Learning single-image 3D reconstruction by generative modelling of shape, pose and shading

Paul Henderson · **Vittorio Ferrari**

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, most 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 existing approaches, while also supporting weaker supervision. Importantly, it can be trained purely from 2D images, without pose annotations, and with only a single view per instance. We employ meshes as an output representation, instead of voxels used in most prior work. This allows us to reason over lighting parameters and 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 in various settings, showing that: (i) it learns to disentangle shape from pose and lighting; (ii) using shading in the loss improves performance compared to just silhouettes; (iii) when using a standard single white light, our model outperforms state-of-the-art 2D-supervised methods, both with and without pose supervision, thanks to exploiting shading cues; (iv) performance improves further when using multiple coloured lights, even approaching that of state-of-the-art 3D-supervised methods; (v) shapes produced by our model capture smooth surfaces and fine details better than voxel-based approaches; and (vi) our approach supports concave classes such as bathtubs and sofas, which methods based on silhouettes cannot learn.

P. Henderson
School of Informatics, University of Edinburgh, Scotland
E-mail: paul@pmh47.net

V. Ferrari
Google Research, Zürich, Switzerland
E-mail: vittoferrari@google.com

Keywords single-image 3D reconstruction · generative models · shape from shading · neural networks

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 (Seitz et al., 2006; Furukawa and Hernández, 2015; Broadhurst et al., 2001; Laurentini, 1994; De Bonet and Viola, 1999; Gargallo et al., 1999; Liu and Cooper, 2010), 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 (e.g. Fan et al., 2017; Choy et al., 2016; Yan et al., 2016; see Sect. 2.1). 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 (e.g. Wu et al., 2016; Zou et al., 2017; Gadelha et al., 2017; see Sect. 2.2).

Most learning-based methods for reconstruction and generation rely on strong supervision. For generation (e.g. Wu et al., 2016; Zou et al., 2017), this means learning from large collections of manually constructed 3D shapes, typically ShapeNet (Chang et al., 2015) or ModelNet (Wu et al., 2015). For reconstruction (e.g. Choy et al., 2016; Fan et al., 2017; Richter and Roth, 2018), it means learning from images paired with aligned 3D meshes, which is very expensive supervision to obtain (Yang et al., 2018). While a few methods do not rely on 3D ground-truth, they still require keypoint annotations on the 2D training images (Vicente et al., 2014; Kar et al., 2015; Kanazawa et al., 2018), and/or multiple views for each object instance, often with pose annotations (Yan et al., 2016; Wiles and Zisserman, 2017; Kato et al., 2018; Tulsiani et al., 2018; Insafutdinov and Dosovit-

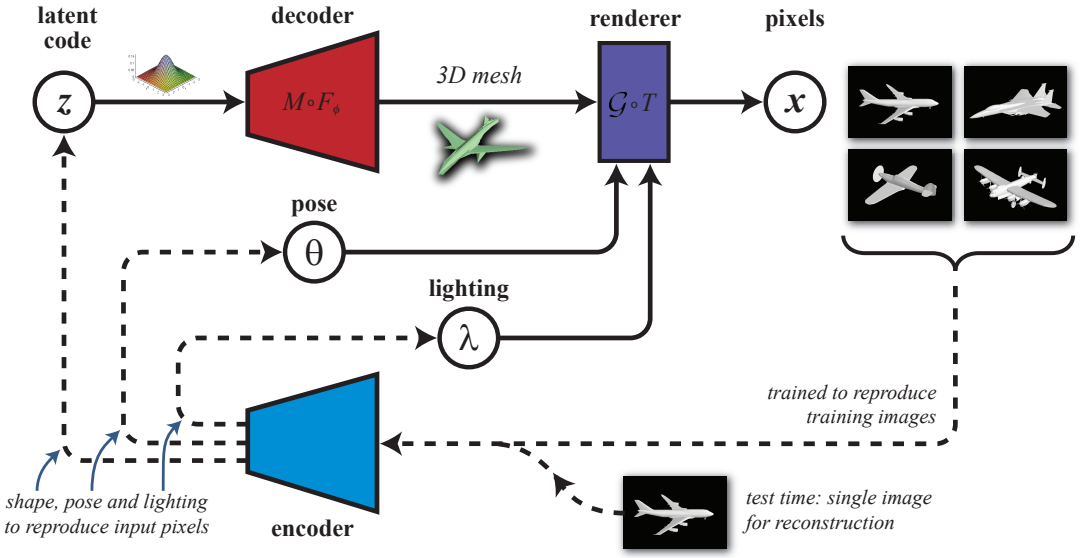


Fig. 1 Given only unannotated 2D images as training data, our model learns (1) to reconstruct and predict the pose of 3D meshes from a single test image, and (2) to generate new 3D mesh samples. The generative process (solid arrows) samples a Gaussian embedding, decodes this to a 3D mesh, renders the resulting mesh, and finally adds Gaussian noise. It is trained end-to-end to reconstruct input images (dashed arrows), via an encoder network that learns to predict and disentangle shape, pose, and lighting. The renderer produces lit, shaded RGB images, allowing us to exploit shading cues in the reconstruction loss.

skiy, 2018). 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.

It is well known that *shading* provides an important cue for 3D understanding (Horn, 1975). 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 (Horn, 1975; Zhang et al., 1999; Barron and Malik, 2015). Unlike learning-based approaches, these methods can only reconstruct non-occluded parts of an object, and achieving good results requires strong priors (Barron and Malik, 2015). 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 majority use voxel grids or point clouds as an output representation. Voxels are easy to work with, but cannot model non-axis-aligned surfaces, while point clouds do not represent surfaces explicitly at all. In both cases, this limits the usefulness of shading cues. To exploit shading information in a learning-based approach, we therefore need to move to a different representation; a natural choice is 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. Our model also explicitly reasons over the lighting parameters, jointly with the object shape, allowing it to exploit shading information even in cases where the lighting parameters are unknown—which classical shape-from-shading methods cannot.

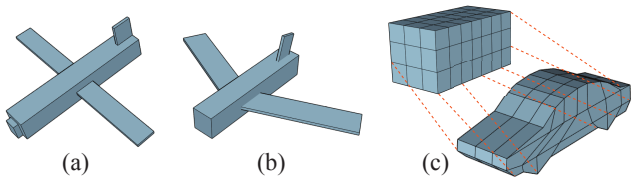
In this paper, we present a unified framework for both reconstruction and generation of 3D shapes, that is trained to model 3D meshes using only 2D supervision (Fig. 1). Our framework is very general, and can be trained in similar settings to existing models (Tulsiani et al., 2017b; Yan et al., 2016; Wiles and Zisserman, 2017; Tulsiani et al., 2018), 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 cannot (Gadelha et al., 2017; Tulsiani et al., 2017b, 2018; Yan et al., 2016; Soltani et al., 2017).
- 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 of a 3D mesh, its pose, and incident lighting are first sampled independently, then a 2D rendering is produced from these (Sect. 3). We use stochastic gradient variational Bayes

(a) (b) (c)

Fig. 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.



(a) (b) (c)

Fig. 3 Mesh parameterisations: **ortho-block** & **full-block** (assembly from cuboidal primitives, of fixed or varying orientation) are suited to objects consisting of compact parts (a-b); **subdivision** (per-vertex deformation of a subdivided cube) is suited to complex continuous surfaces (c).

for training (Kingma and Welling, 2014; Rezende et al., 2014) (Sect. 4). This involves learning an *inference network* that can predict 3D shape, pose and lighting 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 (Kingma and Welling, 2014) 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.

In Sect. 5, we demonstrate our method on 13 diverse object classes. This includes several highly concave classes, which methods relying on silhouettes cannot learn correctly (Yan et al., 2016; Gadelha et al., 2017; Tulsiani et al., 2017b, 2018). We first display samples from the distribution of shapes learnt by our model, showing that (i) the use of meshes yields smoother, more natural samples than those from voxel-based methods (Gadelha et al., 2017), (ii) different mesh parameterisations are better suited to different object classes, and (iii) our samples are diverse and realistic, covering multiple modes of the training distribution. We also demonstrate that our model learns a meaningful latent space, by showing that interpolating between points in it yields realistic intermediate samples. We then quantitatively evaluate performance of our method on single-view reconstruction and pose estimation, showing that: (i) it learns to predict pose, and disentangle it from shape, without either being given as supervision; (ii) exploiting information from shading improves results over using silhouettes in the reconstruction loss, even when the model must learn to estimate the lighting parameters and disentangle them from surface normals; (iii) when using a standard single white light, our model outperforms state-of-the-art 2D-supervised methods (Kato et al., 2018), both with and without pose supervision, thanks to exploiting shading cues; (iv) performance improves further when using multiple coloured lights, even approaching that of state-of-the-art 3D-supervised methods (Fan et al., 2017; Richter and Roth, 2018). Finally, we evaluate the impact of design choices such as different mesh parameterisations and latent space dimensionalities, showing which choices work well for different object classes.

