The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Learning Adversarial 3D Model Generation with 2D Image Enhancer

Jing Zhu, Jin Xie, Yi Fang *
NYU Multimedia and Visual Computing Lab
Department of Electrical and Computer Engineering, NYU Abu Dhabi, UAE
Department of Electrical and Computer Engineering, NYU Tandon School of Engineering, USA
Department of Computer Science and Engineering, NYU Tandon School of Engineering, USA

Abstract

Recent advancements in generative adversarial nets (GANs) and volumetric convolutional neural networks (CNNs) enable generating 3D models from a probabilistic space. In this paper, we have developed a novel GAN-based deep neural network to obtain a better latent space for the generation of 3D models. In the proposed method, an enhancer neural network is introduced to extract information from other corresponding domains (e.g. image) to improve the performance of the 3D model generator, and the discriminative power of the unsupervised shape features learned from the 3D model discriminator. Specifically, we train the 3D generative adversarial networks on 3D volumetric models, and at the same time, the enhancer network learns image features from rendered images. Different from the traditional GAN architecture that uses uninformative random vectors as inputs, we feed the high-level image features learned from the enhancer into the 3D model generator for better training. The evaluations on two large-scale 3D model datasets, ShapeNet and ModelNet, demonstrate that our proposed method can not only generate high-quality 3D models, but also successfully learn discriminative shape representation for classification and retrieval without supervision.

In the recent decades, 3D model generation has attracted increasing interests in computer vision community with applications in a wide range of fields, e.g. engineering, product design. In early 3D model generation systems, new models were usually generated by mixing up several parts from the existing models. With the emergence of depth sensors, such as Microsoft Kinect and 3D LiDAR, it becomes possible to reconstruct 3D models from lower-cost captured RGB-D images or point clouds. However, processing the sensor captured images or point clouds is kind of complicated and time-consuming, especially in some state-of-the-art methods that infer 3D models from multi-view images or depth maps. In this work, we consider constructing a generative model that could effectively synthesize high-quality 3D models without any image or depth map inputs.

The success of generative adversarial networks (GANs) (Goodfellow et al. 2014) in computer vision field provides us a hint to learn a generator via an adversarial process. By mapping a low-dimensional vector into a much more complex target space, GANs have been proved its powerful generative ability with a number of applications mostly in 2D image or text domain. 3D-GAN (Wu et al. 2016) and PrGANs (Gadelha, Maji, and Wang 2016) are the first two attempts to apply GANs technique on addressing 3D model generation problem. Though their works are inspiring, most of their generated models are incomplete with some holes or multiple fragments. The causes might be that 1) they limit their generators to be trained only on a single domain data (3D models or projected images), and 2) their generators are driven by uninformative random vectors. Different from their works, we propose to learn an image-enhancer-driven 3D model generator from both 2D image (learned) features and 3D volumetric data.

In this paper, we build our framework on 3D generative adversarial networks with an enhancer network for better training a 3D model generator. The enhancer contains two deep convolutional neural networks, and learns features from images in an adversarial manner. The high-level learned image features from the enhancer are fed into the 3D model generator for better generation. We train the two networks together, so that our 3D model generator can be learned from 3D data and 2D data simultaneously. Once the framework has been trained, given a random vector, the enhancer first generates corresponding high-level image features, and then the 3D model generator can synthesize a volumetric 3D model based on the image features.

To comprehensively validate the performance of our proposed method, we conduct experiments on two large-scale datasets for different tasks, including 3D model generation, shape classification and shape retrieval. For the generation task, we train our proposed framework on 3D models and their rendered images from ShapeNet, and then use the trained generator and partial enhancer to synthesize volumetric models. The generation results suggest that our proposed method is able to generate high-quality 3D models. For shape classification and shape retrieval tasks, we train the framework on models and rendered images from major categories of ShapeNet, but test it on ModelNet dataset by extracting deep learned features from the trained discriminator as shape representations. We report quantitative analysis of shape classification and shape retrieval on two popular subsets of ModelNet dataset (ModelNet10

*Corresponding Author (Email: yfang@nyu.edu)

and ModelNet40). Our method achieves impressive performance over other unsupervised state-of-the-art approaches on shape classification and retrieval. In addition, we further verify the effectiveness of the enhancer by conducting experiments with the same setting using our framework without enhancer. The large gap of performances demonstrates that our enhancer can improve the training power of the framework.

In summary, the main contributions of our work are threefold:

- To address the challenging 3D model generation problem, we propose to learn a GAN-based 3D model generator from 2D images and 3D models simultaneously.
- Instead of directly using the uninformative random vectors, we introduce an image-enhancer-driven framework, where an enhancer network learns and feeds the image features into the 3D model generator for better training.
- The experimental results demonstrate that our proposed framework can synthesize high-quality 3D models. Moreover, the unsupervised shape features learned by our framework can achieve superior performance over most of the state-of-the-art methods for shape classification and shape retrieval on ModelNet dataset.

