VSFormer: Mining Correlations in Flexible View Set for Multi-view 3D Shape Understanding

Hongyu Sun, Yongcai Wang, *Member, IEEE,* Peng Wang, Haoran Deng, Xudong Cai, Deying Li

Abstract—View-based methods have demonstrated promising performances in 3D shape understanding. However, they tend to make strong assumptions about the relations between views or learn the multi-view correlations indirectly, which limits the flexibility of exploring inter-view correlations and the effectiveness of target tasks. To overcome the above problems, this paper investigates flexible organization and explicit correlation learning for multiple views. In particular, we propose to incorporate different views of a 3D shape into a permutation-invariant set, referred to as *View Set*, which removes rigid relation assumptions and facilitates adequate information exchange and fusion among views. Based on that, we devise a nimble Transformer model, named *VSFormer*, to explicitly capture pairwise and higher-order correlations of all elements in the set. Meanwhile, we theoretically reveal a natural correspondence between the Cartesian product of a view set and the correlation matrix in the attention mechanism, which supports our model design. Comprehensive experiments suggest that VSFormer has better flexibility, efficient inference efficiency and superior performance. Notably, VSFormer reaches state-of-the-art results on various 3d recognition datasets, including ModelNet40, ScanObjectNN and RGBD. It also establishes new records on the SHREC'17 retrieval benchmark. The code and datasets are available at https://github.com/auniquesun/VSFormer.

Index Terms—Multi-view 3D Shape Recognition and Retrieval, Multi-view 3D Shape Analysis, View Set, Attention Mechanism

I. INTRODUCTION

WITH the advancement of 3D perception devices (Li-DAR, RGBD camera, etc.), 3D assets like point clouds, volumetric grids, polygon meshes, RGBD images become more and more common in daily life and industrial production [1]–[5]. 3D object recognition and retrieval are basic requirements for understanding the 3D contents and the development of these technologies will benefit downstream applications like VR/AR/MR, 3D printing, autopilot, etc.

Existing methods for 3D shape analysis can be roughly divided into three categories according to the input representation: (1) point-based [6]–[18], (2) voxel-based [19]–[23], and (3) view-based methods [24]–[44]. Among them, view-based methods recognize a 3D object according to its rendered or projected images, termed *multiple views*. Generally, methods in this line [25], [39]–[46] outperform the point- and voxel-based counterparts [13], [14], [16], [17], [21]. On one hand, view-based methods benefit from massive image datasets and the advances in image recognition over the past decade. On the other hand, the multiple views of a 3D shape contain

All authors are with the Department of Computer Science, School of Information, Renmin University of China, Beijing 100872, China. Corresponding author: Yongcai Wang. {sunhongyu, ycw, peng.wang, xudongcai, denghaoran, deyingli}@ruc.edu.cn
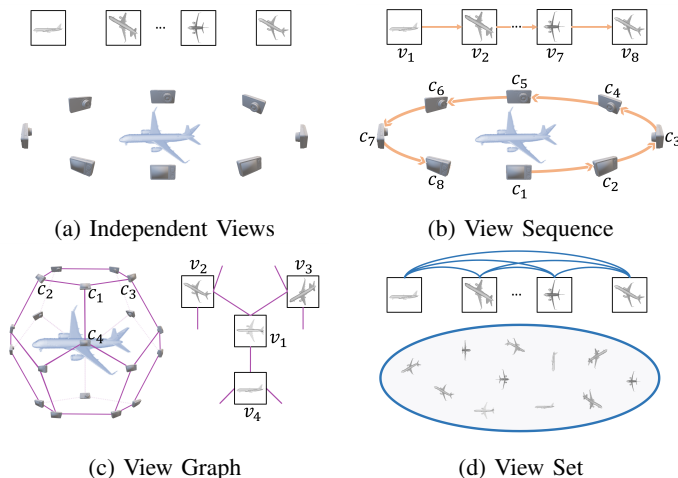

(a) Independent Views

(b) View Sequence

(c) View Graph

(d) View Set

Fig. 1: **A division for multi-view 3D shape analysis methods.** The division is based on how they organize views and aggregate multi-view information. View Set is adopted by VSFormer that the views of a 3D shape are organized in a set.

richer visual and semantic signals than the point or voxel form. For example, one may not be able to decide whether two 3D shapes belong to the same category by observing them from one view, but the answer becomes clear after watching other views of these shapes. The example inspires a central problem, how to effectively exploit multi-view information for a better understanding of 3D shape.

This paper systematically investigates existing methods on how they aggregate the multi-view information and the findings are summarized in Figure 1. In the early stage, MVCNN [24] and its follow-up works [25]–[27], [29], [32], [33], [47], [48] independently process different views of a 3D shape by a shared CNN. The extracted features are fused with pooling operation or some variants to form a compact 3D shape descriptor. We group these methods into *Independent Views*, shown in Figure 1a. Although the simple design made them stand out at the time, the interaction among different views was insufficient. In the second category, a growing number of methods model multiple views as a sequence [28], [34]–[37] to increase information exchange, which are grouped into *View Sequence* in Figure 1b. They deploy RNNs, like GRU [49] and LSTM [50], to learn the view relations. However, a strong assumption behind *View Sequence* is that the views are collected from a circle around the 3D shape. In many cases, the assumption may be invalid since the

views can be rendered from random viewpoints, so they are unordered. To alleviate this limitation, later methods describe views with a more flexible structure, graph [39], [41], [44] or hyper-graph [30], [38], [42], and develop graph convolution networks (GCNs) to propagate features among views, called *View Graph* in Figure 1c. Methods in this category show both flexibility and promising gains, whereas they require to construct a view graph for each 3D shape according to the positions of camera viewpoints, which introduces additional computation overheads. Meanwhile, the viewpoints may be unknown and the message propagation on the graph may not be straightforward for distant views. Some other methods also explore rotations [31], [51], region-to-region relations [52], multi-layered height-maps [53], view correspondences [46], viewpoints selection [40], voint cloud representations [43] when recognizing 3D shapes. They can hardly be divided into the above categories, but multi-view correlations in these methods still need to be enriched.

By revisiting existing works, two aspects are identified critical for improving multi-view 3D shape analysis but are not explicitly pointed out in previous literature. The first is how to organize the views so they can communicate flexibly and freely. The second is how to model multi-view correlations directly and explicitly. It is worth noting that the second ingredient is usually coupled with the first, just like GCNs designed for view graphs and RNNs customized for the view sequences.

In this paper, we propose to organize the multiple views of a 3D shape into a more flexible structure, *e.g.*, *View Set*, shown in Figure 1d, where elements are permutation invariant. This is consistent with the fact that 3D shape understanding is actually not dependent on the order of input views. For example, in Figure 1b, whether the side view is placed first, middle or last in the inputs, the recognition result produced by the model should always be `airplane`. Unlike existing methods analyzed above, this perspective removes inappropriate assumptions and restrictions about the relations between the views, thus is more practical and reasonable in real-world applications.

More importantly, a *ViewSet Transformer* (**VSFormer**) is devised to release the power of multiple views and adaptively learn the pairwise and higher-order relations among the views and integrate multi-view information. The attention architecture is a natural choice because it aligns with the view set's characteristics. First, we theoretically reveal that the Cartesian product of a view set can be formulated by the correlation matrix, which can be decomposed into attention operations mathematically. Second, the attention mechanism is essentially a set operator and inherently good at capturing correlations between the elements in a set. Third, this mechanism is flexible enough that it makes minimal assumptions about the inputs, which matches our expectation that there are no predefined relations or restrictions for views. Overall, the proposed approach presents a one-stop solution that directly captures the correlations of all view pairs in the set, which promotes the flexible and free exchange of multi-view information.

Several critical designs are presented in VSFormer. (1) The position encodings of input views are removed since views are permutation invariant. (2) The class token is removed because it is irrelevant to capturing the correlations of view pairs in the set. (3) The number of attention blocks is greatly reduced as the size of a view set is relatively small (≤ 20 in most cases).

The details of the proposed approach will be elaborated in Section III. Systematic experiments suggest that VSFormer around the flexible set and explicit relation grasping unleashes astonishing capabilities and obtains new records in downstream tasks. In short, the contributions of this paper include:

- We identify two key aspects of multi-view 3D shape understanding, organizing views reasonably and modeling their relations explicitly, albeit they are critical for performance improvement but absent in previous literature.
- We propose a Transformer-based model, named VSFormer, to capture the correlations of all view pairs directly for better multi-view information exchange and fusion. At the same time, a theoretical analysis is accompanied to support such a design.
- Extensive experiments demonstrate the superb performances of the proposed approach and the ablation studies shed light on the various sources of performance gains. For the recognition task, VSFormer reaches 98.4%(+4.1%), 95.9%(+1.9%), 98.8%(+1.1%) overall accuracy on RGBD, ScanObjectNN, ModelNet40, respectively. The results surpass all existing methods and achieve new state of the arts. For 3D shape retrieval, VSFormer also sets new records in multiple dimensions on the SHREC'17 benchmark.

II. RELATED WORK

In this section, we review the multi-view 3D shape analysis methods, explore the deployment of set and attention in these methods, and discuss the latest progress in the field.

A. Multi-view 3D Shape Analysis

Existing methods aggregate multi-view information for 3D shape understanding in different ways.

1) Independent Views: Early work like MVCNN series [24], [25] and its follow-up works [26], [27], [29], [32], [33], [47], [48] extract view features independently using a shared CNN, then fuse the extracted features using the pooling operation or some variants. The simple strategy may discard a lot of useful information and the views are not well treated as a whole thus information flow among views needs to be increased.

2) View Sequence: Researchers perceive the problems and propose various descriptions to incorporate multiple views of a 3D shape into specific data structures. For example, RNN-based [28], [34]–[37] and ViT-based [45], [54] methods are proposed to operate on the view sequence.

