SeqViews2SeqLabels: Learning 3D Global Features via Aggregating Sequential Views by RNN with Attention

Zhizhong Han, Mingyang Shang, Zhenbao Liu, *Member, IEEE,* Chi-Man Vong, *Senior Member, IEEE,* Yu-Shen Liu, *Member, IEEE,* Matthias Zwicker, *Member, IEEE,* Junwei Han, *Senior Member, IEEE,* C.L. Philip Chen, *Fellow, IEEE*

***Abstract*—Learning 3D global features by aggregating multiple views has been introduced as a successful strategy for 3D shape analysis. In recent deep learning models with end-to-end training, pooling is a widely adopted procedure for view aggregation. However, pooling merely retains the max or mean value over all views, which disregards the content information of almost all views and also the spatial information among the views. To resolve these issues, we propose Sequential Views To Sequential Labels (SeqViews2SeqLabels) as a novel deep learning model with an encoder-decoder structure based on Recurrent Neural Networks (RNNs) with attention. SeqViews2SeqLabels consists of two connected parts, an encoder-RNN followed by a decoder-RNN, that aim to learn the global features by aggregating sequential views and then performing shape classification from the learned global features, respectively. Specifically, the encoder-RNN learns the global features by simultaneously encoding the spatial and content information of sequential views, which captures the semantics of the view sequence. With the proposed prediction of sequential labels, the decoder-RNN performs more accurate classification using the learned global features by predicting sequential labels step-by-step. Learning to predict sequential labels provides more and finer discriminative information among shape classes to learn, which alleviates the overfitting problem inherent in training using a limited number of 3D shapes. Moreover, we introduce an attention mechanism to further improve the discriminative ability of SeqViews2SeqLabels. This mechanism increases the weight of views that are distinctive to each shape class, and it dramatically reduces the effect of selecting the first view position. Shape classification and retrieval results under three large-scale benchmarks verify that SeqViews2SeqLabels learns more discriminative global features by more effectively aggregating sequential views than state-of-the-art methods.**

***Index Terms*—3D feature learning, Sequential views, Sequential labels, View aggregation, RNN, Attention.**

I. Introduction

THE 2D views taken around 3D shapes have been shown to be effective for learning 3D global features for 3D shape analysis, such as 3D shape classification and retrieval [1]–[8]. View-based methods understand a 3D shape by learning its global feature via aggregating the multiple views taken around it. Due to their independence of 3D geometry processing, view-based methods are capable of understanding both manifold and non-manifold 3D shapes. More importantly, this advantage also alleviates the difficulty of learning features directly from irregular 3D shapes (i.e., arbitrary vertex resolution, irregular vertex topology and orientation ambiguity on 3D surface) [9]–[12], especially for deep learning models [9], [10], [12]. Therefore, how to aggregate multiple views for 3D feature learning has become an important research topic in 3D shape analysis and understanding.

Recently, deep learning models have been very successful at learning 3D features by aggregating the information of multiple views. To perform end-to-end optimization in deep learning models, max pooling or mean pooling [3], [4], [6]–[8], [13] is always used to aggregate the content information of multiple views into global features. Although pooling can make global features invariant to the rotation of 3D shapes to a certain extent, it was designed as a procedure of information abstraction in deep learning models, and it inevitably loses the content information of almost all views and the spatial information among the views. Thus, it remains a research challenge to learn 3D global features by more effectively aggregating the content and spatial information of multiple views using deep learning.

To tackle this challenge, we propose *Sequential Views To Sequential Labels* (SeqViews2SeqLabels), a novel deep learning model that learns 3D global features by simultaneously aggregating the content and spatial information of multiple views of a 3D shape. To enhance the discriminability of learned features via efficiently using the spatial information among views, multiple views are taken from a circle surrounding the 3D shapes. This forms the sequential views to be learned from in our work. SeqViews2SeqLabels forms an encoder-decoder structure based on Recurrent Neural Network (RNN) [14]. Specifically, an encoder-RNN learns the global feature of a 3D shape by simultaneously aggregating the content information of all sequential views and the spatial information among them. In this way, the semantics of the view sequence, which is robust to the first view position, can

Z. Han is with the Tsinghua University, P. R. China and the University of Maryland, College Park, USA (email: h312h@mail.nwpu.edu.cn).

Z. Liu, J. Han are with the Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P. R. China (email: liuzhenbao, jhan@nwpu.edu.cn).

Yu-Shen Liu and Mingyang Shang are with the Tsinghua University, Beijing 100084, P. R. China (liuyushen@tsinghua.edu.cn, s-my16@mails.tsinghua.edu.cn). Yu-Shen Liu is the corresponding author.

Matthias Zwicker is with the University of Maryland, College Park, 20737, USA (email: zwicker@cs.umd.edu).

C. L. P. Chen and C.M. Vong are with University of Macau, Macau 99999, China (email: philip.chen@ieee.org, cmvong@umac.mo).

This work was supported by National Key R&D Program of China (2018YFB0505400), in part by the National Natural Science Foundation of China under Grant 61472202, 61672430, Swiss National Science Foundation project nr. 169151, MYRG2018-00138-FST, MYRG2016-00134-FST, FDCT/273/2017/A, and NWPU Basic Research Fund under Grant 3102018jcc001.

be learned. Subsequently, an decoder-RNN maps the learned feature into sequential labels, which are also organized in a sequential manner for shape classification. The learning of sequential label prediction is proposed to present more and finer discriminative information among different shape classes for the decoder-RNN to capture. This alleviates the overfitting problem inherent in training using a limited number of 3D shapes. Moreover, the decoder-RNN also introduces an attention mechanism to further increase the discriminative ability of SeqViews2SeqLabels. The attention mechanism adaptively learns to weigh the content information of sequential views to predict each sequential label. The attention weight highlights the views that are distinctive to the shape class indicated by a sequential label and suppresses other views. This assists the encoder-RNN to learn the semantic meaning of the view sequence and dramatically reduces the effect of choosing the first view position. In summary, our main contributions are as follows:

i) We propose SeqViews2SeqLabels as a novel deep learning model for 3D global feature learning by more effectively aggregating sequential views, preserving the content information of all sequential views and the spatial information among the views.
ii) To the best of our knowledge, SeqViews2SeqLabels is the first fully RNN-based 3D global feature learning method based on aggregating multiple views, which verifies the usefulness of RNN for 3D global feature learning.
iii) We propose to perform shape classification by predicting sequential labels in a step-by-step way, where the task of predicting sequential labels provides more and finer discriminative information among the shape classes to learn. This alleviates the overfitting problem inherent in training using a limited number of 3D shapes.
iv) We propose an attention mechanism to further increase the discriminative ability of SeqViews2SeqLabels by increasing the weight of distinctive views for each shape class. This also assists the encoder-RNN to learn the semantic meaning of the view sequence and it dramatically reduces the effect of choosing the first view position.

This paper is organized as follows: We review the related work in Section II, and present the details of SeqViews2SeqLabels in Section III. We describe our experimental setup and results in Section IV and Section V, respectively. Finally, we draw conclusions in Section VI.

II. Related work

In this section, the methods of learning 3D features by deep learning models are reviewed. These methods are categorized in terms of different raw 3D representations that are learned from, including meshes, voxels and views. In addition, the existing view aggregation procedures are emphasized in the reviewed methods, which highlights the novelty of our RNN-based view aggregation employed in SeqViews2SeqLabels. Finally, we also review the methods with similar structure of SeqViews2SeqLabels in other applications.

A. Mesh-based methods

3D mesh is an important raw representation for 3D shapes. A 3D mesh is composed of vertices which are connected by edges. To learn features from 3D meshes directly, several deep learning models have been proposed. Han et al. [9] proposed circle convolutional restricted boltzmann machine to learn 3D local features based on a novel circle convolution in an unsupervised way. To learn global features via hierarchically abstracting from local information, Han et al. [10] further proposed mesh convolutional restricted boltzmann machine, which simultaneously encodes the geometry of local regions and the spatiality among them. With heat diffusion based descriptor, Jin et at. [15] proposed DeepShape to learn 3D global features. Similarly, Jonathan et al. [16] learned 3D features from hand-crafted features on 3D surface by a novel geodesic convolutional neural network. To explore the feasibility of learning features in spectral domain, Davide et al. [17] proposed localized spectral convolutional network to perform supervised local feature learning. Also in the spectral domain, Jin et al. [18] learned binary spectral shape descriptor for 3D shape correspondence. By encoding the spatial relationships among virtual words on 3D meshes, Han et al. proposed deep spatiality [19] to simultaneously learn 3D global and local features with novel coupled softmax. However, these methods can only be used to learn features from smooth manifold meshes.

B. Voxel-based methods

Voxel-based methods learn 3D features from voxels which represent 3D shapes by the distribution of corresponding binary variables. Wu et al. [20] proposed 3D ShapeNets to learn global features from voxelized 3D shapes based on convolutional restricted boltzmann machine. Sharma et al. [21] employed fully convolutional denoising autoencoder to robustly perform unsupervised global feature learning via decomposing and reconstructing voxelized 3D shapes. Girdhar et al. [22] combined voxels and the corresponding images to learn global features by a novel T-L network based on CNN. To employ the generative adversarial training manner, Wu et al. [23] learned 3D global features by a novel 3DGAN which is composed of a generator and a discriminator. By analysing the reason why the performance of voxel-based methods are always not as good as view-based methods, Qi et al. [13] employed CNN to learn global features from novel voxel representations, where max pooling is used to aggregate the information captured from different orientations. To speed up the learning from voxels by deep learning models, Wang et al. [24] proposed O-CNN to learn global features based on a novel octree data structure. To learn local features from voxles, Han et al. [12] proposed a novel voxelization permutation strategy to eliminate the effect of rotation and orientation ambiguity on the 3D surface. Although voxel-based methods have the advantage of generating 3D shapes, they not only need heavy computational cost but also require 3D shapes to be aligned. In addition, this kind of methods always perform discriminating shapes worse than the following view-based methods.

C. View-based methods

Light Field Descriptor (LFD) [25] is the pioneer view-based 3D descriptor, which employs features of 2D silhouettes in multiple views of 3D shapes. Instead of aggregating multi-view information into global features, LFD evaluates the dissimilarity between two shapes via comparing 2D features of their corresponding two view sets in a greedy way. By the same strategy, GIFT [5] measures the difference between two shapes by the Hausdorff distance between their corresponding view sets. To bridge 2D sketches and 3D shapes for shape retrieval, barycentric representations of 3D shapes were proposed to be learned from multiple views [26].

