Sequential learning with fine-grained representations for sketch-based 3D model retrieval

Michael Shell, *Member, IEEE,* John Doe, *Fellow, OSA,* and Jane Doe, *Life Fellow, IEEE*

***Abstract*—Sketch-based 3D model retrieval suffers the large visual discrepancy between 2D sketches and 3D models. Most existing state-of-the-art methods directly project 3D models and 2D sketches into a deep feature space to learn common semantic embedding for both modalities to alleviate the discrepancy. We argue that simultaneous learning of the semantic embedding for two significantly different modalities would restrict the discrimination of 3D model representation, which will cause inferior retrieval results and visual dissimilarity. In this work, we propose a novel Sequential Learning (SL) framework for sketch-based 3D model retrieval to learn representations of 3D models and 2D sketches separately and sequentially. Specifically, the SL framework is composed of two modules, a 3D Model Network (3DMN) which learns representations of 3D models firstly, and a 2D Sketch Network (2DSN) which performs feature learning of 2D sketches guided by the learned 3D representations to align feature distributions across domains. Concretely, the 3DMN removes 2D sketches to eliminate the interference of 2D modality, and a discriminative loss is formulated only on 3D models to promote the discrimination of 3D model features. In addition, we further consider the implicit fine-grained classes of 3D models. The 2DSN pulls sketches close to the closest fine-grained class representations obtained by clustering algorithms with a correlation loss formulated on sketch features and fine-grained 3D representations to learn better 2D sketch features and improve the 3D model retrieval accuracy and visual similarity accordingly. Extensive experiments on three large-scale benchmark datasets for 3D model retrieval demonstrate the effectiveness of the proposed SL framework and fine-grained representations and the competitiveness of our approach outperforming the current state-of-the-art.**

***Index Terms*—sketch-based retrieval, 3D model retrieval, sequential learning.**

I. INTRODUCTION

RETRIEVING 3D models from 2D sketches has been widely used in various fields, such as 3D printing [1], medical digital imaging [2], product conceptual design [3] and animation creation [4]. Comparing to text [5] and 3D models [6], [7] as queries , 2D sketches provides an easier interaction way for users to search the desired 3D models. While being intuitive and convenient, sketches are also informative enough to specify shapes.

The main challenge of sketch-based 3D model retrieval is that 2D sketches and 3D models are from two different modalities with large visual discrepancy. The features extracted from these two domains also follow different distributions, which leads to inferior performance by directly searching 3D models through sketch features.

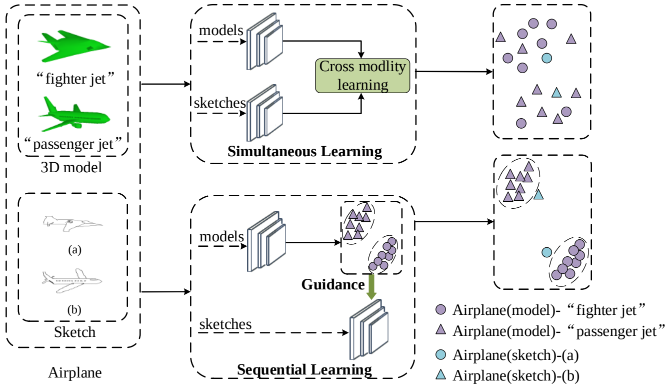


Fig. 1. The motivation of proposed sequential learning with fine-grained representations. Simultaneous learning learns common features of 3D models and 2D sketches through cross-modality learning. Sketch (a) is visually like 3D model ”fighter jet” but closer to some ”passenger jet” in the shared feature space. Sequential learning learns 3D model features without 2D sketches and achieve better discrimination. The samples of class ”Airplane” are separated into two clusters seen as two implicit subcategories. Sketch (a) and (b) are pulled close to the 3D models ”fighter jet” and ”passenger jet”, respectively.

Recently, sketch-based 3D model retrieval methods can be roughly divided into model-based [8]-[9] and view-based [10]-[11]. Model-based methods are very straightforward which directly extracts original features from polygon meshes [12]-[13], voxel grids [14]-[15], or point clouds [8]-[9] and then input the vectors into the deep learning model to generate the final features. However, this kind of method is limited by the high computational complexity [16]. View-based methods firstly convert 3D models into 2D views using line rendering algorithms [17], then extract deep features from these 2D projections, and represent 3D models with more discriminative and compact features by max pooling [10], group aggregating [18] or recursive neural networks (RNNs) [11]. Finally, a metric is learned to eliminate the discrepancy between 2D views of 3D models and 2D sketches. View-based methods have achieved better performance than model-based ones. In this paper, we focus on view-based methods and use 2D projections of 3D models for retrieval.

Sketch-based 3D model retrieval is a problem of matching 2D sketches with 3D models. The matching results is inferior when 3D model features are not discriminative. For example, ”barn” and ”house” are two 3D model classes with high visual similarity. The 2D sketches of class ”barn” and ”house” can not match with the corresponding 3D models if the 3D models of the same class are hard to distinguish in the learned feature

M. Shell was with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA e-mail: (see http://www.michaelshell.org/contact.html).

J. Doe and J. Doe are with Anonymous University.

Manuscript received April 19, 2005; revised August 26, 2015.

space. Almost all state-of-the-art approaches simultaneously learn features of 3D models and 2D sketches for modality-invariant representations as shown in Fig. 1. We argue that the 3D features learned by simultaneous learning are not discriminative enough due to the huge discrepancy between two modalities, which reduces the final retrieval accuracy. In this work, we propose a sequential learning framework to learn representations of 3D models and 2D sketches separately and sequentially. The proposed SL framework firstly learns 3D model features without 2D sketches, and then learns 2D sketch features with the guidance of learned 3D model representations. Note that, without the interference of 2D sketches, the discrimination of 3D representations can be further promoted. Since the learned 3D model features are more discriminative than joint learning, we can learn better 2D sketch feature by feature aligning approaches and improve the final retrieval accuracy accordingly. As shown in Fig. 1, the 3D model features learned by SL are more discriminative than that by simultaneous learning and the final retrieval results is better. Table II shows that, the higher the classification accuracy of 3D models which indicates better discrimination of 3D model representations, the higher the mAP value of the 3D model retrieval.

We further consider the fact that some categories of 3D models may have many implicit fine-grained subcategories. For instance, A and B are two implicit subcategories of a 3D model class. Given a query sketch visually like A, the retrieved 3D models like A, which are most relevant to query sketch, should rank at the top of retrieval list. In this work, we propose to exploit the fine-grained class information of 3D models to guide the feature learning of 2D sketches. We firstly calculate the fine-grained class centers of 3D model features in one class by unsupervised clustering algorithms. Then, the similarity learning can be completed by computing the distance between sketch features and fine-grained class centers, thus mining better 2D sketch representations. While previous simultaneous learning methods only consider the common representations of 2D sketches and 3D models, we further explore the intra-class discrimination. With the guidance of fine-grained 3D model representations, 2D sketches are as close as possible to their corresponding subclass centers. Fig. 1 shows that given a query sketch (a) like a "fighter jet", it is closer to the 3D models like "fighter jet" than those like "passenger jet".

To summarize, we propose a novel sequential learning framework for sketch-based 3D model retrieval. We learn the representations of 3D models and 2D sketches in two phases. In the first phase, we construct a 3DMN network to perform discriminative learning of 3D models. Firstly, a 3D model is rendered into multiple 2D views and MVCNN [10] is adopted to extract features from these views. Then, a 3D model is represented by an integrated vector of those view features. A discriminative loss is proposed to learn a class representation for each class by increasing the inter-class distance and reducing the intra-class distance. In the second phase, we use the fine-grained 3D representation to guide the feature learning of 2D sketches by 2DSN. We exploit clustering methods to obtain the fine-grained class information of 3D models. Then, a correlation loss is formulated on 2D sketch features and corresponding fine-grained class representations of 3D models to mine the discrimination of 2D sketches. In this case, sketches are as close as possible to the nearest fine-grained class of 3D models in the embedding space. Comprehensive experimental results on three large-scale benchmark datasets validate the priority of our proposed methods over the state-of-the-art methods.

Our main contributions are as follows:

- We propose a novel sequential learning framework for sketch-based 3D model retrieval to learn representations of 3D models and 2D sketches separately and sequentially. Different from simultaneous learning, our proposed SL framework learns 3D model features without 2D sketches to promote the discrimination of 3D model representations, thus improving the final retrieval accuracy accordingly.
- We consider the fine-grained classes of 3D models to mine the discrimination of 2D sketches. The 2DSN is proposed to learn 2D sketch features with a correlation loss formulated on 2D sketch features and the fine-grained 3D representations calculated by clustering algorithms. Combining fine-grained class information into 2DSN makes the retrieved 3D models more visually similar to the query sketch.
- We achieve superior results than the state-of-the-arts on the threee 3D model retrieval benchmark datasets SHREC 2013 [19], SHREC 2014 [20] and SHREC 2016 [21] datasets.

