Learning Single-View 3D Reconstruction with Limited Pose Supervision

Guandao Yang[1](✉), Yin Cui[1,2], Serge Belongie[1,2], and Bharath Hariharan[1]

[1] Department of Computer Science, Cornell University, Ithaca, USA
gy46@cornell.edu
[2] Cornell Tech, New York, USA

Abstract. It is expensive to label images with 3D structure or precise camera pose. Yet, this is precisely the kind of annotation required to train single-view 3D reconstruction models. In contrast, unlabeled images or images with just category labels are easy to acquire, but few current models can use this weak supervision. We present a unified framework that can combine both types of supervision: a small amount of camera pose annotations are used to enforce pose-invariance and view-point consistency, and unlabeled images combined with an adversarial loss are used to enforce the realism of rendered, generated models. We use this unified framework to measure the impact of each form of supervision in three paradigms: semi-supervised, multi-task, and transfer learning. We show that with a combination of these ideas, we can train single-view reconstruction models that improve up to 7 points in performance (AP) when using only 1% pose annotated training data.

Keywords: Single-image 3D-reconstruction · Few-shot learning
GANs

1 Introduction

The ability to understand 3D structure from single images is a hallmark of the human visual system and a crucial step in visual reasoning and interaction. Of course, a single image by itself does not have enough information to allow 3D reconstruction, and a machine vision system must rely on some prior over shape: all cars have wheels, for example. The crucial question is how a machine vision system can acquire such priors.

One possibility is to leverage datasets of 3D shapes [4], but obtaining such a dataset for a wide variety of categories requires either 3D modeling expertise or 3D scanning tools and is therefore expensive. Another option, extensively explored recently [21, 27], is to show the machine many different views of a multitude of objects from calibrated cameras. The machine can then use photometric

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-01267-0_6) contains supplementary material, which is available to authorized users.

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11219, pp. 90–105, 2018.
https://doi.org/10.1007/978-3-030-01267-0_6

consistency between rendered views of hypothesized shape and the corresponding view of the real object as a learning signal. Although more tractable than collecting 3D models, this approach is still very expensive in practice: one needs to either physically acquire *thousands* of objects and place them on a turntable, or ask human annotators to annotate images in the wild with both the camera parameters and the precise *instance* that the image depicts. The assumption that *multiple*, *calibrated* views of *thousands* of objects are available is also biologically implausible: a human infant must physically interact with objects to acquire such training data, but most humans can understand airplane shape very easily despite having played with very few airplanes.

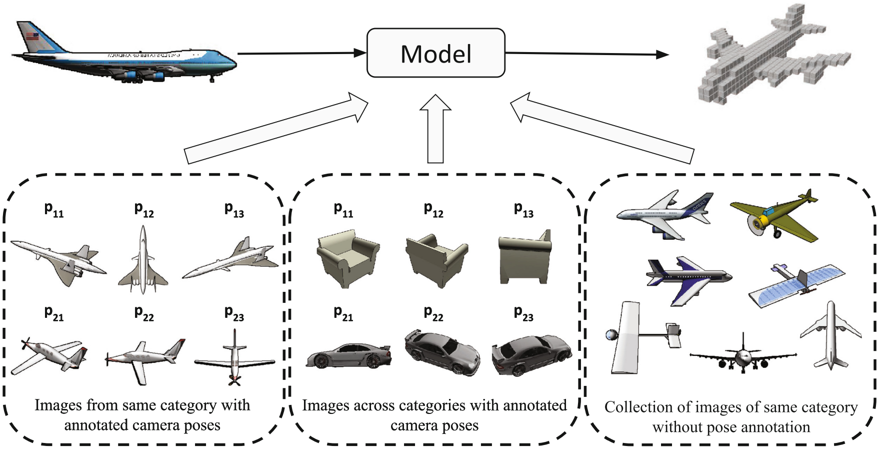
Model
p_{11}
p_{12}
p_{13}
p_{21}
p_{22}
p_{23}
Images from same category with annotated camera poses
p_{11}
p_{12}
p_{13}
p_{21}
p_{22}
p_{23}
Images across categories with annotated camera poses
Collection of images of same category without pose annotation

Fig. 1. We propose a unified framework for single-view 3D reconstruction. Our model can be trained with different types of data, including pose-annotated images from the same object category or across multiple categories, and unlabeled images.

Our goal in this paper is to learn effective single-view 3D reconstruction models when calibrated multi-view images are available for very few objects. To do so we look at two additional sources of information. First, what if we had a large collection of images of a category but without any annotation of the precise instance or pose? Such a dataset is easy to acquire by simply downloading images of this category from the web (Fig. 1, lower right). While it might be hard to extract 3D information from such images, they can capture the distribution of the visual appearance of objects from this category. Second, we look at annotations from other semantic classes (Fig. 1, lower middle). These other classes might not tell us about the nuances of a particular class, but they can still help delineate what *shapes in general* look like. For example, most shapes are compact, smooth, tend to be convex, etc.

This paper presents a framework that can effectively use all these sources of information. First, we design a unified model architecture and loss functions that combine pose supervision with weaker supervision from unlabeled images. Then, we use our model and training framework to evaluate and compare many training paradigms and forms of supervision to come up with the best way of using a small number of pose annotations effectively. In particular, we show that:

1. Images without instance or pose annotations are indeed useful and can provide significant gains in performance (up to 5 points in AP). At the same time a little bit of pose supervision (<50 objects) gives a large gain (>20 points AP) when compared to not using pose information at all.
2. Category-agnostic priors obtained by pooling training data across classes work just as well as, but not better than, category-specific priors trained on each class individually.
3. *Fine-tuning* category-agnostic models for a novel semantic class using a small amount (i.e. only 1%) of pose supervision significantly improves performance (up to 7 points in AP).
4. When faced with a novel category with nothing but a tiny set of pose-annotated images, a category-agnostic model trained on pooled data and fine-tuned on the category of interest outperforms a baseline trained on only the novel category by an enormous margin (up to 20 points in AP).