DeepPano [6] was proposed to learn features from PANORAMA views using CNN, where a PANORAMA view can be regarded as the seamless aggregation of multiple views captured on a circle. To eliminate the effect of rotation about the up-oriented direction, row-wise max pooling was introduced in DeepPano. With pose normalization, Sfikas et al. [27] used CNN to learn 3D features from multiple PANORAMA views which were stacked together in a consistent order. Similarly, using another hand-crafted feature, geometry image, Sinha et al. [28] proposed to learn 3D features from geometry images. In addition, RotationNet [29] is proposed to learn global features by treating pose labels as latent variables which are optimized to self-align in an unsupervised manner.

Recently, Su et al. [3] proposed Multi-View CNN to learn 3D global features from multiple views. To describe a 3D shape by multiple views, the content information within multiple views is aggregated into the global feature through max pooling. Similarly, max pooling is also employed to aggregate multiple views to learn local features for shape segmentation or correspondence [4]. To employ more content information in each view, Li et al. [30] concatenated all view features for hierarchical abstraction in the CNN-based model. By decomposing a view sequence into a set of view pairs, Johns et al. [31] classified each view pair independently, and then, learned an object classifier by weighting the contribution of each view pair, which allowed 3D shape recognition over arbitrary camera trajectories. To perform pooling more efficiently, Wang et al. [8] proposed dominant set clustering to cluster views token form each shape, where pooling is performed in each cluster.

Although pooling resolves the effect of rotation of 3D shapes, it still suffers from two kinds of information loss, i.e., the content information of almost all views and the spatial information among the views. The spatial information between pairwise views is also disregarded by the view pair decomposition [31]. In [30], Li et al. compensated these two kinds of loss by concatenation of all views, however, it is sensitive to the first view position.

To resolve the aforementioned issues, SeqViews2SeqLabels is proposed to learn 3D features via aggregating sequential views by RNN. The RNN-based aggregation not only preserves the content information of all views and the spatial information among the views, but also becomes capable of learning the semantics of view sequence, which is robust to the first view position.

D. CNN-RNN based and RNN-RNN based models

SeqViews2SeqLabels is similar to CNN-RNN based and RNN-RNN based models. Different from multiple views, Miyagi et al. [32] employed multiple voxel slices to learn 3D global features. They used CNN to extract the feature of each voxel slice, and then, used RNN for view aggregation, where a softmax was employed to conduct 3D shape classification. Using a two-layer RNN, Truc et al. [33] proposed a CNN-RNN model to segment 3D shapes, where multiple edge images were predicted to estimate the different parts on a 3D shape. In addition, RNN-RNN based models, especially seq2seq models, were originally proposed for text understanding. Due to their powerful learning ability, they have been successfully employed for image and speech understanding, such as scene text recognition [34], [35], image caption generation [36] and speech recognition [37]. The models in [34]–[36] were proposed to recognize what are in a single image. For example, [34], [35] focused on how to recognize the characters in an image, [36] focused on how to recognize the concepts in an image. Different from their tasks, SeqViews2SeqLabels recognize what a view sequence of multiple views is. This difference makes the involved attention play different roles. In our method, we want to use attention to highlight the views with distinctive characteristics to each shape class and depress the views with ambiguous appearance. Thus, our attention weights are computed at the image level. In the methods of [35], [36], attention is used to highlight the parts with a specified meaning in an image, although multiple feature maps are involved. Thus, their attention weights are computed at the part level. To represent the characteristics of each shape class at each step of decoder, we propose a novel attention mechanism which is different from the one employed in [35], [37].

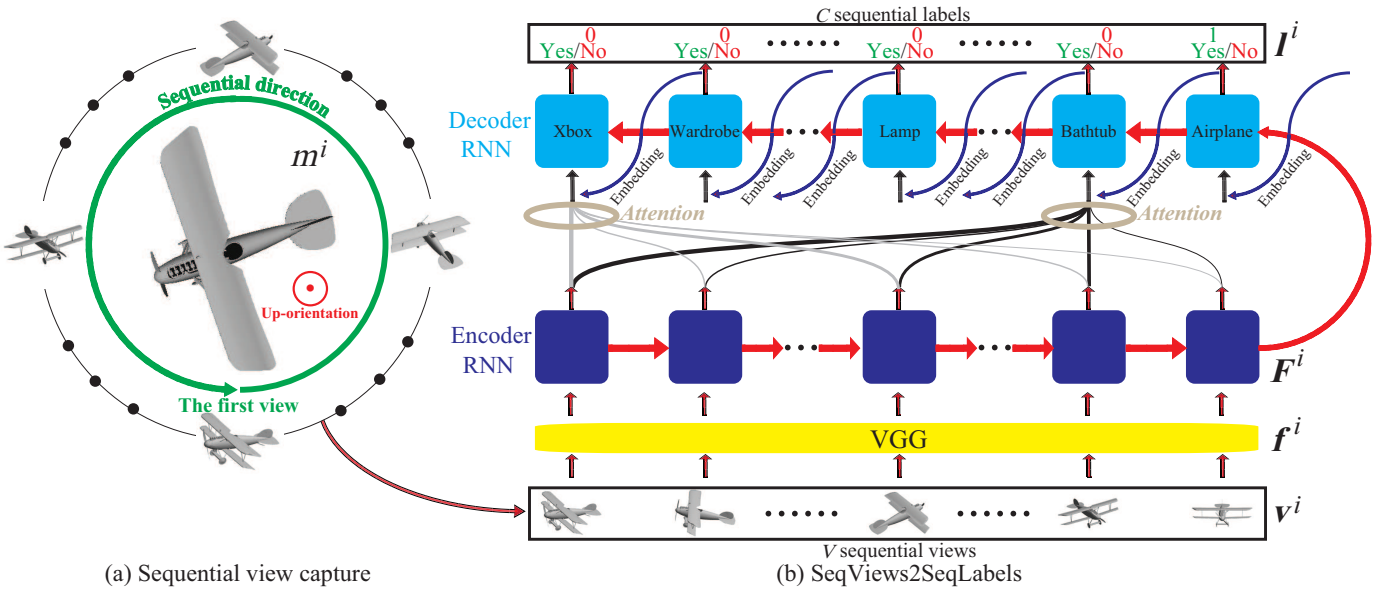
III. SeqViews2SeqLabels

In this section, SeqViews2SeqLabels is introduced in detail. First, we provide an overview and then describe the key elements, including capturing sequential views, view feature extraction, the encoder-RNN, the decoder-RNN, and the attention mechanism in the subsequent five subsections.

A. Overview

The framework of SeqViews2SeqLabels is illustrated in Fig. 1, where SeqViews2SeqLabels consists of the encoder-RNN and the decoder-RNN as shown in Fig. 1 (b). First, a *view sequence* $\boldsymbol{v}^i$ is captured on a circle around each 3D shape m^i in a set of M 3D shapes, where $i \in [1, M]$, as shown in Fig. 1 (a). The view sequence $\boldsymbol{v}^i$ is composed of V sequential views v_j^i, such that $\boldsymbol{v}^i = [v_1^i, ..., v_j^i]$ and $j \in [1, V]$. Then, *the global feature* of m^i, namely $\boldsymbol{F}^i$, is learned from $\boldsymbol{v}^i$ by the encoder-RNN. Finally, the decoder-RNN classifies m^i into one of C shape classes based on the global feature $\boldsymbol{F}^i$ learned by the encoder-RNN.

To learn $\boldsymbol{F}^i$, the encoder-RNN not only aggregates the content information of each single view v_j^i in $\boldsymbol{v}^i$, but also preserves the spatial information between successive views, such as v_j^i and v_{j+1}^i. This enables the learning of semantics



(a) Sequential view capture (b) SeqViews2SeqLabels

Fig. 1. The framework of SeqViews2SeqLabels. The sequential views are first captured around each up-oriented 3D shapes on a circle in (a). Then, they are learned by SeqViews2SeqLabels which consists of encoder-RNN and decoder-RNN.

of the view sequence $\boldsymbol{v}^i$, which makes SeqViews2SeqLabels robust to the first view position with the assistance of the attention mechanism introduced later. The content information of v_j^i is described by its low-level feature $\boldsymbol{f}_j^i$, which is extracted by the fined-tuned VGG19 [38] deep neural network.

In addition, the decoder-RNN classifies shape m^i into one of C shape classes by predicting a *label sequence* $\boldsymbol{l}^i$ based on $\boldsymbol{F}^i$ learned by the encoder-RNN. The label sequence $\boldsymbol{l}^i$ is composed of C *sequential labels* l_c^i, such that $\boldsymbol{l}^i = [l_1^i, ..., l_c^i]$, where $c \in [1, C]$, $l_c^i \in \{0, 1\}$ and $\sum_{c=1}^{C} \boldsymbol{l}^i = 1$. $l_c^i = 1$ indicates the positive prediction of the c-th label for m^i, which means m^i is classified into the c-th shape class, while $l_c^i = 0$ indicates the negative prediction of the c-th label for m^i.

We employ sequential labels in $\boldsymbol{l}^i$ to provide more and finer discriminative information among different shape classes. Sequential labels change the traditional classification task of learning a mapping from a sequence (sequential views) to a scalar (shape class index) to an extended mapping of learning a mapping from a sequence (sequential views) to another sequence (sequential labels). This extended mapping effectively alleviates the overfitting problem inherent in training under a limited number of 3D shapes. The prediction of l_c^i is only conducted at the c-th step of the decoder-RNN. The prediction of sequential labels in a step-by-step manner enables to comprehensively refer to view aggregation at each step of the encoder-RNN, the characteristics of forward (from the 1-th to the $(c-1)$-th) shape classes, the characteristics of backward (from the $(c+1)$-th to the C-th) shape classes, and the label prediction l_{c-1}^i at the previous $(c-1)$-th step. Note that the order of shape classes to be predicted in the decoder-RNN does not affect the discriminative ability of SeqViews2SeqLables, because the prediction of each sequential label is always conducted based on the characteristics of all shape classes.

More importantly, we also introduce an attention mechanism to further increase the discriminative ability of SeqViews2SeqLabels for higher classification accuracy than merely using the encoder-decoder structure. The attention mechanism is implemented by weighting the low-level feature $\boldsymbol{f}_j^i$ of all sequential views for each shape class. That is, the views that are distinctive to one shape class are emphasized, and otherwise the views are suppressed. This ability of observing all views for each sequential label prediction also assists the encoder-RNN to learn the semantic meaning of the view sequence by dramatically reducing the effect of choosing the first view position.

