Generative Adversarial Modeling of 3D Shapes

by

Chengkai Zhang

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2018

Author . . . .
Department of Electrical Engineering and Computer Science
May 25, 2018

Certified by . . . .
Prof. Joshua B. Tenenbaum
Professor of Computational Cognitive Science
Thesis Supervisor

Accepted by . . . .
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Generative Adversarial Modeling of 3D Shapes

by

Chengkai Zhang

Submitted to the Department of Electrical Engineering and Computer Science
on May 25, 2018, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Given a 3D shape, humans are capable of telling whether it looks natural. This shape priors, namely the perception of whether a shape looks realistic, are formed over years of our interactions with surrounding 3D objects, and go beyond simple definition of objects. In this thesis, we propose two models, 3D Generative Adversarial Network and ShapeHD, to learn shape priors from existing 3D shapes via generative-adversarial modeling, pushing the limits of shape generation, single-view shape completion and reconstruction. For shape generation, we demonstrate that our 3D-GAN generates high-quality 3D objects, and our unsupervisedly learned features achieve impressive performance on 3D object recognition, comparable with those of supervised learning methods; for single-view shape completion and reconstruction, we show that ShapeHD recovers fine details for 3D shapes, and outperforms state-of-the-art by a large margin on both tasks.

Thesis Supervisor: Prof. Joshua B. Tenenbaum
Title: Professor of Computational Cognitive Science

To my parents

Acknowledgments

First, I would like to express my gratitude to my advisers, Prof. Joshua Tenenbaum and Prof. William Freeman, for their guidance and support over the past two and a half years.

I would like to thank Jiajun Wu, for leading me into the world of computer vision and machine learning, for the inspiring thoughts during discussion, and for the tremendous help and support over the years of our collaboration.

The thesis would not have been possible without the support of my collaborators, and I would like to express my appreciation to them, Dr. Tianfan Xue, Xiuming Zhang, Zhoutong Zhang. I am also thankful for my other collaborators, Dr. Jun-yan Zhu, Renqiao Zhang and Xingyuan Sun. It has been a pleasure working with you.

I would like to extend my appreciation to all my dear friends, for their help and support that led me through my hardest times at MIT.

I would like to thank MIT for the wonderful four years' experience and the great opportunities to work with all the excellent people I have met.

I would like to thank the freezing weather of Boston, that has kept me indoors most of the time and greatly accelerated the progress of my research.

Finally, I'm grateful to my parents, for their unconditional love and support. I would never have become the person I am today without you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Given a 3D shape, humans are capable of telling whether it looks natural. In addition, humans can learn to recognize novel objects with only very limited examples [60]. As an example, an object created by putting together the legs and seat from chairs of different styles fits every definition of a chair, but is highly unlikely to considered realistic by humans. This shape priors, *i.e.* the knowledge about realistic 3D shapes, are developed over years of our interactions with surrounding objects, and go beyond simple definition of objects.

In this thesis, we study how to extract shape priors from existing 3D shapes, and apply it to the tasks of 3D shape generation, completion and reconstruction. We discuss these topics in details over the following chapters.

In Chapter 2 we study the problem of 3D object generation. We propose a novel framework, namely 3D Generative Adversarial Network (3D-GAN), which generates 3D objects from a probabilistic space by leveraging recent advances in volumetric convolutional networks and generative adversarial nets. The benefits of our model are three-fold: first, the use of an adversarial criterion, instead of traditional heuristic criteria, enables the generator to capture object structure implicitly and to synthesize high-quality 3D objects; second, the generator establishes a mapping from a low-dimensional probabilistic space to the space of 3D objects, so that we can sample objects without a reference image or CAD models, and explore the 3D object manifold; third, the adversarial discriminator provides a powerful 3D shape descriptor which, learned without supervision, has wide applications in 3D object recognition. Experiments demonstrate that our method generates high-quality 3D objects, and our unsupervisedly learned features achieve impressive performance on 3D object recognition, comparable with those of supervised learning methods.

In Chapter 3, we study the problem of 3D shape completion and reconstruction. This problem is quite challenging, because among the many possible shapes that explain an observation, most are implausible and do not correspond to natural objects. Recent research in the field has tackled this problem by exploiting the expressiveness of deep convolutional networks. In fact, there is another level of ambiguity that is often overlooked: among plausible shapes, there are still multiple shapes that fit the 2D image equally well; *i.e.*, the ground truth shape is non-deterministic given an input. Existing fully supervised approaches fail to address this issue, and often produce blurry mean shapes with smooth surfaces but no fine details.

Therefore we propose *ShapeHD*, pushing the limit of single-view shape completion and reconstruction by integrating deep generative models with adversarially learned shape priors. The learned priors serve as a regularizer, penalizing the model only if its output is unrealistic, not if it deviates from the ground truth. Our design thus overcomes both levels of ambiguity aforementioned. Experiments demonstrate that ShapeHD outperforms state-of-the-art by a large margin on both shape completion and shape reconstruction.

Chapter 2

Learning a Probabilistic Latent Space of 3D Shapes

2.1 Introduction

What makes a 3D generative model of object shapes appealing? We believe a good generative model should be able to synthesize 3D objects that are both highly varied and realistic. Specifically, for 3D objects to have variations, a generative model should be able to go beyond memorizing and recombining parts or pieces from a pre-defined repository to produce novel shapes; and for objects to be realistic, there need to be fine details in the generated examples.

In the past decades, researchers have made impressive progress on 3D object modeling and synthesis [6, 73, 80], mostly based on meshes or skeletons. Many of these traditional methods synthesize new objects by borrowing parts from objects in existing CAD model libraries. Therefore, the synthesized objects look realistic, but not conceptually novel.

Recently, with the advances in deep representation learning and the introduction of large 3D CAD datasets like ShapeNet [8, 87], there have been some inspiring attempts in learning deep object representations based on voxelized objects [20, 55, 69]. Different from part-based methods, many of these generative approaches do not explicitly model the concept of parts or retrieve them from an object repository; instead, they synthesize new objects based on learned object representations. This is a challenging problem because, compared to the space of 2D images, it is more difficult to model the space of 3D shapes due to its higher dimensionality. Their current results are encouraging, but often there still exist artifacts (*e.g.*, fragments or holes) in the generated objects.

In this chapter, we demonstrate that modeling volumetric objects in a general-adversarial manner could be a promising solution to generate objects that are both novel and realistic. Our approach combines the merits of both general-adversarial modeling [21, 56] and volumetric convolutional networks [50, 87]. Different from traditional heuristic criteria, generative-adversarial modeling introduces an adversarial discriminator to classify whether an object is synthesized or real. This could be a particularly favorable framework for 3D object modeling: as 3D objects are highly structured, a generative-adversarial criterion, but not a voxel-wise independent heuristic one, has the potential to capture the structural difference of two 3D objects. The use of a generative-adversarial loss may also avoid possible criterion-dependent overfitting (*e.g.*, generating mean-shape-like blurred objects when minimizing a mean squared error).

Modeling 3D objects in a generative-adversarial way offers additional distinctive advantages. First, it becomes possible to sample novel 3D objects from a probabilistic latent space such as a Gaussian or uniform distribution. Second, the discriminator in the generative-adversarial approach carries informative features for 3D object recognition, as demonstrated in experiments (Section 2.4). From a different perspective, instead of learning a single feature representation for both generating and recognizing objects [20, 62], our framework learns disentangled generative and discriminative representations for 3D objects without supervision, and applies them on generation and recognition tasks, respectively.

We show that our generative representation can be used to synthesize high-quality realistic objects, and our discriminative representation can be used for 3D object recognition, achieving comparable performance with recent supervised methods [50, 63], and outperforming other unsupervised methods by a large margin. The learned generative and discriminative representations also have wide applications. For example, we show that our network can be combined with a variational autoencoder [42, 43] to directly reconstruct a 3D object from a 2D input image. Further, we explore the space of object representations and demonstrate that both our generative and discriminative representations carry rich semantic information about 3D objects.

2.2 Related Work

Modeling and synthesizing 3D shapes 3D object understanding and generation is an important problem in the graphics and vision community, and the relevant literature is very rich [1, 4, 6, 9, 35, 36, 73, 80, 84, 93]. Since decades ago, AI and vision researchers have made inspiring attempts to design or learn 3D object representations, mostly based on meshes and skeletons. Many of these shape synthesis algorithms are nonparametric and they synthesize new objects by retrieving and combining shapes and parts from a database. Recently, Huang et al. [28] explored generating 3D shapes with pre-trained templates and producing both object structure and surface geometry. Our framework synthesizes objects without explicitly borrow parts from a repository, and requires no supervision during training.

Deep learning for 3D data The vision community have witnessed rapid development of deep networks for various tasks. In the field of 3D object recognition, Girdhar et al. [20], Li et al. [47], Su et al. [70] proposed to learn a joint embedding of 3D shapes and synthesized images, Qi et al. [55], Su et al. [69] focused on learning discriminative representations for 3D object recognition, Choy et al. [12], Wu et al. [84], Xiang et al. [89] discussed 3D object reconstruction from in-the-wild images, possibly with a recurrent network, and Girdhar et al. [20], Sharma et al. [62] explored autoencoder-based networks for learning voxel-based object representations. Rezende et al. [57], Wu et al. [87], Yan et al. [95] attempted to generate 3D objects with deep networks, some using 2D images during training with a 3D to 2D projection layer. Many of these networks can be used for 3D shape classification [50, 62, 69], 3D shape retrieval [63, 69], and single image 3D reconstruction [1, 20, 36], mostly with full supervision. In comparison, our framework requires no supervision for training, is able to generate objects from a probabilistic space, and comes with a rich discriminative 3D shape representation.

Learning with an adversarial net Generative Adversarial Nets (GAN) [21] proposed to incorporate an adversarial discriminator into the procedure of generative modeling. More recently, LAPGAN [14] and DC-GAN [56] adopted GAN with convolutional networks for

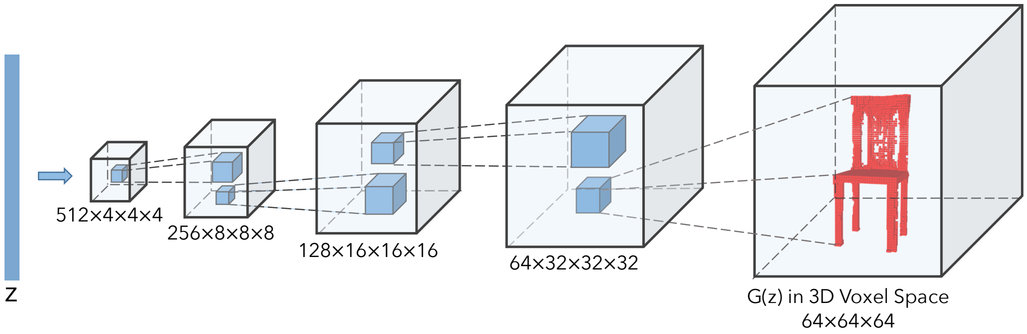


Figure 2-1: The generator in 3D-GAN. The discriminator mostly mirrors the generator.

image synthesis, and achieved impressive performance. Researchers have also explored the use of GAN for other vision problems. To name a few, Wang and Gupta [81] discussed how to model image style and structure with sequential GANs, Li and Wand [45] and Zhu et al. [98] used GAN for texture synthesis and image editing, respectively, and Im et al. [29] developed a recurrent adversarial network for image generation. While previous approaches focus on modeling 2D images, we discuss the use of an adversarial component in modeling 3D objects.

2.3 Models

In this section we introduce our model for 3D object generation. We first discuss how we build our framework, 3D Generative Adversarial Network (3D-GAN), by leveraging previous advances on volumetric convolutional networks and generative adversarial nets. We then show how to train a variational autoencoder [42] simultaneously so that our framework can capture a mapping from a 2D image to a 3D object.

2.3.1 3D Generative Adversarial Network (3D-GAN)

As proposed by Goodfellow et al. [21], the Generative Adversarial Network (GAN) consists of a generator and a discriminator, where the discriminator tries to classify real objects and objects synthesized by the generator, and the generator attempts to confuse the discriminator. In our 3D Generative Adversarial Network (3D-GAN), the generator G maps a 200-dimensional latent vector z, randomly sampled from a probabilistic latent space, to a $64 \times 64 \times 64$ cube, representing an object $G(z)$ in 3D voxel space. The discriminator D outputs a confidence value $D(x)$ of whether a 3D object input x is real or synthetic.

Following Goodfellow et al. [21], we use binary cross entropy as the classification loss, and present our overall adversarial loss function as

$$L_{\text{3D-GAN}} = \log D(x) + \log(1 - D(G(z))), \tag{2.1}$$

where x is a real object in a $64 \times 64 \times 64$ space, and z is a randomly sampled noise vector from a distribution $p(z)$. In this work, each dimension of z is an i.i.d. uniform distribution over $[0, 1]$.

