International Journal of Computational Intelligence Systems
Vol. **12(2)**, 2019, *pp.* 697–705
DOI: https://doi.org/10.2991/ijcis.d.190617.001; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

3D Model Generation and Reconstruction Using Conditional Generative Adversarial Network

Haisheng Li[1,2,3,*], Yanping Zheng[1,2,3], Xiaoqun Wu[1,2,3], Qiang Cai[1,2,3]

[1]*School of Computer and Information Engineering, Beijing Technology and Business University, No. 33, Fucheng Road, Haidian District, Beijing, 100048, China*
[2]*Beijing Key Laboratory of Big Data Technology for Food Safety, No. 33, Fucheng Road, Haidian District, Beijing, 100048, China*
[3]*National Engineering Laboratory For Agri-product Quality Traceability, No. 33, Fucheng Road, Haidian District, Beijing, 100048, China*

ARTICLE INFO

Article History
Received 05 May 2019
Accepted 08 Jun 2019

Keywords
3D model generation
3D model reconstruction
Generative adversarial network
Class information

ABSTRACT

Generative adversarial network (GANs) has significant progress in 3D model generation and reconstruction recently years. GANs can generate 3D models by sampling from uniform noise distribution. But they generate randomly and are often not easy to control. To address this problem, we add the class information to both generator and discriminator and construct a new network named 3D conditional GAN. Moreover, to better guide generator to reconstruct 3D model from a single image in high quality, we propose a new 3D model reconstruction network by integrating a classifier into the traditional system. Experimental results on ModelNet10 dataset show that our method can effectively generate realistic 3D models corresponding to the given class labels. And the qualities of 3D model reconstruction have been improved considerably by using proposed method in IKEA dataset.

1. INTRODUCTION

Compared with texts or images, three-dimensional (3D) scenes composed by 3D models can provide more details since they well match the human visual perception. With the rapid development of computer storage space and 3D data acquisition technology, the number of 3D models increases significantly. 3D models become more and more important and have wide applications in the fields of industrial product design, cultural relics restoration, medical diagnosis, 3D games, and so on. The traditional way of designing and constructing 3D models is very cumbersome, which hinders the enthusiasm of ordinary users for creative design and the satisfaction of 3D models that meet their own requirements. It is not beneficial to the popularization and further application of 3D models. In recent years, people can use CATIA, UG, SolidWorks, MAYA, and other modeling software or 3D scanners to obtain digital 3D models, but it is generally very tedious and exhaustive. Therefore, exploring effective 3D model generation methods is an important topic in the field of computer graphics and computer vision [1].

Since AlexNet [2] achieved excellent results in the 2012 ImageNet [3] competition, deep learning-based algorithms have been widely used in various fields and establish their dominance in 2D computer vision problems. The core idea of deep learning is to make the output of low-level arithmetic units as the input of high-level arithmetic units, by passing through multi-layer nonlinear operation units. It is data-driven and abstract features can be obtained with less domain knowledge [4]. Recently, lots of work aims to utilize those algorithms to address the problems of understanding 3D scenes, such as 3D object retrieval, 3D shape classification/recognition, 3D shape segmentation [5]. As an important embranchment of deep learning, deep generative models have been proven that they have definite advantages whether in dimension reduced analysis, information retrieval, or feature extraction. By employing deep generative models to extract the structure features of 3D models on existing datasets, it becomes possible to automatically generate 3D models that conform to the semantic constraints.

As a typical representative of deep generative models, generative adversarial networks (GANs) are considered to have the ability to generate new data by effectively learning the data distribution, and many efforts have been made in images synthesis, such as EBGAN [6], LSGAN [7], and pix2pix [8]. 3D shape generation and reconstruction is one of key research topics, while using traditional GAN to generate 3D models, the results may be random and uncontrollable [9]. Because generator samples vectors by random guessing.

In this paper, we add constraints in training of 3D model generation to better satisfy the requirements and avoid mode collapse. Our method takes class information (labels) as input, and then generates realistic and novel 3D models. Generated models are similar but different from models in training set.

Further, we explore generative network for 3D reconstruction based on a single image. We combine conditional variational auto-encoder (CVAE) [10] with GAN. Encoder is trained to map input images to a latent space which contains detail 3D structure.

*Corresponding author. Email: lihsh@th.btbu.edu.cn

Generator, which acts as decoder, recovers 3D volume according to the input image. Discriminator learns to determine the plausibility of generated 3D shape, and discriminate between models generated and those from dataset. In addition, we adapt a 3D volume classification network to keep the category features and ensure the correspondence between generated models and input images. By synchronizing among encoding, generation, discrimination, and classification tasks, our reconstruction network is more stable and can get higher quality models.

The main contributions can be summarized as follows:

- A 3D conditional GAN by adding class information is proposed in this paper. It enables the network to learn complex data distribution in diverse model categories which is hard to train by using traditional GAN.
- We also propose a new 3D model reconstruction network which consists of an encoder, a generator, a discriminator, and a classifier. The network relies on an improved loss function to establish convergence in four major components.
- Experiments in ModelNet10 dataset [11] and IKEA dataset [12] demonstrate that the proposed method is effective. And we achieve more satisfactory results outperforming state-of-the-art methods with a mean average precision score increases 9.2% on the IKEA dataset.

2. RELATED WORK

2.1. 3D Model Reconstruction

Previous work of processing 3D data usually utilized multiple views of objects [13]. And 3D model reconstruction methods based on multiple view geometry (MVG) [14] have been widely applied in many areas. Early work recovered geometric shapes by extracting and matching dense features from multiple views, or minimizing the reprojection error of models directly. They have been successfully used in sfM [15] and SLAM [16] for large-scale scene reconstruction or navigation. Nevertheless, it's difficult to get enough views of objects and cannot reconstruct the unseen part, which limits the application of the MVG. While learning-based approach can break this limitation.