Our paper is organized as follows: in Section Related Work, we review some recent works and concepts closely related to our work, including generation models that inferred 3D models from images or depth maps or random vectors. In Section Approach, we describe the pipeline and technical details of our approach. In Section Experiments, we provide the qualitative 3D model generation result, evaluate the learned features on shape classification as well as shape retrieval, and analyse the influences when feeding image features from different layers of the enhancer. Finally, we conclude our work in Section Conclusion.

Related Work

As one of the most significant topics in 3D computer vision area, 3D model generation has received many attentions for years. Early attempts to generate 3D models were mostly based on some templates or parts of existing 3D models, which synthesized new models by replacing or combining some parts of original models (Chaudhuri et al. 2011; Funkhouser et al. 2004; Kalogerakis et al. 2012; Kim et al. 2013). With the advances in depth sensors, RGB-D images of 3D models can be acquired easily, and as a consequence, some researcher started to make some efforts on inferring 3D models from RGB images or depth maps. For example, the system of Kar et al. (Kar et al. 2015) first segmented objects from the input image, then predicted the viewpoint of the image, and finally generated the 3D model from Silhouettes. In another example, Huang et al. (Huang, Wang, and Koltun 2015) reconstructed 3D models from web image by estimating the viewpoint on the image and then matching correspondence between the image and existing models.

On the other hand, inspiring by the great success of applying deep learning techniques on various applications in graphic and vision community, such as retrieval, classification (LeCun et al. 1998; Sermanet et al. 2013; Fang et al. 2015; Xie et al. 2015; Zhu et al. 2015; Xie et al. 2017), many researchers also tried to develop generative models using deep learning techniques in their recent works. For example, Choy et al. (Choy et al. 2016) proposed a recurrent deep neural network to learn a mapping from image to volumetric 3D object generation. Based on image input, Fan et al. (Fan, Su, and Guibas 2016) also used deep neural network to generate 3D models of point clouds. Different from the above methods that worked on static models and images, Slavcheva's work (Slavcheva et al. 2016) focused on real-time 3D object generation. For the depth image field, Wu et al. (Wu et al. 2015) pre-trained a deep neural network on volumetric models to learn shape representations, and then used the pre-trained network to generate and complete a volumetric model from a single depth image. Although recent efforts on deep learning have made impressive progress on tackling 3D model generation problem, most of the current existing methods require images or depth maps as inputs when generating models.

Comparing to methods inferring 3D models from images or depth maps, it is much more difficult to learn a generation model that can synthesize 3D models without image inputs. However, the recent advance of generative adversarial networks (GANs) technique (Radford, Metz, and Chintala 2015; Shrivastava et al. 2016) provides us a great platform to implement such kind of generative model. 3D-GAN (Wu et al. 2016) is the first work to apply GANs technique in 3D model generation task, where a classic GANs architecture is trained to map low-dimensional probabilistic space to 3D model space. They trained their generation model only on 3D data. Another attempt is PrGANs(Gadelha, Maji, and Wang 2016), where the authors trained a projector together with a GANs framework. Their generator learned to generate 3D models, while their discriminator was trained to distinguish projected images of real models from those projected from generative models. In this paper, we also focus on the challenging 3D model generation problem. Desiring the generative power of the GANs architecture, we build our framework on the GANs architecture, and introduce an enhancer that feeds high-level image features into 3D model generator for better training. Different from the above two GAN-based 3D model generation approaches, our framework is trained on both images and 3D models simultaneously.

Approach

In this section, we provide details of our method for 3D model generation. We briefly describe the basic structure and concepts of general generative adversarial networks followed by the presentation for our proposed framework architecture.

Generative Adversarial Networks (GANs)

Proposed by Goodfellow et al. (Goodfellow et al. 2014), a classic generative adversarial networks (GANs) consist of one generator G and one discriminator D, and both of the generator and discriminator are multilayer neural networks. Let $\boldsymbol{x}$ represents the real data (e.g. image, 3D model), and

$\boldsymbol{z}$ be a vector randomly sampled from a uniform distribution or Gaussian distribution. The generator takes $\boldsymbol{z}$ as input, and outputs a generative data $G(\boldsymbol{z})$, where $G(\boldsymbol{z})$ should have the same format as real data $\boldsymbol{x}$. The discriminator takes either real data $\boldsymbol{x}$ or generative data $G(\boldsymbol{z})$ as input, and outputs a confidence score (denoted as $D(\cdot)$) whether the input data is real or not. Ideally, the score is 1 when the discriminator considers the input data is more like a real data. Otherwise, the score is 0. During the training process, the generator is learning to synthesize data $G(\boldsymbol{z})$ as real as real data $\boldsymbol{x}$, while the discriminator is learning to improve its distinguish ability of real data. The generator and discriminator are usually trained as a two-player minimax game with a competing loss as

$$\min_G \max_D L = E_{\boldsymbol{x}\sim p_x} log D(\boldsymbol{x}) + E_{\boldsymbol{z}\sim p_z} log(1 - D(G(\boldsymbol{z}))). \tag{1}$$

The optimization of above loss function can be solved by applying classical back-propagation algorithm. The parameters in the generator and the discriminator will be updated separately in each epoch. Global optimum of the parameters can be acquired when the generative data distribution $p_{G(z)}$ is equal to the real data distribution p_x ($p_{G(z)} = p_x$).