In summary, our results convincingly show large accuracy gains to be accrued from combining multiple sources of data (unlabeled or labeled from different classes) with a single unified model.

2 Related Work

Despite many successes in reconstructing 3D scenes from multiple images [1,22], doing it on a single image remains challenging. Classic work on single-image 3D reconstruction relies on having access to images labeled with the 3D structure [19]. This is also true for many recent deep learning approaches [5,6,8,18,24]. To get away from this need for precise 3D models, some work leverages keypoint and silhouette annotations [12,23]. More recent approaches assume multiple views with calibrated cameras for training [10,17,21,27], and design training loss functions that leverage photometric consistency and/or enforce invariance to pose. Among these, our encoder-decoder architecture is similar to the one proposed in PTN [27], but our model is trained end-to-end and is additionally able to leverage unlabeled images to deal with limited supervision. In terms of the required supervision, Tulsiani et al. [20] remove the requirement for *pose* annotations but still require images to be annotated with the *instance* they correspond to. PrGAN [7] reduces the supervision requirement further by *only* using unlabeled images. As we show in this paper, this makes the problem needlessly challenging, while adding small amounts of pose supervision leads to large accuracy gains.

Recovering 3D structure from a single image requires strong priors about shape, and another line of work has focused on better capturing the manifold of shape. Classic work has used low-dimensional parametric models [3,12]. More recently, the rediscovery of convolutional networks has led to a resurgence in interest in deep generative models. Wu et al. [26] used deep belief nets to model 3D shapes while Rezende et al. [17] consider variants of variational autoencoders. Generative adversarial networks or GANs [9] can also be used to build generative models of shapes [25]. The challenge is to train them *without* 3D data: Gadelha et al. [7] show that this is indeed possible. While we use an adversarial loss as they suggest, our generator is trained jointly with an encoder end-to-end on a combination of pose-supervised and unlabeled images.

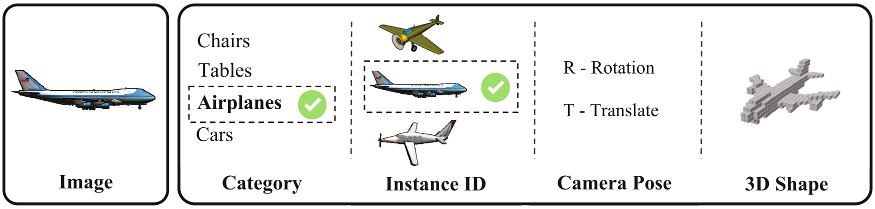

Fig. 2. Different forms of training annotations for single-view 3D reconstruction. Note that some annotations (e.g. category) are cheaper to obtain than others (e.g. 3D shapes); and conversely some offer a better training signal than others.

3 Training Paradigms

For single-view 3D reconstruction, we consider four types of annotations for an image as illustrated in Fig. 2. Our goal is to minimize the need for the more expensive annotations (instance ID, camera pose and 3D shape). Towards this end, we look at three different training paradigms.

3.1 Semi-supervised Single-Category

In this setting, we assume all images are from a single category. Noting the fact that camera pose and model-instance annotations are difficult to collect in the wild, we restrict to a semi-supervised setting where only some of the images are labeled with camera pose and most of them are unlabeled. Formally, we are given a dataset of images annotated with both camera pose and the instance ID: $\mathcal{X}_l = \{(\mathbf{x}_{ij}, \mathbf{p}_{ij}, i)\}_{i,j}$, where $\mathbf{x}_{ij}$ represents the j-th image of the i-th instance when projected with camera pose $\mathbf{p}_{ij}$. We also have a dataset without any annotation: $\mathcal{X}_u = \{\mathbf{x}_i\}_i$. The goal is to use $\mathcal{X}_l$ and $\mathcal{X}_u$ to learn a category-specific model for single image 3D reconstruction.

3.2 Semi-supervised Multi-category

An alternative to building a separate model for each category is to build a category-agnostic model. This allows one to combine training data across multiple categories, and even use training images that do not have any category labels. Thus, instead of a separate labeled training set $\mathcal{X}_l^c$ for each category c, here we only assume a combined dataset $\mathcal{X}_l^{multi} = \mathcal{X}_l^{c_1} \cup \mathcal{X}_l^{c_2} \cup \cdots \cup \mathcal{X}_l^{c_n}$. Similarly, we assume access to an unlabeled set of images $\mathcal{X}_u^{multi}$ (now without category labels). Note that this multi-category setting is harder than the single-category since it introduces cross-category confusion, but it also allows the model to learn category-agnostic shape information across different categories.

3.3 Few-Shot Transfer Learning

Collecting a large dataset that can cover all categories we would ever encounter is infeasible. Therefore, we also need a way to adapt a pre-trained model to a new category. This strategy can also be used for adapting a *category-agnostic* model to a specific category. We assume that for this adaptation, a dataset $\mathcal{X}_l^{(new)}$ containing a very small number of images with pose and instance annotations ($<$100) are available for the category of interest. We also assume that the semi-supervised multi-category dataset described above is available as a pre-training dataset: $\mathcal{X}_l^{pre} = \mathcal{X}_l^{multi}$ and $\mathcal{X}_u^{pre} = \mathcal{X}_u^{multi}$.