3) View Graph: The graph-based models [30], [38], [39], [41], [42], [44] assume the relations among views as graphs and develop GCNs to capture multi-view interaction. However, message propagation between distant nodes on a view graph may not be straightforward and graph construction leads to additional computation overheads.
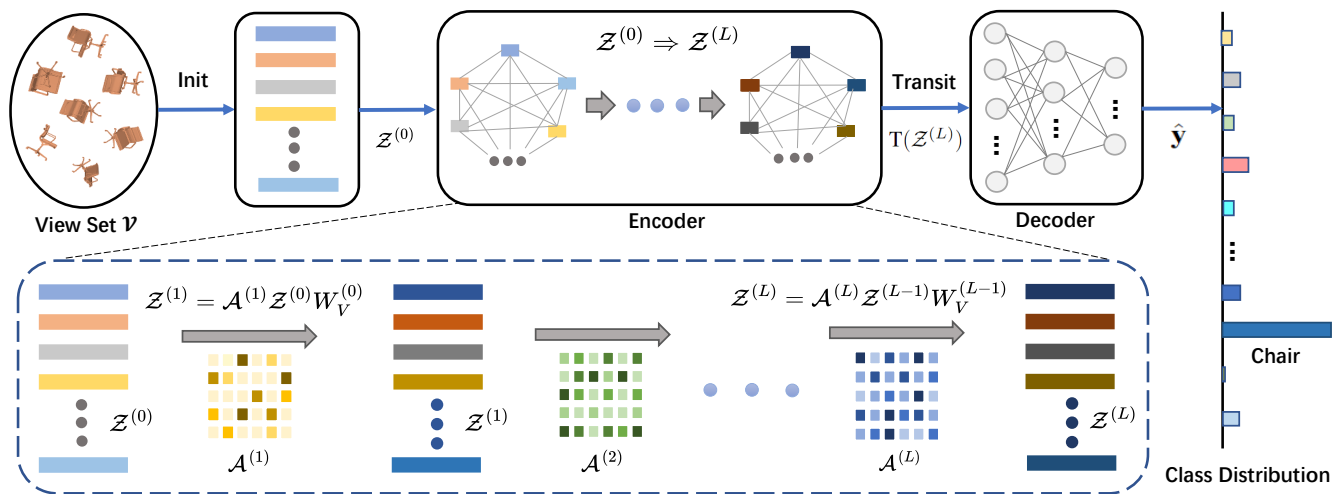
Fig. 2: **The overall architecture of VSFormer.** It consists of 4 modules: Initializer (Init), Encoder, Transition (Transit) and Decoder. Encoder is responsible for grasping pairwise and higher-order correlations of views in a set.

4) View Set: This paper presents a more flexible and practical structure, *View Set*, which neither makes assumptions about views nor introduces additional overheads. Based on that, a view set attention model is devised to adaptively grasp the correlations for all view pairs.

Some other methods also explore rotations [31], [51], region-to-region relations [52], multi-layered height-maps representations [53], view correspondences [46], viewpoints selection [40], voint cloud representations [43] when analyzing 3D shapes. Their multi-view interaction still needs to be strengthened.

B. Set in Multi-view 3D Shape Analysis

Previous works also mention "set" in multi-view 3D shape analysis. But they basically refer to different concepts from the proposed one. For instance, RCPCNN [29] introduces a dominant set clustering and pooling module to improve MVCNN [24]. Johns *et al.* [55] decompose a sequence of views into a set of view pairs. They classify each pair independently and weigh the contribution of each pair. MHBN [47] considers patches-to-patches (set-to-set) similarity of different views and aggregates local features using bilinear pooling. Yu *et al.* extend MHBN by introducing VLAD layer [48], where the similarity between two sets of local patches is calculated by exploiting bilinear and VLAD pooling operations, while our view set perspective provides a foundation for learning the correlations of all view pairs adaptively.

C. Attention in Multi-view 3D Shape Analysis

The attention mechanisms have been embedded in existing multi-view 3D shape analysis methods but vary in motivation, practice and effectiveness. VERAM [36] uses a recurrent attention model to select a sequence of views to classify 3D shapes. SeqViews2SeqLabels [34] introduces the attention mechanism to increase the discriminative ability for the RNN-based model and reduces the effect of selecting the first view

position. 3D2SeqViews [35] proposes hierarchical attention to incorporate view-level and class-level importance for 3D shape analysis. Nevertheless, there are three points worth noting for the attention in the above methods. Firstly, the attention modules in these methods have nothing to do with *view set perspective* and are not designed for handling an unordered view set. Secondly, these modules differ from the multi-head self-attention in standard Transformer [56]. Thirdly, previous methods equipped with attention modules do not seem to produce satisfactory performances.

Another work MVT [45] also explores the attention architecture for view-based 3D recognition. However, MVT is inspired by the success of ViT [57] and simply applies ViT to the views without modification. The position encodings in MVT are preserved. Thus the method is also irrelevant to the idea of organizing views in an unordered set. Besides, MVT deploys a ViT to extract patch-level features and adopts another ViT to learn the correlations of all patches in different views. In contrast, VSFormer shows it is unnecessary to take the patch-level interactions into account to achieve better results, thus the computation budgets are significantly reduced. Recent work MRVA-Net [54] investigates multi-range view aggregation with ViT-based feature fusion for 3D shape retrieval. This method belongs to the category of *View Sequence* as it assumes the views are rendered along a circle. The short, mid and long ranges are defined by human priors and the model may be sensitive to the range definition. Then dilated convolutions are conducted on the CNN-initialized view features to attain multi-range features. After that, MRVA-Net applies ViT without adaptation to fuse the multi-range features. Instead, our approach is more flexible and efficient by directly processing an unordered set of views in parallel, without the sequence assumption, multiple convolution operations, and hard-encoded ranges for views.

D. Latest Progress in Multi-view 3D Shape Analysis

Here we discuss several very recent works and highlight the differences with our method. Hamdi *et al.* propose a novel Voint Cloud representation and develop VointNet [43] to conduct multiple 3D tasks. This model is built on conventional 2D backbones and combines multi-view information with 3D point clouds while our model is devised upon standard attention mechanism and only exploits views as inputs. HGNN$^+$ [42] proposes hyperedge groups and hypergraph convolution to explore multi-modal data correlation. In our cases, we only have views and do not emphasize multi-modal data correlation. View-GCN++ [41] wants to deal with rotation sensitivity and upgrades the prior version by developing local attentional graph convolution and rotation robust view sampling. Instead, our method directly processes aligned or rotated views by exploring the correlations of all view pairs in parallel, which demonstrates superior performances. MVPNet [44] improves View-GCN [39] by generating an ordered path on the view graph and aggregating the ordered features along the path with ViT [57]. It falls into the View Graph category and may suffer from the shortcomings as we analyzed above, *e.g.*, indirect relation modeling and additional overheads introduced by graph construction and path generation for each 3D object.

III. METHODOLOGY

In this section, we firstly formulate the problem of multi-view 3D shape analysis based on the view set, then elaborate on the devised VSFormer and how it handles a set of views.

A. Problem Formulation

1) View Set: The views of a 3D shape refer to its rendered or projected RGB images. For example, a 3D shape $\mathcal{S}$ corresponds to views $v_1, v_2, \ldots, v_M \in \mathbb{R}^{H \times W \times C}$, where M is the number of views and $H \times W \times C$ indicates the image size. In *our perspective*, the views of $\mathcal{S}$ form a set $\mathcal{V} = \{v_1, v_2, \ldots, v_M\}$, where elements are permutation invariant. Thus, $\mathcal{V}_\pi = \{v_{\pi_{(1)}}, v_{\pi_{(2)}}, \ldots, v_{\pi_{(M)}}\}$ is always equivalent to $\mathcal{V}$ when $\pi_{(1)}, \pi_{(2)}, \ldots, \pi_{(M)}$ is a random permutation of 1, 2, . . . , M.

In fact, organizing different views in order (view sequence) is a special case of random permutation. Random permutation does not introduce additional overheads compared to view sequence, and considerably saves computation budgets compared to graph construction in view graph methods.

2) 3D Shape Recognition & Retrieval: In many cases, 3D shape retrieval can be regarded as a classification problem [58]. It aims to find the most relevant shapes to the query. Meanwhile, the relevance is defined according to the query's ground truth class and subclass, which means if a retrieved shape has the same class and subclass as the query, they match perfectly. Therefore, the tasks of 3D shape retrieval and recognition can be unified by predicting a category distribution $\hat{\mathbf{y}} \in \mathbb{R}^K$ of the target shape $\mathcal{S}$, where K is the number of 3D shape categories. In this paper, we design a view set attention model $\mathcal{F}$ to predict the distribution. The input of $\mathcal{F}$ is a view set $\mathcal{V} \in \mathbb{R}^{M \times H \times W \times C}$ of the shape $\mathcal{S}$, and the output is a class distribution $\hat{\mathbf{y}} = \mathcal{F}(\mathcal{V})$.

B. View Set Attention Model

The proposed model aims to facilitate flexible communication and adequate information fusion among views in a set. The overall architecture of VSFormer is presented in Figure 2.

The input of VSFormer is a view set $\mathcal{V} = \{v_1, \ldots, v_M\}$. First, we initialize $\mathcal{V}$ with lightweight modules (*e.g.*, ResNet18 [59], AlexNet [60]) to map the views to hidden representations $\mathcal{Z}^{(0)} = \{\mathbf{z}_1^{(0)}, \ldots, \mathbf{z}_M^{(0)}\} \in \mathbb{R}^{M \times D}$, where $\mathbf{z}_i^{(0)}$ is a D-dimensional encoding of the view v_i. So $\mathcal{Z}^{(0)}$ contains information from M independent views, without any clue of their correlations. Second, to enrich the interaction and fusion of multi-view information, the designed model computes the correlations of all view pairs directly through iterative attention.

We show that (1) in theory, there is a natural correspondence between the view pairs in a set and the correlation matrix in a standard attention model. (2) in practice, the proposed model can be constructed according to the view set and attention theory.

1) View Set and Attention Theory: The first problem is how to represent view pairs in a set. For the initialized $\mathcal{Z}^{(0)}$, all view pairs can be formulated by its *Cartesian product* $\mathcal{P}^{(0)} = \mathcal{Z}^{(0)} \times \mathcal{Z}^{(0)} = \{ (\mathbf{z}_i^{(0)}, \mathbf{z}_j^{(0)}) \mid i, j \in 1, \ldots, M \}$. Let us denote $p_{i,j}^{(0)} = (\mathbf{z}_i^{(0)}, \mathbf{z}_j^{(0)})$, so all view pairs in $\mathcal{Z}^{(0)}$ can be expressed with $\mathcal{P}^{(0)} = \{ p_{i,j}^{(0)} \mid i, j \in 1, \ldots, M \}$.

Theorem 1 (Correspondence between the Cartesian product of a view set and the correlation matrix in the attention mechanism): The Cartesian product $\mathcal{P}$ of a view set $\mathcal{V}$ can be formulated by a correlation matrix $\mathcal{A}$ and computed by the attention mechanism.