Enhancer Driven 3D Model Generation

By introducing an enhancer into a 3D model generator, our approach aims to utilize the features learned from image to improve the generative ability of the 3D model generator. Different from the traditional GAN architecture taking uninformative random vectors as inputs, we feed the learned image features into the 3D generator for better generation. Therefore, our 3D model generator can be learned not only from 3D data but also 2D images. Figure 1 shows the pipeline of our proposed method, including the training framework and those used for 3D model generation and shape feature extraction (testing). Figure 1a is the network architecture of our method for training, which mainly consists of three parts: an enhancer, a 3D model generator and a 3D model discriminator. We will present the structure of the three parts separately in detail below.

Enhancer The purpose of the enhancer is to learn and feed image features into the generator for better training without supervision. We construct our enhancer with two deep neural networks, which are trained in an adversarial manner. We call them enhancer-generator G_E and enhancer-discriminator D_E. The input to the enhancer is a 100-dimension vector $\boldsymbol{z}$ randomly sampled from uniform distribution $\boldsymbol{z} \in \cup[-1, 1]$. For the enhancer-generator G_E, it has six deconvolution layers with different numbers of channels $\{2048, 1024, 512, 256, 128, 3\}$, same kernel size (5×5) and stride (2). Tangent activation is applied on the output of last deconvolution layer to synthesize a $128 \times 128 \times 3$ image. The structure of the enhancer-discriminator D_E is similar to the G_E but with four convolution layers. The channel sizes in the enhancer-discriminator D_E are set to $\{64, 128, 256, 512\}$ in each convolution layer, respectively. A fully-connected layer is attached after the final convolution layer to compute a final output, which is an estimated probability $D_E(\cdot)$ of whether the image is real or not. The purpose of D_E is to help the enhancer-generator G_E better learn the high-level features from images via an adversarial manner. The loss function of the enhancer is described as

$$\begin{aligned}\min_{G_E} \max_{D_E} L_E =& E_{\boldsymbol{x}\sim p_x} log D_E(\boldsymbol{x}) + \\ & E_{\boldsymbol{z}\sim p_z} log(1 - D_E(G_E(\boldsymbol{z}))),\end{aligned} \tag{2}$$

where $\boldsymbol{x}$ here represents a real image from training dataset, and $G_E(\boldsymbol{z})$ denotes a generative image synthesized from G_E. For convenience, we construct an image training dataset using images rendered from 3D training models to train the enhancer. Batch normalization and ReLU layer are added between each two (de)convolution layers.

3D model generator and discriminator Our 3D model generator G_M is a deep neural network that maps the outputs of the enhancer-generator (learned image features) into a complex volumetric 3D space. It includes four deconvolution layers with channel size $\{256,128,64,1\}$, kernel size $4 \times 4 \times 4$ and stride 2. Batch normalization and ReLU layer are added to connect deconvolution layers. A Sigmoid layer is applied after the final deconvolution layer. Taking image features from the i^{th} layer of the enhancer-generator $G_E^i(\boldsymbol{z})$ as input, the 3D model generator is able to calculate a $64 \times 64 \times 64$ volumetric 3D model (denoted as $G_M(G_E^i(\boldsymbol{z}))$), as seen in Figure 1a. The 3D model generator is optimized by a 3D model discriminator D_M with opposite structure of the generator. There are four convolution layers in the 3D model discriminator D_M with channel size $\{64,128,256,512\}$, kernel size $4 \times 4 \times 4$ and stride 2. The output of the final convolution layer then passes through a fully connected layer to calculate a probability $D_M(\cdot)$. The loss function for the 3D model generator and discriminator network can be designed as

$$\begin{aligned}\min_{G_M} \max_{D_M} L_M =& E_{\boldsymbol{y}\sim p_y} log D_M(\boldsymbol{y}) + \\ & E_{\boldsymbol{z}\sim p_z} log(1 - D_M(G_M(G_E^i(\boldsymbol{z})))),\end{aligned} \tag{3}$$