Network structure Inspired by Radford et al. [56], we design an all-convolutional neural network to generate 3D objects. As shown in Figure 2-1, the generator consists of five volumetric fully convolutional layers of kernel sizes $4 \times 4 \times 4$ and strides 2, with batch normalization and ReLU layers added in between and a Sigmoid layer at the end. The discriminator basically mirrors the generator, except that it uses Leaky ReLU [49] instead of ReLU layers. There are no pooling or linear layers in our network. More details can be found in the supplementary material.

Training details A straightforward training procedure is to update both the generator and the discriminator in every batch. However, the discriminator usually learns much faster than the generator, possibly because generating objects in a 3D voxel space is more difficult than differentiating between real and synthetic objects [21, 56]. It then becomes hard for the generator to extract signals for improvement from a discriminator that is way ahead, as all examples it generated would be correctly identified as synthetic with high confidence. Therefore, to keep the training of both networks in pace, we employ an adaptive training strategy: for each batch, the discriminator only gets updated if its accuracy in the last batch is not higher than 80%. We observe this helps to stabilize the training and to produce better results. We set the learning rate of G to 0.0025, D to 10^{-5}, and use a batch size of 100. We use ADAM [40] for optimization, with $\beta = 0.5$.

2.3.2 3D-VAE-GAN

We have discussed how to generate 3D objects by sampling a latent vector z and mapping it to the object space. In practice, it would also be helpful to infer these latent vectors from observations. For example, if there exists a mapping from a 2D image to the latent representation, we can then recover the 3D object corresponding to that 2D image.

Following this idea, we introduce 3D-VAE-GAN as an extension to 3D-GAN. We add an additional image encoder E, which takes a 2D image x as input and outputs the latent representation vector z. This is inspired by VAE-GAN proposed by [43], which combines VAE and GAN by sharing the decoder of VAE with the generator of GAN.

The 3D-VAE-GAN therefore consists of three components: an image encoder E, a decoder (the generator G in 3D-GAN), and a discriminator D. The image encoder consists of five spatial convolution layers with kernel size $\{11, 5, 5, 5, 8\}$ and strides $\{4, 2, 2, 2, 1\}$, respectively. There are batch normalization and ReLU layers in between, and a sampler at the end to sample a 200 dimensional vector used by the 3D-GAN. The structures of the generator and the discriminator are the same as those in Section 2.3.1.

Similar to VAE-GAN [43], our loss function consists of three parts: an object reconstruction loss L_{recon}, a cross entropy loss $L_{\text{3D-GAN}}$ for 3D-GAN, and a KL divergence loss L_{KL} to restrict the distribution of the output of the encoder. Formally, these loss functions write as

$$L = L_{\text{3D-GAN}} + \alpha_1 L_{\text{KL}} + \alpha_2 L_{\text{recon}}, \tag{2.2}$$

where α_1 and α_2 are weights of the KL divergence loss and the reconstruction loss. We have

$$L_{\text{3D-GAN}} = \log D(x) + \log(1 - D(G(z))), \tag{2.3}$$

$$L_{\text{KL}} = D_{\text{KL}}(q(z|y) \mid\mid p(z)), \tag{2.4}$$

$$L_{\text{recon}} = ||G(E(y)) - x||_2, \tag{2.5}$$

where x is a 3D shape from the training set, y is its corresponding 2D image, and $q(z|y)$ is the variational distribution of the latent representation z. The KL-divergence pushes this variational distribution towards to the prior distribution $p(z)$, so that the generator can sample the latent representation z from the same distribution $p(z)$. In this work, we choose $p(z)$ a multivariate Gaussian distribution with zero-mean and unit variance. For more details, please refer to Larsen et al. [43].

Training 3D-VAE-GAN requires both 2D images and their corresponding 3D models.

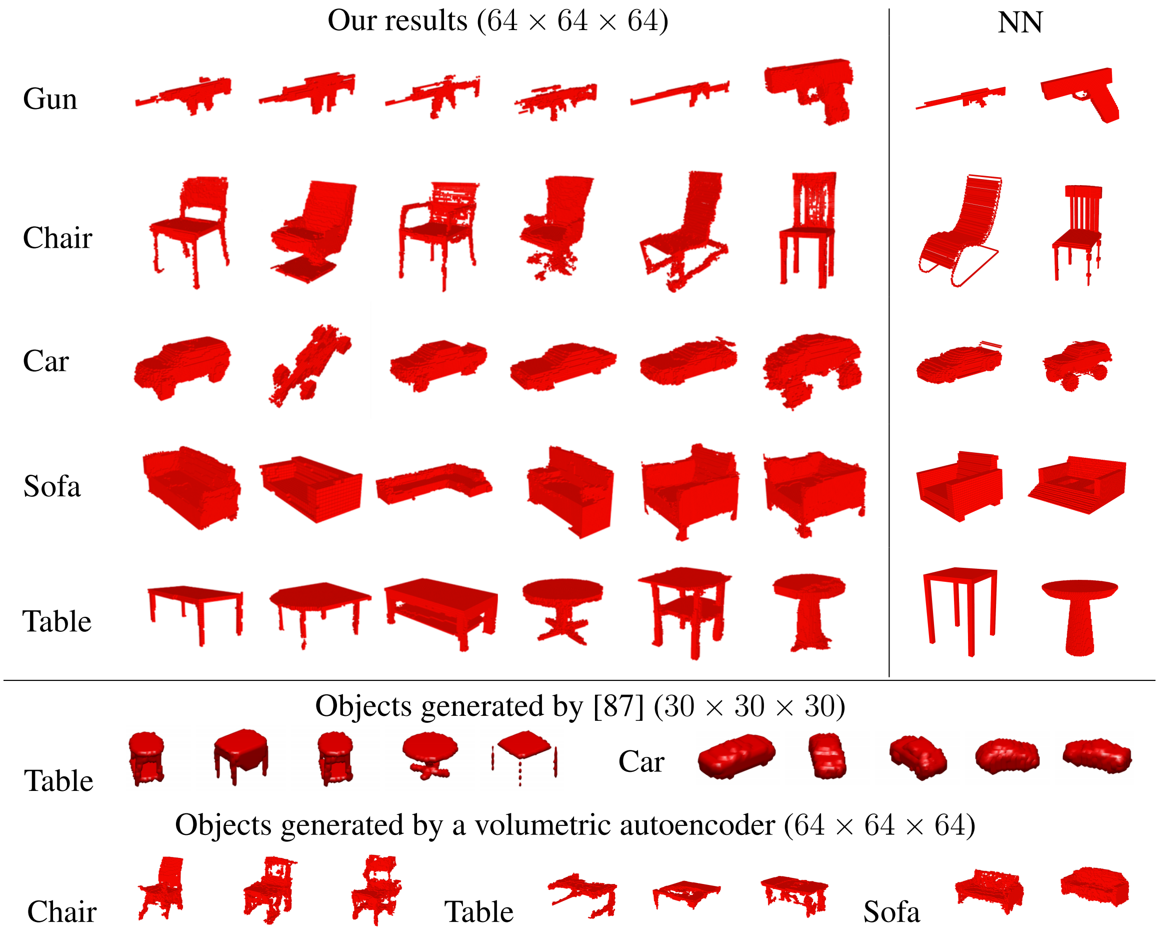

Our results (64 × 64 × 64)
NN
Gun
Chair
Car
Sofa
Table
Objects generated by [87] (30 × 30 × 30)
Table
Car
Objects generated by a volumetric autoencoder (64 × 64 × 64)
Chair
Table
Sofa

Figure 2-2: Objects generated by 3D-GAN from vectors, without a reference image/object. We show, for the last two objects in each row, the nearest neighbor retrieved from the training set. We see that the generated objects are similar, but not identical, to examples in the training set. For comparison, we show objects generated by the previous state-of-the-art [87] (results supplied by the authors). We also show objects generated by autoencoders trained on a single object category, with latent vectors sampled from empirical distribution. See text for details.

We render 3D shapes in front of background images ($16,913$ indoor images from the SUN database [91]) in 72 views (from 24 angles and 3 elevations). We set $\alpha_1 = 5$, $\alpha_2 = 10^{-4}$, and use a similar training strategy as in Section 2.3.1. See our supplementary material for more details.

2.4 Evaluation

In this section, we evaluate our framework from various aspects. We first show qualitative results of generated 3D objects. We then evaluate the unsupervisedly learned representation

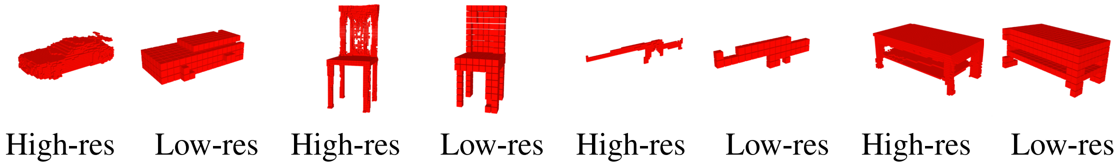


Figure 2-3: We present each object at high resolution ($64 \times 64 \times 64$) on the left and at low resolution (down-sampled to $16 \times 16 \times 16$) on the right. While humans can perceive object structure at a relatively low resolution, fine details and variations only appear in high-res objects.

from the discriminator by using them as features for 3D object classification. We show both qualitative and quantitative results on the popular benchmark ModelNet [87]. Further, we evaluate our 3D-VAE-GAN on 3D object reconstruction from a single image, and show both qualitative and quantitative results on the IKEA dataset [48].

2.4.1 3D Object Generation

Figure 2-2 shows 3D objects generated by our 3D-GAN. For this experiment, we train one 3D-GAN for each object category. For generation, we sample 200-dimensional vectors following an i.i.d. uniform distribution over $[0, 1]$, and render the largest connected component of each generated object. We compare 3D-GAN with Wu et al. [87], the state-of-the-art in 3D object synthesis from a probabilistic space, and with a volumetric autoencoder, whose variants have been employed by multiple recent methods [20, 62]. Because an autoencoder does not restrict the distribution of its latent representation, we compute the empirical distribution $p_0(z)$ of the latent vector z of all training examples, fit a Gaussian distribution g_0 to p_0, and sample from g_0. Our algorithm produces 3D objects with much higher quality and more fine-grained details.

Compared with previous works, our 3D-GAN can synthesize high-resolution 3D objects with detailed geometries. Figure 2-3 shows both high-res voxels and down-sampled low-res voxels for comparison. Note that it is relatively easy to synthesize a low-res object, but is much harder to obtain a high-res one due to the rapid growth of 3D space. However, object details are only revealed in high resolution.

A natural concern to our generative model is whether it is simply memorizing objects from training data. To demonstrate that the network can generalize beyond the training set,

Supervision	Pretraining	Method	Classification (Accuracy)	
			ModelNet40	ModelNet10
Category labels	ImageNet	MVCNN [69]	90.1%	-
		MVCNN-MultiRes [55]	**91.4**%	-
	None	3D ShapeNets [87]	77.3%	83.5%
		DeepPano [63]	77.6%	85.5%
		VoxNet [50]	83.0%	92.0%
		ORION [61]	-	**93.8**%
Unsupervised	-	SPH [38]	68.2%	79.8%
		LFD [10]	75.5%	79.9%
		T-L Network [20]	74.4%	-
		VConv-DAE [62]	75.5%	80.5%
		3D-GAN (ours)	**83.3**%	**91.0**%

Table 2.1: Classification results on the ModelNet dataset. Our 3D-GAN outperforms other unsupervised learning methods by a large margin, and is comparable to some recent supervised learning frameworks.

we compare synthesized objects with their nearest neighbor in the training set. Since the retrieval objects based on ℓ^2 distance in the voxel space are visually very different from the queries, we use the output of the last convolutional layer in our discriminator (with a 2x pooling) as features for retrieval instead. Figure 2-2 shows that generated objects are similar, but not identical, to the nearest examples in the training set.

2.4.2 3D Object Classification

We then evaluate the representations learned by our discriminator. A typical way of evaluating representations learned without supervision is to use them as features for classification. To obtain features for an input 3D object, we concatenate the responses of the second, third, and fourth convolution layers in the discriminator, and apply max pooling of kernel sizes $\{8, 4, 2\}$, respectively. We use one-versus-all linear SVM classifiers for classification.

