Open Project Program of the State Key Lab of CAD&CG, Zhejiang University under Grant A2304. in part by The Hong Kong Polytechnic University (PolyU) under Grant P0048387, Grant P0042740, Grant P0044520, Grant P0043906, Grant P0049586, and Grant P0050657, and in part by the PolyU Research Institute for Sports Science and Technology under Grant P0044571. *(Corresponding author: Meie Fang.)*

Feiwei Qin and Gaoyang Zhan are with the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China (e-mail: qinfeiwei@hdu.edu.cn; tom514199242@gmail.com).

Meie Fang is with the Metaverse Research Institute, School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 511400, China (e-mail: fme@gzhu.edu.cn).

C. L. Philip Chen is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, and also with the Pazhou Lab, Guangzhou 510335, China (e-mail: philip.chen@ieee.org).

Ping Li is with the Department of Computing, the School of Design, and the Research Institute for Sports Science and Technology, The Hong Kong Polytechnic University, Hong Kong (e-mail: p.li@polyu.edu.hk).

I. Introduction

THE increasing quantity and complexity of 3D CAD models have made their retrieval and reuse a heightened topic of research interest. 3D CAD models are crucial for industrial manufacturing because they serve as the foundation for the entire product life cycle. According to statistics, the reuse rates of 3D CAD models are close to 75% [1]. In practical applications, the model designer selects a model that is similar to the requirements in the existing model library and makes a slight modification to obtain a new model. Therefore, a high-performance retrieval approach that can effectively reuse models plays a key role in the product life cycle and is an important factor in improving enterprises' core competitiveness. In recent years, with the development of computer technology, scholars have introduced computer vision and artificial intelligence for 3D model retrieval, proposed various methods [2]–[6], and achieved great success. Retrieval approaches for 3D models can be categorized into voxel-based, point-cloud-based and view-based approaches. However, none of the existing approaches are specifically tailored for 3D CAD model retrieval, as industry-standard 3D CAD models predominantly employ boundary representation (B-rep). B-rep is a precise representation containing rich engineering semantics information, namely, comprehensive geometric and topological data, as well as higher-level information such as design intent, design constraints and other implicit knowledge embedded within models [7]. Moreover, the existing 3D CAD model retrieval approaches cannot effectively describe and represent 3D CAD models: hence, the retrieval performance is not sufficient to meet the requirements of industrial applications.

This paper presents VGNet, a joint convolutional network of attribute graphs and multiple views for 3D CAD model retrieval. The motivation behind this method lies in addressing the limitations of using either multi-view data or graph structural data representations alone for expressing 3D CAD models, which are inherently constrained in their expressive capabilities. Therefore, the approach leverages the joint learning of features from both modalities to represent 3D CAD models. By employing image modal information, specifically multi-view features, to guide the representation of the graph structural features, the method enriches the expression of 3D CAD models with multimodal information, resulting in a more comprehensive and nuanced representation of the models. VGNet aims to combine the strengths of both representations to achieve more accurate and efficient model retrieval. VGNet is divided into three modules: feature extraction module, multi-head attention enhancement module,

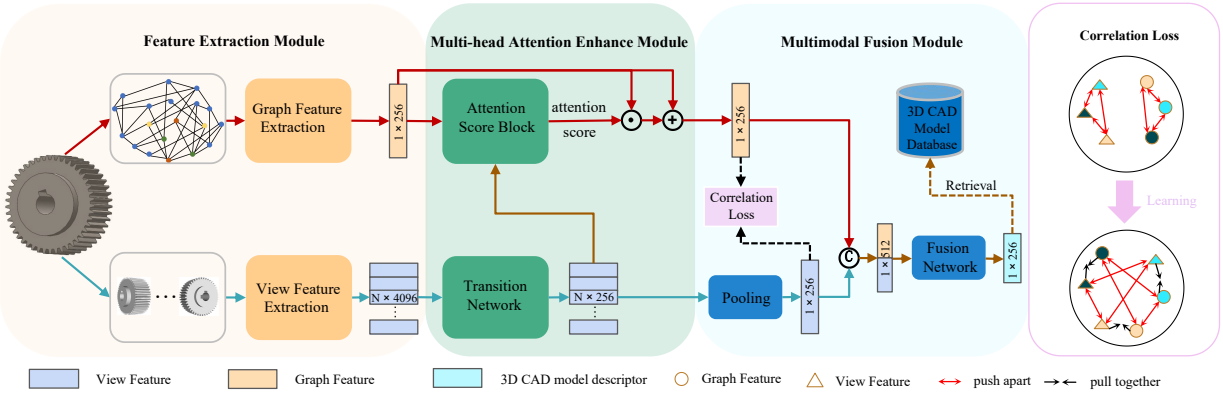


Fig. 1: Architecture of the proposed VGNet. VGNet takes two modalities of 3D CAD models as input: views and B-rep attribute graph. VGNet consists of three modules: Feature Extraction Module, Multi-head Attention Enhancement Module, and Multimodal Fusion Module. The Feature Extraction Module extracts features from two modalities of 3D CAD models respectively. Then, the extracted feature vectors are used as inputs to the Multi-head Attention Enhancement Module. The view features are passed through a transition network and interact with the graph features in the Attention Score Block to generate attention scores, which are then used to enhance the graph features. In the Multimodal Fusion Module, the pooled view features are combined with the enhanced graph features to produce the final descriptor for the 3D CAD model. The correlation loss function aims to minimize the distance between the features of different modalities from the same model.

and multimodal fusion module. The feature extraction module uses an improved convolutional neural network branch and a graph neural network branch to extract and abstract features from the two modalities of CAD models respectively. Since the multi-view and B-rep representations refer to the same objective object, there is a latent relationship between them. Therefore, this paper proposes a multi-head attention enhancement module that uses multi-view information to enhance B-rep information, effectively improving the discriminability of B-rep features. In the multimodal fusion module, a joint loss function is designed. The main contributions of this paper can be summarized as:

- We introduce a novel multimodal approach for 3D CAD model retrieval that leverages both B-rep and multi-view data in a joint learning framework. This method capitalizes on the inherent complementarity between these two modalities to enhance the retrieval performance.
- A specific convolutional neural network branch within VGNet is meticulously designed to extract view features from CAD models, with improved ResBlocks tailored to capture appearance characteristics of solid models. These blocks are engineered to focus on visual aspects that are critical for model differentiation. Concurrently, the graph neural network branch employs specialized graph convolution and pooling operations, e.g., BAGConv and BAGPool, to aggregate hierarchical graph features. This design ensures that rich and informative B-rep attribute graph information is preserved, providing a detailed structural representation that complements the visual data.
- An effective multi-head attention enhancement module is designed for enriching the B-rep graph features based on multi-view information. This module is capable of dynamically focusing on the most informative features from the multi-view data, thereby enhancing the representational capacity of the B-rep graph features and improving the overall retrieval performance of VGNet.
- Multimodal fusion is designed to integrate graph features with view features, harnessing both topological and appearance information for a robust multimodal representation of 3D CAD models, which ensures a more cohesive and effective representation for 3D CAD model retrieval.

II. Related Work

This section presents the existing approaches that relate to the research topic of this article, including voxel-based 3D model retrieval approaches, point-cloud-based 3D model retrieval approaches, view-based 3D model retrieval approaches, and B-rep-based 3D CAD model retrieval approaches.

A. Voxel-based 3D model retrieval approaches

The voxel-based approaches use a set of voxels in the three-dimensional space to represent the 3D model and then establish the neural network based on the voxel to extract the features and complete the recognition and retrieval of the 3D model. Wu et al. [8] proposed the 3D ShapeNets network and 3D models are represented as the probability distribution on the 3D voxel grid, where each voxel is represented by a binary vector that indicates its presence or absence in the grid. The 3D grid's voxel characteristics of the 3D model are learned using a deep convolutional network. However, the use of full-voxel-based models may incur significant computational and memory costs, especially as the voxel resolution increases. To address this issue, Wang et al. [9] and Riegler et al. [10] employed tree structures that selectively perform computations on the occupied voxels and discard empty grids. While Octree-based approaches exhibit good performance, preserving the fine details of input data needs high-resolution representations. Insufficient resolution will result in several points being aggregated in the same voxel grid, leading to a loss of distinguishability.

B. Point-cloud-based 3D model retrieval approaches

Point-cloud-based models utilize a collection of points that are sampled from a 3D shape's surface as the input data. While they retain more complete structural information, their unstructured and irregular nature makes them unsuitable for conventional 2D CNN. PointNet [11] was the first to introduce a deep neural network to handle disordered point cloud data. To ensure permutation invariance, PointNet applied symmetric function-max pooling and used a spatial transform block. Furthermore, In PointNet++ [12], the PointNet module was used for local points and local features were aggregated in a hierarchical way. Kd-Network [13] extracted and gathered features by subdividing points on Kd-trees. DGCNN [14] proposed EdgeConv operation to better exploit local structure information and obtain edge features among points. Li et al. [15] proposed the χ-Conv operation, which can extract local patch features. DSACNN [16] employed novel convolutional operations to process point cloud data directly and also dynamically focused on local semantic information. CurveNet [17] proposed a long-distance point cloud feature extraction operator, which can provide additional geometric information for the whole point cloud features by using guided walk on the isomorphic graph to gather a series of curves of fixed number and length. DCNet [18] is a network used to solve fine-grained 3D point cloud classification tasks, and a novel mechanism of mutual complementarity is designed between the attention block and the dynamic sample confusion block. RECON [19] unifies contrastive and generative modeling for 3D representation multimodal learning, achieving state-of-the-art performance by combining the strengths of both paradigms. FGPNet [20] is the first weakly supervised network for fine-grained 3D point clouds task. Unlike supervised fine-grained classification approaches that use category labels and other manually annotated information, FGPNet has developed a unified framework that only uses category labels as input to process local geometric details and global spatial structures. However, such approaches lose some structural information and topological information of the model in the process of sampling as a point set.

