Learning View Priors for Single-view 3D Reconstruction

Hiroharu Kato[1] and Tatsuya Harada[1,2]
[1]The University of Tokyo, [2]RIKEN
{kato,harada}@mi.t.u-tokyo.ac.jp

Abstract

There is some ambiguity in the 3D shape of an object when the number of observed views is small. Because of this ambiguity, although a 3D object reconstructor can be trained using a single view or a few views per object, reconstructed shapes only fit the observed views and appear incorrect from the unobserved viewpoints. To reconstruct shapes that look reasonable from any viewpoint, we propose to train a discriminator that learns prior knowledge regarding possible views. The discriminator is trained to distinguish the reconstructed views of the observed viewpoints from those of the unobserved viewpoints. The reconstructor is trained to correct unobserved views by fooling the discriminator. Our method outperforms current state-of-the-art methods on both synthetic and natural image datasets; this validates the effectiveness of our method.

1. Introduction

Humans can estimate the 3D structure of an object in a single glance. We utilize this ability to grasp objects, avoid obstacles, create 3D models using CAD, and so on. This is possible because we have gained prior knowledge about the shapes of 3D objects.

Can machines also acquire this ability? This problem is called *single-view 3D object reconstruction* in computer vision. A straightforward approach is to train a reconstructor using 2D images and their corresponding ground truth 3D models [3, 5, 9, 13, 16, 27, 30]. However, creating 3D annotations requires extraordinary effort from professional 3D designers. Another approach is to train a reconstructor using a single view or multiple views of an object without explicit 3D supervision [15, 17, 18, 31, 36]. We call this approach *view-based training*. This approach typically requires annotations of silhouettes of objects and viewpoints, which are relatively easy to obtain.

Because a ground truth 3D shape is not given in view-based training, there is some ambiguity in the possible shapes. In other words, several different 3D shapes can be projected into the same 2D view, as shown in the upper half and lower half of Figure 1. To reduce this ambiguity, twenty or more views per object are typically used in training [18, 36]. However, this is not practical in many cases in terms of feasibility and scalability. When creating a dataset by taking photos, if an object is moving or deforming, it is difficult to take photos from many viewpoints. In addition, when creating a dataset using a large number of photos from the Internet, it is not always possible to collect multiple views of an object. Therefore, it is desirable that a reconstructor can be trained using a few views or even a single view of an object.

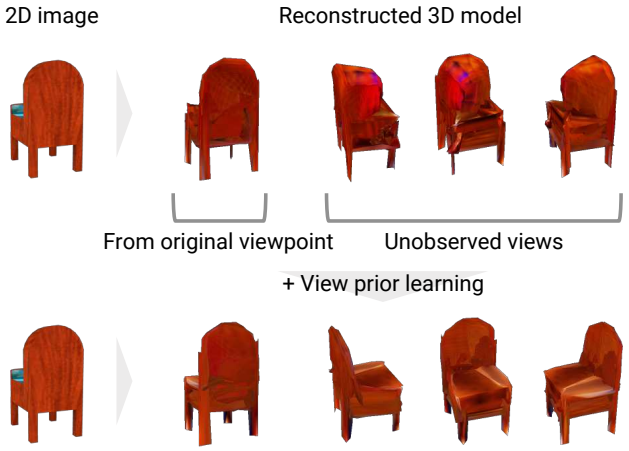

Figure 1. When a 3D reconstructor is trained using only a single view per object, because of ambiguity in the 3D shape of an object, it reconstructs a shape which only fits the observed view and looks incorrect from unobserved viewpoints (upper). By introducing a discriminator that learns prior knowledge of correct views, the reconstructor is able to generate a shape that is viewed as reasonable from any viewpoint (lower).

In this work, we focus on training a reconstructor using a single view or a few views for single-view 3D object reconstruction. In this case, the ambiguity in shapes in training is not negligible. The upper half of Figure 1 shows the result of single-view 3D reconstruction using a conventional approach [18]. Although this method originally uses multiple views per object for training, a single view is used in this experiment. As a result, the reconstructed shape looks

	Work	Class A	Class B
(a)	[13, 37]	Predicted 3D shapes	Their corresponding ground truth 3D shapes
(b)	[9, 34]	Predicted 3D shapes	3D shape collections
(c)	Ours	Views of predicted 3D shapes from observed viewpoints	Views of predicted 3D shapes from random viewpoints
(d)	-	Views of predicted 3D shapes	Views in a training dataset

Table 1. Summary of discriminators in learning-based 3D reconstruction. Discriminator (d) is described in Section 3.2.

correct when viewed from the same viewpoint as the input image, however, it looks incorrect from other viewpoints. This is because the reconstructor is unaware of unobserved views and generates shapes that only fit the observed views.

How can a reconstructor overcome shape ambiguity and correctly estimate shapes? The hint is shown in Figure 1. Humans can recognize that the three views of the chair in the upper-right of Figure 1 are incorrect because we have prior knowledge of how a chair looks, having seen many chairs in the past. If machines also have knowledge regarding the correct views, they would use it to estimate shapes more accurately.

