View N-gram Network for 3D Object Retrieval

Xinwei He[1*]   Tengteng Huang[1*]   Song Bai[2]   Xiang Bai[1†]

[1]Huazhong University of Science and Technology
[2]University of Oxford

{eriche.hust, songbai.site}@gmail.com   {huangtengtng, xbai}@hust.edu.cn

Abstract

How to aggregate multi-view representations of a 3D object into an informative and discriminative one remains a key challenge for multi-view 3D object retrieval. Existing methods either use view-wise pooling strategies which neglect the spatial information across different views or employ recurrent neural networks which may face the efficiency problem. To address these issues, we propose an effective and efficient framework called View N-gram Network (VNN). Inspired by n-gram models in natural language processing, VNN divides the view sequence into a set of visual n-grams, which involve overlapping consecutive view sub-sequences. By doing so, spatial information across multiple views is captured, which helps to learn a discriminative global embedding for each 3D object. Experiments on 3D shape retrieval benchmarks, including ModelNet10, ModelNet40 and ShapeNetCore55 datasets, demonstrate the superiority of our proposed method.

1. Introduction

3D object retrieval is an important topic in computer vision and has received a surge of research attention owing to its close relationship with various geometry related applications, *e.g.*, VR/AR [20, 26, 48], medical imaging [25, 6, 56] and 3D printing [49]. With the development of 3D model acquisition technology, large amounts of 3D models are available for free, *e.g.*, the large-scale repository ShapeNet [8]. Thanks to the advances of data-driven based deep learning techniques, dramatic progresses have been achieved in this field. Nowadays, research trend has shifted from designing hand-crafted features [5, 10, 16, 24] to learn 3D shape representations directly via deep architectures [46, 37, 34, 23, 3].

In general, learning deep 3D shape representation can be coarsely divided into two mainstreams, *i.e.*, model-based and view-based methods. Model-based methods [37, 39, 53, 34, 54] learn 3D shape representations directly from the raw representation (*e.g.*, point cloud, voxel) of 3D shapes. View-based methods [46, 2, 51, 44, 55] usually first represent a 3D object with a set of 2D view images, then extract features of each view image, and finally aggregate them into a compact 3D shape descriptor. Comparing with model-based methods, view-based methods are more flexible and can benefit from recent developments in 2D image analysis, for instance, well-established architectures. Besides, in the real-world scenario, view images for 3D objects are easier to obtain, hence more efficient.

However, for multi-view based methods, one challenge is how to effectively aggregate the multiple view cues. To this end, existing works can be mainly categorized into two representative branches, *i.e.*, view-wise pooling strategies [46, 51] (see Fig. 1(a)) and recurrent neural networks (RNN) based strategies [12, 17] (see Fig. 1(b)). Although great progresses have been achieved, these methods have certain limitations and cannot fully leverage the latent view embeddings. For instance, view-wise pooling strategies lose the spatial information across different views, while recurrent neural networks suffer from inefficiency due to the sequential working mechanism which requires much more computational resources [50, 32].

Considering these limitations, we propose an effective and efficient framework called View N-gram Network (VNN) based on the concept of n-gram (see Fig. 1(c)) in natural language processing. In a typical n-gram model, a sentence is first decomposed into a sequence of n-grams, each of which has n consecutive words. Due to its capability of capturing word order in short context and its simplicity, n-gram has been widely used for various NLP tasks like language modeling [40], text classification [27], and machine translation [33]. Though its success has been demonstrated on learning text features, its effectiveness on learning 3D shape representations is unexplored.

However, n-gram also has intrinsic advantages for learning multi-view 3D shape representation. Specifically, for a 3D shape, we regard each view image as a "word" and di-

---

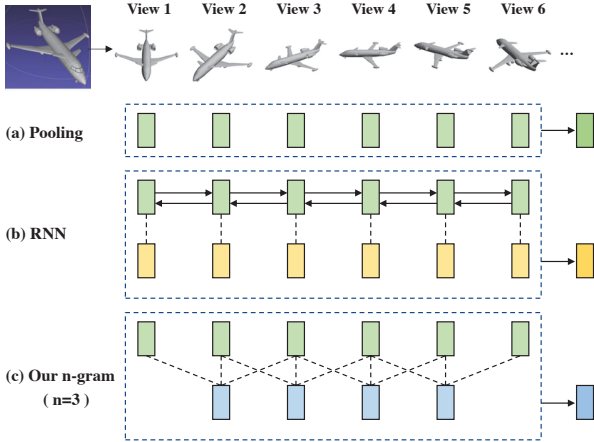*indicates equal contributions.
†corresponding author.

Figure 1. Illustration of different aggregation strategies for multiple view images, including (a) view-wise pooling, (b) recurrent neural networks, and (c) the proposed visual n-grams.

vide the sequence of all the views into a set of overlapping consecutive sub-sequences which we call visual n-grams. To this end, VNN utilizes an n-Gram Learning Unit (n-GLU) for efficient n-gram partitioning and intra-gram feature learning, thus the local spatial information among multiple views is well exploited. Compared with the RNN-based aggregation which works in a sequential manner, n-GLU is computationally more efficient as each visual n-gram only involves several consecutive views and can be fed in parallel to the network. Another merit of VNN is that the learned shape representation is rotation-invariant to a certain degree since the local adjacent relationship captured by visual n-grams is robust to rotation. Considering that visual n-grams of different sizes capture different scales of spatial information in the view sequence, we propose to combine the multi-scale representations learned with different n-gram sizes. Moreover, we further propose a parameterless attention model to selectively pack the partitioned n-gram features into a compact and global shape representation.

To summarize, our main contributions are as follows:

1. We present a novel framework named VNN to effectively model spatial information across the local context of the view sequence of each 3D shape. The proposed VNN firstly treats the rendered view sequence as a set of visual n-grams, and then computes rich n-gram features based on them, which produces more discriminative representations that are robust to rotation for 3D shape retrieval.

2. To capture different scales of the local spatial information across the view sequence, we propose to combine the learning of distinct visual n-gram sizes, which can lead to further improvements.