C. View-based 3D model retrieval approaches

View-based models are often used to describe 3D objects through multiple views from various angles. Handcraft descriptors are investigated in the beginning. The initial and classic view-based 3D descriptor, known as Lighting Field descriptor [21], consists of a collection of ten views obtained from a hemisphere. The similarity between two 3D objects is measured using probabilistic matching in [22], [23]. Many deep neural network based models have been extensively studied with the development of deep learning. For example, Su et al. [24] proposed a convolutional neural network for multi-view that employs a weight-shared CNN to generate features for each view. Furthermore, Feng et al. [25], proposed a group based approach to mine the relationship among views. Due to the high maturity of deep neural networks in 2D images, this kind of approaches can capture a wealth of appearance information. However, due to the projection from different angles, some details are lost and the internal structural information of 3D model cannot be used effectively [26].

D. B-rep-based 3D CAD model retrieval approaches

There are few neural networks that have the ability to directly process B-rep data. Before the maturity of deep learning, Mohamed et al. [27], [28] first converted the CAD model into an attribute graph representing its topological structure, where the vertices correspond to the faces of the model, the lines between vertices correspond to the edges of the model and the attribute values in the graph represent the spatial geometry information in the 3D CAD model. Then, models are compared by inexact graph matching using the attribute graph. However, the performance of this approach cannot meet the requirements of the industry. With the rapid development of deep learning, many new approaches have been proposed. Jayaraman et al. [29] proposed the UV-Net network architecture, which uses two-dimensional UV coordinates to represent the geometric shape information of 3D models and efficiently combines image convolutional neural networks with hierarchical graph neural networks to achieve retrieval. Colligan et al. [30] used hierarchical graphs to represent 3D CAD models to complete feature recognition. Mandelli et al. [31] proposed the CADGCN method, which uses GNNs for classification and retrieval by converting 3D models in CAD format into graph data for processing. In addition, Bai et al. [32] completed the partial retrieval of 3D CAD models by constructing hierarchical descriptors based on B-rep and feature graphs. Although these approaches can capture the internal structure and topology information of 3D models well, some important appearance information could be ignored.

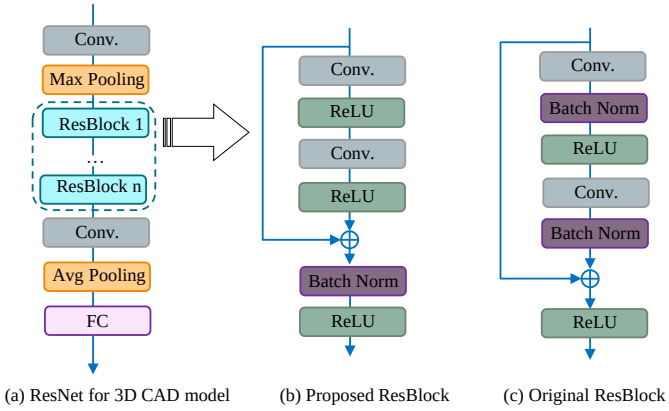
III. METHOD

In this section, we introduce the proposed novel multimodal learning framework VGNet for 3D CAD model retrieval in detail. VGNet presents an innovative approach that effectively leverages both B-rep and multi-view data to enhance retrieval performance. This method capitalizes on the complementary nature of these two modalities, providing a more comprehensive and robust representation for accurate and efficient retrieval of 3D CAD models. Fig. 1 depicts the overall network architecture, which takes the B-rep attribute graph and multi-views as inputs to the VGNet's respective branches. The output of VGNet is a 3D CAD model descriptor. VGNet consists of three modules: *Feature Extraction Module*, *Multi-head Attention Enhancement Module*, and *Multimodal Fusion Module*. In the feature extraction module, we propose an innovative feature extraction method tailored for solid models. In the view feature extraction network, we employ an improved ResNet [33], which is more suitable for capturing the nuanced visual details and geometric features from 3D CAD models. In the B-rep attribute graph feature extraction network, novel graph convolutional and pooling operations are designed to fully leverage the rich topological and geometric information inherent in B-rep data. Based on the characteristics of the B-rep attribute graph feature and multiple view features, an

effective multi-head attention enhancement module is designed for mining the relationship between the two modalities, and enriching the B-rep graph features with multi-view information, leading to a more comprehensive and accurate representation of 3D CAD models. In the multimodal fusion module, a carefully designed joint loss function and a fusion network are implemented to seamlessly integrate the complementary features extracted from B-rep graphs and multi-view images. This integration strategy is pivotal in enhancing the discriminative power of the final CAD model descriptor, thereby improving the overall retrieval performance of VGNet.

A. *Feature extraction module*

Different modalities of data, i.e. graph and views, are input respective branches in the feature extraction module. In the view branch, we utilize multiple convolutional layers, specifically designed for CAD models, to extract features from projected views of the model. In the B-rep attribute graph branch, based on the rich attribute information contained in the graph, the graph convolutional layer and the graph pooling layer that can fully utilize attributes are designed to extract graph features. In addition, we combine the different levels of features to get a B-rep attribute graph feature that contains structural information and hierarchical information of the 3D CAD model.



(a) ResNet for 3D CAD model (b) Proposed ResBlock (c) Original ResBlock

Fig. 2: Details of the view branch network.

View branch: People can distinguish different models by observing the appearance of 3D CAD models, which shows that the appearance of models contains rich recognizable information. The effect of people viewing a 3D model from a specific angle is the same as a 2D image. Hence, in this paper, the multi-view approach is utilized to obtain the appearance information of the 3D CAD model. Each 3D CAD model is represented by 12 rendered views captured with a predefined camera array. These cameras surround the model every 30 degrees. In addition, residual learning is good at capturing view features, hence we construct residual blocks. As shown in Fig. 2(a), we use the ResBlocks to construct a ResNet for extracting view features. To be more accommodating to the retrieval of 3D CAD models, we adopt a new way to implement ResBlock [Fig. 2(b)]. In comparison to the original ResBlock [Fig. 2(c)], we remove the batch normalization layers from residual mapping. This is because, in the industrial application field, 3D CAD models are often more structured and regular due to they are constrained by engineering specifications and industry standards. Hence, the normalization of features by the batch normalization layer might be less effective for 3D CAD model retrieval. The implementation of ResBlock is as follows: the residual mapping consists of two convolutional layers followed by an activation function ReLU. The input and output of the residual mapping are added together, and then the output of ResBlock is obtained by passing through a batch normalization layer and a ReLU activation function.

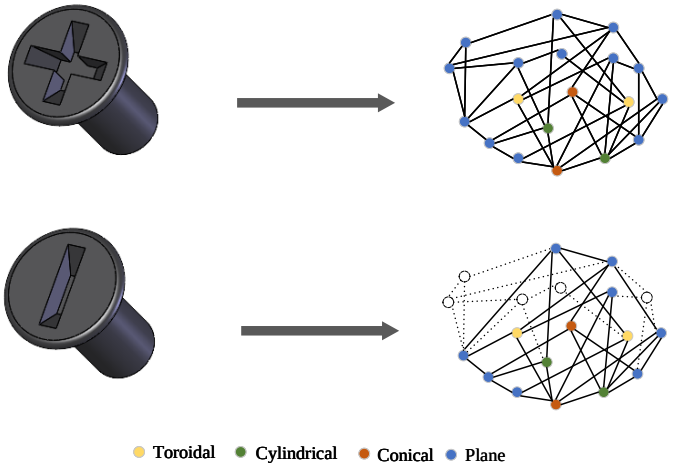
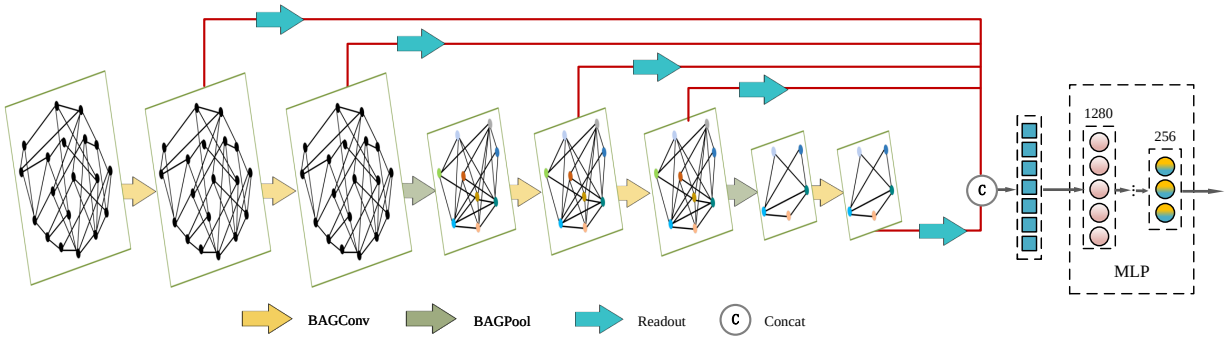


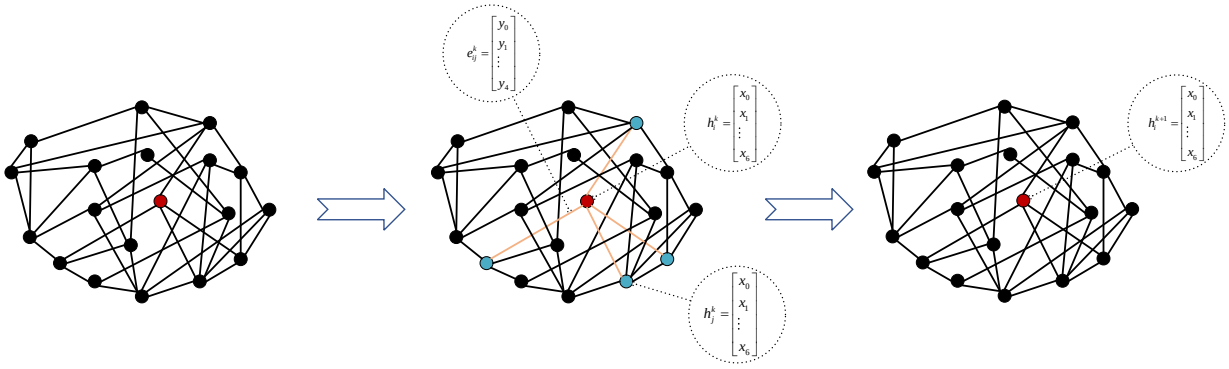
Fig. 3: Modification of the bolt part results in changes to the B-rep attribute graph.

