Learning to Generate and Reconstruct 3D Meshes with only 2D Supervision

Paul Henderson
p.m.henderson@ed.ac.uk

School of Informatics
University of Edinburgh
Edinburgh, Scotland

Vittorio Ferrari
vittoferrari@google.com

Google Research
Zürich, Switzerland

Abstract

We present a unified framework tackling two problems: class-specific 3D reconstruction from a single image, and generation of new 3D shape samples. These tasks have received considerable attention recently; however, existing approaches rely on 3D supervision, annotation of 2D images with keypoints or poses, and/or training with multiple views of each object instance. Our framework is very general: it can be trained in similar settings to these existing approaches, while also supporting weaker supervision scenarios. Importantly, it can be trained purely from 2D images, without ground-truth pose annotations, and with a single view per instance. We employ meshes as an output representation, instead of voxels used in most prior work. This allows us to exploit shading information during training, which previous 2D-supervised methods cannot. Thus, our method can learn to generate and reconstruct concave object classes. We evaluate our approach on synthetic data in various settings, showing that (i) it learns to disentangle shape from pose; (ii) using shading in the loss improves performance; (iii) our model is comparable or superior to state-of-the-art voxel-based approaches on quantitative metrics, while producing results that are visually more pleasing; (iv) it still performs well when given supervision weaker than in prior works.

1 Introduction

Reconstructing 3D objects from 2D images is a long-standing research area in computer vision. While traditional methods rely on multiple images of the same object instance [2, 6, 8, 10, 21, 23, 28], there has recently been a surge of interest in learning-based methods that can infer 3D structure from a single image, assuming that it shows an object of a class seen during training [5, 7, 11, 14, 31, 33, 36, 38]. A related problem to reconstruction is that of generating new 3D shapes from a given object class *a priori*, i.e. without conditioning on an image. Again, there have recently been several works that apply deep learning techniques to this task [9, 22, 29, 34, 39].

Most learning-based methods for reconstruction and generation rely on strong supervision. For generation, [22, 26, 29, 34, 39] use large collections of manually constructed 3D shapes [4, 35]. For reconstruction, [5, 7, 11, 38] require training images paired with aligned 3D meshes; [14] relaxes this slightly by not requiring the images and meshes to be paired. While other methods do not rely on 3D ground-truth, they still require annotations on the 2D

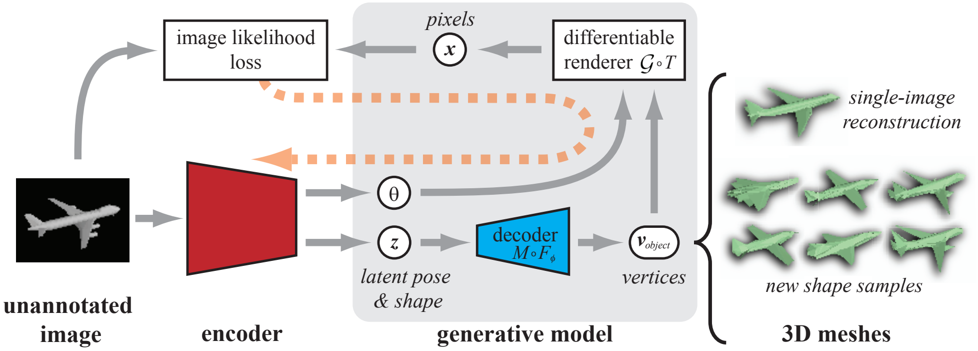

Figure 1: Given only unannotated 2D images as training data, our model learns (1) to reconstruct and predict the pose of 3D meshes from a single test image, and (2) to generate new 3D mesh samples. It is trained end-to-end (orange dashed arrow) to reconstruct input images, via a differentiable renderer that produces lit, shaded RGB images, allowing us to exploit shading cues in the loss.

training images such as keypoints [17, 32] and object poses [31, 33, 36]. Furthermore, some of them also require multiple views for each object instance [26, 33, 36]. In this paper, we consider the more challenging setting where we only have access to unannotated 2D images for training, without ground-truth pose, keypoints, or 3D shape, and with a single view per object instance; this setting is considered in just one previous work [9].

It is well known that *shading* provides an important cue for 3D understanding [16]. It allows determination of surface orientations, if the lighting and material characteristics are known; this has been explored in numerous works on shape-from-shading over the years [1, 16, 37]. Unlike learning-based approaches, these methods can only reconstruct non-occluded parts of an object, and achieving good results requires strong priors [1]. Conversely, existing learning-based generation and reconstruction methods can reason over occluded or visually-ambiguous areas, but do not leverage shading information in their loss. Furthermore, the vast majority use voxel grids as an output representation (except [29, 39]); while easy to work with, these cannot model surfaces that are not axis-aligned, limiting the usefulness of shading cues. To exploit shading information in a learning-based approach, we therefore need to move beyond voxels; a natural choice of representation is then 3D *meshes*. Meshes are ubiquitous in computer graphics, and have desirable properties for our task: they can represent surfaces of arbitrary orientation and dimensions at fixed cost, and are able to capture fine details. Thus, they avoid the visually displeasing 'blocky' reconstructions that result from voxels. We also go beyond monochromatic light, considering the case of coloured directional lighting; this provides even stronger shading cues when combined with arbitrarily-oriented mesh surfaces.