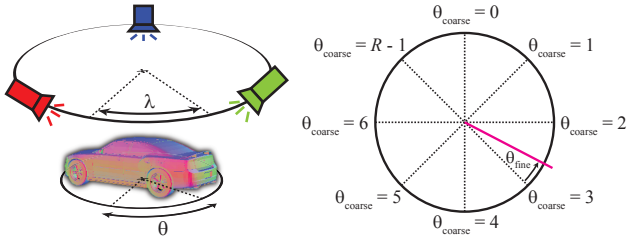
A preliminary version of this work appeared as Henderson and Ferrari (2018). That earlier version assumed fixed, known lighting parameters rather than explicitly reasoning over them; also here we present a much more extensive experimental evaluation.

2 Related Work

2.1 Learning single-image 3D reconstruction

In the last three years, there has been a surge of interest in single-image 3D reconstruction; this has been enabled both by the growing maturity of deep learning techniques, and by the availability of large datasets of 3D shapes (Chang et al., 2015; Wu et al., 2015). Among such methods, we differentiate between those requiring full 3D supervision (i.e. 3D shapes paired with images), and those that need only weaker 2D supervision (e.g. pose annotations); our work here falls into the second category.

3D-supervised methods. Choy et al. (2016) apply a CNN to the input image, then pass the resulting features to a 3D deconvolutional network, that maps them to to occupancies of a 32^3 voxel grid. Girdhar et al. (2016) and Wu et al. (2016) proceed similarly, but pre-train a model to auto-encode or generate 3D shapes respectively, and regress images to the latent features of this model. Instead of directly producing voxels, Soltani et al. (2017), Shin et al. (2018) and Richter



(a) **(b)**

Fig. 4 **(a)** We parameterise the object pose relative to the camera by the azimuth angle θ, and rotate the lights around the object as a group according to a second azimuth angle λ. **(b)** To avoid degenerate solutions, we discretise θ into coarse and fine components, with θ_{coarse} categorically distributed over R bins, and θ_{fine} specifying a small offset relative to this. For example, to represent the azimuth indicated by the pink line, $\theta_{\text{coarse}} = 3$ and $\theta_{\text{fine}} = -18°$. The encoder network outputs softmax logits ρ for a categorical variational distribution over θ_{coarse}, and the mean ξ and standard deviation ζ of a Gaussian variational distribution over θ_{fine}, with ξ bounded to the range $(-\pi/R, \pi/R)$.

and Roth (2018) output multiple depth-maps and/or silhouettes, from known (fixed) viewpoints; these are subsequently fused if a voxel reconstruction is required. Fan et al. (2017) and Mandikal et al. (2018) generate point clouds as the output, with networks and losses specialised to their order-invariant structure. Like ours, the concurrent work of Wang et al. (2018) predicts meshes, but parameterises them differently to us. Tulsiani et al. (2017a) and Niu et al. (2018) both learn to map images to sets of cuboidal primitives, of fixed and variable cardinality respectively. Finally, Gwak et al. (2017) and Zhu et al. (2017) present methods with slightly weaker requirements on ground-truth. As in the previous works, they require large numbers of 3D shapes and images; however, these do not need to be paired with each other. Instead, the images are annotated only with silhouettes.

2D-supervised methods. A few recent learning-based reconstruction techniques do not rely on 3D ground-truth; these are the closest in spirit to our own. They typically work by passing input images through a CNN, which predicts a 3D representation, which is then rendered to form a reconstructed 2D silhouette; the loss is defined to minimise the difference between the reconstructed and original silhouettes. This reliance on silhouettes means they cannot exploit shading and cannot learn to reconstruct concave object classes—in contrast to our approach. Moreover, all these methods require stronger supervision than our own—they must be trained with ground-truth pose or keypoint annotations, and/or multiple views of each instance presented together during training.

Rezende et al. (2016) briefly discuss single-image reconstruction using a conditional generative model over meshes. This models radial offsets to vertices of a spherical base mesh, conditioning on an input image. The model is trained in a variational framework to maximise the reconstructed pixel likelihood. It is demonstrated only on simple shapes such as cubes and cylinders.

Yan et al. (2016) present a method that takes single image as input, and yields a voxel reconstruction. This is trained to predict voxels that reproject correctly to the input pixels, assuming the object poses for the training images are known. The voxels are projected by computing a max operation along rays cast from each pixel into the voxel grid, at poses matching the input images. The training objective is then to maximise the IOU between these projected silhouettes and the silhouettes of the original images. Kato et al. (2018) present a very similar method, but using meshes instead of voxels as the output representation. It is again trained using the silhouette IOU as the loss, but also adds a smoothness regularisation term, penalising sharply creased edges. Wiles and Zisserman (2017) propose a method that takes silhouette images as input, and produces rotated silhouettes as output; the input and output poses are provided. To generate the rotated silhouettes, they predict voxels in 3D space, and project them by a max operation along rays.

Tulsiani et al. (2017b) also regress a voxel grid from a single image; however, the values in this voxel grid are treated as occupancy probabilities, which allows use of probabilistic ray termination (Broadhurst et al., 2001) to enforce consistency with a silhouette or depth map. Two concurrent works to ours, Tulsiani et al. (2018) and Insafutdinov and Dosovitskiy (2018), extend this approach to the case where pose is not given at training time. To disentangle shape and pose, they require that multiple views of each object instance be presented together during training; the model is then trained to reconstruct the silhouette for each view using its own predicted pose, but the shape predicted from some other view. Yang et al. (2018) use the same principle to disentangle shape and pose, but assume that a small number of images are annotated with poses, which improves the accuracy significantly.

Vicente et al. (2014) jointly reconstruct thousands of object instances in the PASCAL VOC 2012 dataset using keypoint and silhouette annotations, but without learning a model that can be applied to unseen images. Kar et al. (2015) train a CNN to predict keypoints, pose, and silhouette from an input image, and then optimise the parameters of a deformable model to fit the resulting estimates. Concurrently with our work, Kanazawa et al. (2018) present a method that takes a single image as input, and produces a textured 3D mesh as output. The mesh is parameterised by offsets to the vertices of a learnt mean shape. These three methods all require silhouette and keypoint annotations on the training images, but only a single view of each instance.

Novotny et al. (2017) learn to perform single-image reconstruction using videos as supervision. Classical multi-view stereo methods are used to reconstruct the object instance in each video, and the reconstructions are used as

ground-truth to train a regression model mapping images to 3D shapes.

2.2 Generative models of 3D shape

The last three years have also seen increasing interest in deep generative models of 3D shapes. Again, these must typically be trained using large datasets of 3D shapes, while just one work requires only images (Gadelha et al., 2017).

3D-supervised methods. Wu et al. (2015) and Xie et al. (2018) train deep energy-based models on voxel grids; Huang et al. (2015) train one on surface points of 3D shapes, jointly with a decomposition into parts. Wu et al. (2016) and Zhu et al. (2018) present generative adversarial networks (GANs; Goodfellow et al., 2014) that directly model voxels using 3D convolutions; Zhu et al. (2018) also fine-tune theirs using 2D renderings. Rezende et al. (2016) and Balashova et al. (2018) both describe models of voxels, based on the variational autoencoder (VAE; Kingma and Welling, 2014). Nash and Williams (2017) and Gadelha et al. (2018) model point clouds, using different VAE-based formulations. Achlioptas et al. (2018) train an autoencoder for dimensionality reduction of point clouds, then a GAN on its embeddings. Li et al. (2017) and Zou et al. (2017) model shapes as assembled from cuboidal primitives; Li et al. (2017) also add detail by modelling voxels within each primitive. Tan et al. (2018) present a VAE over parameters of meshes. Calculating the actual vertex locations from these parameters requires a further energy-based optimisation, separate to their model. Their method is not directly applicable to datasets with varying mesh topology, including ShapeNet and ModelNet.

2D-supervised methods. Soltani et al. (2017) train a VAE over groups of silhouettes from a set of known viewpoints; these may be fused to give a true 3D shape as a post-processing stage, separate to the probabilistic model. The only prior work that learns a true generative model of 3D shapes given just 2D images is Gadelha et al. (2017); this is therefore the most similar in spirit to our own. They use a GAN over voxels; these are projected to images by a simple max operation along rays, to give silhouettes. A discriminator network ensures that projections of sampled voxels are indistinguishable from projections of ground-truth data. This method does not require pose annotations, but they restrict poses to a set of just eight predefined viewpoints. In contrast to our work, this method cannot learn concave shapes, due to its reliance on silhouettes. Moreover, like other voxel-based methods, it cannot output smooth, arbitrarily-oriented surfaces. Yang et al. (2018) apply this model as a prior for single-image reconstruction, but they require multiple views per instance during training.

3 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 three independent latent components—the shape of the mesh, its pose relative to the camera, and the lighting. These are modelled by three low-dimensional random variables, $\mathbf{z}$, θ, and λ respectively. The joint distribution over these and the resulting pixels $\mathbf{x}$ factorises as $P(\mathbf{x}, \mathbf{z}, \theta, \lambda) = P(\mathbf{z})P(\theta)P(\lambda)P(\mathbf{x}\,|\,\mathbf{z}, \theta, \lambda)$.