B. Capturing sequential views

The sequential views are captured around each 3D shape on a circle, which forms a view sequence, as shown in Fig. 1 (a). The sequential views are formed by V views in order which are uniformly distributed on the circle. Here, the cameras are elevated $30°$ from the ground plane, pointing to the centroid of the 3D shape. The first view in the view sequence is taken from a fixed position that can be randomly selected on the circle. Then, the subsequent views are taken with an angle interval of $360°/V$ in a consistent sequential direction. The sequential direction is determined by the right hand rule, that is, it is along the direction of wrapping one's right hand when the thumb is in the same direction of the up-orientation, as demonstrated by the green arrow surrounding the 3D shape in Fig. 1 (a).

Different from traditional multiple view capture [5], [25], the sequential views are captured on a circle rather than a unit sphere. Although the sequential views cannot fully cover the top or the bottom of 3D shapes, the low-level features of sequential views can be more efficiently aggregated while preserving the spatial information among the views for 3D global feature learning.

C. Low-level view feature extraction

The low-level feature of each single view can be extracted through fine-tuning existing deep neural networks, such as VGG19 [38] and Alexnet [39]. In our work, we employ VGG19 to extract the low-level feature $\boldsymbol{f}_j^i$ of each single view v_j^i in $\boldsymbol{v}^i$, since VGG19 and its pre-trained parameters are easy to access. VGG19 is originally trained under the ImageNet benchmark for large scale image classification [38].

VGG19 is formed by 19 weight layers including 16 convolutional layers and 3 fully connected layers. With a softmax layer, VGG19 is capable of classifying images belonging to 1000 categories. In our work, the VGG19 pre-trained under ImageNet is fine-tuned by all sequential views of 3D shapes in the training set, where each view is classified into one of C shape classes by another softmax layer. When a view v_j^i is forwarded through the fine-tuned VGG19, its low-level feature $\boldsymbol{f}_j^i$ is extracted as a 4096 dimensional vector from the last fully connected layer of the VGG19.

D. Encoder-RNN

To benefit from the powerful ability of learning sequential data, SeqViews2SeqLabels employs an RNN as the encoding procedure to learn 3D global feature $\boldsymbol{F}^i$. The encoder-RNN learns $\boldsymbol{F}^i$ via aggregating $\boldsymbol{f}_j^i$ of all sequential views v_j^i in the view sequence $\boldsymbol{v}^i$ while preserving the spatial information among them.

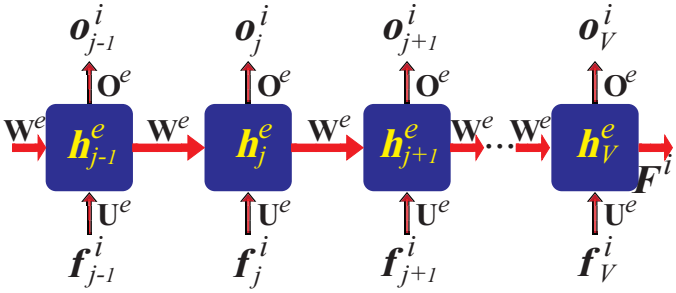


Fig. 2. The structure of the encoder-RNN aggregates low-level features of views while preserving the spatial information among them.

The general structure of the encoder-RNN for aggregating sequential views in $\boldsymbol{v}^i$ is illustrated in Fig. 2, where the RNN cell shown as a blue square at each step can be a Long Short Term Memory (LSTM) [14] or Gated Recurrent Unit (GRU) [40]. The encoder-RNN learns from the sequential views v_j^i in $\boldsymbol{v}^i$ step-by-step, where all low-level features of v_j^i, i.e., $[\boldsymbol{f}_1^i, ..., \boldsymbol{f}_j^i, ..., \boldsymbol{f}_V^i]$, are sequentially aggregated while preserving the spatial information among them.

An $\boldsymbol{f}_j^i$ is conveyed to the encoder-RNN as the input at the j-th step. At the j-th step, a *hidden state* $\boldsymbol{h}_j^e$ plays the role of "memory" of the encoder-RNN, where the superscript, e, is the abbreviation of the encoder. This is because $\boldsymbol{h}_j^e$ is calculated based on the hidden state $\boldsymbol{h}_{j-1}^e$ at the previous $j-1$-th step and the input $\boldsymbol{f}_j^i$ at the current step j, as defined in Eq. (1),

$$\boldsymbol{h}_j^e = ReLU(\mathbf{U}^e \boldsymbol{f}_j^i + \mathbf{W}^e \boldsymbol{h}_{j-1}^e), \tag{1}$$

where $ReLU(\cdot)$ is a non-linear function defined as $max(0, \cdot)$, $\mathbf{U}^e$ and $\mathbf{W}^e$ are learnable parameters. $\boldsymbol{h}_0^e$ required to calculate $\boldsymbol{h}_1^e$ is initialized to all zeros.

In addition, an output is obtained at each step of the encoder-RNN. The *output* at the j-th step, $\boldsymbol{o}_j^e$, is provided to the decoder-RNN for the prediction of sequential labels, and $\boldsymbol{o}_j^e$ can be calculated as in Eq. (2),

$$\boldsymbol{o}_j^e = \mathbf{O}^e \boldsymbol{h}_j^e + \boldsymbol{b}^e. \tag{2}$$

where $\mathbf{O}^e$ and $\boldsymbol{b}^e$ are learnable weight parameters. Moreover, *the hidden state at the last step*, $\boldsymbol{h}_V^e$, describes m^i as its global feature $\boldsymbol{F}^i$ after aggregating all sequential views in $\boldsymbol{v}^i$, such that $\boldsymbol{F}^i = \boldsymbol{h}_V^e$.

E. Decoder-RNN

Overview. Similar to the encoder-RNN, the decoder-RNN is also implemented by an RNN, which leads to the encoder-decoder structure of SeqViews2SeqLabels. According to the global feature $\boldsymbol{F}^i$ of m^i provided by the encoder-RNN, the decoder-RNN aims to classify m^i into one of C shape classes by predicting the sequential labels l_c^i in $\boldsymbol{l}^i$ step by step, as shown in Fig. 1 (b).

Based on sequential labels, the decoder-RNN regards the shape classification as finding a mapping from a view sequence $\boldsymbol{v}^i$ to a label sequence $\boldsymbol{l}^i$, which is different from the traditional mapping from $\boldsymbol{v}^i$ to a shape class index. This facilities the decoder-RNN to learn from more and finer discriminative information among different shape classes, which effectively alleviates overfitting inherent in training a powerful RNN-based model under a limited number of 3D shapes.

The decoder-RNN predicts one label l_c^i in $\boldsymbol{l}_c^i$ at each c-th step. The prediction of l_c^i indicates whether the shape m^i belongs to the c-th shape class. The positive prediction ($l_c^i = 1$) indicates that m^i belongs to the c-th shape class. Otherwise, the negative prediction ($l_c^i = 0$) is provided.

Structure. The details of the decoder-RNN are briefly illustrated in Fig. 3, where only two steps for predicting the sequential labels of "Airplane" and "Bathtub" shape classes are demonstrated. Generally, each sequential label l_c^i is predicted according to several aspects of information, such as the view-level information ($\boldsymbol{g}_c$) combined by the attention mechanism at the current step, the information ($\boldsymbol{k}_{c-1}$) of the sequential label predicted at the previous step, the characteristics ($\boldsymbol{h}_{c-1}^d$) of forward shape classes, and the characteristics of backward shape classes.

The hidden state at the current step. For the prediction of label l_c^i at the c-th step, *the hidden state* at the c-th step, $\boldsymbol{h}_c^d$, is first computed, where the superscript, d, is the abbreviation of the decoder. To compute $\boldsymbol{h}_c^d$, the hidden state at the previous step $\boldsymbol{h}_{c-1}^d$ and the embedding $\boldsymbol{k}_{c-1}$ of label l_{c-1}^i predicted at the previous step are employed. $\boldsymbol{h}_{c-1}^d$ comprehensively encodes the characteristics of forward shape classes, while $\boldsymbol{k}_{c-1}$ especially highlights the label prediction at the previous step, as defined as follows,

$$\boldsymbol{h}_c^d = ReLU(\mathbf{U}^d \boldsymbol{k}_{c-1} + \mathbf{W}^d \boldsymbol{h}_{c-1}^d). \tag{3}$$

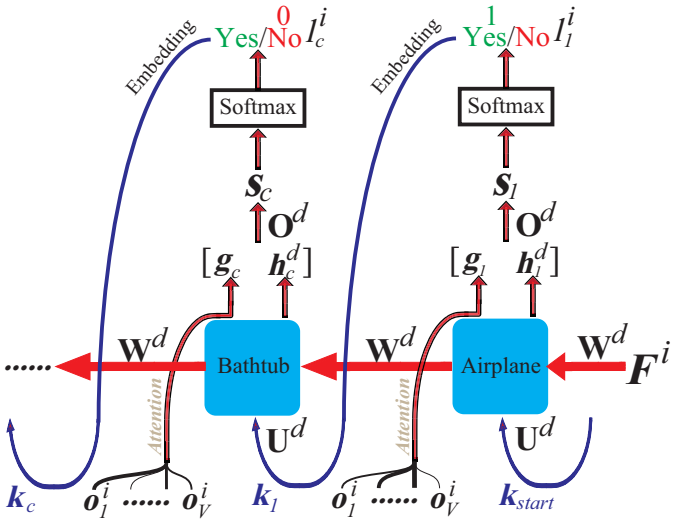


Fig. 3. We illustrate the structure of the decoder-RNN, where only two steps of the decoder-RNN for predicting the sequential labels of "Airplane" and "Bathtub" shape classes are shown.

where $\mathbf{U}^d$ and $\mathbf{W}^d$ are learnable parameters.