Graph branch: The Standard for the Exchange of Product model data (STEP) [34] file plays an important role in geometric data exchange as an intermediate exchange format for 3D CAD. This paper extracts information from STEP files to obtain B-rep attribute graphs. As shown in Fig. 3, the left side shows the 3D CAD model of a bolt, and the right side shows its corresponding generated B-rep attribute graph. In the B-rep attribute graph, the node represents the surface in the CAD model, and the edge represents the curve between intersecting surfaces. Considering the tradeoff between the precision of representation and the computational cost, the attributes in the B-rep attribute graph are defined based on the particular information extracted from the 3D CAD model, e.g. surface type, normal vector, tangent vector as face attributes, and curve type, direction, length as edge attributes, and they could be adjusted according to specific tasks. The B-rep attribute graph is associated with the structure of the 3D CAD model, and the farther the distance between nodes, the weaker the mutual dependence of the structure in the CAD model.

After obtaining the B-rep attribute graph, this paper uses a graph neural network to extract information from it. GCN [35] is a typical graph neural network that can capture global information of graphs and effectively represent node features. However, the convolution operation in GCN requires the entire graph to be stored in memory, which can be very memory-intensive and limit its ability to handle large graphs. Additionally, GCN requires information of the entire graph's structure during training, which limits its inference capabilities. In the

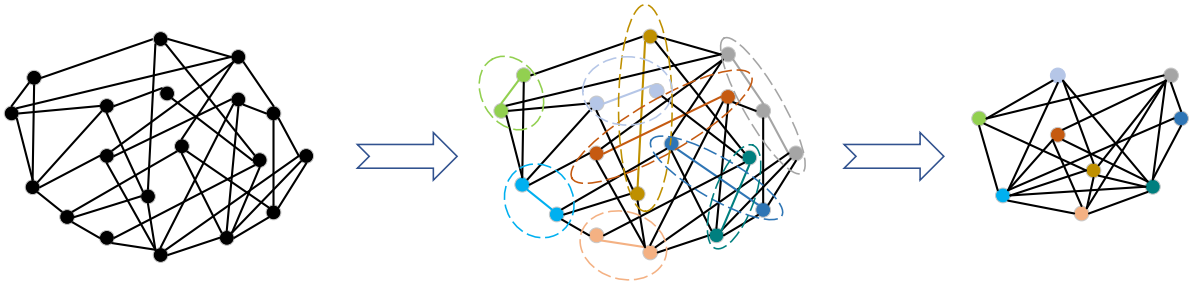


(a) Graph feature extraction network for B-rep attribute graphs consists of three components: BAGConv, BAGPool, and residual connection structure.



① Select the nodes to be updated ② Sample neighbor nodes and edges ③ Aggregate and update the features of the node

(b) Visual illustration of BAGConv. It considers both the node attributes and edge attributes of B-rep attribute graph during the convolution process.



① B-rep attribute graph ② Calculate the importance scores ③ Merge to generate new nodes

(c) Visual illustration of BAGPool. It is a pooling mechanism based on edge contraction, which preserves the original structural information of the graph during the pooling process while utilizing the node and edge attributes of the B-rep attribute graph.

Fig. 4: Details of the graph branch network. A graph neural network, as shown in (a), is designed to extract features from the B-rep attribute graph. (b) and (c) depict the BAGConv convolution operation and BAGPool pooling operation, respectively.

scenario of 3D CAD model reuse, 3D CAD model modifications are usually involved, which means that the corresponding B-rep attribute graph will change frequently. For example, In Fig. 3, compared to the CAD model in the upper part, the CAD model in the lower part removes a groove, resulting in significant changes to nodes in local regions of the B-rep attribute graph. This indicates that scenarios involving the reuse of 3D CAD models require strong inference capabilities, which GCN does not provide. Therefore, this paper proposes a network architecture shown in Fig. 4(a). In this architecture, the BAGConv (B-rep Attribute Graph Convolution) convolutional layer is designed with reference to SAGEConv [36], the BAGPool (B-rep Attribute Graph Pooling) pooling layer is designed with reference to EdgePool [37], and the residual connections are adopted to transfer shallow graph features to deep networks to supplement local information.

BAGConv completes the convolution by considering adjacent node attributes and edge attributes at the same time, as shown in Fig. 4(b). Firstly, select a node to be updated, which is represented by a red node in Fig. 4(b). Secondly, sample the neighborhood of the node to be updated with variable proportions, obtaining the features of the sampled neighbor nodes and corresponding edge features, which are respectively represented by blue nodes and yellow lines in the figure. Finally, node updating is performed using an update function that leverages both node and edge features. BAGConv can be

represented in the following as:

$$h_i^{(k+1)} = \sigma(W^{(k)} \cdot [h_i^{(k)}, \text{Mean}(f_\Delta^{(k)}(e_{ij}^{(k)}) \odot h_j^{(k)}, \forall j \in \xi_\epsilon(\mathcal{N}(i)))]), \tag{1}$$

where $h_i^{(k+1)}$ represents the newly generated feature vector of the i-th node in the (k+1)-th layer. $e_{ij}^{(k)}$ represents the edge features between node i and node j, and f_Δ is a linear projection used for projecting the edge to node feature space. $\{h_j^{(k)}, \forall j \in \xi_\epsilon(\mathcal{N}(i))\}$ represents the sampled neighbor nodes, where $\mathcal{N}(i)$ represents the neighbor nodes of node i, ξ represents the sampling function and ϵ represents the sampling rate. The Mean function calculates the average of the sampled neighbor nodes along each dimension. W is the weight matrix. The resulting vector is then passed through a nonlinear activation function σ to produce the (k+1)-th layer representation for the target node. In the BAGConv convolutional operation, both surface features and curve features are used, resulting in learned graph features that contain rich attribute characteristics.

Algorithm 1 B-rep Attribute Graph Pooling

Input: B-rep attribute graph $G^{(k)}(V, E)$, node features $\{h_i^{(k)}, \forall i \in V\}$, edge features $\{e_{ij}^{(k)}, \forall ij \in E\}$
Output: B-rep attribute graph $G^{(k+1)}(V, E)$

```
Calculate the score for each edge using Eq. (2);
Sort the edges in descending order by their scores;
for each edge do
    if nodes at the ends of edge haven't been merged then
        Merge two nodes using Eq. (3);
    end if
end for
for unmerged nodes do
    Connected to adjacent merged subgraph;
end for
Update the edges using Eq. (4);
return B-rep attribute graph G^(k+1)(V, E);
```

In addition, for graph neural networks, graph pooling can be roughly divided into three categories: graph collapse, edge contraction, and top-K. In the B-rep attribute graph, edges reflect the relationship between adjacent nodes, and edge attributes contain important information. Therefore, pooling should be based on these characteristics. However, graph collapse and top-K do not make use of edges, hence this paper proposes BAGPool. Algorithm 1 shows the process of BAGPool. Firstly, the importance scores between each pair of adjacent nodes are calculated using node and edge features as:

$$\text{Score}_{ij}^{(k)} = \text{Softmax}(W^{(k)}(f_\Delta^{(k)}(e_{ij}^{(k)}) \odot [h_i{}^{(k)}, h_j{}^{(k)}]) + b^{(k)}), \tag{2}$$

where $\text{Score}_{ij}^{(k)}$ is the importance scores between node i and node j. $h_i^{(k)}$ is the feature vector of node i, $W^{(k)}$ is the weight matrix and $b^{(k)}$ is the bias. f_Δ is a fully connected layer $\text{FC}(5, 14)$ used for aligning feature dimensions.

After obtaining the importance scores, the edges in the graph are sorted in descending order based on these scores, and the nodes on both sides of each edge are merged. This ensures that the important nodes are prioritized for merging. The feature of the newly merged node is obtained as:

$$h_{ij}^{(k+1)} = \text{Score}_{ij}^{(k)} \odot (h_i^{(k)} + h_j^{(k)}), \tag{3}$$

where $+$ is a summation operation and $h_{ij}^{(k+1)}$ represents the feature vector of the new node. The updated equation for the edge features after pooling is as follow:

$$e_{ij}^{(k+1)} = \text{Score}_{ij}^{(k)} \odot e_{ij}^{(k)}, \tag{4}$$

the edge features are updated by using the importance score. Based on the connection relationship of nodes, the edge contraction pooling operation does not affect the structural information of the graph. This pooling operation merges neighbor nodes without losing attributes, making it more consistent with the characteristics of accurate representation of 3D CAD models represented by B-rep.

With the deepening of the convolution layer and pooling layer, the perceptual range of the network becomes larger and larger and the semantic information it contains becomes richer and richer. However, the representation of the deep layers is more inclined to the overall characteristics of the 3D CAD model and will ignore the local information contained in the shallow layers. Hence, the graph embedding of the shallow layers can be transmitted to the deep layers to supplement the local information that is ignored in the deep layers. Inspired by JKNet [38], we transfer graph embedding from shallow layers to deep layers and combine them by concatenation so that the final graph embedding contains rich hierarchical structure information. To obtain graph embedding from the B-rep attribute graph, we utilize the readout mechanism, which is defined as follows:

$$H^k = \{h_1^k, h_2^k, ..., h_n^k\}, \tag{5}$$

$$\mathcal{R}(H^k) = \sigma(\frac{1}{n}\sum_{i=1}^{n} h_i^k), \tag{6}$$

where H^k represents the set of nodes of the graph of the k-th layer. h_i^k denotes the feature of the i-th node in layer k. $\mathcal{R}$ means the Readout function. After the graph embedding of each layer is obtained, they are concatenated and aggregated, which is defined as:

$$g = \text{MLP}([\mathcal{R}(H^1), \mathcal{R}(H^2), ..., \mathcal{R}(H^k)]), \tag{7}$$

where MLP consists of two fully connected layers $\text{FC}(1280, 256) \rightarrow \text{FC}(256, 256)$. The graph embeddings of the middle layers are concatenated at the end and the graph embedding g of the 3D CAD model is obtained by MLP.