3. We design a parameterless attention model that efficiently and effectively aggregates learned visual n-gram

features, which proves to be a better aggregation method than max-pooling for view-based 3D object retrieval task.

4. Extensive experiments are conducted on both aligned and unaligned 3D shape benchmarks, and significant improvements are achieved over state-of-the-art methods.

The rest of the paper is organized as follows. We briefly review related works in Section 2. Then, we elaborate the proposed VNN in Section 3 and present the experimental evaluation in Section 4. Conclusions are given in Section 5.

2. Related work

3D object retrieval has received increasing attention and great efforts have been devoted to constructing discriminative 3D shape descriptors. Early works mainly focus on designing handcrafted features to represent 3D shapes. Various types of 3D shape descriptors have been proposed, *e.g.*, Light Field Descriptor (LFD) [10], Spherical Harmonic descriptor [24], and Heat Kernel Signatures [7]. With the recent development of deep learning techniques, learning 3D shape representations by deep neural networks has become a hot topic in the 3D object retrieval field. Generally speaking, existing methods can be coarsely divided into two categories, *i.e.* model-based methods and view-based methods.

Model-based methods [54, 13] deal with the raw representations of 3D shapes directly, *i.e.*, voxel, point cloud, and polygon mesh. Wu *et al.* [53] propose 3D ShapeNets which uses a Convolutional Deep Belief Network (CDBN) to learn representations directly on the voxelized 3D objects. Similarly, Maturana and Scherer propose VoxNet [34], which employs a 3D convolutional neural network and deals with 3D volumetric representations directly. Meanwhile, Qi *et al.* [38] propose Multi-Orientation Volumetric CNN (MO-VCNN) which aims at fusing the learned representations of 3D voxels from various orientations. However, methods based on the sparse volumetric representations are limited to the resolution of only 32^3 due to the cubically increasing computational complexity and memory overhead. To address this problem, Wang *et al.* [52] propose O-CNN, which is built upon a memory-efficient data structure named octree for the 3D object. Their method can process 3D shapes in the resolutions up to 256^3. As for the representation of point clouds, PointNet [37] is a pioneering work, which uses fully connected layers to embed the 3D coordinates into higher dimensional space and fuses them using max-pooling operation. Qi *et al.* [39] further propose PointNet++ to extract local features with increasing contextual scales followed by a hierarchical aggregation mechanism. Klokov *et al.* [29] propose Kd-Networks, which carry out computation based on the subdivision of point clouds using kd-trees. Overall, model-based methods are capable of exploiting the geometric information of 3D objects.

View-based methods usually first project the raw 3D shapes to a panoramic view [44, 43] or a set of 2D view images [46, 51, 24, 9, 55]. We mainly review the works which attempt to make rational use of multiple view images since they have a closer relation to our method. Bai *et al.* [2] propose a real-time 3D shape search engine based on the projective images. Su *et al.* [46] propose multi-view CNN, which uses a max-pooling operation to aggregate the multi-view representations outputted by a shared CNN. Dai *et al.* [12] propose a siamese CNN-BiLSTM for 3D shape representation learning, where they use BiLSTM to capture features across different views of a 3D shape. In addition, Han *et al.* [17] propose to use an RNN with attention to aggregate sequential views of each 3D object and promising results on several 3D shape retrieval benchmarks are obtained. Leng *et al.* [31] propose a score generation unit to evaluate the quality of the projected image and weight the view image features.

Our proposed View N-gram Network (VNN) borrows the idea of n-gram to aggregate multi-view representations. The idea of n-gram has been widely applied to language models [28, 22] and witnesses its success in text recognition [21, 36]. For a given sequence, it can be sliced into a set of overlapping sub-sequences consisting of n consecutive words or characters, which are of great importance for exploring the pattern of sequences. However, there is no attempt to adapt the spirit of n-gram for 3D shape related tasks in the literature. In this paper, we divide the multi-view images into a set of small overlapping subgroups (hence we name it view N-gram), and perform view feature enhancement based on each subgroup. Finally, an attention mechanism is adopted to aggregate the enhanced features. We will detail our method in the following section.

3. View n-gram network

Given a 3D object H, we first render it into a set of 2D greyscale images $V_H = \{v_1, v_2, ..., v_{|V|}\}$, where v_j denotes the j-th view image and $|V|$ represents the number of view images. Our goal is to learn a robust and discriminative representation for H under the multi-view setting.

As Fig. 2 shows, the pipeline of the proposed View N-gram Network can be divided into 3 stages, *i.e.*, feature extraction stage, n-gram feature learning and aggregation stage, and recognition stage. The first stage uses a shared convolutional neural network (CNN) that extracts features for each view, which is detailed in subsection 3.1. The second stage is the core part of our framework, which is a multi-branch network with each branch consisting of an n-Gram Learning Unit (n-GLU) (see subsection 3.2) and a parameterless attentional feature aggregator (see subsection 3.3) to learn and aggregate n-gram features for particular visual n-gram size. The recognition stage and other supplementary details are given in subsection 3.4.

3.1. View feature extraction

To extract the view features, a shared CNN is used. For each view image $v_i \in V_H$, the output of CNN is a D-dimensional feature $f_i \in \mathbb{R}^D$. Then, each 3D object can be represented as the multi-view embedding matrix $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, ..., \mathbf{f}_{|V|}]^T \in \mathbb{R}^{|V| \times D}$ by concatenating $\mathbf{f}_i$ $(1 \leq i \leq |V|)$ according to the rendered order.

Note that any off-the-shelf convolutional neural network (*e.g.*, AlexNet [30], GoogLeNet [47], ResNet [18]) can be used as the view feature extractor. In our work, we employ VGG-11 (also named VGG-A) with batch normalization [45] pre-trained on ImageNet [41] as our backbone. The original VGG-A has 11 layers, which consists of 8 convolutional layers (*conv 1-8*) and 3 fully connected layers (*fc 9-11*). In our experiments, we pre-train it and construct the feature extractor by removing the last two fully connected layers of the VGG-A. In this case, D is 4096.