II. Related Work

Sketch-based 3D models retrieval has drawn more and more attention from computer vision community, with the introduction of large-scale 3D model datasets (such as PSB [22] and ShapeNets [23]) and 3D retrieval contests (such as SHREC 2013 [19], SHREC 2014 [20] and SHREC 2016 [21]). Due to the powerful representation ability, deep learning based methods perform better than traditional methods on many tasks, such as object detection [24], semantic segmentation [25], and 3D model retrieval [26]. In this section, we mainly discuss the sketch-based 3D model retrieval methods based on deep learning algorithms. More specifically, we cover the sketch-based 3D model retrieval methods from the perspective of 3D model processing methods and metric learning.

A. 3D model processing methods

1) Model-based methods: The model-based methods learn shape features directly from the 3D models. PCDNN [13] employs the view-invariant Local Depth Scale-Invariant Feature Transform (LD-SIFT [27]) feature to represent 3D shapes. Then, a pre-trained 3D model neural network is used for these 3D features. [28], [26] use locality-constrained linear coding (LLC) [29] to get a global model descriptor, and then use the deep neural network to transform these raw features into a semantic space. In order to alleviate the large cross-domain discrepancy between sketches and 3D models, [28], [26] propose a deep correlated metric learning method that consists of two parts, discriminative part which mainly addresses the

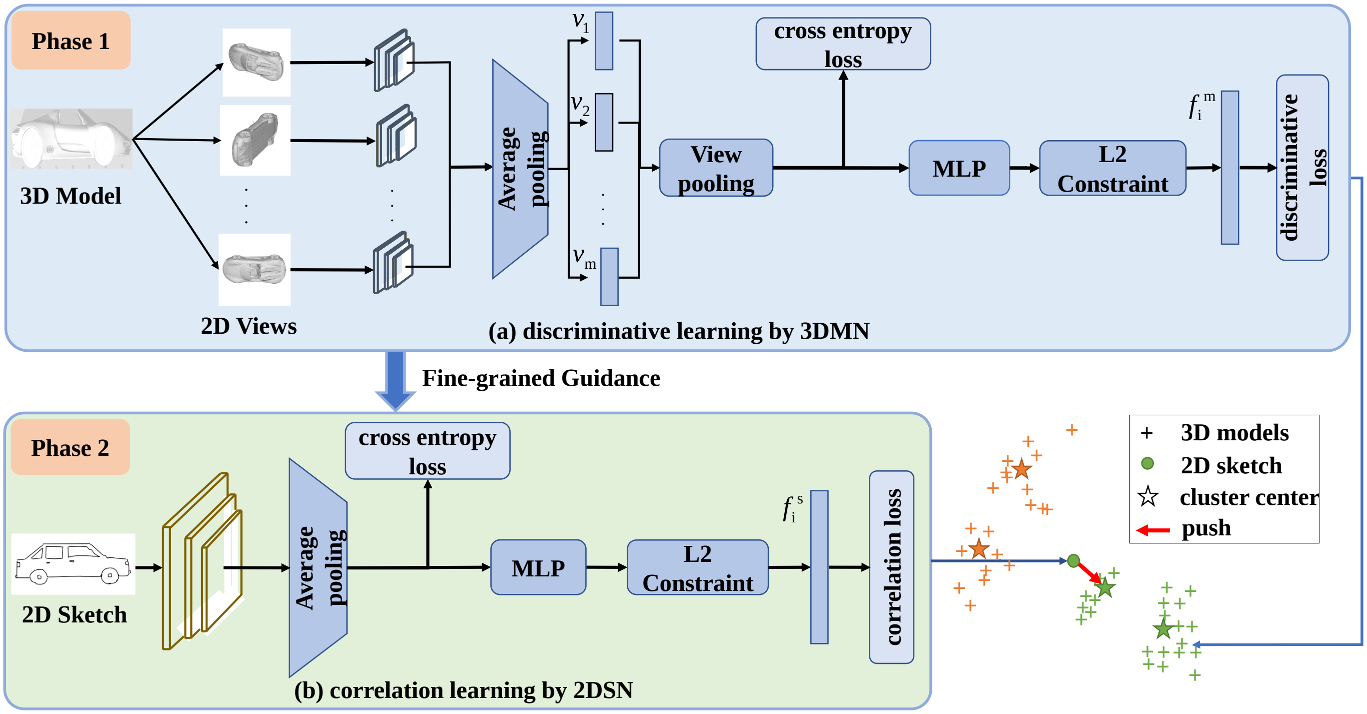


Fig. 2. Overview of the proposed sequential learning (SL) framework for sketch-based 3D model retrieval. By rendering 3D models into multiple views, we extract deep CNN features of 2D projections. 3D models are represented by the pooled views. With the metric network of MLP and L2 Constraint, we formulate a cross entropy loss and a discriminative loss to learn 3D model features in phase 1. By clustering algorithms, we compute the fine-grained 3D representations to guide the feature learning of 2D sketches. We extract deep CNN features of 2D sketches. A cross entropy loss and a correlation loss are formulated to learn 2D sketch feature with the same metric network architecture in phase 2. The 2D sketch features are pushed to the closest subcategory of 3D models. Color represents category.

difference within each domain, and correlation part which solves the difference across domains. These two parts jointly train the neural network to make the feature distributions of two modalities tend to be consistent. [30] employs PointNet [8] to process 3D point clouds, and proposes to project sketches and 3D models into a joint semantic embedding space by classification and triplet ranking losses. In [31], a DLAN network is proposed to combine rotation-invariant 3D local features, which describes local 3D regions of a 3D model by using a set of 3D geometric features. The DLAN network aggregates the set of features of each 3D model into a globally rotation-invariant and compact feature for retrieval task. [32] proposes Kd-network and [7] proposes Oc-tree to work with unstructured point clouds from 3D models. Both of them use the learned features to perform retrieval task. Furuya et al. [31] used normal vectors of the 3D model surfaces as input, and then a voxel-based convolution neural network is designed for this data format. The above methods either construct hand-crafted features for 3D models or automatically learn shallow features directly from the 3D model data formats, such as voxel [15], point clouds [33]. However, methods based on hand-crafted features underperform for lack of discrimination [34], [35]. And methods based on point clouds [30] and voxels [31] are limited by the inefficiency of related technologies or suffer high computational complexity.

2) View-based methods: The view-based methods describe 3D models by a group of projected views, and then use features of these projected views extracted by convolutional neural network (CNNs) for retrieval or classification task. Wang et al. [36] selected two representative views to represent the 3D model, and then exploit Siamese network to extract the features of sketches and views with within-domain and cross-domain similarity loss functions. MVCNN [10] aggregates features from multiple views of a 3D model into a compact model descriptor with a view-pooling layer. Many 3D model retrieval methods based on MVCNN have been proposed. Bai et al. [37] proposed a real-time 3D shape retrieval system. Feng et al. [18] proposed a group-view convolutional neural network (GVCNN) framework for hierarchical correlation modeling towards discriminative 3D shape description. LWBR [38] computes the Wasserstein barycenters of the extracted features from a group of views of 3D models, and then uses these Wasserstein barycenters of 3D models and sketches to learn discriminative 3D model and sketch features for retrieval. DCA [39] combines multi-view features by average view-pooling operation, and introduces adversarial learning for sketches and 3D models. Lei et al. [16] introduced a Representative-View Selection (RVS) module to select the most representative views of a 3D model, and propose DPSML framework to alleviate the difference between sketches and 3D models for retrieval. Nie et al. [11] utilized RNNs to aggregate the extracted deep features of the rendered views. In general, view-based deep learning methods learn more compact features for 3D models and achieve better retrieval performance in most cases [39], [16].

B. Deep metric learning

To alleviate the large cross-domain discrepancy between 2D sketches and 3D models, many deep metric learning based methods have been proposed. Wang et al. [36] used a Siamese network to learn the similarity between samples of 3D models and 2D sketches. Dai et al. [28] proposed a discriminative loss to minimize intra-class distance, maximize inter-class distance within each domain, and optimize the distance across domains simultaneously. Qi et al. [30] and Li et al. [40] proposed metric learning methods with triplet structure to optimize intra-class and inter-class distance jointly. However, these methods jointly learn two deep nonlinear transformations to map the two modalities into a common feature space to solve the intra-domain and inter-domain difference simultaneously and the triplet loss suffers from the problem of time-consuming mining of hard triplets. We propose losses to optimize the intra-domain and inter-domain distances separately and sequentially without requiring additional hard samples mining stage.