Kar *et al.* [17] proposed a 2D–3D correspondence method to reconstruct 3D models. The estimated instance segmentation and predicted viewpoints that obtained from the input image were used to generate a high frequency 2.5D depth map and a complete 3D mesh. Fan *et al.* [18] proposed PointOutNet to recover 3D models from images. Models are represented by point set. The network took a single image (RGB or RGB-D) as input, then the positions of points in 3D space were determined by the image and inferred viewpoint position. In MarrNet, proposed by Wu *et al.* [19], objects' normal, depth, and silhouette sketches were first recovered from RGB images to compose the 2.5D sketches, then 3D models can be reconstructed from predicted 2.5D sketches. The reprojection consistency loss function was applied to ensure the estimated 3D models align 2.5D sketches. Kato *et al.* [20] proposed a rasterized approximation gradient that allows rendering to be integrated into the neural network so that the mesh model can be used directly as input to the neural network. By using this renderer, the offset of a given spherical mesh vertex can be predicted to reconstruct 3D mesh model from a single image, without 3D supervision.

For learning-based 3D model generation and reconstruction, the representation of 3D shapes plays a key role. Using point sets to represent 3D models is flexible for geometric transformations and deformations, but it may be inefficient to represent continuous 3D geometry because of losing surface connectivity [18]. Mesh-based generative methods can infer the surface representation naturally, while they are limited to the topology of given template mesh and require complicated network architectures [21]. We address the problem of generating 3D models based on volumetric representations, for it's easy to learn and can be simply employed in GANs to generate realistic and novel models.

2.2. 3D GANs

The adversarial architecture was first proposed by Goodfellow *et al.* [22], and its main idea is to simultaneously train two models, the generator and the discriminator, and make them both stronger in adversarial learning. Then, DCGAN was proposed by Radford *et al.* [23], which implemented by deep convolutional layer. It has excellent performance in generating images and been popular employed in solving 2D-view problems. The traditional GANs is hard to train and then improved solution WGAN [24] and WGAN-GP [25] were proposed to enable the network training steadily.

3D-GAN [26] applied GAN in learning latent 3D space, and it can generate 3D voxel models from the latent space by extending 2D convolution into 3D convolution. Combining 3D-GAN with WGAN-GP, 3D-IWGAN [27] can generate high-quality 3D models with a more stable training process. Wang *et al.* [28] utilized an encoder–decoder as generator of the adversarial network to address 3D shape inpainting. Then a long-term recurrent convolutional network (LRCN) was employed to refine the generated results to obtain more complete 3D models in higher resolution. Chen *et al.* [29] proposed text2shape system which combined 3D generation with natural language processing. The network encoded the text, then regarded the results as a condition, and utilized WGAN to decode it into a 3D model related to input text.

3. METHODS

3.1. Adversarial Network Architecture

GAN is a framework for generating objects through adversarial process estimation, which has a profound impact on the development of generation methods. GAN is implemented by combining a generator G and a discriminator D. D classifies whether its input is generated or sampled from the "real" data. G captures the data distribution and attempts to falsify the "real" data to cause the discriminator to make a wrong judgment [22]. G and D can be regarded as two players of a min–max game and train simultaneously, the objective is stated as follows:

$$\min_G \max_D V(D,G) = E_{p_r}\left[\log D(x)\right] + E_{p_z}\left[\log\left(1 - D(G(z))\right)\right]. \quad (1)$$

where p_r is the data distribution of training set, and z is a randomly sampled vector from the prior noise distribution p_z, $D(x)$ is the output scalar of D that indicates the possibility of x is sampled from

the training set. And G is trained to build a map of z between data space $G(z)$, minimize probability that D distinguishes $G(z)$ came from p_g rather than p_r.

Training of standard GANs only requires annotation information (true or false) of the data source and is optimized according to discriminator's output. The conditional GAN [9] refers to adding conditions c in real data, G and D. The function of c, which can be class information or other additional information, is used to supervise network training. And the loss function can be defined as

$$L_D^{CGAN} = E_{p_r}\left[\log D\left(x|c\right)\right] + E_{p_z}\left[log\left(1 - D\left(G\left(z|c\right)\right)\right)\right]. \quad (2)$$

For purpose of resolving the gradient disappearance problem that may appear during training, and avoiding instability of the training process, WGAN-GP [25] is utilized to train the whole model. Our objective function is as follows:

$$L_D = -E_{p_r}\left[D\left(x|c\right)\right] + E_{p_z}\left[D\left(G\left(z|c\right)\right)\right] + \alpha E_{p_{\hat{x}}}\left[\left(\|\nabla_{\hat{x}} D\left(\hat{x}\right)\|_2 - 1\right)^2\right]. \quad (3)$$

where $p_{\hat{x}}$ denotes the distribution which is sampled uniformly on a straight line between p_g and p_r. In this work, p_z is defined to the uniform distribution over [0, 1], and the loss of discriminator is added by gradient penalty with $\alpha = 10$ as suggested in WGAN-GP [25].

3.2. 3D Model Generation

In this paper, the generator is learned to map a low dimensional probability space to 3D models, to explore 3D manifold. And models are generated based on the specified condition without reference images and CAD models. The discriminator distinguishes the generated 3D models from "real" data and feeds the result back to the generator for guiding its training.