In this paper, we present a unified framework for both reconstruction and generation of 3D shapes, that is trained with only 2D supervision, and models 3D meshes rather than voxels (Fig. 1). Our framework is very general, and can be trained in similar settings to existing models [31, 33, 36], while also supporting weaker supervision scenarios. It allows:

- use of different **mesh parameterisations**, which lets us incorporate useful modeling priors such as smoothness or composition from primitives
- exploitation of **shading cues** due to monochromatic or coloured directional lighting, letting us discover concave structures that silhouette-based methods [9, 31, 36] cannot

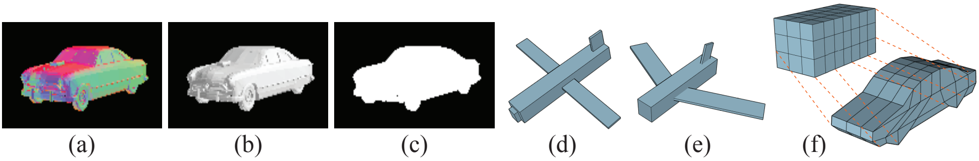

Figure 2: **Lighting**: Coloured directional lighting (a) provides strong cues for surface orientation; white light (b) provides less information; silhouettes (c) provide none at all. Our model is able to exploit the shading information from coloured or white lighting. **Mesh parameterisations**: **ortho-block** & **full-block** (assembly from cuboidal primitives, of fixed or varying orientation) are suited to objects consisting of compact parts (d-e); **subdivision** (per-vertex deformation of a subdivided cube) is suited to complex continuous surfaces (f).

- training with **varying degrees of supervision**: single or multiple views per instance, with or without ground-truth pose annotations

To achieve this, we design a probabilistic generative model that captures the full image formation process, whereby the shape and pose of a 3D mesh are first sampled independently, then a 2D rendering is produced from these (Sec. 2). We use stochastic gradient variational Bayes [19, 27] for training (Sec. 3). This involves learning an *inference network* that can predict 3D shape and pose from a single image, with the shape placed in a canonical frame of reference, i.e. disentangled from the pose. Together, the model plus its inference network resemble a variational autoencoder [19] on pixels. It represents 3D shapes in a compact latent embedding space, and has extra layers in the decoder corresponding to the mesh representation and renderer. As we do not provide 3D supervision, the encoder and decoder must bootstrap and guide one another during training. The decoder learns the manifold of shapes, while at the same time the encoder learns to map images onto this. This learning process is driven purely by the objective of reconstructing the training images. While this is an ambiguous task and the model cannot guarantee to reconstruct the true shape of an object from a single image, its generative capability means that it always produces a plausible instance of the relevant class; the encoder ensures that this is consistent with the observed image. This works because the generative model must learn to produce shapes that reproject well over *all* training images, starting from low-dimensional latent representations. This creates an inductive bias towards regularity, which avoids degenerate solutions with unrealistic shapes that could, in isolation, explain each individual training image.

We display samples from our model in Sec. 4.1, showing that (i) the use of meshes yields more natural samples than those from voxel-based methods, and (ii) our samples are diverse and realistic. In Sec. 4.2, we quantitatively evaluate the performance of our method on single-view reconstruction and pose estimation, in the various settings described above. We show that (i) it learns to predict pose, and disentangle it from shape; (ii) exploiting information from shading improves the results; (iii) it achieves comparable or better performance than prior works with equivalent supervision; and (iv) it still performs well when given weaker supervision than supported by prior works.

2 Generative Model

Our goal is to build a probabilistic generative model of 3D meshes for a given object class. For this to be trainable with 2D supervision, we cast the entire image-formation process as a directed model (Fig. 1). We assume that the content of an image can be explained by two independent latent components—the shape of the mesh, and its pose relative to the camera. These are modelled by two low-dimensional random variables, $\mathbf{z}$ and θ respectively. The joint distribution over these and the resulting pixels $\mathbf{x}$ factorises as $P(\mathbf{x}, \mathbf{z}, \theta) = P(\theta)P(\mathbf{z})P(\mathbf{x} \,|\, \mathbf{z}, \theta)$.

Following [9, 31, 33, 36], we assume that the pose θ is parameterised by just the azimuth angle, with $\theta \sim \mathrm{Uniform}(-\pi, \pi)$. The camera is then placed at fixed distance and elevation relative to the object. Following recent works on deep latent variable models [12, 19], we assume that $\mathbf{z}$ is drawn from a standard isotropic Gaussian, and then transformed by a deterministic *decoder network*, F_ϕ, parameterised by weights ϕ which are to be learnt. This produces the *mesh parameters* $\Pi = F_\phi(\mathbf{z})$. Intuitively, the decoder network F_ϕ transforms and entangles the dimensions of $\mathbf{z}$ such that all values in the latent space map to plausible values for Π, even if these lie on a highly nonlinear manifold. Note that our approach contrasts with previous models that directly output pixels [12, 19] or voxels [9, 34] from a decoder network.