Following Gadelha et al. (2017), Yan et al. (2016), Tulsiani et al. (2017b), and Wiles and Zisserman (2017), we assume that the pose θ is parameterised by just the azimuth angle, with $\theta \sim \text{Uniform}(-\pi, \pi)$ (Fig. 4a, bottom). The camera is then placed at fixed distance and elevation relative to the object. We similarly take λ to be a single azimuth angle with uniform distribution, which specifies how a predefined set of directional light sources are to be rotated around the origin (Fig. 4a, top). The number of lights, their colours, elevations, and relative azimuths are kept fixed. We are free to choose these; our experiments include tri-directional coloured lighting, and a single white directional light source plus an ambient component.

Following recent works on deep latent variable models (Kingma and Welling, 2014; Goodfellow et al., 2014), we assume that the embedding vector $\mathbf{z}$ 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 (Appendix A details the architecture of this network). 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 (Kingma and Welling, 2014; Goodfellow et al., 2014) or voxels (Wu et al., 2016; Gadelha et al., 2017; Zhu et al., 2018; Balashova et al., 2018) from a decoder network.

We use Π as inputs to a fixed mesh parameterisation function $M(\Pi)$, which yields vertices $\mathbf{v}_{\text{object}}$ 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: $\mathbf{v}_{\text{camera}} = T(\mathbf{v}_{\text{object}}, \theta)$. They are then rendered into an RGB image $I_0 = \mathscr{G}(\mathbf{v}_{\text{camera}}, \lambda)$ by a rasteriser $\mathscr{G}$ using Gouraud shading (Gouraud, 1971) and Lambertian surface reflectance (Lambert, 1760).

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 (Burt and Adelson, 1983), whose base level equals I_0, and whose L^{th} level has

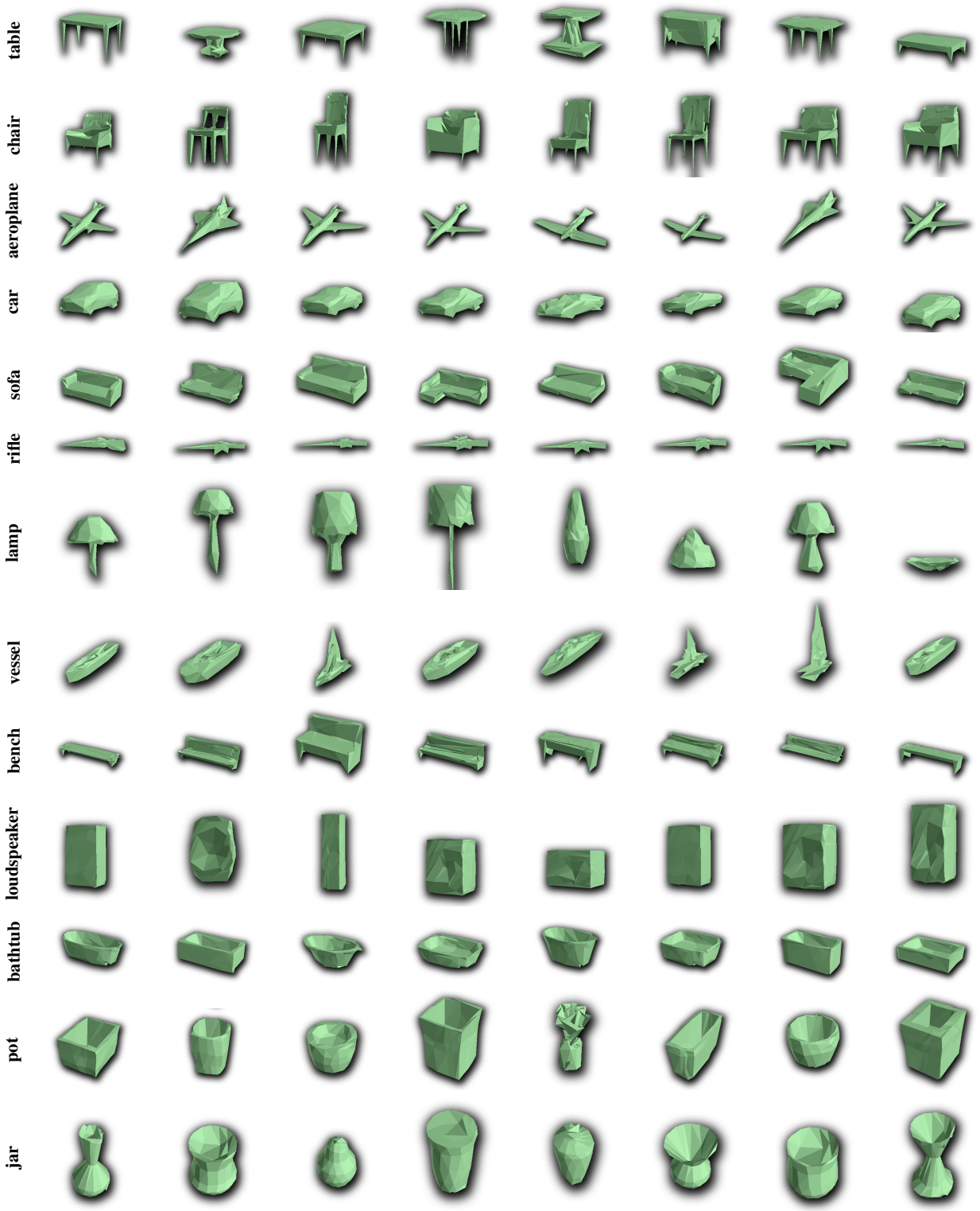


Fig. 5 Samples from our model for the ten most frequent classes in ShapeNet in order of decreasing frequency, plus three other interesting classes. Note the diversity and realism of our samples, which faithfully capture multimodal shape distributions, e.g. both straight and right-angled sofas, boats with and without sails, and straight- and delta-wing aeroplanes. We successfully learn models for the highly concave classes sofa, bathtub, pot, and jar, enabled by the fact that we exploit shading cues during training. Experimental setting: subdivision, fixed colour lighting, shading loss.

Fig. 6 Qualitative examples of reconstructions for different object classes. Each group of three images shows (i) ShapeNet ground-truth; (ii) our reconstruction; (iii) reconstruction placed in a canonical pose, with the different viewpoint revealing hidden parts of the shape. Experimental setting: subdivision, single-view training, fixed colour lighting, shading loss.

smallest dimension equal to one:

$$P_\phi(\mathbf{x}\,|\,\mathbf{z},\,\theta,\,\lambda) = \prod_l P_\phi(\mathbf{x}_l\,|\,\mathbf{z},\,\theta,\,\lambda) \tag{1}$$

$$\mathbf{x}_l \sim \text{Normal}\left(I_l,\,\tfrac{\varepsilon}{2^l}\right) \tag{2}$$

$$I_0 = \mathscr{G}(T(M(F_\phi(\mathbf{z})),\,\theta),\,\lambda) \tag{3}$$

$$I_{l+1} = I_l * k_G \tag{4}$$

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. 3a), from which the mesh is assembled (Zou et al., 2017; Tulsiani et al., 2017a). Changing Π can produce configurations with different topologies, depending which blocks touch or overlap, but all surfaces will always be axis-aligned. The scale and location of each primitive are represented by 3D vectors, resulting in a total of six parameters per primitive. 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 associate each with a rotation, in addition to its location and scale. Each rotation is parameterised as three Euler angles, yielding a total of nine parameters per primitive. In our experiments we call this **full-block** (Fig. 3b).

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. 3c). This parameterisation is based on a single cuboid, centred at the origin; the edges and faces of the cuboid are subdivided several times along each axis. Then, Π specifies a list of 3D displacements, one per vertex, which deform the subdivided cube into the required shape. In practice, we subdivide each edge into four segments, resulting in 98 vertices, hence 294 parameters.

4 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 object with unknown shape, at an unknown pose, under unknown lighting. Note that we do *not* require that there are multiple views of each object (in contrast with Yan et al. (2016) and Tulsiani et al. (2018)), nor that the object poses are given as supervision (in contrast with Yan et al. (2016), Tulsiani et al. (2017b), Wiles and Zisserman (2017), and Kato et al. (2018)).