4 A Unified Framework

We need a model and a training framework that can utilize both images with pose and instance annotations, and images without any labels. The former set of images can be used to enforce the consistency of the predicted 3D shape across views, as well as the similarity between the rendered 3D shape and the corresponding view of the real object. The latter set of images can only provide constraints on the realism of the generated shapes. To capture all these constraints, we propose a unified model architecture with three main components:

1. An **Encoder** E that takes an image (silhouette) as input and produces a latent representation of shape.
2. A **Generator** G that takes a latent representation of shape as input and produces a voxel grid.
3. A **Discriminator** D that tries to distinguish between rendered views of the voxel output by the generator and views of the real objects.

In addition, we make use of a "projector" module P that takes a voxel and a viewpoint as input, and it renders the voxel from the inputted viewpoint. We use a differentiable projector similar to the one in PrGAN [7]. We extend it to perspective projection. P has no trainable parameters.

The training process alternates between an iteration on images labeled with pose and instance, and an iteration on unlabeled images. The two sets of iterations use different loss functions but update the same model.

4.1 Training on Pose-Annotated Images

In each pass on the annotated images, the encoder is provided with pairs of images x_{i1}, x_{i2} of the same 3D object i taken from different camera poses $\mathbf{p}_1$ and $\mathbf{p}_2$. The encoder E embeds each image into latent vectors $\mathbf{z}_1$, $\mathbf{z}_2$. The generator (decoder) G is tasked with predicting the 3D voxel grid from $\mathbf{z}_1$ and $\mathbf{z}_2$.

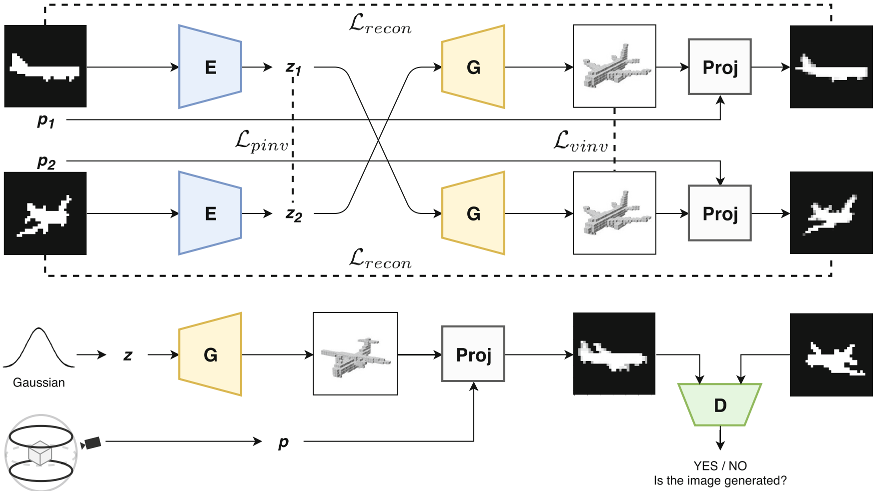

Fig. 3. Overview of the proposed model architecture. An encoder E and a generator G with pose consistency (on the top) learn from images with pose supervision, and a discriminator D (on the bottom) helps G to learn from unlabeled images. Notice that two encoders E and three generator G in the diagram all share parameters, respectively.

The 3D voxel grid produced by the generator should be: (1) a good reconstruction of the object and (2) invariant to the pose of the input image [27]. This requires that the latent shape representation also be invariant to the camera pose of the input image. To ensure the pose invariance of the learned latent representation $\mathbf{z}_1$, the predicted 3D voxel from $\mathbf{z}_1$ should be able to reconstruct the second input image when projected to the second viewpoint $\mathbf{p}_2$, and vice versa.

With these intuitions in mind, we explore the following three losses.

Reconstruction Loss: The predicted 3D model, when projected with a certain camera pose, should be consistent with the ground truth image projected from that camera pose. More specifically, let $(\mathbf{x}_1, \mathbf{p}_1)$ and $(\mathbf{x}_2, \mathbf{p}_2)$ be two pairs of image-pose pair sampled from a 3D-model, then the voxel reconstructed from $E(\mathbf{x}_1)$ should produce the same image as $\mathbf{x}_2$ if projected from camera pose $\mathbf{p}_2$. Same for the other view. Let $P(\mathbf{v}, \mathbf{p})$ represent the image generated by projecting voxel $\mathbf{v}$ using camera pose $\mathbf{p}$. We define the reconstruction loss to address this consistency requirement as:

$$\mathcal{L}_{recon} = \|P(G(E(\mathbf{x}_2)), \mathbf{p}_1) - \mathbf{x}_1\|_{1+2} + \|P(G(E(\mathbf{x}_1)), \mathbf{p}_2) - \mathbf{x}_2\|_{1+2} \quad (1)$$

where $\|\cdot\|_{1+2} = \|\cdot\|_1 + \|\cdot\|_2$ is the summation of ℓ_1 and ℓ_2 reconstruction losses. Such reconstruction loss has been used in prior work [27]. We add ℓ_1 loss since ℓ_1 loss could better cope with sparse vectors such as silhouette images.

Pose-Invariance Loss on Representations: Given two randomly sampled views of an object, the encoder E should be able to embed their latent representations close by, irrespective of pose. Therefore, we define a *pose-invariance* loss on the latent representations:

$$\mathcal{L}_{pinv} = \|E(\mathbf{x}_1) - E(\mathbf{x}_2)\|_2 \quad (2)$$

Pose-Invariance Loss on Voxels: Similarly, the 3D voxel output reconstructed by the generator G from two different views of the same object should be the same. Thus, we introduce a *voxel-based* pose invariance loss:

$$\mathcal{L}_{vinv} = \|G(E(\mathbf{x}_1)) - G(E(\mathbf{x}_2))\|_1 \quad (3)$$

Losses are illustrated by the dashed lines in Fig. 3. Each training step on the images with pose annotations tries to minimize the combined supervised loss:

$$\mathcal{L}_{supervised} = \mathcal{L}_{recon} + \alpha\mathcal{L}_{pinv} + \beta\mathcal{L}_{vinv} \quad (4)$$

where α and β are weights for $\mathcal{L}_{pinv}$ and $\mathcal{L}_{vinv}$, respectively. We use $\alpha = \beta = 0.1$ in all of our experiments.