3.2. N-gram learning unit

N-gram is a basic concept in natural language processing, and it has been widely used for language modeling. Let $S = (w_1, w_2, ..., w_m)$ be a sentence composed of m words. An n-gram is defined as a substring consisting of n consecutive words $(w_i, w_{i+1}, ..., w_{i+n-1})$ from S. N-gram can adaptively model the temporal dependency of the n consecutive words [32].

As suggested in [17], the spatial relationships among the view images play an important role in multi-view 3D shape analysis. Therefore, we propose to model the spatial dependency of view images in the form of n-gram inspired by its success in modeling temporal dependency. We regard each view image of a 3D object as one "word", then a sequence of $|V|$ rendered view images can be analogized to a sentence of $|V|$ words. Similarly, we can further decompose the view sequence into a set of $|V| + 1 - n$ n-grams, each of which is composed of n consecutive view images (we call it visual n-gram). Each visual n-gram depicts a certain pattern of its corresponding 3D shape. Intuitively, 3D shapes from the same category should share similar n-gram patterns while those from different categories should differ in their n-gram patterns. Therefore, it would be beneficial for understanding a 3D shape to learn the n-gram patterns by capturing the local spatial dependency among consecutive view images. To this end, we propose a novel module named n-Gram Learning Unit (n-GLU). And the mechanism works as follows.

Recall that the multi-view representation for a 3D shape is denoted as an embedding matrix $\mathbf{F} \in \mathbb{R}^{|V| \times D}$ arranged in the rendering order. Similar to an n-gram based sentence classification network [27], we adopt a sliding window strategy of size $n \times D$ over $\mathbf{F}$ for the partition of n-grams as illustrated in Fig. 3. In particular, for each visual n-gram which corresponds to local consecutive n images of
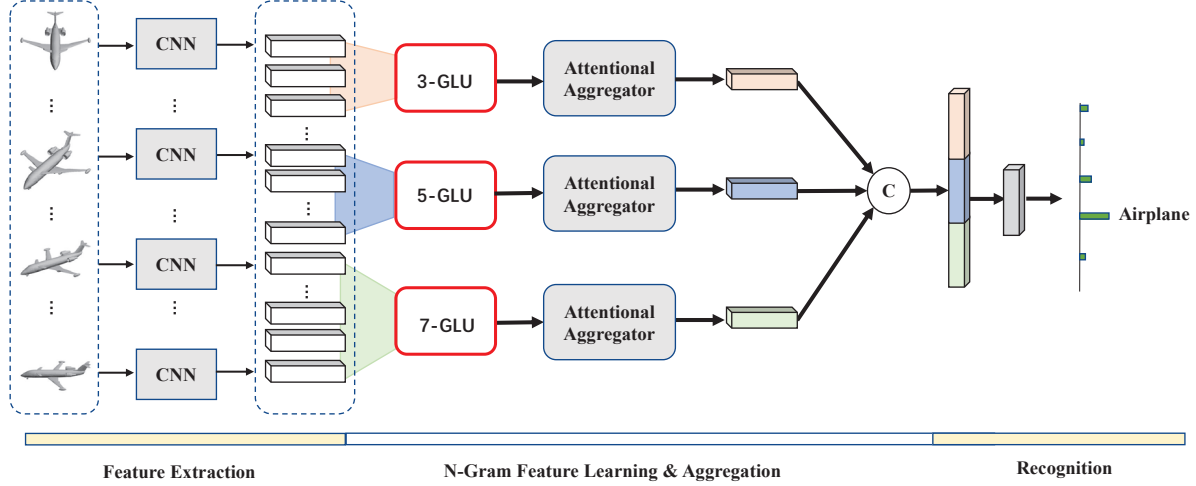
Figure 2. The pipeline of View N-gram Network. A shared CNN is used to extract features for each view images of the 3D object. The raw feature sequence is enhanced by modeling the dependency of consecutive view features in a convolutional manner by our GLU module and then aggregated into a global descriptor. Three parallel branches are employed to exploit spatial information in different scales, resulting in a more discriminative representation for the 3D shape.

the sequence, we compute the enhanced visual n-gram features by using 2D convolution filter of size $D' \times D \times n \times 1$, where D' is the dimension of the output features enhanced by the visual n-gram. It is straightforward that the enhanced D'-dimensional representation has encoded the local spatial information of the corresponding visual n-grams. Since there are $|V| - n + 1$ visual n-grams for each 3D object, therefore, the final enhanced compact representation $\mathbf{G}$ is of size $(|V| - n + 1) \times D'$ for each 3D shape. In our experiments, D' is set to 512.

N-gram learning unit possesses two desirable advantages for robust 3D representations. First, n-GLU can adaptively learn typical patterns for 3D shapes by exploring the spatial relation of several local consecutive views. Different from RNN which aims at modeling the long-range dependency among all the views, the proposed method can better capture local and fine-grained patterns. Second, our framework is endowed with the rotational invariance to some extent since the local adjacent relation of views is quite robust to rotation transformation. This holds for both aligned and unaligned 3D shapes. However, RNN based methods may be sensitive to the predefined viewpoints and thus cannot ensure the rotational invariance of the learned features, as suggested in [11].

3.3. Attentional feature aggregation

N-gram learning unit outputs a new sequence of visual n-gram features with local spatial information incorporated. And for the downstream retrieval task, we need to aggregate them into a compact and discriminative representation.

Max-pooling is a simple yet effective strategy. However, it may lead to sub-optimal performance as much information is lost by only keeping the maximum value. For better
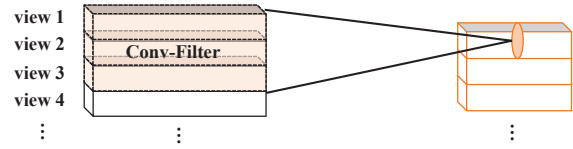

Figure 3. A diagram of the GLU model used in our method. Here we choose the view n-gram size to be 3 for illustration purpose, $i.e.$, 3-GLU. In essence, GLU is a 2D convolutional filter, which convolves with the multi-view feature embedding matrix and produces visual n-gram features.