We use Π as inputs to a fixed mesh parameterisation function $M(\Pi)$, which yields vertices $\mathbf{v}_{\mathrm{object}}$ of triangles defining the shape of the object in 3D space, in a canonical pose (different options for M are described below). The vertices are transformed into camera space according to the pose θ, by a fixed function T: $\mathbf{v}_{\mathrm{camera}} = T(\mathbf{v}_{\mathrm{object}}, \theta)$. They are then rendered into an RGB image $I_0 = \mathcal{G}(\mathbf{v}_{\mathrm{camera}})$ by a rasteriser $\mathcal{G}$ with Gouraud shading [13] and Lambertian directional lighting [20]. We are free to choose the lighting parameters: our experiments include tri-directional coloured lighting, and white directional lighting with an ambient component.

The final observed pixel values $\mathbf{x}$ are modelled as independent Gaussian random variables, with means equal to the values in an L-level Gaussian pyramid [3], whose base level equals I_0, and whose L^{th} level has smallest dimension equal to 1:

$$P_\phi(\mathbf{x} \,|\, \mathbf{z}, \theta) = \prod_l P_\phi(\mathbf{x}_l \,|\, \mathbf{z}, \theta) \qquad \mathbf{x}_l \sim \mathrm{Normal}\left(I_l, \tfrac{\varepsilon}{2^l}\right) \tag{1}$$

$$I_0 = \mathcal{G}(T(M(F_\phi(\mathbf{z})), \theta)) \qquad I_{l+1} = I_l * k_G \tag{2}$$

where k_G is a small Gaussian kernel, ε is the noise magnitude at the base scale, and $*$ denotes convolution with stride two. We use a multi-scale pyramid instead of just the raw pixel values to ensure that, during training, there will be gradient forces over long distances in the image, thus avoiding bad local minima where the reconstruction is far from the input.

Mesh parameterisations. After the decoder network has transformed the latent embedding $\mathbf{z}$ into the mesh parameters Π, these are converted to actual 3D vertices using a simple, non-learnt mesh-parameterisation function M. One possible choice for M is the identity function, in which case the decoder network directly outputs vertex locations. However, initial experiments showed that this does not work well: it produces very irregular meshes with large numbers of intersecting triangles. Conversely, using a more sophisticated form for M enforces regularity of the mesh. We use three different parameterisations in our experiments.

In our first parameterisation, Π specifies the locations and scales of a fixed number of axis-aligned cuboidal *primitives* (Fig. 2d), from which the mesh is assembled [30, 39]. Changing Π can produce configurations with different topologies, depending which blocks touch or overlap, but all surfaces will necessarily be axis-aligned. In our experiments we call this **ortho-block**.

Our second parameterisation is strictly more powerful than the first: we still assemble the mesh from cuboidal primitives, but now parameterise each with a 3D rotation, in addition to its location and scale. In our experiments we call this **full-block** (Fig. 2e).

The above parameterisations are naturally suited to objects composed of compact parts, but cannot represent complex continuous surfaces. For these, we define a third parameterisation, **subdivision** (Fig. 2f). This parameterisation is based on a single unit cube, centred at the origin; the edges and faces of the cube are subdivided several times along each axis. Then, Π specifies a list of displacements, one per vertex, which deform the subdivided cube into the required shape.

3 Variational Training

We wish to learn the parameters of our model from a training set of 2D images of objects of a single class. More precisely, we assume access to a set of images $\{\mathbf{x}^{(i)}\}$, each showing an unknown object instance at unknown pose. Note that we do *not* require that there are multiple views of each object (in contrast with [36]), nor that the object poses are given as supervision (in contrast with [31, 33, 36]).

We seek to maximise the marginal log-likelihood of the training set, which is given by $\sum_i \log P_\phi(\mathbf{x}^{(i)})$, with respect to ϕ. For each image, we have

$$\log P_\phi(\mathbf{x}^{(i)}) = \log \int_{\mathbf{z},\theta} P_\phi(\mathbf{x}^{(i)} \,|\, \mathbf{z}, \theta) P(\mathbf{z}) P(\theta) d\mathbf{z} d\theta. \tag{3}$$

Unfortunately this is intractable, due to the integral over the latent space $\mathbf{z}, \theta$. Hence, we use amortised variational inference, in the form of stochastic gradient variational Bayes [19, 27]. This introduces an approximate posterior $Q_\omega(\mathbf{z}, \theta \,|\, \mathbf{x})$, parameterised by some ω that we learn jointly with the model parameters ϕ. Intuitively, Q maps an image $\mathbf{x}$ to a distribution over likely values of the latent variables $\mathbf{z}$ and θ. Instead of the log-likelihood (3), we then maximise the *evidence lower bound* (ELBO):

$$\mathop{\mathbb{E}}_{\mathbf{z},\theta \sim Q_\omega(\mathbf{z},\theta \,|\, \mathbf{x}^{(i)})} \left[\log P_\phi(\mathbf{x}^{(i)} \,|\, \mathbf{z}, \theta) \right] - KL \left[Q_\omega(\mathbf{z}, \theta \,|\, \mathbf{x}^{(i)}) \middle|\middle| P(\mathbf{z}) P(\theta) \right] \leq \log P_\phi(\mathbf{x}^{(i)}). \tag{4}$$

This lower-bound on the log-likelihood can be evaluated efficiently, as the necessary expectation is now with respect to Q, for which we are free to choose a tractable form. The expectation can then be approximated using a single sample.