Note that, for the prediction of the first label l_1^i, the hidden state at the previous step is replaced by the global feature $\boldsymbol{F}^i$ of m^i. Moreover, a special embedding, $\boldsymbol{k}_{start}$, is employed to indicate the start of the prediction of sequential labels. Then, Eq. (3) is rewritten to calculate the hidden state at the first step $\boldsymbol{h}_1^d$, as defined below,

$$\boldsymbol{h}_1^d = ReLU(\mathbf{U}^d \boldsymbol{k}_{start} + \mathbf{W}^d \boldsymbol{F}^i). \tag{4}$$

The prediction vector. The prediction of sequential label l_c^i also considers the view-level information at each step of the encoder-RNN, which is represented by the *attention vector* $\boldsymbol{g}_c$. $\boldsymbol{g}_c$ is obtained through the attention mechanism as detailed in the following subsection. We expect sequential label l_c^i can be predicted through simultaneously observing the view-level information and the class-level information. Therefore, the prediction of l_c^i is carried out based on a *prediction vector* $\boldsymbol{s}_c$ that is formed by the concatenation of the attention vector $\boldsymbol{g}_c$ and the characteristics of forward shape classes $\boldsymbol{h}_c^d$, as defined below,

$$\boldsymbol{s}_c = \mathbf{O}^d[\boldsymbol{g}_c \ \boldsymbol{h}_c^d] + \boldsymbol{b}^d, \tag{5}$$

where $\mathbf{O}^d$ and $\boldsymbol{b}^d$ are learnable parameters. To represent the characteristic of each shape class at each step, the view-level information is not directly involved in producing the class-level information as in other methods [35], [37]. This design makes the decoder-RNN learn the distribution of sequential labels mainly based on the characteristics of shape classes.

Similar to rewriting Eq. (3) as Eq. (4), Eq. (5) can be rewrittern as Eq. (6) for the prediction of the first label l_1^i,

$$\boldsymbol{s}_1 = \mathbf{O}^d[\boldsymbol{g}_1 \ \boldsymbol{h}_1^d] + \boldsymbol{b}^d. \tag{6}$$

Sequential label prediction. In our scenario, the c-th label l_c^i can only be predicted at the c-th step of the decoder-RNN. Thus, the sum of probabilities over both positive and negative predictions of l_c^i is supposed to be one, where l_c^i only equals to either one or zero to indicate whether shape m^i belongs to the c-th shape class. Thus, the probability of positive prediction of l_c^i can be obviously computed by a sigmoid function according to the prediction vector $\boldsymbol{s}_c$, while the probability of negative prediction of l_c^i is the supplementary. However, $\boldsymbol{s}_c$ merely considers the characteristics of forward shape classes, which means the sigmoid function can not observe the characteristics of backward shape classes. As a result, there is a loss of discriminative information among shape classes when predicting sequential labels, resulting in low classification accuracy.

To resolve this issue, the characteristics of all shape classes are comprehensively considered when predicting each label at each step by a softmax layer, as shown in Fig. 3. The softmax layer captures more discriminative information among different shape classes via minimizing the probabilities that a shape belongs to wrong shape classes and maximizing the probabilities that it belongs to the correct shape class in the training procedure. More importantly, the softmax layer also efficiently employs the characteristics of backward shape classes, which overcomes the disadvantage that only the characteristics of forward shape classes are encoded as the hidden state $\boldsymbol{h}_{c-1}^d$ for the c-th label prediction.

Specifically, the softmax layer regards the positive and negative label predictions of each shape class as two independent categories, that is, the sum of probabilities over both positive label prediction and negative label prediction is not guaranteed to be one. Thus, there are totally $2C$ categories for the softmax layer to classify at each step of predicting sequential labels. With the softmax layer, the probabilities of positive and negative predictions of l_c^i are respectively defined based on the prediction vector $\boldsymbol{s}_c$ as below,

$$\boldsymbol{y}_c = \mathbf{W}\boldsymbol{s}_c + \boldsymbol{b}, \tag{7}$$

$$p(l_c^i = 1 | [l_1^i, l_2^i, ..., l_{c-1}^i], \boldsymbol{v}^i) = \frac{exp(y_c^1)}{\sum\limits_{a\in[1,C]} \sum\limits_{b\in\{0,1\}} exp(y_a^b)}, \tag{8}$$

$$p(l_c^i = 0 | [l_1^i, l_2^i, ..., l_{c-1}^i], \boldsymbol{v}^i) = \frac{exp(y_c^0)}{\sum\limits_{a\in[1,C]} \sum\limits_{b\in\{0,1\}} exp(y_a^b)}, \tag{9}$$

where $\boldsymbol{y}_c = [y_1^0, y_1^1, ..., y_a^b, ..., y_C^0, y_C^1]$, $a \in [1, C]$, and $b \in \{0, 1\}$, $\mathbf{W}$ and $\boldsymbol{b}$ are learnable parameters in the softmax layer. Finally, a joint probability is defined over the sequential labels l_c^i in $\boldsymbol{l}^i$ by sequentially conditional probabilities as follows,

$$p(\boldsymbol{l}^i) = \prod_{c\in[1,C]} p(l_c^i | [l_1^i, l_2^i, ..., l_{c-1}^i], \boldsymbol{v}^i), \tag{10}$$

where the label sequence $\boldsymbol{l}^i = [l_1^i, ..., l_c^i]$ is for the i-th shape m^i. It means to evaluate the probability of $\boldsymbol{l}^i$ to be inferred based on the given view sequence $\boldsymbol{v}^i$.

Objective function. Based on Eq. (10), we want the decoder-RNN to predict the sequential labels as accurately as possible. Thus, the objective function of SeqViews2SeqLabels is to

maximize the log-likelihood of joint probabilities of predicting l_c^i in $\boldsymbol{l}^i$ for all M shapes in the training set, as defined below,

$$\max \frac{1}{M} \sum_{i \in [1,M]} \log p(\boldsymbol{l}^i). \tag{11}$$

A testing shape is classified according to $p(l_c^i = 1|[l_1^i, l_2^i, ..., l_{c-1}^i], \boldsymbol{v}^i)$, such that the shape class that the testing shape belongs to is determined by $argmax_c \ p(l_c^i = 1|[l_1^i, l_2^i, ..., l_{c-1}^i], \boldsymbol{v}^i)$. In the next subsection, the attention mechanism is introduced in detail, which describes how to compute the attention vector $\boldsymbol{g}_c$.

F. Attention mechanism

For each sequential label prediction, the attention mechanism determines which views should be paid more attention for more accurate prediction of sequential labels.

To predict the c-th label l_c^i for shape m^i, the degree of attention paid to the j-th view v_j^i is measured by *the attention weight* $\alpha_{c,j}^i$, where all attention weights form an *attention weight vector* $\boldsymbol{\alpha}_c^i = [\alpha_{c,1}^i, ..., \alpha_{c,j}^i]$, $j \in [1, V]$ and $\sum_{j=1}^V \alpha_{c,j}^i = 1$. If $l_c^i = 1$ is finally predicted, a higher value of $\alpha_{c,j}^i$ means that the shape appearance in the j-th view v_j^i is more distinctive to the characteristics of the c-th shape class, and maybe, no other views are needed for the positive prediction of l_c^i. Otherwise, if $l_c^i = 0$ is finally predicted, a higher value of $\alpha_{c,j}^i$ means that the shape appearance in the j-th view v_j^i is more different from the characteristics of the c-th shape class, and maybe, no other views are needed for the negative prediction of l_c^i. Thus, the attention vector $\boldsymbol{g}_c$ is computed via weighting the output $\boldsymbol{o}_j^i$ at each step of the encoder-RNN by the attention weights $\alpha_{c,j}^i$, defined as,

$$\boldsymbol{g}_c = \sum_{j=1}^{V} \alpha_{c,j}^i \boldsymbol{o}_j^i. \tag{12}$$

Inspired by the attention mechanism for machine translation [41], we compute the attention weight $\alpha_{c,j}^i$ in a similar way. $\alpha_{c,j}^i$ measures the similarity between the view v_j^i and the c-th shape class, which indicates the distinctiveness of v_j^i to the c-th shape class. Different from the attention involved in [35], [37], the attention weights are computed according to the hidden state at the current step rather than the hidden state at the previous step. Therefore, the computation of $\alpha_{c,j}^i$ is implemented by a single-layer neural network involving $\boldsymbol{o}_j^i$ and $\boldsymbol{h}_c^d$, as defined as follows,

$$\beta_{c,j}^i = \boldsymbol{x}^{\mathrm{T}} tanh(\mathbf{Y}\boldsymbol{o}_j^i + \mathbf{Z}\boldsymbol{h}_c^d), \tag{13}$$

$$\alpha_{c,j}^i = \frac{exp(\beta_{c,j}^i)}{\sum_{q=1}^V exp(\beta_{c,q}^i)}, \tag{14}$$

where the vector $\boldsymbol{x}$, and the matrices $\mathbf{Y}$ and $\mathbf{Z}$ are learnable parameters of SeqViews2SeqLabels for learning the attention weight vector $\boldsymbol{\alpha}_c^i$. These parameters are optimized along with other parameters involved in SeqViews2SeqLabels in the learning procedure via maximizing Eq. (11).

IV. Experimental setup

In this section, different shape benchmarks and performance measures for global shape classification and retrieval are respectively described to evaluate the 3D global features learned by SeqViews2SeqLabels. In addition, the setup of parameters involved in SeqViews2SeqLabels is also discussed.

A. Benchmarks and evaluations

The global shape classification and retrieval experiments are conducted under three large-scale 3D shape benchmarks, including ModelNet40 [20], ModelNet10 [20] and ShapeNetvCore55 [42].

ModelNet40 and ModelNet10 are two subsets of ModelNet which contains 151,128 3D shapes categorized into 660 shape classes. As smaller subsets, ModelNet40 is formed by 40 shape classes with a total of 12,311 3D shapes, while ModelNet10 consists of 4,899 3D shapes split into 10 shape classes. The training and testing sets of ModelNet40 consist of 9,843 and 2,468 shapes, respectively. In addition, the training and testing sets of ModelNet10 consist of 3,991 and 908 shapes, respectively. ShapeNetCore55 is a subset of the ShapeNet dataset, and it contains 51,190 3D shapes of 55 shape classes.

In 3D shape classification experiments, the metrics employed for evaluating the performance of different methods include *average instance accuracy* and *average class accuracy*. In 3D shape retrieval experiments, *mean Average Precision* (mAP), *Precision and Recall* (PR) curves, *precision* (P), *recall* (R), *F1 score* (F1) and *Normalized Discounted Cumulative Gain* (NDCG) are presented to compare the performances of different methods under different benchmarks.

B. The setup of parameters

In this subsection, the key parameters involved in SeqViews2SeqLabels are set by exploring their impacts on the performance of SeqViews2SeqLabels in shape classification experiments under ModelNet40. The average instance accuracy is used as the metric for the performance comparison, and the GRU cell is employed in SeqViews2SeqLabels.

The key parameters include the dimension of hidden state, the embedding dimension of sequential labels, the learning rate, and the number of views in the view sequence captured around each 3D shape.