utilization of different n-gram view features and the reduction of information loss, we further propose to use attentional feature aggregation mechanism, which can be seen as a variant of self-attention module [50]. The main difference is that we estimate the correlation between the global feature and each n-gram feature instead of all pairs of n-gram features. Besides, our attention module is free of parameters. The pipeline of our attention feature aggregation is presented in Fig. 4, which can be formulated as

$$\beta_j = \frac{\exp(\phi_j(\mathbf{G_j}, \mathbf{g}_p))}{\sum_{i=1}^{|V|+1-n} \exp(\phi_i(\mathbf{G_i}, \mathbf{g}_p))} \qquad (1)$$

$$\phi_j = \frac{\mathbf{G_j} * \mathbf{g}_p}{\sqrt{D'}} \qquad (2)$$

$$\mathbf{g}_a = \sum_{j=1}^{|V|+1-n} \beta_j \mathbf{G}_j \qquad (3)$$

where $\mathbf{g}_p$ and $\mathbf{g}_a$ denote the global representation for the sequence (via max pooling operation over $\mathbf{G}$) and the attentional output, respectively. $\mathbf{G_j}$ represents the j-th row of matrix $\mathbf{G}$ ($i.e.$, j-th visual n-gram feature) and β_j is the
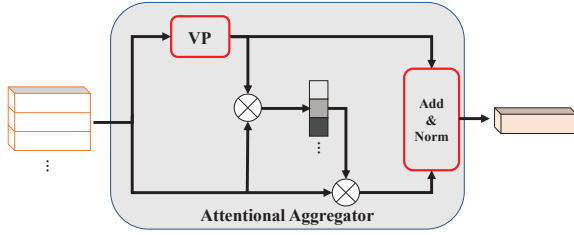
Figure 4. The detailed structure of our proposed attention aggregator. Here VP represents view-pooling operation (*i.e.* max-pooling), which outputs global n-gram feature. The attention aggregator assigns weights to different n-gram features according to their similarities to the proxy global n-gram feature. A compact descriptor is obtained by combining all the n-gram features in a weighted sum manner.

corresponding attention score. ϕ denotes the inner product with scale normalization to avoid extremely small gradients when the result is in large magnitude as suggested in [50]. We elaborate the details of our attention aggregation mechanism below, which includes three main steps.

Estimation of attention score. We first adopt max-pooling operation over the set of the visual n-gram features $\mathbf{G}$ to obtain a global descriptor $\mathbf{g}_p$. We then assign attention scores to different n-gram features by estimating their correlations to the global descriptor $\mathbf{g}_p$ in an inner product manner, as formulated in Eq. (1) and Eq. (2).

View aggregation. With the estimated attention scores, the n-gram view features are aggregated into a compact representation $\mathbf{g}_a$ in a weighted sum fashion (see Eq. (3)). In this way, different n-gram views, which contain different local spatial information, are effectively combined.

Residual connection. Similar to self-attention [50], we further introduce a residual connection to add the global descriptor $\mathbf{g}_p$ back to the aggregated feature $\mathbf{g}_a$ to reduce information loss, which is then normalized by a layer normalization operation [1].

Note that the full pipeline of our attentional feature aggregation does not involve any learnable parameters. Therefore, it is very efficient.

3.4. Remarks

Multi-scale n-gram feature fusion. Varying the n-gram size, n-GLU module can capture the spatial information in different scales (or the number of consecutive view images) and learn discriminative patterns of 3D shapes in distinct granularities. Therefore, we propose to employ multiple parallel branches for comprehensively characterizing the 3D shape from different scales. Each branch performs n-gram feature learning and attentional aggregation separately with different n-gram sizes. We then combine the aggregated features from all the branches in a concatenation

manner. In this way, spatial information among the view images is better exploited. As presented in Fig. 2, we use three n-gram branches with n-gram sizes of 3, 5 and 7 in our VNN framework.

Training details. For the final recognition stage, we employ a simple network which is composed of only two fully connected layers. The first layer takes the combined feature as input and maps it into a lower-dimensional vector $\mathbf{g}_r \in \mathbb{R}^{512}$. The last layer predicts the category distribution based on $\mathbf{g}_r$. In our experiments, we adopt *softmax* loss as the training objective.

4. Experiments

4.1. Datasets

In our experiments, we evaluate the proposed method on three public 3D shape retrieval datasets, *i.e.*, ModelNet40 [53], ModelNet10 [53] and ShapeNetCore55 [8]. ModelNet40 and ModelNet10 datasets are two subsets of the Princeton ModelNet dataset which contains 151,128 3D shapes divided into 660 categories. The ShapeNetCore55 dataset is a large-scale dataset from ShapeNet.

The ModelNet40 contains 12,311 shapes from 40 common categories. In our experiments, we use the same training/test splits as popular MVCNN [46] and 3D ShapeNets [53]. The ModelNet10 dataset consists of 4,899 3D models in total from 10 classes. The training and testing sets contain 3,991 and 908 models, respectively.

The ShapeNetCore55 dataset is introduced in the SHape REtrieval Contest (SHREC) 2016 competition track. It is a large-scale dataset, which is composed of 51,190 3D shapes from 55 shape categories. Each model in the dataset is also attached with a fine-grained subcategory from 204 subcategories in addition to the label from the 55 categories. Among these 3D shape data, 35,765 3D shapes (70%) are provided for training and another 5,159 3D shapes (10%) are for validation. The remaining 10,266 shapes (20%) form the testing set. The dataset has two versions, *i.e.*, "normal" and "perturbed" versions. For the "normal" version, the 3D shapes are aligned. For the "perturbed" version, each 3D shape is arbitrarily oriented. Hence, the latter version is more challenging. To test the robustness of our method, we conduct experiments on both versions.

4.2. Evaluation metrics

In our experiments, we adopt five common metrics to evaluate the retrieval performance of our method against state-of-the-art methods, which are defined as follows:

- Precision-Recall (PR) curve is used to visualize the retrieval performance.
- Mean Average Precision (mAP) of the PR curve is used to give a quantitative evaluation.

- Area Under Curve (AUC) is the mean area under the PR curve.
- F-Measure is the harmonic mean of recall and precision.
- Normalized Discounted Cumulative Gain (NDCG) is a measure that assigns more weights to the relevant results at the top of the ranked list.