Currently, existing sketch-based 3D model retrieval methods [36], [28], [26] address 3D model domain variations and discrepancy between 3D model domain and 2D sketch domain simultaneously in an end-to-end manner. However, there is a huge semantic gap between the 3D models and 2D sketches. In this work, we propose to learn 3D model representations and 2D sketch representations separately and sequentially. Firstly, we learn 3D model representations without the interference of 2D sketches to promote the discrimination of 3D models. Subsequently, the network only focuses on learning 2D sketch features with the guidance of the learned 3D representations of the same class. In addition, existing methods ignore the fine-grained classes of 3D models. We formulate a correlation loss on 2D sketch features and fine-grained 3D model representations obtained by clustering algorithms. At the testing stage, 3D models belonging to the fine-grained class closest to the query sketch are expected to rank at the top of retrieval list.

III. Proposed Method

In this section, we present our novel SL framework for sketch-based 3D model retrieval. Fig. 2 shows the architecture of our proposed method, which consists of two main modules, one for 3D model domain, referred as 3D model network (3DMN), one for sketch domain, referred as 2D sketch network (2DSN). Note that, 3DMN will guide the learning of 2DSN. The proposed method trains 3DMN and 2DSN networks in two phases separately and sequentially.

In the first phase, we train 3DMN with a discriminative loss function to minimize intra-class variations and maximize inter-class variations within 3D model domain. In the second phase, we train 2DSN with a correlation loss on sketch features and learned 3D representations to learn the cross domain similarity. The details for each phase are introduced as follows.

A. Discriminative learning of 3D models

We denote training examples from 3D model domain as $D_{model} = \{(x_i^m, y_i^m)\}_{i=1}^{n_m}$, where x_i^m is a 3D model, $y_i^m \in C = \{1, 2, \ldots, n_c\}$ is the associated label, and n_c denotes the number of categories. These samples are encoded into d-dimensional vectors with a neural network denoted as $f_{\theta_m}(\cdot)$. Note that, directly performing Euclidean distance between feature vectors in high-dimensional space will cause numerical instability. Therefore, we normalize $f_{\theta_m}(x_i^m)$ to get f_i^m using Equation 1.

$$f_i^m = \frac{f_{\theta_m}(x_i^m)}{||f_{\theta_m}(x_i^m)||_2}. \tag{1}$$

3DMN learns to promote the discrimination of 3D model domain to facilitate the subsequent cross domain similarity learning. We propose a discriminative loss which simultaneously minimize the intra-class distance and maximize the inter-class distance without requiring additional hard samples mining stage [41]. Assume that 3D models in the same class share one common class representation, thus, we initialize n_c feature vectors, $\{c_1, c_2, \ldots, c_{n_c}\}$ for each class, where $c \in R^d$ and $||c||_2^2 = 1$. We update the parametric class representation vectors based on a batch of data at each iteration without designing sample pairs. Given a batch of training data with N_m samples, the discriminative loss $\mathcal{L}_d$ is defined as:

$$\mathcal{L}_d = \sum_{i=1}^{N_m} \max(0, m + D(f_i^m, c_{y_i^m}) - \min_{j \in C, j \neq y_i^m} D(f_i^m, c_j)), \tag{2}$$

where N_m is batch size, m is a predefined margin and $D(\cdot)$ is the cosine distance function denoted as:

$$\begin{aligned} D(f(x_i^m), c_{y_i^m}) &= 1 - \frac{< f_i^m, c_{y_i^m} >}{||f_i^m||_2 ||c_{y_i^m}||_2} \\ &= 1 - < f_i^m, c_{y_i^m} >. \end{aligned} \tag{3}$$

The learning objective of $\mathcal{L}_d$ is to pull samples closer to their corresponding class representation vector $c_{y_i^m}$ than the nearest class representation vector c_j where $j \neq y_i^m$. Since the numerical value of $\mathcal{L}_d$ is very small, the extracted 3D model features in the same class will all approach the class representation vector. In this case, there is no discrimination among samples in the same class, which is detrimental for mining fine-grained categories. Therefore, we introduce the cross entropy loss $\mathcal{L}_s$ to accommodate large intra-class variations and train the 3DMN networks with joint supervision of $\mathcal{L}_s$ and $\mathcal{L}_d$. The overall loss of 3DMN is given in Equation 4.

$$\begin{aligned} \mathcal{L}_{3DMN} &= \mathcal{L}_s + \lambda \mathcal{L}_d \\ &= -\sum_{i=1}^{N_m} \log \frac{e^{W_{y_i}^T f_{\theta_m}(x_i^m) + b_{y_i}}}{\sum\limits_{j=1}^{|C|} e^{W_j^T f_{\theta_m}(x_i^m) + b_j}} + \lambda \mathcal{L}_d, \end{aligned} \tag{4}$$

where λ is a hyper-parameter used to balance $\mathcal{L}_s$ and $\mathcal{L}_d$. We conduct experiments to illustrate how λ affects the learning of 3DMN networks.

B. Correlation Learning for 2D sketches

In this subsection, we present to learn discriminative sketch features for 3D model retrieval. Since 2D sketches and 3D models come from two different modalities with great visual discrepancy. Even 3D models and 2D sketches are projected into the same embedding space, the gap of feature distribution

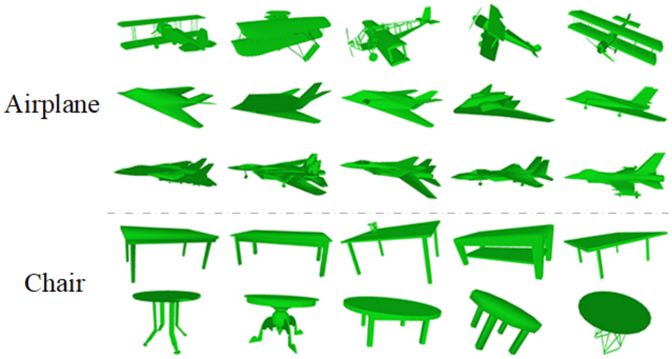


Fig. 3. An illustration of fine-grained classes in 3D models. Airplane and Chair are two model classes from the SHREC 2013 dataset. According to visual similarity, Airplane and Chair have three and two fine-grained classes, respectively.

between them is still significant. In order to guarantee the effectiveness of cross-domain retrieval, we should ensure that the feature distributions of two domains are as consistent as possible.

The training examples of 2D sketches are denoted as $D_{sketch} = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, where x_i^s is a sketch, $y_i^s \in C$ is the corresponding label. Note that, 2D sketches and 3D models share the same label space. Sketches are embedded into d-dimensional vectors by 2DSN network denoted as $f_{\theta_s}(\cdot)$. Here, we also use Equation 1 to normalize the $f_{\theta_s}(x_i^s)$ to get $f_i^s \in R^d$.

The learning objective of 2DSN networks is to align the feature distribution of 2D sketches with that of 3D models. Most recent works use triplet loss to optimize the intra-class and inter-class distance of sketch-model pairs. However, training with triplet loss is subjected to the carefully designed hard sample mining strategy and suffers the high computation cost of triplets. Considering the overall class information, each class of 3D model can be represented by a class representation vector as $c_{y_i^s}$. When training 2DSN network, we impel the sketch x_i^s to approach the learned 3D class representation vector $c_{y_i^s}$ by 3DMN. Given a batch of training data with N_s samples, we define correlation loss $\mathcal{L}_c$ as:

$$\mathcal{L}_c = \sum_{i=1}^{N_s} D(f_i^s, c_{y^i}). \tag{5}$$

As we observe, a query sketch is visually similar with only a part of 3D models in a class when subcategories exists. Different from previous methods which match 2D sketches with all 3D modelss in a same class, we further consider fine-grained classes of 3D models (see Fig. 3). Using clustering algorithms such as K-Means, we calculate multiple fine-grained class centers for each class, denoted as $\{r_1^i, r_2^i, \ldots, r_{k_i}^i\}$, where $r \in R^d$ is a cluster center of class $i \in C$, k_i denotes the number of fine-grained categories in class i. Given a batch of N_s samples, we define correlation loss $\mathcal{L}_c$ as:

$$\mathcal{L}_c = \sum_{i=1}^{N_s} \min_{j \in [1, k_{y_i^s}]} D(f_i^s, r_j^{y_i^s}). \tag{6}$$

The intuition of this objective function is that if the sketch feature is close to a fine-grained class center, there is a high probability that the sketch belongs to the fine-grained class. If we only supervise 2DSN networks by $\mathcal{L}_c$, the deeply learned features will be close to each other and lack fine-grained discrimination, which causes visual dissimilarity of retrieval results. So we also combine softmax loss $\mathcal{L}_s$ with $\mathcal{L}_c$ to jointly supervise the 2DSN network. The overall objective function for 2DSN networks is given in Equation 7.

$$\begin{aligned} \mathcal{L}_{2DSN} &= \mathcal{L}_s + \alpha \mathcal{L}_c \\ &= -\sum_{i=1}^{N_s} \log \frac{e^{W_{y_i^s}^T f_{\theta_s}(x_i^s)} + b_{y_i^s}}{\sum_{j=1}^{|C|} e^{W_j^T f_{\theta_s}(x_i^s)} + b_j} + \alpha \mathcal{L}_c, \end{aligned} \tag{7}$$

where α is a trade-off parameter between $\mathcal{L}_s$ and $\mathcal{L}_c$.