Data We train a single 3D-GAN on the seven major object categories (chairs, sofas, tables, boats, airplanes, rifles, and cars) of ShapeNet [8]. We use ModelNet [87] for testing, following Maturana and Scherer [50], Qi et al. [55], Sharma et al. [62].[*] Specifically, we

[*] For ModelNet, there are two train/test splits typically used. Maturana and Scherer [50], Qi et al. [55], Shi et al. [63] used the train/test split included in the dataset, which we also follow; Sharma et al. [62], Su et al. [69], Wu et al. [87] used 80 training points and 20 test points in each category for experiments, possibly with

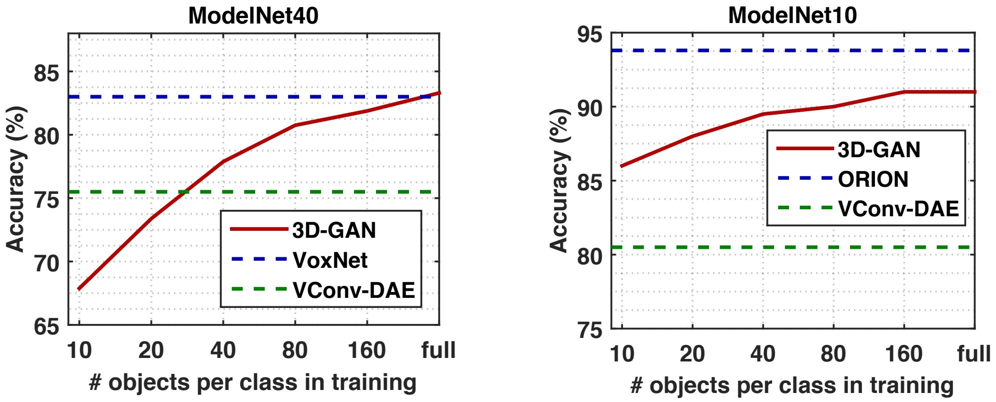


Figure 2-4: Classification accuracy with limited training data, on ModelNet40 and ModelNet10.

evaluate our model on both ModelNet10 and ModelNet40, two subsets of ModelNet that are often used as benchmarks for 3D object classification. Note that the training and test categories are not identical, which also shows the out-of-category generalization power of our 3D-GAN.

Results We compare with the state-of-the-art methods [20, 61, 62, 87] and show per-class accuracy in Table 2.1. Our representation outperforms other features learned without supervision by a large margin (83.3% vs. 75.5% on ModelNet40, and 91.0% vs 80.5% on ModelNet10) [20, 62]. Further, our classification accuracy is also higher than some recent supervised methods [63], and is close to the state-of-the-art voxel-based supervised learning approaches [50, 61]. Multi-view CNNs [55, 69] outperform us, though their methods are designed for classification, and require rendered multi-view images and an ImageNet-pretrained model.

3D-GAN also works well with limited training data. As shown in Figure 2-4, with roughly 25 training samples per class, 3D-GAN achieves comparable performance on ModelNet40 with other unsupervised learning methods trained with at least 80 samples per class.

viewpoint augmentation.

Method	Bed	Bookcase	Chair	Desk	Sofa	Table	Mean
AlexNet-fc8 [20]	29.5	17.3	20.4	19.7	38.8	16.0	23.6
AlexNet-conv4 [20]	38.2	26.6	31.4	26.6	69.3	19.1	35.2
T-L Network [20]	56.3	30.2	32.9	25.8	71.7	23.3	40.0
3D-VAE-GAN (jointly trained)	49.1	31.9	42.6	34.8	**79.8**	33.1	45.2
3D-VAE-GAN (separately trained)	**63.2**	**46.3**	**47.2**	**40.7**	78.8	**42.3**	**53.1**

Table 2.2: Average precision for voxel prediction on the IKEA dataset.†

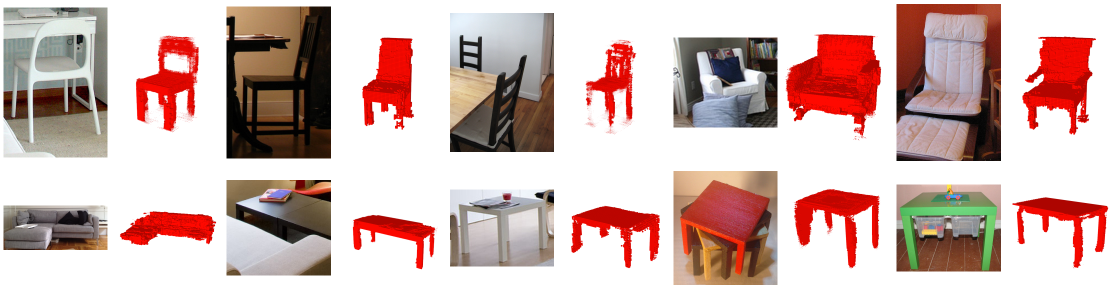


Figure 2-5: Qualitative results of single image 3D reconstruction on the IKEA dataset

2.4.3 Single Image 3D Reconstruction

As an application, our show that the 3D-VAE-GAN can perform well on single image 3D reconstruction. Following previous work [20], we test it on the IKEA dataset [48], and show both qualitative and quantitative results.

Data The IKEA dataset consists of images with IKEA objects. We crop the images so that the objects are centered in the images. Our test set consists of $1,039$ objects cropped from 759 images (supplied by the author). The IKEA dataset is challenging because all images are captured in the wild, often with heavy occlusions. We test on all six categories of objects: bed, bookcase, chair, desk, sofa, and table.

Results We show our results in Figure 2-5 and Table 2.2, with performance of a single 3D-VAE-GAN jointly trained on all six categories, as well as the results of six 3D-VAE-GANs separately trained on each class. Following Girdhar et al. [20], we evaluate results at resolution $20 \times 20 \times 20$, use the average precision as our evaluation metric, and attempt to align each prediction with the ground-truth over permutations, flips, and translational alignments (up to 10%), as IKEA ground truth objects are not in a canonical viewpoint. In all categories, our model consistently outperforms previous state-of-the-art in voxel-level

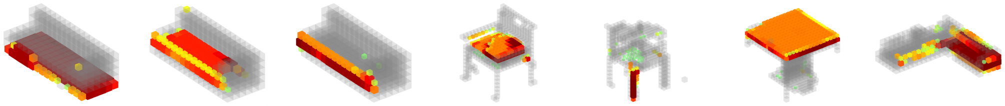


Figure 2-6: The effects of individual dimensions of the object vector.

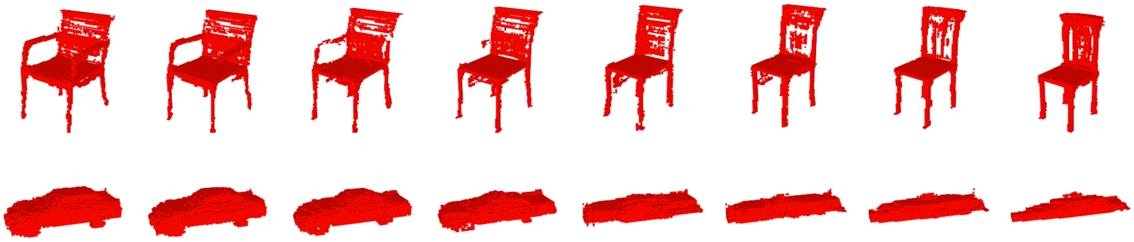


Figure 2-7: Intra/inter-class interpolation between object vectors.

prediction and other baseline methods.[†]

2.5 Analyzing Learned Representations

In this section, we look deep into the representations learned by both the generator and the discriminator of 3D-GAN. We start with the 200-dimensional object vector, from which the generator produces various objects. We then visualize neurons in the discriminator, and demonstrate that these units capture informative semantic knowledge of the objects, which justifies its good performance on object classification presented in Section 2.4.

2.5.1 The Generative Representation

We explore three methods for understanding the latent space of vectors for object generation. We first visualize what an individual dimension of the vector represents; we then explore the possibility of interpolating between two object vectors and observe how the generated objects change; last, we present how we can apply shape arithmetic in the latent space.

Visualizing the object vector To visualize the semantic meaning of each dimension, we gradually increase its value, and observe how it affects the generated 3D object. In Figure 2-6, each column corresponds to one dimension of the object vector, where the red region marks the voxels affected by changing values of that dimension. We observe that some

[†] For methods from Girdhar et al. [20], the mean values in the last column are higher than the originals in their paper, because we compute per-class accuracy instead of per-instance accuracy.

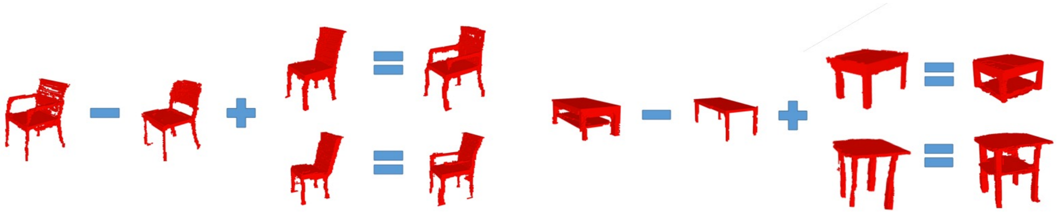


Figure 2-8: Shape arithmetic for chairs and tables. The left images show the obtained “arm” vector can be added to other chairs, and the right ones show the “layer” vector can be added to other tables.

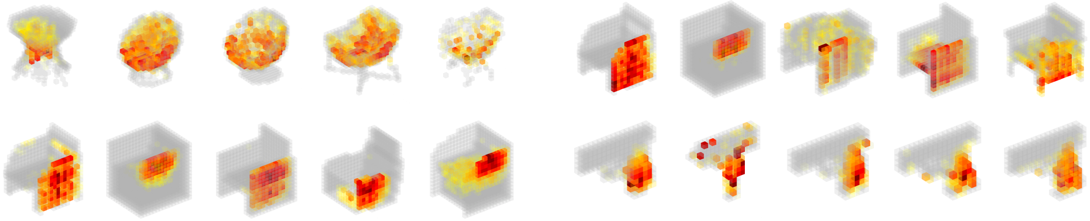


Figure 2-9: Objects and parts that activate specific neurons in the discriminator. For each neuron, we show five objects that activate it most strongly, with colors representing gradients of activations with respect to input voxels.

dimensions in the object vector carries semantic knowledge of the object, *e.g.*, the thickness or width of surfaces.

Interpolation We show results of interpolating between two object vectors in Figure 2-7. Earlier works demonstrated interpolation between two 2D images of the same category [16, 56]. Here we show interpolations both within and across object categories. We observe that for both cases walking over the latent space gives smooth transitions between objects.

Arithmetic Another way of exploring the learned representations is to show arithmetic in the latent space. Previously, Dosovitskiy et al. [16], Radford et al. [56] presented that their generative nets are able to encode semantic knowledge of chair or face images in its latent space; Girdhar et al. [20] also showed that the learned representation for 3D objects behave similarly. We show our shape arithmetic in Figure 2-8. Different from Girdhar et al. [20], all of our objects are randomly sampled, requiring no existing 3D CAD models as input.

2.5.2 The Discriminative Representation

We now visualize the neurons in the discriminator. Specifically, we would like to show what input objects, and which part of them produce the highest intensity values for each neuron.

To do that, for each neuron in the second to last convolutional layer of the discriminator, we iterate through all training objects and exhibit the ones activating the unit most strongly. We further use guided back-propagation [68] to visualize the parts that produce the activation.

Figure 2-9 shows the results. There are two main observations: first, for a single neuron, the objects producing strongest activations have very similar shapes, showing the neuron is selective in terms of the overall object shape; second, the parts that activate the neuron, shown in red, are consistent across these objects, indicating the neuron is also learning semantic knowledge about object parts.

Chapter 3

Learning Shape Priors for Single-View 3D Completion and Reconstruction

3.1 Introduction

Let's start with a game: each of the two instances in Figure 3-1 shows a depth or color image and two different 3D shape interpretations. Which one looks better?

We asked this question to 100 people on Amazon Mechanical Turk. 59% of them preferred interpretation A of the airplane, and 35% preferred interpretation A of the car. These numbers suggest that people's opinions diverge on these two cases, indicating that the quality of these reconstructions is close, and their perceptual differences are relatively minor.

Actually, for each instance, one of the reconstructions is the output of the model introduced in this chapter, and the other is the ground truth shape. Answers are available in the footnote.

In this chapter, we aim to push the limit of 3D shape completion from a single depth image, and of 3D shape reconstruction from a single color image. Recently, researchers have made impressive progress on the these tasks [12, 13, 79], making use of gigantic 3D datasets [7, 88, 90]. Many of these methods tackle the ill-posed nature of the problem by using deep convolutional networks to regress possible 3D shapes. Leveraging the power of deep generative networks, models learn to avoid producing implausible shapes (Figure 3-2b).