4.2 Training on Unlabeled Images

In order to learn from unlabeled images, we use an adversarial loss, as illustrated in the bottom of Fig. 3. The intuition is to let the generator G learn to *generate* 3D voxel grids. When projected from a random viewpoint, the 3D voxel grid should be able to produce an image that is indistinguishable from a real image. Another advantage of an adversarial loss is regularization, as in the McRecon approach [10]. Specifically, we first sample a vector $\mathbf{z} \sim \mathcal{N}(0, I)$ and a viewpoint $\mathbf{p}$ uniformly sampled from the range of camera poses observed in the training set. Then the generator G will take the latent vector $\mathbf{z}$ and reconstruct a 3D shape. This 3D shape will be projected to an image using the random pose $\mathbf{p}$. No matter which camera pose we project, the projected image should look like an image sampled from the dataset. We update the generator and discriminator by using an adversarial loss similar to the one used by PrGAN [7]:

$$\mathcal{L}_D = \mathbb{E}_{\mathbf{z},\mathbf{p}}[\log(1 - D(P(G(\mathbf{z}), \mathbf{p})))] + \mathbb{E}_{\mathbf{x}\sim\mathcal{X}}[\log D(\mathbf{x})] \quad (5)$$

$$\mathcal{L}_G = -\mathbb{E}_{\mathbf{z},\mathbf{p}}[\log D(P(G(\mathbf{z}), \mathbf{p}))] \quad (6)$$

Note that instead of normally distributed $\mathbf{z}$ vectors, one could also use the encoder output on sampled training images. However, encouraging G to produce meaningful shapes even on noise input might force G to capture shape priors.

4.3 Implementation Details

The detailed architectures of encoder, generator and discriminator are illustrated in Fig. 4. In the projector (not shown in Fig. 4), we first rotate the voxelized 3D model by its center and then use perspective projection to produce the image according to the camera pose. The whole model is trained end-to-end by alternating between iterations on pose-annotated and unlabeled images. We use Adam optimizer [13] with learning rates of 10^{-3}, 10^{-4}, and 10^{-4} for encoder, generator, and discriminator respectively. While training using the adversarial loss, we used the gradient penalty introduced by DRAGAN [14] to improve training stability. Codes are available at https://github.com/stevenygd/3d-recon.

5 Experiments

5.1 Dataset

We use voxelized $32 \times 32 \times 32$ 3D shapes from the ShapeNetCore [4] dataset. We look at 10 categories: `airplanes`, `cars`, `chairs`, `displays`, `phones`, `speakers`, `tables`, `benches`, `vessels`, and `cabinets`. For each category, we use ShapeNet's default split for training, validation, and test. While generating the training images, we first rotate the voxelized 3D model around its center using a rotation vector $\mathbf{r} = [r_x, r_y, r_z]$, where $r_x \in [-20°, 40°]$ and $r_y \in [0°, 360°]$ are uniformly sampled rotation angles of Altitude and Azimuth; we always set $r_z = 0$. We then project the rotated 3D voxel into a binary mask as the image for training, validation, and testing, where the rotation vector $\mathbf{r}$ is the camera pose. For each 3D shape, we generate 5 masks from different camera poses. During the

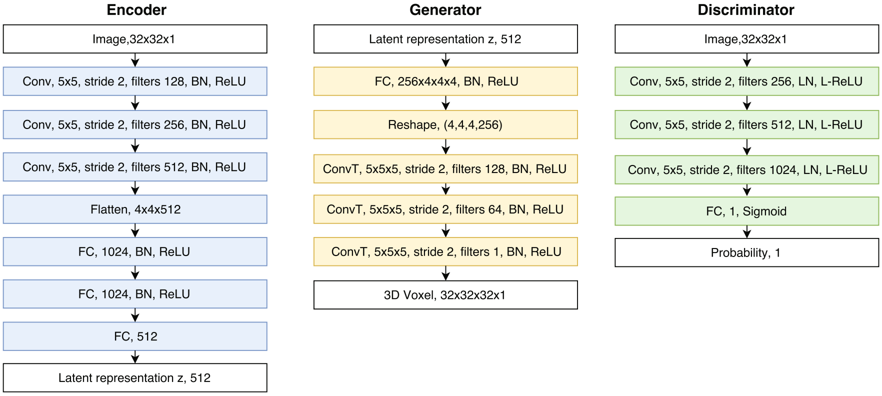
Encoder
Image,32x32x1
Conv, 5x5, stride 2, filters 128, BN, ReLU
Conv, 5x5, stride 2, filters 256, BN, ReLU
Conv, 5x5, stride 2, filters 512, BN, ReLU
Flatten, 4x4x512
FC, 1024, BN, ReLU
FC, 1024, BN, ReLU
FC, 512
Latent representation z, 512
Generator
Latent representation z, 512
FC, 256x4x4x4, BN, ReLU
Reshape, (4,4,4,256)
ConvT, 5x5x5, stride 2, filters 128, BN, ReLU
ConvT, 5x5x5, stride 2, filters 64, BN, ReLU
ConvT, 5x5x5, stride 2, filters 1, BN, ReLU
3D Voxel, 32x32x32x1
Discriminator
Image,32x32x1
Conv, 5x5, stride 2, filters 256, LN, L-ReLU
Conv, 5x5, stride 2, filters 512, LN, L-ReLU
Conv, 5x5, stride 2, filters 1024, LN, L-ReLU
FC, 1, Sigmoid
Probability, 1