The dimension of the hidden state. The hidden states of the encoder-RNN and the decoder-RNN have the same dimension in SeqViews2SeqLabels. In this experiment, the results obtained with different candidate dimensions of hidden state are compared as shown in Table I, where the dimension of label embedding is set to 256, and the learning rate is 0.0001.

TABLE I
THE DIMENSION OF HIDDEN STATE COMPARISON UNDER MODELNET40, EMBEDDING=256, RATE=0.0001.

Hidden state dimension	64	128	256	512
Accuracy(%)	92.91	**93.11**	92.83	92.95

The candidate dimensions of hidden state form a set $\{64, 128, 256, 512\}$. From the comparison shown in Table I,

all results obtained with these candidate dimensions are very good, and the best result is achieved with 128. Thus, the dimension of the hidden state is set to 128 in the following experiments.

The embedding dimension of sequential labels. We conduct a comparison of different embedding dimensions of sequential labels using 128 dimensional hidden states and a learning rate of 0.0001.

TABLE II
THE DIMENSION OF LABEL EMBEDDING COMPARISON UNDER MODELNET40, HIDDEN=128, RATE=0.0001.

Label embedding dimension	64	128	256	512
Accuracy(%)	92.54	92.91	**93.11**	92.99

The results with candidate embedding dimensions $\{64, 128, 256, 512\}$ of sequential labels are compared in Table II. The best result is achieved with 256, which is used in the following experiments.

The former two comparisons also imply that the performance of SeqViews2SeqLabels can not be further improved by increasing the dimension of hidden states and the embedding dimension of sequential labels under ModelNet40. However, we believe the learning ability of SeqViews2SeqLabels could be increased via enlarging the dimension of the hidden state and the embedding dimension of sequential labels if more training samples were available.

The learning rate. The learning rate affects the optimization of parameters in SeqViews2SeqLabels. In this experiment, the results obtained with different learning rates are compared. As shown in Table III, the result obtained with learning rate of 0.0002 is better than the ones obtained in the former experiments, which achieves an accuracy of 93.31%. This comparison is conducted with the 128 dimensional hidden state and the 256 dimensional embedding of sequential labels, respectively. In the following experiments, SeqViews2SeqLabels is trained with the learning rate of 0.0002.

TABLE III
THE LEARNING RATE COMPARISON UNDER MODELNET40, HIDDEN=128, EMBEDDING=256.

Learning rate	0.00005	0.0001	0.0002	0.0004
Accuracy(%)	92.63	93.11	**93.31**	92.99

The number of views. The number of views in view sequence is also a factor of affecting the performance of SeqVews2SeqLabels. In the former experiments, 12 views in view sequence are captured around each 3D shape, which is employed for learning global features. In this experiment, different numbers of views are compared to explore the effect of number of views.

TABLE IV
THE NUMBER OF VIEWS UNDER MODELNET40, HIDDEN=128, EMBEDDING=256, RATE=0.0002.

View number	3	6	12	24
Accuracy(%)	92.71	92.78	**93.31**	92.46

In the comparison shown in Table IV, the best result is obtained with 12 views. Similar to the effect of dimension of hidden state and the embedding dimension of sequential labels, the performance of SeqViews2SeqLabels cannot be further improved by increasing the number of views, as indicated by the result with 24 views. The same phenomenon is observed under ModelNet10 as shown in Table V, where the best result is also achieved with 12 views. The reason is analysed in the following paragraph.

Although more sequential views in view sequences provide more information of each shape, it would become more difficult to aggregate more views for effective feature learning. In other words, the ability of learning long sequential data is still limited even if LSTM and GRU are specially designed to learn from long sequences. In the following experiments, 12 views in the view sequences captured around each shape are used to learn global features.

TABLE V
THE NUMBER OF VIEWS UNDER MODELNET10, HIDDEN=128, EMBEDDING=256, RATE=0.0002.

View number	3	6	12	24
Accuracy(%)	93.94	94.27	**94.71**	94.05

V. RESULTS AND ANALYSIS

In this section, the performance of SeqViews2SeqLabels is evaluated against the state-of-the-art methods in shape classification and shape retrieval under ModelNet40, ModelNet10 and ShapeNetCore55, respectively. For fair comparison, the results obtained by the state-of-the-art methods are computed from single modality, such as image, voxel or point cloud.

A. Shape classification

ModelNet40. Under ModelNet40 for shape classification, the comparison is shown in Table VI, where the modality and numbers of views are also presented. The evaluation metrics, both average class precision and average instance precision, are presented in the table if they are reported in the original paper.

Using views captured from 3D shapes in the training set of ModelNet40, VGG is fine-tuned via classifying each single view into one of 40 shape classes. The accuracy of single view classification is 89.47%, as the result named as "VGG (ModelNet40)". By voting the classification of single view over all views in each view sequence, namely "VGG (Voting)", the average instance accuracy of classifying 3D shapes is 92.50%. Fine-tuning is important to extract low-level features of views by VGG. This is because VGG is pre-trained by color images from ImageNet while the views are captured without colors. Thus, the results listed as "Ours (No finetune)" are not as good as our best results described in the following paragraph, where SeqViews2SeqLabels is trained under low-level features obtained from no fine-tuned VGG.

With SeqViews2SeqLabels employing GRU cell, our results named as "Ours" achieve 91.12% and 93.31%, as shown by the bold numbers. Our results are the best results among all reported results in terms of both average class accuracy and average instance accuracy. For fair comparison, the result of

VRN [43] is presented with a single CNN, where twice more views than ours are employed, and the result of RotationNet [29] is presented with views taken by the default camera system orientation which keeps identical with other methods. In addition, another result of ours listed as "Ours1" achieves 91.24% and 93.15%, which is also a state-of-the-art result. The comparison between "Ours" and "Ours1" implies that the unbalanced number of shapes in each shape class makes average class accuracy and average instance accuracy not positively correlated.

SeqViews2SeqLabels is able to learn the semantics of sequential views via aggregating views by the encoder-RNN, which makes SeqViews2SeqLabels insensitive to the first view position. To verify this point, the result named as "Ours (Start)" is obtained via training SeqViews2SeqLabels by sequential views with random first view position. Although the first view position is not fixed for training, the result obtained as "Ours (Start)" is still comparable to our best result.

In addition, the effect of different kinds of RNN cells is also explored in the comparison. The result listed as "Ours (LSTM)" is obtained by replacing GRU with LSTM in SeqViews2SeqLabels. The effect of different kinds of RNN cells is insignificant, as implied by the comparable result to our best result.

The effect of the attention mechanism is also highlighted in the comparison. The result listed as "Ours (No attention)" is obtained based on SeqViews2SeqLabels without attention vector for sequential labels prediction. The degenerated result implies that the attention mechanism is important for the prediction of sequential labels, especially when sequential views are with large number and complex to understand.

The result listed as "Ours (No decoder)" emphasizes the importance of sequential labels. "Ours (No decoder)" is implemented by replacing the decoder-RNN with a softmax classifier. The degenerated result shows that, by learning and predicting labels in a sequential way, the decoder-RNN successively captures more discriminative information among different shape classes than the softmax classifier. Sequential labels effectively alleviate overfitting, which increases the classification accuracy.

In addition, we also conduct an experiment to verify the effectiveness of the softmax layer for sequential labels prediction at each step of the decoder-RNN. By replacing the softmax layer with a sigmoid function, the result listed as "Ours (Sigmoid)" is obtained by minimizing the least squares error of predicted sequential labels. However, the result listed as "Ours (Sigmoid)" is not satisfactory. This is because the characteristics of backward shape classes cannot be observed for sequential labels prediction by the sigmoid function at each step.

Finally, we highlight our novel view aggregation by comparing it with widely used max pooling and mean pooling. To conduct a fair comparison, we employ the same low-level view features as the ones ("VGG (ModelNet40)") involved in our best results of "Ours". Moreover, the structure of MVCNN is trained with max pooling and mean pooling respectively, as shown by the results of "Ours(Maxpooling)" and "Ours(Meanpooling)". Due to the loss of content information in most of the views and the spatial information among the views, these results are not better than ours.

TABLE VI
CLASSIFICATION COMPARISON UNDER MODELNET40, HIDDEN=128, EMBEDDING=256, RATE=0.0002.

Methods	Modality	Views	Class(%)	Instance(%)
SHD	Mesh	-	68.23	-
LFD	Image	10	75.47	-
PyramidHoG-LFD	Image	20	87.2	90.5
Fisher vector [3]	-	12	84.8	-
3DShapeNets [20]	Voxel	12	77.32	-
DeepPano [6]	Image	1	77.6	-
Geometry image [28]	Image	1	83.9	-
VoxNet [44]	Voxel	-	83.0	-
VRN [43]	Voxel	24	-	91.33
FPNN [45]	Voxel	-	88.4	-
T-L Network [46]	Voxel	-	74.4	-
3DGAN [23]	Voxel	-	83.3	-
PointNet [47]	Point	1	86.2	89.2
PointNet++ [48]	Point	1	-	91.9
FoldingNet [49]	Point	1	-	88.4
Octree [24]	Voxel	12	90.6	-
PANORAMA [27]	Image	6	90.70	-
Pairwise [31]	Image	12	90.7	-
GIFT [5]	Image	64	89.5	-
Dominant Set [8]	Image	12	-	92.2
Su-MVCNN [3]	Image	80	90.1	-
MVCNN [13]	Image	20	89.7	92.0
MVCNN-Sphere [13]	Voxel	20	86.6	89.5
RotationNet [29]	Image	12	-	90.65
SO-Net [50]	Point	1	87.3	90.9
SliceVoxel [32]	Voxel	1	-	85.73
VGG(ModelNet40)	Image	1	-	89.47
VGG(Voting)	Image	12	90.37	92.50
Ours	Image	12	**91.12**	**93.31**
Ours1	Image	12	**91.38**	**93.07**
Ours (No finetune)	Image	12	88.63	91.57
Ours (Start)	Image	12	91.10	92.95
Ours (LSTM)	Image	12	91.14	92.99
Ours (No attention)	Image	12	88.99	91.13
Ours (No decoder)	Image	12	90.50	92.50
Ours (Sigmoid)	Image	12	63.79	77.63
Ours (Maxpooling)	Image	12	89.77	91.53
Ours (Meanpooling)	Image	12	89.97	91.57

ModelNet10. The performance of SeqViews2SeqLabels is further evaluated under ModelNet10 for shape classification. The comparison is shown in Table VII.

The VGG fine-tuned by the views from ModelNet40 is first used to extract the low-level features of sequential views which are captured from the 3D shapes in ModelNet10.