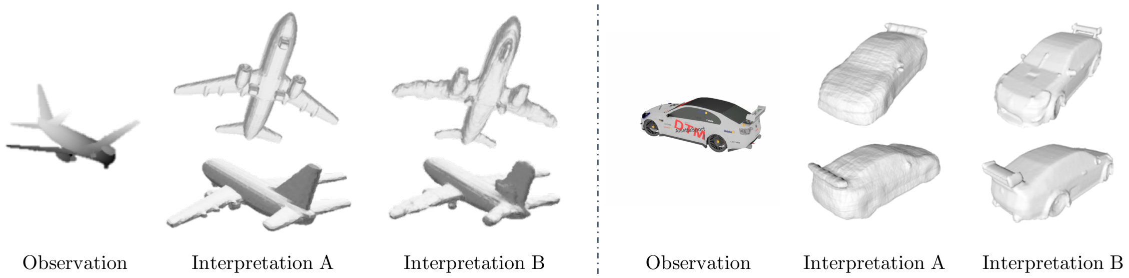


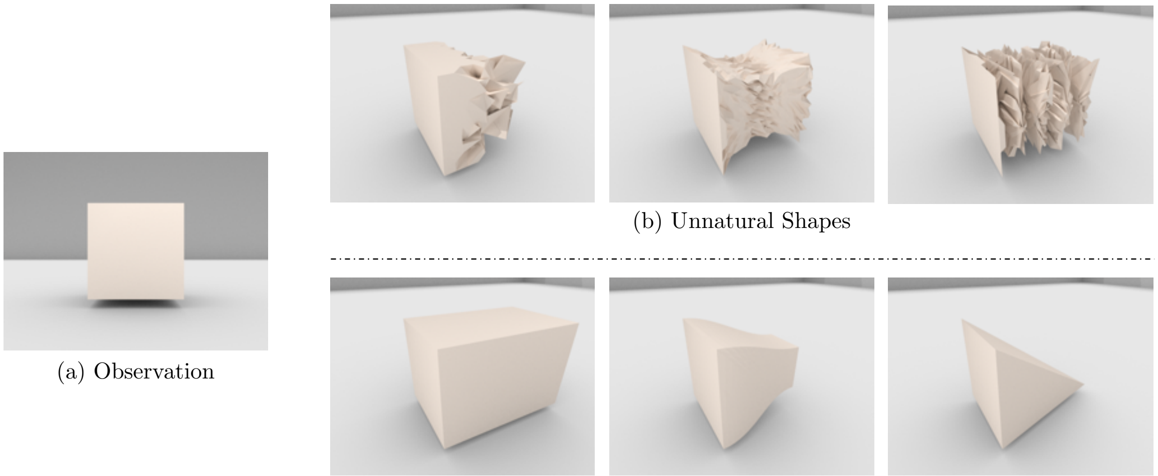
Figure 3-1: Our model completes or reconstructs the object's full 3D shape with fine details from a single depth or RGB image. In this figure, we show two examples, each consisting of an input image, two views of its ground truth shape, and two views of our results. Our reconstructions are of high quality with fine details, and are preferred by humans 41% and 35% of the time in behavioral studies, respectively. Our model takes a single feed-forward pass without any post-processing during testing, and is thus highly efficient (<100 ms) and practically useful. Answers are available in the footnote.

However, from Figure 3-2c we realize that there is still ambiguity that a supervisedly trained network fails to model. From just a single view, there exist multiple natural shapes that explain the observation equally well. In other words, there is no deterministic ground truth for each observation. Through pure supervised learning, the network tends to generate mean shapes that minimize its penalty precisely due to this ambiguity.

To tackle this, we propose ShapeHD, which completes or reconstructs a 3D shape by combining deep volumetric convolutional networks with adversarially learned shape priors. The learned shape priors penalize the model only if the generated shape is unrealistic, not if it deviates from the ground truth. This overcomes the difficulty discussed above. Our model characterizes this naturalness loss through adversarial learning, a research topic that has received immense attention in recent years and is still rapidly growing [21, 56, 85].

Experiments on multiple synthetic and real datasets suggest that ShapeHD performs well on single-view 3D shape completion, outperforming methods by a large margin in quantitative evaluations. With an additional depth estimation module, our model also does well in single-image 3D reconstruction, achieving better results than state-of-the-art reconstruction systems. Further analyses suggest that the network is learning to attend to meaningful object parts, and the naturalness module helps to characterize shape details.

Our reconstructions: B, A



(a) Observation

(b) Unnatural Shapes

(c) Natural Shapes

Figure 3-2: Two levels of ambiguity in single-view 3D shape perception. For each 2D observation (a), there exist many possible 3D shapes that explain this observation equally well (b, c), but only a small fraction of them correspond to real, daily shapes (c). Methods that exploit deep networks for recognition reduce, to a certain extent, ambiguity on this level. By using an adversarially learned naturalness model, our ShapeHD aims to model ambiguity on the next level: even among the realistic shapes, there are still multiple shapes explaining the observation well (c).

3.2 Related Work

3D shape completion. Shape completion is an essential task in geometry processing and has wide applications. Traditional methods have attempted to complete shapes with local surface primitives, or to formulate it as an optimization problem [53, 67], *e.g.*, Poisson surface reconstruction solves an indicator function on a voxel grid via the Poisson equation [37, 39]. Recently, there have also been a growing number of papers on exploiting shape structures and regularities [52, 78], and papers on leveraging strong database priors [5, 46, 72]. These methods, however, often require the database to contain exact parts of the shape, and thus have limited generalization power.

With the advances in large-scale shape repositories like ShapeNet [7], researchers began to develop fully data-driven methods, some building upon deep convolutional networks. To name a few, Voxlets [19] employs random forests for predicting unknown voxel neighborhoods. 3D ShapeNets [87] uses a deep belief network to obtain a generative model for a given shape database, and Nguyen *et al.* [77] extends the method for mesh repairing.

Probably the most related paper to ours is the 3D-EPN from Dai *et al*. [13]. 3D-EPN achieves impressive results on 3D shape completion from partial depth scans by levering 3D convolutional networks and nonparametric patch-based shape synthesis methods. Our model has advantages over 3D-EPN in two aspects. First, with naturalness losses, ShapeHD can choose among multiple hypotheses that explain the observation, therefore reconstructing a high-quality 3D shape with fine details; in contrast, the output from 3D-EPN without nonparametric shape synthesis is often blurry. Second, our completion takes a single feed-forward pass without any post-processing, and is thus much faster (<100ms) than 3D-EPN.

Single-image 3D reconstruction. The problem of recovering the object shape from a single image is challenging, as it requires both powerful recognition systems and prior shape knowledge. With the development of large-scale shape repositories like ShapeNet [7] and methods like deep convolutional networks, researchers have made significant progress in recent years [12, 20, 25, 36, 54, 57, 75, 79, 85, 86, 94]. While most of these approaches encode objects in voxels, there have also been attempts to reconstruct objects in point clouds [18, 22] or octave trees [58, 59, 76].

A related direction is to estimate 2.5D sketches (*e.g*., depth and surface normal maps) from an RGB image. In the past, researchers have explored recovering 2.5D sketches from shading, texture, or color images [2, 3, 27, 74, 83, 96]. With the development of depth sensors [31] and larger-scale RGB-D datasets [51, 65, 66], there have also been papers on estimating depth [11, 17], surface normals [1, 82], and other intrinsic images [33, 64] with deep networks. In this chapter, in addition to shape completion, we show that our model can be augmented with a depth estimation module, enabling 3D reconstruction from a single RGB image.

Perceptual losses and adversarial learning. Researchers recently proposed to evaluate the quality of 2D images using perceptual losses [15, 34]. The idea has been applied to many image tasks like style transfer and super-resolution [34, 44]. Furthermore, the idea has been extended to learn a perceptual loss function with generative adversarial nets (GAN) [21]. GANs incorporate an adversarial discriminator into the procedure of generative modeling, and achieve impressive performance on tasks like image synthesis [56]. Isola *et al*. [30] and

Zhu *et al*. [98] use GANs for image translation with and without supervision, respectively.

In 3D vision, Wu *et al*. [85] extends GANs for 3D shape synthesis. However, their model for shape reconstruction (3D-VAE-GAN) often produces a noisy, incomplete shape given an RGB image. This is because training GANs jointly with recognition networks could be highly unstable. Many other researchers have also noticed this issue: although adversarial modeling of 3D shape space may resolve the ambiguity discussed earlier, its training could be challenging [13]. Addressing this, when Gwak *et al*. [24] explored adversarial nets for single-image 3D reconstruction and chose to use GANs to model 2D projections instead of 3D shapes. This weakly supervised setting, however, hampers their reconstructions. In this chapter, we develop our naturalness loss by adversarial modeling of the 3D shape space, outperforming the state-of-the-art significantly.

3.3 Approach

Our model for single-view 3D shape reconstruction consists of three components: first, a 2.5D sketch estimator that predicts the object's depth, surface normal, and silhouette from an RGB image (Figure 3-3-I); second, a 3D shape estimator that predicts a 3D shape from an object's 2.5D sketches (Figure 3-3-II); third, a deep naturalness model that penalizes the shape estimator if the predicted shape is unnatural (Figure 3-3-III). Models trained with a supervised reconstruction loss alone often generate blurry mean shapes. Our learned naturalness model helps to avoid this issue.

2.5D sketch estimation network. The first component of our model is a 2.5D sketch estimator with an encoder-decoder structure to predict the object's depth, surface normals, and silhouette from an RGB image (Figure 3-3-I). We use a ResNet-18 [26] to encode a RGB image of resolution 256×256 into 512 feature maps of size 8×8. The decoder consists of four transposed convolutional layers with a kernel size of 5×5 and a stride and padding of 2. The predicted depth and surface normal images are then masked by the predicted silhouette and used as the input to our shape completion network.

3D shape completion network. The second component of our model (Figure 3-3-II) is an encoder-decoder network that predicts a 3D shape in canonical view from single-

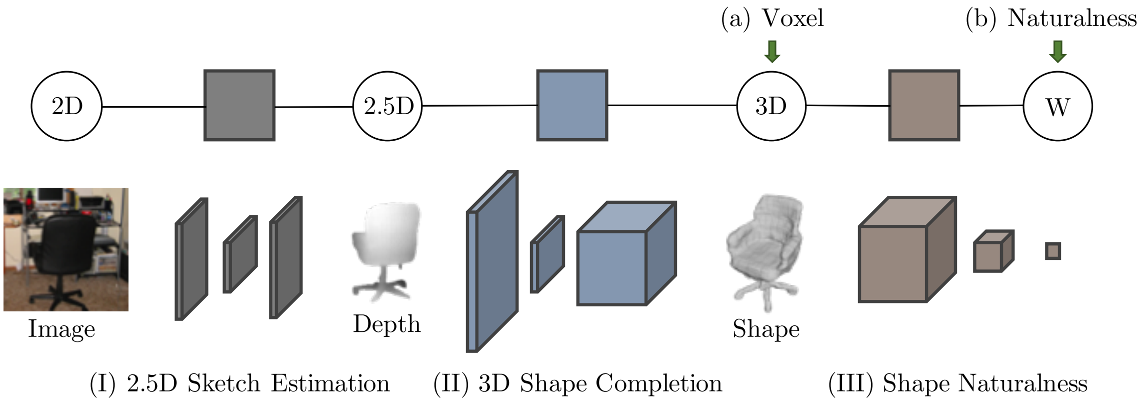


Figure 3-3: For single-view shape reconstruction, ShapeHD contains three components: (I) a 2.5D sketch estimator that predicts depth, surface normal and silhouette images from a single image; (II) a 3D shape completion module that regresses 3D shapes from silhouette-masked depth and surface normal images; (III) an adversarially pretrained convolutional net that serves as the naturalness loss function. While fine-tuning the 3D shape completion net, we use two losses: a supervised loss on the output shape, and a naturalness loss offered by the pretrained discriminator.

view depth and surface normal data. The encoder is adapted from ResNet-18 [26] to take this four-channel 256×256 image as input (one for depth, three for surface normals) and outputs a 200-D latent vector. The vector then goes through a decoder consisting of five transposed convolutional layers and ReLU layers to output a 128×128×128 voxel-based shape reconstruction. Binary cross-entropy losses between predicted and target voxels are used as the supervised loss L_{voxel}.

3.3.1 Shape Naturalness Network

Due to the inherent uncertainty of 3D shape reconstruction from a single view, shape completion networks using encoder-decoder structure with only supervised loss usually predict unrealistic mean shapes. By doing so, they minimize the loss when there exist multiple possible ground truth shapes. We instead introduce an adversarially trained deep naturalness regularizer that penalizes the network for such unrealistic shapes.

We pre-train a 3D generative adversarial network [21] to determine whether a shape is realistic. Its generator synthesizes a 3D shape from a randomly sampled vector, and its discriminator distinguishes generated shapes from real ones. Therefore, the discriminator has the ability to model the real shape distribution and can be used as a naturalness loss for the shape completion network. The generator is not involved in our later training process. Following 3D-GAN [85], we use 5 transposed convolutional layers with batch normalization and ReLU for the generator, and 5 convolutional layers with leaky ReLU for the discriminator.