We implement this idea on machines by using a discriminator and adversarial training [7]. One can see from the upper half of Figure 1 that, with the conventional method, views of estimated shapes from observed viewpoints converge to the correct views, while unobserved views do not always become correct. Therefore, we train the discriminator to distinguish the observed views of estimated shapes from the unobserved views. This results in the discriminator obtaining knowledge regarding the correct views. By training the reconstructor to fool the discriminator, reconstructed shapes from all viewpoints become indistinguishable and to be viewed as reasonable from any viewpoint. The lower half of Figure 1 shows the results from the proposed method.

Learning prior knowledge of 3D shapes using 3D models was tackled in other publications [9, 34]. In contrast, we focus on prior knowledge of 2D views rather than 3D shapes. Because our method does not require any 3D models for training, ours can scale to various categories where 3D models are difficult to obtain.

The major contributions can be summarized as follows.

- In view-based training of single-view 3D reconstruction, we propose a method to predict shapes which are viewed as reasonable from any viewpoint by learning prior knowledge of object views using a discriminator. Our method does not require any 3D models for training.
- We conducted experiments on both synthetic and natural image datasets and we observed a significant performance improvement for both datasets. The advantages and limitations of the method are also examined via extensive experimentation.

2. Related work

A simple and popular approach for learning-based 3D reconstruction is to use 3D annotations. Recent studies focus on integrating multiple views [3, 16], memory efficiency problem of voxels [30], point cloud generation [5], mesh generation [8, 32], advanced loss functions [13], and neural network architectures [27].

To reduce the cost of 3D annotation, view-based training has recently become an active research area. The key of training is to define a differentiable loss function for view reconstruction. A loss function of silhouettes using chamfer distance [17], differentiable projection of voxels [31, 33, 36, 38], point clouds [11, 23], and meshes [18] is proposed. Instead of using view reconstruction, 3D shapes can be reconstructed via view synthesis [29].

As mentioned in the previous section, it is not easy to train reconstructors using a small number of views. For this problem, some methods use human knowledge of shapes as regularizers or constraints. For example, the graph Laplacian of meshes was regularized [15, 32], and shapes were assumed to be symmetric [15]. Instead of using manually-designed constraints, others attempted to acquire prior knowledge of shapes from data. Learning category-specific mean shapes [15, 17] is an example. Adversarial training is another way to learn shape priors. Yang *et al*. [37] and Jiang *et al*. [13] used discriminators on an estimated shape and its corresponding ground truth shape to make the estimated shapes more realistic. Gwak *et al*. [9] and Wu *et al*. [34] used discriminators on generated shapes and a shape collection. In contrast, our method does not require 3D models to learn prior knowledge. Table 1 lists a summary of these discriminators.

3. View-based training of single-view 3D object reconstructors with view prior learning

In this section, we introduce a simple view-based method to train 3D reconstructors based on [18]. Then, we describe our main technique, called *view prior learning (VPL)*. We also explain a technique to further improve reconstruction accuracy by applying internal pressure to shapes. Figure 2 shows the architecture of our method.

For training, our method requires a dataset that contains single or multiple views of objects, and their silhouette and viewpoint annotations, similar to previous studies [15, 18, 31, 36]. Additionally, ours can also use class

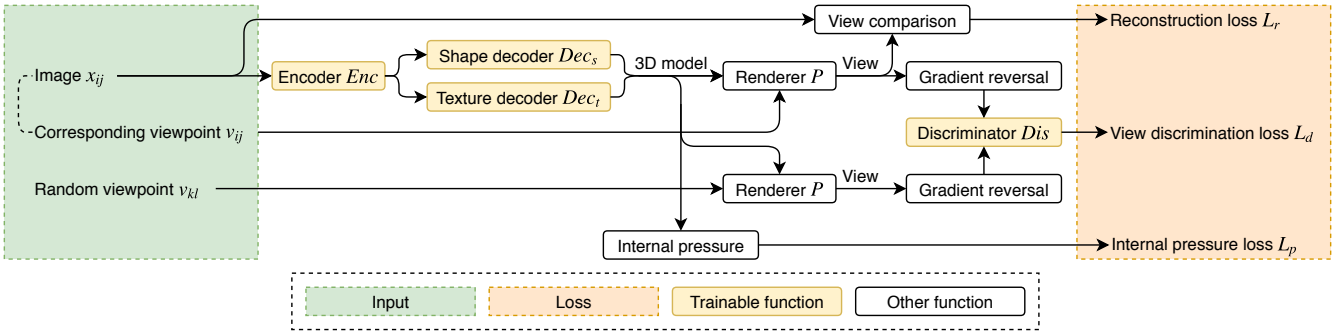

Figure 2. Architecture of the proposed method. The main point of our method is the use of discrimination loss to learn priors of views. While the discriminator aims to minimize discrimination loss, the encoder and decoders try to maximize it using a gradient reversal layer.

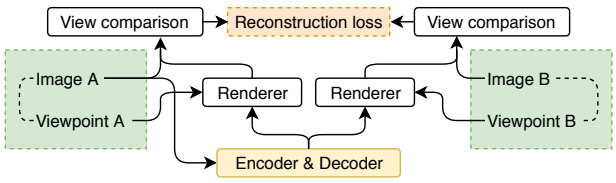

Figure 3. Reconstruction loss in multi-view training. Images A and B are views of the same object. Although only the loss with respect to a view reconstructed from image A is shown in this figure, the loss with respect to image B is also computed.

labels of views if they are available. After training, reconstruction is performed without silhouette, viewpoint, and class label annotations.