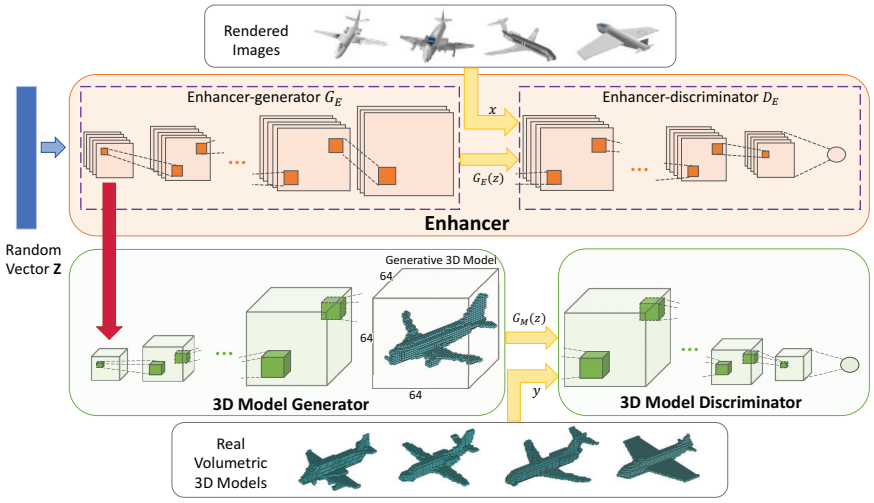
where $\boldsymbol{y}$ is a $64 \times 64 \times 64$ 3D model voxelized from a training model, $D_M(\boldsymbol{y})$ denotes the probability that input model $\boldsymbol{y}$ is real, and $D_M(G_M(G_E^i(\boldsymbol{z})))$ represents the probability that generated model $G_M(G_E^i(\boldsymbol{z}))$ is real.

Learning We train the enhancer network and the 3D model generator and discriminator network together by optimizing the following objective function

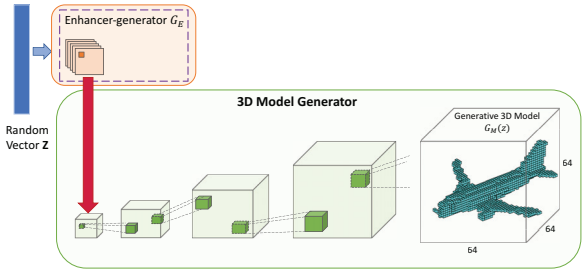
$$\min_{G_E, G_M} \max_{D_E, D_M} L = L_E + L_M. \tag{4}$$

For generators (G_E and G_M), they are optimized towards minimizing the value of the objective function, while discriminators (D_E and D_M) are trained to maximize the value of the objective function.

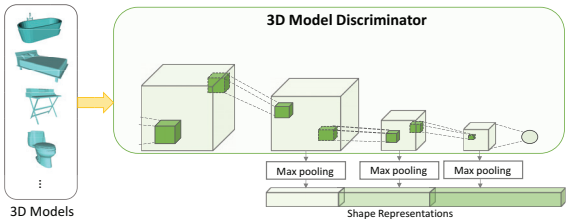
We use ADAM (Kingma and Ba 2014) optimizer to obtain the optimal network parameters with beta value $\beta = 0.5$ and learning rate 0.0002 for generators and discriminators. Observing that the discriminators always learn faster than the generators, we use a simple but useful strategy that updates the generators twice more than the



(a) The framework of our proposed method for training. It consists of three parts: an enhancer, a 3D model generator and a 3D model discriminator. The enhancer contains two deep neural networks and learns features from rendered images via an adversarial manner. The 3D model generator is trained on 3D data with the 3D model discriminator. By feeding the outputs from the first layer of the enhancer into the 3D model generator, the learned high-level image feature from enhancer can be utilized for better training a 3D model generator.



(b) The framework for 3D model generation (testing). After training, given the outputs of the first layer of the enhancer (computed based on a random vector), our trained 3D model generator is able to synthesize a $64 \times 64 \times 64$ 3D volumetric model.



(c) The framework of feature extraction for classification and retrieval. Given a volumetric 3D model, we concatenate the max pooling outputs of the last three convolution layers in the discriminator as the shape representation.

Figure 1: The framework of our proposed method. Figure 1a is the network architecture of our method for training, while figures 1b and 1c show the frameworks that we use to generate 3D models (testing) and extract features for classification and retrieval (testing), respectively.

discriminators in each batch when training our framework. The batch size is set to 64. We implement our framework using the popular deep learning tool TensorFlow (Abadi et al. 2016) and train it on a desktop with Intel Xeon E5-2603 CPU and NVIDIA Tesla K80 GPU.

Generating 3D model and shape representation Due to the properties of adversarial learning, obtaining a better 3D model generator can result in a greater 3D model discriminator, which can generate more discriminative shape representations. After training, we only use the 3D model generator and partial enhancer for generating 3D models (as shown in Figure 1b). Given a 100-dimension random vector $\boldsymbol{z}$ sampled from uniform distribution $\boldsymbol{z} \in \cup[-1, 1]$, we first compute the outputs $G_E^i(\boldsymbol{z})$ from the trained enhancer, and then our trained 3D model generator can synthesize a volumetric 3D model $G_M(G_E^i(\boldsymbol{z}))$ without any inferences from images. For shape representation, given a volumetric 3D model, we concatenate the outputs after max pooling the last three convolution layers of the 3D model discriminator as the representation (as shown in Figure 1c).