Due to the high dimensionality of 3D shapes (128×128×128), training a GAN becomes highly unstable. To deal with this issue, we follow Gulrajani *et al.* [23] and use the Wasserstein GAN loss with a gradient penalty to train our adversarial generative network. Specifically,

$$L_{\mathrm{WGAN}} = \mathop{\mathbb{E}}_{\tilde{x}\sim P_g}[D(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim P_r}[D(x)] + \lambda \mathop{\mathbb{E}}_{\hat{x}\sim P_x}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2], \tag{3.1}$$

where D is the discriminator, P_g and P_r are distributions of generated shapes and real shapes, respectively. The last term is the gradient penalty from Gulrajani *et al.* [23]. During training, the discriminator attempts to minimize the overall loss L_{WGAN} while the generator attempts to maximize the loss via the first term in Equation 3.1, so we can define our naturalness loss as

$$L_{\mathrm{natural}} = -\mathop{\mathbb{E}}_{\tilde{x}\sim P_c}[D(\tilde{x})], \tag{3.2}$$

where P_c are the reconstructed shapes from our completion network.

3.3.2 Training Paradigm

We train our network in two stages. We first pre-train the three components of our model separately. The shape completion network is then fine-tuned with both voxel loss and naturalness losses.

Our 2.5D sketch estimation network and 3D completion network are trained with images rendered with ShapeNet [7] objects (see Sections 3.4.1 and 3.5 for details). We train the 2.5D sketch estimation network using a L2 loss and stochastic gradient descent with a learning rate of 0.001 for 120 epochs. For the 3D completion network, we only use the supervised loss L_{voxel} at this stage, again with SGD, a learning rate of 0.1, and a momentum of 0.9 for 80 epochs. The naturalness network is trained with the same set of objects in an adversarial manner, where we use Adam [41] with a learning rate of 0.001 and a batch size of 4 for 80 epochs. We set $\lambda = 10$ as suggested in Gulrajani *et al.* [23].

We then fine-tune our completion network with both voxel loss and naturalness losses as $L = L_{\text{voxel}} + \alpha L_{\text{natural}}$. We compare the scale of gradients from the losses and train our completion network with $\alpha = 3 \times 10^{-11}$ using SGD for 80 epochs. Our model is robust to these parameters; they are only for ensuring gradients of various losses are of the same magnitude.

An alternative is to jointly train the naturalness module with the completion network from scratch using both losses. It seems tempting, but in practice we find that Wasserstein GANs have large losses and gradients, resulting in unstable outputs. We therefore choose to use our pre-training and fine-tuning setup.

3.4 Single-View Shape Completion

We present results on 3D shape completion from a single depth image. Here, we only use the last two modules of the model: the 3D shape estimator and deep naturalness network.

3.4.1 Setup

Data. We render each of the ShapeNet Core55 [7] objects from the aeroplane, car and chair categories in 20 random, fully unconstrained views. Specifically, for each view, we randomly set the azimuth and elevation angles of the camera, but the camera up vector is fixed to be the world $+y$ axis, and the camera always looks at the object center. The focal length is fixed at 50mm with a 35mm film. We use Mitsuba [32], a physically-based graphics engine, for all our renderings.

We render the ground-truth depth image of each object in all 20 views. Depth values are measured from the camera center (*i.e.*, ray depth), rather than from the image plane. To approximate depth scanner data, we also generate the accompanying ground-truth surface normal images from the raw depth data, as surface normal maps are the common by-products of depth scanning. All our rendered surface normal vectors are defined in the camera space.

Baselines. We compare with the state-of-the-art: 3D-EPN [13]. To ensure a fair comparison, we convert depth maps to partial surfaces registered in a canonical global coordinate defined

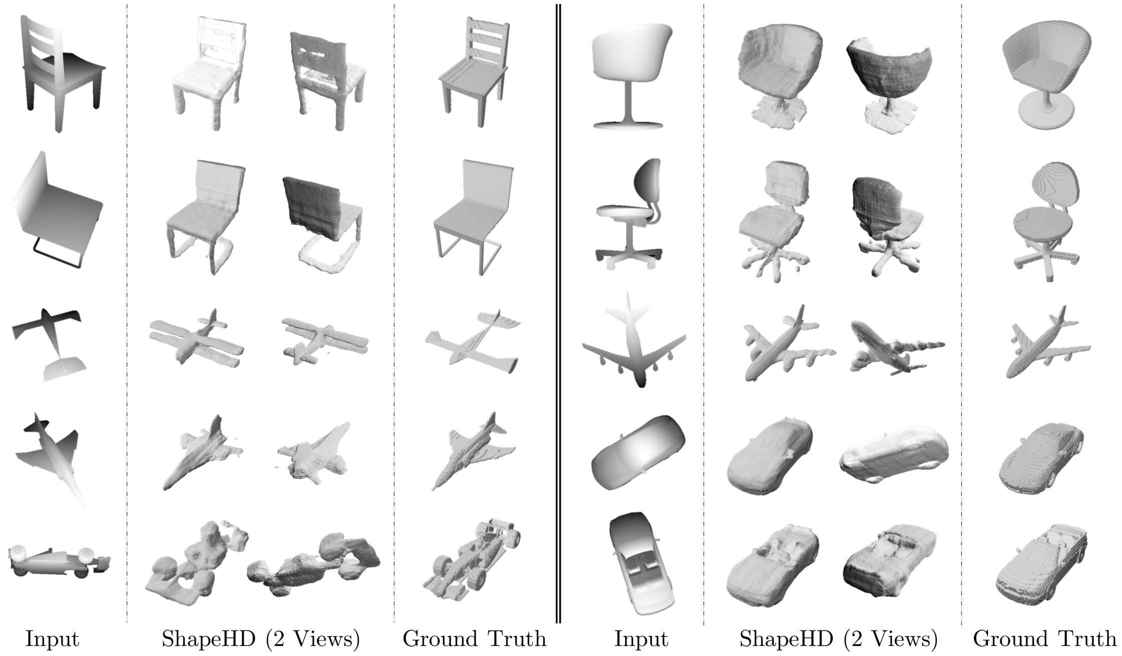


Figure 3-4: Results on 3D shape completion from single-view depth. From left to right: input depth maps, shapes reconstructed by ShapeHD in the canonical view and a novel view, and ground truth shapes in the canonical view. Assisted by the adversarially learned naturalness losses, ShapeHD recovers highly accurate 3D shapes with fine details. Sometimes the reconstructed shape deviates from the ground truth, but can be viewed as another plausible explanation of the input (*e.g.*, the airplane on the left, third row).

by ShapeNet Core55 [7], which is required by 3D-EPN.

Metrics. We use two standard metrics for quantitative comparisons: Intersection over Union (IoU) and Chamfer Distance (CD). In particular, Chamfer distance can be applied to various shape representations including voxels (by sampling points on the isosurface) and point clouds.

3.4.2 Results on ShapeNet

Qualitative results. In Figure 3-4, we show 3D shapes predicted by ShapeHD from single-view depth images. While common encoder-decoder structure usually generates mean shapes with few details, our ShapeHD predicts shapes with large variance and fine details. In addition, even when there is strong occlusion in the depth image, our model can predict a high-quality, plausible 3D shape that looks good perceptually, and infer parts not

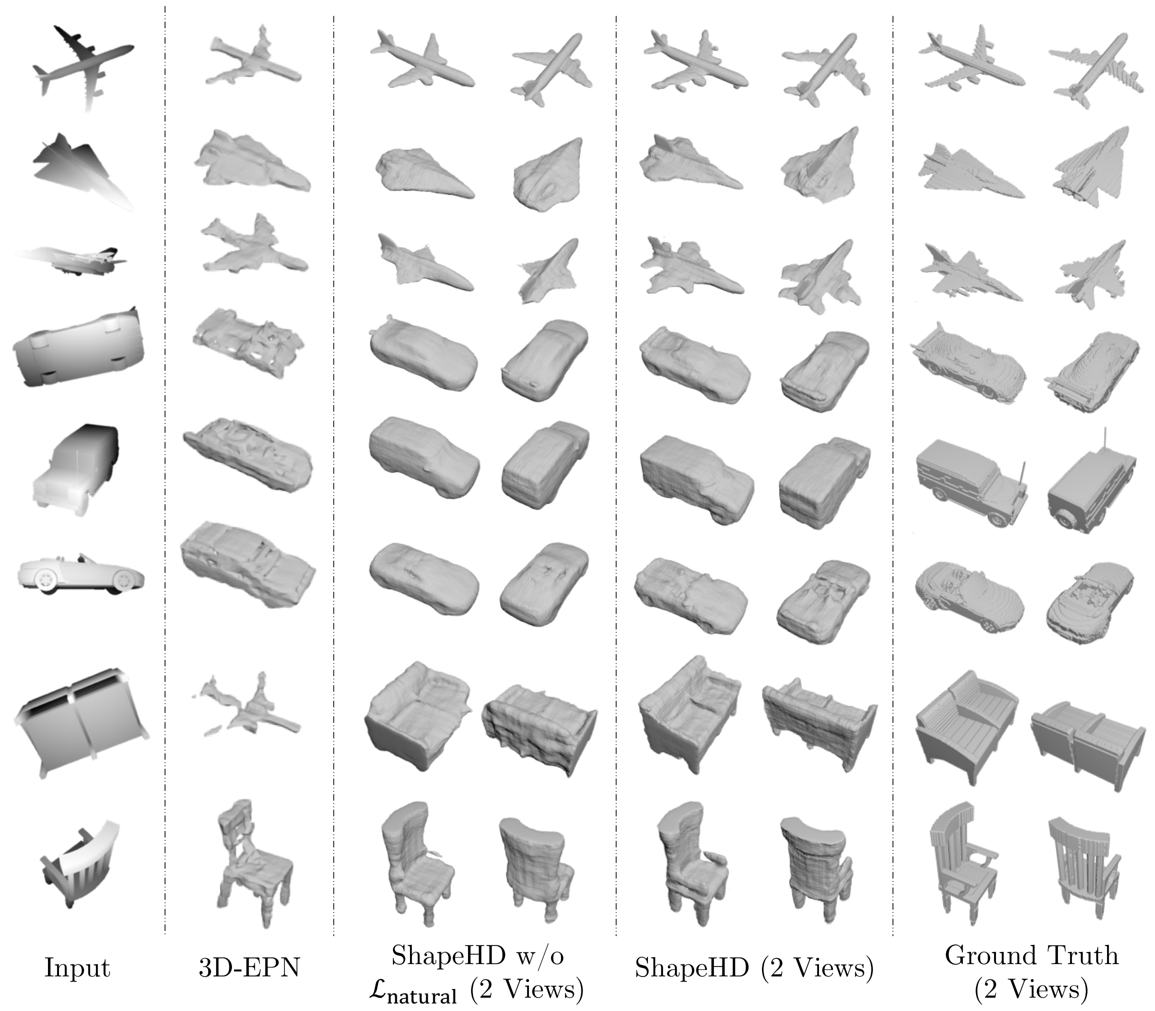


Figure 3-5: Our results on 3D shape completion, compared with the state-of-the-art, 3D-EPN [13], and our model but without naturalness losses. Our results contain more details than 3D-EPN. We observe that the adversarially trained naturalness losses help fix errors, add details (*e.g.*, the plane wings in row 3, car seats in row 6, and chair arms in row 8), and smooth planar surfaces (*e.g.*, the sofa back in row 7).

present in the input images.

Ablation. When using naturalness loss, the network is penalized for generating mean shapes that are unreasonable but minimize the supervised loss. In Figure 3-5, we show reconstructed shapes from our ShapeHD with and without naturalness loss (*i.e.* before fine-tuning with L_{natural}), together with ground truth shapes and shapes predicted by 3D-EPN [13]. Our results contain finer details compared with those from 3D-EPN. Also, the performance of ShapeHD improves greatly with the naturalness loss, which predicts more reasonable and complete shapes.

Methods	IoU				CD			
	chair	car	plane	avg	chair	car	plane	avg
3D-EPN [13]	.147	.274	.155	.181	.227	.200	.125	.192
ShapeHD w/o L_{natural}	.466	**.698**	**.488**	**.529**	.112	.083	.071	.093
ShapeHD	**.488**	**.698**	.452	**.529**	**.096**	**.078**	**.068**	**.084**

Table 3.1: Average IoU scores (32^3) and CDs for 3D shape completion on ShapeNet [7]. Our model outperforms the state-of-the-art by a large margin. The learned naturalness losses significantly lower the CDs between our completion results and ground truth across all categories.

Quantitative results. We present quantitative results in Table 3.1. Our ShapeHD outperforms the state-of-the-art by a margin in all metrics. Our method outputs shapes at the resolution of 128^3, while shapes produced by 3D-EPN are of resolution 32^3. Therefore, for a fair comparison, we downsample our predicted shapes to 32^3 and report results of both methods in that resolution. The original 3D-EPN paper suggests a post-processing step that retrieves similar patches from a shape database for results of a higher resolution. Practically, we find this steps takes 18 hours for a single image. We therefore report results without post-processing for both methods.

Table 3.1 also suggests the naturalness loss improve the completion results, achieving comparable IoU scores and better (lower) CDs. CD has been reported to be better at capturing human perception of shape quality [71].