Fig. 4. Model Architectures of encoder, generator and discriminator. `Conv`: Convolution, `BN`: Batch Normalization [11], `LN`: Layer Normalization [2], `L-ReLU`: leaky ReLU with slope of 0.2 [15], `ConvT`: Transposed Convolution that is often used in generation tasks [16,25]. `FC,` k: Fully-Connected layers with k outputs. The discriminator outputs a probability that the image is generated.

experiments, we also want to restrict the amount of pose supervision. A model is trained with $r\%$ of pose supervision if $r\%$ of model instances are annotated with poses. We will explore 100%, 50%, 10%, and 1% of pose annotations in different settings. All training images, no matter whether they have pose annotations or not, are used as unlabeled images in all settings.

Table 1. Comparison between synthetic datasets used in prior work and ours. The key difference is the amount of pose annotations available during training. We experiment with multiple settings.

Dataset properties	MVC [20]	McRecon [10]	PTN [27]	Ours
Input image	64 × 64/RGB	127 × 127/RGB	64 × 64/RGB	32 × 32/Grayscale
Supervision image	64 × 64/Mask	127 × 127/Mask	32 × 32/Mask	32 × 32/Mask
Supervision level	2D	2D + U3D	2D	2D
Pose annotations	100%	100%	100%	**0–100%**
#views per image	5	Unavailable	8–24	5
Pose selection	Random	Random	Fixed discrete	Random

Note that our data settings are different from prior work, and indeed the settings in prior works differ from each other. We use input images with the lowest resolution (32 × 32) and no color cues (grayscale) compared to the synthetic dataset from Tulsiani *et al.* [20], McRecon [10], and PTN [27]. We use fewer viewpoints than PTN [27], and our viewpoints are sampled randomly, making it a more difficult task. Our data setting only provides 2D supervision with camera pose, which is different from McRecon [10] that also used unlabeled 3D supervision (U3D). The precise data setting is orthogonal to our focus, which is on combining pose supervision and unlabeled images. As such, we select the setting with less information provided when compared to prior works. A detailed comparison is presented in Table 1.

5.2 Evaluation Metrics

To evaluate the performance of our model, we use the Intersection-over-Union (IoU) between the ground truth voxel grid and the predicted one, averaged over all objects. Computing the IoU requires thresholding the probability output of voxels from the generator. As suggested by Tulsiani *et al.* [20], we sweep over thresholds and report the maximum average IoU. We also report $\text{IoU}_{0.4}$ and $\text{IoU}_{0.5}$ for comparison with previous work, and the Average Precision (AP).

5.3 Semi-supervised Single-Category

We use 6 categories: `airplanes`, `benches`, `cars`, `chairs`, `sofas`, and `tables` for single-category experiments under semi-supervised setting. In this setting, we train a separate model for each category. We experiment with varying amounts of pose supervision from 0% to 100%.

Comparison with Prior Work: We first compare with prior work that uses full pose/instance supervision. We train our models with 50% of the images annotated with instance and pose. The models are trained for 20,000 iterations with early stopping (i.e., keeping the model with the best performance in validation set). Performance comparisons are shown in Table 2. The performance of our model is comparable with prior work across multiple metrics. The results suggest that while using only 50% of pose supervisions, our model outperforms McRecon [10] and MVC [20], but it performs worse than PTN [27] in terms of IoU$_{0.5}$. However, note that due to differences in the setting across different approaches, the numbers are not exactly commensurate.

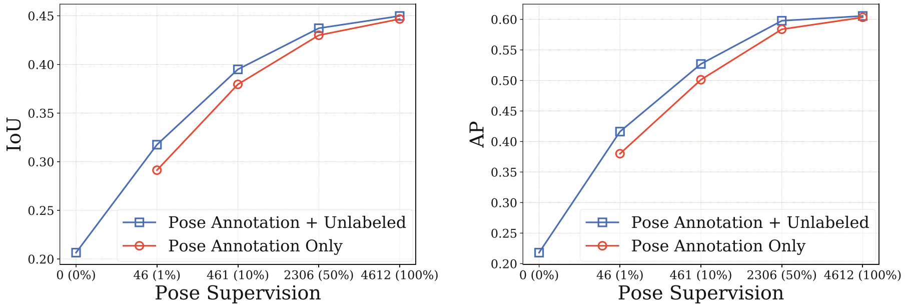

Fig. 5. Comparison between three variations of our models trained with: (1) combined pose-annotated and unlabeled images, (2) pose-annotated images only, and (3) unlabeled images only. Our model is able to leverage both data with pose annotation and unlabeled data. Unlabeled data is especially helpful in the case of limited supervision.

Table 2. Comparison between our model and prior work on single-view 3D reconstruction. All models are trained with images from a single category. Our model's performance is comparable with prior models while using only 50% pose supervision.