B. Multi-head attention enhancement module

In order to make full use of the view features and B-rep attribute graph feature, it is necessary to mine the latent information in each view and its relationship with the B-rep attribute graph. Inspired by Transformer [39], this paper adopts the multi-head attention mechanism in Fig. 5 to mine the latent information. The view features are used to assign

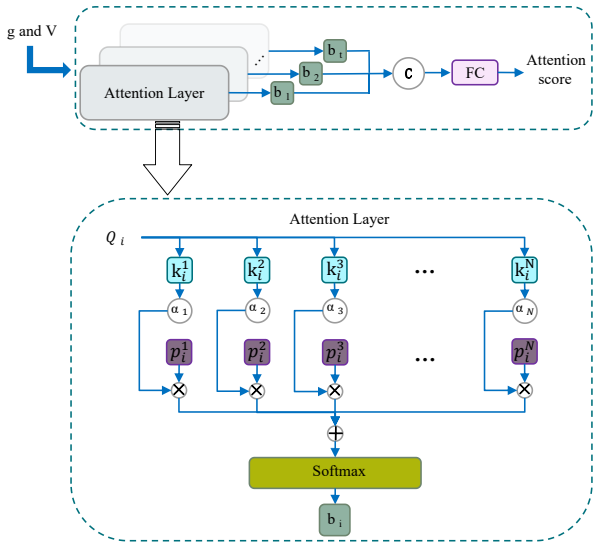


Fig. 5: Attention Score Block using a multi-head attention mechanism. g represents graph feature, while V represent view features. Q_i is operated with each k_i to obtain a similarity score α_i. p_i are then weighted and summed based on the similarity scores to obtain the attention score of this attention layer. Finally, the attention scores from each layer are concatenated and passed through a fully connected layer to obtain the final attention score.

weights to the graph feature, so as to enhance the graph feature. As shown in Fig. 1, after the view features are obtained, they are fed to the transition network to make view features and graph feature have the same vector dimension. In this paper, the transition network consists of fully connected layers $\text{FC}(4096, 1024) \rightarrow \text{FC}(1024, 256)$. The attribute graph feature and the transition view features interact to produce the attention score $\mathcal{S}(g, V)$, which is defined as:

$$\mathcal{S}(g, V) = \mathcal{A}(Q, K, P) = \mathcal{A}(L_1 g, L_2 V, L_3 V), \tag{8}$$

where the B-rep attribute graph feature is represented by $g \in \mathbb{R}^{1\times n_1}$ and view features is represented by $V = \{v_1, v_2, \ldots, v_N\} \subseteq \mathbb{R}^{N\times n_1}$. n_1 is the feature dimension. V is the collection of all view features of the model and v_i is a $1{\times}n_1$ tensor representing the feature of the i-th view. Q is equal to $L_1 g$, K is equal to $L_2 V$ and P is equal to $L_3 V$. L_1, L_2 and L_3 represent the linear projection layer. The relationship between the B-rep attribute graph and views is reasoned by the function $\mathcal{A}$, which is defined as:

$$\mathcal{A}(Q, K, P) = \vartheta(\text{Concat}(b_1, \ldots, b_t)), \tag{9}$$

$$b_i = \text{Softmax}(\frac{Q_i K_i^T}{\sqrt{n_1}})P_i, \tag{10}$$

where $b_i \in \mathbb{R}^{1\times \frac{n_1}{t}}$ represents the attention score generated by each head. In this paper, the number of headers is equal to 8, which means t is equal to 8. The generated b_i are concatenated and subsequently fed into a fully connected layer $\vartheta = \text{FC}(256, 256)$ to obtain the final attention score. Moreover, Q, K, P are divided into Q_i, K_i and P_i. $Q_i \in \mathbb{R}^{1\times \frac{n_1}{t}}$, $K_i = \{k_i^1, k_i^2, ..., k_i^N\} \subseteq \mathbb{R}^{N\times \frac{n_1}{t}}$ and $P_i = \{p_i^1, p_i^2, ..., p_i^N\} \subseteq \mathbb{R}^{N\times \frac{n_1}{t}}$. The final output attention score $\mathcal{S}(g, V) \in (0, 1)$. Attention score reflects the strength of the correlation between views and attribute graph. The higher the score, the stronger the correlation. On the contrary, the weaker. For feature enhancement, local information that is more relevant to cross-modal features should be given greater importance. So we use the attention score $\mathcal{S}(g, V)$ to enhance the B-rep attribute feature through residual connections:

$$\tilde{g} = g + \mathcal{S}(g, V) \odot g, \tag{11}$$

where $\tilde{g}$ represents the enhanced B-rep attribute feature.

C. Multimodal fusion module

To fuse the view features and attribute graph feature, this paper adopts max pooling to obtain the global view feature from the transition view features and uses the global view feature and the enhanced graph feature for cross-modal fusion. To make the feature representation of two modalities of the same 3D CAD model close to each other in the embedded space, this paper proposes a correlation loss function, which is defined as:

$$L_c = ||\delta(v) - \delta(\tilde{g})||_2, \tag{12}$$

where v is the global view feature and $\delta = \text{Sigmoid}(\log(|\cdot|))$ represents a normalized activation function. Aiming to learn separable features of each of the categories in the dataset, it could preliminarily filter models in the relevant categories for a given input by calculating classification probability. To this end, the cross-entropy loss function is used for VGNet learning, which is defined as:

$$L_s = -\frac{1}{n}\sum_{i=1}^{n}(y^{(i)} \log p^{(i)} + (1 - y^{(i)}) \log(1 - p^{(i)})), \tag{13}$$

where $p = \{p_1, p_2, ..., p_k\}$ is the output scores of all k categories in VGNet, y is the labels of the input and n is the minibatch size. Based on the above two loss functions, the final loss function is defined as:

$$L_m = \alpha L_s + \beta L_c, \tag{14}$$

where α and β are assigned weights. In this paper, both α and β are equal to 0.5. In addition, the B-rep attribute graph feature and view feature are concatenated and the final 3D CAD model descriptor $\mathcal{M}$ is obtained through the fusion network, which is defined as:

$$\mathcal{M} = \psi([\tilde{g}, v]), \tag{15}$$

where ψ is the fusion network, which is a multi-layer perceptron with two fully connected layers $\text{FC}(512, 256) \rightarrow \text{FC}(256, 256)$ in this paper.

IV. Experiments

In this section, the dataset CADNet30 used in the experiments is first introduced, then the experiments of VGNet in 3D CAD model retrieval are introduced, and the performance

Fig. 6: Some model examples of the CADNet30 dataset.

of the proposed approach in this paper is analyzed by comparing with the state-of-the-art approaches, including point-cloud-based approaches, view-based approaches. To verify the effectiveness of the proposed module in this paper, we also performed ablation experiments.

A. Experiment preparation

This section provides a detailed description of the datasets and training strategies used for the experiments.

TABLE I: Attributes of the B-rep attribute graph.

Attributes	Explanations
F_{type}	Face types in the CAD model, including Plane, Torus, Sphere, Cylinder, Cone, and B-spline.
F_{area}	Area of the face in the CAD model.
F_{normal}	Normal vector of the face centroid in the CAD model.
F_{tangent}	Tangent vector of the face centroid in the CAD model.
E_{type}	Edge types in the CAD model, including Line, Circle, and B-spline.
$E_{\text{direction}}$	The direction of the edge (if the edge is a line), otherwise the line connecting both ends of the edge is taken as the direction.
E_{length}	Length of the edge in the CAD model.

Dataset: Now there are many 3D model databases on the Internet. Some of them are based on grids and point clouds, such as the Princeton ModelNet dataset. However, such inexactly represented models are not practical for industrial manufacturing with high precision requirements. There are also databases whose representations are exact, such as SolidLetters. However these datasets are not real models used in industrial manufacturing. Based on this situation, we collected industrial models from local companies and factories and constructed a new 3D CAD model dataset, called CADNet30. These part models are used in real production, they are accurate and can be converted to STEP representations. The dataset is constructed under professional guidance and consists of 10,365 parts. It is categorized into 30 categories based on the standard mechanical parts catalog. Some of the categories include normal gear, flange, bolt, bevel gear, wire tensioner, coupling sleeve, nut, lifting hook, etc. In this paper, experiments are performed based on this dataset to confirm the effectiveness of VGNet. Some models in the dataset are shown in Fig. 6. In the B-rep attribute graph, nodes represent faces of the CAD model, and edges represent links between faces. Taking into account both performance and computational cost, we selected the face attributes and edge attributes shown in Table I as node features and edge features from the 3D CAD model's B-rep data. Furthermore, to verify the universality of VGNet, we also completed the performance evaluation on the publicly available FabWave dataset [40]. The FabWave dataset is similar to CADNet30, as both contain realistic industrial manufacturing models. The FabWave dataset consists of 4475 3D CAD models, which are divided into 45 categories.

Training strategy: The training process of the VGNet framework was performed in an end-to-end manner, where both the B-rep attribute graph branch and the multi-view branch extracted features using their respective feature extraction networks. The train-to-test ratios were set to 3, and we trained the VGNet for a maximum of 50 epochs. In addition, we employed a decayed learning rate, the ADAM optimizer, and

a loss function composed of correlation loss and cross-entropy loss. The experiments were performed with an Intel® Xeon® E5-2686 v4 @ 2.30GHz CPU and an Nvidia GeForce RTX™ 3080 Ti GPU.

B. Comparison to state-of-the-art approaches

TABLE II: Retrieval results on the CADNet30 dataset. In experiments, our proposed framework VGNet is compared with state-of-the-art models that use different representations of 3D CAD models. MVCNN (GoogLeNet) means that GoogLeNet is employed as the base architecture for weight-shared CNN in MVCNN.