C. Network architecture.

Both 3DMN and 2DSN adopt AlexNet [42] or ResNet50 [43] as CNN backbone for feature extraction. For AlexNet, we follow the same network architecture as [28]. For ResNet50, the layers before pooling5 layer (inclusive) are remained. Using the ReLU activation function [43], all the extracted features are distributed in the same quadrant, which might cause the CNNs to be difficult to learn. We use two fully connected layers (fc) with LeakyReLU activation function (fc(2048,2048)->BN->LeakyReLU->fc(2048,2048)) to make the feature distribution more reasonable. Moreover, we perform L2 regularization on the features as L2 constraint to make the values more stable. The 3DMN network is built upon the framework of MVCNN [10] which combines a series of 2D projection views to the CNNs in an end-to-end training manner. Following MVCNN, we employ Phong reflection model [17] to render a 3D model into n views representation as $V = \{v_i, 1 \leq i \leq n, v_i \in R^{H \times W \times 3}\}$, where $H \times W$ is the size of views.

IV. Experimental Setups

We conduct extensive experiments on three large-scale benchmark datasets, i.e., the SHREC 2013 [19], SHREC 2014 [20] and SHREC 2016 [21] sketch track benchmark datasets. We first introduce the experimental settings, including detailed description of above benchmark datasets, evaluation metrics and implementation details. Then, we conduct ablation experiments to demonstrate the effectiveness of the proposed method. Next, we calculate seven metrics to investigate the retrieval performance and compare our results against the state-of-the-arts. We also give detailed analysis of the normalization module and hyper-parameters including λ, m and α. Finally, we visualize the distribution of our learned sketch and 3D shape features, retrieval results of our proposed method.

A. 3D model retrieval datasets

SHREC 2013 [19] is a large-scale benchmark dataset for sketch-based 3D model retrieval. This dataset collects 7200 sketches from human-drawn sketch dataset [44] and 1258 models from Princeton Shape Benchmark [22], divided into

90 classes. The number of sketches in each class is equal, and there are 80 in each class. The number of 3D models in each class ranges from a minimum of 4 to a maximum of 184. Refer to [19], we use 50 sketches from each class for training and the remaining 30 sketches per class for testing, while the 1258 relevant models are remained as the target for retrieval.

SHREC 2014 [20] is a much larger scale benchmark dataset, which is composed of 13680 sketches and 8987 3D models from 171 classes. The number of sketches of each class is also equal to 80. The number of 3D models in each class varies from 1 to 632. Following [19], we use 50 sketches of each class for training and 30 sketches for testing. All of the 3D models are used for retrieval.

SHREC 2016 [21] is a new benchmark dataset, which takes hand-drawn 3D sketches as queries for 3D model retrieval. This dataset is composed of 300 3D sketches collected by a Microsoft Kinect device, and 1258 3D models acquired from SHREC 2013 dataset. The sketches are evenly divided into 30 classes, among which only 21 have corresponding 3D models. For each class, there are 10 sketches, where 7 sketches are used for training and 3 sketches for testing.

B. Evaluation metrics

Following the state-of-the-arts, we employ seven widely used performance metrics [19] in our experiments. They are Precision-Recall curve (PR curve), Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), E-measure (E), Discounted Cumulated Gain (DCG) and mean Average Precision (mAP).

1) PR: Precision-Recall curve is the most common metric to evaluate sketch-based 3D model retrieval. Precision is the ratio of retrieved 3D models that are relevant to all retrieved 3D models in the ranked list. Recall is the ratio of retrieved 3D models that are relevant to all relevant 3D models.

2) NN, FT, ST: These three evaluation metrics check the ratio of 3D models in the query's class that also appear in the ranked list within the length of K, where K can be 1, the size of the query's class, or the double size of the query's class. Specifically, for a class with $|C|$ members, K=1 for NN, K=$|C|$-1 for FT, and K=2*($|C|$-1) for ST.

3) E: F-Measure is the weighted harmonic mean of precision and recall. The E-Measure is defined as E = 1- F. Note that, its maximum value is 1.0, and higher E value indicate better retrieval performance.

4) DCG: The relevance level of each 3D model is used as a gain value measures for its ranked position, the gain is summed progressively from position 1 to n. The discounted cumulated gain vector DCG reduces the 3D model weight as its rank increases.

5) mAP: Average precision (AP) is the area under the precision-recall curve. The AP is then averaged over all classes to get mAP.

C. Implementation details

We implement the proposed method based on the PyTorch framework [45] with a NVIDIA GeForce GTX 1060 GPU. In our proposed method, we obtain 12 rendered views for each 3D model as [17]. The deep CNN features are extracted from L2 constraint layer for both 2DSN and 3DMN networks. In practice, we are limited by GPU memory, which is not large enough to hold large batch of data. In the experiments, we adopt the SGD with momentum [46] of 0.9 for optimization and set mini-batch size to be 32 for 3DMN and 16 for 2DSN, respectively. We conduct our experiments for total 90 epochs with cosine learning rate [47] for all experiments. The initial learning rate is set to be 1e-3 and 1e-2 for optimizing backbone and other layers respectively. A weight decay of 1e-4 is used. We randomly initialize the class representation features of each 3D model class and the learning rate of these features is set to be 0.1 for our experiments. λ, m and α are set to be 12, 1.5, and 28, respectively. In this work, we adopt K-Means to process the learned 3D model features. For classes in SHREC 2013 dataset, we set K to be 2, 3, 4, and 5 with the minimum 3D model number of 8, 24, 64, and 160. For SHREC 2014 dataset, we set K to be 2, 3, 4, and 5 with the minimum 3D model number of 16, 48, 128, 320. The setting of K for SHREC 2016 is same with SHREC 2013 for the same source of 3D models.

V. Experimental results

A. Evaluation of the proposed method

We conduct some ablation experiments to illustrate the effectiveness of our proposed SL framework and fine-grained representations. We evaluate the methods on SHREC 2013 and exploit AlexNet as the CNNs backbone. First, we extract features of 3D models and 2D sketches using 3DMN and 2DSN and train them with a triplet loss end-to-end, which is set as the baseline. Then, we adopt the SL framework to train 3DMN and 2DSN networks separately and sequentially, which achieves performance improvement in terms of all metrics. To promote the discrimination of 3D model features, we train 3DMN with $\mathcal{L}_d$ in Equation 4, which parameterize the class representation of each class. We can exploit the learned class representations as coarse-grained class information to guide the representation learning of 2DSN with $\mathcal{L}_c$ in Equation 5, considering a possible scenario that no classes of 3D models have subclasses. Further, we introduce fine-grained representation learning of 2D sketch features to address a more practical scenario that most classes have subclasses. 3D models and 2D sketches belonging to different subcategory differs in visual appearance such as shape and texture. We use K-Means to obtain the fine-grained class representations and train 2DSN with $\mathcal{L}_c$ in Equation 6. As shown in Table I, BL+SL outperforms BL, and both BL+SL+C and BL+SL+C+F outperform BL+SL in terms of all evaluation metrics. Note that, the SHREC 2013 datesets contains 90 classes of 3D models, few of which have subclasses due to limited samples. Therefore, we also conduct experiments on SHREC 2014 dataset which have more samples and classes. As shown in Table I, the SL+C+F(SHREC 2014) explicitly surpass the SL+C(SHREC 2014) in terms of all evaluation metrics. Ablation experiments indicate that the proposed method has learned discriminative features of 3D models and 2D sketches, and efficiently align these features across two domains, effectively improving the retrieval accuracy.