As the results listed as "Ours" and "Ours (LSTM)" shown, SeqViews2SeqLabels achieves the best results under ModelNet10. Comparing with the GRU cell, LSTM cell performs better under ModelNet10, where average class accuracy and average instance accuracy achieve up to 94.80% and 94.82%, respectively.

The effects of attention mechanism and sequential labels are also highlighted in the comparison. Although both the results listed as "Ours (No attention)" and "Ours (No decoder)" are better than the ones of other state-of-the-art methods, they are degenerated compared with "Ours" or "Ours (LSTM)" due to the lack of attention mechanism and sequential labels, respectively.

With the low-level features provided by VGG which is fine-tuned under the views captured from the shapes in Model-

Net10, we explore whether better results could be achieved. As the result listed as "VGG (ModelNet10)", the accuracy of classifying single view into one of 10 shape classes is 91.87%. By voting the classification of single view over all sequential views in each view sequence, the accuracy of classifying shapes is achieved to 93.83%, as listed as "VGG (Voting)". Although the results of "Ours1" and "Ours1 (LSTM)" are slightly degenerated compared with the results of "Ours" and "Ours (LSTM)", they are still the state-of-the-art results among all reported results.

Under ModelNet10, we repeat the experiments of "Ours (Start)", "Ours(Maxpooling)" and "Ours(Meanpooling)" conducted under ModelNet40. As the results shown in Table VII, the same phenomenons are observed.

TABLE VII
CLASSIFICATION COMPARISON UNDER MODELNET10, HIDDEN=128, EMBEDDING=256, RATE=0.0002.

Methods	Modality	Views	Class(%)	Instance(%)
SHD	Mesh	-	79.79	-
LFD	Mesh	10	79.87	-
3DShapeNets [20]	Voxel	12	83.54	-
DeepPano [6]	Image	1	85.5	-
Geometry image [28]	Image	1	88.4	-
VoxNet [44]	Image	-	92.0	-
VRN [43]	Voxel	24	-	93.8
3DGAN [23]	Voxel	-	91.0	-
ORION [51]	Voxel	-	93.8	-
FoldingNet [49]	Point	1	-	94.4
PANORAMA [27]	Image	6	91.12	-
Pairwise [31]	Image	12	92.8	-
GIFT [5]	Image	64	91.5	-
RotationNet [29]	Image	12	-	93.84
3DDescriptorNet [52]	Voxel	-	-	92.4
SO-Net [50]	Point	1	93.9	94.1
SliceVoxel [32]	Voxel	1	-	91.40
Ours	Image	12	**94.56**	**94.71**
Ours (LSTM)	Image	12	**94.80**	**94.82**
Ours (No attention)	Image	12	93.15	93.17
Ours (No decoder)	Image	12	93.75	93.83
Ours (Start)	Image	12	**94.55**	**94.60**
Ours (Maxpooling)	Image	12	92.00	92.07
Ours (Meanpooling)	Image	12	93.12	93.17
VGG (ModelNet10)	Image	1	-	91.87
VGG (Voting)	Image	12	93.83	93.83
Ours1	Image	12	**94.51**	**94.60**
Ours1 (LSTM)	Image	12	**94.12**	**94.27**

ShapeNetCore55. In this experiment, the performance of SeqViews2SeqLabels is evaluated under ShapeNetCore55. For each 3D shape, 12 sequential views rendered without colors are used to train SeqViews2SeqLabels. In addition, we also explore whether sequential views rendered with colors can be used to improve the performance of SeqViews2SeqLabels. The sequential views with colors are downloaded from the web page of ShapeNet, however, there are only 8 sequential views in each view sequence. The results are shown in Table VIII.

In Table VIII, the results named as "VGG (ShapeNetCore55)" and "VGG1 (ShapeNetCore55)" are obtained via fine-tuning VGG by the views without colors and the views with colors, respectively, where the results obtained by voting are correspondingly listed as "VGG (Voting)" and "VGG1 (Voting)". Because of the highly unbalanced number of shapes in each shape class, we only present our best results in terms of average class accuracy, as listed as "Ours" and "Ours1" which are obtained by low-level view features from "VGG (ShapeNetCore55)" and "VGG1 (ShapeNetCore55)", respectively. The comparison between these results implies that the color is slightly helpful to increase the performance of SeqViews2SeqLabels in terms of average class accuracy, from 74.81% and 76.91%. We also try to explore whether the performance of SeqViews2SeqLabels could be improved via increasing the dimension of hidden state, as the results of "Ours (512)" and "Ours1 (512)". However, the results with higher dimension of hidden states are comparable to "Ours" or "Ours1" respectively, which implies that the 128 dimensional hidden states are sufficiently good to learn from shapes for the scale of ShapeNetCore55.

TABLE VIII
CLASSIFICATION COMPARISON UNDER SHAPENETCORE55, HIDDEN=128, EMBEDDING=256, RATE=0.0002.

Methods	Modality	Views	Class(%)	Instance(%)
VGG(ShapeNetCore55)	Image	1	-	83.85
VGG(Voting)	Image	12	71.84	86.78
Ours	Image	12	**74.81**	85.47
Ours (512)	Image	12	**75.11**	85.10
VGG1 (ShapeNetCore55)	Image	1	-	83.68
VGG1 (Voting)	Image	8	76.03	87.04
Ours1	Image	8	**76.91**	86.61
Ours1 (512)	Image	8	**76.84**	85.94

B. The effect of shape class order

In this subsection, we explore the effect of shape class order under ModelNet40 and ModelNet10 in shape classification. In the experiments above, we use the default shape class order provided by the benchmark, while we employ randomized shape class order in this experiment. Specifically, we randomize shape class order 40 times under each benchmark. Using each randomized shape class order, we repeat the shape classification with the parameters of "Ours" in Table VI or Table VII. Finally, we compute the mean, standard deviation and maximum over the 40 groups of results in terms of average instance accuracy and average class accuracy, as shown in Table IX.

TABLE IX
THE EFFECT OF SHAPE CLASS ORDER UNDER MODELNET40 AND MODELNET10, HIDDEN=128, EMBEDDING=256, RATE=0.0002.

Metrics (%)	ModelNet40	ModelNet10
Instance mean	93.20	94.52
Instance std	0.09	0.11
Class mean	91.02	94.45
Class std	0.14	0.12
Ours(ClassMax)-Instance	**93.40**	**94.71**
Ours(ClassMax)-Class	**91.10**	**94.65**

The statistic results show that the effect of shape class order is subtle. Under both benchmarks, the mean values are high, and the standard deviations are quite small, in terms of both average instance accuracy and average class accuracy. In addition, we even obtain a higher instance accuracy than our best results with default shape class order under ModelNet40, as shown by "Ours(ClassMax)-Instance".

C. Attention visualization

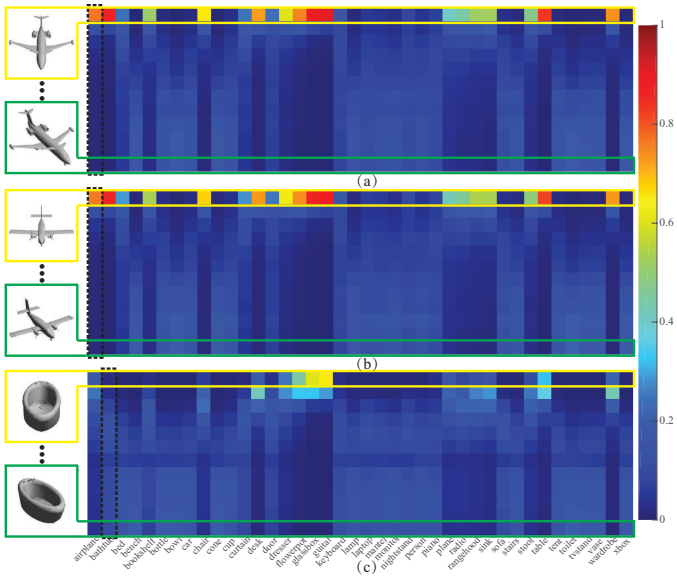


Fig. 4. The attention weights learned by SeqViews2SeqLabels for two airplanes and one bathtub from ModelNet40, as shown in (a), (b) and (c) respectively.

In this subsection, the attention learned by SeqViews2SeqLabels under ModelNet40 is visualized to analyse how SeqViews2SeqLabels recognizes 3D shapes by understanding sequential views. As shown in Fig. 4, the attention weight vectors $\{\boldsymbol{\alpha}_c^i\}$ for all shape classes over sequential views in $\boldsymbol{v}^i$ are visualized as a matrix, such as the ones of two airplanes in Fig. 4 (a), (b) and the one of a bathtub in Fig. 4 (c), where red represents high attention weight and each $\boldsymbol{\alpha}_c^i$ is the c-th column of the matrix. SeqViews2SeqLabels learns the attention weights of two airplanes with similar patterns which are much different from the ones of bathtub. In addition, the learned attention weights conform to the human cognition of objects. Specifically, for shapes like airplanes with distinctive characteristics, most shape classes can make certain label predictions upon merely the first view. This can be observed in most red entries in the first row of matrices in Fig. 4 (a) and (b). In contrast, for shapes without distinctive characteristics, such as the bathtub which is similar to "cup" or "flowerpot", most shape classes need almost all views to predict each sequential label in label sequence, as shown by the inapparent entries in most columns of the matrix in Fig. 4 (c).

D. Shape retrieval

The performance of SeqViews2SeqLabels is also evaluated using the learned global features in shape retrieval experiments under ModelNet40, ModelNet10 and ShapeNetCore55, respectively. Under ModelNet40 and ModelNet10, our results are produced with the global features learned by the trained SeqViews2SeqLabels named as "Ours" in the corresponding Table VI, Table VII.

The shapes in ModelNet40 and ModelNet10 are originally split into a training set and a testing set. Thus, to comprehensively evaluate the performance of SeqViews2SeqLabels for shape retrieval, four experiments are conducted under each benchmark. The four experiments are named as "Test-Test", "Test-Train", "Train-Train", and "All-All", indicating which data set the query and retrieved shapes come from, respectively. For example, "Test-Train" indicates that the shapes in the testing set are used as query for shape retrieval from the training set.