Experiments

To comprehensively validate our proposed framework, we conduct three different experiments on large-scale 3D model datasets, including 3D model generation, shape classification and shape retrieval. We present the experiment settings, qualitative generation result, quantitative analysis for shape classification and retrieval. The experimental result demonstrates that our method can generate high-quality 3D models, and successfully learn unsupervised features as shape representation. In the shape classification and retrieval exper-

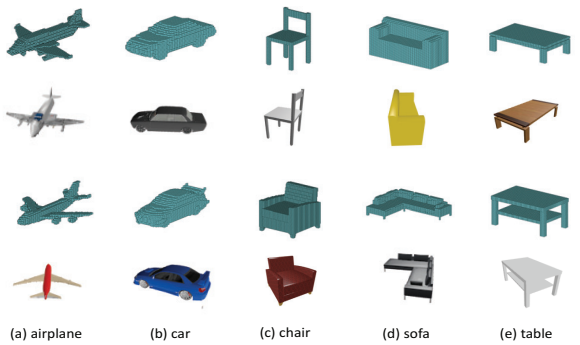
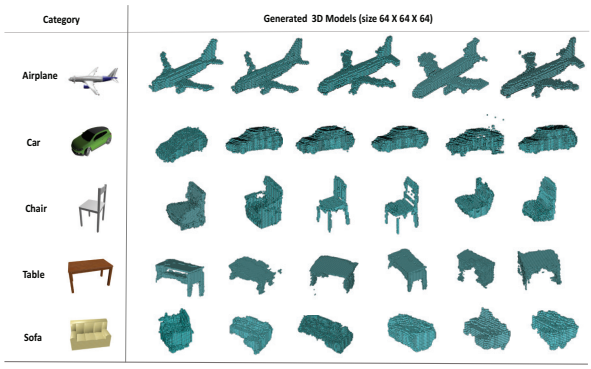
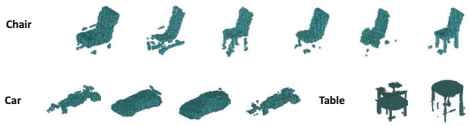


Figure 2: Examples of our training dataset. 3D models from ShapeNet (Chang et al. 2015) dataset are voxelized as $64 \times 64 \times 64$ volumetric ones. Corresponding images for each 3D model are randomly selected from one of 23 views of rendered images in 3D-R2N2 dataset (Choy et al. 2016). As we can see from the figure, view points of images might be different.



(a) Examples of 3D models generated by our proposed method.



(b) Examples of 3D models generated by (Wu et al. 2016).

Figure 3: Comparion of 3D model generation results. Figure 3a shows examples of 3D models generated from our trained generators, one row for each category (e.g. airplane, car, chair, table, sofa). For comparison, we also show some models generated using state-of-the-art 3D-GAN (Wu et al. 2016) (in Figure 3b). Our framework can generate high-quality 3D models with size $64 \times 64 \times 64$, which is comparable even better than those generated from 3D-GAN.

iments, our method outperforms the-state-of-the-arts with highest classification accuracy and retrieval precision. In addition, we verify the effectiveness of the enhancer by comparing shape classification and retrieval results using features extracted from our framework trained with and without enhancer. In all the three experiments, we choose to feed the outputs of the first layer of the enhancer-generator into the 3D model generator. Discussion about the choices of layers is also provided.

3D Model Generation

In this task, we train our proposed framework on large-scale ShapeNet (Chang et al. 2015) dataset that contains more than $50,000$ 3D models with 55 categories. All 3D training models are voxelized as $64 \times 64 \times 64$ volumetric ones. 3D-R2N2 (Choy et al. 2016) provides a dataset that includes rendered images from 3D models in ShapeNet from 23 different views. We randomly select one rendered image for each training model to construct the image dataset to train the enhancer. Some examples of our training dataset are shown in Figure 2. The volumetric models are input into 3D model discriminator, while the rendered images are inputs to the enhancer-discriminator. For each training epoch, volumetric 3D models and their corresponding images are utilized together to train our framework.

To obtain better generative models, we train one 3D model generator for each category. After training, we randomly sample a 100-dimension vector from uniform distribution $[-1, 1]$ and then pass it through the first layer of the enhancer-generator and the 3D model generator to synthesize a volumetric 3D model. No image is needed when generating 3D models.

Figure 3 shows some generative models for major categories in ShapeNet. Models in Figure 3a are generated by our proposed method, including airplane, car, chair, table and sofa. The generation result suggests that our 3D model generator can synthesize varied 3D models with high resolution size $64 \times 64 \times 64$. For comparison, we also show some models generated by the state-of-the-art 3D-GAN (Wu et al. 2016). All 3D models in Figure 3b are synthesized using the pretrained generators provided by the authors. As we can see from the figure, our framework can generate high-quality 3D models, which is comparable even better than those generated from 3D-GAN. In addition, we also observe that most of the table models generated by (Wu et al. 2016) are more likely to be small side tables, but ours are bigger rectangle tables. The reason could be most of table models in their training dataset are manually selected side tables, but we used the table dataset in ShapeNet where most of the tables have rectangle shapes. Due to a lack of quantitative criteria to evaluate the generation quality, we below provide shape classification and retrieval results for quantitative comparison.