We let Q be a mean-field approximation, factorised as $Q_\omega(\mathbf{z}, \theta \,|\, \mathbf{x}) = Q_\omega(\mathbf{z} \,|\, \mathbf{x}) Q_\omega(\theta \,|\, \mathbf{x})$. $Q_\omega(\mathbf{z} \,|\, \mathbf{x})$ is a multivariate Gaussian with diagonal covariance. The mean and variance of each latent dimension are given by an *encoder network*, $\mathrm{enc}_\omega(\mathbf{x})$, which takes the image $\mathbf{x}$ as input. For this encoder network we use a CNN with architecture similar to [33]. When training with multiple views per instance, we apply the encoder network to each image separately, then calculate the final shape embedding $\mathbf{z}$ by max-pooling each dimension over all views.

For the pose θ, we could similarly use a Gaussian posterior. However, many objects are roughly symmetric with respect to rotation, and so the true posterior is typically multi-modal. We capture this multi-modality by decomposing the rotation into coarse and fine parts [25]: an integer random variable θ_{coarse} that chooses from R rotation bins, and a small Gaussian offset θ_{fine} relative to this. We apply this transformation in both the generative $P(\theta)$ and variational $Q_\omega(\theta)$, giving

$$\theta = -\pi + \theta_{\mathrm{coarse}} \frac{2\pi}{R} + \theta_{\mathrm{fine}} \tag{5}$$

$$P(\theta_{\mathrm{coarse}} = r) = 1/R, \qquad P(\theta_{\mathrm{fine}}) = \mathrm{Normal}(\theta_{\mathrm{fine}} \,|\, 0, \pi/R) \tag{6}$$

$$Q_\omega\left(\theta_{\mathrm{coarse}} = r \,\middle|\, \mathbf{x}^{(i)}\right) = \rho_r\left(\mathbf{x}^{(i)}\right), \qquad Q_\omega(\theta_{\mathrm{fine}}) = \mathrm{Normal}\left(\theta_{\mathrm{fine}} \,\middle|\, \xi(\mathbf{x}^{(i)}), \zeta(\mathbf{x}^{(i)})\right) \tag{7}$$

where the variational parameters ρ_r, ξ, ζ for image $\mathbf{x}^{(i)}$ are again estimated by the encoder

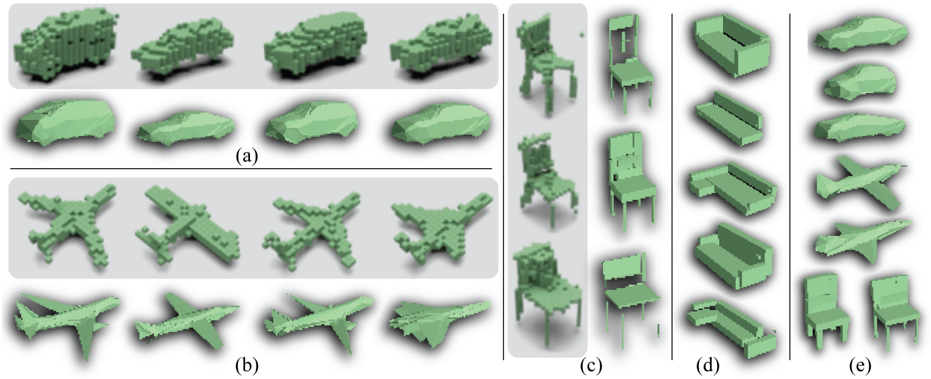

Figure 3: **(a-c)** Samples from [9] (grey background), shown next to stylistically-similar samples from our model (white background). Both are trained with a single view per instance, and without ground-truth pose. However, our model outputs meshes, and uses shading in the loss. **(d)** For *sofa*, we only show samples from our model, as [9] cannot handle sofas due to the concavities. **(e)** Additional samples from our model, showing their diversity and quality

network $\text{enc}_\omega(\mathbf{x}^{(i)})$. Provided R is sufficiently small, we can integrate directly with respect to θ_{coarse} when evaluating (4), i.e. sum over all possible rotations. We found in initial experiments that this significantly improves performance.