Among these metrics, we use the PR curve, mAP, AUC on ModelNet40 and mAP, AUC on ModelNet10 to evaluate the retrieval performance. For ShapeNetCore55 dataset, F-Measure, mAP, and NDCG are adopted.

4.3. Implementation details

We render a set of 2D greyscale images of size 224×224 for each 3D object, following the same rendering protocol as MVCNN [46]. For the aligned datasets, *i.e.*, ModelNet40, ModelNet10 and ShapeNetCore55 "normal" datasets, we place virtual cameras around the 3D object every 30 degrees, obtaining 12 view images. For the unaligned ShapeNetCore55 "perturbed" dataset, 80 view images are rendered by placing virtual cameras at 20 vertices of the icosahedron and taking 4 views per camera using 0, 90, 180, and 270 degrees in-plane rotations.

During training, we adopt stochastic gradient descent (SGD) for optimization with momentum of 0.9 and weight decay of 0.0001. The learning rate is set to 0.001. We clip the gradients into the range [-0.01, 0.01] for training stability. And we train the model for 150 epochs with a mini-batch size of 8. In the inference time, we extract the output of the penultimate layer of our network, which is 512-dimensional, as the descriptor for each 3D object.

We implement our method using PyTorch [35], and all the experiments are conducted on a server with eight NVIDIA Titan-X GPUs, an Intel i7 CPU and 64GB RAM.

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

To validate the effectiveness of our method, we first conduct experiments on two common 3D shape datasets-ModelNet40 and ModelNet10, in which 3D objects are assumed to be aligned. We then further conduct experiments on the more challenging ShapeNetCore55 dataset, which includes two versions, *i.e.*, the "normal" version with aligned 3D objects and the "perturbed" version where the orientations of 3D shapes remain unknown.

Comparison on ModelNet40. The comparison with state-of-the-art methods is listed in Table 1. We present the results of three representative model-based methods including SPH [24], 3DShapeNet [53] and DLAN [15], and the results of several representative view-based methods including LFD [10], DeepPano [44], GIFT [2], MVCNN [46], GVCNN [14], RED [4], TCL [19] and SeqViews [17] for extensive comparison. Moreover, we also re-implement

Methods	ModelNet40		ModelNet10	
	AUC	mAP	AUC	mAP
SPH [24]	34.5	33.3	46.0	44.1
3DShapeNet [53]	49.9	49.2	69.3	68.3
DLAN [15]	-	85.0	-	90.6
LFD [10]	42.0	40.9	51.7	49.8
DeepPano [44]	77.6	76.8	85.5	84.2
GIFT [2][V-S]	83.1	81.9	92.4	91.1
MVCNN [46][V-M]	-	70.1	-	-
MVCNN* [46][V-M]	-	80.2	-	-
RED [4][R-50]	87.0	86.3	93.2	92.2
GVCNN [14][G]	-	85.7	-	-
TCL [19][V-A]	89.0	88.0	-	-
SeqViews [17][V-19]	-	89.1	-	91.4
MVCNN[†][V-A]	73.7	72.9	80.8	80.1
Ours[V-A]	89.6	88.9	**93.5**	**92.8**
Ours[V-19]	**90.2**	**89.3**	-	-

Table 1. The comparison with state-of-the-art methods on ModelNet40 and ModelNet10. On the top are results of model-based methods. Results of view-based methods are listed in the middle. * means employing metric learning. † represents reproduced MVCNN result which is conducted under the same setting (backbone network and pooling position) of the proposed method as the baseline. V-S, V-M, V-A, V-19, G and R-50 represent using VGG-S, VGG-M, VGG-A, VGG-19, GoogLeNet and ResNet-50 architectures, respectively.

MVCNN by inserting max-pooling operation at the fc-9 layer of VGG-A with batch normalization, as our baseline. It can be observed that our method achieves very competitive performance, reaching 89.6% in AUC and 88.9% in mAP, outperforming most existing methods. Concretely, our method surpasses the best model-based method DLAN by 3.9% in terms of mAP. And when compared with other view-based methods, VNN outperforms GIFT, GVCNN, and TCL by 7.0%, 3.2%, and 0.9% in mAP, respectively. It should be noted that GVCNN employs a stronger backbone GoogLeNet. Compared with RED, which is a sophisticated similarity fusion method, we outperform it by 2.6% in mAP. Besides, our method achieves comparable performance with SeqViews for mAP (88.9% *vs.* 89.1%) while SeqViews leverages a stronger backbone VGG-19 for view feature extraction. To make a fair comparison, we further use VGG-19 for the experiment and achieve slightly better performance (89.3% *vs.* 89.1% in mAP). When compared with our baseline, VNN obtains significant improvements, *e.g.*, 89.6% *vs.* 73.7% for AUC and 88.9% *vs.* 72.9% for mAP. The consistent gains over the baseline and state-of-the-art methods demonstrate the superiority of our method.

We present several retrieval examples on the Model-Net40 dataset in Fig. 5. It can be seen that our method can retrieve highly relevant 3D objects for the queries. Note that the retrieved false positives also share similar shapes with the query, *e.g.*, the vase in the last row.

ShapeNetCore55	Methods	microALL			macroALL			microALL + macroALL		
		F1	mAP	NDCG	F1	mAP	NDCG	F1	mAP	NDCG
normal	Wang [42]	39.1	82.3	88.6	28.6	66.1	82.0	33.8	74.2	85.3
	Li [42]	58.2	82.9	90.4	20.1	71.1	84.6	39.2	77.0	87.5
	MVCNN [46][V-M]	76.4	87.3	89.9	57.5	81.7	88.0	66.9	84.5	89.0
	GIFT [2][V-S]	68.9	82.5	89.6	45.4	74.0	85.0	57.2	78.3	87.3
	Kd-network [29]	74.3	85.0	90.5	51.9	74.6	86.4	63.1	79.8	88.5
	Ours[V-A]	**78.9**	**90.3**	**92.8**	**61.4**	**85.2**	**91.7**	**70.2**	**87.8**	**92.3**
perturbed	Wang [42]	24.6	60.0	77.6	16.3	47.8	69.5	20.5	53.9	73.6
	Li [42]	53.4	74.9	86.5	18.2	57.9	76.7	35.8	66.4	81.6
	MVCNN [46][V-M]	61.2	73.4	84.3	41.6	66.2	79.3	51.4	69.8	81.8
	GIFT [2][V-S]	66.1	81.1	88.9	42.3	73.0	84.3	54.2	77.0	86.6
	Kd-network [29]	45.1	61.7	81.4	24.1	48.4	72.6	34.6	55.1	77.0
	TCL [19][V-A]	67.9	84.0	89.5	43.9	**78.3**	**86.9**	55.9	**81.2**	88.2
	Ours[V-A]	**71.3**	**84.3**	**89.7**	**50.1**	78.0	86.8	**60.7**	**81.2**	**88.3**

Table 2. The performance (%) comparison on the ShapeNetCore55 dataset. On the top is the performance comparison on the "normal" version dataset, while comparison results on "perturbed" version are listed in the bottom rows.