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

$$\log P_\phi(\mathbf{x}^{(i)}) = \log \int_{\mathbf{z},\theta,\lambda} P_\phi(\mathbf{x}^{(i)}\,|\,\mathbf{z},\theta,\lambda) P(\mathbf{z}) P(\theta) P(\lambda)\,\mathrm{d}\mathbf{z}\,\mathrm{d}\theta\,\mathrm{d}\lambda \tag{5}$$

Unfortunately this is intractable, due to the integral over the latent variables $\mathbf{z}$ (shape), θ (pose), and λ (lighting). Hence, we use amortised variational inference, in the form of stochastic gradient variational Bayes (Kingma and Welling, 2014; Rezende et al., 2014). This introduces an approximate posterior $Q_\omega(\mathbf{z},\theta,\lambda\,|\,\mathbf{x})$, 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 $\mathbf{z}$, θ, and λ. Instead of the log-likelihood (5), we then maximise the *evidence lower bound* (ELBO):

$$\begin{aligned}&\mathop{\mathbb{E}}_{\mathbf{z},\theta,\lambda\sim Q_\omega(\mathbf{z},\theta,\lambda\,|\,\mathbf{x}^{(i)})}\Big[\log P_\phi(\mathbf{x}^{(i)}\,|\,\mathbf{z},\,\theta,\,\lambda)\Big]\\ &- KL\left[Q_\omega(\mathbf{z},\,\theta,\,\lambda\,|\,\mathbf{x}^{(i)})\,\middle|\middle|\,P(\mathbf{z})P(\theta)P(\lambda)\right] \le \log P_\phi(\mathbf{x}^{(i)})\end{aligned} \tag{6}$$

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, i.e. given by a product of independent variational distributions:

$$Q_\omega(\mathbf{z},\theta,\lambda\,|\,\mathbf{x}) = Q_\omega(\mathbf{z}\,|\,\mathbf{x}) Q_\omega(\theta\,|\,\mathbf{x}) Q_\omega(\lambda\,|\,\mathbf{x}) \tag{7}$$

The parameters of these distributions are produced by an *encoder network*, $\text{enc}_\omega(\mathbf{x})$, which takes the image $\mathbf{x}$ as input. For this encoder network we use a small CNN with architecture similar to Wiles and Zisserman (2017) (see Appendix A). We now describe the form of the variational distribution for each of the variables $\mathbf{z}$, θ, and λ.

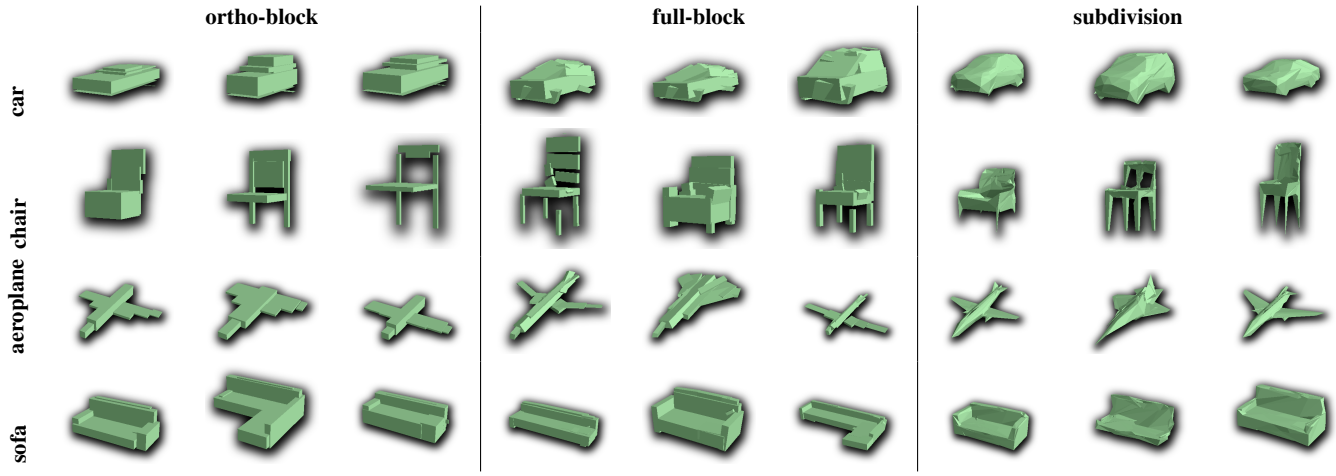


Fig. 7 Samples for four object classes, using our three different mesh parameterisations. **ortho-block** and **full-block** perform well for sofas and reasonably for chairs, but are less well-suited to aeroplanes and cars, which are naturally represented as smooth surfaces. **subdivision** gives good results for all four object classes.

Shape. For the shape embedding $\mathbf{z}$, the variational posterior distribution $Q_\omega(\mathbf{z}\,|\,\mathbf{x})$ is a multivariate Gaussian with diagonal covariance. The mean and variance of each latent dimension are produced by the encoder network. 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.

Pose. 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 (Mousavian et al., 2017): an integer random variable θ_{coarse} that chooses from R_θ rotation bins, and a small Gaussian offset θ_{fine} relative to this (Fig. 4b):

$$\theta = -\pi + \theta_{\text{coarse}}\frac{2\pi}{R_\theta} + \theta_{\text{fine}} \tag{8}$$

We apply this transformation in both the generative $P(\theta)$ and variational $Q_\omega(\theta)$, giving

$$P(\theta_{\text{coarse}} = r) = 1/R_\theta \tag{9}$$

$$P(\theta_{\text{fine}}) = \text{Normal}(\theta_{\text{fine}}\,|\,0,\, \pi/R_\theta) \tag{10}$$

$$Q_\omega\left(\theta_{\text{coarse}} = r \,\middle|\, \mathbf{x}^{(i)}\right) = \rho_r^\theta\left(\mathbf{x}^{(i)}\right) \tag{11}$$

$$Q_\omega(\theta_{\text{fine}}) = \text{Normal}\left(\theta_{\text{fine}} \,\middle|\, \xi^\theta(\mathbf{x}^{(i)}),\, \zeta^\theta(\mathbf{x}^{(i)})\right) \tag{12}$$

where the variational parameters $\rho_r^\theta, \xi^\theta, \zeta^\theta$ for image $\mathbf{x}^{(i)}$ are again estimated by the encoder network $\text{enc}_\omega(\mathbf{x}^{(i)})$. Specifically, the encoder uses a softmax output to parameterise ρ^θ, and restricts ξ^θ to lie in the range $(-\pi/R_\theta, \pi/R_\theta)$, ensuring that the fine rotation is indeed a small perturbation, so the model must correctly use it in conjunction with θ_{coarse}.

Provided R_θ is sufficiently small, we can integrate directly with respect to θ_{coarse} when evaluating (6), i.e. sum over all possible rotations. While this 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 (9). The model is therefore relying on the shape embedding $\mathbf{z}$ to model all variability, rather than disentangling shape and pose. The ELBO (6) 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

$$\sum_r^{R_\theta}\left|\left\langle Q_\omega\left(\theta_{\text{coarse}} = r\,|\,\mathbf{x}^{(i)}\right)\right\rangle_i - P\left(\theta_{\text{coarse}} = r\right)\right| = \sum_r^{R_\theta}\left|\left\langle \rho_r^\theta(\mathbf{x}^{(i)})\right\rangle_i - \frac{1}{R_\theta}\right| \tag{13}$$

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

Lighting. For the lighting angle λ, we perform the same decomposition into coarse and fine components as for θ,

giving new variables λ_{coarse} and λ_{fine}, with λ_{coarse} selecting from among R_λ bins. Analogously to pose, λ_{coarse} has a categorical variational distribution parameterised by a softmax output ρ^λ from the encoder, and λ_{fine} has a Gaussian variational distribution with parameters ξ^λ and ζ^λ. Again, we integrate over λ_{coarse}, so the training process reasons over many possible lighting angles for each image, increasing the predicted probability of the one giving the best reconstruction. We also regularise the aggregated posterior distribution of λ_{coarse} towards a uniform distribution.

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

$$\begin{aligned}
\sum_{r_\theta}^{R_\theta}\sum_{r_\lambda}^{R_\lambda}\Bigg\{ &-\Bigg\langle \mathop{\mathbb{E}}_{\mathbf{z},\theta_{\text{fine}},\lambda_{\text{fine}}\sim Q_\omega}\Bigg[\\
&\qquad \log P_\phi\left(\mathbf{x}^{(i)}\,\middle|\,\mathbf{z},\,\theta_{\text{coarse}}=r_\theta,\,\theta_{\text{fine}},\,\lambda_{\text{coarse}}=r_\lambda,\,\lambda_{\text{fine}}\right) \\
&\Bigg]\Bigg\rangle_{i\in\mathscr{B}} \rho^\theta_{r_\theta}\left(\mathbf{x}^{(i)}\right)\rho^\lambda_{r_\lambda}\left(\mathbf{x}^{(i)}\right)\Bigg\} \\
&+\alpha\sum_{r}^{R_\theta}\left\{\left|\left\langle\rho^\theta_r\left(\mathbf{x}^{(i)}\right)\right\rangle_{i\in\mathscr{B}}-\frac{1}{R_\theta}\right|\right\} \\
&+\alpha\sum_{r}^{R_\lambda}\left\{\left|\left\langle\rho^\lambda_r\left(\mathbf{x}^{(i)}\right)\right\rangle_{i\in\mathscr{B}}-\frac{1}{R_\lambda}\right|\right\} \\
&+\beta\left\langle KL\left[Q_\omega\left(\mathbf{z},\theta_{\text{fine}},\lambda_{\text{fine}}\,\middle|\,\mathbf{x}^{(i)}\right)\,\middle|\,\middle|\,P(\mathbf{z})P(\theta_{\text{fine}})P(\lambda_{\text{fine}})\right]\right\rangle_{i\in\mathscr{B}}
\end{aligned} \tag{14}$$

where β increases the relative weight of the KL term as in Higgins et al. (2017), and α controls the strength of the prior-matching terms for pose and lighting. We minimise (14) with respect to ϕ and ω using ADAM (Kingma and Ba, 2015), applying the reparameterisation trick to handle the Gaussian random variables (Kingma and Welling, 2014; Rezende et al., 2014).