Imposing a uniform pose prior. While the above allows our training process to reason over different poses, it is still prone to predicting the same pose θ for every image; clearly this does not correspond to the prior on θ given by (6). The model is therefore relying on the shape embedding $\mathbf{z}$ to model all variability, rather than disentangling shape and pose. The ELBO (4) does include a KL-divergence term that should encourage latent variables to match their prior. However, it does not have a useful effect for θ_{coarse}: minimising the KL divergence from a uniform distribution for each sample individually corresponds to independently minimising all the probabilities $Q_\omega(\theta_{\text{coarse}})$, which does not encourage uniformity of the full distribution. The effect we desire is to match the aggregated posterior distribution $\left\langle Q_\omega(\theta \,|\, \mathbf{x}^{(i)})\right\rangle_i$ to the prior $P(\theta)$, where $\langle \cdot \rangle_i$ is the empirical mean over the training set. As θ_{coarse} follows a categorical distribution in both generative and variational models, we can directly minimise the L1 distance between the aggregated posterior and the prior:

$$\sum_r^R \left| \left\langle Q_\omega\left(\theta_{\text{coarse}} = r \,|\, \mathbf{x}^{(i)}\right)\right\rangle_i - P\left(\theta_{\text{coarse}} = r\right)\right| = \sum_r^R \left| \left\langle \rho_r(\mathbf{x}^{(i)})\right\rangle_i - \frac{1}{R}\right|. \tag{8}$$

We use this term in place of $KL\left[Q(\theta_{\text{coarse}} \,|\, \mathbf{x}^{(i)}) \,\middle\|\, P(\theta_{\text{coarse}})\right]$ in our loss, approximating the empirical mean with a single minibatch.

Loss. Our final loss function for a minibatch $\mathcal{B}$ is then given by

$$\begin{aligned}\sum_r^R \Bigg\{ -\left\langle \rho_r(\mathbf{x}^{(i)}) \mathop{\mathbb{E}}_{\mathbf{z},\theta_{\text{fine}} \sim Q_\omega} \left[\log P_\phi\left(\mathbf{x}^{(i)} \,\middle|\, \mathbf{z}, \theta_{\text{coarse}} = r, \theta_{\text{fine}}\right)\right]\right\rangle_{i \in \mathcal{B}} + \alpha \left| \left\langle \rho_r(\mathbf{x}^{(i)})\right\rangle_{i \in \mathcal{B}} - \frac{1}{R}\right| \Bigg\} \\ + \beta \left\langle KL\left[Q_\omega\left(\mathbf{z}, \theta_{\text{fine}} \,\middle|\, \mathbf{x}^{(i)}\right) \,\middle\|\, P(\mathbf{z}) P(\theta_{\text{fine}})\right]\right\rangle_{i \in \mathcal{B}}\end{aligned} \tag{9}$$

where β increases the relative weight of the KL term as in [15], and α controls the strength of the pose prior matching. We minimise (9) with respect to ϕ and ω using ADAM [18], applying the reparameterisation trick [19, 27] to handle the Gaussian random variables.

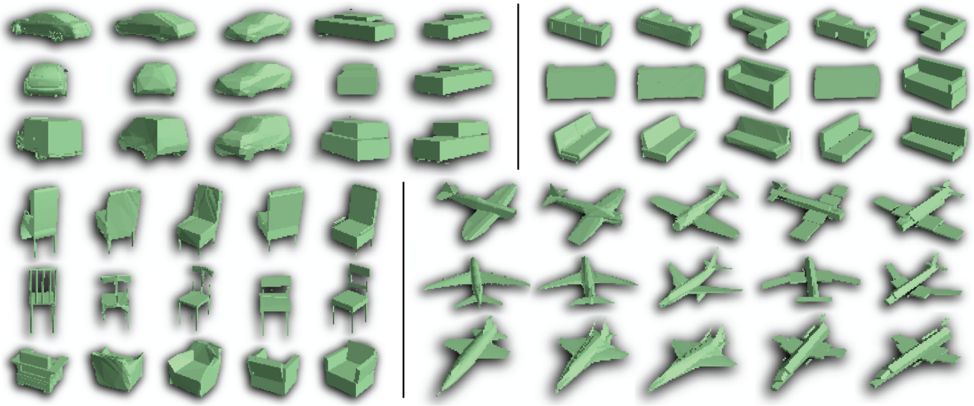

Figure 4: Qualitative examples of reconstructions. Each row of five images shows (i) ShapeNet ground-truth; (ii) our reconstruction with **subdivision** parameterisation; (iii) reconstruction aligned to canonical pose; (iv) our reconstruction with **blocks**; (v) aligned reconstruction. Experimental setting: single-view training, colour lighting, shading loss.

Differentiable rendering. Note that optimising (9) by gradient descent requires differentiating through the mesh-rendering operation $\mathcal{G}$ used to calculate $P_\phi(\mathbf{x}\,|\,\mathbf{z}, \boldsymbol{\theta})$, to find the derivative of the pixels with respect to the vertex locations and colours. While computing exact derivatives of $\mathcal{G}$ is very expensive, [24] describes an efficient approximation. We employ a similar technique here, and have made our TensorFlow implementation publicly available[1].