Shape Classification

Following the experiment setting in 3D-GAN (Wu et al. 2016) for fair comparison, we pretrain a framework on 3D models and their rendered images from seven major categories (e.g. chair, couch, gun, airplane, watercraft, table and car) in ShapeNet without label information. A max pooling layer is added after each convolution layer of the trained 3D model discriminator with kernel sizes {8,4,2}, stride size {4,2,1}, respectively. Then, we input 3D models from ModelNet (Wu et al. 2015) into the trained 3D

Table 1: Performance comparison of shape classification with state-of-the-art methods on two benchmarks (ModelNet10 and ModelNet40) of ModelNet dataset.

Method	Supervised?	ModelNet10 (%)	ModelNet40 (%)
3D ShapeNets(Wu et al. 2015)	✓	83.54	77.32
VoxNet (Maturana and Scherer 2015)	✓	92.00	83.00
Geometry Image (Sinha, Bai, and Ramani 2016)	✓	88.40	83.90
PointNet (Qi et al. 2017)	✓	77.60	–
GIFT (Bai et al. 2016)	✓	92.35	83.10
FusionNet (Hegde and Zadeh 2016), fine-tuned	✓	93.11	90.80
SPH (Kazhdan, Funkhouser, and Rusinkiewicz)	×	79.79	68.23
LFD (Chen et al. 2003)	×	79.87	75.47
VConv-DAE (Sharma, Grau, and Fritz 2016)	×	80.50	75.50
3D-GAN (Wu et al. 2016)	×	91.00	83.30
Our Method without Enhancer	×	**88.88**	**85.53**
Our Method with Enhancer	×	**91.63**	**87.85**

model discriminator and concatenate the features extracted from each convolution layer (after max pooling) as shape representations. Finally, we train a linear SVM upon the extracted $57,344$-dimensional features of models from ModelNet training sets, while the 3D models from the ModelNet test sets are used for testing.

We apply our proposed framework on two popular subsets of ModelNet (ModelNet10 and ModelNet40) for shape classification, and present the comparison with other state-of-the-art methods. The ModelNet10 subset contains $4,899$ models from 10 different categories, which are split into a training set with $3,991$ models and a testing set with 908 models. The ModelNet40 subset has a total of $12,311$ models from 40 categories, split into a training set and a testing set with size $9,843$ and $2,468$, respectively. We report the classification performance on testing sets in Table 1. The classification accuracy of our method is pretty high (91.63% on ModelNet10 and 87.85% on ModelNet40), which demonstrates that the shape features our framework learned are highly discriminative.

We collect the publicly available results of state-of-the-art approaches from the ModelNet dataset website [1] for comparison, including supervised methods and unsupervised descriptors. Seen from Table 1, though our method is training on a subset of ShapeNet, it can obtain comparable performance with some supervised methods, such as VoxNet (Maturana and Scherer 2015), GIFT (Bai et al. 2016) and FusionNet (Hegde and Zadeh 2016), and get higher classification accuracy than other supervised methods, e.g., 3D ShapeNets (Wu et al. 2015), Geometry Image (Sinha, Bai, and Ramani 2016), PointNet (Qi et al. 2017). Besides, we compare our method with some state-of-the-art unsupervised methods, such as SPherical Harmonic descriptor (SPH) (Kazhdan, Funkhouser, and Rusinkiewicz), Light Field Descriptor (LFD) (Chen et al. 2003), VConv-DAE (Sharma, Grau, and Fritz 2016) and 3D-GAN (Wu et al. 2016). Our method achieves the best performance over above methods on both ModelNet10 and ModelNet40 dataset. We also provide the classification performance using our framework without enhancer with exactly same experimental settings as the one with enhancer, e.g., batch size, learning rate, epoch. Same max pooling are applied to exact features. Though our framework without enhancer has the same architecture as 3D-GAN, it obtains a worse classification accuracy in ModelNet10 and a better accuracy in ModelNet40 than that reported in 3D-GAN paper. The reason could be the differences of initial parameter setting, training strategy and the number of training epochs. Importantly, the improvement of the performance using the framework with the enhancer clearly demonstrates the effectiveness of the enhancer.

[1] http://modelnet.cs.princeton.edu

Shape Retrieval

In addition to shape classification, we use the learned features for shape retrieval on ModelNet10 and ModelNet40 datasets. In this task, we extract shape features following the same way as mentioned in above shape classification task. Models in test sets are used as queries to retrieve relevant models in the same set. For each pair of models, Euclidean distance between their $57,344$-dimensional representations is calculated. The smaller the distance is, the more relevant the two models are. For each query, we can obtain a ranked list of models based on the calculated distance in an ascending order. Only models from the same category as the query's are considered as relevant models. The best case is that all relevant models are ranked at the top of the retrieval list.