Figure 1 illustrates the architecture of 3D conditional GAN. Before generator, there's a fully connected layer with 2,048 units, which takes a set of 200-dimensional vectors as input, together with category labels. And the vectors are sampled from a standard normal distribution randomly. The generator consists of four 3D deconvolutional layers with strides two and kernels size of four. All deconvolution layers are separated by a batch normalization layer and a ReLU activation layer, and the last one is followed by one tanh activation layer and outputs the voxel grids of 32^3. The generator's output and category labels are taken as input of discriminator. In the discriminator, there are four 3D convolutional layers, with strides two and kernels size of four, and one fully connected layers. All the convolution are connected by a leaky ReLU activation layer. Outputs of discriminator are the probabilities of whether the input models are from raw data. The detailed configuration and primary parameters of generative network and discriminative network are shown in Tables 1 and 2, respectively. We use proper zero-padding in both networks. And in order for the network to converge more stably, the discriminator trains in each batch, while the generator trains every five batches, following the 3D-IWGAN [27] recommendation. The whole training pipeline is illustrated in Algorithm 1.

Algorithm 1 The training procedure of the 3D model generation network

Initialize: Batch size n; number of classes m; parameter of the generator θ_G; parameter of the discriminator θ_D.
1: **while** algorithm is not convergence **do**
2: Sample $(x_1, c_1), ..., (x_n, c_n)$ from the distribution of real data p_r;
3: Sample z from the prior distribution p_z;
4: $x' \leftarrow G\left(z|c\right)$
5: $L_G \leftarrow D\left(x'\right)$
6: Sample $\hat{x}$ from straight line between p_g and p_r;
7: $L_D \leftarrow -D\left(x|c\right) + D\left(x'|c\right) + \alpha\left(\|\nabla D\left(\hat{x}\right)\|_2 - 1\right)^2$
8: $\theta_D \leftarrow -\nabla_{\theta_D}\left(L_D\right)$
9: **if** iteration % 5 == 0 **then**
10: $\theta_G \leftarrow -\nabla_{\theta_G}\left(L_G\right)$
11: **end while**

Table 1 The network details of generator.

Index	Type	Kernels	Filter Size	Strides
1	FC	2048		
2	3D Deconv	256	$4 \times 4 \times 4$	2
3	3D Deconv	128	$4 \times 4 \times 4$	2
4	3D Deconv	64	$4 \times 4 \times 4$	2
5	3D Deconv	1	$4 \times 4 \times 4$	2

Table 2 The network details of discriminator.

Index	Type	Kernels	Filter Size	Strides
1	3D Conv	32	$4 \times 4 \times 4$	2
2	3D Conv	64	$4 \times 4 \times 4$	2
3	3D Conv	128	$4 \times 4 \times 4$	2
4	3D Conv	256	$4 \times 4 \times 4$	2
5	FC	1		

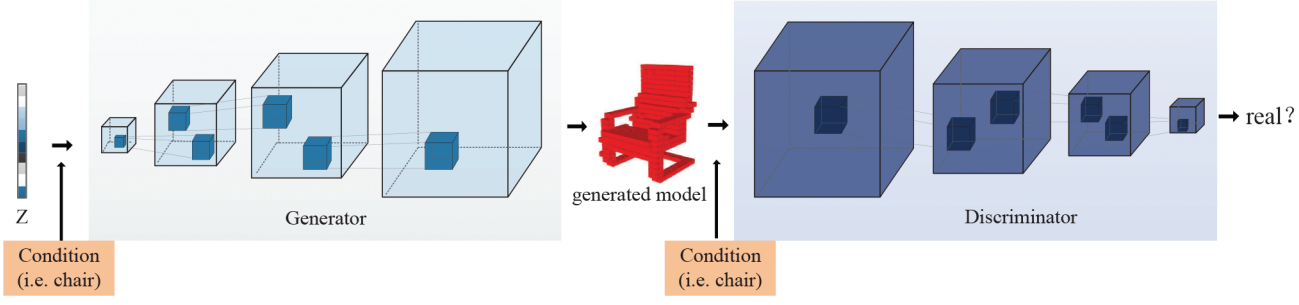

Figure 1 The architecture of three-dimensional (3D) model generation network.

3.3. 3D Object Reconstruction from a Single Image

If a 3D model of a specified condition can be generated from a random vector, it's possible to generate a 3D model corresponding to an image by encoding the image and using encoded result as input to generator. Due to its excellent results in image generation, we modify the encoder of CVAE [10] to get encoding vectors of images in a latent space. And inspired by CVAE-GAN [30], we add a 3D volume classification network beyond above 3D conditional generation network to classify generated model. The classification results are also fed back to the generator to guide the generation of better 3D models. Therefore, the network used for reconstruction tasks has four major components, corresponding to four neural networks, which cooperate and promote each other, as depicted in Figure 2.

3.3.1. Encoder (E)

Given an image i and category label c of the model that is contained in the image, encoder (E) encodes them into a latent variable z. E is built by stacking five 2D convolutional layers, each with a leaky ReLU activation layer. And there are two fully connected layers followed. All layers are separated by a batch normalization layer, and proper zero-paddings are employed in all convolutional layers. More details of the encoder network are illustrated in Table 3. The objective of E is to minimize:

$$L_E = L_{KL} + L_{recon}. \tag{4}$$

where L_{KL} is the KL-divergence that proposed to push the variational encoder distribution p_e towards to the prior distribution p_z. And L_{recon} is presented to minimize the gap between real 3D models and models generated from the encoding. In other words, L_{recon} is tasked with making p_g close to the real distribution of 3D data p_r. They can be defined as

$$L_{KL} = D_{KL}\left(p_e\left(z|i,c\right)|p_z\right). \tag{5}$$

$$L_{recon} = \|G\left(E\left(i,c\right)\right) - x\|_2. \tag{6}$$

where x denotes real 3D model from the training set, c is its category label and i is the corresponding image.