4 Experiments

We follow recent works [7, 9, 31, 36] and evaluate our approach using the ShapeNet dataset [4]. Using synthetic data has two advantages: it allows (i) controlled experiments modifying lighting and other parameters; (ii) benchmarking the performance of the reconstruction network against ground-truth 3D shapes. Our experiments focus on the four classes *aeroplane*, *car*, *chair*, and *sofa*. The first three are used in [9, 31, 36], while the fourth is an example of a highly concave class that is not easily handled by silhouette-based approaches.

To rigorously evaluate the performance of our model, we vary several factors:

- **Mesh parameterisations:** We evaluate the three parameterisations described in Sec. 2.
- **Lighting:** Unlike previous works [9, 31, 33, 36], our method is able to exploit shading in the images. We test in two settings, illumination by (i) three coloured directional lights (**colour**), and (ii) one white directional light plus a white ambient component (**white**).
- **Reconstruction loss:** We typically calculate the reconstruction loss (pixel log-likelihood) over the RGB shaded image (**shading**), but for comparison with [31, 33, 36] we also experiment with using only the silhouette in the loss (**silhouette**), disregarding differences in shading between the input and reconstructed pixels.
- **Pose supervision:** Previous works that train for 3D reconstruction with 2D supervision require the ground-truth pose of each training instance [31, 33, 36]. Although our method does not need this, we evaluate whether it can benefit from it.
- **Multiple views:** [33, 36] require that multiple views of each instance are presented together in each training batch, and [31] also focuses on this setting. Our model does

[1] *DIRT: a fast Differentiable Renderer for TensorFlow*, `https://github.com/pmh47/dirt`

	car				**chair**				**aeroplane**				**sofa**			
	iou	*err*	*acc*	*iou*$\vert\theta$	*iou*	*err*	*acc*	*iou*$\vert\theta$	*iou*	*err*	*acc*	*iou*$\vert\theta$	*iou*	*err*	*acc*	*iou*$\vert\theta$
ortho-block	0.71	7.3	**0.84**	0.74	0.41	9.2	**0.69**	0.49	0.30	7.9	0.73	0.24	**0.59**	**7.3**	**0.94**	**0.69**
full-block	0.54	6.5	0.82	0.63	**0.46**	**4.6**	**0.69**	**0.51**	**0.51**	**4.4**	**0.89**	**0.57**	0.39	9.1	0.70	0.68
subdivision	**0.77**	**4.7**	**0.84**	**0.81**	0.39	7.9	0.65	**0.51**	0.49	6.7	0.64	**0.57**	0.39	14.7	0.52	0.59

Table 1: Reconstruction performance for four classes, with three different mesh parameterisations (Sec. 2). For each class, the first three columns are in the default setting of no pose supervision and correspond to the metrics in Sec. 4.2; iou$\vert\theta$ is the IOU when trained with pose supervision. Higher is better for *iou* and *acc*; lower is better for *err*. Experimental setting: single-view training, colour lighting, shading loss.

not require this, but for comparison we include results with four views per instance at training time, and either one or four at test time.

During training, we construct each minibatch by randomly sampling 128 meshes from the relevant ShapeNet class uniformly with replacement. For each selected mesh, we render a single image, using a pose sampled from Uniform$(-\pi, \pi)$. Only these images are used to train the model, not the meshes themselves. In experiments using multiple views, we instead sample 32 meshes and four poses per mesh, and correspondingly render four images.

4.1 Generation

Fig. 3 shows examples of meshes sampled from our model, using the same setting as [9] (i.e. single-view training without pose supervision). That is the only prior work that learns a 3D generative model with just images as supervision. We manually selected samples from our model that are stylistically similar to those from [9] to allow side-by-side comparison.

We see that in all cases, generating meshes tends to give cleaner, more visually-pleasing samples than voxels (as used by [9]). For *chair*, our model is able to capture the very narrow legs; for *aeroplane*, it captures the diagonal edges of the wings; for *car*, it captures the smoothly curved edges. We have also successfully learnt a model for the concave class *sofa*—which is impossible for [9] as it does not consider shading. Finally, note that our samples are diverse: the model generates various different styles for each class.

4.2 Reconstruction

We now evaluate the performance of our model on 3D reconstruction from a single image. We benchmark on a held-out test set, following the protocol of [36], where each object is presented at 24 different poses, and statistics aggregated across objects and poses. We evaluate according to the following measures:

- *iou*: to measure the shape reconstruction error, we calculate the mean intersection-over-union between the predicted mesh and ground-truth; this follows recent works on reconstruction [31, 36]. To calculate this, we voxelise both meshes at a resolution of 32^3
- *err*: to measure the pose estimation error, we calculate the median error in degrees of predicted rotations
- *acc*: again to evaluate pose estimation, we measure the fraction of instances whose predicted rotation is within $\pi/6$ of the ground-truth rotation.