The comparison between SeqViews2SeqLabels and the state-of-the-art methods is shown in terms of mAP in Table X, where the retrieval range is also explained. Under ModelNet40, the mAPs obtained by SeqViews2SeqLabels are the best, which achieves 89.00% and 96.73% in the "Test-Test" and "All-All" experiments, respectively, as shown by the bold numbers. Under ModelNet10, the mAPs of SeqViews2SeqLabels achieve 89.55% and 97.85% in the "Test-Test" and "All-All" experiments, respectively. The corresponding PR curves of our results obtained under ModelNet40 and ModelNet10 are shown in Fig. 5 (a) and (b), respectively, where the PR curves of our results show a high performance of SeqViews2SeqLabels.

We believe our results are also the best as shown in bold, even if GIFT obtains a higher mAP. This is because, the dataset used by GIFT is formed by randomly selecting 100 shapes from each shape category, which is much simpler than the whole benchmark that we used. To verify this point, we employ the same low-level view features to compare with GIFT (64 clusters) under the whole ModelNet40 and ModelNet10, as shown by "GIFT1". In addition, for better analysis of SeqViews2SeqLabels in shape retrieval, we also present the retrieval results with the features learned by the variants of SeqViews2SeqLabels compared in the shape classification experiments, such as "Ours(LSTM)", "Ours(Start)","Ours(No attention)", "Ours(No decoder)" and "Ours(ClassMax)". The corresponding PR curves are presented in Fig. 6.

TABLE X
RETRIEVAL COMPARISON UNDER MODELNET40 AND MODELNET10, HIDDEN=128, EMBEDDING=256, RATE=0.0002.

Methods	Range	ModelNet40	ModelNet10
SHD	Test-Test	33.26	44.05
LFD	Test-Test	40.91	49.82
3DShapeNets	Test-Test	49.23	68.26
Geometry image	Test-Test	51.30	74.90
DeepPano	Test-Test	76.81	84.18
su-MVCNN	Test-Test	79.50	-
PANORAMA	Test-Test	83.45	87.39
GIFT	Random	81.94	**91.12**
Triplet-Center [53]	Test-Test	88.0	-
SliceVoxel [32]	Test-Test	77.48	85.34
Ours	Test-Test	**89.00**	**89.55**
Ours	Test-Train	**92.41**	**93.56**
Ours	Train-Train	**98.76**	**99.65**
Ours	All-All	**96.73**	**97.85**
Ours(LSTM)	Test-Test	88.83	**91.43**
Ours(Start)	Test-Test	88.09	**89.80**
Ours(No attention)	Test-Test	88.54	88.46
Ours(No decoder)	Test-Test	87.49	86.66
GIFT1	Test-Test	86.56	89.04
Ours(ClassMax)	Test-Test	**89.09**	**89.45**

Under the three subsets of ShapeNetCore55, i.e., training set, validation set and testing set, the retrieval performance of SeqViews2SeqLabels is compared with other state-of-the-art

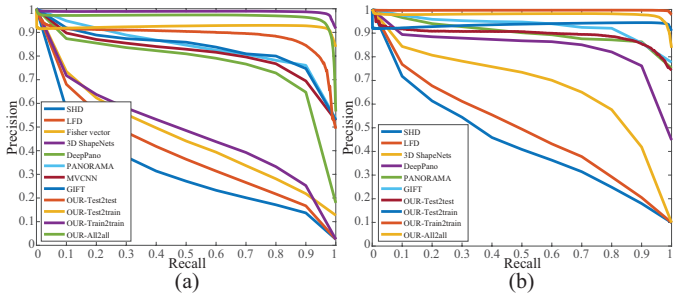


Fig. 5. The comparison between precision and recall cures obtained by different methods under (a) ModelNet40 and (b) ModelNet10.

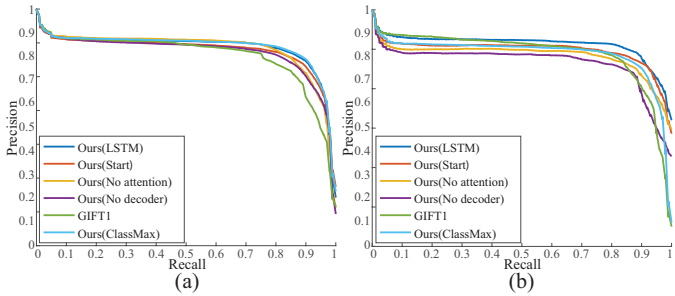


Fig. 6. The comparison between precision and recall cures obtained by GIFT and different variants of SeqViews2SeqLabels under (a) ModelNet40 and (b) ModelNet10 based on the same low-level view features.

methods in terms of different metrics. Considering that there is no comparison results under training set and validation set in [54], the results of state-of-the-art methods under testing set are from the SHREC2017 retrieval contest [54], while the ones under training set and validation set are from the SHREC2016 retrieval contest [30]. All involved 3D shapes under ShapeNetCore55 are normal and are not perturbed by rotation. In Table XI, we present the performance obtained by SeqViews2SeqLabels respectively trained under views without colors and views with colors, as the ones named as "Ours (512)" and "Ours1" in Table VIII. The comparison shown in Table XI implies that the performance of SeqViews2SeqLabels for shape retrieval is the best among all state-of-the-art methods under all subsets, where our results under views without colors and views with colors are listed as "Ours" and "Ours (C)", respectively. In addition, the comparison between results of "Ours" and "Ours (C)" also demonstrate that colors in views for training do not significantly improve the retrieval performance of SeqViews2SeqLabels.

VI. Conclusions, Limitations and Future Work

A. Conclusions

In this paper, a novel deep learning model, SeqViews2SeqLabels, is proposed to learn 3D global features via aggregating sequential views captured around 3D shapes on a circle. In existing methods, a pooling procedure is employed to aggregate multiple views but suffers from two issues, i.e., the lack of content information of almost all views and the lack of spatial information among the views. To resolve these disadvantages, SeqViews2SeqLabels employs an encoder-RNN to aggregate sequential views, which effectively learns global features with semantics. In addition, the other part of the encoder-decoder structure of SeqViews2SeqLabels, the decoder-RNN, predicts sequential labels based on the learned global features. The decoder-RNN is able to capture more and finer discriminative information among all shape classes to effectively alleviate overfitting for higher classification accuracy. Finally, an attention mechanism is integrated in the decoder-RNN, which assigns heavier weights on the low-level features of distinctive views for each shape class. The introduced attention assists the encoder-RNN in learning the semantic meaning of view sequences by dramatically reducing the effect of the first view position. The attention mechanism is experimentally verified to further improve the discriminative ability of SeqViews2SeqLabels.

B. Limitations and future work

Although SeqViews2SeqLabels learns 3D global features with high performance, it still suffers from two disadvantages. First, SeqViews2SeqLabels can only learn features via aggregating sequential views rather than any kind of unordered views, such as views captured on a unit sphere centered at 3D shapes. Second, although RNNs are good at aggregating sequential data, their ability is limited when the sequence contains a large number of data, especially for the complex data, such as views in this work. Thus, SeqViews2SeqLabels merely performs well under limited number of sequential views, even with the help of the attention mechanism.

In the future, it is worth to explore how to aggregate large numbers of sequential views in view sequences with novel deep learning models, since more views could provide more information to learn for discriminating 3D shapes.

References

[1] K. Lu, R. Ji, J. Tang, and Y. Gao, "Learning-based bipartite graph matching for view-based 3D model retrieval," *IEEE Transactions on Image Processing*, vol. 23, no. 10, pp. 4553–4563, 2014.

[2] H. Guo, J. Wang, Y. Gao, J. Li, and H. Lu, "Multi-view 3D object retrieval with deep embedding network," *IEEE Transactions on Image Processing*, vol. 25, pp. 5526–5537, 2016.

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

[4] H. Huang, E. Kalegorakis, S. Chaudhuri, D. Ceylan, V. Kim, and E. Yumer, "Learning local shape descriptors with view-based convolutional neural networks," *ACM Transactions on Graphics*, 2017.

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

[6] B. Shi, S. Bai, Z. Zhou, and X. Bai, "Deeppano: Deep panoramic representation for 3D shape recognition," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2339–2343, 2015.

[7] T. Furuya and R. Ohbuchi, "Deep aggregation of local 3D geometric features for 3D model retrieval," in *Proceedings of the British Machine Vision Conference*, 2016.

[8] C. Wang, M. Pelillo, and K. Siddiqi, "Dominant set clustering and pooling for multi-view 3D object recognition," in *Proceedings of British Machine Vision Conference*, 2017.

[9] Z. Han, Z. Liu, J. Han, C.-M. Vong, S. Bu, and X. Li, "Unsupervised 3D local feature learning by circle convolutional restricted boltzmann machine," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5331–5344, 2016.

[10] Z. Han, Z. Liu, J. Han, C.-M. Vong, S. Bu, and C. Chen, "Mesh convolutional restricted boltzmann machines for unsupervised learning of features with structure preservation on 3D meshes," *IEEE Transactions on Neural Network and Learning Systems*, vol. 28, no. 10, pp. 2268 – 2281, 2017.

TABLE XI
RETRIEVAL COMPARISON UNDER SHAPENETCORE55, HIDDEN=128, EMBEDDING=256, RATE=0.0002.