3.4.3 Results on Real Depth Scans

We now show results of ShapeHD on real depth scans. We capture six depth maps of different chairs using a Structure sensor* and use the captured depth maps to evaluate our model. All the corresponding normal maps used as inputs are estimated from depth measurements. Figure 3-6 shows that ShapeHD completes 3D shapes well given a single-view depth map. Our ShapeHD is more flexible than 3D-EPN, as we do not need any camera intrinsics or extrinsics to register depth maps. In our case, none of these parameters are known and thus 3D-EPN cannot be applied.

*`http://structure.io`

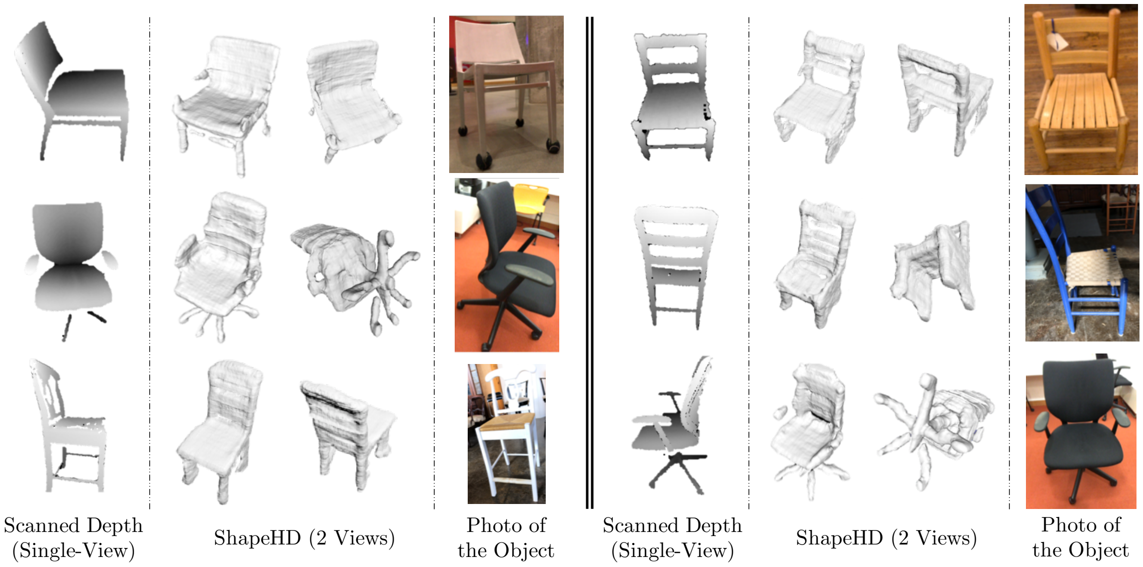


Figure 3-6: Results of 3D shape completion on depth data from a physical scanner. Our model is able to reconstruct the shape well from just a single view. From left to right: input depth, two views of our single-view completion results, and a color image of the object from a side view.

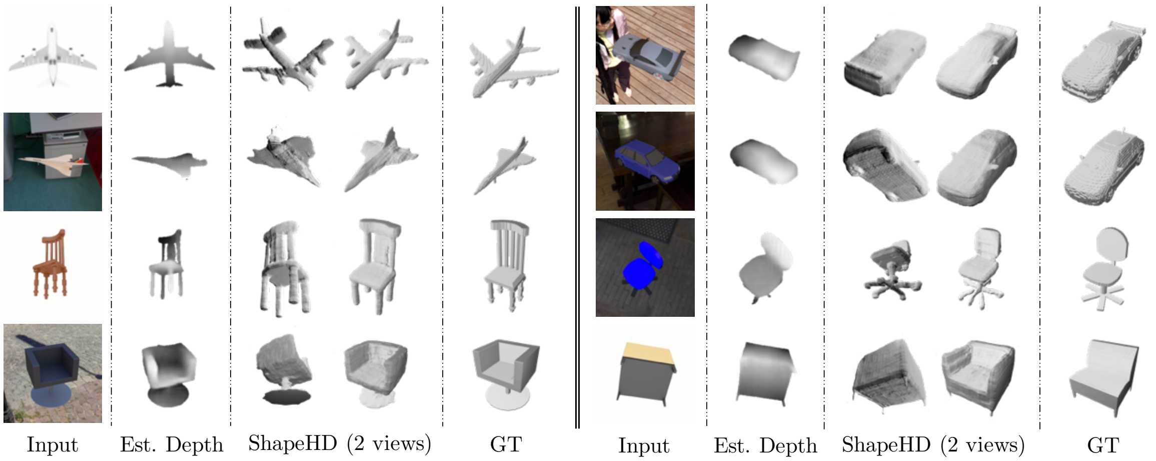


Figure 3-7: Results on 3D reconstruction from a single RGB image. ShapeHD is able to accurately recover the full object shape from an image. From left to right: input images, estimated depth maps, shapes reconstructed by ShapeHD, and ground truth shapes.

3.5 3D Shape Reconstruction

We now evaluate ShapeHD on 3D shape reconstruction from a single color image.

RGB image preparation. For the task of single-image 3D reconstruction, we need to render RGB images that correspond to the depth images for training. We follow the same

Methods	IoU (32^3/128^3)				CD			
	chair	car	plane	avg	chair	car	plane	avg
3D-R2N2 [12]	.167/.120	.255/.184	.149/.105	.183/.131	.242	.255	.262	.251
DRC (3D) [79]	.221/.169	.334/.277	.246/.167	.255/.195	.184	.131	.176	.169
PSGN [18]*	-	-	-	-	.208	.157	.164	.183
AtlasNet [22]*	-	-	-	-	.152	.162	.154	.155
OGN [76]	-	.398/.324	-	-	-	.087	-	-
ShapeHD	**.426/.352**	**.621/.478**	**.412/.348**	**.470/.382**	**.106**	**.078**	**.073**	**.090**

Table 3.2: IoU scores and CDs for 3D shape reconstruction on ShapeNet [7]. Our rendering of ShapeNet is more challenging than that from Choy *et al.* [12]; as such, the numbers for other methods may differ from those reported by the original authors. All methods were trained with full 3D supervision. *3D-R2N2, DRC, OGN, and ShapeHD take a single image as input, while PSGN and AltasNet require ground truth object silhouettes as additional input. Also, PSGN and AtlasNet generate surface point clouds without guaranteeing watertight meshes and therefore cannot be evaluated in IoU. OGN is trained only on cars and hence evaluated the same way.

camera setup specified earlier. Additionally, to boost the realism of the rendered RGB images, we put three different types of backgrounds behind the object during rendering. One third of the images are rendered in a clean white background; one third are rendered in high-dynamic-range backgrounds with illumination channels that produce realistic lighting. We render the remaining one third images with backgrounds randomly sampled from the SUN database [92].

Baselines. We compare our ShapeHD with the state-of-the-art in 3D shape reconstruction, including 3D-R2N2 [12], point set generation network (PSGN) [18], differentiable ray consistency (DRC) [79], octree generating network (OGN) [76], and AtlasNet [22]. All these models are trained on rendered images using ShapeNet objects with ground truth 3D shapes as supervision. The DRC model is fine-tuned on PASCAL 3D+ using ray consistency losses after training on rendered images.

3D-R2N2, DRC, OGN, and our ShapeHD take a single image as input, while PSGN and AltasNet require ground truth object silhouettes as additional input. Also, PSGN and AtlasNet generate surface point clouds without guaranteeing watertight meshes or voxels and therefore cannot be evaluated in IoU.

Qualitative results. We evaluate on three datasets: a synthetic dataset using renderings

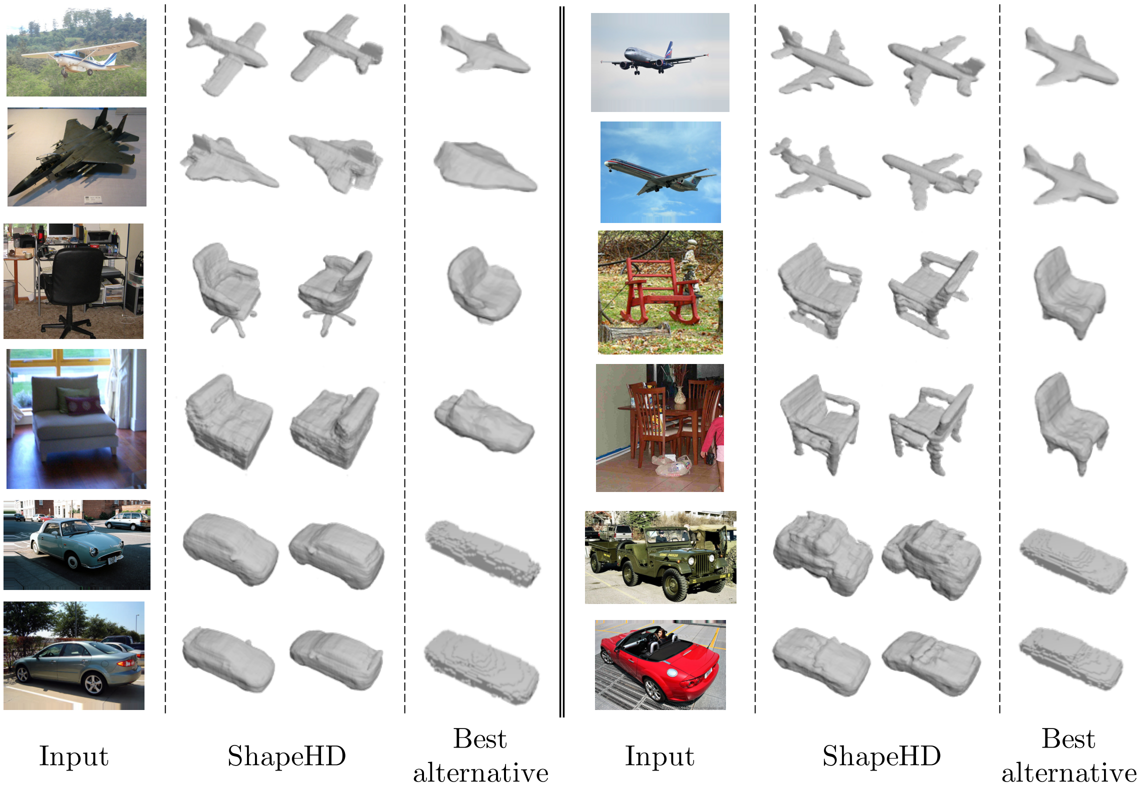
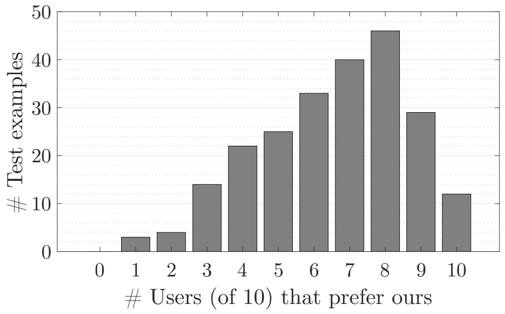


Figure 3-8: Results on 3D shape reconstruction from a single RGB image on the PASCAL 3D+ dataset [88]. From left to right: input, two views of reconstructions from ShapeHD, and reconstructions by the best alternative methods in Table 3.3. Assisted by the learned naturalness losses, ShapeHD recovers accurate 3D shapes with fine details.