Differentiable rendering. Note that optimising (14) by gradient descent requires differentiating through the mesh-rendering operation $\mathscr{G}$ used to calculate $P_\phi(\mathbf{x}\,|\,\mathbf{z},\theta,\lambda)$, to find the derivative of the pixels with respect to the vertex locations and colours. While computing exact derivatives of $\mathscr{G}$ is very expensive, Loper and Black (2014) describe an efficient approximation. We employ a similar technique here, and have made our TensorFlow implementation publicly available[1].

5 Experiments

We follow recent works (Gadelha et al., 2017; Yan et al., 2016; Tulsiani et al., 2017b; Fan et al., 2017; Kato et al., 2018; Tulsiani et al., 2018; Richter and Roth, 2018; Yang et al., 2018) and evaluate our approach using the ShapeNet dataset (Chang et al., 2015). Using synthetic data has two advantages: it allows controlled experiments modifying lighting and other parameters, and it lets us evaluate the reconstruction accuracy using the ground-truth 3D shapes.

Table 1 Reconstruction and pose estimation performance for the ten most-frequent classes in ShapeNet (first ten rows), plus three smooth, concave classes that methods based on voxels and silhouettes cannot handle (last three rows). Metrics: *iou* measures shape reconstruction accuracy when pose supervision is not given; *err* and *acc* measure pose estimation in this case (which requires the model to disentangle shape and pose); *iou* | θ measures shape reconstruction accuracy when pose supervision is given during training. Note that table, lamp, pot, and jar all typically have rotational symmetry, and as such, it is not possible to define an unambiguous reference frame; this results in high values for *err* and low for *acc*. Experimental setting: subdivision, single-view training, fixed colour lighting, shading loss.

	iou *(shape)*	err *(pose)*	acc *(pose)*	iou | θ *(shape)*
table	0.44	89.3	0.39	0.49
chair	0.39	7.9	0.65	0.51
airplane	0.55	1.4	0.90	0.59
car	0.77	4.7	0.84	0.82
sofa	0.59	6.5	0.88	0.71
rifle	0.54	9.0	0.68	0.61
lamp	0.40	87.7	0.19	0.41
vessel	0.48	9.8	0.59	0.58
bench	0.35	5.1	0.71	0.44
loudspeaker	0.41	81.7	0.28	0.54
bathtub	0.54	9.7	0.54	0.57
pot	0.49	90.4	0.20	0.53
jar	0.49	93.1	0.16	0.52

We begin by demonstrating that our method successfully learns to generate and reconstruct 13 different object classes (Sect. 5.1). These include the top ten most frequent classes of ShapeNet, plus three others (*bathtub*, *jar*, and *pot*) that we select because they are smooth and concave, meaning that prior methods using voxels and silhouettes cannot learn and represent them faithfully, as shading information is needed to handle them correctly.

We then rigorously evaluate the performance of our model in different settings, focusing on four classes (*aeroplane*, *car*, *chair*, and *sofa*). The first three are used in Yan et al. (2016), Tulsiani et al. (2017b), Kato et al. (2018), and Tulsiani et al. (2018), while the fourth is a highly concave class that is hard to handle by silhouette-based approaches. We conduct experiments varying the following factors:

- **Mesh parameterisations** (Sect. 5.2): We evaluate the three parameterisations described in Sect. 3: **ortho-block**, **full-block**, and **subdivision**.
- **Single white light vs. three coloured lights** (Sect. 5.3): Unlike previous works using silhouettes (Sect. 2), our method is able to exploit shading in the training images. We test in two settings: (i) illumination by three coloured directional lights (**colour**); and (ii) illumination by one

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

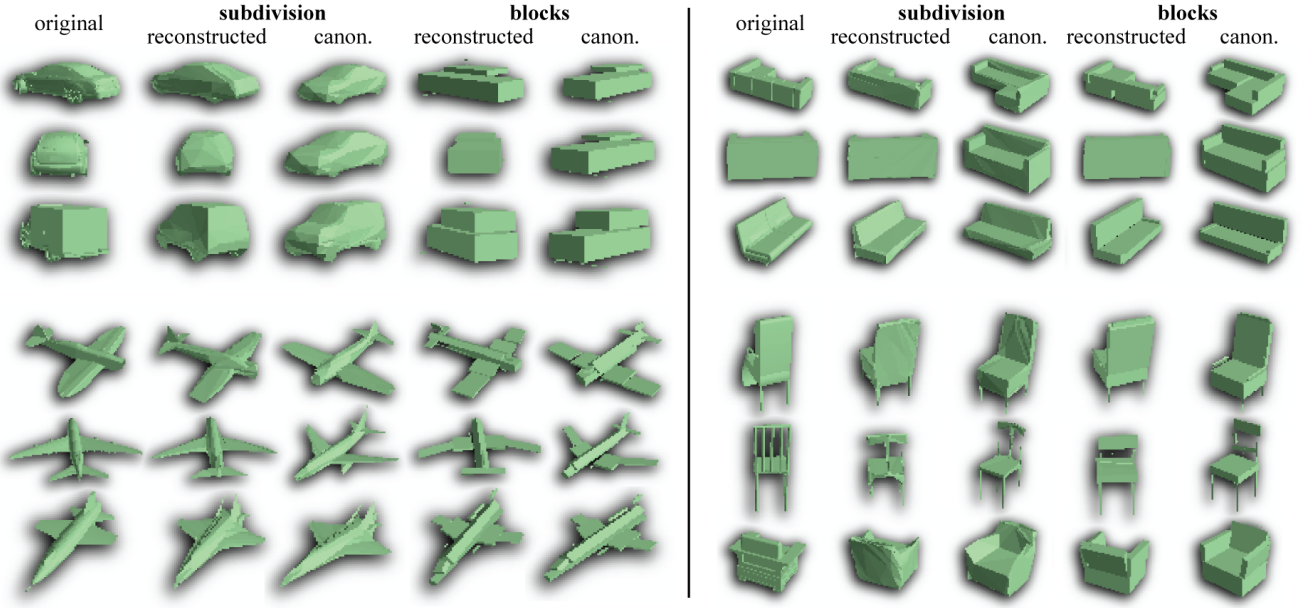


Fig. 8 Qualitative examples of reconstructions, using different mesh parameterisations. Each row of five images shows (i) ShapeNet ground-truth; (ii) our reconstruction with **subdivision** parameterisation; (iii) reconstruction placed in a canonical pose; (iv) our reconstruction with **blocks**; (v) canonical-pose reconstruction. Experimental setting: single-view training, fixed colour lighting, shading loss.

white directional light plus a white ambient component (**white**).

- **Fixed vs. varying lighting** (Sect. 5.3): The variable λ represents a rotation of all the lights together around the vertical axis (Sect. 3). We conduct experiments in two settings: (i) λ is kept fixed across all training and test images, and is known to the generative model (**fixed**); and (ii) λ is chosen randomly for each training/test image, and is not provided to the model (**varying**). In the latter setting, the model must learn to disentangle the effects of lighting angle and surface orientation on the observed shading.
- **Silhouette vs. shading in the loss** (Sect. 5.3): We typically calculate the reconstruction loss (pixel log-likelihood) over the RGB shaded image (**shading**), but for comparison with 2D-supervised silhouette-based methods (Sect. 2) we also experiment with using only the silhouette in the loss (**silhouette**), disregarding differences in shading between the input and reconstructed pixels.
- **Latent space dimensionality** (Sect. 5.4): We experiment with different sizes for the latent shape embedding $\mathbf{z}$, which affects the representational power of our model. We found that 12 dimensions gave good results in initial experiments, and use this value for all experiments apart from Sect. 5.4, where we evaluate its impact.
- **Multiple views** (Sect. 5.5): Yan et al. (2016), Wiles and Zisserman (2017), Tulsiani et al. (2018) and Yang et al. (2018) require that multiple views of each instance are presented together in each training batch, and Tulsiani et al. (2017b) also focus on this setting. Our model does not require this, but for comparison we include results with three views per instance at training time, and either one or three at test time.
- **Pose supervision:** Most previous works that train for 3D reconstruction with 2D supervision require the ground-truth pose of each training instance (Yan et al., 2016; Wiles and Zisserman, 2017; Tulsiani et al., 2017b). While our method does not need this, we evaluate whether it can benefit from it, in each of the settings described above (we report these results in their corresponding sections).

Finally, we compare the performance of our model to several prior and concurrent works on generation and reconstruction, using various degrees of supervision (Sect. 5.6).

Evaluation metrics. We benchmark our reconstruction and pose estimation accuracy on a held-out test set, following the protocol of Yan et al. (2016), where each object is presented at 24 different poses, and statistics are aggregated across objects and poses. We use the following measures:

- *iou*: to measure the shape reconstruction error, we calculate the mean intersection-over-union between the predicted and ground-truth meshes; this follows recent works on reconstruction (e.g. Yan et al., 2016; Tulsiani et al., 2017b). For 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 30° of the ground-truth rotation.