To evaluate the performance of retrieval, we compute retrieval precision for each query and report the average in Table 2. Our method (with enhancer) achieves high precision 65.00% in ModelNet10 with large margin over other unsupervised state-of-the-art methods, such as 20% higher than SPherical Harmonic descriptor (SPH) (Kazhdan, Funkhouser, and Rusinkiewicz), 15% higher than Light Field Descriptor (LFD) (Chen et al. 2003). For ModelNet40, our method obtain a 44.44% MAP, which is 10% and 4% higher than SPH and LFD, respectively. Since the author of 3D-GAN did not provide a pretrained model for feature extraction or source code of a trainable model, we cannot obtain the retrieval performance using the original 3D-GAN

Table 2: Mean average precision (MAP) comparison of shape retrieval with state-of-the-art unsupervised methods on two benchmarks (ModelNet10 and ModelNet40) of ModelNet dataset.

Method	retrieval MAP (%)	
	ModelNet10	ModelNet40
SPH	44.05	33.26
LFD	49.82	40.91
Ours without Enhancer	**61.82**	**40.81**
Ours with Enhancer	**65.00**	**44.44**

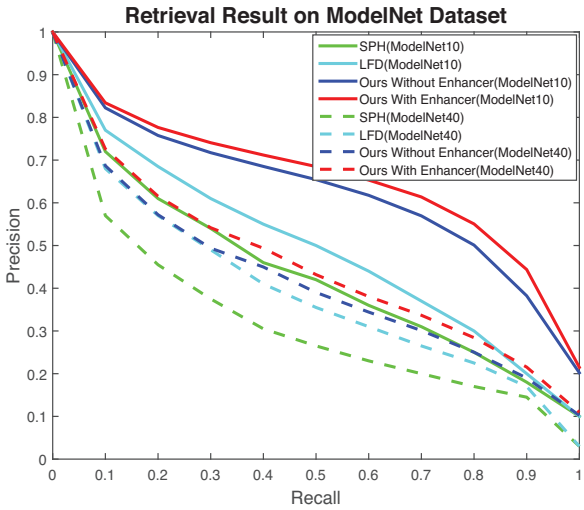


Figure 4: Precision-Recall plots for shape retrieval comparison of state-of-the-art methods on two benchmarks (ModelNet10 and ModelNet40) of ModelNet dataset.

model. However, our method without enhancer actually is a self-implemented version of 3D-GAN, so we provide the performance using our method without enhancer as a comparison. As we can see in the Table 2, our method without enhancer can obtain 61.82% precision in ModelNet10 dataset and 40.81% in ModelNet40 dataset. Although performance of our method without enhancer is comparable with state-of-the-art unsupervised descriptors, our method with enhancer performs the best. Moreover, the large gap of the MAPs between them illustrates that the augmented enhancer can significantly improve the learning performance.

Precision-recall (PR) curve is usually used to visually indicate the relation between precision and recall for all queries. We plot the PR-curves of all compared methods in Figure 4, where our method outperforms other unsupervised approaches with more than 10% in ModelNet10 and more than 5% in ModelNet40 when recall reaches 1.0.

Analysis on Image Feature Layers

In this subsection, we discuss the influences when different image feature layer outputs are chosen to feed into the model generator. We remain the same network structure of the 3D model generator and discriminator (as mentioned in Section Approach) but change some channel sizes of the enhancer-generator, so that the outputs from different layers of the enhancer-generator can be fed into the 3D model generator. We report the classification performances when training our method with the image features generated from $1st$ to $3rd$ layer of the enhancer-generator in Table 3. As we can see from the table, the accuracy has a slight decrease when feeding the image features from the layer closer to the final output of the enhancer-generator. It is reasonable because the layer closer to the final output generates lower-level features, which looks more like an "image" but further away from a better latent space. Therefore, the framework would generate easy-identified "unreal" 3D models. As a consequence, the discriminator cannot be improved. Since our proposed method obtains the best performance when training with the image features from the first layer, we choose to feed the outputs from the first layer of the enhancer-generator into the 3D model generator in other experiments, such as generation, classification and retrieval.

Table 3: Shape classification accuracy comparison of our proposed method with image features generated from different layer of the enhancer-generator on ModelNet dataset.

Image Feature Layer	Classification Accuracy (%)	
	ModelNet10	ModelNet40
None	88.88	85.53
1st	**91.63**	**87.85**
2nd	91.52	87.12
3rd	90.42	86.34

Conclusion

In this paper, we tackle the challenging 3D model generation problem by learning a 3D model generator with image features. We propose to design an enhancer to learn features from rendered images and feed the high-level image features generated from the first layer of the enhancer into the 3D model generator for better training. After training, given random vector inputs, the trained 3D model generator can be used to synthesize volumetric models based on the first layer outputs of the enhancer-generator. The qualitative generation results on ShapeNet demonstrate that our proposed framework is able to generate high-quality 3D models with high resolution. Moreover, we use the shape features learned from our framework to classify and retrieve shapes on two subsets of ModelNet dataset, including ModelNet10 and ModelNet40. The superior classification and retrieval performance over state-of-the-art methods suggests that our framework can learn a highly discriminative shape representation without supervision. In order to verify the effectiveness of our designed enhancer, we compare the shape classification and retrieval performances of our frameworks with enhancer and without enhancer. The higher classification and retrieval accuracies imply the training power of the

enhancer.