Table 3 The network details of encoder.

Index	Type	Kernels	Filter Size	Strides
1	2D Conv	64	11 × 11	4
2	2D Conv	128	5 × 5	4
3	2D Conv	256	5 × 5	2
4	2D Conv	512	5 × 5	2
5	2D Conv	400	8 × 8	1
6	FC	200		
7	FC	200		

3.3.2. Generator (G)

Generator (G) serves as a decoder which recovers a corresponding 3D model by given a latent variable and model class information c. Structure of G is consistent with generator in the 3D conditional generation network described in Section 3.2. While the output size is changed to 20^3 for comparing with other methods in IKEA dataset. The change is minor and it demonstrates that our method can extend to other resolutions naturally. Loss function of G is

$$L_G = D\left(x'\right) + \lambda L_{recon}. \tag{7}$$

L_{recon} is also utilized to measure similarities between generated models and models in input images. And we set $\lambda = 10$ as recommended by 3D-IWGAN [27]. In addition, the 3D models generated by G are expected to be similar with real data as far as possible, and can be identified by C as belonging to class c. G also tries to minimize:

$$L_{G\_C} = \sum |fc\left(x\right) - fc\left(x'\right)|_2 - E_{p_z}\left[P\left(c|x'\right)\right]. \tag{8}$$

where $fc\left(x\right)$ is defined as feature maps of the last convolutional layer in the classifier network, $P\left(c|x\right)$ indicates the probability output.

3.3.3. Classifier (C)

Given a 3D model, Classifier (C) outputs the category label c of it. Inspired by VoxNet [31], the network structure takes 3D volume as input directly. In C, there are four 3D convolutional layers and two fully connected layers. Table 4 illustrates details of C. And we use no padding for convolutional layers. Minimizing the entropy loss of model category prediction is the target of C, and it can be defined as

$$L_C = -E_{p_r}\left[P\left(c|x\right)\right]. \tag{9}$$

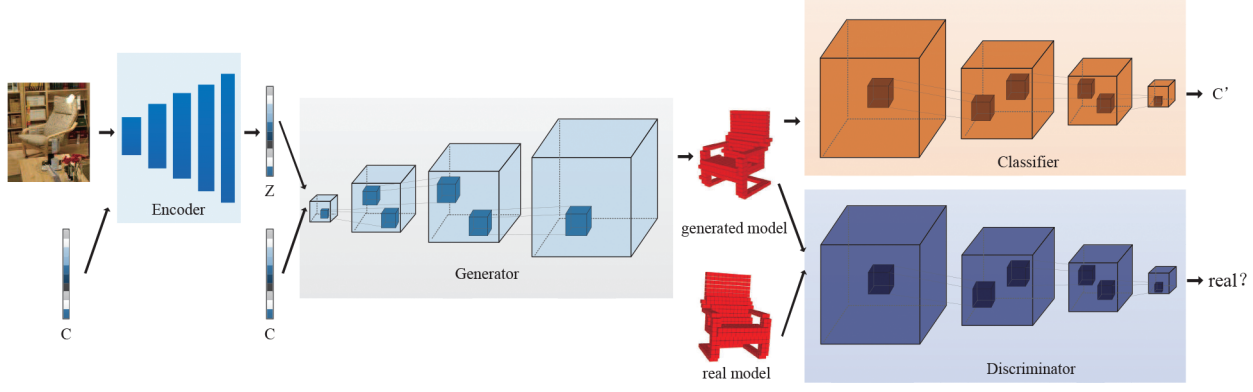

Figure 2 Three-dimensional (3D) model reconstruction framework.

Table 4 The network details of classifier.

Index	Type	Kernels	Filter Size	Strides
1	3D Conv	40	$5 \times 5 \times 5$	2
2	3D Conv	40	$3 \times 3 \times 3$	1
3	3D Conv	80	$2 \times 2 \times 2$	2
4	3D Conv	80	$2 \times 2 \times 2$	2
5	FC	80		
6	FC	10/6		

3.3.4. Discriminator (D)

Discriminator (D) is adversarially learned to discriminate between the generating models and real data. The network structure is identical to discriminator in the 3D generation network described in Section 3.2. And it is trained with a loss function which is almost the same as Equation (3). Except category labels are not input.

Overall, the 3D reconstruction network minimizes the total loss:

$$L = L_E + L_G + L_{G\_C} + L_C + L_D. \tag{10}$$

where each term is meaningful and has been described in Equations (3–9). All these objectives are related to the four networks mentioned above respectively, and complementary to each other, which enables our method to reconstruct 3D models from single images eventually. The whole training pipeline is illustrated in Algorithm 2. In our experiments, Adam is used to optimize network parameters, with the learning rate of 10^{-4}.

Algorithm 2 The training procedure of the 3D model reconstruction network

Initialize: Batch size n; number of classes m; parameter of the encoder θ_E; parameter of the generator θ_G; parameter of the discriminator θ_D; parameter of the classifier θ_C.

while algorithm is not convergence **do**

Sample $(x_1, i_1, c_1), ..., (x_n, i_n, c_n)$ from the distribution of real data p_r;

$z \leftarrow E(i, c)$

$L_{KL} \leftarrow D_{KL}\left(p_e\left(z|i,c\right)|p_z\right)$

$x' \leftarrow G(z)$

$L_{recon} \leftarrow \|x' - x\|_2$

$L_E \leftarrow L_{KL} + L_{recon}$

$L_G \leftarrow D\left(x'\right) + \lambda L_{recon}$

Extract feature map $fc(x)$ and $fc\left(x'\right)$ of x and x' in the classifier network;