TABLE I

THE ABLATION RETRIEVAL RESULTS OF DIFFERENT VARIANTS OF PROPOSED METHOD ON THE SHREC 2013 AND SHREC 2014 DATASETS. "BL" REFERS TO THE BASELINE METHOD TRAINING 3DMN AND 2DSN END-TO-END; "SL" REFERS TO PROPOSED SEQUENTIAL LEARNING STRATEGY; "C" REFERS TO THE COARSE-GRAINED REPRESENTATIONS; "F" REFERS TO PROPOSED FINE-GRAINED REPRESENTATIONS. THE BEST RESULTS ARE IN BOLD FONT.

Methods	Backbones	NN	FT	ST	E	DCG	mAP
BL	AlexNet	65.74	64.81	76.64	37.26	78.50	69.63
BL+SL	AlexNet	70.90	67.50	78.50	38.40	80.40	71.60
BL+SL+C	AlexNet	**78.30**	**80.30**	84.20	39.40	**85.50**	**81.70**
BL+SL+C+F	AlexNet	76.70	79.52	**84.48**	**39.42**	85.20	81.10
BL+SL+C(SHREC 2014)	ResNet50	78.00	**80.40**	84.40	41.50	87.10	81.90
BL+SL+C+F(SHREC 2014)	ResNet50	**78.10**	**80.40**	**84.70**	**41.53**	**87.20**	**82.00**

TABLE II

THE ABLATION CLASSIFICATION RESULTS (MCA, %) FOR 3D MODELS AND RETRIEVAL RESULTS (MAP, %) OF BASELINE METHOD AND DIFFERENT VARIANTS OF PROPOSED METHOD ON SHREC 2013 BENCHMARK DATASET. THE BEST RESULTS ARE IN BOLD FONT.

Methods	MCA(%) 3D	mAP(%)
BL	60.57	69.63
BL+SL	66.02	71.60
BL+SL+C	**69.67**	**81.70**

Better 3D model representation, better 3D model retrieval. We assume that the more discriminative 3D representation, the more accurate the 3D model retrieval. To verify this assumption, we conduct experiments on SHREC 2013 datasets, and compare the classification and retrieval performance of BL, BL+SL, and BL+SL+C using AlexNet as the CNNs backbone for feature extraction. Specifically, BL+SL learns better 3D model representation than BL with a gain of 5.45 % (i.e., 66.02% versus 60.57) in terms of mean class accuracy (MCA) as shown in Table II. For 3D model retrieval, BL+SL outperforms BL and achieves an increase of about 2% in terms of mAP. Furthermore, BL+SL+C obtains a gain of 3.65% in terms of MCA for 3D models over BL+SL, and achieves an increase of 10.1% in terms of mAP. As the discrimination of 3D model representation improve, the final retrieval accuracy increases. The experimental results demonstrate that the discrimination of 3D representation is a vital factor of 3D model retrieval.

B. Comparison with state-of-the-art methods

In this section, we evaluate our proposed method on SHREC 2013 [19], SHREC 2014 [20] and SHREC 2016 [21] datasets and compare it with the state-of-the-art methods. For SHREC 2013 dataset, we consider coarse-grained class representations for 3D models. For SHREC 2014 and SHREC 2016 datasets, we further consider the fine-grained classes of 3D models to mine the discrimination of 2D sketch features.

1) Evaluation on the SHREC 2013 dataset: A quantitative comparison of the proposed method with the state-of-the-arts is shown in Fig. 4 based on PR curves. For a fair comparison, we conduct experiments based on different backbones (e.g., AlexNet and ResNet50) according to [39] and [16]. It can be seen that, the precision value of the proposed method steadily exceeds the state-of-the-art methods while the recall value increases from 0 to 1. Note that, the precision value of the proposed method increases consistently, while the precision value of other methods decreases especially when the recall value is bigger than 0.4. It demonstrates that, given a query sketch, our method obtains all relevant 3D models within less retrieval items than other methods. This is because that our method guides 2D sketches to approach the most similar 3D models in an implicit fine-grained class, while other methods pull sketches close to all 3D models in the whole class. Compared with the most recently published works [13], [26], [39], [16], our proposed method achieve superior retrieval performance based on the CNN backbones of both AlexNet and RetNet50.

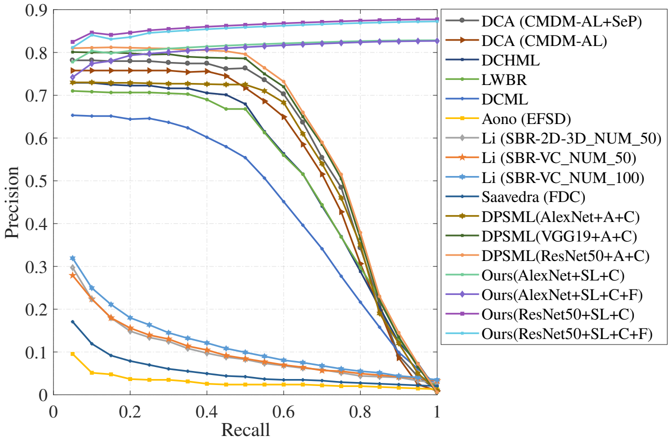


Fig. 4. Precision-recall comparisons on the SHREC 2013 dataset.

Table III shows a quantitative comparison of the proposed method with the state-of-the-art methods on six standard evaluation metrics, including NN, FT, ST, E DCG and mAP. It can be observed that the proposed method outperforms other state-of-the-art methods for all evaluation metrics. Considering that few classes contain fine-grained categories in SHREC 2013 dataset, we implement the "S+C" method. Experimental results on SHREC 2013 benchmark dataset have verified the effectiveness of the proposed method.

2) Evaluation on the SHREC 2014 dataset: In this section, we evaluate the proposed method on the SHREC 2014 dataset which is more challenging than the SHREC 2013 for more classes and larger variations within each class. For a fair comparison experiments, we also implement our proposed method based on two CNN backbones (e.g., AlexNet and ResNet50) as the works [39], [16].

TABLE III
RETRIEVAL RESULTS ON THE SHREC 2013 DATASET. THE BEST RESULTS ARE IN BOLD FONT.

Methods	Backbones	NN	FT	ST	E	DCG	mAP
CDMR[48]	-	27.90	20.30	29.60	16.60	45.80	25.00
SBR-VC[19]	-	16.40	9.70	14.90	8.50	34.80	11.60
Siamese[36]	-	40.50	40.30	54.80	28.70	60.70	46.90
DCML[28]	AlexNet	65.00	63.40	71.90	34.80	76.60	67.40
LWBR[38]	AlexNet	71.20	72.50	78.50	36.90	81.40	75.20
DCHML[26]	AlexNet	73.00	71.50	77.30	36.80	81.60	74.40
DCA[39]	ResNet50	78.30	79.60	82.90	37.60	85.60	81.30
DPSML[16]	AlexNet	74.10	76.10	82.10	38.50	83.60	78.50
DPSML[16]	ResNet50	81.90	83.40	87.50	41.50	89.20	85.70
Ours(SL+C)	AlexNet	78.30	80.30	84.20	39.40	85.50	81.70
Ours(SL+C)	ResNet50	**82.70**	**84.10**	**89.50**	**41.80**	**89.50**	**86.40**

TABLE IV
RETRIEVAL RESULTS ON THE SHREC 2014 DATASET. THE BEST RESULTS ARE IN BOLD FONT.