3.1. View-based training for 3D reconstruction

In this section, we describe our baseline method for 3D reconstruction. We extend a method which uses silhouettes in training [18] to handle textures using a texture decoder and perceptual loss [14].

Overview. The common approach to view-based training of 3D reconstructors is to minimize the difference between views of a reconstructed shape and views of a ground truth shape. Let x_{ij} be the view of an object o_i from a viewpoint v_{ij}, N_o be the number of objects in the training dataset, N_v be the number of viewpoints per object, $R(\cdot)$ be a reconstructor that takes an image and outputs a 3D model, $P(\cdot, \cdot)$ be a renderer that takes a 3D model and a viewpoint and outputs the view of the given model from the given viewpoint, and $\mathcal{L}_v(\cdot, \cdot)$ be a function that measures the difference between two views. Then, reconstruction loss is defined as

$$\mathcal{L}_r(x, v) = \sum_{i=1}^{N_o} \sum_{j=1}^{N_v} \sum_{k=1}^{N_v} \mathcal{L}_v(P(R(x_{ij}), v_{ik}), x_{ik}). \quad (1)$$

We call the case where $N_v = 1$ *single-view training*. In this case, the reconstruction loss is simplified to $\mathcal{L}_r(x, v) = \sum_{i=1}^{N_o} \mathcal{L}_v(P(R(x_{i1}), v_{i1}), x_{i1})$. We call the case where $2 \leq N_v$ *multi-view training*.

3D representation and renderer. Some works use voxels as a 3D representation in view-based training [31, 36]. However, voxels are not well suited to view-based training because using high-resolution views of voxels is difficult as voxels are memory inefficient. Recently, this problem was overcome by Kato *et al*. [18] by using a mesh as a 3D representation and a differentiable mesh renderer. Following this work, we also use a mesh and their renderer[1].

Reconstructor. In this work, a 3D model is represented by a pair of a shape and a texture. Our reconstructor $R(\cdot)$ uses an encoder-decoder architecture. An encoder $Enc(\cdot)$ encodes an input image, and a shape decoder $Dec_s(\cdot)$ and texture decoder $Dec_t(\cdot)$ generate a 3D mesh and a texture image, respectively. Following recent learning-based mesh reconstruction methods [15, 18, 32], we generate a shape by moving the vertices of a pre-defined mesh. Therefore, the output of the shape decoder is the coordinates of the estimated vertices. The details of the encoder and the decoders are described in the supplementary material.

View comparison function. Color images (RGB channels) and silhouettes (alpha channels) are processed separately in $\mathcal{L}_v(\cdot, \cdot)$. Let x and $\hat{x} = P(R(x), v)$ be a ground truth view and an estimated view, x_c, $\hat{x}_c$ be the RGB channels of x, $\hat{x}$, and x_s, $\hat{x}_s$ be the alpha channels of x, $\hat{x}$. The silhouette at the i-th pixel x_{s_i} is set to one if an object exists at the pixel and to zero if the pixel is part of the background. x_s can take a value between zero and one owing to anti-aliasing of the renderer. To compare color images $x_c, \hat{x}_c$, we use perceptual loss $\mathcal{L}_p$ [14] with additional feature normalization. Let $F_m(\cdot)$ be the m-th feature map of N_f maps in a pre-trained CNN for image classification. In addition, let C_m, H_m, W_m be the channel size, height, and width of $F_m(\cdot)$, respectively. Specifically, we use the five

[1] We modified the approximate differentiation of the renderer. Details are described in the supplementary material.

feature maps after convolution layers of AlexNet [20] for $F_m(\cdot)$. Then, using $D_m = C_m H_m W_m$, the perceptual loss is defined as

$$\mathcal{L}_c(\hat{x_c}, x_c) = \sum_{m=1}^{N_f} \frac{1}{D_m} \left| \frac{F_m(\hat{x}_c)}{|F_m(\hat{x}_c)|} - \frac{F_m(x_c)}{|F_m(x_c)|} \right|^2. \quad (2)$$

For silhouettes $x_s, \hat{x}_s$, we use their multi-scale cosine distance. Let x_s^i be an image obtained by down-sampling x_s 2^{i-1} times, and N_s be the number of scales. We define the loss function as

$$\mathcal{L}_s(x_s, \hat{x}_s) = \sum_{i=1}^{N_s} \left(1 - \frac{x_s^i \cdot \hat{x}_s^i}{|x_s^i||\hat{x}_s^i|}\right). \quad (3)$$

We also use negative intersection over union (IoU) of silhouettes, as was used in [18]. Let $\odot$ be an elementwise product. This loss is defined as

$$\mathcal{L}_s(x_s, \hat{x}_s) = 1 - \frac{|x_s \odot \hat{x}_s|_1}{|x_s + \hat{x}_x - x_s \odot \hat{x}_s|_1}. \quad (4)$$

The total reconstruction loss is $\mathcal{L}_v = \mathcal{L}_s + \lambda_c \mathcal{L}_c$. λ_c is a hyper-parameter.

Training. We optimize $R(\cdot)$ using mini-batch gradient descent. Figure 2 shows the architecture of single-view training. In multi-view training, we randomly take two views of an object in one minibatch. The architecture for computing $\mathcal{L}_r$ in this case is shown in Figure 3.