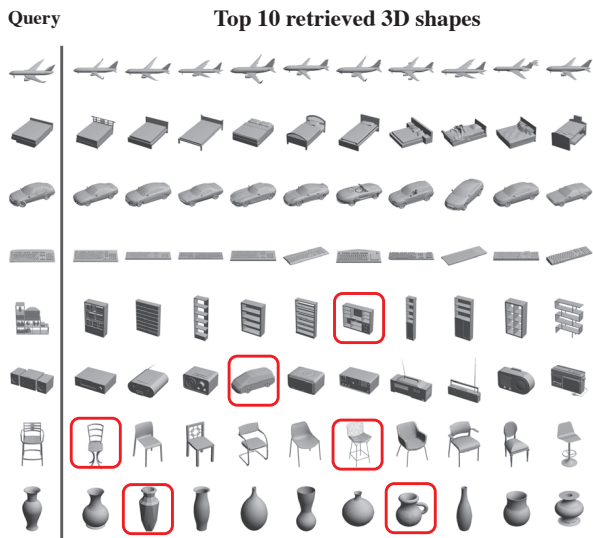**Query**   **Top 10 retrieved 3D shapes**

Figure 5. Illustration of the retrieval examples on the ModelNet40 dataset. The query shapes are put on the first left column, and the top 10 retrieved shapes are on the right side. The retrieved false positives are highlighted by red boxes.

Comparison on ModelNet10. We present the results on the Modelnet10 dataset in Table 1. As shown, our method yields an AUC of 93.5% and an mAP of 92.8%, considerably outperforming state-of-the-art methods. Comparing with DLAN [15], which is a superior model-based method utilizing rich rotation-invariant 3D local features, we achieve an improvement of 2.2% in terms of mAP. In addition, VNN outperforms DeepPano [44], GIFT [2] and RED [4] by 8.6%, 1.7%, and 0.6% in terms of mAP, respectively. The comparison with SeqViews [17] is especially valuable since SeqView leverages a stronger backbone network for view feature extraction. Nevertheless, the proposed method surpasses it by 1.4% in terms of mAP. We also provide the reproduced result of MVCNN under

the same settings with VNN as a baseline. We can observe that VNN significantly boosts the mAP of MCNNN by 12.7%, suggesting the importance of local spatial relation for 3D object retrieval and the effectiveness of the proposed method.

Comparison on ShapeNetCore55 dataset. We conduct experiments on both the "normal" and the "perturbed" versions of the ShapeNetCore55 dataset. For the convenience of comparison with other methods, we provide the average value for the two kinds of evaluation metrics (*i.e.*, microALL and macroALL). The comparison for both versions is listed in Table 2. As shown, significant improvements over state-of-the-art methods are achieved on both versions of the dataset. In particular, on the "normal" version, when compared with Kd-network [29], which is model-based, we improve the mAP by 9%. And compared with MVCNN [46], improvements of over 3% in terms of all the evaluation metrics are obtained. At the same time, the comparison results on the more challenging "perturbed" version also prove the effectiveness of our method. When compared with TCL [19], which adopts an end-to-end metric learning loss function, we achieve comparable performance in terms of both mAP and NDCG. However, for F1, nearly 5% improvements are obtained. It demonstrates that the proposed method can be naturally extended to the cases where orientations of 3D shapes are agnostic.

4.5. Ablation study

In this section, we present ablation experiments of the proposed method on the ModelNet40 dataset. Concretely, we will study the effects of n-gram size, combinations of different n-gram sizes, attentional aggregation mechanism and its complementarity with metric learning methods.

Effect of n-gram size. we first explore the effect of n-gram size on the model performance. The n-gram size is a

N-gram size	Max-Pooling		With Attention	
	AUC	mAP	AUC	mAP
1	80.3	79.5	81.5	80.7
3	84.0	83.2	85.6	84.9
5	85.5	84.7	88.0	87.3
7	85.1	84.3	86.6	85.9
3+5	88.3	87.6	88.7	88.0
3+7	87.8	87.0	88.3	87.5
5+7	87.9	87.1	88.9	88.1
3+5+7	**88.9**	**88.2**	**89.6**	**88.9**

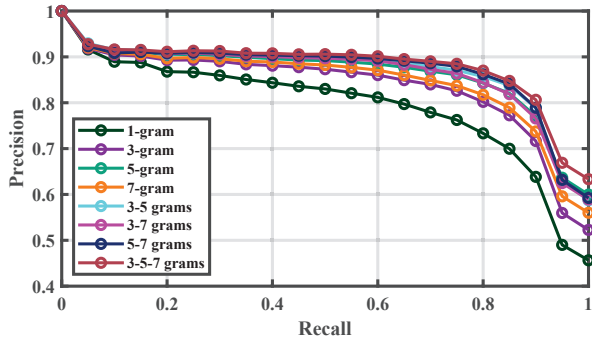Table 3. Ablation analysis on the ModelNet40 dataset.