$L_{G\_C} \leftarrow \sum_{c_i} |fc(x) - fc\left(x'\right)|^2 - E_c\left[P\left(c_i|x'\right)\right]$

$L_C \leftarrow -E_c\left[P\left(c_i|x\right)\right]$

Sample $\hat{x}$ from straight line between p_g and p_r;

$L_D \leftarrow -D\left(x|c\right) + D\left(x'|c\right) + \alpha\left(\|\nabla D(\hat{x})\|_2 - 1\right)^2\right)$

$\theta_C \leftarrow -\nabla_{\theta_C}(L_C)$

$\theta_D \leftarrow -\nabla_{\theta_D}(L_D)$

$\theta_E \leftarrow -\nabla_{\theta_E}(L_E)$

if iteration % 5 == 0 **then**

$\theta_G \leftarrow -\nabla_{\theta_G}\left(L_G + L_{G\_C}\right)$

end while

4. EXPERIMENTS

4.1. 3D Model Generation

- **Dataset**
  ModelNet10 dataset, a subset of ModelNet dataset [11], is used to train the generation network. ModelNet10 contains 4899 3D models from 10 classes. All models are in 32^3 resolution and rotated in 12 evenly orientations. The network is trained jointly in all classes of ModelNet10 dataset, and category labels are regarded as the condition information.

- **Results**
  We show 3D models generated by proposed method in Figure 3. The first and sixth column is the input condition information (labels), and the followed columns are corresponding 3D models generated. All models are rotated to the appropriate direction for better observation. Visual results verify that our method has the power to effectively generate 3D models based on given conditions, and maintain a diversity of the same class. After the training is completed, given a category label c, the network can automatically generate realistic and varied 3D models in an infinite amount, and without reference models and images.

- **Comparison**
  Both 3D-GAN [26] and 3D-IWGAN [27] can generate 3D models from randomly sampled noise by learning data characteristics in the dataset. However, due to the lack of controllability, what kind of model will be generated is unknown, which cause the result is not ideal when training in a full dataset of multiple categories. We introduce condition information to guide the generation of models. Generator not only generates models similar to real data, but also needs to satisfy the given condition, which is class information in this paper. This indicates that 3D conditional generation network can learn more complex data distribution and generate models of multiple categories more stable with higher quality.

4.2. 3D Object Reconstruction

- **Dataset**
  A synthetic dataset is constructed in this paper. The models that contain 10 classes (bathtub, bed, boat, bookcase, car, chair, monitor, plane, sofa, table) are selected from ShapeNet dataset [32]. Random images of the Internet are used as the background. Models are placed in a random pose, illumination, and distance and paste the random texture of the texture dataset [33]. Then, a set of views are obtained by rendering. And each image has a corresponding 3D model in ShapeNet as the ground truth. The IKEA dataset [12], which consists of 759 images related to models in six categories (bed, bookcase, chair, desk, sofa, table), is also used to test our method and compare with other methods. The network is jointly trained across all classes on both the synthetic dataset and the IKEA dataset.

- **Results**
  We show 3D object reconstruction results of the synthetic dataset and the IKEA dataset in Figures 4 and 5, respectively. These results indicate that 3D models can be reconstructed successfully from input images by using our method. And it makes reasonable reconstruction even in the unseen parts.

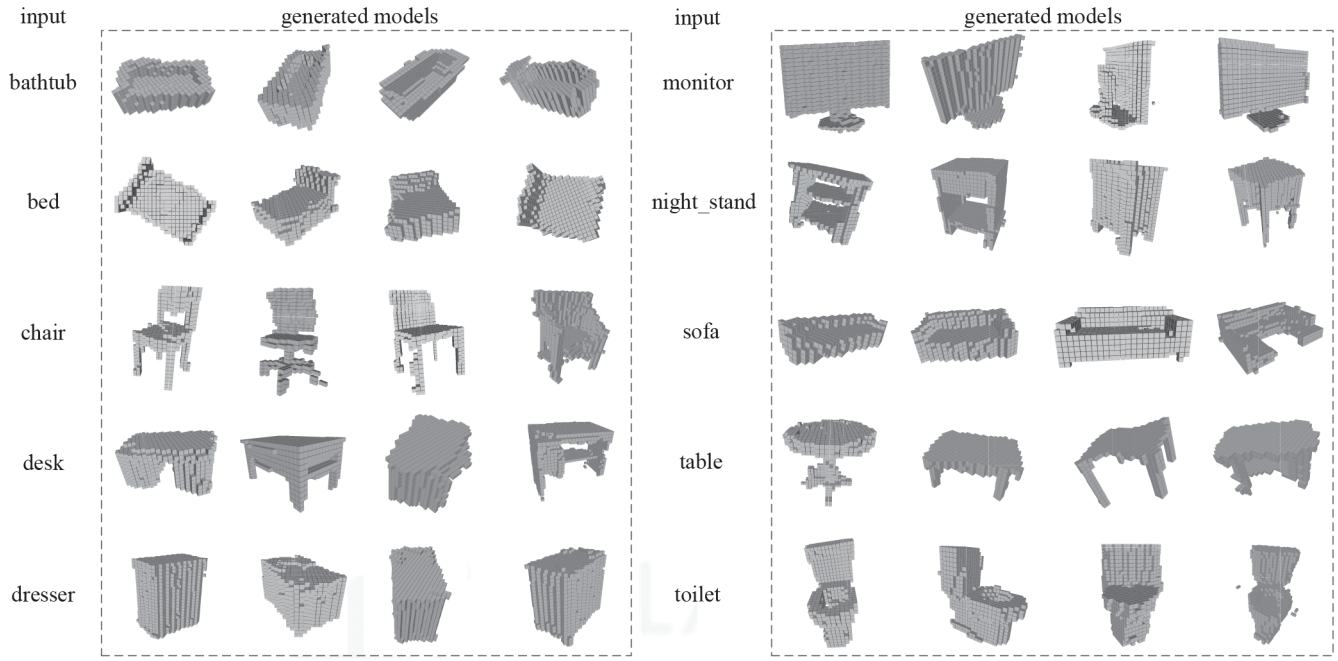

Figure 3 | The three-dimensional (3D) model generated by 3D conditional generative adversarial network (GAN).