Proof is provided in the subsection A of the Supplementary Material. We further elaborate on the theorem from the following three aspects.

(i) Pairwise Correlations in a View Set. Generally, the attention model characterizes the pairwise correlations of different elements by a correlation matrix. The model receives an input $\mathcal{I}$ that consist of N elements, $\mathbf{e}_1, \ldots, \mathbf{e}_N$, where $\mathbf{e}_i$ is a E-dimensional vector. $\mathcal{I}$ is regarded as a set $\{\mathbf{e}_1, \ldots, \mathbf{e}_N\}$ as the model is unaware of the order of the elements. Thereby, the pairwise correlations for $\mathcal{I}$ learned by the attention mechanism can be represented by a correlation matrix $\mathcal{A}^{(1)} = \{ a_{i,j}^{(1)} \mid i, j \in 1, \ldots, N \}$, shown in Eq. 1, where $a_{i,j}^{(1)}$ is the attention score that $\mathbf{e}_i$ receives from $\mathbf{e}_j$.

$$\mathcal{A}^{(1)} = \begin{bmatrix} a_{1,1}^{(1)} & a_{1,2}^{(1)} & \ldots & a_{1,N}^{(1)} \\ a_{2,1}^{(1)} & a_{2,2}^{(1)} & \ldots & a_{2,N}^{(1)} \\ \ldots & \ldots & \ldots & \ldots \\ a_{N,1}^{(1)} & a_{N,2}^{(1)} & \ldots & a_{N,N}^{(1)} \end{bmatrix} \tag{1}$$

We define $\mathcal{A}^{(1)}$ as the *first-order correlation* matrix. It is noticed that $\mathcal{A}^{(1)}$ has same form as the *Cartesian product* $\mathcal{P}^{(0)}$. Let $\mathbf{a}_i^{(1)} = (a_{i,1}^{(1)}, \ldots, a_{i,N}^{(1)})$ and $i \in 1, \ldots, N$, $\mathbf{a}_i^{(1)}$ represents the correlations that the ith element receives from all elements in $\mathcal{I}$. Hence, $\mathcal{A}^{(1)}$ can be further converted into another form in Eq. 2 and $\mathbf{a}_i$ can be calculated with Eq. 3.

$$\mathcal{A}^{(1)} = \begin{bmatrix} \mathbf{a}_1^{(1)} & \mathbf{a}_2^{(1)} & \ldots & \mathbf{a}_N^{(1)} \end{bmatrix}^{\mathrm{T}} \tag{2}$$

$$\mathbf{a}_i^{(1)} = \text{Norm}(Q_i^{(0)} K^{(0)\text{T}} / \tau)$$
$$Q_i^{(0)} = \mathbf{e}_i W_Q^{(0)} \qquad K^{(0)} = \mathcal{I} W_K^{(0)} \qquad (3)$$

Here τ is a temperature coefficient to adapt the product and Norm is a normalized function to ensure the attention scores are in the range of [0,1]. Both $W_Q^{(0)}$ and $W_K^{(0)} \in \mathbb{R}^{E \times E}$ are learnable embeddings to project the input $\mathcal{I}$.

Due to $\mathcal{P}^{(0)}$ and $\mathcal{A}^{(1)}$ having the same mathematical form, it is easy to transfer the above process to capture the pairwise correlations of all views in a set. The only thing we need to do is make $N = M$, $E = D$ and $\mathcal{I} = \mathcal{Z}^{(0)}$.

(ii) Injecting the Correlations into Views. Once the *first-order correlations* $\mathcal{A}^{(1)}$ are obtained, we can inject this kind of knowledge into the initialized $\mathcal{Z}^{(0)}$ using Eq. 4, to enable information flow between views. The idea is to update each element of $\mathcal{Z}^{(0)}$ according to its correlations with other elements, resulting in $\mathcal{Z}^{(1)} = \{\mathbf{z}_1^{(1)}, \ldots, \mathbf{z}_M^{(1)}\} \in \mathbb{R}^{M \times D}$.

$$\mathcal{Z}^{(1)} = \mathcal{A}^{(1)} \mathcal{Z}^{(0)} W_V^{(0)} \qquad (4)$$

Here $W_V^{(0)} \in \mathbb{R}^{D \times D}$ is a learnable embedding that is to map the initialized $\mathcal{Z}^{(0)}$. The new representations $\mathcal{Z}^{(1)}$ achieve a basic understanding of the relations among views thus take the first step toward multi-view information fusion. We call $\mathcal{Z}^{(1)}$ the *first-order representations* of the view set $\mathcal{V}$.

However, the first-order representations may not be sufficient to grasp various view relations in complex scenarios. It is expected that the *higher-order* interactions among views are also adaptively explored.

(iii) Higher-order Correlations in a View Set. The attention model can go beyond capturing pairwise correlations for elements in a set. Assuming there are a total of L attention blocks in the model, since the correlation matrix $\mathcal{A}^{(\ell)}$ in ℓth block is always constructed based on $\mathcal{A}^{(\ell-1)}$, the higher-order interaction can be learned by deepening the attention blocks. We derive the ℓth-order multi-view representations $\mathcal{Z}^{(\ell)}$ using Eq. 5, where $W_Q^{(\ell-1)}, W_K^{(\ell-1)}$ and $W_V^{(\ell-1)}$ are learnable embeddings of shape $\mathbb{R}^{D \times D}$ to transform $\mathcal{Z}^{(\ell-1)}$, $\ell \in 1, \ldots, L$.

$$\mathcal{Z}^{(\ell)} = \mathcal{A}^{(\ell)} \mathcal{Z}^{(\ell-1)} W_V^{(\ell-1)}$$
$$\mathcal{A}^{(\ell)} = \text{Norm}(Q^{(\ell-1)} K^{(\ell-1)\text{T}} / \tau)$$
$$Q^{(\ell-1)} = \mathcal{Z}^{(\ell-1)} W_Q^{(\ell-1)} \qquad (5)$$
$$K^{(\ell-1)} = \mathcal{Z}^{(\ell-1)} W_K^{(\ell-1)}$$

By going through L attention blocks in the view set encoder, the representations $\mathcal{Z}^{(\ell)}$ iteratively compute the correlation matrix and update themselves with the latest knowledge to obtain higher-order understanding of the correlations among views.

2) Constructing VSFormer: According to the above analysis, VSFormer can be built with the following modules: Initializer, Encoder, Transition and Decoder, shown in Figure 2.

Lightweight neural networks can serve as the initializer (*e.g.*, AlexNet [60]). The encoder receives the initialized view set $\mathcal{Z}^{(0)} \in \mathbb{R}^{M \times D}$ and processes it with L attention blocks. Each attention block stacks the multi-head self-attention [56] (MSA) and MLP layers with residual connections. LayerNorm

(LN) is deployed before MSA and MLP, whereas Dropout (DP) is applied afterward. The procedure in the ℓth block is summarized by Eq. 6, where $\ell = 1, \ldots, L$.

$$\hat{\mathcal{Z}}^{(\ell)} = \text{DP}(\text{MSA}(\text{LN}(\mathcal{Z}^{(\ell-1)}))) + \mathcal{Z}^{(\ell-1)}$$
$$\mathcal{Z}^{(\ell)} = \text{DP}(\text{MLP}(\text{LN}(\hat{\mathcal{Z}}^{(\ell)}))) + \hat{\mathcal{Z}}^{(\ell)} \qquad (6)$$

Note the input of each attention block is not equipped with the *position encoding* as in standard Transformer [56], since the views are permutation invariant in the set. Also, VSFormer does not insert the *class token* in the input as the goal is to grasp the correlations within views rather than learning the relations between the class token and views. Surprisingly, a lightweight view set encoder (2.7M parameters) that is only composite of two attention blocks can work quite well (99.0% overall accuracy), validated by extensive experiments in Section IV.

For 3D shape recognition or retrieval tasks, it is necessary to convert the learned higher-order view set representations into an expressive descriptor $\mathbf{d} \in \mathbb{R}^G$ via a transition module (*e.g.*, concatenation of max and mean pooling). Then the descriptor is processed by a decoder to generate the prediction $\hat{\mathbf{y}} \in \mathbb{R}^K$.

$$\mathbf{d} = \text{Transition}(\mathcal{Z}^{(L)}) \qquad (7)$$
$$\hat{\mathbf{y}} = \text{Decoder}(\mathbf{d}) \qquad (8)$$

The objective to be optimized is defined as *Cross Entropy* loss for 3D shape recognition, where $\mathbf{y}_i$ is the predicted class distribution of ith object and θ_W denotes all learnable parameters in the model.

$$\mathcal{L}_{CE}(\mathbf{y}_i, \hat{\mathbf{y}}_i; \theta_W) = \sum_i -\mathbf{y}_i \log \hat{\mathbf{y}}_i \qquad (9)$$

These modules have various design choices, we will examine the design choices of each component through the ablation studies in Section IV-C.

C. Implementation Details

Architecture. For Initializer, we adopt lightweight CNNs. There are several candidates (AlexNet, ResNet18, *etc.*) and we will compare them later. $\mathcal{V}$ is instantiated as a random permutation of the views, $v_i \in \mathcal{V}$ is mapped to a D=512 dimensional vector through Initializer. For Encoder, there are L=4 attention blocks and within each block, the MSA layer has 8 attention heads and the widening factor of the MLP hidden layer is 2. The normalized function Norm is defined as softmax($\cdot$) and the temperature coefficient τ is set to $\sqrt{\frac{D}{8}}$. The Transition module converts $\mathcal{Z}^{(L)}$ into a G=1024 dimensional descriptor $\mathbf{d}$. Finally, the descriptor is projected to a category distribution by Decoder, which is a 2-layer MLP of shape $\{1024, 512, K\}$. The design choices are verified by ablated studies in Section IV-C.

Optimization Strategy. The optimization objective $\mathcal{L}_{CE}$ has a label smooth of 0.1. Following previous methods [25], [39], the learning is divided into two stages. In the first stage, the initializer is individually trained on the target dataset for 3D shape recognition. The purpose is to provide good initializations for views. In the second stage, the pre-trained initializer is loaded and jointly optimized with other modules

TABLE I: Comparison of 3D shape recognition on Model-Net40.