Figure 6. PR curves on the ModelNet40 dataset for different settings of n-gram size.

very important hyper-parameter because it affects the context window for computing the n-gram features. We set the n-gram size to be 1, 3, 5, 7 and study their impacts on ModelNet40. The comparison results are reported in Table 3. As shown, when the n-gram size is 1, which means adopting uni-gram without considering the context of each view images, we can only reach an mAP of 80.7%. However, when we increase the n-gram size to 3, we see an improvement of nearly 4%. It suggests that more discriminative representations can be obtained by incorporating local spatial information via the visual n-grams (n>1). The best result is obtained when we set the n-gram size to be 5, which reaches 88.0% and 87.3% in terms of AUC and mAP, respectively. When the n-gram size is greater than 5, we see some degree of degradation on retrieval performance.

Effect of fusion on different n-grams. We further discuss how the combinations of different n-gram sizes influence the retrieval performance. Concretely, we enumerate all the combinations of three n-gram sizes (*i.e.*, 3-gram, 5-gram and 7-gram). As shown in Table 3, combining different kinds of gram sizes yields consistent improvements over only utilizing a single gram size, suggesting different n-gram modules are complementary to each other and combining them can effectively improve the retrieval performance. Besides, it should be noted that different combinations in Table 3 share the same retrieval efficiency since they output representations of the same dimension (*i.e.* 512).

The PR curves are shown in Fig. 6, which intuitively

Methods	AUC	mAP
Ours	89.6	88.9
Ours + TCL [19]	**90.5**	**89.5**

Table 4. The performance (%) of our method with and without TCL on ModelNet40.

demonstrate the retrieval performance based on each n-gram size and their different combinations.

Effect of the attentional aggregation mechanism. We compare the attentional aggregation mechanism with the widely adopted max-pooling operation under different settings for n-gram size. As presented in Table 3, the attentional aggregation mechanism demonstrates consistent improvements over max-pooling. It should be noted that our attentional aggregation mechanism is parameter-free, suggesting it can serve as a more powerful alternative to max-pooling for better utilization of multi-view features.

Complementary to metric learning. Triplet-Center Loss (TCL) [18] is an effective metric learning loss and has achieved superior performance on multiple 3D shape benchmarks. Hence, we further combine the softmax loss with the triplet-center loss to validate the complementarity of our method with metric learning method. As shown in Table 4, TCL yields an improvement of AUC 0.9% and mAP of 0.6%, demonstrating the potential of our method when combined with existing metric learning methods.

5. Conclusion

In this paper, we propose a novel framework named View N-gram Network (VNN) to model the spatial relationships of multi-view images for 3D objects, which can learn discriminative representations for 3D object retrieval task. The core component of VNN is the n-Gram Learning Unit (n-GLU), which first divides multiple view images into a set of visual n-grams efficiently and then learns the intra-gram feature effectively. In this way, local spatial information is leveraged. Attentional aggregation mechanism is adopted over the learned n-gram features. Moreover, we propose to fuse representations under different n-gram sizes. Experimental results on multiple 3D shape benchmarks demonstrate the superiority of the learned 3D shape representations from the proposed method. In the future, we would like to explore other alternatives to the concatenation method for better fusing the multi-scale n-gram features.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 5

[2] Song Bai, Xiang Bai, Zhichao Zhou, Zhaoxiang Zhang, and Longin Jan Latecki. Gift: A real-time and scalable 3d shape search engine. In *CVPR*, pages 5023–5032, 2016. 1, 3, 6, 7

[3] Song Bai, Peng Tang, Philip HS Torr, and Longin Jan Latecki. Re-ranking via metric fusion for object retrieval and person re-identification. In *Proc. CVPR*, pages 740–749, 2019. 1

[4] Song Bai, Zhichao Zhou, Jingdong Wang, Xiang Bai, Longin Jan Latecki, and Qi Tian. Ensemble diffusion for retrieval. In *Proc. ICCV*, pages 774–783, 2017. 6, 7

[5] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *Proc. NIPS*, pages 831–837, 2001. 1

[6] Leila Bergamasco, Karla Lima, Carlos Rochitte, and Fátima de Lourdes dos Santos Nunes. 3d medical objects retrieval approach using spharms descriptor and network flow as similarity measure. In *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 329–336. IEEE, 2018. 1

[7] Alexander M. Bronstein, Michael M. Bronstein, Leonidas J. Guibas, and Maks Ovsjanikov. Shape google: Geometric words and expressions for invariant shape retrieval. *Acm Transactions on Graphics*, 30(1):1–20, 2011. 2

[8] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1, 5

[9] Dapeng Chen, Dan Xu, Hongsheng Li, Nicu Sebe, and Xiaogang Wang. Group consistent similarity learning via deep crf for person re-identification. In *Proc. CVPR*, pages 8649–8658, 2018. 3

[10] Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. On visual similarity based 3d model retrieval. In *Comput. Graph. Forum*, pages 223–232, 2003. 1, 2, 6

[11] Songle Chen, Lintao Zheng, Yan Zhang, Zhixin Sun, and Kai Xu. Veram: View-enhanced recurrent attention model for 3d shape classification. *IEEE Trans. Vis. Comput. Graphics*, 2018. 4

[12] Guoxian Dai, Jin Xie, and Yi Fang. Siamese cnn-bilstm architecture for 3d shape representation learning. In *Proc. IJCAI*, 2018. 1, 3

[13] Yi Fang, Jin Xie, Guoxian Dai, Meng Wang, Fan Zhu, Tiantian Xu, and Edward Wong. 3d deep shape descriptor. In *Proc. CVPR*, pages 2319–2328, 2015. 2

[14] Yifan Feng, Zizhao Zhang, Xibin Zhao, Rongrong Ji, and Yue Gao. Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In *Proc. CVPR*, pages 264–272, 2018. 6

[15] Takahiko Furuya and Ryutarou Ohbuchi. Deep aggregation of local 3d geometric features for 3d model retrieval. In *Proc. BMVC*, 2016. 6, 7

[16] Yue Gao, You Yang, Qionghai Dai, and Naiyao Zhang. 3d object retrieval with bag-of-region-words. In *Proc. ACM Int. Conf. Multimedia*, pages 955–958, 2010. 1

[17] Zhizhong Han, Mingyang Shang, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Matthias Zwicker, Junwei Han, and CL Philip Chen. Seqviews2seqlabels: Learning 3d global features via aggregating sequential views by rnn with attention. *IEEE Trans. Image Process.*, 28(2):658–672, 2018. 1, 3, 6, 7