Methods	CD			
	chair	car	plane	avg
3D-R2N2 [12]	0.238	0.305	0.305	0.284
DRC (3D) [79]	0.158	0.099	0.112	0.122
OGN [76]	-	**0.087**	-	-
ShapeHD	**0.137**	0.129	**0.094**	**0.119**

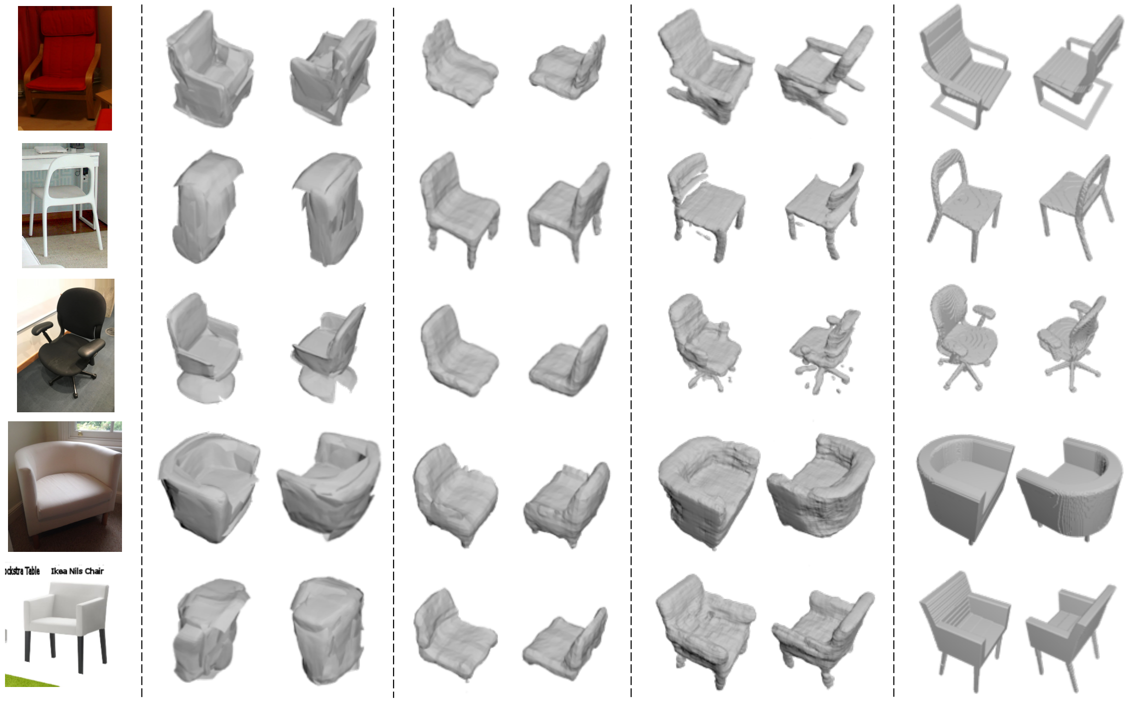
(a) CDs on PASCAL 3D+ [88]



(b) Human Study results

Table 3.3: Results for 3D shape reconstruction on PASCAL 3D+ [88]. (a) We compare our ShapeHD with 3D-R2N2, DRC, and OGN. PSGN and AtlasNet are not evaluated, because they require object masks as additional input, but PASCAL 3D+ has only inaccurate masks. (b) In the behavioral study, most users prefer our constructions on most images. Overall, our reconstructions are preferred 64.5% of the time to OGN's.

of ShapeNet [7], and two real datasets, PASCAL 3D+ [88] and Pix3D [71]. We present reconstructed 3D shapes in Figures 3-7, 3-8 and 3-9. In general, our ShapeHD is able to



(a) Input (b) AtlasNet (c) DRC (3D) (d) ShapeHD (e) GT

Figure 3-9: Results of 3D shape reconstruction from a single RGB image on the Pix3D dataset [71]. For each input image, we show reconstruction results by AtlasNet, DRC, our ShapeHD, and ground truth. Our ShapeHD reconstructs complete 3D shapes with fine details that resemble the ground truth.

	3D-R2N2 [12]	DRC (3D) [79]	PSGN [18]*	AtlasNet [22]*	ShapeHD
IoU (32^3)	0.136	0.265	-	-	**0.284**
IoU (128^3)	0.089	0.185	-	-	**0.205**
CD	0.239	0.160	0.199	0.126	**0.123**

Table 3.4: 3D shape reconstruction results on Pix3D [71]. All methods were trained with full 3D supervision on rendered images of ShapeNet objects. *3D-R2N2, DRC, and ShapeHD take a single image as input, while PSGN and AtlasNet require the ground truth mask as input. Also, PSGN and AtlasNet generate surface point clouds without guaranteeing watertight meshes and therefore cannot be evaluated in IoU.

predict 3D shapes that closely resemble the ground truth shapes, giving fine details that make the reconstructed shapes more realistic. Additionally, ShapeHD infers a reasonable shape even in the presence of strong self-occlusions. Our ShapeHD also reconstructs the final 3D shapes with fine details on the two real datasets.

In particular, in Figure 3-8, we compare our reconstructions with the best-performing alternatives (DRC on chairs and airplanes, and AtlasNet on cars). In addition to preserving details, our model captures the shape variations of the objects, while the competitors produce similar reconstructions across instances.

Quantitative results. Quantitatively, Tables 3.2, 3.3, and 3.4 suggest that ShapeHD performs significantly better than the other methods in almost all metrics. The only exception is the CD on PASCAL 3D+ cars, where OGN performs the best. However, as PASCAL 3D+ only has around 10 CAD models for each object category as ground truth 3D shapes, the ground truth labels and the scores can be inaccurate, failing to reflect human perception [79].

We therefore conduct an additional user study, where we show an input image and its two reconstructions (from ShapeHD and from OGN, each in two views) to users on Amazon Mechanical Turk, and ask them to choose the shape that looks closer to the object in the image. For each image, we collect 10 responses from "Masters" (workers who have demonstrated excellence across a wide range of HITs). Table 3.3b suggests that on most images, most users prefer our reconstruction to OGN's. In general, our reconstructions are preferred 64.5% of the time.

3.6 Analyses

We want to understand what the network has learned. In this section, we present a few analyses to visualize what the network is learning, analyze the effect of the naturalness loss function over time, and discuss common failure modes.

Network visualization. As the network successfully reconstructs object shape and parts, it is natural to ask if it learns object or part detectors implicitly. To this end, we visualize the top activating regions across all validation images for units in the last convolutional layer of the encoder in our 3D completion network, using the method proposed by Zhou *et al.* [97]. As shown in Figure 3-10, the network indeed learns a diverse and rich set of object and part detectors. There are detectors that attend to car wheels, chair backs, chair arms, chair legs, and airplane engines. Also note that many detectors respond to certain patterns (*e.g.*, strided) in particular, which is probably contributing to the fine details in the reconstruction.

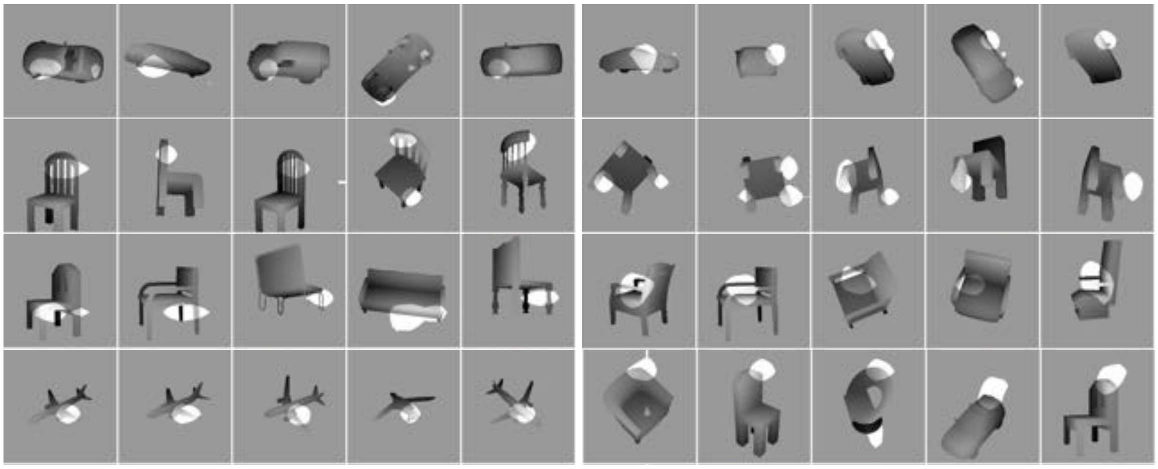


Figure 3-10: Visualizations on how ShapeHD attends to details in depth maps. Row 1: car wheel detectors. Row 2: chair back and chair leg detectors. The left one responds to the strided pattern in particular. Row 3: chair leg and chair arm detectors. Row 4: airplane engine and curved surface detectors. Note that the right one responds to a specific pattern across classes.

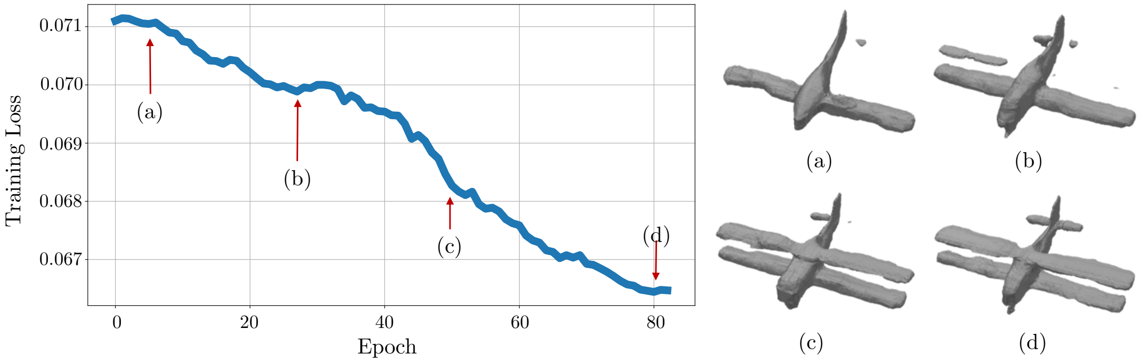


Figure 3-11: Visualizations on how ShapeHD evolves over time with naturalness losses: the predicted shape becomes increasingly realistic as details are being added.

Additionally, there are units that respond to generic shape patterns across categories, like the curve detector in the bottom right.

Training with naturalness loss over time. We study how the effect of the naturalness loss evolves over time. In Figure 3-11, we plot the losses of the completion network with respect to the fine-tuning iterations. We realize the voxel loss consistently goes down, but slowly and marginally. However, if we visualize the reconstructed examples at different timestamps, we clearly see details are being added to the shapes. These fine details occupy a small region in the voxel grid, and thus training with supervised loss alone is unlikely to recover them. In contrast, with adversarially training perceptual losses, our model recovers

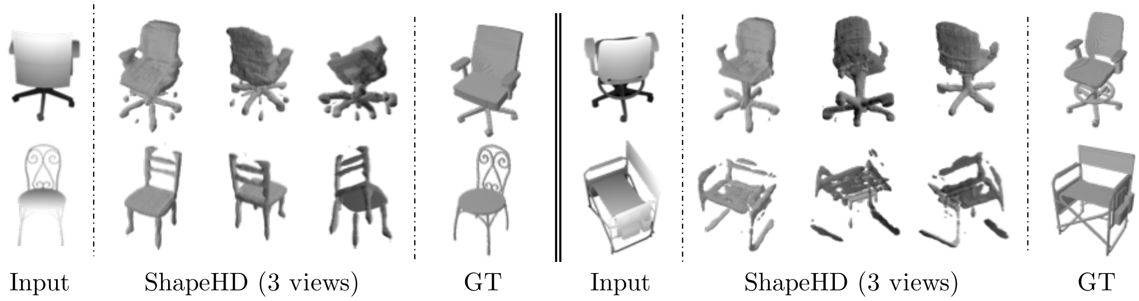


Figure 3-12: Common failure modes of our system. Top left: the model sometimes gets confused by deformable object parts (*e.g.*, wheels). Top right: the model might miss uncommon object parts (the ring above the wheels). Bottom row: the model has difficulty in recovering very thin structure, and may generate other structure patterns instead.

details successfully.

Failure cases. We present failure cases in Figure 3-12 to understand the limitations of our model. We observe our model has these common failing modes: the model sometimes gets confused by the deformable object parts (*e.g.*, wheels on the top left); the model might miss uncommon object parts (top right, the ring above the wheels); the model has difficulty in recovering very thin structure (bottom right), and may generate other structure patterns instead (bottom left).

Chapter 4

Conclusion

In this thesis, we explored the learning of 3D shape priors via generative-adversarial modeling and applied it to the tasks of 3D shape generation, completion and reconstruction.

In Chapter 2, we proposed 3D-GAN for 3D object generation, as well as 3D-VAE-GAN for learning an image to 3D model mapping. We demonstrated that our models are able to generate novel objects and to reconstruct 3D objects from images. We showed that the discriminator in GAN, learned without supervision, can be used as an informative feature representation for 3D objects, achieving impressive performance on shape classification. We also explored the latent space of object vectors, and presented results on object interpolation, shape arithmetic, and neuron visualization.

In Chapter 3, we have proposed to use learned shape priors to overcome the 2D-3D ambiguity and to learn from the multiple hypotheses that explain a single-view observation. With an adversarially learned shape prior loss function, our ShapeHD achieves state-of-the-art results on 3D shape completion both qualitatively and quantitatively. We have also included additional analyses to reveal the learned representation, present how the prior module contributes over time, and discuss failure cases. ShapeHD has also been extended to perform 3D shape reconstruction from a single RGB image. We hope our results will inspire further research in 3D shape modeling.

Appendix A

Technical Details for 3D-GAN and 3D-VAE-GAN

A.1 Network Structure

Here we give the network structures of the generator, the discriminator, and the image encoder.

Generator The generator consists of five fully convolution layers with numbers of channels $\{512, 256, 128, 64, 1\}$, kernel sizes $\{4, 4, 4, 4, 4\}$, and strides $\{1, 2, 2, 2, 2\}$. We add ReLU and batch normalization layers between convolutional layers, and a Sigmoid layer at the end. The input is a 200-dimensional vector, and the output is a $64 \times 64 \times 64$ matrix with values in $[0, 1]$.