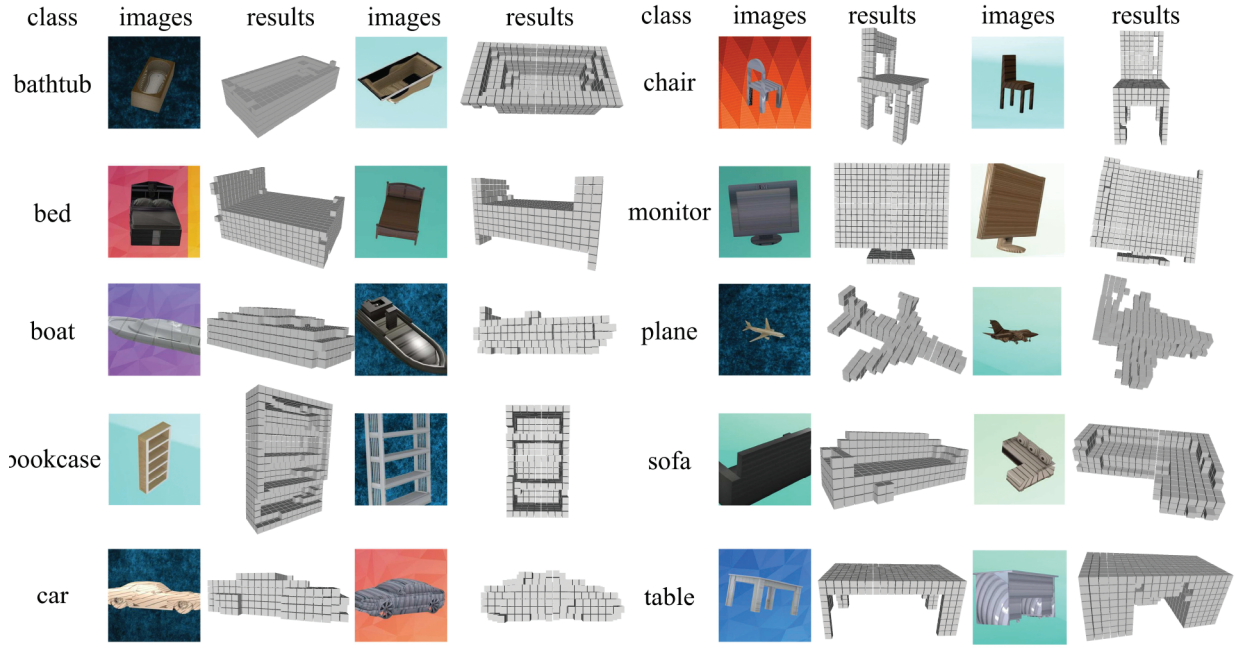

Figure 4 | The reconstruction results on the synthetic dataset.

- **Comparison**
  Addressing 3D reconstruction in the IKEA dataset is difficult, for images of the dataset are captured from realistic scenes, which means many images are heavily occluded. While Figure 5 shows our superior results. And Table 5 provides the comparative quantification of our method and other state-of-the-art methods in the IKEA dataset. As shown in these results, proposed method achieves markedly better improvement than previous methods, with a mean average precision of 70.9 across all classes in jointly trained, a 9.2% increases.

- **Discussion**
  For verifying the impact of condition and classifier, we do "exclude Condition" and "exclude Classifier" experiments for the ablation study. We keep same parameter settings and training epoches, for example, we use the same settings with Section 3.3 and only delete input condition of the generator G and the encoder E in "exclude Condition" experiment. Experimental results are presented in Figure 5 (second and third columns from right). As we can see, without the classifier, G can generate corresponding models under the guidance of