Object classes and mesh parameterisations. Table 1 shows the performance of our model on four different classes, comparing the three mesh parameterisations of Sec. 2. This focuses on our default setting of colour lighting, shading loss, single-view training without pose supervision (columns *iou, err, acc*); we also give *iou* when trained with pose supervision (column *iou*$\vert\theta$). We see that different parameterisations are better suited to different classes, in

	car				**chair**				**aeroplane**				**sofa**			
	iou	*err*	*acc*	*iou*|θ	*iou*	*err*	*acc*	*iou*|θ	*iou*	*err*	*acc*	*iou*|θ	*iou*	*err*	*acc*	*iou*|θ
colour	**0.77**	**4.7**	**0.84**	**0.81**	**0.46**	**4.6**	**0.69**	**0.51**	**0.51**	**4.4**	**0.89**	**0.57**	**0.59**	**7.3**	**0.94**	0.69
white	0.58	13.8	0.82	0.81	0.25	33.6	0.49	0.42	0.42	7.7	0.85	0.54	0.51	56.1	0.49	**0.71**
cl.+sil.	0.46	65.2	0.29	0.64	0.28	51.7	0.35	0.48	0.20	17.8	0.57	0.47	0.27	89.8	0.15	0.57

Table 2: Reconstruction performance with different lighting and loss. *colour* indicates three coloured directional lights with shading loss; *white* indicates a single white directional light plus white ambient, with shading loss; *cl.+sil.* indicates coloured lighting with only the silhouette used in the loss. Our model can exploit the extra information gained by considering shading in the loss, and coloured directional lighting helps further. Experimental setting: single-view training, best mesh parameterisations from Table 1.

	car				**chair**			
	iou	*err*	*acc*	*iou*|θ	*iou*	*err*	*acc*	*iou*|θ
single-view	0.77	4.7	0.84	0.81	0.46	4.6	0.69	0.51
4-view train, 4-view test	**0.83**	**2.6**	**0.94**	**0.86**	**0.51**	4.7	0.72	**0.55**
4-view train, 1-view test	0.81	5.1	0.93	0.83	0.46	**2.5**	**0.78**	0.50

Table 3: Reconstruction performance with multiple views at train/test time. Our model is able to exploit the extra information gained through multiple views, and can benefit even when testing with a single view. Experimental setting: best mesh parameterisations from Table 1, colour lighting, shading loss.

line with our expectations. Cars have smoothly curved edges, and are well-approximated by a single simply-connected surface; hence, **subdivision** performs well. Chairs vary in topology (e.g. the back may be solid or slatted) and sometimes have non-axis-aligned surfaces, so the flexible **full-block** parameterisation performs best. Aeroplanes have one dominant topology and include non-axis-aligned surfaces; both **full-block** and **subdivision** perform well here. Sofas often consist of axis-aligned blocks, so the **ortho-block** parameterisation is expressive enough to model them. We hypothesise that it performs better than the other more flexible parameterisations as it is easier for training to find a good solution in a more restricted representation space. This is effectively a form of regularisation. Overall, the best reconstruction performance is achieved for cars, which accords with [7, 31, 36].

The low values of *err* (and corresponding high values of *acc*) indicate that the model has indeed learnt to disentangle pose from shape. This is noteworthy given the model has seen only unannotated 2D images with arbitrary poses—disentanglement of these factors presumably arises because it is easier for the model to learn to reconstruct in a canonical frame, given that it is encouraged by our loss to predict diverse poses. However, providing the ground-truth poses as input improves reconstruction performance further in almost all cases (column *iou*|θ vs. *iou*).

Benefit of lighting. Table 2 shows how reconstruction performance varies with the different choices of lighting, **colour** and **white**, using **shading** loss. Coloured directional lighting provides more information during training than white lighting, and the results are correspondingly better. We also show performance with **silhouette** loss for coloured light. This performs significantly worse than with shading in the loss, in spite of the input images being identical. Thus, back-propagating information from shading through the renderer does indeed help with learning—it is not merely that colour images contain more information for the encoder network. As in the previous experiment, we see that pose supervision helps the model (column *iou*|θ vs. *iou*). In particular, only with pose supervision are silhouettes informative enough for the model to learn a canonical frame of reference reliably, as evidenced

	lighting	*loss*	**car**	**chair**	**aeroplane**	**sofa**
DRC [31]	*white*	*silhouette*	0.73	0.43	0.50	-
DRC [31]	*white*	*depth*	0.74	0.44	0.49	-
PTN [36]	*white*	*silhouette*	0.71	0.50	0.56	0.62
PTN, our images	*colour*	*silhouette*	0.66	0.22	0.42	0.46
ours	*white*	*silhouette*	0.71	0.25	0.53	0.68
ours	*white*	*shading*	0.79	0.44	0.54	**0.69**
ours	*colour*	*shading*	**0.83**	**0.51**	**0.57**	**0.69**
PSG [7]	*white*	*3D*	*0.83*	*0.54*	*0.60*	*0.71*

Table 4: Reconstruction performance (iou|θ) in a setting matching [31, 36] (multi-view training; best parameterisations from Table 1), but with mesh output instead of voxels. *PTN, our images* is running the unmodified public code of [36] with its normal silhouette loss, on our coloured images. The final row shows performance of a state-of-the-art method [7] with full 3D supervision—note that our colour results are comparable with this, in spite of using only unannotated 2D images as supervision

by the high median rotation errors without (column *err*).