Table 2 Reconstruction performance for four classes, with three different mesh parameterisations (Sect. 3). For each class, the first three columns are in the default setting of no pose supervision and correspond to the metrics in Sect. 5; iou$\mid\theta$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, fixed colour lighting, shading loss.

	car				**chair**				**aeroplane**				**sofa**			
	iou	err	acc	iou$\mid\theta$	iou	err	acc	iou$\mid\theta$	iou	err	acc	iou$\mid\theta$	iou	err	acc	iou$\mid\theta$
ortho-block	0.72	7.6	**0.90**	0.78	0.41	9.2	**0.69**	0.49	0.30	7.9	0.73	0.24	**0.59**	7.3	**0.94**	**0.74**
full-block	0.54	6.5	0.82	0.63	**0.46**	**4.6**	**0.69**	**0.51**	**0.55**	1.7	**0.90**	0.57	0.39	9.1	0.70	0.68
subdivision	**0.77**	**4.7**	0.84	**0.82**	0.39	7.9	0.65	**0.51**	**0.55**	**1.4**	**0.90**	**0.59**	**0.59**	**6.5**	0.88	0.71

Table 3 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; *col+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 2, fixed lighting rotation.

	car				**chair**				**aeroplane**				**sofa**			
	iou	err	acc	iou$\mid\theta$	iou	err	acc	iou$\mid\theta$	iou	err	acc	iou$\mid\theta$	iou	err	acc	iou$\mid\theta$
colour	**0.77**	**4.7**	**0.84**	**0.82**	**0.46**	**4.6**	**0.69**	**0.51**	**0.55**	**1.4**	**0.90**	**0.59**	**0.59**	**7.3**	**0.94**	**0.74**
white	0.58	13.8	0.82	0.81	0.31	37.7	0.43	0.42	0.42	7.7	0.85	0.54	0.51	56.1	0.49	0.71
col+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

Note that the metrics *err* and *acc* are used by Tulsiani et al. (2018) to evaluate pose estimation in a similar setting to ours.

Training minibatches. 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$(-\pi, \pi)$ (and also sampling a lighting angle for experiments with varying lighting). Only these images are used to train the model, not the meshes themselves. In experiments using multiple views, we instead sample 64 meshes and three poses per mesh, and correspondingly render three images.

5.1 Generating and reconstructing diverse object classes

We train a separate model for each of the 13 object classes mentioned above, using **subdivision** parameterisation. Samples generated from these models are shown in Fig. 5. We see that the sampled shapes are realistic, and the models have learnt a prior that encompasses the space of valid shapes for each class. Moreover, the samples are diverse: the models generate various different styles for each class. For example, for *sofa*, both straight and right-angled (modular) designs are sampled; for *aeroplane*, both civilian airliners and military (delta-wing) styles are sampled; for *pot*, square, round, and elongated, forms are sampled; and, for *vessel*, boats both with and without sails are sampled. Note also that our samples incorporate smoothly curved surfaces (e.g. *car*, *jar*) and slanted edges (e.g. *aeroplane*), which voxel-based methods cannot represent (Sect. 5.6 gives a detailed comparison with one such method (Gadelha et al., 2017)).

Reconstruction results are given in Table 1, with qualitative results in Fig. 6. We use fixed colour lighting, shading loss, single-view training, and no pose supervision (columns *iou, err, acc*); we also report *iou* when using pose supervision in column *iou*$\mid\theta$. We see that the highest reconstruction accuracy (*iou*) is achieved for cars, sofas, and aeroplanes, and the lowest for benches, chairs, and lamps. Providing the ground-truth poses as supervision improves reconstruction performance in all cases (*iou*$\mid\theta$). Note that performance for the concave classes sofa, bathtub, pot, and jar is comparable or higher than several non-concave classes, indicating that our model can indeed learn them by exploiting shading cues.

Note that in almost all cases, the reconstructed image is very close to the input (Fig. 6); thus, the model has learnt to reconstruct pixels successfully. Moreover, even when the input is particularly ambiguous due to self-occlusion (e.g. the rightmost car and sofa examples), we see that the model infers a plausible completion of the hidden part of the shape (visible in the third column).

The low values of the pose estimation error *err* (and corresponding high values of *acc*) for most classes indicate that the model has indeed learnt to disentangle pose from shape, without supervision. 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 reference frame, given that it is encouraged by our loss to predict diverse poses. While the pose estimation appears inaccurate for table, lamp, pot, and jar note that these classes exhibit rotational symmetry about the vertical axis. Hence, it is not possible to define (nor indeed to learn) a single, unambiguous canonical frame of reference for them.

Table 4 Reconstruction performance with fixed and varying lighting. In the *varying* case, our model must learn to predict the lighting angle, simultaneously with exploiting the shading cues it provides. Experimental setting: single-view training, best mesh parameterisations from Table 2, shading loss.

		car				**chair**				**aeroplane**				**sofa**			
		iou	err	acc	iou $\mid \theta$	iou	err	acc	iou $\mid \theta$	iou	err	acc	iou $\mid \theta$	iou	err	acc	iou $\mid \theta$
fixed	white	0.58	13.8	0.82	0.81	0.31	37.7	0.43	0.42	0.42	7.7	0.85	0.54	0.51	56.1	0.49	0.71
varying	white	0.48	23.6	0.58	0.79	0.31	31.1	0.47	0.43	0.40	2.5	0.82	0.55	0.47	60.7	0.47	0.71
fixed	colour	0.77	4.7	0.84	0.82	0.46	4.6	0.69	0.51	0.55	1.4	0.90	0.59	0.59	7.3	0.94	0.74
varying	colour	0.60	10.5	0.82	0.79	0.32	36.5	0.42	0.46	0.52	2.4	0.89	0.59	0.69	7.5	0.96	0.73

Table 5 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 2, fixed colour lighting, shading loss.

views		**car**				**chair**			
train	test	iou	err	acc	iou $\mid \theta$	iou	err	acc	iou $\mid \theta$
1	1	0.77	4.7	0.84	0.82	0.46	4.6	0.69	0.51
3	1	0.82	**1.3**	**0.94**	0.83	0.50	**2.1**	**0.83**	0.52
3	3	**0.83**	1.7	**0.94**	**0.84**	**0.53**	3.1	0.80	**0.56**

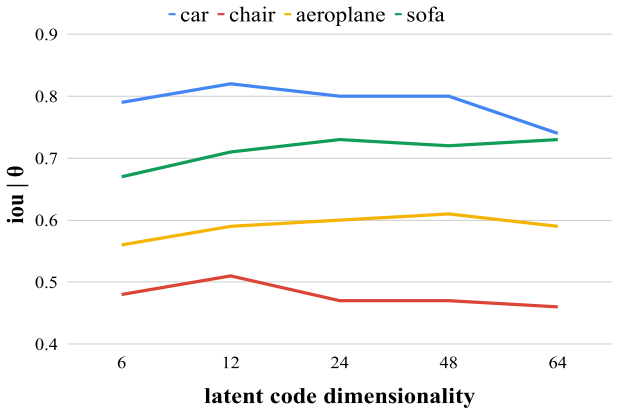


Fig. 9 Effect of varying the dimensionality of the latent embedding vector $\mathbf{z}$ on reconstruction performance (iou $\mid \theta$). Experimental setting: subdivision, fixed colour lighting, shading loss.

5.2 Comparing mesh parameterisations

We now compare the three mesh parameterisations of Sect. 3, considering the four classes *car*, *chair*, *aeroplane*, and *sofa*. We show qualitative results for generation (Fig. 7) and reconstruction (Fig. 8); Table 2 gives quantitative results for reconstruction. Again we use fixed colour lighting, shading loss and single-view training.

We see that different parameterisations are better suited to different classes, in 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 more flexible **full-block** 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 Tulsiani et al. (2017b), Yan et al. (2016), and Fan et al. (2017). On average over the four classes, the best parameterisation is **subdivision**, both with and without pose supervision.

5.3 Lighting

Fixed lighting rotation. Table 3 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 considers just the silhouette in the reconstruction loss, instead of the shaded pixels. To implement it, we differentiably binarise both our reconstructed pixels I_0 and the ground-truth pixels $\mathbf{x}^{(i)}$ prior to calculating the reconstruction loss. Specifically, we transform each pixel p into $p/(p+\eta)$, where η is a small constant. 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* $\mid \theta$ 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 by the high median rotation errors without (column *err*).

Table 6 Reconstruction performance (iou $|\,\theta$) in a setting matching Yan et al. (2016), Tulsiani et al. (2017b), Kato et al. (2018), and Yang et al. (2018), which are silhouette-based methods trained with pose supervision and multiple views (to be precise, Yang et al. (2018) provide pose annotations for 50% of all training images). *PTN, our images* is running the unmodified public code of Yan et al. (2016) with their normal silhouette loss, on our coloured images. N_{views} indicates the number of views of each instance provided together in each minibatch during training. The final rows show performance of two state-of-the-art methods with full 3D supervision (Fan et al., 2017; Richter and Roth, 2018)—note that our colour results are comparable with these, in spite of using only 2D images. Experimental setting: subdivision, three views per object during training, fixed lighting rotation.