References

Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Bai, S.; Bai, X.; Zhou, Z.; Zhang, Z.; and Latecki, L. J. 2016. Gift: A real-time and scalable 3d shape search engine.

Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; Xiao, J.; Yi, L.; and Yu, F. 2015. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago.

Chaudhuri, S.; Kalogerakis, E.; Guibas, L.; and Koltun, V. 2011. Probabilistic reasoning for assembly-based 3d modeling. In *ACM Transactions on Graphics (TOG)*, volume 30, 35. ACM.

Chen, D.-Y.; Tian, X.-P.; Shen, Y.-T.; and Ouhyoung, M. 2003. On visual similarity based 3d model retrieval. In *Computer graphics forum*, volume 22, 223–232. Wiley Online Library.

Choy, C. B.; Xu, D.; Gwak, J.; Chen, K.; and Savarese, S. 2016. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European Conference on Computer Vision*, 628–644. Springer.

Fan, H.; Su, H.; and Guibas, L. 2016. A point set generation network for 3d object reconstruction from a single image. *arXiv preprint arXiv:1612.00603*.

Fang, Y.; Xie, J.; Dai, G.; Wang, M.; Zhu, F.; Xu, T.; and Wong, E. 2015. 3d deep shape descriptor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2319–2328.

Funkhouser, T.; Kazhdan, M.; Shilane, P.; Min, P.; Kiefer, W.; Tal, A.; Rusinkiewicz, S.; and Dobkin, D. 2004. Modeling by example. In *ACM Transactions on Graphics (TOG)*, volume 23, 652–663. ACM.

Gadelha, M.; Maji, S.; and Wang, R. 2016. 3d shape induction from 2d views of multiple objects. *arXiv preprint arXiv:1612.05872*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Hegde, V., and Zadeh, R. 2016. Fusionnet: 3d object classification using multiple data representations. *arXiv preprint arXiv:1607.05695*.

Huang, Q.; Wang, H.; and Koltun, V. 2015. Single-view reconstruction via joint analysis of image and shape collections. *ACM Transactions on Graphics (TOG)* 34(4):87.

Kalogerakis, E.; Chaudhuri, S.; Koller, D.; and Koltun, V. 2012. A probabilistic model for component-based shape synthesis. *ACM Transactions on Graphics (TOG)* 31(4):55.

Kar, A.; Tulsiani, S.; Carreira, J.; and Malik, J. 2015. Category-specific object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1966–1974.

Kazhdan, M.; Funkhouser, T.; and Rusinkiewicz, S. Rotation invariant spherical harmonic representation of 3 d shape descriptors.

Kim, V. G.; Li, W.; Mitra, N. J.; Chaudhuri, S.; DiVerdi, S.; and Funkhouser, T. 2013. Learning part-based templates from large collections of 3d shapes. *ACM Transactions on Graphics (TOG)* 32(4):70.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

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

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation.

Radford, A.; Metz, L.; and Chintala, S. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; and LeCun, Y. 2013. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

Sharma, A.; Grau, O.; and Fritz, M. 2016. Vconv-dae: Deep volumetric shape learning without object labels. In *Computer Vision–ECCV 2016 Workshops*, 236–250. Springer.

Shrivastava, A.; Pfister, T.; Tuzel, O.; Susskind, J.; Wang, W.; and Webb, R. 2016. Learning from simulated and unsupervised images through adversarial training. *arXiv preprint arXiv:1612.07828*.

Sinha, A.; Bai, J.; and Ramani, K. 2016. Deep learning 3d shape surfaces using geometry images. In *European Conference on Computer Vision*, 223–240. Springer.

Slavcheva, M.; Kehl, W.; Navab, N.; and Ilic, S. 2016. Sdf-2-sdf: Highly accurate 3d object reconstruction. In *European Conference on Computer Vision*, 680–696. Springer.

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1912–1920.

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

Xie, J.; Fang, Y.; Zhu, F.; and Wong, E. K. 2015. Deepshape: Deep learned shape descriptor for 3d shape matching and retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, 1275–1283.

Xie, J.; Dai, G.; Zhu, F.; Wong, E. K.; and Fang, Y. 2017. Deepshape: Deep-learned shape descriptor for 3d shape retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* 39(7):1335–1345.

Zhu, J.; Zhu, F.; Wong, E. K.; and Fang, Y. 2015. Learning pairwise neural network encoder for depth image-based 3d model retrieval. In *Proceedings of the 23rd ACM international conference on Multimedia*, 1227–1230. ACM.