Method	Input	Class Acc. (%)	Inst. Acc. (%)
3DShapeNets [19]		77.3	–
VoxNet [20]		83.0	–
VRN Ensemble [61]	Voxels	–	95.5
HCNN [23]		–	89.4
MVCNN-MR [21]		91.4	93.8
PointNet++ [7]		–	91.9
DGCNN [8]		90.2	92.9
RSCNN [12]		–	93.6
KPConv [9]		–	92.9
SimpleView [62]	Points	90.5	93.0
CurveNet [14]		–	93.8
PointMLP [17]		91.3	94.1
RIF [18]		–	89.3
MVCNN [24]		90.1	90.1
MVCNN-new [25]		92.4	95.0
MHBN [47]		93.1	94.7
GVCNN [26]		90.7	93.1
RCPCNN [29]		–	93.8
RN [52]		92.3	94.3
3D2SeqViews [35]		91.5	93.4
SV2SL [34]		91.1	93.3
VERAM [36]		92.1	93.7
Ma *et al.* [37]		–	91.5
iMHL [38]	Views	–	97.2
HGNN [30]		–	96.7
HGNN$^+$ [42]		–	96.9
View-GCN [39]		96.5	97.6
View-GCN++ [41]		96.5	97.6
DeepCCFV [63]		–	92.5
EMV [31]		92.6	94.7
RotationNet [51]		–	97.4
MVT [45]		–	97.5
CARNet [46]		–	97.7
MVTN [40]		92.2	93.5
MVPNet [44]		96.8	97.9
VSFormer	Views	**98.9**	**98.8**

TABLE II: Comparison of 3D shape recognition on ScanObjectNN.

Method	Input	Class Acc. (%)	Inst. Acc. (%)
PointNet++ [35]		82.1	84.3
SpiderCNN [70]	Points	77.4	79.5
PointCNN [71]		–	85.5
DGCNN [8]		84.0	86.2
MVCNN-M		85.0	86.8
RotationNet [51]		84.9	86.9
CVR [69]		88.4	90.8
View-GCN [39]	Views	88.3	90.4
View-GCN++ [41]		89.1	91.3
MVTN [34]		–	92.3
VointNet [43]		–	94.0
VSFormer	Views	**94.6**	**95.9**

TABLE III: Comparison of 3D shape recognition on RGBD.

Method	#Views	Inst. Acc. (%)
CFK [72]	$\geq$ 120	86.8
MMDCNN [73]	$\geq$ 120	89.5
MDSICNN [74]	$\geq$ 120	89.6
MVCNN [24]	12	86.1
RotationNet [51]	12	89.3
View-GCN(ResNet18) [39]	12	94.3
View-GCN(ResNet50) [39]	12	93.9
VSFormer(ResNet18)	12	**98.4**
VSFormer(ResNet50)	12	**95.6**

on the same dataset. Experiments in Figure 7a show this strategy will significantly improve performance in a shorter period.

Network Training. For Initializer, we train it 30 epochs on the target dataset using SGD [64], with an initial learning rate of 0.01 and CosineAnnealingLR scheduler. After that, the pre-trained weights of Initializer are loaded into VSFormer to be optimized with other modules jointly. Specifically, VSFormer is trained 300 epochs on the target dataset using AdamW [65], with an initial peak learning rate of 0.001 and CosAnnealing-WarmupRestartsLR scheduler [66]. The restart interval is 100 epochs and the warmup happens in the first 5 epochs of each interval. The learning rate increases to the peak linearly during warmup and the peak decays by 40% after each interval. The learning rate curve is visualized in Figure 7b.

IV. EXPERIMENTS

In this section, VSFormer is evaluated on 3D shape recognition and retrieval tasks. Then we conduct controlled experiments to examine the design choices of the proposed method.

A. 3D Shape Recognition

Datasets. We conduct 3D shape recognition on three datasets, ModelNet40 [19], ScanObjectNN [67] and RGBD [68].

- ModelNet40 includes 12,311 objects across 40 categories and we use its rendered version as in previous work [25], [39], where each object corresponds to 20 views.
- ScanObjectNN is collected from real-world scans and poses great challenges to existing methods. There are 2,902 objects distributed in 15 categories. We follow previous work [39]–[41], [69] to adopt the OBJ_ONLY split and render 20 views for each object. To ease reproduction, we wrote a detailed instruction to explain the rendering procedure, and please refer to this blog[1].
- RGBD is a large-scale, hierarchical multi-view object dataset [68], containing 300 objects organized into 51 classes. In RGBD, we use 12 views for each 3D object as in [39].

Metrics. Two evaluation metrics are computed for 3D shape recognition: mean class accuracy (Class Acc.) and instance accuracy (Inst. Acc.). We record the best results of these metrics during optimization.

Results. Table I compares representative methods on Model-Net40 and these methods have different input formats: voxels, points and views. VSFormer achieves 98.9% mean class accuracy and 98.8% overall accuracy, surpassing the voxel-

[1]https://auniquesun.github.io/2023-06-16-multi-view-rendering/

TABLE IV: Comparison of 3D shape retrieval on the normal version of ShapeNet Core55.

Method	micro					macro				
	P@N	R@N	F1@N	mAP	NDCG	P@N	R@N	F1@N	mAP	NDCG
ZFDR	53.5	25.6	28.2	19.9	33.0	21.9	40.9	19.7	25.5	37.7
DeepVoxNet	79.3	21.1	25.3	19.2	27.7	59.8	28.3	25.8	23.2	33.7
DLAN	81.8	68.9	71.2	66.3	76.2	61.8	53.3	50.5	47.7	56.3
GIFT [75]	70.6	69.5	68.9	64.0	76.5	44.4	53.1	45.4	44.7	54.8
Improved GIFT [76]	78.6	77.3	76.7	72.2	82.7	59.2	65.4	58.1	57.5	65.7
ReVGG	76.5	80.3	77.2	74.9	82.8	51.8	60.1	51.9	49.6	55.9
MVFusionNet	74.3	67.7	69.2	62.2	73.2	52.3	49.4	48.4	41.8	50.2
CM-VGG5-6DB	41.8	71.7	47.9	54.0	65.4	12.2	66.7	16.6	33.9	40.4
MVCNN [24]	77.0	77.0	76.4	73.5	81.5	57.1	62.5	57.5	56.6	64.0
RotationNet [51]	81.0	80.1	79.8	77.2	86.5	60.2	63.9	59.0	58.3	65.6
View-GCN [39]	81.8	80.9	80.6	78.4	85.2	62.9	65.2	61.1	60.2	66.5
View-GCN++ [41]	81.2	79.9	80.0	77.5	83.9	61.2	65.8	61.1	59.0	63.8
MVPNet [44]	81.2	81.1	80.8	78.5	86.3	61.5	66.9	62.5	61.2	66.5
MRVA-Net [54]	80.0	**83.0**	80.8	79.2	**87.3**	61.2	**69.8**	62.3	63.1	69.6
VSFormer	**82.3**	82.6	**82.0**	**79.3**	82.4	**67.0**	69.7	**66.7**	**64.2**	**70.4**

TABLE V: Ablation Study: the architecture of Encoder.

#Blocks	2	2	2	2	4	4	4	4	6	6	6	6
#Heads	6	8	6	8	6	8	6	8	6	8	6	8
Ratio_{mlp}	2	2	4	4	2	2	4	4	2	2	4	4
Dim_{view}	384	512	384	512	384	512	384	512	384	512	384	512
#Params (M)	2.7	4.8	3.9	6.9	5.0	9.0	7.4	13.2	7.4	13.2	11.0	19.5
ModelNet40												
Class Acc. (%)	98.8	98.7	98.4	97.2	97.4	98.9	**99.1**	98.2	98.7	98.2	98.4	98.1
Inst. Acc. (%)	**99.0**	98.8	98.5	98.1	97.6	98.8	98.5	98.5	98.3	98.3	98.1	98.3

and point-based counterparts. Also, it sets new records in view-based methods. For example, compared to early works [24]–[26], [29], [47] that aggregate multi-view information independently by pooling or some variants, VSFormer exceeds their instance accuracies by 3.8% at least. VSFormer also significantly improves the results of methods built on view sequence, such as RelationNet [52], 3D2SeqViews [35], SeqViews2SeqLabels [34], VERAM [36]. Methods defined on view graph and hyper-graph achieve decent performances [30], [38], [39], [41], [42] because of enhanced information flow among views. VSFormer still outreaches the strongest baseline of this category, increasing 2.4% Class Acc. and 1.2% Inst Acc. over View-GCN [39].

Table II exhibits the evaluation results of various methods on the real-scan ScanObjectNN [67] dataset. Apparently, VSFormer outreaches existing strong baselines in terms of class and instance recognition accuracies, leading View-GCN++ [41] by 5.5% class accuracy and VointNet [43] by 0.9% instance accuracy. The results confirm the proposed method still works well when handling cluttered and occluded views.

Table III records the comparison with related work on the challenging RGBD [68] dataset. The dataset designs 10-fold cross-validation for multi-view 3D object recognition. We follow this setting and report the average instance accuracy of 10 folds. VSFormer shows consistent improvements over View-GCN under the same initializations. Especially, it gets 98.4% accuracy, which is a 4.1% absolute improvement over the runner-up, suggesting VSFormer can produce more expressive shape descriptors when dealing with challenging cases.

B. 3D Shape Retrieval

Datasets. 3D shape retrieval aims to find a rank list of shapes most relevant to the query in a given dataset. We conduct this task on ShapeNet Core55 [2], [58]. The dataset is split into train/val/test set and there are 35764, 5133 and 10265 meshes, respectively. 20 views are rendered for each mesh as in [39] and we explain the procedure in this blog[2]. ShapeNet Core55 has two rendered versions (normal and perturbed) and we report results on the normal one as in previous work [37], [39], [51].

Metrics. According to the SHREC'17 benchmark [58], the rank list is evaluated based on the ground truth category and subcategory. If a retrieved shape in a rank list has the same category as the query, it is positive. Otherwise, it is negative. The evaluation metrics include micro and macro P@N, R@N, F1@N, mAP and NDCG. P@N and R@N mean the precision and recall when the length of the returned rank list is N (1,000 by default). NDCG that represents normalized discounted cumulative gain is a measure of ranking quality. Please refer to [58] for more details about the metrics.