Methods	Backbones	NN	FT	ST	E	DCG	mAP
CDMR[48]	-	10.90	5.70	8.90	4.10	32.80	5.40
SBR-VC[20]	-	9.50	5.00	8.10	3.70	31.90	5.00
Siamese[36]	-	23.90	21.20	31.60	14.00	49.60	22.80
DCML[28]	AlexNet	27.20	27.50	34.50	17.10	49.80	28.60
LWBR[38]	AlexNet	40.30	37.80	45.50	23.60	58.10	40.10
DCHML[26]	AlexNet	40.30	32.90	39.40	20.10	54.40	33.60
DCA[39]	ResNet50	77.00	78.90	82.30	39.80	85.90	80.30
DPSML[16]	AlexNet	67.70	73.20	79.50	37.90	83.00	75.10
DPSML[16]	ResNet50	77.40	79.80	**84.90**	41.50	**87.70**	81.30
Ours(SL+C)	AlexNet	72.30	74.10	77.00	37.90	81.30	75.10
Ours(SL+C)	ResNet50	78.00	**80.40**	84.40	41.50	87.10	81.90
Ours(SL+C+F)	AlexNet	71.60	75.70	77.80	37.80	82.40	75.70
Ours(SL+C+F)	ResNet50	**78.10**	**80.40**	84.70	**41.53**	87.20	**82.00**

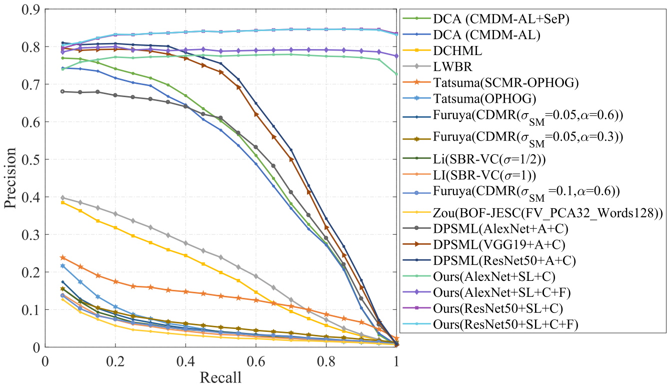


Fig. 5. Precision-recall comparisons on the SHREC 2014 dataset.

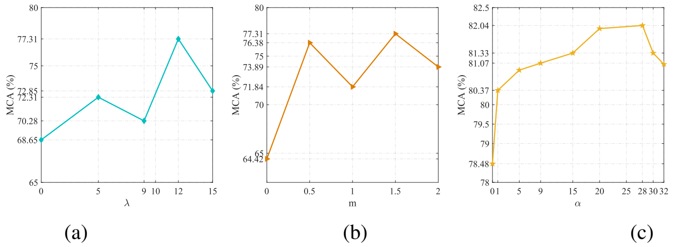
Fig. 5 illustrates the PR curves of the proposed method and comparison with the state-of-the-art methods. It can be seen that our proposed method significantly outperforms the state-of-the-art methods. More specially, the precision value of our proposed method (both SL+C and SL+C+F) using AlexNet as CNN backbone always keeps higher than that of one most recent work [16]. Moreover, when the deeper CNN backbone is applied, the proposed method still surpass the work [16]. In addition, as the recall value increases from 0 to 1, the precision value of the proposed method maintains an ascending trend and only has a slight decrease when the recall value is great than 0.95. The PR curves have verified the superior performance and robustness of the proposed methods.

Table IV provides a quantitative comparisons of the proposed method with the state-of-the-art methods on the SHREC 2014 benchmark dataset. We evaluate the proposed method on six metrics, including NN, FT, ST, E, DCG, and mAP. It can be observed that the proposed method outperforms the state-of-the-art methods on NN, FT, E, and mAP, and has a comparative results on ST and DCG. More specially, the proposed method surpasses the most recent work DPSML [16] in terms of the metric NN with a gain of 4.6% (i.e., 72.30% versus 67.70%) adopted AlexNet as the backbone and 0.7% adopted ResNet50 as the backbone. Generally, people are more likely to examine top retrieved objects, instead of latter one, due to time and efforts. For the SHREC 2014 dataset, ST checks the ratio of models in the query's class that also appear within the top K matches and K is always 20. The ST value of our proposed method degrades for some classes with few examples. The appearance similarity of hard categories (e.g., barn and house, car_sedan and suv) might account for the slight difference of DCG value of the proposed method and DPSML [16]. It can be observed that the retrieval performances of "SL+C+F" is better than that of "SL+C". This is because that there are more classed of 3D models in SHREC 2014, and more classes contains fine-grained subcategories. The experimental results on SHREC 2014 validate the effectiveness of our proposed method, especially the fine-grained discrimination mining.

3) Evaluation on the SHREC 2016 dataset: This section presents a comparison of the proposed method with the state-of-the-art methods on the SHREC 2016 dataset. For adopting 3D sketches as queries, the SHREC 2016 dataset is more challenging than both SHREC 2013 and SHREC 2014 dataset-

TABLE V
THE RETRIEVAL RESULTS ON THE SHREC 2016 DATASET. THE BEST RESULTS ARE IN BOLD FONT.

Methods	Backbone	NN	FT	ST	E	DCG	mAP
Siamese[36]	-	0.00	3.10	10.80	4.80	29.30	7.20
CNN-SBR[21]	-	22.20	25.10	32.00	28.60	47.10	31.40
DCHML[26]	AlexNet	11.70	10.60	14.80	8.60	32.70	14.70
DPSML[16]	AlexNet	42.90	47.80	56.30	27.90	60.90	49.90
DPSML[16]	VGG19	47.60	51.00	57.20	29.00	64.00	53.30
Ours(SL+C)	AlexNet	**63.49**	66.57	72.33	**33.62**	76.80	69.10
Ours(SL+C+F)	AlexNet	58.37	**66.90**	**76.91**	31.57	**77.40**	**69.57**



(a) (b) (c)

Fig. 6. The mean class accuracy (MCA) of the proposed method versus λ and α when testing on the SHREC 2013 dataset. (a) varying λ when m is fixed to 1.5, (b) varying m and (c) varying α.

s. Moreover, 3D sketches are represented with sparse point clouds, which are more abstract than 2D sketches.

Only few works [36], [21], [26], [16] are evaluated on the SHREC 2016. For fairness, we employ the front view of 3D sketch as its representation following [26]. We evaluated the proposed method on NN, FT, ST, E, DCG and mAP. The corresponding quantitative comparison is illustrated in Table V. The proposed method outperforms the state-of-the-art methods in terms of all metrics. Specifically, the proposed learned method significantly surpasses the most recent work [16] with gain of 20.59 % in terms of the metric NN when AlexNet is applied as backbone. Note that, our proposed method achieves better retrieval performance than [16], even they use a deeper backbone of VGG19 [49]. The retrieval performance on the SHREC 2016 dataset further demonstrate the effectiveness of the proposed method and generalization ability to 3D sketch-based 3D model retrieval task. Without interference from noisy 3D point cloud sketch, a better 3D model representation can be learned.

C. Detailed analysis of proposed method

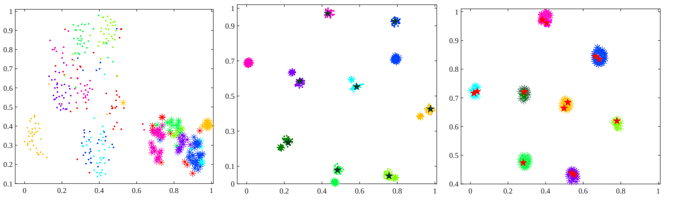
We have conducted some experiments to evaluate the effectiveness of different hyper-parameters and the normalization module in proposed method. To select the optimal hyper-parameters, we randomly choose 25% samples of 3D models and 2D sketches from SHREC 2013 dataset as the validation set for the following contrast experiments. Note that, we validate hyper-parameters with the classification results of 3D models or 2D sketches measured by the mean class accuracy (MCA). We validate the effectiveness of the normalization module with the retrieval results.

Effect of hyper-parameters α, λ and m. As described in Equation 4 and Equation 7, both the overall loss of proposed 3DMN and 2DSN contains two terms. The hyper-parameters α and λ balance the associated loss functions, respectively. In Equation 4, there are two hyper-parameters λ and m. We set $m = 1.5$ and vary λ from 0 to 15 to learn different models. The effect of λ is shown in Fig. 6 (a). Experimental results show that supervising the model learning using only $\mathcal{L}_s$ loss ($\lambda = 0$) leads to poor verification performance. Introducing the $\mathcal{L}_d$ loss can improve the verification accuracy. The $\mathcal{L}_s$ loss focuses on how to map 3D model features to discrete labels, and the $\mathcal{L}_d$ learns features directly that can enforce 3D model feature to be embedded into more compact spaces faster. In order to improve the generalization ability of the network, we introduce m into the $\mathcal{L}_d$ loss. It can be seen from Fig. 6 (c) that when m is not equal to 0, the network has a strong generalization ability. We also conduct many validation experiments on α. We vary α from 0 to 32, where $\alpha = 0$ obtains the worst performance when only softmax loss is used to learn the mapping from features to labels. However, the $\mathcal{L}_c$ loss pay more attention to the learning between feature domains when the label space is fixed. Since we use the ”L2 constraint” module, we can set a larger value for α and λ, which can prompt the network to learn more discriminative features and avoid gradient explosion.

TABLE VI
THE CLASSIFICATION RESULTS (MCA, %) AND RETRIEVAL SPEED (MS) OF PROPOSED SL+C AND SL+C+F. THE BEST RESULTS ARE IN BOLD FONT.