	N_{views}	lighting	loss	car	chair	aeroplane	sofa
PTN (Yan et al., 2016)	24	white	silhouette	0.71	**0.50**	0.56	0.62
DRC (Tulsiani et al., 2017b)	5	white	silhouette	0.73	0.43	0.50	-
DRC (Tulsiani et al., 2017b)	5	white	depth	0.74	0.44	0.49	-
NMR (Kato et al., 2018)	2	white	silhouette	0.71	**0.50**	**0.62**	0.67
LPS (Yang et al., 2018)	2	white	silhouette	0.78	0.44	0.57	0.54
PTN, our images	24	colour	silhouette	0.66	0.22	0.42	0.46
ours	3	white	silhouette	0.79	0.46	0.58	0.67
ours	3	white	shading	0.81	0.48	0.60	0.67
ours	3	colour	shading	**0.83**	**0.50**	0.61	**0.73**
PSG (Fan et al., 2017)	-	*white*	*3D*	*0.83*	*0.54*	*0.60*	*0.71*
MN (Richter and Roth, 2018)	-	*white*	*3D*	*0.85*	*0.55*	*0.65*	*0.68*

Table 7 Comparison of our method with the concurrent work MVC (Tulsiani et al., 2018) in different settings, on the three classes for which they report results. Note that they vary elevation as well as azimuth, and their images are rendered with texturing under white light; hence, this comparison to our method is only approximate. Experimental setting: subdivision, three views per object during training, fixed lighting rotation.

	lighting	**loss**	**car**				**chair**				**aeroplane**			
			iou	err	acc	iou $\mid \theta$	iou	err	acc	iou $\mid \theta$	iou	err	acc	iou $\mid \theta$
ours	white	silhouette	0.62	19.4	0.55	0.79	0.45	13.1	0.60	0.46	0.56	1.4	0.83	0.58
ours	white	shading	0.77	3.0	0.91	0.81	0.46	4.2	**0.83**	0.48	0.57	1.0	**0.89**	0.60
ours	colour	shading	**0.82**	**1.3**	**0.94**	**0.83**	**0.47**	**2.7**	0.82	**0.50**	**0.58**	**0.9**	0.88	**0.61**
MVC	white	silhouette	0.74	5.2	0.87	0.75	0.40	7.8	0.81	0.42	0.52	14.3	0.69	0.55
MVC	white	depth	0.71	4.9	0.85	0.69	0.43	8.6	**0.83**	0.45	0.44	21.7	0.60	0.43

Varying lighting rotation. We have shown that shading cues are helpful for training our model. We now evaluate whether it can still learn successfully when the lighting angle varies across training samples (**varying**). Table 4 shows that our method can indeed reconstruct shapes even in this case. When the object pose is given as supervision (column *iou* $|\,\theta$), the reconstruction accuracy is on average only slightly lower than in the case of fixed, known lighting. Thus, the encoder successfully learns to disentangle the lighting angle from the surface normal orientation, while still exploiting the shading information to aid reconstruction. When the object pose is not given as supervision (column *iou*), the model must learn to simultaneously disentangle shape, pose and lighting. Interestingly, even in this extremely hard setting our method still manages to produce good reconstructions, although of course the accuracy is usually lower than with fixed lighting. Finally, note that our results with varying lighting are better than those with fixed lighting from the final row of Table 3, using only the silhouette in the reconstruction loss. This demonstrates that even when the model does not have access to the lighting parameters, it still learns to benefit from shading cues, rather than simply using the silhouette.

5.4 Latent space structure

The shape of a specific object instance must be entirely captured by the latent embedding vector $\mathbf{z}$. On the one hand, using a higher dimensionality for $\mathbf{z}$ should result in better reconstructions, due to the greater representational power. On the other hand, a lower dimensionality makes it easier for the model to learn to map *any* point in $\mathbf{z}$ to a reasonable shape, and to avoid over-fitting the training set. To evaluate this trade-off, we ran experiments with different dimensionalities for $\mathbf{z}$ (Fig. 9). We see that for all classes, increasing from 6 to 12 dimensions improves reconstruction performance on the test set. Beyond 12 dimensions, the effect differs between classes. For car and chair, higher dimensionalities yield lower performance (indicating over-fitting or other training difficulties). Instead, aeroplane and sofa continue to benefit from higher and higher dimensionalities, up to 48 for aeroplane and 64 (and maybe beyond) for sofa.

For all our other experiments, we use a 12-dimensional embedding, as this gives good performance on average across classes. Note that our embedding dimensionality is much smaller than its counterpart in other works. For example, Tulsiani et al. (2017b) have a bottleneck layer with dimen-

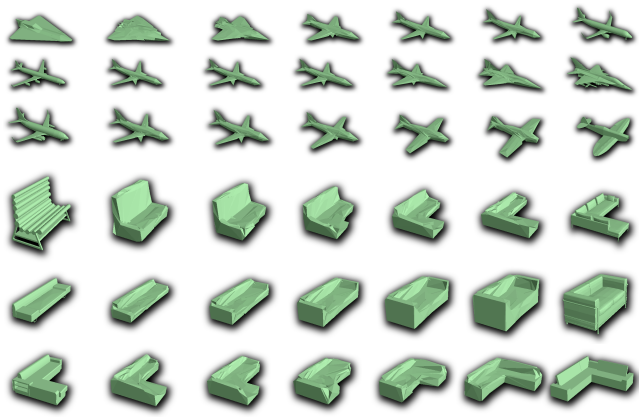


Fig. 10 Interpolating between shapes in latent space. In each row, the leftmost and rightmost images show ground-truth shapes from ShapeNet, and the adjacent columns show the result of reconstructing each using our model with **subdivision** parameterisation. In the centre three columns, we interpolate between the resulting latent embeddings, and display the decoded shapes. In each case, we see a semantically-plausible, gradual deformation of one shape into the other.

sionality 100, while Wiles and Zisserman (2017) use dimensionality 160. This low dimensionality of our embeddings facilitates the encoder mapping images to a compact region of the embedding space centred at the origin; this in turn allows modelling the embeddings by a simple Gaussian from which samples can be drawn.

Interpolating in the latent space. To demonstrate that our models have learnt a well-behaved manifold of shapes for each class, we select pairs of ground-truth shapes, reconstruct these using our model, and linearly interpolate between their latent embeddings (Fig. 10). We see that the resulting intermediate shapes give a gradual, smooth deformation of one shape into the other, showing that all regions of latent space that we traverse correspond to realistic samples.

5.5 Multi-view training/testing

Table 5 shows results when we provide multiple views of each object instance to the model, either at training time only, or during both training and testing. In both cases, this improves results over using just a single view—the model has learnt to exploit the additional information about each instance. Note that when training with three views but testing with one, the network has not been optimised for the single-view task; however, the additional information present during training means it has learnt a stronger model of valid shapes, and this knowledge transfers to the test-time scenario of reconstruction from a single image.

5.6 Comparison to previous and concurrent works

Generation. Fig. 11 compares samples from our model, to samples from that of Gadelha et al. (2017), on the four object classes we have in common. This is the only prior work that trains a 3D generative model using only single views of instances, and without pose supervision. Note however that unlike us, all images in the training set of Gadelha et al. (2017) are taken from one of a fixed set of eight poses, making their task a little easier. We manually selected samples from our model that are stylistically similar to those shown in Gadelha et al. (2017) to allow side-by-side comparison. We see that in all cases, generating meshes tends to give cleaner, more visually-pleasing samples than their use of voxels. For *chair*, our model is able to capture the very narrow legs; for *aeroplane*, it captures the diagonal edges of the wings; for *car* and *vase*, it captures the smoothly curved edges. Note that as shown in Fig. 5, our model also successfully learns models for concave classes such as *bathtub* and *sofa*—which is impossible for Gadelha et al. (2017) as they do not consider shading.

Reconstruction. Table 6 compares our results with previous and concurrent 2D-supervised methods that input object pose at training time. We consider works that appeared in 2018 to be concurrent to ours (Henderson and Ferrari, 2018). Here, we conduct experiments in a setting matching Yan et al. (2016), Tulsiani et al. (2017b), Kato et al. (2018), and Yang et al. (2018): multiple views at training time, with ground-truth pose supervision (given for 50% of images in Yang et al. (2018)).

Even when using only silhouettes during training, our results are about as good as the best of the works we compare to, that of Kato et al. (2018), which is a concurrent work. Our results are somewhat worse than theirs for aeroplanes and chairs, better for cars, and identical for sofas. On average over the four classes, we reach the same iou of 62.5%. When we add shading information to the loss, our results show a significant improvement. Importantly, Yan et al. (2016), Tulsiani et al. (2017b) and Yang et al. (2018) cannot exploit shading, as they are based on voxels. Coloured lighting helps all classes even further, leading to a final performance higher than than all other methods on car and sofa, and comparable to the best other method on chair and aeroplane (Kato et al., 2018). On average we reach 66.8% iou, compared to 62.5% for Kato et al. (2018).