3.2. View prior learning

As described in Section 1, in view-based training, a reconstructor can generate a shape that looks unrealistic from unobserved viewpoints. In order to reconstruct a shape that is viewed as realistic from any viewpoint, it is necessary to (1) learn the difference between correct views and incorrect views, and (2) tell the reconstructor how to modify incorrect views. In view-based training, reconstructed views from observed viewpoints converge to the real views in a training dataset by minimizing the reconstruction loss, and views from unobserved viewpoints do not always become correct. Therefore, the former can be regarded as correct and realistic views, and the latter can be regarded as incorrect and unrealistic views. Based on this assumption, we propose to train a discriminator that distinguishes estimated views at observed viewpoints from estimated views at unobserved viewpoints to learn the correctness of views. The discriminator can pass this knowledge to the reconstructor by back-propagating the gradient of the discrimination loss into the reconstructor via estimated views and shapes as with adversarial training in image generation [7] and domain adaptation [6].

Concretely, let $Dis(\cdot, \cdot)$ be a trainable discriminator that takes a view and its viewpoint and outputs the probability that the view is correct, and $\mathcal{V}$ be the set of all viewpoints in the training dataset. Using cross-entropy, we define *view disrcimination loss* as

$$\mathcal{L}_d(x_{ij}, v_{ij}) = -\log(Dis(P(R(x_{ij}), v_{ij}), v_{ij})) - \sum_{v_u \in \mathcal{V}, v_u \neq v_{ij}} \frac{\log(1 - (Dis(P(R(x_{ij}), v_u), v_u)))}{|\mathcal{V} - 1|}. \quad (5)$$

In minibatch training, we sample one random view for each reconstructed object to compute $\mathcal{L}_d$.

Stability of training. Although adversarial training is generally not stable, training of our proposed method is stable. It is known that training of GANs fails when the discriminator is too strong to be fooled by the generator. This problem is explained from the distinction of the supports of *real* and *fake* samples [1]. However, in our case, it is very difficult to distinguish views correctly in the earlier training stage because view reconstruction is not accurate and views are incorrect from any viewpoint. Even in the later stage, the reconstructor can easily fool the discriminator by slightly breaking the correct views. Therefore, the discriminator cannot be dominant in our method.

Optimization of the reconstructor. The original procedure of adversarial training requires optimizing a discriminator and a generator iteratively [7]. Subsequently, Ganin *et al*. [6] proposed to train a generator using the reversed gradient of discrimination loss. The proposed gradient reversal layer does nothing in the forward pass, although it reverses the sign of gradients and scales them λ_d times in the backward pass. This layer is posed on the right before a discriminator. Because this optimization procedure is not iterative, the training time is shorter than in iterative optimization. Furthermore, we experimentally found that the performance of the gradient reversal and iterative optimization are nearly the same in our problem. Therefore, we use the gradient reversal layer for training the reconstructor.

Image type for the discriminator. The discriminator can take both RGBA images and silhouette images. We give it RGBA images when texture prediction is conducted, otherwise we give it silhouettes.

Class conditioning. In addition, a discriminator can be conditioned on class labels using the conditional GAN [24] framework. When class labels are known, view discrimination becomes easier and the discriminator becomes more reliable. We use the projection discriminator [26] for class conditioning. Note that the test phase does not require class labels even in this case.

Another possible discriminator. We propose to train a discriminator on views of reconstructed shapes at observed and unobserved viewpoints. Another possible approach is

to distinguish reconstructed views from real views in a training dataset. In fact, this discriminator does not work well because generating a view that is difficult to distinguish from the real view is very difficult. This is caused by the limitation of the representation ability of the reconstructor and renderer. Table 1 shows a summary of the discriminators we have explained thus far.

3.3. Internal pressure

One of the most popular methods in multi-view 3D reconstruction is visual hull [21]. In visual hull, a point inside all silhouettes is assumed to be inside the object. In other words, in terms of shape ambiguity, visual hull produces the shape with the largest volume. Following this policy, we inflate the volume of the estimated shapes by giving them internal pressure in order to maximize their volume. Concretely, we add a gradient along the normal of the face for each vertex of a triangle face. Let p_i be one of the vertices of a triangle face, and n be the normal of the face. We add a loss term $\mathcal{L}_p$ that satisfies $\frac{\partial \mathcal{L}_p(p_i)}{\partial p_i} = -n$.

3.4. Summary

In addition to using reconstruction loss $\mathcal{L}_r = \mathcal{L}_s + \lambda_c \mathcal{L}_c$, we propose to use view discrimination loss $\mathcal{L}_d$ to reconstruct realistic views and internal pressure loss $\mathcal{L}_p$ to inflate reconstructed shapes. The total loss is $\mathcal{L} = \mathcal{L}_s + \lambda_c \mathcal{L}_c + \mathcal{L}_d + \lambda_p \mathcal{L}_p$. The hyperparameters of loss weighting are λ_c, λ_p, and λ_d. Because λ_d is used in the gradient reversal layer, it does not appear in $\mathcal{L}$. The entire architecture is shown in Figure 2.

4. Experiments

We tested our proposed view prior learning (VPL) on synthetic and natural image datasets. We conducted an extensive evaluation of our proposed method using a synthetic dataset because it consists of a large number of objects with accurate silhouette and viewpoint annotations.