Methods	MCA(%) 3D	MCA(%) 2D	Speed(ms)
SL+C (w/o norm)	70.95	82.00	28.60
SL+C	**77.37**	82.04	**4.40**
SL+C+F (w/o norm)	70.95	82.33	28.60
SL+C+F	**77.37**	**83.48**	**4.40**

Effect of normalization. It can be seen from Table VI that the ”L2 constraint” module can significantly improve the classification performance. In Fig. 2, after the features passes the ”MLP” module, it will be constrained by Equation 1. After the ”L2 constraint” module, features are normalized to the unit hypersphere with a maximum distance of 2 between features. So we can conveniently get a more reasonable value for m in Equation 2. We set $m = 1.5$ in all experiments. Using this module can get about 7% performance improvement in terms of mean class accuracy (MCA) on 3D model dataset as shown in Table VI. As for retrieval speed, after the ”L2 constraint” module, using the cosine distance to calculate the similarity between features is equivalent to the scalar product. From Table VI, we can see that the normalized retrieval speed is nearly 7 times faster than that without normalization.



(a) pretrained CNNs (b) SL+C (c) SL+C+F

Fig. 7. A visualization of learned features from 8 classes (i.e., sword, race car, sailboat, pickup truck, brain, floor lamp, door and church) of sketches and 3D models in SHREC 2013 benchmark dataset using pretrained CNNs (a) SL+C (b) and SL+C+F (c). Star represents 3D models, and dot represents 2D sketches. Color indicates classes. Green pentacle is class representation and red pentacle is fine-grained class representation.

D. Qualitative results

Visualization of learned representations. We adopt t-SNE [50] to visualize the learned features from 8 classes (i.e., sword, race car, sailboat, pickup truck, brain, floor lamp, door and church) of 2D sketches and 3D models in SHREC 2013 benchmark dataset using pretrained CNNs on ImageNet [51], ”SL+C”, and ”SL+C+F”. Fig. 7(a) illustrates features of 2D sketches and 3D models extracted by pretrained CNNs. It can be seen that the features of two modalities are far from each other and features of different classes in each modality are mixed together. Fig. 7 (b) shows the features by our proposed ”SL+C” method. It can be observed that the 3D model features learned by 3DMN are divergent to each other, and the sketch features learned by 2DSN gather around the learned class representation. Although the class representation of sword is a little far from 3D models due to visually similar to other classes, other learned class representations are very close to corresponding 3D models. Fig. 7 (c) illustrates that the 3D model and 2D sketch features learned by the proposed ”SL+C+F” are obviously divided into eight compact clusters. The fine-grained class representation calculated by clustering algorithms are more accurate than the learned class representation. As shown in Fig. 7 (c), the feature distribution of 2D sketches learned by 2DSN are almost consistent to that of 3D models. Comparing the learned features of Fig. 7(b) and (c), we can conclude that combining fine-grained class information into the training of 2DSN is beneficial to align the feature distribution of 2D sketches with that of 3D models, which avails the 3D model retrieval.

Visualization of retrieval results. The top 10 retrieval results of 7 query sketches from the SHREC 2013 dataset are showed in Fig. 8. The left column is the query sketches (i.e., dog, ice cream cone, pickup truck, potted plant, and shovel) and the right column is the corresponding top 10 retrieval 3D models with rank score from 1 to 10. The correct 3D models are marked in green color and the incorrect ones are marked in yellow. For the classes with abundant examples like ice cream cone and potted plant, all 10 retrieval 3D models are correct, while for the classes with limited examples like bicycle, dog, pickup truck, and shovel, the first few retrieved models are correct and the last few ones are incorrect. Note that, there are only 7, 7, 8, and 5 examples provided for the classes of bicycle, dog, pickup truck, and shovel, which means all correct 3D models are retrieved by our proposed method for these classes, either. Specifically, the retrieval results are obtained by ”SL+C” using the CNN backbone of AlexNet as some recent works [26], [13], [39], [16] whose top 10 retrieval results of above classes include incorrect or visually dissimilar 3D models. Note that, the relevant 3D models of query sketches are retrieved and rank in front of the retrieval list, which is an intuitive demonstration of the aforementioned PR curves.

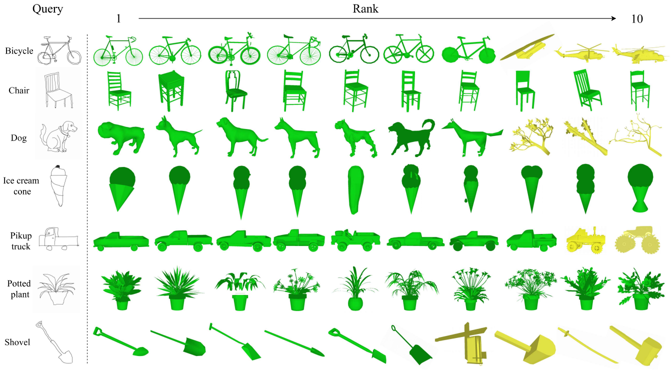


Fig. 8. Top 10 retrieved models of query sketches (i.e., bicycle, chair, dog, ice cream cone, pickup truck, potted plant, and shovel) by the proposed SL+C method on the SHREC 2013 dataset. Query sketches are listed on the left and the retrieved 3D models are on the right. Note that, the correct retrieval are marked as green and the incorrect ones are marked as yellow.

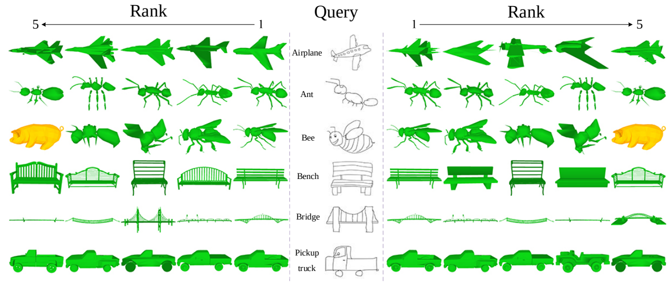


Fig. 9. The top 5 retrieval results of the proposed SL+C+F (left) and SL+C (right) on the SHREC 2013 dataset. The correct 3D models are marked with green color and the incorrect ones with yellow color. The retrieval results are sorted by visual similarity with rank from 1 to 5.

Fig. 9 shows the top 5 retrieval results of 6 sketches from the SHREC 2013 dataset using the ”SL+C” and ”SL+C+F” method. 3D models in green color are correct retrieval results, and that in yellow color are incorrect. The rank score indicates the relevance to the query sketch and the retrieved 3D model with rank 1 is most relevant. Note that, there are only 4 3D models of class ”Bee” in SHREC 2013 dataset. It can be seen that the retrieved 3D models using ”SL+C+F” are more visually similar to the query sketch than that of ”SL+C”. Fig. 9 shows that the top 5 retrieval results of the ”SL+C+F” remains more texture and shape information for Airplane, Bench, and Bridge.

VI. Conclusion

WeproposeanovelSLframeworkforsketch-based 3Dmodelretrieval.Different from simultaneous learning, the SL framework learns representations of 3D models and 2D sketches in two phases sequentially. In the first phase, we render 3D models to multiple views, extract the deep features of these views, and then learn features of 3D models by formulating a discriminative loss only on 3D model features. Without the interference of 2D sketches, the discrimination of 3D model feature is greatly promoted, which is demonstrated to be vital for 3D model retrieval. In the second phase, we learn features of 2D sketches by the guidance of learned 3D features. We further consider the fine-grained classes of 3D models and exploit clustering algorithms to obtain the representations of these fine-grained classes. A correlation loss is formulated on sketch features and the fine-grained 3D model representations to obtain better 2D sketch features. We demonstratetheeffectivenessof our proposed SL framework and fine-grained 3D representations on threewidely-used large-scale datasets (i.e., SHREC 2013, SHREC 2014, and SHREC 2016 datasets) and achieve superior retrieval performance over the state-of-the-arts.

VII. Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61976038, 61932020, and 61772108.