Retrieval. We generate the rank list for each query shape in two steps. First, VSFormer is trained to recognize the shape categories in ShapeNet Core55 [2]. We retrieve shapes that have the same predicted class as the query $\mathcal{Q}$ and rank the retrieved shapes according to class probabilities in descending order, resulting in L_1. Second, we train another VSFormer to recognize the shape subcategories of ShapeNet Core55 [2], then re-rank L_1 to ensure shapes that have same predicted subcategory as the query $\mathcal{Q}$ rank before shapes that are not in same subcategory with $\mathcal{Q}$ and keep the remaining unchanged,

[2]https://auniquesun.github.io/2023-01-15-shapenetcore55-rendering/

resulting in L_2, which is regarded as the final rank list for the query $\mathcal{Q}$.

Results. VSFormer is compared with the methods that report results on SHREC'17 benchmark [58], shown in Table IV. The methods in the first three rows use voxels as inputs, while the remaining ones exploit views. The overall performances of view-based methods are better than voxel-based ones. Previously, MRVA-Net achieved state-of-the-art results by extracting multi-range view features and fusing the features with ViT. But experiments show VSFormer goes beyond MRVA-Net in 7 out 10 metrics, including micro P@N, F1@N and mAP as well as macro P@N, F1@N, mAP and NDCG. In particular, we achieve 2.3% and 5.8% absolute improvements for micro and macro P@N over MRVA-Net. On the other hand, the Transition module implemented as a concatenation of max and mean pooling will inevitably lose a bunch of grasped correlations. The model can be confused by the highly similar 3D shapes in appearance and suffers from poor rankings for the shapes, which results in lower micro NDCG.

C. Ablation Studies

We conducted controlled experiments to verify the choices of different modules in VSFormer design, analyze the impact of patch-level correlations and the number of views. The used dataset is ModelNet40.

1) Encoder: **The Architecture of Encoder.** We provide ablations to justify the design choices of Encoder. The controlled variables of Encoder are the number of attention blocks (#Blocks), the number of attention heads in MSA (#Heads), the widening ratio of MLP hidden layer ($Ratio_{mlp}$) and the dimension of the view representations (Dim_{view}). The mean class acc. and instance acc. of VSFormer with different encoder structures are compared in Table V. All design variants show high-level performances and surpass the existing state of the art. Surprisingly, the encoder consisting of only 2 attention blocks can facilitate VSFormer to achieve 99.0% overall accuracy. The results are in line with expectations as the size of a view set is relatively small thus, it is unnecessary to design a heavy encoder. At the same time, it is inspiring that a shallow encoder can enrich and grasp the pairwise and higher-order correlations of elements in the view set well. Finally, we select the design that takes *the second place* in both mean class and instance accuracy, namely #Blocks = 4, #Heads = 8, $Ratio_{mlp}$ = 2 and Dim_{view} = 512.

Performance Gains Delivered by Our Encoder. We investigate the performance gains delivered by the devised view set encoder. There are two settings. 1) The initializer is individually trained to recognize 3D shapes. 2) The devised encoder is appended upon the pre-trained initializer to further capture the feature interactions among views. Table VI compares different configurations described above. Notable performance gains are obtained over different initializers. For example, by appending only 2 attention blocks (4.8M #Params) on the AlexNet initializer, our model achieves 18.3% and 13.7% absolute improvements for mean class accuracy and instance accuracy.

Position Encoding. According to the view set perspective, VSFormer should be unaware of the order of elements in the

TABLE VI: Ablation study: the performance gains brought by the devised encoder.

Module	#Params(M)	Class Acc.(%)	Inst. Acc.(%)
AlexNet			
w/o encoder	42.3	80.6	85.1
w/ encoder	47.1↑ 4.8	**98.9**↑ 18.3	**98.8**↑ 13.7
ResNet18			
w/o encoder	11.2	88.7	91.8
w/ encoder	16.0↑ 4.8	**96.7**↑ 8.0	**97.6**↑ 5.8

view set, so we remove the position encoding from the devised encoder. We examine this design in Table VII. The results show if learnable position embeddings are forcibly injected into the initialized view features to make the model position-aware, the performance will be hindered, dropping by 0.5% for class accuracy and 0.3% for overall accuracy.

Class Token. Unlike standard Transformer [56], the proposed method does not insert the class token into the inputs since it is irrelevant to the target of capturing the correlations among views in the set. This claim is supported by the results in Table VII, which shows that inserting the class token decreases recognition accuracies.

TABLE VII: Ablation study: position encoding and class token.

Variants	Class Acc. (%)	Inst. Acc. (%)
w/ pos. enc.	98.4	98.5
w/o pos. enc.	**98.9**↑ 0.5	**98.8**↑ 0.3
w/ cls. token	98.8	98.5
w/o cls. token	**98.9**↑ 0.1	**98.8**↑ 0.3

Number of Attention Blocks. In VSFormer, the number of attention blocks in Encoder is considerably compressed because the size of a view set is relatively small and it is unnecessary to deploy a deeper encoder to model the interactions between the views in the set. The results in Table VIII demonstrate the encoder can be highly lightweight, as light as 2 attention blocks, but with 98.8% overall accuracy that exceeds all existing methods. The results also indicate increasing the attention blocks does not receive gains but additional parameters and overheads.

TABLE VIII: Ablation study: number of attention blocks.

Module	#Params (M)	Inst. Acc. (%)
AlexNet	42.3	85.1
+ 2 Attn. Blocks	**4.8**	**98.8**↑ 13.7
+ 4 Attn. Blocks	9.0	**98.8**↑ 13.7
+ 6 Attn. Blocks	13.2	98.3↑ 13.2

2) Initializer: We explore different means to initialize view representations, including shallow convolution operations and lightweight CNNs. The idea of shallow convolution operation is inspired by the image patch projection (1x1 Conv) in ViT [57] and the specific configurations are explained in Table XVII. Table IX compares their recognition accuracies.

We observe that initializations by 1- and 2-layer convolution operations do not yield satisfactory results. Instead, lightweight CNNs work well, especially when receiving the initialized features from AlexNet and jointly optimizing with other modules, VSFormer reaches 98.9% class accuracy and 98.8% overall accuracy, both are new records on ModelNet40. By default, AlexNet serves as the Initializer module.

TABLE IX: Ablation study: choices for Initializer.

Initializer	#Params (M)	Class Acc. (%)	Inst. Acc. (%)
1-layer Conv	102.8	90.1	92.5
2-layer Conv	12.9	88.9	93.7
alexnet	42.3	**98.9**	**98.8**
resnet18	11.2	96.7	97.6
resnet34	21.3	96.9	97.1

3) Transition: We investigate three kinds of operations for the Transition module. The results are reported in Table X. We find the simple pooling operations (Max and Mean) can work well (98.0+% Acc.) and both outreach the performances of the previous state of the art. By concatenating the outputs of max and mean pooling, the optimization is more stable and the overall accuracy is lifted to 98.8%. It is worth noting that the same pooling operations are adopted by MVCNN [24] and its variants [25], [26], [29], [47], [48], but their accuracies are up to 95.0%, implying that the view set descriptors learned by our encoder are more informative.

TABLE X: Ablation study: choices for Transition.

Transition	Class Acc. (%)	Inst. Acc. (%)
Max pooling	**99.1**	98.5
Mean pooling	98.5	98.5
Concat(Max&Mean)	98.9	**98.8**

4) Decoder: The decoder projects the view set descriptor to a shape category distribution. The choices for the decoder are compared in Table XI. VSFormer with a decoder of a single Linear can recognize 3D shapes at 98.1% instance accuracy, which outperforms all existing methods and again, reflects the summarized view set descriptor is highly discriminative. The advantage is enlarged when the decoder is deepened to a 2-layer MLP. However, further tests show it is unnecessary to exploit deeper transformations.

TABLE XI: Ablation study: choices for Decoder.

Decoder	Class Acc. (%)	Inst. Acc. (%)
1-layer	97.9	98.1
2-layer	**98.9**	**98.8**
3-layer	98.5	98.5

5) Effect of the Patch-level Feature Correlations: Some other methods, such as MHBN [47], [48], MVT [45], Car-Net [46], also consider patch-level interactions. They want to enhance multi-view information flow by integrating patch-level features. In this work, we examine the effect of patch-level feature correlations by injecting them into each attention block of the encoder. The results in Table XII show injecting patch-level features is redundant and unnecessary. A major reason is the view-level correlations are already well understood by our model for 3d shape analysis. Many fine-grained patches are similar in different views and lack the sense of overall shape, some of them are even blank backgrounds, thus contribute little to the target task. However, no matter with or without the patch-level correlations, VSFormer maintains high-level performances (98.1% class and inst. accuracies) and surpasses all existing models.

TABLE XII: Ablation study: effect of the patch-level correlations.

Variants	Class Acc. (%)	Inst. Acc. (%)
w/ patch	98.1	98.1
w/o patch	**98.9**↑ 0.8	**98.8**↑ 0.7

6) Effect of the Number of Views: We investigate the effect of the number of views on the recognition performance, shown in Table XIII. There are up to 20 views for each 3D shape and M views are randomly selected for each shape during training and evaluation, where $M \in \{1, 4, 8, 12, 16, 20\}$. When $M = 1$, the problem is equivalent to single-view object recognition, so there is no interaction among views. In this case, a lightweight ResNet18 [59] is trained for recognition and it achieves 89.0% mean class accuracy and 91.8% instance accuracy. When the number of views increases, the performances are quickly improved. For instance, after aggregating the correlations from 4 views, VSFormer lifts 8.4% and 5.3% absolute points in class and instance accuracy, respectively. But exploiting more views does not necessarily lead to better accuracy. The 8-view VSFormer reaches 98.0% class accuracy and 98.8% overall accuracy, outperforming 12- and 16-view versions. The performance is optimal when exploiting all 20 views and we choose this version to compare with other view-based methods.

TABLE XIII: Ablation study: effect of the number of views.

#Views	1	4	8	12	16	20
Class Acc. (%)	89.0	97.4	98.0	97.5	97.7	**98.9**
Inst. Acc. (%)	91.8	97.1	**98.8**	97.6	98.3	**98.8**

V. VISUALIZATION

This section exhibits various visualizations of the predictions and intermediate results given by VSFormer, which are helpful for having a better understanding of our method.