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, pages 770–778, 2016. 3, 8

[19] Xinwei He, Yang Zhou, Zhichao Zhou, Song Bai, and Xiang Bai. Triplet-center loss for multi-view 3d object retrieval. *Proc. CVPR*, 2018. 6, 7, 8

[20] Hiroyasu Ichida, Yuichi Itoh, Yoshifumi Kitamura, and Fumio Kishino. Interactive retrieval of 3d virtual shapes using physical objects. In *IEEE Virtual Reality 2004*, pages 231–232. IEEE, 2004. 1

[21] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep structured output learning for unconstrained text recognition. *arXiv preprint arXiv:1412.5903*, 2014. 3

[22] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016. 3

[23] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *Proc. CVPR*, pages 5010–5019, 2018. 1

[24] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Rotation invariant spherical harmonic representation of 3 d shape descriptors. In *Symposium on geometry processing*, volume 6, pages 156–164, 2003. 1, 2, 3, 6

[25] Daniel Keysers, Jörg Dahmen, Hermann Ney, Berthold B Wein, and Thomas Martin Lehmann. Statistical framework for model-based image retrieval in medical applications. *Journal of Electronic Imaging*, 12(1):59–69, 2003. 1

[26] Kyungyoon Kim, Rebekah L Lawrence, Nikki Kyllonen, Paula M Ludewig, Arin M Ellingson, and Daniel F Keefe. Anatomical 2d/3d shape-matching in virtual reality: A user interface for quantifying joint kinematics with radiographic imaging. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 243–244. IEEE, 2017. 1

[27] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014. 1, 3

[28] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. Character-aware neural language models. In *Proc. AAAI*, 2016. 3

[29] Roman Klokov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *Proc. ICCV*, pages 863–872, 2017. 2, 7

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. NIPS*, pages 1097–1105, 2012. 3

[31] Biao Leng, Cheng Zhang, Xiaochen Zhou, Cheng Xu, and Kai Xu. Learning discriminative 3d shape representations by view discerning networks. *IEEE transactions on visualization and computer graphics*, 2018. 3

[32] Bofang Li, Zhe Zhao, Tao Liu, Puwei Wang, and Xiaoyong Du. Weighted neural bag-of-n-grams model: New baselines for text classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1591–1600, 2016. 1, 3

[33] José B Marino, Rafael E Banchs, Josep M Crego, Adrià de Gispert, Patrik Lambert, José AR Fonollosa, and Marta R Costa-Jussà. N-gram-based machine translation. *Computational linguistics*, 32(4):527–549, 2006. 1

[34] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Proc. Int. Conf. Intell. Robots Syst.*, pages 922–928, 2015. 1, 2

[35] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 6

[36] Arik Poznanski and Lior Wolf. Cnn-n-gram for handwriting word recognition. In *Proc. CVPR*, pages 2305–2314, 2016. 3

[37] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proc. CVPR*, pages 652–660, 2017. 1, 2

[38] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proc. CVPR*, pages 5648–5656, 2016. 2

[39] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proc. NIPS*, pages 5099–5108, 2017. 1, 2

[40] Brian Roark, Murat Saraclar, and Michael Collins. Discriminative n-gram language modeling. *Computer Speech & Language*, 21(2):373–392, 2007. 1

[41] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 3

[42] Manolis Savva, Fisher Yu, Hao Su, M Aono, B Chen, D Cohen-Or, W Deng, Hang Su, Song Bai, Xiang Bai, et al. Shrec16 track: largescale 3d shape retrieval from shapenet core55. In *Proc. EG Workshop 3D Object Recognit.*, 2016. 7

[43] Konstantinos Sfikas, Ioannis Pratikakis, and Theoharis Theoharis. Ensemble of panorama-based convolutional neural networks for 3d model classification and retrieval. *Computers & Graphics*, 71:208–218, 2018. 3

[44] Baoguang Shi, Song Bai, Zhichao Zhou, and Xiang Bai. Deeppano: Deep panoramic representation for 3-d shape recognition. *IEEE Signal Process. Lett.*, 22(12):2339–2343, 2015. 1, 3, 6, 7

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[46] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. CVPR*, pages 945–953, 2015. 1, 3, 5, 6, 7

[47] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, pages 1–9, 2015. 3

[48] Gabriel Takacs, Vijay Chandrasekhar, Natasha Gelfand, Yingen Xiong, Wei-Chao Chen, Thanos Bismpigiannis, Radek Grzeszczuk, Kari Pulli, and Bernd Girod. Outdoors augmented reality on mobile phone using loxel-based visual feature organization. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, pages 427–434. ACM, 2008. 1

[49] Francesca Uccheddu, Monica Carfagni, Lapo Governi, Rocco Furferi, Yary Volpe, and Erica Nocerino. 3d printing of cardiac structures from medical images: an overview of methods and interactive tools. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 12(2):597–609, 2018. 1

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. NIPS*, pages 5998–6008, 2017. 1, 4, 5

[51] Chu Wang, Marcello Pelillo, and Kaleem Siddiqi. Dominant set clustering and pooling for multi-view 3d object recognition. In *Proc. BMVC*, volume 12, 2017. 1, 3

[52] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Trans. Graph.*, 36(4):72, 2017. 2

[53] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proc. CVPR*, pages 1912–1920, 2015. 1, 2, 5, 6

[54] Jin Xie, Guoxian Dai, Fan Zhu, Edward K Wong, and Yi Fang. Deepshape: Deep-learned shape descriptor for 3d shape retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(7):1335–1345, 2017. 1, 2

[55] Tan Yu, Jingjing Meng, and Junsong Yuan. Multi-view harmonized bilinear network for 3d object recognition. In *Proc. CVPR*, pages 186–194, 2018. 1, 3

[56] Yuyin Zhou, Yan Wang, Peng Tang, Song Bai, Wei Shen, Elliot Fishman, and Alan Yuille. Semi-supervised 3d abdominal multi-organ segmentation via deep multi-planar co-training. In *Proc. WACV*, pages 121–140, 2019. 1