As a metric of the reconstruction accuracy, we used intersection over union (IoU) of a predicted shape and a ground truth that was used in many previous publications [3, 5, 15, 16, 18, 27, 30, 31, 36]. To fairly compare our results with those in the literature, we computed IoU after converting a mesh into a volume of 32^3 voxels[2].

4.1. Synthetic dataset

As a synthetic dataset, we used ShapeNet [2], a large-scale dataset of 3D CAD models. We use 43,784 objects in thirteen categories from ShapeNet. By using ShapeNet and a renderer, a dataset of views, silhouettes, viewpoints, and ground truth 3D shapes can be synthetically created. We used ground truth 3D shapes only for validation and testing. We used rendered views and train/val/test splits provided by Kar *et al*. [16]. In this dataset, each 3D model is rendered from twenty random viewpoints. Each image has a resolution of 224×224. We augmented the training images by random color channel flipping and horizontal flipping, as was used in [16, 27][3]. We use all or a subset of views for training, and all views were used for testing.

We used Batch Normalization [12] and Spectral Normalization [25] in the discriminator. The parameters were optimized with the Adam optimizer [19]. The architecture of the encoder and decoders, hyperparameters, and optimizers are described in the supplementary material. The hyperparameters were tuned using the validation set. We used Equation 3 as the view comparison function for silhouettes.

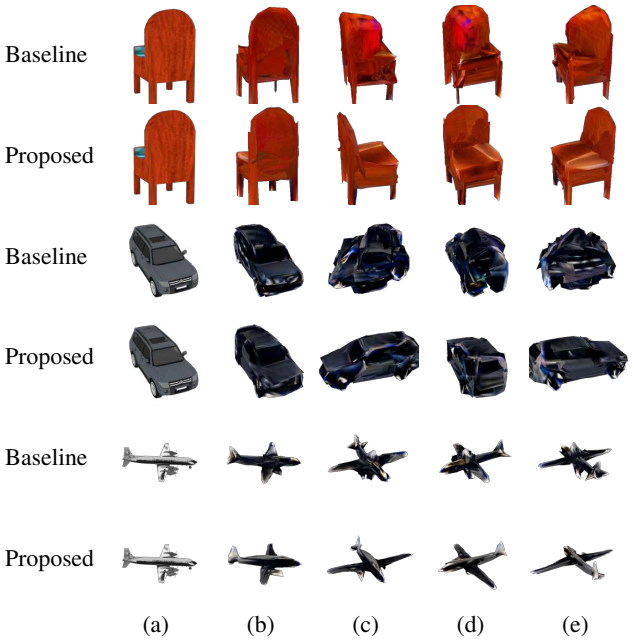

Figure 4. Examples of single-view training on the ShapeNet dataset. (a) Input images. (b) Reconstructed shapes viewed from the original viewpoints. (c–e) Reconstructed shapes viewed from other viewpoints.

4.1.1 Single-view training

At first, we trained reconstructors in single-view training described in Section 3.1. Namely, we used only one randomly selected view out of twenty views for each object in training.

Figure 4 shows examples of reconstructed shapes with and without VPL. When viewed from the original viewpoints (b), the estimated shapes appear valid in all cases. However, without VPL, the shapes appear incorrect when

[2]Another popular metric is the chamfer distance of point clouds. However, this metric is not suitable for use in view-based learning. Because it commonly assumes that points are distributed on surfaces, it is influenced by invisible structures inside shapes, which are impossible to learn in view-based training. This problem does not arise when using IoU because it commonly assumes that the interior of a shape is filled.

[3]When flipping images, we also flip the corresponding viewpoints.

viewed from other viewpoints (c–e). For example, the backrest of the chair is too thick, the car is completely broken, and the airplane has a strange prominence in the center. When VPL is used, the shapes look reasonable from any viewpoint. These results clearly indicate that the discriminator informed the reconstructor regarding knowledge of feasible views.

Table 2 shows a quantitative evaluation of single-view training. VPL provides significantly improved reconstruction performance. This improvement is further boosted when the discriminator is class conditioned. We can tell that conducting texture prediction also helps train accurate reconstructors.

VPL is particularly effective with the *phone*, *display*, *bench*, and *sofa* categories. In contrast, VPL is not effective with the *lamp* category. Typical examples in these categories are shown in the supplementary material. In the case of *phone* and *display* categories, because the silhouettes are very simple, the shapes are ambiguous and various shapes can fit into one view. Although integrating texture prediction reduces the ambiguity, VPL is much more effective. In the case of *bench* and *sofa* categories, learning their long shapes is difficult without considering several views. Because the shapes in the *lamp* category are diverse and the training dataset is relatively small, the discriminator cannot learn meaningful priors.

4.1.2 Multi-view training

Second, we trained reconstructors using multi-view training as described in Section 3.1. Namely, we used two or more views out of twenty views for each object in training.

Table 3 shows the relationship between the reconstruction accuracy and the number of views per object N_v used for training. Texture prediction was not conducted in this experiment, and the difference between the proposed method and the baseline is the use of VPL with class conditioning. Our proposed method outperforms the baseline in all cases, which indicates that VPL is also effective in multi-view training. The effect of VPL increases as N_v decreases, as expected. Figure 5 shows reconstructed shapes with texture prediction when $N_v = 2$. When VPL is used, the shape details become more accurate.