Multi-view Attention in Colored Lines. We randomly select a 3D shape that is a nightstand, then visualize the multi-view correlations of eight views of this shape, referring to Figure 3. The correlations are represented by the attention scores emitted by the last attention block of VSFormer. The scores are normalized to map to the color bar on the far right of the figure. Our model distributes more weights to the 2nd, 3rd and 6th views from the 5th. The results seem reasonable

since these views are more discriminative according to visual appearances.
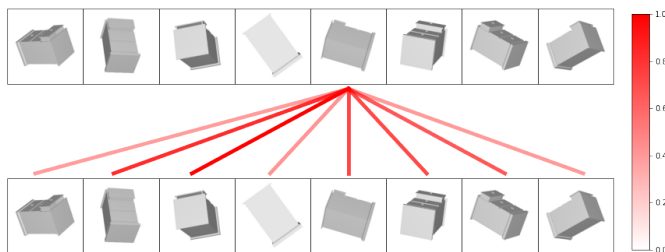

Fig. 3: Visualization of multi-view attention of 8 views of a nightstand in colored lines.

Multi-view Attention Map. For better understanding, we visualize the attention map of eight views of a 3D airplane in Figure 4. The attention scores are taken from the outputs of the last attention block of our model. We normalize the attention scores so that each score is rounded to three decimal digits and the sum of the scores for each row is equivalent to 1. Based on the 3rd or 7th view, one may not be confident that the shape is a airplane, and our model assigns relatively small weights to them. Instead, the map indicates the 6th view is representative since it receives more attention from other views. On the other hand, we can manually infer the 6th view is representative based on the visual appearances of these views. The results reflect that the proposed model can adaptively capture the multi-view correlations and allocate different views with reasonable weights for recognition.

Fig. 4: Visualization of the attention scores for 8 views of a 3D airplane.

3D Shape Recognition. We visualize the feature distribution for different shape categories on ScanObjectNN, ModelNet40 and RGBD using t-SNE [77], shown in Figure 5. It shows different shape categories in different datasets are successfully distinguished by the proposed method, demonstrating that VS-

Former understands multi-view information well by explicitly modeling the correlations for all view pairs in the view set.
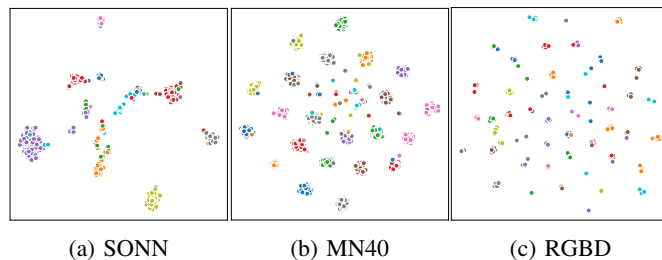

Fig. 5: Visualization of 3D shape **feature distribution** on (a) ScanObjectNN (SONN) of 15 classes (b) ModelNet40 (MN40) of 40 classes (c) RGBD of 51 classes.

3D Shape Retrieval. We visualize the top 10 retrieved shapes for 10 typical queries in Figure 6. The retrieval happens in the ShapeNet Core55 validation set. Each retrieved shape is represented by its random view. We observe the top 10 results are 100% relevant to the query, which means they belong to the same category. The 5th retrieved shape in the 3rd row may be confusing. But when observing other views of this shape in ShapeNet, it is also a cup.
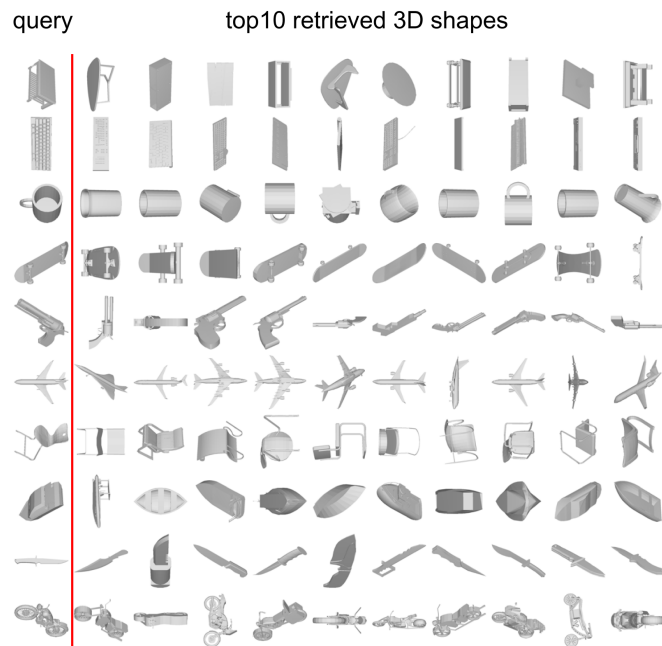

Fig. 6: Visualization of the top 10 retrieved results for each query shape.

VI. ADDITIONAL ANALYSIS

We provide additional analysis of the proposed approach, including network training, inference speed, and auxiliary observations in the initializer module.

A. Network Training

The following settings will verify the adopted 2-stage training strategy and excellent learning efficiency of VSFormer.
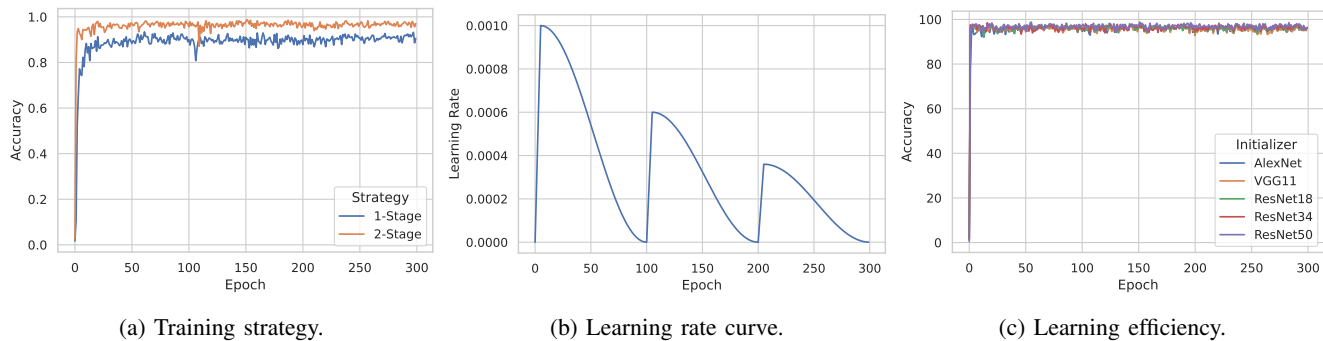
Fig. 7: (a) Comparison of instance accuracy using 1-stage and 2-stage optimization on ModelNet40. (b) The learning rate curve of AdamW for VSFormer. (c) Learning efficiency of the VSFormer variants using different initializers.

Optimization Strategy. We compare the effectiveness of 1-stage and 2-stage optimization on ModelNet40. For 2-stage optimization, Initializer is trained on the dataset individually, then the pre-trained weights of Initializer are loaded into VSFormer to be jointly optimized with other modules. The 1-stage optimization means VSFormer learns in an end-to-end way and all parameters are randomly initialized. Figure 7a shows the recognition accuracy achieved by 2-stage optimization is significantly better than that of 1-stage training. The results demonstrate that VSFormer receives gains from the well-initialized view representations provided by the first stage.

Learning Efficiency. We explore the learning efficiency of VSFormer by freezing the weights of the pre-trained Initializer. Figure 7c displays the recognition accuracy curves of the VSFormer variants with different initializers on ModelNet40 during training. Regardless of Initializer used, all variants' performances soared after a short training and approached the highest. For instance, VSFormer with ResNet34 Initializer reaches 97.6% instance accuracy after *only 2 epochs*, while View-GCN achieves the same performance with 7.5x longer optimization. The results reflect the proposed method has higher learning efficiency than the previous state of the art.

B. Inference Speed

Table XIV compares the number of parameters, inference speed and recognition accuracy of different models. The experiments take place on ModelNet40 using a 2080Ti GPU. The inference speed counts the number of 3D objects each model processes per second (obj/s). Only the forward pass that outputs the prediction is considered, not including data preparation and loss computation. Note that each object corresponds to 20 views. OA is short for overall accuracy and mAcc refers to mean class accuracy.

We select MVCNN and View-GCN for comparison. The choice considers the availability and usability of the official code. The former is a pioneer work in the field and the latter was the previous SOTA method. MVCNN has the least #Params and runs much faster than View-GCN and VSFormer. View-GCN has the most #Params and significantly improves recognition accuracy over MVCNN. But when combining these metrics as a whole, VSFormer demonstrates consistent advantages under different initializers, achieving efficient

inference speed and best recognition results. The proposed method is 3.8-6.4x faster than View-GCN when employing different initializers. Note that a large portion of parameters in VSFormer are attributed to the initializer, and the view set encoder is highly lightweight. For example, our encoder contains only 9.0M parameters (51.3M-42.3M), compared to 42.3M in AlexNet. Readers may notice that VSFormer with AlexNet (51.3M #Params) is faster than that with ResNet18 (20.1M #Params). It is due to AlexNet having fewer multiply-add operations than ResNet18, *e.g.*, 0.72 vs. 1.82 in GFLOPS.

TABLE XIV: Comparison of the number of parameters, inference speed and accuracy of different methods.

Method	Initializer	#Params (M)	Speed (obj/s)	OA (%)	mAcc (%)
MVCNN [24]		**11.2**	**1621.3**	91.7	87.3
View-GCN [39]	ResNet18	33.9	131.8	96.1	96.1
VSFormer		20.1	979.4	**97.1**	**96.4**
MVCNN [24]		**42.3**	**2710.0**	90.0	86.4
View-GCN [39]	AlexNet	84.6	325.7	91.5	88.9
VSFormer		51.3	1562.5	**97.8**	**97.5**

C. Bad Case Analysis

Here we carry out a bad case analysis of the proposed model. The study covers recognition and retrieval tasks on different datasets, including ModelNet40 and SHREC'17. The input views come from the corresponding test set and the incorrect predictions of our model are visualized in Figure 8. For the recognition task, the model is confused by shapes with highly similar appearances, resulting in incorrect outputs on occasion. As the subfigure 8a displays, our model predicts the *bathtub* views as the *bowl* category, the *bookshelf* views as *table* category, and the *bottle* views as *flower pot* category. The *bottle* and *flower pot* have close appearances and share the function of holding water. Note that the views in the right part (Prediction) are found in the training set and they are hard to distinguish from corresponding input views, even for human beings.

For 3D shape retrieval, its performance is affected by classification accuracy since the misclassified result of a query shape will propagate in the retrieval process, where the model tries to find shapes that have the same category as the query.