Approach	Classification (Overall Accuracy)	Retrieval (mAP)
PointNet [11]	88.7%	81.6%
PointNet++ [12]	89.3%	84.0%
PointCNN [15]	90.5%	85.7%
DGCNN [14]	92.6%	87.3%
CurveNet [17]	96.8%	91.9%
MVCNN [24]	95.3%	90.6%
MVCNN (GoogLeNet)	97.2%	94.5%
GVCNN [25]	97.8%	95.3%
GCN [35]	84.3%	71.7%
GraphSAGE [36]	88.2%	81.2%
UV-Net [29]	98.2%	95.2%
CADGCN [31]	85.9%	77.1%
RECON [19]	97.8%	95.5%
VGNet	**98.8%**	**96.1%**

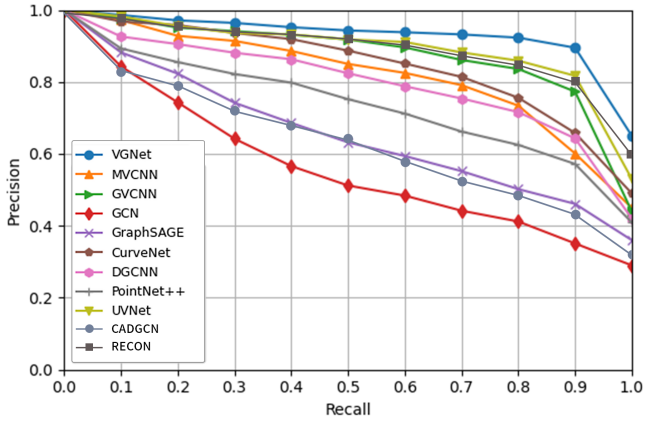


Fig. 7: The PR curves of our VGNet and other compared approaches on the CADNet30 dataset. In MVCNN model, GoogLeNet is employed as the base network.

To demonstrate the superiority of our approach, we compare our VGNet to several state-of-the-art (SOTA) approaches, including point-cloud-based 3D model approaches (PointNet [11], PointNet++ [12], PointCNN [15], DGCNN [14], CurveNet [17], view-based 3D model approaches (MVCNN [24], GVCNN [25]), graph neural network based approaches (GCN [35], GraphSAGE [36], UV-Net [29], CADGCN [31]), and the multimodal approach RECON [19] on CADNet30 dataset and FabWave dataset. For a fair comparison, we trained the other approaches again using the same training datasets.

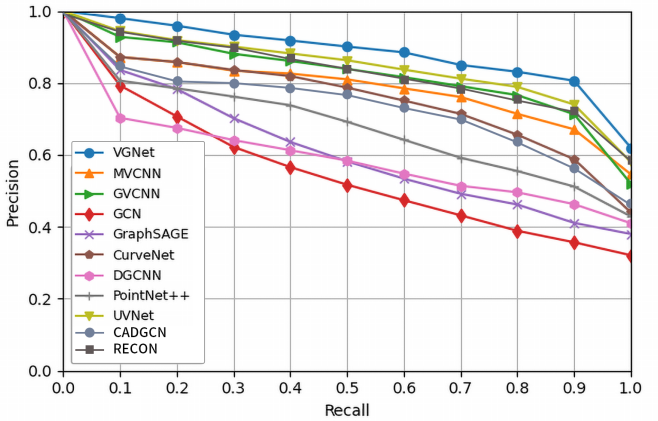


Fig. 8: The PR curves of VGNet and the other compared approaches on the FabWave dataset. In the MVCNN model, GoogLeNet is employed as the base network.

TABLE III: Retrieval results of our VGNet and the other approaches on the FabWave dataset.

Approach	Classification (Overall Accuracy)	Retrieval (mAP)
PointNet [11]	76.8%	68.4%
PointNet++ [12]	80.6%	72.4%
PointCNN [15]	82.7%	70.2%
DGCNN [14]	70.3%	58.1%
CurveNet [17]	87.2%	78.9%
MVCNN [24]	87.2%	80.3%
MVCNN (GoogLeNet)	92.1%	83.7%
GVCNN [25]	92.8%	87.6%
GCN [35]	79.3%	70.2%
GraphSAGE [36]	83.6%	76.8%
UV-Net [29]	94.4%	89.1%
CADGCN [31]	84.5%	79.4%
RECON [19]	93.8%	86.2%
VGNet	**98.0%**	**92.9%**

Quantitative Results: Table II presents the quantitative results of the experiment. Compared to other approaches, VGNet proposed in this paper achieved the best performance in terms of accuracy and mAP. The classification accuracy reached 98.8%, and the retrieval mAP reached 96.1%. Compared to the typical MVCNN in multi-view approaches, VGNet has shown significant performance improvement in both classification and retrieval tasks. Furthermore, compared with the state-of-the-art GVCNN model in view-based approaches, VGNet achieved 1.0% and 0.8% improvement in classification accuracy and retrieval accuracy, respectively. VGNet also outperformed point-cloud-based retrieval approaches, where it significantly

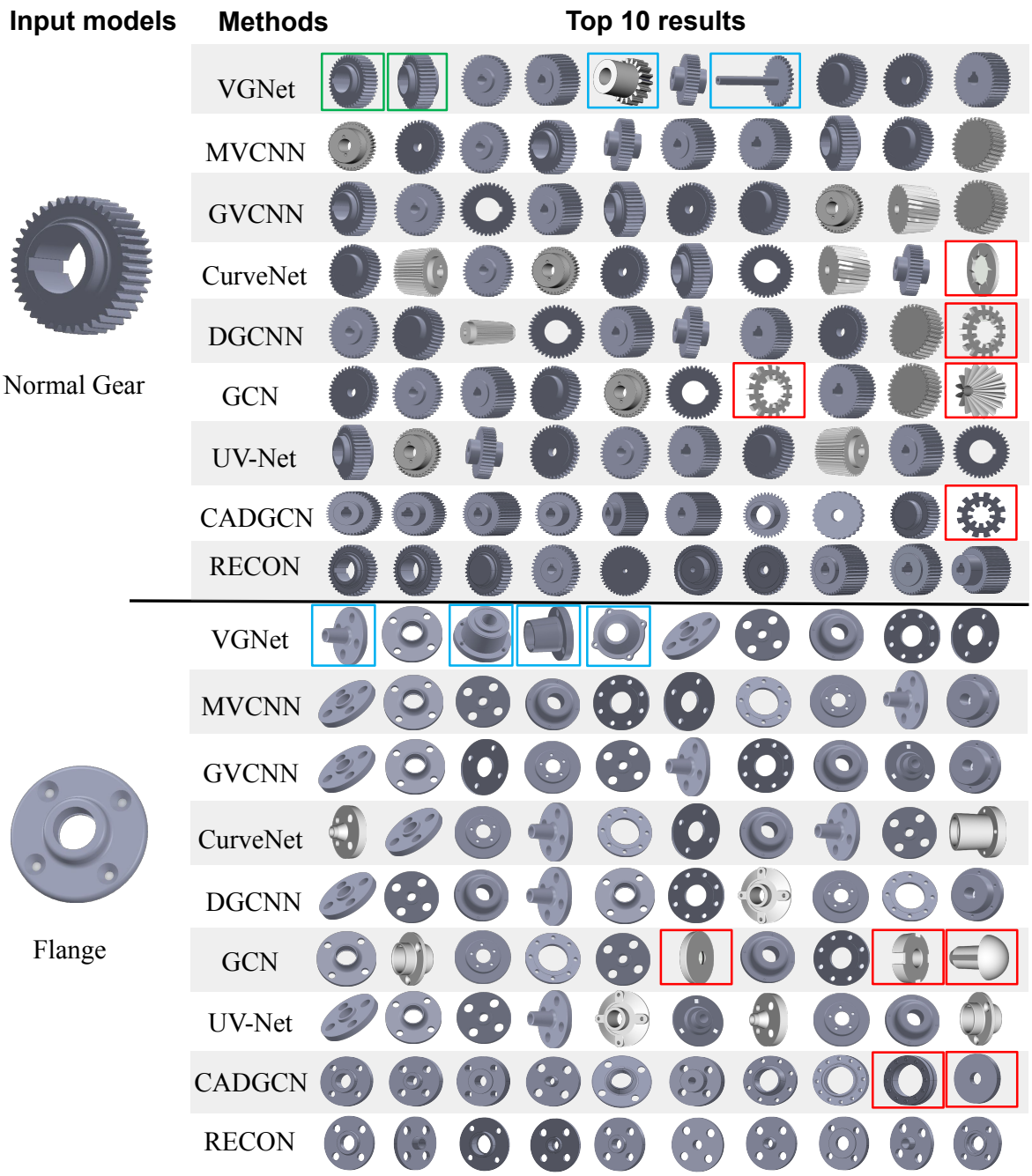


Fig. 9: When given inputs normal gear and flange, different retrieval methods yield varying results.

improved classification and retrieval accuracy compared to the typical PointNet model in point cloud. Additionally, VGNet outperformed network models based on GNN. Compared to UV-Net, VGNet improved 0.6% and 0.9% in classification and retrieval accuracy, respectively. Furthermore, VGNet outperformed the multimodal method RECON. Fig. 7 shows the precision-recall (PR) curves for all approaches listed in Table II. It can be seen that VGNet has the best retrieval performance, and compared with other approaches, the proposed approach can maintain a high level of precision when the recall rate is high. Moreover, the area enclosed by the PR curve of VGNet is the largest, indicating that VGNet has better stability.

To validate the generalization ability of VGNet proposed in this paper, we evaluated its performance on the publicly available FabWave dataset. As presented in Table III, VGNet also exhibits the highest performance. Fig. 8 shows the PR curve of each approach on FabWave. As can be seen from the figure, VGNet achieved the best balance between recall and precision, demonstrating the best performance. Compared to other approaches, VGNet showed higher precision in the high recall rate area. For example, when the recall rate is 0.9, VGNet had a precision rate of 0.81, while the precision rates of other approaches were all below 0.8. These experimental results indicate that VGNet proposed in this paper showed good retrieval performance on the FabWave dataset, which demonstrates its good generalization ability.