		micro					macro				
Datasets	Methods	P@N	R@N	F1@N	mAP@N	NDCG@N	P@N	R@N	F1@N	mAP@N	NDCG@N
Tesing	Kanezaki	0.810	0.801	**0.798**	0.772	0.865	0.602	0.639	**0.590**	0.583	0.656
	Zhou	0.786	0.773	0.767	0.722	0.827	0.592	0.654	0.581	0.575	0.657
	Tatsuma	0.765	0.803	0.772	0.749	0.828	0.518	0.601	0.519	0.496	0.559
	Furuya	**0.818**	0.689	0.712	0.663	0.762	**0.618**	0.533	0.505	0.477	0.563
	Thermos	0.743	0.677	0.692	0.622	0.732	0.523	0.494	0.484	0.418	0.502
	Deng	0.418	0.717	0.479	0.540	0.654	0.122	0.667	0.166	0.339	0.404
	Li	0.535	0.256	0.282	0.199	0.330	0.219	0.409	0.197	0.255	0.377
	Mk	0.793	0.211	0.253	0.192	0.277	0.598	0.283	0.258	0.232	0.337
	Su	0.770	0.770	0.764	0.735	0.815	0.571	0.625	0.575	0.566	0.640
	Bai	0.706	0.695	0.689	0.640	0.765	0.444	0.531	0.454	0.447	0.548
	Taco [55]	0.701	0.711	0.699	0.676	0.756	-	-	-	-	-
	Ours	0.5964	**0.8034**	0.6105	**0.8373**	**0.9022**	0.1862	**0.8144**	0.2375	**0.6816**	**0.8364**
	Ours(C)	0.6012	**0.8122**	0.6158	**0.8567**	**0.9082**	0.1883	**0.8285**	0.2405	**0.7266**	**0.8560**
Validation	Su	0.805	0.800	**0.798**	0.910	0.938	0.641	0.671	**0.642**	0.879	0.920
	Bai	0.747	0.743	0.736	0.872	0.929	0.504	0.571	0.516	0.817	0.889
	Li	0.343	**0.924**	0.443	0.861	0.930	0.087	**0.873**	0.132	0.742	0.854
	Wang	0.682	0.527	0.488	0.812	0.881	0.247	0.643	0.266	0.575	0.712
	Tatsuma	0.306	0.763	0.378	0.722	0.886	0.096	0.828	0.140	0.601	0.801
	Ours	**0.8736**	0.1036	0.1507	**0.9556**	**0.9553**	**0.6478**	0.3395	0.3534	**0.9240**	**0.9425**
	Ours(C)	**0.8771**	0.1374	0.1893	**0.9496**	**0.9524**	**0.6443**	0.4224	0.4010	**0.9146**	**0.9394**
Training	Su	0.939	0.944	**0.941**	0.964	0.923	0.909	0.935	**0.921**	0.964	0.947
	Bai	0.841	0.571	0.620	0.907	0.912	0.634	0.452	0.472	0.815	0.891
	Li	0.827	**0.996**	0.864	0.990	0.978	0.374	**0.997**	0.460	0.982	0.986
	Wang	0.884	0.260	0.363	0.917	0.891	0.586	0.497	0.428	0.775	0.863
	Ours	**0.9954**	0.0058	0.0115	**0.9996**	**0.9844**	**0.9930**	0.0221	0.0424	**0.9995**	**0.9909**
	Ours(C)	**0.9972**	0.0059	0.0115	**0.9997**	**0.9842**	**0.9969**	0.0222	0.0426	**0.9997**	**0.9909**

[11] Z. Han, Z. Liu, C.-M. Vong, Y.-S. Liu, S. Bu, J. Han, and C. Chen, "BoSCC: Bag of spatial context correlations for spatially enhanced 3D shape representation," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3707–3720, 2017.

[12] Z. Han, Z. Liu, J. Han, C. Vong, S. Bu, and C. Chen, "Unsupervised learning of 3D local features from raw voxels based on a novel permutation voxelization strategy," *IEEE Transactions on Cybernetics*, 2017, doi:10.1109/TCYB.2017.2778764.

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

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] J. Xie, Y. Fang, F. Zhu, and E. Wong, "DeepShape: Deep learned shape descriptor for 3D shape matching and retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1275–1283.

[16] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on riemannian manifolds," in *Proc. of the International IEEE Workshop on 3D Representation and Recognition*, 2015.

[17] D. Boscaini, J. Masci, S. Melzi, M. M. Bronstein, U. Castellani, and P. Vandergheynst, "Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks," *Computer Graphics Forum*, vol. 34, no. 5, pp. 13–23, 2015.

[18] J. Xie, M. Wang, and Y. Fang, "Learned binary spectral shape descriptor for 3d shape correspondence," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[19] Z. Han, Z. Liu, C. Vong, Y.-S. Liu, S. Bu, J. Han, and C. Chen, "Deep spatiality: Unsupervised learning of spatially-enhanced global and local 3D features by deep neural network with coupled softmax," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 3049–3063, 2018.

[20] 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 IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1912–1920.

[21] A. Sharma, O. Grau, and M. Fritz, "VConv-DAE: Deep volumetric shape learning without object labels," in *Proceedings of European Conference on Computer Vision*, 2016, pp. 236–250.

[22] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta, "Learning a predictable and generative vector representation for objects," in *Proceedings of European Conference on Computer Vision*, 2016, pp. 484–499.

[23] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," in *Advances in Neural Information Processing Systems*, 2016, pp. 82–90.

[24] 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. 72:1–72:11, 2017.

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

[26] J. Xie, G. Dai, F. Zhu, and Y. Fang, "Learning barycentric representations of 3D shapes for sketch-based 3D shape retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[27] K. Sfikas, T. Theoharis, and I. Pratikakis, "Exploiting the PANORAMA Representation for Convolutional Neural Network Classification and Retrieval," in *Eurographics Workshop on 3D Object Retrieval*, 2017, pp. 1–7.

[28] A. Sinha, J. Bai, and K. Ramani, "Deep learning 3D shape surfaces using geometry images," in *European Conference on Computer Vision*, 2016, pp. 223–240.

[29] A. Kanezaki, Y. Matsushita, and Y. Nishida, "Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[30] M. Savva, F. Yu, H. Su, M. Aono, B. Chen, D. Cohen-Or, W. Deng, H. Su, S. Bai, and X. Bai, "Shrec'16 track large-scale 3D shape retrieval from shapeNet core55," in *EG 2016 workshop on 3D Object Recognition*, 2016.

[31] E. Johns, S. Leutenegger, and A. J. Davison, "Pairwise decomposition of image sequences for active multi-view recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3813–3822.

[32] R. Miyagi and M. Aono, "Sliced voxel representations with LSTM and CNN for 3D shape recognition," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2017.

[33] T. Le, G. Bui, and Y. Duan, "A multi-view recurrent neural network for 3D mesh segmentation," *Computers and Graphics*, vol. 66, pp. 103–112, 2017.

[34] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298–2304, 2017.

[35] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recog-

nition with automatic rectification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4168–4176.

[36] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *ICML*, vol. 37, 2015, pp. 2048–2057.

[37] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems*, 2015, pp. 577–585.

[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, vol. 25, pp. 1097–1105.

[40] K. Cho, B. V. Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *Computer Science*, 2014.

[41] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014.

[42] 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.

[43] A. Brock, T. Lim, J. Ritchie, and N. Weston, "Generative and discriminative voxel modeling with convolutional neural networks," in *3D deep learning workshop (NIPS)*, 2016.

[44] D. Maturana and S. S., "Voxnet: A 3D convolutional neural network for real-time object recognition," in *International Conference on Intelligent Robots and Systems*, 2015, pp. 922–928.

[45] Y. Li, S. Pirk, H. Su, C. R. Qi, and L. J. Guibas, "FPNN: Field probing neural networks for 3D data." in *NIPS*, 2016, pp. 307–315.

[46] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta, "Learning a predictable and generative vector representation for objects," in *European Conference on Computer Vision*, 2016, pp. 484–499.

[47] 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.

[48] 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*, 2017, pp. 5105–5114.

[49] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[50] J. Li, B. M. Chen, and G. H. Lee, "SO-Net: Self-organizing network for point cloud analysis," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[51] N. Sedaghat, M. Zolfaghari, E. Amiri, and T. Brox, "Orientation-boosted voxel nets for 3D object recognition," in *British Machine Vision Conference*, 2017.

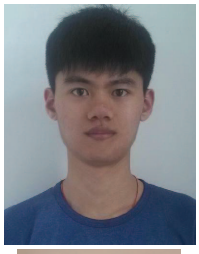
[52] J. Xie, Z. Zheng, R. Gao, W. Wang, S.-C. Zhu, and Y. N. Wu, "Learning descriptor networks for 3D shape synthesis and analysis," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[53] X. He, Y. Zhou, Z. Zhou, S. Bai, and X. Bai, "Triplet-center loss for multi-view 3D object retrieval," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[54] 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, "SHREC'17 Large-Scale 3D Shape Retrieval from ShapeNet Core55," in *Eurographics Workshop on 3D Object Retrieval*, 2017.

[55] T. S. Cohen, M. Geiger, J. Khler, and M. Welling, "Spherical CNNs," in *International Conference on Learning Representations*, 2018.

Zhizhong Han is a PhD student with Northwestern Polytechnical University, China. He is majored in pattern recognition and machine intelligence. His research interests include machine learning, pattern recognition, feature learning and digital geometry processing.



Mingyang Shang is currently a master candidate in the School of Software at Tsinghua University. He received his BS in Software Engineering from Dalian University of Technology, China, 2016. His research interests include deep learning, shape analysis and pattern recognition and NLP.

Zhenbao Liu (M'11) is currently a Professor with Northwestern Polytechnical University, China. He received the Ph.D. degree from the College of Systems and Information Engineering, University of Tsukuba, Tsukuba, Japan, in 2009. He was a visiting scholar with Simon Fraser University, Canada, in 2012. He has published approximately 50 papers in major international journals and conferences. His research interests include pattern recognition, computer vision, and shape analysis.

Chi-Man Vong (M'09-SM'14) received the M.S. and Ph.D. degrees in Software Engineering from the University of Macau in 2000 and 2005, respectively. He is currently an Associate Professor with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau. His research interests include machine learning methods and intelligent systems.

Yu-shen Liu is an Associate Professor in School of Software at Tsinghua University, Beijing, China. He received his BS in mathematics from Jilin University, China, in 2000. He earned his PhD in the Department of Computer Science and Technology at Tsinghua University, China, in 2006. He spent three years as a post doctoral researcher in Purdue University from 2006 to 2009. His research interests include shape analysis pattern recognition, machine learning and semantic search.

Matthias Zwicker is a professor at the Department of Computer Science, University of Maryland, College Park, where he holds the Reginald Allan Hahne Endowed E-nnovate chair. He obtained his PhD from ETH in Zurich, Switzerland, in 2003. Before joining University of Maryland, he was an Assistant Professor at the University of California, San Diego, and a professor at the University of Bern, Switzerland. His research in computer graphics covers signal processing for high-quality rendering, point-based methods for rendering and modeling, 3D geometry processing, and data-driven modeling and animation.

Junwei Han (M'12-SM'15) is a currently a Professor with Northwestern Polytechnical University, Xi'an, China. He received his Ph.D. degree in pattern recognition and intelligent systems from the School of Automation, Northwestern Polytechnical University in 2003. His research interests include multimedia processing and brain imaging analysis. He is an Associate Editor of IEEE Trans. on Human-Machine Systems, Neurocomputing, and Multidimensional Systems and Signal Processing.

C.L. Philip Chen (S'88CM'88CSM'94CF'07) received his M.S.degree in electrical engineering from University of Michigan, Ann Arbor, in 1985 and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, in 1988. After having worked at U.S. for 23 years as a tenured professor, as a department head and associate dean in two different universities, he is currently the Dean of the Faculty of Science and Technology, University of Macau, Macau, China and a Chair Professor of the Department of Computer and Information Science.