(a) Incorrect predictions for multi-view shape recognition on Model-Net40

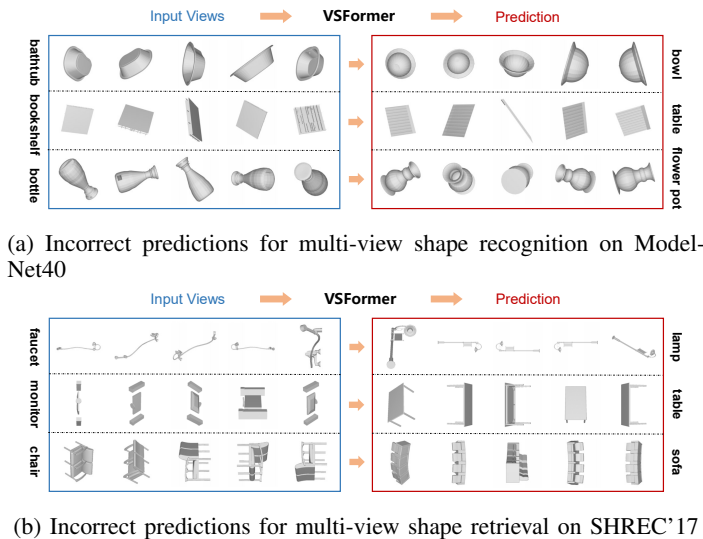(b) Incorrect predictions for multi-view shape retrieval on SHREC'17

Fig. 8: Bad case analysis of the proposed model for the multi-view recognition and retrieval tasks.

Here we visualize the misclassification of several query shapes on the SHREC'17 benchmark, exhibited in the subfigure 8b. For instance, the query in the first row is a faucet but our model recognizes it as a lamp. The misclassification is somewhat understandable as there are views of *lamp* in the training set with extremely close appearances with *faucet*, seeing the corresponding prediction part. Interestingly, in the third row, our model regards two chairs side by side as *sofa*, probably because it learns the common sense that *sofa* is more likely to have consecutive seats than *chair*.

D. Auxiliary Observations

Optimizer and Scheduler. We examine whether VSFormer is sensitive to the optimizer and lr scheduler by replacing them with common settings, such as Adam and CosineAnnealingLR. The recognition accuracies on 3 datasets are recorded in Table XV, where the last column measures the change relative to the default configuration. We noticed that there was a slight drop in accuracy when training with Adam and CosineAnnealingLR.

TABLE XV: The recognition accuracy of VSFormer when optimizing with Adam and CosineAnnealingLR (CosLR). ModelNet40: MN40, ScanObjectNN: SONN.

Method	Optim.	Sched.	Dataset	OA (%)	Δ (%)
VSFormer	Adam	CosLR	MN40	98.3	0.5 ↓
			SONN	95.9	0.0
			RGBD	98.0	0.4 ↓

Different Methods Using Same Initializer. To be fair, we use the same Initializer for different methods to inspect their recognition accuracies on ModelNet40. The chosen methods are strong baselines, RotationNet [51] and View-GCN [39]. The results in Table XVI show VSFormer can achieve higher-level performance no matter which initializer is used, exceeding View-GCN(AlexNet) and View-GCN(ResNet) by 1.6%

and 1.5%, respectively. The results also indicate the proposed approach is better at grasping multi-view information for recognition since the initialized view features are identical.

TABLE XVI: Comparison of different multi-view methods with same Initializer.

Method	Initializer	Inst. Acc. (%)
RotationNet	AlexNet	96.4
View-GCN		97.2
VSFormer		**98.8**
RotationNet	ResNet50	96.9
View-GCN		97.3
VSFormer		**98.8**

Shallow Convolutions in Initializer. We investigate the performances of VSFormer when deploying shallow convolution operations as Initializer, *e.g.*, 1- and 2-layer convolution. Table XVII explains their specific configurations. Due to the increased number of strides, 2-layer convolution has much lower parameters than 1-layer operation. However, VSFormer with shallow convolution initializations does not lead to decent 3D shape recognition. The best instance accuracy is 93.7%, much lower than 98.8% given by VSFormer with lightweight CNN (AlexNet) Initializer, suggesting lightweight CNNs are reasonable choices for the Init module.

TABLE XVII: The configurations of shallow convolutions in Initializer.

Conv(s)	1-layer	2-layer
View Size	$224 \times 224 \times 3$	
1st Conv	Conv2d(in=3,out=64,k=7,s=2,p=3)	
	BathNorm2d(num=64)	
	ReLU(inplace=True)	
	MaxPool2d(k=3,s=2,p=1)	
2nd Conv	None	Conv2d(in=64,out=32,k=3,s=2,p=1)
		BatchNorm2d(num=32)
		ReLU(inplace=True)
#Params (M)	102.8	**12.9**
Class Acc. (%)	**90.1**	88.9
Inst. Acc. (%)	92.5	**93.7**

E. Limitations

It is worth noting that VSFormer has some limitations. First, the Transition module may be a weak point. In many related works, this module is designed as a pooling operation or some variant. We employ a concatenation of max and mean pooling to summarize the higher-order view correlations into a descriptor, which will inevitably lose a part of well-learned correlations. Second, it may not be necessary to adopt the same architecture to execute recognition and retrieval tasks. 3D shape retrieval is more complex as it requires finding highly relevant shapes and generating a rank list for them. The recognition and retrieval models can share an encoder but vary in the decoder. For challenging scenarios, the decoder in the retrieval model can directly operate on the grasped higher-order view representations instead of the compressed descriptor.

VII. CONCLUSION

This paper presents VSFormer, a succinct and effective multi-view 3D shape analysis method. We organize the different views of a 3D shape into a permutation-invariant set and devise a lightweight attention model to capture the correlations of all view pairs. A theoretical analysis is provided to bridge the view set and attention mechanism. VSFormer shows outstanding performances across different datasets and sets new records for recognition and retrieval tasks.

In the future, we plan to investigate new paradigms of aggregating the well-learned multi-view correlations without losing useful information in the transition. Besides, we are interested in exploring more sophisticated designs for the retrieval task and evaluating on diverse benchmarks. As this paper suggested, pairwise similarities are not sufficient to capture the intrinsic structure of the data manifold [78]. It is also challenging but worthwhile to extend 3D shape analysis to scene-level tasks, such as multi-view 3D semantic segmentation and object detection.

ACKNOWLEDGMENTS

REFERENCES

[1] A. Huang, G. Nielson, A. Razdan, G. Farin, D. Baluch, and D. Capco, "Thin structure segmentation and visualization in three-dimensional biomedical images: a shape-based approach," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 1, pp. 93–102, 2006. 1

[2] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "Shapenet: An information-rich 3d model repository," *CoRR*, vol. abs/1512.03012, 2015. [Online]. Available: http://arxiv.org/abs/1512.03012 1, 7

[3] L. Gao, Y.-P. Cao, Y.-K. Lai, H.-Z. Huang, L. Kobbelt, and S.-M. Hu, "Active exploration of large 3d model repositories," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 12, pp. 1390–1402, 2015. 1

[4] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi, "Objaverse: A universe of annotated 3d objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 13 142–13 153. 1

[5] T. Wu, J. Zhang, X. Fu, Y. Wang, J. Ren, L. Pan, W. Wu, L. Yang, J. Wang, C. Qian, D. Lin, and Z. Liu, "Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 803–814. 1

[6] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1

[7] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf 1, 6

[8] 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 (TOG)*, 2019. 1, 6

[9] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1, 6

[10] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[11] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia, "Pointweb: Enhancing local neighborhood features for point cloud processing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[12] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8895–8904. 1, 6

[13] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, "Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1

[14] T. Xiang, C. Zhang, Y. Song, J. Yu, and W. Cai, "Walk in the cloud: Learning curves for point clouds shape analysis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 915–924. 1, 6

[15] S. S. Mohammadi, Y. Wang, and A. D. Bue, "Pointview-gcn: 3d shape classification with multi-view point clouds," in *2021 IEEE International Conference on Image Processing (ICIP)*, 2021, pp. 3103–3107. 1

[16] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 16 259–16 268. 1

[17] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, "Rethinking network design and local geometry in point cloud: A simple residual MLP framework," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=3Pbra-_u76D 1, 6

[18] X. Li, R. Li, G. Chen, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "A rotation-invariant framework for deep point cloud analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 12, pp. 4503–4514, 2022. 1, 6

[19] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1, 6

[20] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 922–928. 1, 6

[21] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view cnns for object classification on 3d data," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5648–5656. 1, 6

[22] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[23] T. Shao, Y. Yang, Y. Weng, Q. Hou, and K. Zhou, "H-cnn: Spatial hashing based cnn for 3d shape analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 7, pp. 2403–2416, 2020. 1, 6

[24] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 945–953. 1, 2, 3, 6, 7, 9, 11

[25] J.-C. Su, M. Gadelha, R. Wang, and S. Maji, "A deeper look at 3d shape classifiers," in *Computer Vision – ECCV 2018 Workshops*, L. Leal-Taixé and S. Roth, Eds. Cham: Springer International Publishing, 2018, pp. 645–661. 1, 2, 5, 6, 7, 9

[26] Y. Feng, Z. Zhang, X. Zhao, R. Ji, and Y. Gao, "Gvcnn: Group-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2, 6, 7, 9

[27] H. You, Y. Feng, R. Ji, and Y. Gao, "Pvnet: A joint convolutional network of point cloud and multi-view for 3d shape recognition," in *Proceedings of the 26th ACM International Conference on Multimedia*, ser. MM'18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1310–1318. [Online]. Available: https://doi.org/10.1145/3240508.3240702 1, 2

[28] C. Xu, B. Leng, C. Zhang, and X. Zhou, "Emphasizing 3d properties in recurrent multi-view aggregation for 3d shape retrieval," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr.