Category	MVC [20]	McRecon [10]		PTN [27]	Ours (50% pose annotations)			
	IoU	AP	IoU$_{0.4}$	IoU$_{0.5}$	IoU	AP	IoU$_{0.4}$	IoU$_{0.5}$
airplanes	0.55	0.59	0.37	–	**0.57**	**0.75**	**0.56**	0.57
benches	–	0.39	0.30	–	0.36	**0.48**	**0.35**	0.35
cars	0.75	0.82	0.56	–	**0.78**	**0.92**	**0.77**	0.77
chairs	0.42	0.48	0.35	**0.49**	**0.44**	**0.60**	**0.43**	0.42
sofas	–	0.56	0.38	–	0.54	**0.69**	**0.53**	0.52
tables	–	0.46	0.35	–	0.44	**0.63**	**0.43**	0.42

Are unlabeled images useful? We next ask if using unlabeled images and an adversarial loss to provide additional supervision and regularization is useful. We compare three models: (1) a model trained with both pose-annotated and unlabeled images; (2) a model trained on just the pose-annotated images; and (3) a model trained on only the unlabeled images. In the third case, since the model doesn't have data to train the encoder, we adopt the training scheme of PrGAN [7] by first training the generator G and discriminator D together as a GAN, and then using the generator to train an encoder E once the GAN training is done. We compare these models on the `chair` category with different amounts of pose supervision. Results are presented in Fig. 5.

First, compared to the purely unsupervised approach (0 pose supervision), when only 1% of the data has pose annotations (45 models, 225 images), performance increases significantly. This suggests that pose supervision is necessary, and that our model could successfully leverage such supervision to make better predictions. Second, the model that combines pose annotations with unlabeled images outperforms the one that uses only pose-annotated images. The lesser the pose annotation available, the larger the gain, indicating that an adversarial loss on unlabeled images is useful especially in the case when pose supervisions and viewpoints are limited ($\leq$10%). Third, given enough pose supervision (50% or even 100%), the performance gap between the pose-supervision-only model and the combined model is greatly reduced. This suggests that when there are enough images with pose annotations, leveraging unlabeled data is unnecessary.

5.4 Semi-supervised Multi-category

We next experiment with a *category-agnostic* model on combined training data from 7 categories: `airplanes`, `cars`, `chairs`, `displays`, `phones`, `speakers`, and `tables`. This experiment is also conducted with different amount of pose annotations. Results are reported in Table 3.

In general, using more pose supervision yields better performance of category-agnostic model. With the same amount of pose supervision (50%) for each category, the category-agnostic model achieves similar performance compared with the category-specific models. This suggests that the model is able to remedy the removal of category information by learning a category-agnostic representation.

5.5 Few-Shot Transfer Learning

What happens when a new class comes along that the system has not seen before? In this case, the model should be able to *transfer* the knowledge it has acquired and adapt it to the new class with very limited annotated training data.

To evaluate if this is possible, we use the category-agnostic model, pre-trained on the dataset described in Sect. 5.4, and adapt it to three *unseen* categories: `benches`, `vessels`, and `cabinets`. For each of the novel categories, only 1% of the pose-annotated data is provided. As a result, each novel category usually has about 13 3D-shapes or about 65 pose-annotated images.

Table 3. Performance of category-agnostic models under different amount of pose supervision. Using same amount of supervision (50%), the performance of category-agnostic model is on par with its category-specific counterpart, indicating that we don't need category supervision.

Test categories	Pose supervision and problem setting									
	50% single		100% multi		50% multi		10% multi		1% multi	
	IoU	AP	IoU	AP	IoU	AP	IoU	AP	IoU	AP
Airplanes	0.57	0.75	0.58	0.76	0.57	0.73	0.54	0.75	0.49	0.63
Cars	0.78	0.92	0.79	0.93	0.78	0.93	0.78	0.93	0.71	0.81
Chairs	0.44	0.60	0.45	0.57	0.44	0.57	0.41	0.54	0.31	0.39
Displays	0.44	0.61	0.43	0.59	0.43	0.58	0.36	0.49	0.26	0.32
Phones	0.55	0.69	0.55	0.72	0.56	0.73	0.50	0.64	0.42	0.52
Speakers	0.58	0.73	0.59	0.74	0.59	0.73	0.55	0.69	0.45	0.58
Tables	0.44	0.63	0.46	0.63	0.45	0.61	0.40	0.54	0.29	0.39
Mean	0.54	0.70	0.55	0.71	0.55	0.70	0.51	0.65	0.42	0.52

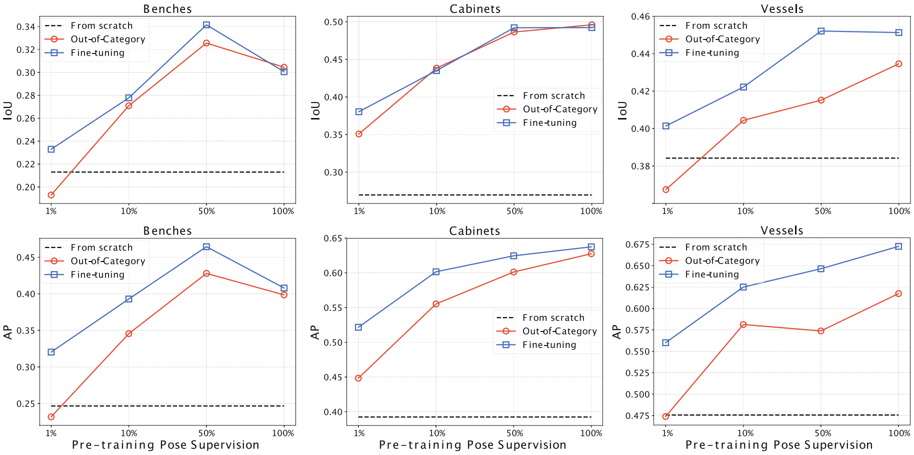

Fig. 6. Few-shot transfer learning on novel categories. Each column represents the performance on a novel category (IoU in top row and AP in bottom row). Notice that the horizontal axis shows the amount of pose annotated supervision in *pre-training*.