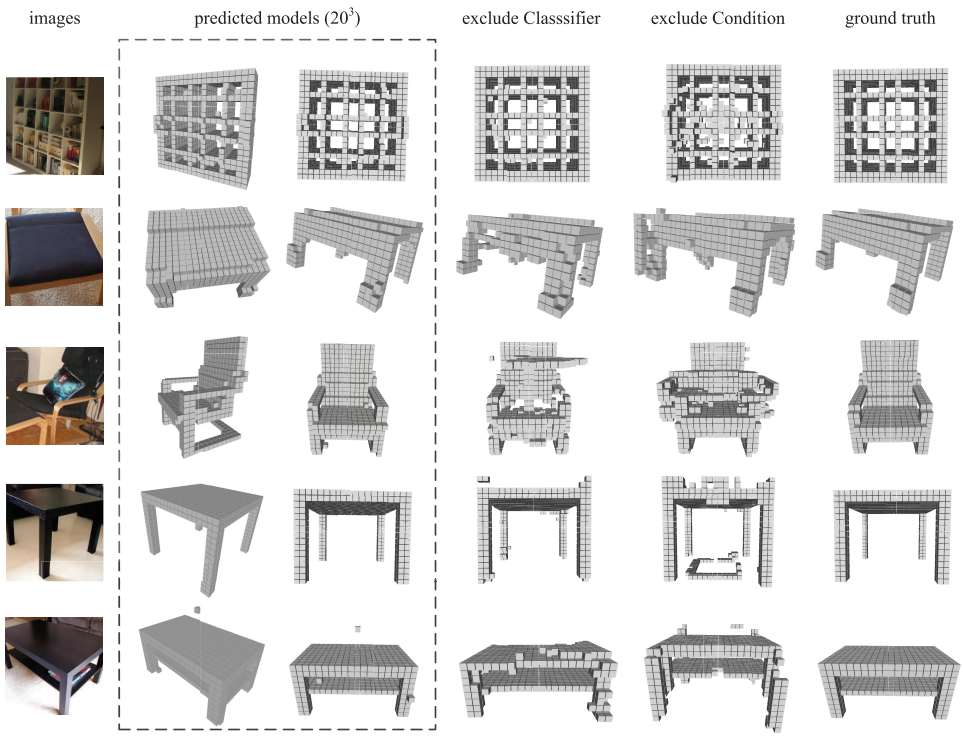

Figure 5 The reconstruction results on the IKEA dataset.

Table 5 The comparison of average precision on the IKEA dataset.

Method	Bed	Bookcase	Chair	Desk	Sofa	Table	Mean
AlexNet-fc8 [34]	29.5	17.3	20.4	19.7	38.8	16.0	23.6
AlexNet-conv4 [34]	38.2	26.6	31.4	26.6	69.3	19.1	35.2
T-L Network [34]	56.3	30.2	32.9	25.8	71.7	23.3	40.0
3D-VAE-GAN (jointly trained) [26]	49.1	31.9	42.6	34.8	79.8	33.1	41.2
3D-VAE-GAN (separately trained) [26]	63.2	46.3	47.2	40.7	78.8	42.3	53.1
3D-VAE-IWGAN (jointly trained) [27]	65.7	44.2	49.3	50.6	68.0	52.2	55.0
3D-VAE-IWGAN (separately trained) [27]	77.7	51.8	56.2	49.8	**82.0**	52.6	61.7
Ours	**86.8**	**85.2**	**60.3**	**52.4**	80.2	**60.1**	**70.9**

condition, but there may have more noise point. While E and G do not have explicit class information in "exclude Condition" experiment, the generated models may have a large offset from required, as shown in third row of Figure 5, generated chair has parts from sofa, table in forth row has parts of chair. We therefore believe that condition and classifier are both necessary. Moreover, to prove that our method can be extended to other resolutions and generate higher quality models. We also provide our reconstruction results at resolution 64^3, as shown in Figure 6. And we only change the output size of G compares with Section 3.3.

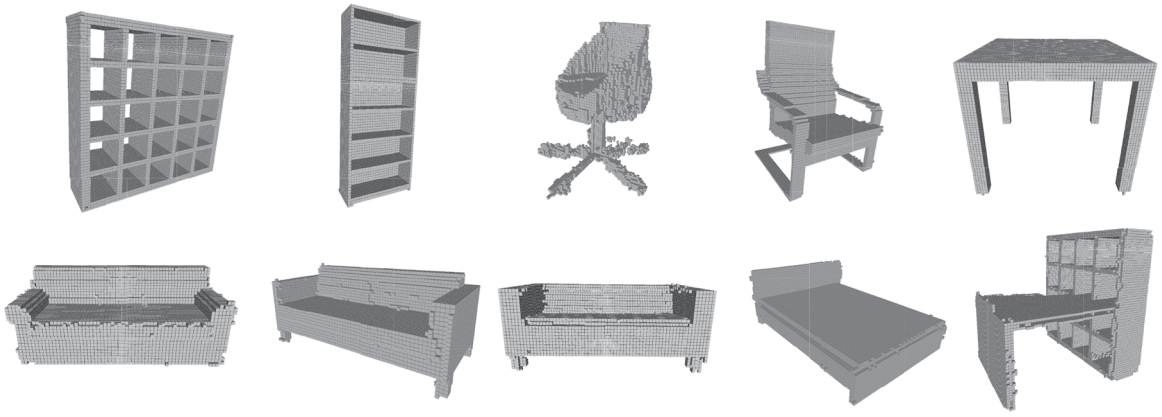

Figure 6 Our reconstruction results at resolution 64^3.

5. CONCLUSION

A 3D conditional GAN which employs class information to generate 3D models from the given class labels is proposed in this paper. The network can learn complex data distribution of multiple categories which is hard to train by using traditional GAN, and the diversity of the generator is well ensured. Moreover, a 3D reconstruction network that is produced by combining this new generation system with CVAE and volume classification network is further introduced. They promote each other for reconstructing 3D model from a single image. Experimental results in ModelNet10 dataset demonstrate that our proposed 3D conditional GAN can generate realistic and novel models effectively. And our method successfully reconstructs high-quality 3D models in jointly trained on IKEA dataset and achieves a higher mean average precision, outperforming state-of-the-art methods which trained separately on classes.

Future work 3D convolutional networks require more GPU memory than 2D convolutional networks, and the computational speed is slow. Due to computational power and time constraints, our results are limited to low resolution. Our future work will focus on how to adapt the super-resolution technology in images to achieve 3D models super-resolution or end-to-end generating 3D models in high-resolution.

ACKNOWLEDGMENTS

This work is partially supported by National Natural Science Foundation of China (No. 61877002, No. 61602015), High-level Teachers in Beijing Municipal Universities in the Period of 13th Plan CIT&TCD201904036, Beijing Municipal Commission of Education PXM2019_014213_000007 and Science and Technology Development Program of Beijing Municipal Education Commission KM201910011012.

REFERENCES

[1] C.B. Choy, D. Xu, J.Y. Gwak, K. Chen, S. Savarese, 3d-r2n2: a unified approach for single and multi-view 3d object reconstruction, in Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, Netherlands, 2016, pp. 628–644.