4.1.3 Discriminator and optimization

We discussed two types of discriminators in the last paragraph of Section 3.2 and emphasized the importance of discriminating between estimated views rather than estimated views and real views. We validated this statement with an experiment. We ran experiments in single-view training using the discriminator of Table 1 (d). We also tested the iterative optimization used in GAN [7] instead of using a gradient reversal layer [6]. However, in both cases, we were unable to observe any meaningful improvements from the baseline by tuning λ_d. This fact indicates that the discriminator in Figure 1 (d) does not work well in practice, and discriminating estimated views is key to effective training.

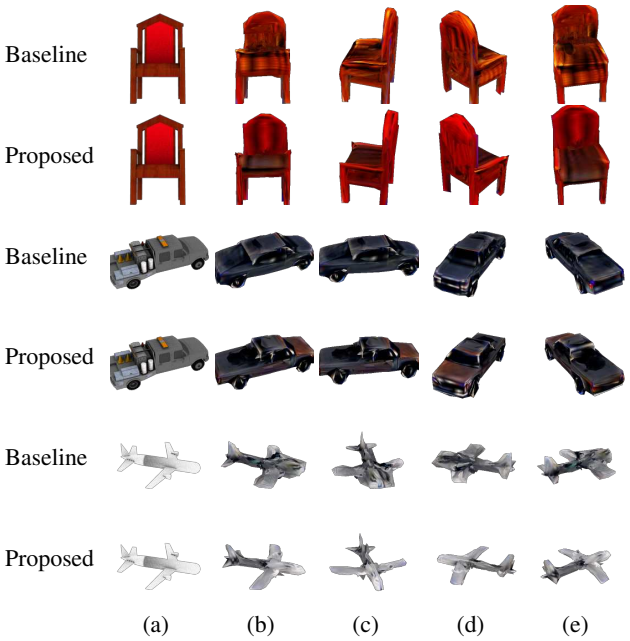

Figure 5. Examples of multi-view training on ShapeNet ($N_v = 2$). Panels (a–e) are the same as in Figure 4.

4.1.4 Comparison with manually-designed priors

Our proposed internal pressure (IP) loss and some regularizers and constraints used in [15, 32] were designed using human knowledge regarding shapes. Table 4 shows a comparison with VPL. This experiment was conducted in single-view training without texture prediction.

This result shows that IP loss improves performance. The symmetricity constraint also improves the performance, however, some objects in ShapeNet are actually not symmetric. By regularizing the graph Laplacian and the edge length of meshes, although the visual quality of the generated meshes became better, improvement of IoU was not observed.

VPL cannot be compared with the learning-based 3D shape priors detailed by Gwak *et al*. [9] and Wu *et al*. [34] because these methods require additional 3D models for training, and their methods are applicable to voxels rather than meshes.

4.1.5 Comparison with state-of-the-arts

Our work also shows the effectiveness of view-based training. Table 5 shows the reconstruction accuracy (IoU) on the ShapeNet dataset using our method and that presented in re-

VPL	CC	TP	airplane	bench	dresser	car	chair	display	lamp	speaker	rifle	sofa	table	phone	vessel	all
			.479	.266	.466	.550	.367	.265	**.454**	.524	.382	.367	.342	.337	.439	.403
✓			.500	.347	.583	.673	.413	.399	.443	.578	.481	.464	.423	.583	.486	.490
✓	✓		.513	.376	**.591**	.701	.444	**.425**	.422	**.596**	.479	.500	.436	.595	.485	.505
		✓	.483	.284	.544	.535	.356	.372	.443	.534	.386	.370	.361	.529	.448	.434
✓		✓	.524	.378	.581	**.705**	.442	.422	.441	.561	.510	.475	.443	**.625**	.490	.508
✓	✓	✓	**.531**	**.385**	**.591**	.701	**.454**	.423	.441	.570	**.521**	**.508**	**.444**	.601	**.498**	**.513**

Table 2. IoU of single-view training on the ShapeNet dataset. VPL: proposed view prior learning. CC: class conditioning in the discriminator. TP: texture prediction.

N_v	2	3	5	10	20
Baseline	.575	.596	.620	.641	.652
Proposed	**.583**	**.600**	**.624**	**.644**	**.655**

Table 3. The relation between the number of views per object N_v and the reconstruction accuracy (IoU) in multi-view training.

Prior	IoU
None	.387
Internal pressure (IP, ours)	.403
IP & Symmetricity [15]	.420
IP & Regularizing graph Laplacian [15, 32]	*.403**
IP & Regularizing edge length [32]	*.403**
IP & View prior learning (ours)	**.505**

Table 4. Comparison of our learning-based prior with manually-designed shape regularizers and constraints. *No meaningful improvement was observed.

	N_v	IoU
Single-view training		
Our best model$^\sharp$	1	**.513**
Multi-view training		
PTN [36]	24	.574
NMR [18]	24	.602
Our best model$^\sharp$	20	**.655**
3D supervision		
3D-R2N2$^\sharp$ [16]	20	.551
3D-R2N2$^\flat$ [3]	24	.560
OGN$^\flat$ [30]	24	.596
LSM$^\sharp$ [16]	20	.615
Matryoshka$^\flat$ [27]	24	.635
PSGN$^\flat$ [5]	24	.640
VTN$^\flat$ [27]	24	**.641**

Table 5. Comparison of our method and state-of-the-art methods on ShapeNet (3D-R2N2) dataset using IoU. Although supervision is weaker, our proposed method outperforms the other models trained using 3D models. $^{\sharp\flat}$Models denoted with the same symbol use the same rendered images.

cent papers[4][5]. Our method outperforms existing view-based training methods [18, 36]. The main differences between our baseline and [18] are the internal pressure and the training dataset. Because the resolution of our training images (224×224) is larger than theirs (64×64) and the elevation range in the viewpoints ($[-20°, 30°]$) is wider than that of theirs ($30°$ only), more accurate and detailed 3D shapes can be learned in our experiments.