We compare three models in this experiment. **From scratch**: a model trained from scratch on the given novel category without using any pre-training; **Out-of-Category** [27]: the pre-trained category-agnostic model directly applied on the novel classes without any additional training; and **Fine-tuning**: a pre-trained category-agnostic model fine-tuned on the given novel category. The fine-tuning is done by fixing the encoder and training the generator only using pose-annotated images for a few iterations. We used the same training strat-

Table 4. Comparing different training strategies on `chairs` with 1% pose annotations. Fine-tuning a category-agnostic model on the target category works the best.

	S, P	**S, U**	**S, P+U**	**M**	**FT**
IoU	0.2913	0.2065	0.3175	0.3104	**0.3250**
AP	0.3800	0.2180	0.4162	0.3859	**0.4247**

egy as mentioned in Sect. 4.3 for all three models. In this experiment, we varies the amount of pose annotations used for pre-training. The results are shown in Fig. 6.

First, we observe that fine-tuning a pre-trained model for a novel category performs much better than training from scratch without pre-training. This suggests that transferring the knowledge learned from a pre-trained model is essential for few-shot learning on new categories. Second, compared with the out-of-category baseline, fine-tuning improves the performance a lot upon directly using the pre-trained model, especially in the case of limited pose supervision. This indicates that our model is able to quickly adapt to a novel category with few training examples via fine-tuning.

5.6 How Best to Use Limited Annotation?

We now have all the ingredients necessary to answer the question: given a very small number of pose annotations, what is the best way to train a single-view 3D reconstruction model?

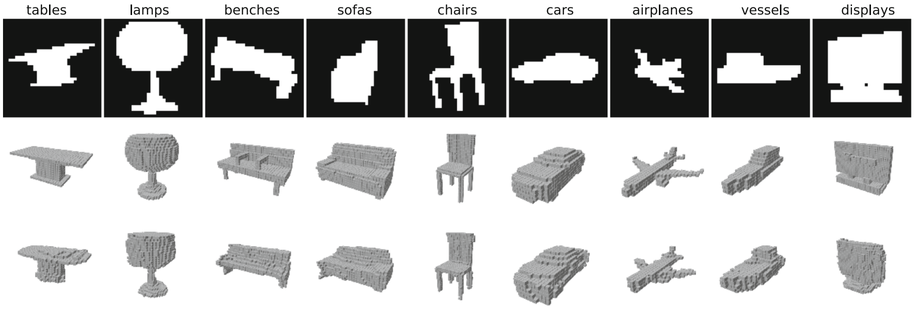

Fig. 7. 3D shape generation on the validation set. The top row shows input images (32 × 32 grayscale). The corresponding ground truth voxels and generated ones are presented in the middle row and bottom row, respectively. The models are trained with semi-supervised single-category setting with 50% pose supervision.

Table 4 compares multiple training strategies on `chairs`: using just the pose-annotated images of `chairs` (**S, P**), using just unlabeled images of `chairs` (**S, U**), using both pose-annotated and unlabeled images of `chairs` (**S, P+U**),

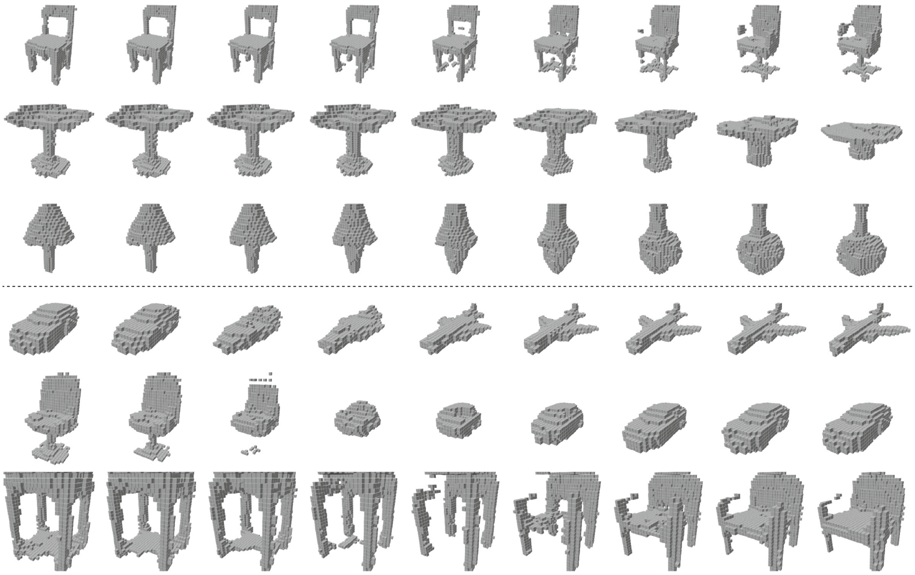

Fig. 8. Interpolation within-category (top 3 rows) and cross-category (bottom 3 rows). Given the latent vector of the left most shape $\mathbf{z}_1$ and the right-most shape $\mathbf{z}_2$, intermediate shapes correspond to $G(\mathbf{z}_1 + \alpha(\mathbf{z}_2 - \mathbf{z}_1))$, where $\alpha \in [0, 1]$.

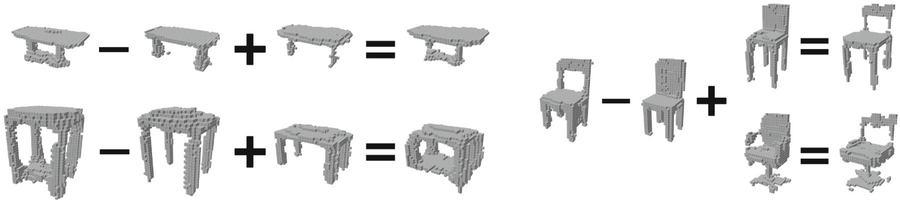

Fig. 9. Latent space arithmetic.