Multi-view training/testing. Table 3 shows results when we provide 4 views of each object instance to the model. Using 4 views at both training and testing time improves results in all cases—the model has learnt to exploit the additional information about each instance. There is also a smaller performance improvement when we train with 4 views, but test with only one—although the network has not been optimised for the single-view task during training.

Comparison to previous works. Table 4 compares our results with previous works. Here, we conduct experiments in a setting matching [31, 36]: multiple views at training time, with ground-truth pose supervision. This shows that our results using meshes are roughly comparable with these previous works using voxels, even when only silhouette supervision is used (our results are worse on 'chair', but better on 'sofa'). Furthermore, when we add shading information to the loss (which these previous works cannot), our results show a significant improvement; coloured lighting helps even further. We also show results for [36] using our coloured lighting images as input, but their silhouette loss. This performs worse than our method on the same images, again showing that shading in the loss is useful—our colour images are not simply more informative to the encoder network than those of [36]. Interestingly, when trained with shading or colour, our method outperforms [31] even when the latter is trained with depth information. When trained with colour, our results are even close to [7], which is a state-of-the-art method trained with full 3D supervision.

5 Conclusion

We have presented a framework for generation and reconstruction of 3D meshes. Our approach is flexible and supports many different supervision settings, including weaker supervision than any prior works (i.e. a single view per training instance, and without pose annotations). Unlike prior works, we can exploit shading cues due to directional lighting; we have shown that this improves performance over silhouettes. Moreover, performance is higher than that of a method with depth supervision [31], and even close to the state-of-the-art results using full 3D supervision [7]. Finally, ours is the first method that can learn a generative model of 3D meshes, trained with only 2D images. We have shown that use of meshes leads to more visually-pleasing results than prior voxel-based works [9].

References

[1] Jonathan T. Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE Trans. on PAMI*, 37(8):1670–1687, 2015.

[2] A. Broadhurst, T. W. Drummond, and R. Cipolla. A probabilistic framework for space carving. In *ICCV*, 2001.

[3] Peter J. Burt and Edward H. Adelson. The laplacian pyramid as a compact image code. *IEEE Trans. on Communications*, COM-31(4):532–540, 1983.

[4] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. *arXiv preprint*, arXiv:1512:03012, 2015.

[5] 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.

[6] Jeremy S. De Bonet and Paul Viola. Roxels: Responsibility weighted 3D volume reconstruction. In *ICCV*, 1999.

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

[8] Yasutaka Furukawa and Carlos Hernández. Multi-view stereo: A tutorial. *FT CGV*, 9 (1–2):1–148, 2015.

[9] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3D shape induction from 2D views of multiple objects. In *3DV*, 2017.

[10] Pau Gargallo, Peter Sturm, and Sergi Pujades. An occupancy-depth generative model of multi-view images. In *ICCV*, 1999.

[11] R. Girdhar, D.F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In *ECCV*, 2016.

[12] I. Goodfellow, J. Pouget-Abadle, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[13] Henri Gouraud. Continuous shading of curved surfaces. *IEEE Trans. on Computers*, C-20(6):623–629, 1971.

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

[15] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-VAE: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

[16] Berthold Horn. Obtaining shape from shading information. In Patrick Henry Winston, editor, *The Psychology of Computer Vision*. 1975.

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

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

[19] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *ICLR*, 2014.

[20] Johann Heinrich Lambert. *Photometria*. Eberhard Klett Verlag, 1760.

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

[22] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. GRASS: Generative recursive autoencoders for shape structures. *Trans. on Graphics*, 36(4), 2017.

[23] Shubao Liu and David B. Cooper. Ray markov random fields for image-based 3D modeling: model and efficient inference. In *CVPR*, 2010.

[24] Matthew M. Loper and Michael J. Black. OpenDR: An approximate differentiable renderer. In *ECCV*, pages 154–169, 2014.

[25] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3D bounding box estimation using deep learning and geometry. In *CVPR*, 2017.

[26] Danilo J. Rezende, S. M. Ali Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3D structure from images. In *NIPS*, 2016.

[27] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

[28] Steven Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, 2006.

[29] Ayan Sinha, Asim Unmesh, Qixing Huang, and Karthik Ramani. SurfNet: Generating 3D shape surfaces using deep residual networks. In *CVPR*, 2017.

[30] Shubham Tulsiani, Hao Su, Leonidas J. Guibas, Alexei A. Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *CVPR*, 2017.

[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.

[32] Sara Vicente, João Carreira, Lourdes Agapito, and Jorge Batista. Reconstructing PASCAL VOC. In *CVPR*, 2014.

[33] Olivia Wiles and Andrew Zisserman. SilNet: Single- and multi-view reconstruction by learning from silhouettes. In *BMVC*, 2017.

[34] 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.

[35] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shape modeling. In *CVPR*, 2015.

[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.

[37] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape-from-shading: A survey. *IEEE Trans. on PAMI*, 21(8):690–706, 1999.

[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.

[39] Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3D-PRNN: Generating shape primitives with recurrent neural networks. In *ICCV*, 2017.