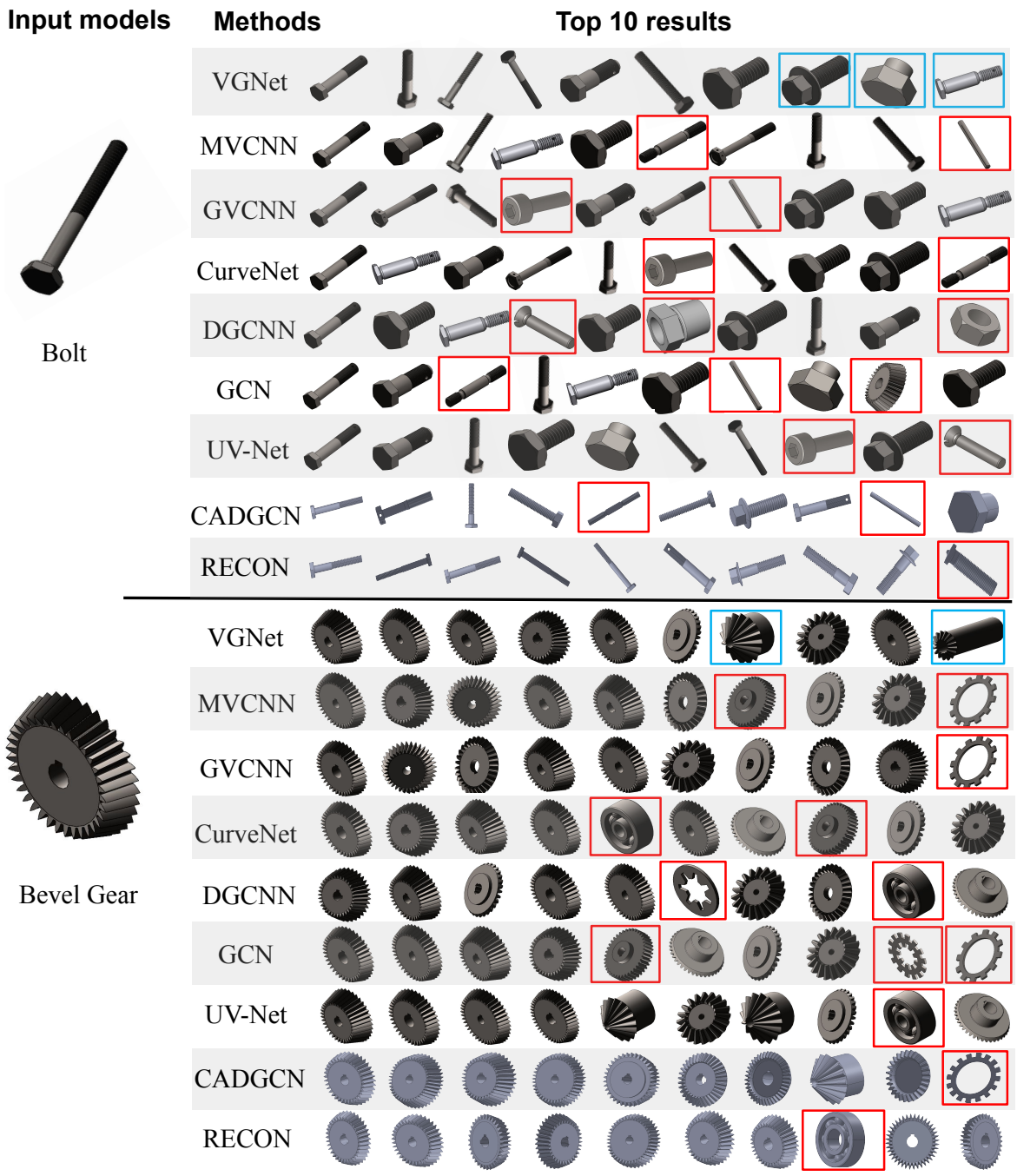


Fig. 10: When given inputs bolt and bevel gear, different retrieval methods yield varying results.

Qualitative Results: Qualitative results are shown in Fig. 9, Fig. 10, and Fig. 11. These figures demonstrate the different models returned by different retrieval methods when inputting a model. Among them, the models highlighted in green boxes indicate that VGNet has better detail capture capability. The models highlighted in blue boxes show that the models returned by VGNet exhibit diversity in geometric appearance, providing more choices for retrieval. The models highlighted in red boxes indicate retrieval errors, as they differ from the category of the input model. Specifically, in Fig. 9, when the input model is a normal gear, the models in the green boxes have rounded corners in the middle protruding section, while the models without rounded corners from other methods have higher rankings. This demonstrates that VGNet has a better capability to capture details. In Fig. 9 and Fig. 10, the models in the blue boxes exhibit significant geometric differences from the input model, but they still belong to the same category as their respective input models. This illustrates that VGNet can provide more choices in model retrieval.

In Fig. 9, Fig. 10 and Fig. 11, the models in the red boxes do not belong to the same category as their corresponding input models. It can be observed that other methods have retrieval errors. For example, in Fig. 9, when the input model is a normal gear, VGNet, RECON, UV-Net, and the view-

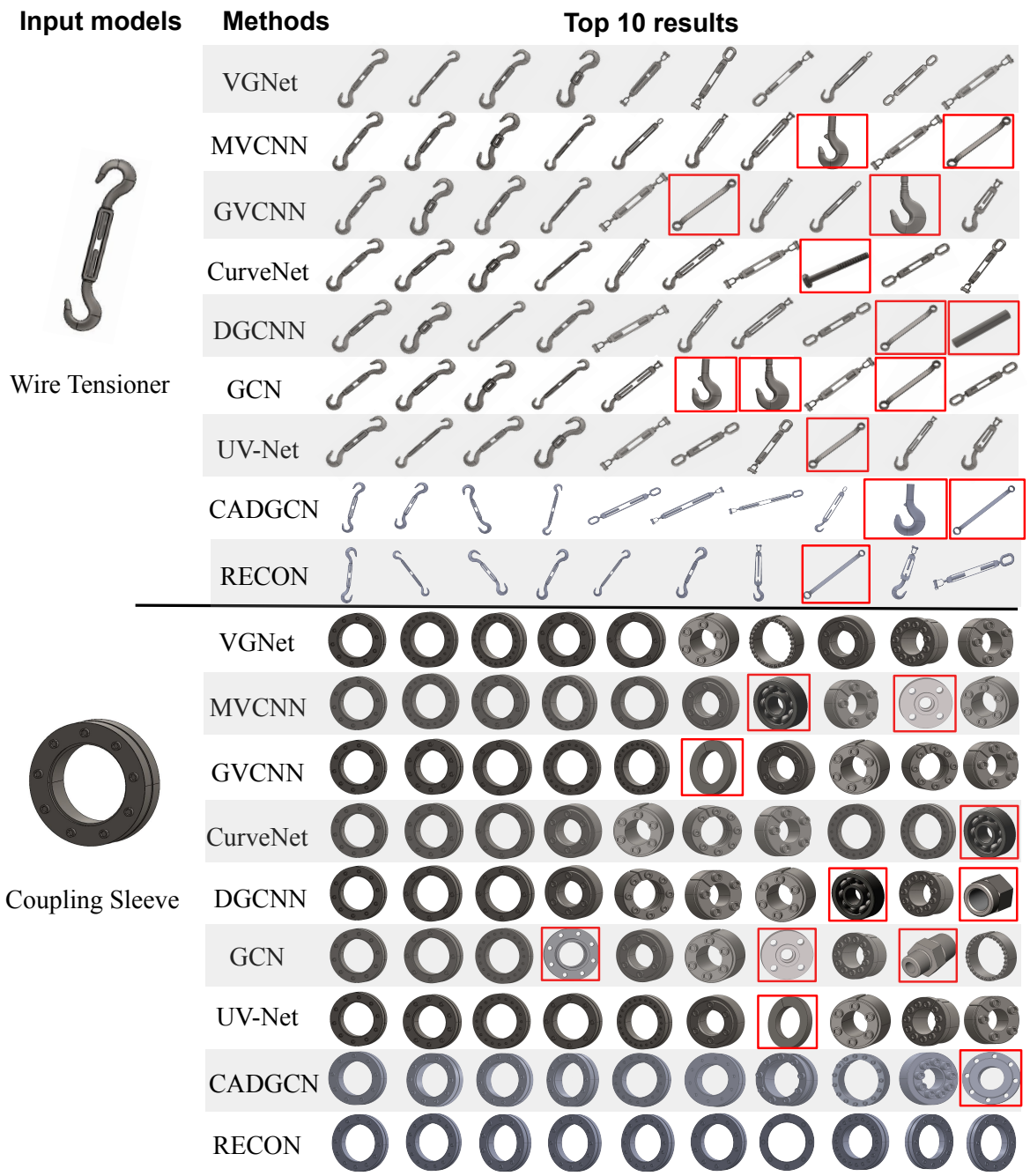


Fig. 11: When given inputs wire tensioner and coupling sleeve, different retrieval methods yield varying results.

based methods MVCNN and GVCNN all returned the correct models. However, the point-cloud-based methods CurveNet and DGCNN both returned an incorrect model, which is a lock washer. The GCN method returned two incorrect models, namely a lock washer and a bevel gear. The CADGCN method returned an incorrect model, which is a lock washer. When the input model is a flange, the GCN method returned three models, namely a normal washer, a round nut, and a grooved pin. The CADGCN method returned two incorrect models, namely a coupling sleeve and a normal washer. All the other methods retrieved the models correctly. In Fig. 10, when the input model is a bolt, MVCNN returned a grooved pin and a stud, while GVCNN returned a grooved pin and a screw. CurveNet, which exhibits better retrieval performance on point clouds, returned a stud and a screw. Similarly, UV-Net also returned two incorrect models, both of which are screws. CADGCN returned two incorrect models, namely a stud and a grooved pin. The multimodal method RECON returned a stud. In contrast, VGNet consistently returned models that belong to the bolt category. When the input model is a bevel gear, MVCNN, GVCNN and CADGCN both returned a lock washer, and MVCNN additionally returned an incorrect model, a normal gear. The point-cloud-based methods CurveNet and DGCNN each returned two incorrect models. CurveNet returned a bearing and a normal gear, while DGCNN returned a lock washer and a bearing. The GCN method returned

three incorrect models, consisting of one normal gear and two lock washers. Both UV-Net and RECON returned one incorrect model, which is a bearing. Similar results can be observed when other models are used as inputs. In Fig. 11, when the input model is a wire tensioner, MVCNN, GVCNN and CADGCN both returned a lifting hook and a wrench, CurveNet returned a bolt. UV-Net and RECON, which have better retrieval performance, also returned a wrench. When the input model is a coupling sleeve, MVCNN returned two incorrect models, namely a bearing and a flange. GVCNN returned one incorrect model, a normal washer. Both CurveNet and DGCNN returned an incorrect model, which is a bearing. In addition, DGCNN also returned an incorrect model, a normal nut. The GCN method returned three incorrect models, consisting of two flanges and one normal nut. UV-Net returned one incorrect model, a normal washer. CADGCN also returned one incorrect model, a flange. It demonstrated that VGNet exhibits superior retrieval performance.