2018. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/12309 1, 2

[29] C. Wang, M. Pelillo, and K. Siddiqi, "Dominant set clustering and pooling for multi-view 3d object recognition," in *British Machine Vision Conference*, 06 2019. 1, 2, 3, 6, 7, 9

[30] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *AAAI*, vol. 33, 2019, pp. 3358–3565. 1, 2, 6, 7

[31] C. Esteves, Y. Xu, C. Allen-Blanchette, and K. Daniilidis, "Equivariant multi-view networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1, 2, 3, 6

[32] Z. Li, C. Xu, and B. Leng, "Angular triplet-center loss for multi-view 3d shape retrieval," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 8682–8689, Jul. 2019. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4890 1, 2

[33] B. Leng, C. Zhang, X. Zhou, C. Xu, and K. Xu, "Learning discriminative 3d shape representations by view discerning networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 10, pp. 2896–2909, 2019. 1, 2

[34] Z. Han, M. Shang, Z. Liu, C.-M. Vong, Y.-S. Liu, M. Zwicker, J. Han, and C. L. P. Chen, "Seqviews2seqlabels: Learning 3d global features via aggregating sequential views by rnn with attention," *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 658–672, 2019. 1, 2, 3, 6, 7

[35] Z. Han, H. Lu, Z. Liu, C.-M. Vong, Y.-S. Liu, M. Zwicker, J. Han, and C. L. P. Chen, "3d2seqviews: Aggregating sequential views for 3d global feature learning by cnn with hierarchical attention aggregation," *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 3986–3999, 2019. 1, 2, 3, 6, 7

[36] S. Chen, L. Zheng, Y. Zhang, Z. Sun, and K. Xu, "Veram: View-enhanced recurrent attention model for 3d shape classification," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 12, pp. 3244–3257, 2019. 1, 2, 3, 6, 7

[37] C. Ma, Y. Guo, J. Yang, and W. An, "Learning multi-view representation with lstm for 3-d shape recognition and retrieval," *IEEE Transactions on Multimedia*, vol. 21, no. 5, pp. 1169–1182, 2019. 1, 2, 6, 7

[38] Z. Zhang, H. Lin, X. Zhao, R. Ji, and Y. Gao, "Inductive multi-hypergraph learning and its application on view-based 3d object classification," *IEEE Transactions on Image Processing*, vol. 27, no. 12, pp. 5957–5968, 2018. 1, 2, 6, 7

[39] X. Wei, R. Yu, and J. Sun, "View-gcn: View-based graph convolutional network for 3d shape analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 4, 5, 6, 7, 11, 12

[40] A. Hamdi, S. Giancola, and B. Ghanem, "Mvtn: Multi-view transformation network for 3d shape recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 1–11. 1, 2, 3, 6

[41] X. Wei, R. Yu, and J. Sun, "Learning view-based graph convolutional network for multi-view 3d shape analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 06, pp. 7525–7541, jun 2023. 1, 2, 4, 6, 7

[42] Y. Gao, Y. Feng, S. Ji, and R. Ji, "Hgnn+: General hypergraph neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3181–3199, 2023. 1, 2, 4, 6, 7

[43] A. Hamdi, S. Giancola, and B. Ghanem, "Voint cloud: Multi-view point cloud representation for 3d understanding," in *International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=IpGgfpMucHj 1, 2, 3, 4, 6, 7

[44] L. Xu, Q. Cui, W. Xu, E. Chen, H. Tong, and Y. Tang, "Walk in views: Multi-view path aggregation graph network for 3d shape analysis," *Information Fusion*, vol. 103, p. 102131, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253523004475 1, 2, 4, 6, 7

[45] S. Chen, T. Yu, and P. Li, "Mvt: Multi-view vision transformer for 3d object recognition," in *British Machine Vision Conference*, 2021. 1, 2, 3, 6, 9

[46] Y. Xu, C. Zheng, R. Xu, Y. Quan, and H. Ling, "Multi-view 3d shape recognition via correspondence-aware deep learning," *IEEE Transactions on Image Processing*, vol. 30, pp. 5299–5312, 2021. 1, 2, 3, 6, 9

[47] T. Yu, J. Meng, and J. Yuan, "Multi-view harmonized bilinear network for 3d object recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2, 3, 6, 7, 9

[48] T. Yu, J. Meng, M. Yang, and J. Yuan, "3d object representation learning: A set-to-set matching perspective," *IEEE Transactions on Image Processing*, vol. 30, pp. 2168–2179, 2021. 1, 2, 3, 9

[49] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014. 1

[50] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 11 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735 1

[51] A. Kanezaki, Y. Matsushita, and Y. Nishida, "Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 3, 6, 7, 12

[52] Z. Yang and L. Wang, "Learning relationships for multi-view 3d object recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2, 3, 6, 7

[53] K. Sarkar, B. Hampiholi, K. Varanasi, and D. Stricker, "Learning 3d shapes as multi-layered height-maps using 2d convolutional networks," *ArXiv*, vol. abs/1807.08485, 2018. 2, 3

[54] D. Lin, Y. Li, Y. Cheng, S. Prasad, A. Guo, and Y. Cao, "Multi-range view aggregation network with vision transformer feature fusion for 3d object retrieval," *IEEE Transactions on Multimedia*, pp. 1–12, 2023. 2, 3, 7

[55] E. Johns, S. Leutenegger, and J. D. Andrew, "Pairwise decomposition of image sequences for active multi-view recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf 3, 5, 8

[57] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy 3, 4, 8

[58] M. Savva, F. Yu, H. Su, A. Kanezaki, T. Furuya, R. Ohbuchi, Z. Zhou, R. Yu, S. Bai, X. Bai, M. Aono, A. Tatsuma, S. Thermos, A. Axenopoulos, G. T. Papadopoulos, P. Daras, X. Deng, Z. Lian, B. Li, H. Johan, Y. Lu, and S. Mk, "Large-Scale 3D Shape Retrieval from ShapeNet Core55," in *Eurographics Workshop on 3D Object Retrieval*, I. Pratikakis, F. Dupont, and M. Ovsjanikov, Eds. The Eurographics Association, 2017. 4, 7, 8

[59] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 4, 9

[60] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf 4, 5

[61] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, "Generative and Discriminative Voxel Modeling with Convolutional Neural Networks," *arXiv e-prints*, p. arXiv:1608.04236, Aug. 2016. 6

[62] A. Goyal, H. Law, B. Liu, A. Newell, and J. Deng, "Revisiting point cloud shape classification with a simple and effective baseline," *International Conference on Machine Learning*, 2021. 6

[63] Z. Huang, Z. Zhao, H. Zhou, X. Zhao, and Y. Gao, "Deepccfv: Camera constraint-free multi-view convolutional neural network for 3d object retrieval," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. [Online]. Available: https://doi.org/10.1609/aaai.v33i01.33018505 6

[64] S. Ruder, "An overview of gradient descent optimization algorithms." 2016, cite arxiv:1609.04747Comment: Added derivations of AdaMax and Nadam. [Online]. Available: http://arxiv.org/abs/1609.04747 6

[65] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7 6

[66] N. Katsura, "Pytorch cosineannealing with warmup restarts," https://github.com/katsura-jp/pytorch-cosine-annealing-with-warmup, 2021. 6

[67] M. A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Revisiting point cloud classification: A new benchmark dataset and clas-

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2024.3381152

JOURNAL OF LATEX CLASS FILES, VOL. 14, NO. 8, AUGUST 2023   15

sification model on real-world data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 6, 7

[68] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 1817–1824. 6, 7

[69] X. Wei, Y. Gong, F. Wang, X. Sun, and J. Sun, "Learning canonical view representation for 3d shape recognition with arbitrary views," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 407–416. 6

[70] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "Spidercnn: Deep learning on point sets with parameterized convolutional filters," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 87–102. 6

[71] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/file/f5f8590cd58a54e94377e6ae2eded4d9-Paper.pdf 6

[72] Y. Cheng, R. Cai, X. Zhao, and K. Huang, "Convolutional fisher kernels for rgb-d object recognition," in *2015 International Conference on 3D Vision*, 2015, pp. 135–143. 6

[73] M. M. Rahman, Y. Tan, J. Xue, and K. Lu, "Rgb-d object recognition with multimodal deep convolutional neural networks," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, 2017, pp. 991–996. 6

[74] U. Asif, M. Bennamoun, and F. A. Sohel, "A multi-modal, discriminative and spatially invariant cnn for rgb-d object labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 9, pp. 2051–2065, 2018. 6

[75] S. Bai, X. Bai, Z. Zhou, Z. Zhang, and L. J. Latecki, "GIFT: A real-time and scalable 3d shape search engine," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 5023–5032. [Online]. Available: https://doi.org/10.1109/CVPR.2016.543 7

[76] 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. PP, pp. 1–1, 01 2017. 7

[77] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008. [Online]. Available: http://jmlr.org/papers/v9/vandermaaten08a.html 10

[78] M. Donoser and H. Bischof, "Diffusion processes for retrieval revisited," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1320–1327. 13

Peng Wang received his bachelor degree in Finance at International School of Business and Finance from Sun Yat-sen University, China, in 2018. He is currently working toward the master degree in the School of Information, Renmin University of China. His research interests include Computer Vision and Multi-Object Tracking.

Haoran Deng received his bachelor degree in Computer Science and Technology at School of Remote Sensing Information Engineering from Wuhan University, China, in 2022. He is currently working toward the master degree in the School of Information, Renmin University of China. His research interests include 3D Point Cloud Analysis and Simultaneous Localization and Mapping (SLAM).

Xudong Cai received his bachelor degree in Computer Science and Technology at School of Computer Science from North China Electric Power University, China, in 2021. He is currently working toward the Ph.D. degree in the School of Information, Renmin University of China. His research interests include Computer Vision and Visual Localization.

Deying Li received the MS degree in Mathematics from Huazhong Normal University (1988) and PhD degree in Computer Science from City University of Hong Kong (2004). She is currently a Professor in Department of Computer Science, Renmin University of China. Her research interests include wireless networks, mobile computing, social network and algorithm design and analysis.

Hongyu Sun obtained his bachelor degree in Computer Science and Technology at School of Computing from Inner Mongolia University, in 2017. He received his master degree in Computer Application Technology at School of Information from Renmin University of China, in 2020. He is currently working toward the Ph.D. degree in the School of Information, Renmin University of China. His research interests include 3D shape analysis and 3D point cloud understanding.

Yongcai Wang received his BS and PhD degrees from Department of Automation Sciences and Engineering, Tsinghua University in 2001 and 2006, respectively. He worked as associated researcher at NEC Labs, China from 2007-2009. He was a research scientist in Institute for Interdisciplinary Information Sciences (IIIS), Tsinghua University from 2009-2015. He was a visiting scholar at Cornell University in 2015. He is currently associate professor at Department of Computer Sciences, Renmin University of China. His research interests include perception and optimization algorithms in intelligent and networked systems.