It may be surprising that our view-based method outperforms reconstructors trained using 3D models. Although view-based training is currently less popular than 3D-based training, one can say that view-based training has much room for further study.

4.2. Natural image dataset

If a 3D model is available, we can synthetically create multiple views with accurate silhouette and viewpoint annotations. However, in practical applications, it is not always possible to obtain many 3D models, and datasets must be created using natural images. In this case, generally, multi-view training is not possible, and silhouette and viewpoint annotations are noisy. Therefore, to measure the practicality of a given method, it is important to evaluate such a case.

Thus, we used the PASCAL dataset preprocessed by Tulsiani *et al*. [31]. This dataset is composed of images in PASCAL VOC [4], annotations of 3D models, silhouettes, and viewpoints in PASCAL 3D+ [35], and additional images in ImageNet [28] with silhouette and viewpoint annotations automatically created using [22]. We conducted single-view training because there is only one view per object. Because this dataset is not large, the variance in the training results is not negligible. Therefore, we report the mean accuracy from five runs with different random seeds. We used the pre-trained ResNet-18 model [10] as the encoder as with [15, 31]. The parameters were optimized with

[4]The most commonly used dataset of ShapeNet for 3D reconstruction was provided by Choy *et al*. [3]. However, we found that this dataset is not suitable for view-based training because there are large occluded regions in the views owing to the narrow range of elevation in the viewpoints. Therefore, we used a dataset by Kar *et al*. [16], in which images were rendered from a variety of viewpoints. A comparison of the results from both datasets is not so unfair because the performance of 3D-R2N2 [3] is close in both datasets.

[5]This table only compares papers that report IoU on 3D-R2N2 dataset. On other metrics and datasets, some works [8, 32] outperform PSGN [5].

	airplane	car	chair	mean
Category-agnostic models				
DRC [31]	.415	.666	.247	.443
Baseline (s)	.448	.652	.272	.458
Proposed (s)	.450	**.672**	.292	.471
Baseline	.440	.640	.280	.454
Proposed	**.460**	.662	**.296**	**.473**
Category-specific models				
CSDM [17]	.398	.600	.291	.429
CMR [15]	.46	.64	n/a	n/a
Baseline (s)	.449	.679	.289	.472
Proposed (s)	.472	**.689**	.303	**.488**
Baseline	.450	.669	.293	.470
Proposed	**.475**	.679	**.304**	.486

Table 6. IoU of single-view 3D reconstruction on the PASCAL dataset. The difference between the proposed method and the baseline is the use of view prior learning. (s) indicates silhouette only training without texture prediction ($\lambda_c = 0$).

the Adam optimizer [19]. The architecture of decoders, discriminators, and other hyperparameters are described in the supplementary material. We constrained estimated shapes to be symmetric, as was the case in a previous study [15]. We used Equation 4 as the view comparison function for silhouettes.

Table 6 shows the reconstruction accuracy on the PASCAL dataset. Our proposed method consistently outperforms the baseline and provides state-of-the-art performance for this dataset, which validates the effectiveness of our proposed method. Category-specific models outperform category-agnostic models because the object shapes in these three categories are not very similar and multitask learning is not beneficial. The performance difference when texture prediction is used is primarily caused by the relative weight of the internal pressure loss.

Figure 6 shows typical improvements that can be gained using our method. Improvements are prominent on the wings of the airplane, the tires of the car, and the front legs of the chair when viewed from unobserved viewpoints.

In this experiment, internal pressure loss plays an important role because observed viewpoints are not diverse. Figure 7 shows a reconstructed shape without internal pressure. The trunk of the car is hollowed, and this hollow cannot be filled by VPL because there are few images taken from viewpoints such as (c–e) in the dataset.

5. Conclusion

In this work, we proposed a method to learn prior knowledge of views for view-based training of 3D object reconstructors. We verified our approach in single-view training on both synthetic and natural image datasets. We also found that our method is effective, even when multiple views are available for training. The key to our success involves using a discriminator with two estimated views from observed and unobserved viewpoints. Our data-driven method works better than existing manually-designed shape regularizers. We also showed that view-based training works as well as methods that use 3D models for training. The experimental results clearly validate these statements.

Our method significantly improves reconstruction accuracy, especially in single-view training. This is important progress because it is easier to create a single-view dataset than to create a multi-view dataset. This fact may enable 3D reconstruction of diverse objects beyond the existing synthetic datasets. The most important limitation of our method is that it requires silhouette and viewpoint annotations. Training end-to-end 3D reconstruction, viewpoint prediction, and silhouette segmentation would be a promising future direction.