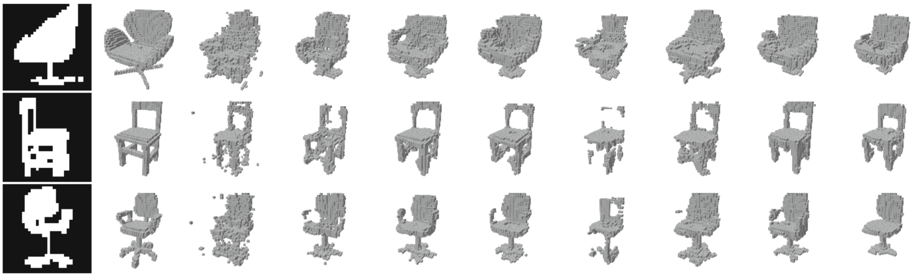

Fig. 10. Shape predictions from models with different amount of pose supervisions. From left to right: input image, ground truth voxel, and then shapes from models presented in Fig. 5. P: training with pose annotation; S: training with unlabeled data. The percentage indicates the amount of pose annotation.

combining multiple categories to train a category-agnostic model (**M**), and fine-tuning a category-agnostic model for `chairs` (**FT**). The fine-tuned model works best, indicating that it is best to combine both pose-annotated and unlabeled images, to leverage multiple categories and to retain category-specificity.

5.7 Qualitative Results

Figure 7 shows some qualitative results from our category-specific model trained with 50% pose annotations. In addition to single-image 3D reconstruction, our model learns a meaningful representation of shape, as shown by the ability to do interpolation and arithmetic in the latent space (Figs. 8 and 9).

The qualitative impact of reducing annotations is shown in Fig. 10. When the amount of supervision is reduced, one sees a significant amount of noise in the 3D reconstructions, which seems to reduce when unlabeled images are included.

6 Conclusions

In conclusion, we propose a unified and end-to-end model to use both images labeled with camera pose and unlabeled images as supervision for single view 3D reconstruction, and evaluate different training strategies when annotations are limited. Our experiments show that one can train a single-view reconstruction model with few pose annotations when leveraging unlabeled data. Future work will include confirming and extending these results on more practical settings with high resolution RGB images and arbitrary camera locations.

References

1. Agarwal, S.: Building rome in a day. Commun. ACM **54**(10), 105–112 (2011)
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
3. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3D faces. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques. ACM Press/Addison-Wesley Publishing Co. (1999)
4. Chang, A.X., et al.: ShapeNet: an information-rich 3D model repository. Technical report arXiv:1512.03012 [cs.GR], Stanford University – Princeton University – Toyota Technological Institute at Chicago (2015)
5. Choy, C.B., Xu, D., Gwak, J.Y., Chen, K., Savarese, S.: 3D-R2N2: a unified approach for single and multi-view 3D object reconstruction. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 628–644. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_38
6. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3D object reconstruction from a single image. In: CVPR, vol. 2, p. 6 (2017)
7. Gadelha, M., Maji, S., Wang, R.: 3D shape induction from 2D views of multiple objects. In: 3DV (2017)

8. Girdhar, R., Fouhey, D.F., Rodriguez, M., Gupta, A.: Learning a predictable and generative vector representation for objects. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 484–499. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_29
9. Goodfellow, I., et al.: Generative adversarial nets. In: NIPS (2014)
10. Gwak, J., Choy, C.B., Garg, A., Chandraker, M., Savarese, S.: Weakly supervised generative adversarial networks for 3D reconstruction. In: 3DV (2017)
11. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: ICML (2015)
12. Kar, A., Tulsiani, S., Carreira, J., Malik, J.: Category-specific object reconstruction from a single image. In: CVPR (2015)
13. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (2015)
14. Kodali, N., Hays, J., Abernethy, J., Kira, Z.: On convergence and stability of GANs (2018)
15. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: ICML (2013)
16. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: ICLR (2016)
17. Rezende, D.J., Eslami, S.A., Mohamed, S., Battaglia, P., Jaderberg, M., Heess, N.: Unsupervised learning of 3D structure from images. In: NIPS (2016)
18. Rock, J., Gupta, T., Thorsen, J., Gwak, J., Shin, D., Hoiem, D.: Completing 3D object shape from one depth image. In: CVPR (2015)
19. Saxena, A., Sun, M., Ng, A.Y.: Make3D: learning 3D scene structure from a single still image. In: PAMI (2009)
20. Tulsiani, S., Efros, A.A., Malik, J.: Multi-view consistency as supervisory signal for learning shape and pose prediction. In: CVPR (2018)
21. Tulsiani, S., Zhou, T., Efros, A.A., Malik, J.: Multi-view supervision for single-view reconstruction via differentiable ray consistency. In: CVPR (2017)
22. Ullman, S.: The interpretation of structure from motion. Proc. R. Soc. Lond. B. **203**(1153), 405–426 (1979)
23. Vicente, S., Carreira, J., Agapito, L., Batista, J.: Reconstructing PASCAL VOC. In: CVPR (2014)
24. Wu, J., et al.: Single image 3D interpreter network. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 365–382. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_22
25. Wu, J., Zhang, C., Xue, T., Freeman, W.T., Tenenbaum, J.B.: Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In: NIPS (2016)
26. Wu, Z., et al.: 3D shapenets: a deep representation for volumetric shapes. In: CVPR (2015)
27. Yan, X., Yang, J., Yumer, E., Guo, Y., Lee, H.: Perspective transformer nets: learning single-view 3D object reconstruction without 3D supervision. In: NIPS (2016)