C. Ablation experiment

In this section, we conduct ablation experiments to systematically evaluate the contributions of VGNet's multimodal feature fusion method, the improved ResBlock, the BAGPool strategy, and the multimodal learning strategy, to the overall performance of the model.

Multimodal fusion module. We analyze the effectiveness of the feature fusion method by comparing the performance of VGNet with different variants of multimodal fusion modules. This comparison enables us to validate the effectiveness of the multimodal fusion module employed in VGNet, ensuring that the integration of view and B-rep features is optimized for the task of 3D CAD model retrieval. These variants are defined as follows:

- Attribute Graph Model: We only conducted experiments using the B-rep attribute graph feature extraction network in the graph branch.
- Multi-view Model: We only conducted experiments using the view feature extraction network in the view branch.
- VGNet (None): We removed the Multi-head Attention Enhancement Module (MAEM) and the Multimodal Fusion Module (MMFM) in this variant for comparison with other variants.
- VGNet (MAEM): We used a multi-head attention mechanism and direct fusion in the fusion module instead of using the fusion approach proposed (MMFM) in this paper.
- VGNet (MMFM): We used a correlation loss function and did not use MAEM proposed in this paper.
- VGNet (Ours): We used the MAEM and MMFM modules, which constitute the final architecture of our model.

First, we conducted retrieval experiments using models based on the B-rep attribute graph and view respectively. As shown in Table IV, the accuracy and mAP of the B-rep attribute graph based model reached 90.6% and 84.2%, respectively. The view-based model achieved an accuracy of 97.2% and a mAP of 94.5%. Then, we conducted a controlled experiment on the two modules.

MAEM: The multi-head attention enhancement module is used to enhance the feature of the B-rep attribute graph, making it more discriminative. To investigate the benefits introduced by this module, we compared VGNet (MAEM) with VGNet (None). As shown in Table IV, if MAEM is removed, accuracy and mAP suffered decreases of 0.9% and 0.3% respectively. This demonstrates the importance of the multi-head attention enhancement module.

MMFM: The multimodal fusion module is used to fuse features from two different modalities. Compared to direct fusion, we use a correlation loss function in the fusion module. To investigate the benefits introduced by this module, we also compared VGNet (MMFM) with VGNet (None). As shown in Table IV, if MMFM is removed, accuracy and mAP suffered decreases of 0.6% and 0.1% respectively. This is because the correlation loss function makes different representations of the same model closer together and the cross-entropy loss function makes the representation of models of different classes farther away.

TABLE IV: Impact Analysis: Evaluating the contribution of VGNet's architectural components to retrieval performance.

Approach	MAEM	MMFM	Accuracy	mAP
Attribute Graph Model			90.6%	84.2%
Multi-view Model			97.2%	94.5%
VGNet (None)			97.6%	95.4%
VGNet (MAEM)	✓		98.5%	95.7%
VGNet (MMFM)		✓	98.2%	95.5%
VGNet (Ours)	✓	✓	**98.8%**	**96.1%**

The above experiments show that both modules can effectively improve performance, hence we also did the experiment of VGNet (Ours) containing these two modules. As shown in Table IV, VGNet (Ours) performed the best with accuracy and mAP reaching 98.8% and 96.1% respectively.

To further validate the effectiveness of the loss function designed in the Multimodal Fusion Module, we conducted ablation experiments on the design of fusion weights. As shown in Fig. 12, the results indicated that the optimal performance was achieved when both α and β were set to 0.5, confirming the utility of our loss function in effectively balancing the contributions from different modalities for enhanced 3D CAD model retrieval.

The improved ResBlock. We assess the role of the improved ResBlock in the view feature extraction by replacing it with a standard ResBlock and observing the changes in retrieval performance. This experiment helps us understand the importance of ResBlock's design tailored for 3D CAD model retrieval. The experimental results, presented in Table V, demonstrate a significant difference in performance when using the improved ResBlock compared to the standard version.

The number of views in view branch. To investigate the impact of the number of views on the retrieval performance of VGNet, we conducted an ablation study using 4, 8, 12, 16, and 20 views. The results, summarized in Table VI, demonstrate how varying the number of views affects both accuracy and mean average precision (mAP). The results indicate that using

TABLE V: Ablation study on the improved ResBlock to evaluate its contribution to retrieval performance.

Approach	Original ResBlock	Improved ResBlock	Accuracy	mAP
VGNet (Ori. ResBlock)	✓		98.2%	95.3%
VGNet (Ours)		✓	**98.8%**	**96.1%**

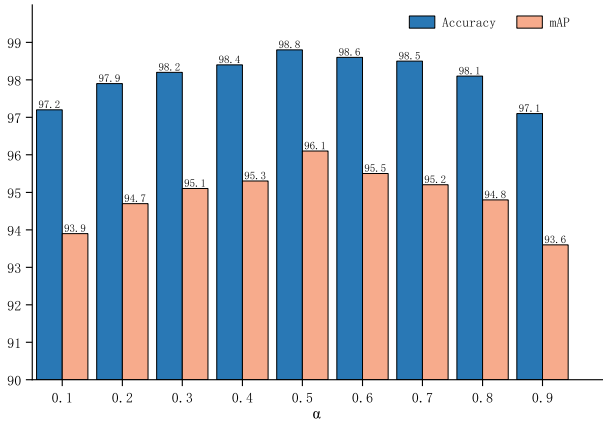


Fig. 12: Ablation study on the loss function designed in the Multimodal Fusion Module to evaluate its contribution to retrieval performance.

TABLE VI: The impact of the number of views on the retrieval performance of VGNet.

Number of Views	Accuracy	mAP
4	98.2%	95.1%
8	98.5%	95.5%
12	98.8%	96.1%
16	98.8%	96.0%
20	98.8%	96.1%

12 views achieves the mAP of 96.1% with an accuracy of 98.8%. Increasing the number of views to 16 and 20 does not lead to significant improvements in performance, and even shows a slight decrease in mAP. Therefore, considering both performance and efficiency, we decided to use 12 views in our final implementation. This choice ensures that VGNet performs effectively without unnecessary computational overhead, making it a practical solution for industrial applications.

BAGPool strategy. We evaluate the effectiveness of the BAGPool strategy by contrasting it with the EdgePool method within the graph neural network branch. This comparison can validate the BAGPool's role in aggregating and preserving the informative attributes of the B-rep graph. By removing BAGPool and substituting it with EdgePool, we can directly measure the impact on the model's performance. As shown in Table VII, the comparison demonstrates that BAGPool outperforms EdgePool in maintaining the informative attributes of the B-rep graph, resulting in improved retrieval accuracy for VGNet.

Multimodal learning strategy. We investigate the performance of various multimodal learning strategies by exploring the synergistic potential between B-rep graph data, multi-view image data, and point cloud data. The experimental approach involved sequentially replacing the graph branch and the view branch in the VGNet architecture with a point cloud branch. This allowed us to explore which pair of modalities exhibited the best complementarity. In the point cloud branch, the multi-head attention enhancement module and multimodal fusion module remained unchanged, while the feature extraction module was adapted to employ a method based on PointNet++. The experimental results shown in Table VIII indicated that the combination of point cloud and view data outperformed the point cloud and graph data, but neither surpassed the performance of the graph and view combination. This finding underscores the varying degrees of complementarity between different modalities and suggests that the graph and view combination is the most effective for our retrieval task.

TABLE VII: Ablation study on the improved BAGPool strategy to evaluate its contribution to retrieval performance.

Approach	EdgePool	BAGPool	Accuracy	mAP
VGNet (EdgePool)	✓		97.9%	94.9%
VGNet (Ours)		✓	**98.8%**	**96.1%**

D. Visualization analysis

To provide a visual understanding of how 3D CAD models are represented in the latent feature space by VGNet, we performed t-SNE visualizations of the test set (see Fig. 13). The t-SNE algorithm reduces high-dimensional features to a two-dimensional space, allowing us to observe the clustering behavior of different categories. The visualizations show that VGNet successfully maps 3D CAD models into a latent feature space where models from the same category are grouped together, and models from different categories are well-separated. This highlights the model's discriminative capability and confirms the effectiveness of our approach.

V. Conclusion

In this paper, we propose a new deep neural network architecture named VGNet for 3D CAD model retrieval. Since the B-rep representation of 3D CAD models contains rich structural information and engineering semantic information, and view representation has abundant appearance information, which describes 3D CAD models from different angles, they are complementary to each other. Hence, in VGNet, firstly the view branch is designed to extract view features from multiple projected views, and the graph branch is designed to extract graph features from the B-rep attribute graph. A multimodal

TABLE VIII: Ablation study on different multimodal learning strategies to evaluate its contribution to retrieval performance.