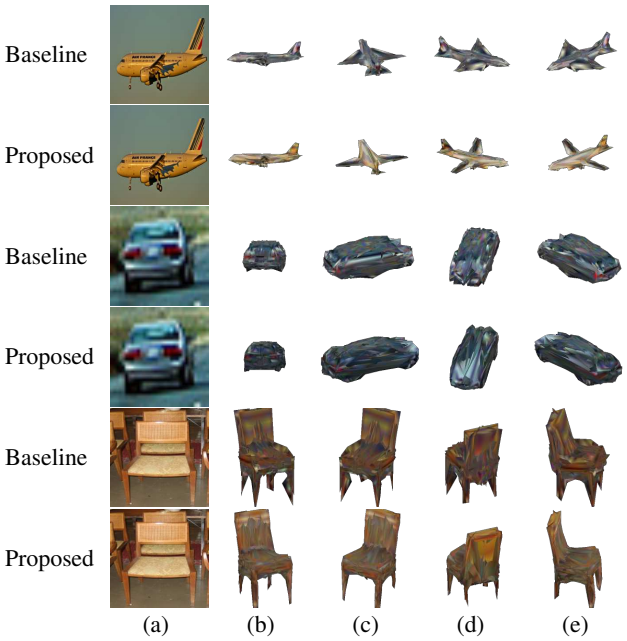

Figure 6. Examples on the PASCAL dataset. Panels (a–e) are the same as in Figure 4.

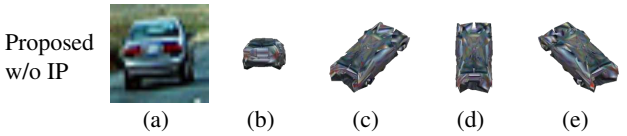

Figure 7. An example of reconstruction without internal pressure (IP). Panels (a–e) are the same as in Figure 4.

Acknowledgment

This work was partially funded by ImPACT Program of Council for Science, Technology and Innovation (Cabinet Office, Government of Japan) and partially supported by JST CREST Grant Number JPMJCR1403, Japan. We would like to thank Antonio Tejero de Pablos, Atsuhiro Noguchi, Kosuke Arase, and Takuhiro Kaneko for helpful discussions.

References

[1] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *ICLR*, 2017. 4

[2] 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*, 2015. 5

[3] 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. 1, 2, 5, 7

[4] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 8

[5] Haoqiang Fan, Hao Su, and Leonidas Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, 2017. 1, 2, 5, 7

[6] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17(1):2096–2030, 2016. 4, 7

[7] 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. 2, 4, 7

[8] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Atlasnet: A papier-m\ˆ ach\'e approach to learning 3d surface generation. In *CVPR*, 2018. 2, 7

[9] JunYoung Gwak, Christopher B Choy, Manmohan Chandraker, Animesh Garg, and Silvio Savarese. Weakly supervised 3d reconstruction with adversarial constraint. In *3DV*, 2017. 1, 2, 7

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 8, 16

[11] Eldar Insafutdinov and Alexey Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. In *NIPS*, 2018. 2

[12] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 5, 17

[13] Li Jiang, Shaoshuai Shi, Xiaojuan Qi, and Jiaya Jia. Gal: Geometric adversarial loss for single-view 3d-object reconstruction. In *ECCV*, 2018. 1, 2

[14] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 3

[15] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018. 1, 2, 3, 5, 7, 8

[16] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *NIPS*, 2017. 1, 2, 5, 7

[17] Abhishek Kar, Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. In *CVPR*, 2015. 1, 2, 8

[18] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *CVPR*, 2018. 1, 2, 3, 4, 5, 7, 13

[19] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5, 8, 16

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 4

[21] Aldo Laurentini. The visual hull concept for silhouette-based image understanding. *PAMI*, 16(2):150–162, 1994. 5

[22] Ke Li, Bharath Hariharan, and Jitendra Malik. Iterative instance segmentation. In *CVPR*, 2016. 8

[23] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *AAAI*, 2018. 2

[24] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv*, 2014. 4

[25] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018. 5, 17

[26] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. In *ICLR*, 2018. 4

[27] Stephan R Richter and Stefan Roth. Matryoshka networks: Predicting 3d geometry via nested shape layers. In *CVPR*, 2018. 1, 2, 5, 7

[28] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 8

[29] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Multi-view 3d models from single images with a convolutional network. In *ECCV*, 2016. 2

[30] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *ICCV*, 2017. 1, 2, 5, 7

[31] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, 2017. 1, 2, 3, 5, 8

[32] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV*, 2018. 2, 3, 7

[33] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, Bill Freeman, and Josh Tenenbaum. Marrnet: 3d shape reconstruction via 2.5 d sketches. In *NIPS*, 2017. 2

[34] Jiajun Wu, Chengkai Zhang, Xiuming Zhang, Zhoutong Zhang, William T Freeman, and Joshua B Tenenbaum. Learning shape priors for single-view 3d completion and reconstruction. In *ECCV*, 2018. 2, 7

[35] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *WACV*, 2014. 8

[36] 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. 1, 2, 3, 5, 7

[37] B. Yang, S. Rosa, A. Markham, N. Trigoni, and H. Wen. Dense 3d object reconstruction from a single depth view. *PAMI*, pages 1–1, 2018. 2

[38] Rui Zhu, Hamed Kiani Galoogahi, Chaoyang Wang, and Simon Lucey. Rethinking reprojection: Closing the loop for pose-aware shape reconstruction from a single image. In *ICCV*, 2017. 2