References

[1] E. Peed and N. Lee, "3d printing, history of," in *Encyclopedia of Computer Graphics and Games*, 2019. [Online]. Available: https://doi.org/10.1007/978-3-319-08234-9_279-2

[2] M. Furukawa, Y. Akagi, Y. Kawai, and H. Kawasaki, "Interactive 3d animation creation and viewing system based on motion graph and pose estimation method," in *Proceedings of the ACM International Conference on Multimedia, MM '14, Orlando, FL, USA, November 03 - 07, 2014*, 2014, pp. 1213–1216. [Online]. Available: https://doi.org/10.1145/2647868.2655055

[3] M. H. Hesamian, W. Jia, X. He, and P. J. Kennedy, "Deep learning techniques for medical image segmentation: Achievements and challenges," *J. Digital Imaging*, vol. 32, no. 4, pp. 582–596, 2019. [Online]. Available: https://doi.org/10.1007/s10278-019-00227-x

[4] S. Saravi, D. Joannou, R. Kalawsky, M. R. N. King, I. P. Marr, M. Hall, P. C. J. Wright, R. Ravindranath, and A. Hill, "A systems engineering hackathon - A methodology involving multiple stakeholders to progress conceptual design of a complex engineered product," *IEEE Access*, vol. 6, pp. 38 399–38 410, 2018. [Online]. Available: https://doi.org/10.1109/ACCESS.2018.2851384

[5] R. E. Banchs, "A comparative evaluation of 2d and 3d visual exploration of document search results," in *Information Retrieval Technology - 10th Asia Information Retrieval Societies Conference, AIRS 2014, Kuching, Malaysia, December 3-5, 2014. Proceedings*, 2014, pp. 100–111.

[6] S. Kawamura, K. Usui, T. Furuya, and R. Ohbuchi, "Local goemetrical feature with spatial context for shape-based 3d model retrieval," in *Eurographics Workshop on 3D Object Retrieval 2012, Cagliari, Italy, May 13, 2012. Proceedings*, 2012, pp. 55–58.

[7] P. Wang, Y. Liu, Y. Guo, C. Sun, and X. Tong, "O-CNN: octree-based convolutional neural networks for 3d shape analysis," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 72:1–72:11, 2017.

[8] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 77–85.

[9] 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 Trans. Graph.*, vol. 38, no. 5, pp. 146:1–146:12, 2019.

[10] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, 2015, pp. 945–953.

[11] W. Nie, K. Wang, H. Wang, and Y. Su, "The assessment of 3d model representation for retrieval with CNN-RNN networks," *Multimedia Tools Appl.*, vol. 78, no. 12, pp. 16 979–16 994, 2019.

[12] J. Xie, G. Dai, F. Zhu, E. K. Wong, and Y. Fang, "Deepshape: Deep-learned shape descriptor for 3d shape retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1335–1345, 2017. [Online]. Available: https://doi.org/10.1109/TPAMI.2016.2596722

[13] F. Zhu, J. Xie, and Y. Fang, "Learning cross-domain neural networks for sketch-based 3d shape retrieval," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, 2016, pp. 3683–3689.

[14] D. Maturana and S. A. 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, Hamburg, Germany, September 28 - October 2, 2015*, 2015, pp. 922–928.

[15] 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, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 5648–5656.

[16] Y. Lei, Z. Zhou, P. Zhang, Y. Guo, Z. Ma, and L. Liu, "Deep point-to-subspace metric learning for sketch-based 3d shape retrieval," *Pattern Recognition*, vol. 96, 2019.

[17] B. T. Phong, "Illumination for computer generated pictures," *Commun. ACM*, vol. 18, no. 6, pp. 311–317, 1975.

[18] Y. Feng, Z. Zhang, X. Zhao, R. Ji, and Y. Gao, "GVCNN: group-view convolutional neural networks for 3d shape recognition," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018, pp. 264–272.

[19] B. Li, Y. Lu, A. Godil, T. Schreck, M. Aono, H. Johan, J. M. Saavedra, and S. Tashiro, "Shrec'13 track: Large scale sketch-based 3d shape retrieval," in *Eurographics Workshop on 3D Object Retrieval, Girona, Spain, 2013. Proceedings*, 2013, pp. 89–96.

[20] B. Li, Y. Lu, A. Godil, T. Schreck, M. Aono, M. Burtscher, H. J. Hongbo Fu and, Takahiko Furuya and *et al.*, "Shrec'14 track: Extended large scale sketch-based 3d shape retrieval," pp. 121–130, 2014.

[21] B. Li, Y. Lu, F. Duan, S. Dong, Y. Fan, L. Qian, H. Laga, H. Li, Y. Li, P. Lui, M. Ovsjanikov, H. Tabia, Y. Ye, H. Yin, and Z. Xu, "Shrec'16 track: 3d sketch-based 3d shape retrieval," in *Eurographics Workshop on 3D Object Retrieval (3DOR) 2016*, 2016.

[22] P. Shilane, P. Min, M. M. Kazhdan, and T. A. Funkhouser, "The princeton shape benchmark," in *SMI 2004), 7-9 June 2004, Genova, Italy*, 2004, pp. 167–178.

[23] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 2015, pp. 1912–1920.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[25] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, 2017.

[26] G. Dai, J. Xie, and Y. Fang, "Deep correlated holistic metric learning for sketch-based 3d shape retrieval," *IEEE Trans. Image Processing*, vol. 27, no. 7, pp. 3374–3386, 2018.

[27] T. Darom and Y. Keller, "Scale-invariant features for 3-d mesh models," *IEEE Trans. Image Processing*, vol. 21, no. 5, pp. 2758–2769, 2012.

[28] G. Dai, J. Xie, F. Zhu, and Y. Fang, "Deep correlated metric learning for sketch-based 3d shape retrieval," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, 2017, pp. 4002–4008.

[29] J. Wang, J. Yang, K. Yu, F. Lv, T. S. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, 2010, pp. 3360–3367.

[30] A. Qi, Y. Song, and T. Xiang, "Semantic embedding for sketch-based 3d shape retrieval," in *British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*, 2018, p. 43.

[31] 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, BMVC 2016, York, UK, September 19-22, 2016*, 2016.

[32] R. Klokov and V. S. Lempitsky, "Escape from cells: Deep kd-networks for the recognition of 3d point cloud models," in *IEEE International*

Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, 2017, pp. 863–872.

[33] 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 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, 2017, pp. 5099–5108.

[34] J. M. Saavedra, B. Bustos, T. Schreck, S. M. Yoon, and M. Scherer, “Sketch-based 3d model retrieval using keyshapes for global and local representation,” in *Eurographics Workshop on 3D Object Retrieval 2012, Cagliari, Italy, May 13, 2012. Proceedings*, 2012, pp. 47–50.

[35] S. M. Yoon, M. Scherer, T. Schreck, and A. Kuijper, “Sketch-based 3d model retrieval using diffusion tensor fields of suggestive contours,” in *Proceedings of the 18th International Conference on Multimedia 2010, Firenze, Italy, October 25-29, 2010*, 2010, pp. 193–200.

[36] F. Wang, L. Kang, and Y. Li, “Sketch-based 3d shape retrieval using convolutional neural networks,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 2015, pp. 1875–1883.

[37] 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*, 2016, pp. 5023–5032.

[38] J. Xie, G. Dai, F. Zhu, and Y. Fang, “Learning barycentric representations of 3d shapes for sketch-based 3d shape retrieval,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 3615–3623.

[39] J. Chen and Y. Fang, “Deep cross-modality adaptation via semantics preserving adversarial learning for sketch-based 3d shape retrieval,” in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII*, 2018, pp. 624–640.

[40] Z. Li, C. Xu, and B. Leng, “Angular triplet-center loss for multi-view 3d shape retrieval,” in *AAAI*, 2019, pp. 8682–8689.

[41] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 2015, pp. 815–823.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, 2012, pp. 1106–1114.

[43] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 770–778.

[44] M. Eitz, J. Hays, and M. Alexa, “How do humans sketch objects?” *ACM Trans. Graph.*, vol. 31, no. 4, pp. 44:1–44:10, 2012.

[45] S. Benoit, D. Zachary, C. Soumith, G. Sam, P. Adam, M. Francisco, L. Adam, C. Gregory, L. Zeming, Y. Edward *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” 2019.

[46] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton, “On the importance of initialization and momentum in deep learning,” in *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, 2013, pp. 1139–1147.

[47] P. Goyal, P. Dollár, R. B. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, “Accurate, large minibatch SGD: training imagenet in 1 hour,” *CoRR*, vol. abs/1706.02677, 2017.

[48] T. Furuya and R. Ohbuchi, “Ranking on cross-domain manifold for sketch-based 3d model retrieval,” in *2013 International Conference on Cyberworlds, Yokohama, Japan, October 21-23, 2013*, 2013, pp. 274–281.

[49] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[50] L. Van Der Maaten and G. E. Hinton, “Visualizing data using t-sne,” in *Journal of Machine Learning Reseach 9, 2579-2625*, 2008.

[51] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, 2009, pp. 248–255. [Online]. Available: https://doi.org/10.1109/CVPRW.2009.5206848