Approach	B-rep graph	Multi-view image	Point cloud	Accuracy	mAP
Attribute Graph Model	✓			90.6%	84.2%
Multi-view Model		✓		97.2%	94.5%
Point cloud Model			✓	89.3%	84.0%
VGNet (B-rep + Point cloud)	✓		✓	91.8%	85.2%
VGNet (Multi-view + Point cloud)		✓	✓	98.0%	92.6%
VGNet (Ours)	✓	✓		**98.8%**	**96.1%**

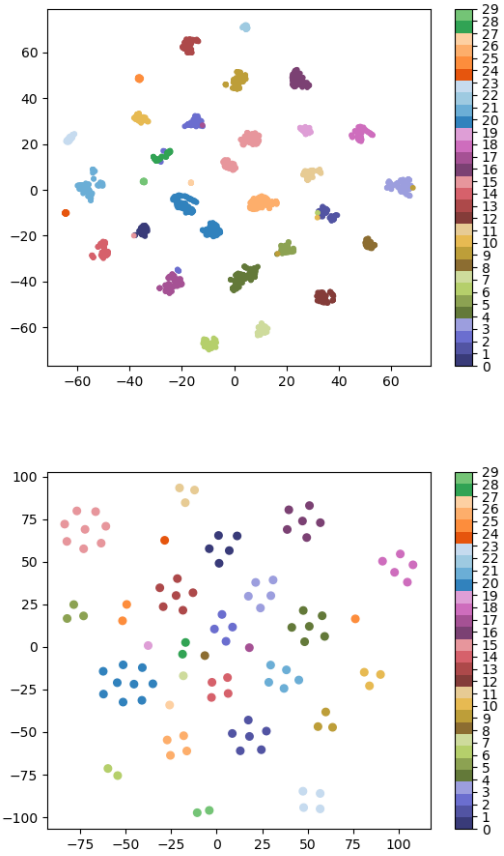


Fig. 13: T-SNE visualization of the latent feature space. (Top) Complete test set. (Bottom) 10% random sample from the test set. Different colors represent different categories.

fusion module is used to fuse these two types of features to generate the final 3D CAD model descriptor. Furthermore, inspired by MixFormer, the multi-head attention enhancement module is designed to enhance the graph feature, making it more informative. Thus, VGNet is very suitable for retrieval needs in industrial manufacturing processes. Comprehensive experiments validated the effectiveness of our approach.

Limitations & future work. Firstly, as the complexity of CAD models increases, the B-rep graphs become more intricate, potentially leading to difficulties for the graph branch in capturing all essential structural information and long-range dependencies, which could impact retrieval performance. In the future, we plan to optimize the graph neural network and conduct experiments on more complex 3D CAD models to improve the scalability and generalization ability of our approach. Secondly, the multi-view representation, while effective, may struggle with capturing fine details in highly complex models, especially if the number of available views is limited and the resolution is constrained. We are investigating techniques to enhance the view-based representation to better capture these details. Additionally, we recognize that the balance between the contributions of different modalities to the final descriptor is crucial. As CAD models grow in complexity, the relative importance of geometric, topological, and appearance features may shift, requiring a more adaptive fusion strategy. We are working on developing mechanisms to dynamically adjust the fusion weights based on the model's complexity. Also, other modal representations of 3D CAD models could be considered to enhance the representation ability of the learned descriptors. Moreover, incorporating contrastive learning into the framework to refine the embedding distribution within the latent space could be an interesting extension. It could be helpful to enhance the system's robustness, enabling it to effectively manage retrieval tasks with finer granularity, even within the context of complex CAD models. Lastly, we are exploring the integration of graph matching methods to facilitate more nuanced and detailed attribute matching and local retrieval. By incorporating these methods, we expect to significantly improve the precision and accuracy of our model in identifying and retrieving CAD models with specific attributes. This will pave the way for more sophisticated applications in the field of CAD model retrieval and analysis.

REFERENCES

[1] C. Zhang and G. Zhou, "A view-based 3D CAD model reuse framework enabling product lifecycle reuse," *Advances in Engineering Software*, vol. 127, pp. 82–89, 2019.

[2] S. Bai, X. Bai, Z. Zhou, Z. Zhang, Q. Tian, and L. J. Latecki, "GIFT: towards scalable 3D shape retrieval," *IEEE Transactions on Multimedia*, vol. 19, no. 6, pp. 1257–1271, 2017.

[3] D. Song, T.-B. Li, W.-H. Li, W.-Z. Nie, W. Liu, and A.-A. Liu, "Universal cross-domain 3D model retrieval," *IEEE Transactions on Multimedia*, vol. 23, pp. 2721–2731, 2020.

[4] H. Du, H. Shi, D. Zeng, X.-P. Zhang, and T. Mei, "The elements of end-to-end deep face recognition: A survey of recent advances," *ACM Computing Surveys*, vol. 54, no. 10s, pp. 1–42, 2022.

[5] W.-Z. Nie, M.-J. Ren, A.-A. Liu, Z. Mao, and J. Nie, "M-GCN: multi-branch graph convolution network for 2D image-based on 3D model retrieval," *IEEE Transactions on Multimedia*, vol. 23, pp. 1962–1976, 2020.

[6] Q. Liang, Q. Li, W. Nie, and A. Liu, "Unsupervised cross-media graph convolutional network for 2D image-based 3D model retrieval," *IEEE Transactions on Multimedia*, vol. 25, pp. 3443–3455, 2023.

[7] M. Groover and E. Zimmers, *CAD/CAM: Computer-aided design and manufacturing*. Pearson Education, 1983, pp. 35–83.

[8] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: a deep representation for volumetric shapes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1912–1920.

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

[10] G. Riegler, A. Osman Ulusoy, and A. Geiger, "OctNet: learning deep 3D representations at high resolutions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3577–3586.

[11] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: deep learning on point sets for 3D classification and segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[12] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: deep hierarchical feature learning on point sets in a metric space," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[13] R. Klokov and V. Lempitsky, "Escape from cells: deep Kd-networks for the recognition of 3D point cloud models," in *IEEE International Conference on Computer Vision*, 2017, pp. 863–872.

[14] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Transactions on Graphics*, vol. 38, no. 5, pp. 1–12, 2019.

[15] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: convolution on x-transformed points," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[16] Y. Song, F. He, L. Fan, J. Dai, and Q. Guo, "DSACNN: dynamically local self-attention CNN for 3D point cloud analysis," *Advanced Engineering Informatics*, vol. 54, p. 101803, 2022.

[17] T. Xiang, C. Zhang, Y. Song, J. Yu, and W. Cai, "Walk in the cloud: learning curves for point clouds shape analysis," in *IEEE International Conference on Computer Vision*, 2021, pp. 915–924.

[18] R. Wu, J. Bai, W. Li, and J. Jiang, "DCNet: exploring fine-grained vision classification for 3D point clouds," *The Visual Computer*, vol. 40, no. 2, pp. 781–797, 2024.

[19] Z. Qi, R. Dong, G. Fan, Z. Ge, X. Zhang, K. Ma, and L. Yi, "Contrast with reconstruct: Contrastive 3D representation learning guided by generative pretraining," in *International Conference on Machine Learning*. PMLR, 2023, pp. 28 223–28 243.

[20] H. Shao, J. Bai, R. Wu, J. Jiang, and H. Liang, "FGPNet: a weakly supervised fine-grained 3D point clouds classification network," *Pattern Recognition*, vol. 139, p. 109509, 2023.

[21] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung, "On visual similarity based 3D model retrieval," *Computer Graphics Forum*, vol. 22, no. 3, pp. 223–232, 2003.

[22] Y. Gao, J. Tang, R. Hong, S. Yan, Q. Dai, N. Zhang, and T.-S. Chua, "Camera constraint-free view-based 3D object retrieval," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2269–2281, 2011.

[23] D. Zeng, S. Chen, B. Chen, and S. Li, "Improving remote sensing scene classification by integrating global-context and local-object features," *Remote Sensing*, vol. 10, no. 5, p. 734, 2018.

[24] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *IEEE International Conference on Computer Vision*, 2015, pp. 945–953.

[25] Y. Feng, Z. Zhang, X. Zhao, R. Ji, Y. Gao, and Gvcnn, "Group-view convolutional neural networks for 3D shape recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 264–272.

[26] S. Ge, C. Li, S. Zhao, and D. Zeng, "Occluded face recognition in the wild by identity-diversity inpainting," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 10, pp. 3387–3397, 2020.

[27] M. El-Mehalawi and R. A. Miller, "A database system of mechanical components based on geometric and topological similarity. Part I: representation," *Computer-Aided Design*, vol. 35, no. 1, pp. 83–94, 2003.

[28] El-Mehalawi, Mohamed and Miller, R Allen, "A database system of mechanical components based on geometric and topological similarity. part ii: indexing, retrieval, matching, and similarity assessment," *Computer-Aided Design*, vol. 35, no. 1, pp. 95–105, 2003.

[29] P. K. Jayaraman, A. Sanghi, J. G. Lambourne, K. D. Willis, T. Davies, H. Shayani, and N. Morris, "UV-Net: learning from boundary representations," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 703–11 712.

[30] A. R. Colligan, T. T. Robinson, D. C. Nolan, Y. Hua, and W. Cao, "Hierarchical CADNet: learning from B-reps for machining feature recognition," *Computer-Aided Design*, vol. 147, p. 103226, 2022.

[31] L. Mandelli and S. Berretti, "CAD 3D model classification by graph neural networks: A new approach based on STEP format," *arXiv preprint arXiv:2210.16815*, pp. 1–11, 2022.

[32] J. Bai, S. Gao, W. Tang, Y. Liu, and S. Guo, "Design reuse oriented partial retrieval of CAD models," *Computer-Aided Design*, vol. 42, no. 12, pp. 1069–1084, 2010.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[34] International Organization for Standardization, "Industrial automation systems and integration – product data representation and exchange – part 111: Integrated application resource: Elements for the procedural modeling of solid shapes," ISO, Standard 10303-111, 2007.

[35] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, pp. 1–14, 2016.

[36] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in Neural Information Processing Systems*, vol. 30, p. 1025–1035, 2017.

[37] F. Diehl, "Edge contraction pooling for graph neural networks," *arXiv preprint arXiv:1905.10990*, pp. 1–9, 2019.

[38] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *International Conference on Machine Learning*, 2018, pp. 5453–5462.

[39] Y. Song, F. He, Y. Duan, T. Si, and J. Bai, "LSLPCT: an enhanced local semantic learning transformer for 3-D point cloud analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.

[40] A. Angrish, B. Craver, and B. Starly, "FabSearch: A 3D CAD model-based search engine for sourcing manufacturing services," *Journal of Computing and Information Science in Engineering*, vol. 19, no. 4, p. 041006, 2019.