[2] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in Advances in Neural Information Processing Systems (NIPS), 2012, Lake Tahoe, Nevada, pp. 1097–1105.

[3] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, F.F. Li, Imagenet: a large-scale hierarchical image database, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, 2009.

[4] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, Cambridge, 2016. http://www.deeplearningbook.org

[5] A. Ioannidou, E. Chatzilari, S. Nikolopoulos, Y. Kompatsiaris, Deep learning advances in computer vision with 3d data: a survey, ACM Comput. Surv. 50 (2017), 1–20.

[6] J. Zhao, M. Mathieu, Y. LeCun, Energy-based generative adversarial network, arXiv: abs/1609.03126, 2016.

[7] X. Mao, Q. Li, H. Xie, R.Y.K.Lau, Z. Wang, S.P. Smolley, Least squares generative adversarial networks, in Proceedings of the International Conference on Computer Vision (ICCV), Venice, 2017, pp. 2794–2802.

[8] P. Isola, J. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, 2017, pp. 5967–5976.

[9] M. Mirza, S. Osindero, Conditional generative adversarial nets, arXiv preprint arXiv: 1411.1784, 2014.

[10] K. Sohn, H. Lee, X. Yan, Learning structured output representation using deep conditional generative models, in Advances in Neural Information Processing Systems (NIPS), Montreal, Canada, 2015, pp. 3483–3491, acmid: 2969628.

[11] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3d shapenets: a deep representation for volumetric shape, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, Massachusetts, 2015, pp. 1912–1920.

[12] J.J. Lim, H. Pirsiavash, A. Torralba, Parsing ikea objects: fine pose estimation, in Proceedings of the International Conference on Computer Vision (ICCV), Sydney, 2013, pp. 2992–2999.

[13] H. Li, L. Sun, S. Dong, X. Zhu, Q. Cai, J. Du, Efficient 3d object retrieval based on compact views and hamming embedding, IEEE Access. 50 (2018), 31854–31861.

[14] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, second ed., Cambridge, 2004.

[15] J.L. Schonberger, J.M. Frahm, Structure-from-motion revisited, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016, pp. 4104–4113.

[16] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, J.J. Leonard, Past, present, and future of simultaneous localization and mapping: toward the robust-perception age, IEEE Trans. Robot. 32 (2016), 1309–1332.

[17] A. Kar, S. Tulsiani, J. Carreira, J. Malik, Category-specific object reconstruction from a single image, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, Massachusetts, 2015, pp. 1966–1974.

[18] H. Fan, H. Su, L.J. Guibas, A point set generation network for 3d object reconstruction from a single image, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, 2017, pp. 605–613.

[19] J. Wu, Y. Wang, T. Xue, X. Sun, B. Freeman, J. Tenenbaum, Marrnet: 3d shape reconstruction via 2.5 d sketches, in Advances in Neural Information Processing Systems (NIPS), Long Beach, California, 2017, pp. 540–550, acmid: 3294823.

[20] H. Kato, Y. Ushiku, T. Harada, Neural 3d mesh renderer, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, 2018, pp. 3907–3916.

[21] E. Smith, S. Fujimoto, D. Meger, Multi-view silhouette and depth decomposition for high resolution 3d object representation, in Advances in Neural Information Processing Systems (NIPS), Montreal, Canada, 2018, pp. 6479–6489.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in Advances in Neural Information Processing Systems (NIPS), Montreal, Canada, 2014, pp. 2672–2680.

[23] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv: 1511.06434, 2015.

[24] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein gan, arXiv preprint arXiv: 1701.07875, 2017.

[25] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A.C. Courville, Improved training of wasserstein gans, in Advances in Neural Information Processing Systems (NIPS), Long Beach, California, 2017,pp. 5767–5777, acmid: 3295327.

[26] J. Wu, C. Zhang, T. Xue, B. Freeman, J. Tenenbaum, Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling, in Advances in Neural Information Processing Systems (NIPS), Barcelona, Spain, 2016, pp. 82–90, acmid: 3157106.

[27] E. Smith, D. Meger, Improved adversarial systems for 3d object generation and reconstruction, in Conference on Robot Learning (CoRL), Mountain View, California, 2017, pp. 87–96.

[28] W. Wang, Q. Huang, S. You, C. Yang, U. Neumann, Shape inpainting using 3d generative adversarial network and recurrent convolutional networks, in Proceedings of the International Conference on Computer Vision (ICCV), Venice, 2017, pp. 2317–2325.

[29] K. Chen, C.B. Choy, M. Savva, A.X. Chang, T. Funkhouser, S. Savarese, Text2shape: generating shapes from natural language by learning joint embeddings, in Proceedings of Asian Conference on Computer Vision (ACCV), Cham, 2018, pp: 100–116.

[30] J. Bao, D. Chen, F. Wen, H. Li, G. Hua, Cvae-gan: fine-grained image generation through asymmetric training, in Proceedings of the International Conference on Computer Vision (ICCV), Venice, 2017, pp. 2745–2754.

[31] D. Maturana, S. Scherer, Voxnet: a 3d convolutional neural network for real-time object recognition, in Proceedings of Intelligent Robots and Systems (IROS), Hamburg, 2015, pp. 922–928.

[32] A.X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, *et al.* Shapenet: an information-rich 3d model repository, arXiv preprint arXiv: 1512.03012, 2015.

[33] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, A. Vedaldi, Describing textures in the wild, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, 2014, pp. 3606–3613.

[34] R. Girdhar, D.F. Fouhey, M. Rodriguez, A. Gupta, Learning a predictable and generative vector representation for objects, in Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, Netherlands, 2016, pp. 484–499.