Discriminator As a mirrored version of the generator, the discriminator takes as input a $64 \times 64 \times 64$ matrix, and outputs a real number in $[0, 1]$. The discriminator consists of 5 volumetric convolution layers, with numbers of channels $\{64, 128, 256, 512, 1\}$, kernel sizes $\{4, 4, 4, 4, 4\}$, and strides $\{2, 2, 2, 2, 1\}$. There are leaky ReLU layers of parameter 0.2 and batch normalization layers in between, and a Sigmoid layer at the end.

Image encoder The image encoder in our 3D-VAE-GAN takes a $3 \times 256 \times 256$ image as input, and outputs a 200-dimensional vector. It consists of five spatial convolution layers with numbers of channels $\{64, 128, 256, 512, 400\}$, kernel sizes $\{11, 5, 5, 5, 8\}$, and strides $\{4, 2, 2, 2, 1\}$, respectively. There are ReLU and batch normalization layers in between. The output of the last convolution layer is a 400-dimensional vector representing a Gaussian distribution in the 200-dimensional space, where 200 dimensions are for the mean and the other 200 dimensions are for the diagonal variance. There is a sampling layer at the end to sample a 200-dimensional vector from the Gaussian distribution, which is later used by the 3D-GAN.

A.2 3D-VAE-GAN Training

Let $\{x_i, y_i\}$ be the set of training pairs, where y_i is a 2D image and x_i is the corresponding 3D shape. In each iteration t of training, we first generate a random sample z_t from $N(\mathbf{0}, \mathbf{I})^*$. Then we update the discriminator D, the image encoder E, and the generator G sequentially. Specifically,

- Step 1: Update the discriminator D by minimizing the following loss function:

$$\log D(x_i) + \log(1 - D(G(z_t))). \tag{A.1}$$

- Step 2: Update the image encoder E by minimizing the following loss function:

$$D_{\text{KL}}\left(N(E_{\text{mean}}(y_i), E_{\text{var}}(y_i)) \mid\mid N(\mathbf{0}, \mathbf{I})\right) + ||G(E(y_i)) - x_i||_2, \tag{A.2}$$

  where $E_{\text{mean}}(y_i)$ and $E_{\text{var}}(y_i)$ are the predicted mean and variance of the latent variable z, respectively.

- Step 3: Update the generator G by minimizing the following loss function:

$$\log(1 - D(G(z_t))) + ||G(E(y_i)) - x_i||_2. \tag{A.3}$$

***I** is an identity matrix.

Bibliography

[1] Aayush Bansal and Bryan Russell. Marr revisited: 2d-3d alignment via surface normal prediction. In *CVPR*, 2016. 19, 34

[2] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE TPAMI*, 37(8):1670–1687, 2015. 34

[3] Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic images in the wild. *ACM TOG*, 33(4):159, 2014. 34

[4] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, 1999. 19

[5] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Generative and discriminative voxel modeling with convolutional neural networks. In *NIPS Workshop*, 2016. 33

[6] Wayne E Carlson. An algorithm and data structure for 3d object synthesis using surface patch intersections. In *SIGGRAPH*, 1982. 17, 19

[7] 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:1512.03012*, 2015. 31, 33, 34, 37, 38, 39, 41, 43, 44

[8] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 17, 25

[9] Siddhartha Chaudhuri, Evangelos Kalogerakis, Leonidas Guibas, and Vladlen Koltun. Probabilistic reasoning for assembly-based 3d modeling. *ACM TOG*, 30(4):35, 2011. 19

[10] Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. On visual similarity based 3d model retrieval. *CGF*, 2003. 25

[11] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. In *NIPS*, 2016. 34

[12] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, 2016. 19, 31, 34, 43, 44, 45

[13] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *CVPR*, 2017. 31, 34, 35, 38, 40, 41

[14] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, 2015. 19

[15] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *NIPS*, 2016. 34

[16] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. In *CVPR*, 2015. 29

[17] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 34

[18] Haoqiang Fan, Hao Su, and Leonidas Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, 2017. 34, 43, 45

[19] Michael Firman, Oisin Mac Aodha, Simon Julier, and Gabriel J Brostow. Structured Completion of Unobserved Voxels from a Single Depth Image. In *CVPR*, 2016. 33

[20] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *ECCV*, 2016. 17, 18, 19, 24, 25, 26, 27, 28, 29, 34

[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 18, 19, 20, 21, 32, 34, 36

[22] Thibault Goueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russel, and Mathieu Aubry. Atlasnet: A papier-mÃćchÃľ approach to learning 3d surface generation. In *CVPR*, 2018. 34, 43, 45

[23] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In *NIPS*, 2017. 37, 38

[24] JunYoung Gwak, Christopher B Choy, Manmohan Chandraker, Animesh Garg, and Silvio Savarese. Weakly supervised 3d reconstruction with adversarial constraint. In *3DV*, 2017. 35

[25] Christian Häne, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3d object reconstruction. In *3DV*, 2017. 34

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2015. 35, 36

[27] Berthold KP Horn and Michael J Brooks. *Shape from shading*. MIT press, 1989. 34

[28] Haibin Huang, Evangelos Kalogerakis, and Benjamin Marlin. Analysis and synthesis of 3D shape families via deep-learned generative models of surfaces. *CGF*, 34(5): 25–38, 2015. 19

[29] Daniel Jiwoong Im, Chris Dongjoo Kim, Hui Jiang, and Roland Memisevic. Generating images with recurrent adversarial networks. *arXiv:1602.05110*, 2016. 20

[30] Phillip Isola, Daniel Zoran, Dilip Krishnan, and Edward H Adelson. Learning visual groups from co-occurrences in space and time. In *ICLR Workshop*, 2016. 34

[31] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard A. Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew J. Davison, and Andrew W. Fitzgibbon. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *UIST*, 2011. 34

[32] Wenzel Jakob. Mitsuba renderer, 2010. http://www.mitsuba-renderer.org. 38

[33] Michael Janner, Jiajun Wu, Tejas Kulkarni, Ilker Yildirim, and Joshua B Tenenbaum. Self-Supervised Intrinsic Image Decomposition. In *NIPS*, 2017. 34

[34] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 34

[35] Evangelos Kalogerakis, Siddhartha Chaudhuri, Daphne Koller, and Vladlen Koltun. A probabilistic model for component-based shape synthesis. *ACM TOG*, 31(4):55, 2012. 19

[36] Abhishek Kar, Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. In *CVPR*, 2015. 19, 34

[37] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM TOG*, 32(3):29, 2013. 33

[38] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Rotation invariant spherical harmonic representation of 3 d shape descriptors. In *SGP*, 2003. 25

[39] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *SGP*, SGP '06, 2006. 33

[40] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 21

[41] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 37

[42] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 18, 20

[43] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *ICML*, 2016. 18, 22

[44] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv:1609.04802*, 2016. 34

[45] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *ECCV*, 2016. 20

[46] Yangyan Li, Angela Dai, Leonidas Guibas, and Matthias Nießner. Database-assisted object retrieval for real-time 3d reconstruction. *CGF*, 34(2):435–446, 2015. 33

[47] Yangyan Li, Hao Su, Charles Ruizhongtai Qi, Noa Fish, Daniel Cohen-Or, and Leonidas J Guibas. Joint embeddings of shapes and images via cnn image purification. In *SIGGRAPH Asia*, volume 34, page 234. ACM, 2015. 19

[48] Joseph J. Lim, Hamed Pirsiavash, and Antonio Torralba. Parsing ikea objects: Fine pose estimation. In *ICCV*, 2013. 24, 27

[49] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, 2013. 21

[50] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *IROS*, 2015. 18, 19, 25, 26

[51] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J Davison. Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? In *ICCV*, 2017. 34

[52] Niloy J Mitra, Leonidas J Guibas, and Mark Pauly. Partial and approximate symmetry detection for 3d geometry. *ACM TOG*, 25(3):560–568, 2006. 33

[53] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Laplacian mesh optimization. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 381–389. ACM, 2006. 33

[54] David Novotny, Diane Larlus, and Andrea Vedaldi. Learning 3d object categories by looking around them. In *ICCV*, 2017. 34

[55] 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 *CVPR*, 2016. 17, 19, 25, 26

[56] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 18, 19, 21, 29, 32, 34

[57] Danilo Jimenez Rezende, SM Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3d structure from images. In *NIPS*, 2016. 19, 34

[58] Gernot Riegler, Ali Osman Ulusoy, Horst Bischof, and Andreas Geiger. Octnetfusion: Learning depth fusion from data. In *3DV*, 2017. 34

[59] Gernot Riegler, Ali Osman Ulusoys, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *CVPR*, 2017. 34

[60] Lauren A Schmidt. *Meaning and compositionality as statistical induction of categories and constraints*. PhD thesis, Massachusetts Institute of Technology, 2009. 15

[61] Nima Sedaghat, Mohammadreza Zolfaghari, and Thomas Brox. Orientation-boosted voxel nets for 3d object recognition. *arXiv preprint arXiv:1604.03351*, 2016. 25, 26

[62] Abhishek Sharma, Oliver Grau, and Mario Fritz. Vconv-dae: Deep volumetric shape learning without object labels. In *ECCV Workshop*, 2016. 18, 19, 24, 25, 26

[63] Baoguang Shi, Song Bai, Zhichao Zhou, and Xiang Bai. Deeppano: Deep panoramic representation for 3-d shape recognition. *IEEE SPL*, 22(12):2339–2343, 2015. 18, 19, 25, 26

[64] Jian Shi, Yue Dong, Hao Su, and Stella X Yu. Learning non-lambertian object intrinsics across shapenet categories. In *CVPR*, 2017. 34

[65] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 34

[66] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, 2017. 34

[67] Olga Sorkine and Daniel Cohen-Or. Least-squares meshes. In *Shape Modeling Applications*, 2004. 33

[68] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR Workshop*, 2015. 30

[69] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *ICCV*, 2015. 17, 19, 25, 26

[70] Hao Su, Charles R Qi, Yangyan Li, and Leonidas Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *ICCV*, 2015. 19

[71] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Tianfan Xue, Joshua B. Tenenbaum, and William T. Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *CVPR*, 2018. 41, 44, 45

[72] Minhyuk Sung, Vladimir G Kim, Roland Angst, and Leonidas Guibas. Data-driven structural priors for shape completion. *ACM TOG*, 34(6):175, 2015. 33

[73] Johan WH Tangelder and Remco C Veltkamp. A survey of content based 3d shape retrieval methods. *Multimedia tools and applications*, 39(3):441–471, 2008. 17, 19

[74] Marshall F Tappen, William T Freeman, and Edward H Adelson. Recovering intrinsic images from a single image. In *NIPS*, 2003. 34

[75] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Multi-view 3d models from single images with a convolutional network. In *ECCV*, 2016. 34

[76] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *ICCV*, 2017. 34, 43, 44

[77] Duc Thanh Nguyen, Binh-Son Hua, Khoi Tran, Quang-Hieu Pham, and Sai-Kit Yeung. A field model for repairing 3d shapes. In *CVPR*, 2016. 33

[78] Sebastian Thrun and Ben Wegbreit. Shape from symmetry. In *ICCV*, 2005. 33

[79] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, 2017. 31, 34, 43, 44, 45, 46

[80] Oliver Van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. *CGF*, 2011. 17, 19

[81] Xiaolong Wang and Abhinav Gupta. Generative image modeling using style and structure adversarial networks. In *ECCV*, 2016. 20

[82] Xiaolong Wang, David Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *CVPR*, 2015. 34

[83] Yair Weiss. Deriving intrinsic images from image sequences. In *ICCV*, 2001. 34

[84] Jiajun Wu, Tianfan Xue, Joseph J Lim, Yuandong Tian, Joshua B Tenenbaum, Antonio Torralba, and William T Freeman. Single image 3d interpreter network. In *ECCV*, 2016. 19

[85] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. In *NIPS*, 2016. 32, 34, 35, 37

[86] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, William T Freeman, and Joshua B Tenenbaum. MarrNet: 3D Shape Reconstruction via 2.5D Sketches. In *NIPS*, 2017. 34

[87] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, 2015. 17, 18, 19, 23, 24, 25, 26, 33

[88] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *WACV*, 2014. 31, 44

[89] Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese. Data-driven 3d voxel patterns for object category recognition. In *CVPR*, 2015. 19

[90] Yu Xiang, Wonhui Kim, Wei Chen, Jingwei Ji, Christopher Choy, Hao Su, Roozbeh Mottaghi, Leonidas Guibas, and Silvio Savarese. Objectnet3d: A large scale database for 3d object recognition. In *ECCV*, 2016. 31

[91] Jianxiong Xiao, James Hays, Krista Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 23

[92] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 43

[93] Tianfan Xue, Jianzhuang Liu, and Xiaoou Tang. Example-based 3d object reconstruction from line drawings. In *CVPR*, 2012. 19

[94] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *NIPS*, 2016. 34

[95] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *NIPS*, 2016. 19

[96] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape-from-shading: a survey. *IEEE TPAMI*, 21(8):690–706, 1999. 34

[97] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. In *ICLR*, 2014. 46

[98] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, 2016. 20, 35