We also show results for Yan et al. (2016) using our coloured lighting images as input, but their silhouette loss. This performs worse than our method on the same images, again showing that incorporating shading in the loss is useful—our colour images are not simply more informative to the encoder network than those of Yan et al. (2016). Interestingly, when trained with shading or colour, our method outper-

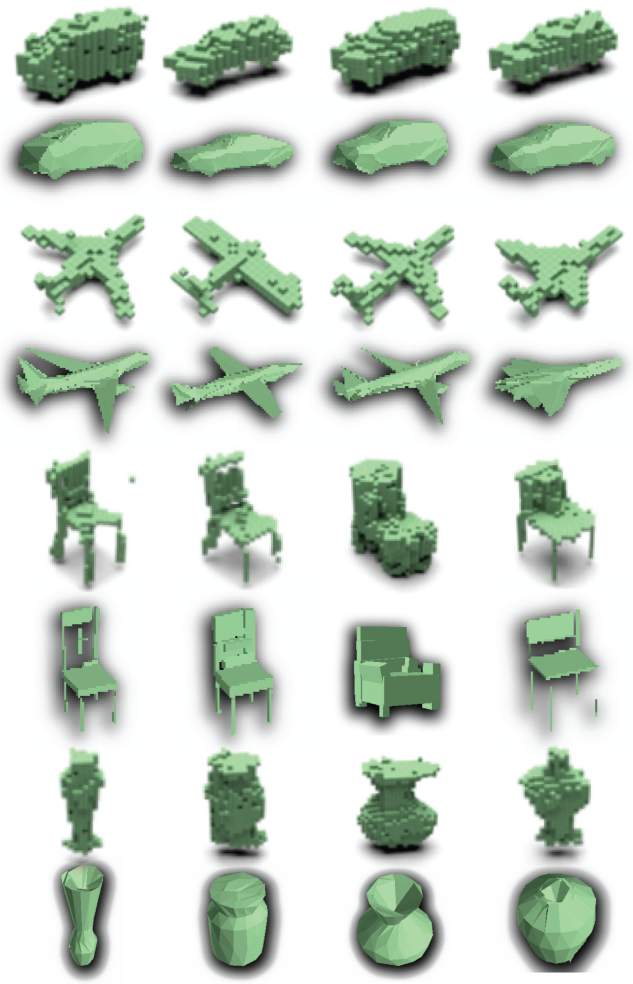


Fig. 11 Samples from the voxel-based method of Gadelha et al. (2017) (odd rows), shown above stylistically-similar samples from our model (even rows). Both methods are trained with a single view per instance, and without pose annotations. However, our model outputs meshes, and uses shading in the loss; hence, it can represent smooth surfaces and learn concave classes such as *vase*.

forms Tulsiani et al. (2017b) even when the latter is trained with depth information. When trained with colour, our results (average 66.8% iou) are even close to those of Fan et al. (2017) (67.0%) and Richter and Roth (2018) (68.2%), which are state-of-the-art methods trained with full 3D supervision.

Table 7 compares our results with those of Tulsiani et al. (2018). This is a concurrent work similar in spirit to our own, that learns reconstruction and pose estimation without 3D supervision nor pose annotations, but requires multiple views of each instance to be presented together during training. We match their experimental setting by training our models on three views per instance; however, they vary elevation as well as azimuth during training, making their task a little harder. We see that the ability of our model to exploit shading cues enables it to significantly outperform Tulsiani et al. (2018), which relies on silhouettes in its loss. This is shown by *iou* and $iou \mid \theta$ being higher for our method with white light and shading loss, than for theirs with white light and silhouette. Indeed, our method outperforms theirs even when they use depth information as supervision. When we use colour lighting, our performance is even higher, due to the stronger information about surface normals. Conversely, when our method is restricted to silhouettes, it performs significantly worse than theirs across all three object classes.

6 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). When pose supervision is not provided, it automatically learns to disentangle the effects of shape and pose on the final image. When the lighting is unknown, it also learns to disentangle the effects of lighting and surface orientation on the shaded pixels. We have shown that exploiting shading cues leads to higher performance than state-of-the-art methods based on silhouettes (Kato et al., 2018). It also allows our model to learn concave classes, unlike these prior works. Moreover, our performance is higher than that of methods with depth supervision (Tulsiani et al., 2017b, 2018), and even close to the state-of-the-art results using full 3D supervision (Fan et al., 2017; Richter and Roth, 2018). 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 (Gadelha et al., 2017).

A Network architectures

In this appendix we briefly describe the architectures of the decoder and encoder neural networks.

The decoder network F_ϕ takes the latent embedding $\mathbf{z}$ as input. This is passed through a fully-connected layer with 32 output channels using ReLU activation. The resulting embedding is processed by a second fully-connected layer that outputs the mesh parameters: vertex offsets for subdivision parameterisation, and locations, scales and rotations for the primitive-based parameterisations. For the primitive scales, we use a softplus activation to ensure they are positive; for the other parameters, we do not use any activation function.

The encoder network $\text{enc}_\omega(\mathbf{x})$ is a CNN operating on RGB images of size 128×96 pixels; its architecture is similar to that of Wiles and Zisserman (2017). Specifically, it has the following layers, each with batch normalisation and ReLU activation:

- 3×3 convolution, 32 channels, stride = 2
- 3×3 convolution, 64 channels, stride = 1
- 2×2 max-pooling, stride = 2
- 3×3 convolution, 96 channels, stride = 1
- 2×2 max-pooling, stride = 2
- 3×3 convolution, 128 channels, stride = 1
- 2×2 max-pooling, stride = 2

– 4×4 convolution, 128 channels, stride = 1
– fully-connected, 128 channels

This yields a 128-dimensional feature vector for the image. The parameters for each variational distribution are produced by a further fully-connected layer, each taking this feature vector as input. For the mean of $\mathbf{z}$, we do not use any activation function; for the mean of θ_{fine} we use tanh activation, scaled by π/R_θ to ensure θ_{coarse} rather than θ_{fine} is used to model large rotations. For the mean of λ_{fine} we analogously use tanh activation scaled by π/R_λ. For the standard deviations of $\mathbf{z}$, θ_{fine}, and λ_{fine}, we use softplus activation, to ensure they are positive. Finally, for θ_{coarse} and λ_{coarse}, we use softmax outputs giving the probabilities of the different coarse rotations.

References

Achlioptas P, Diamanti O, Mitliagkas I, Guibas L (2018) Learning representations and generative models for 3D point clouds. In: International Conference on Machine Learning

Balashova E, Singh V, Wang J, Teixeira B, Chen T, Funkhouser T (2018) Structure-aware shape synthesis. In: 3DV

Barron JT, Malik J (2015) Shape, illumination, and reflectance from shading. IEEE Transactions on Pattern Analysis and Machine Intelligence 37(8):1670–1687

Broadhurst A, Drummond TW, Cipolla R (2001) A probabilistic framework for space carving. In: Proceedings of the International Conference on Computer Vision

Burt PJ, Adelson EH (1983) The laplacian pyramid as a compact image code. IEEE Trans on Communications COM-31(4):532–540

Chang AX, Funkhouser T, Guibas L, Hanrahan P, Huang Q, Li Z, Savarese S, Savva M, Song S, Su H, Xiao J, Yi L, Yu F (2015) ShapeNet: An Information-Rich 3D Model Repository. arXiv preprint arXiv:1512:03012

Choy CB, Xu D, Gwak J, Chen K, Savarese S (2016) 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In: Proceedings of the European Conference on Computer Vision

De Bonet JS, Viola P (1999) Roxels: Responsibility weighted 3D volume reconstruction. In: Proceedings of the International Conference on Computer Vision

Fan H, Su H, Guibas L (2017) A point set generation network for 3D object reconstruction from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

Furukawa Y, Hernández C (2015) Multi-view stereo: A tutorial. Foundations and trends® in Computer Graphics and Vision 9(1–2):1–148

Gadelha M, Maji S, Wang R (2017) 3D shape induction from 2D views of multiple objects. In: 3DV

Gadelha M, Wang R, Maji S (2018) Multiresolution tree networks for 3D point cloud processing. In: Proceedings of the European Conference on Computer Vision

Gargallo P, Sturm P, Pujades S (1999) An occupancy-depth generative model of multi-view images. In: Proceedings of the International Conference on Computer Vision

Girdhar R, Fouhey D, Rodriguez M, Gupta A (2016) Learning a predictable and generative vector representation for objects. In: Proceedings of the European Conference on Computer Vision

Goodfellow I, Pouget-Abadle J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Advances in Neural Information Processing Systems

Gouraud H (1971) Continuous shading of curved surfaces. IEEE Trans on Computers C-20(6):623–629

Gwak J, Choy CB, Chandraker M, Garg A, Savarese S (2017) Weakly supervised 3D reconstruction with adversarial constraint. In: 3DV

Henderson P, Ferrari V (2018) Learning to generate and reconstruct 3D meshes with only 2D supervision. In: Proceedings of the British Machine Vision Conference

Higgins I, Matthey L, Pal A, Burgess C, Glorot X, Botvinick M, Mohamed S, Lerchner A (2017) β-VAE: Learning basic visual concepts with a constrained variational framework. In: International Conference on Learning Representations

Horn B (1975) Obtaining shape from shading information. In: Winston PH (ed) The Psychology of Computer Vision

Huang H, Kalogerakis E, Marlin B (2015) Analysis and synthesis of 3D shape families via deep-learned generative models of surfaces. Computer Graphics Forum 34(5):25–38

Insafutdinov E, Dosovitskiy A (2018) Unsupervised learning of shape and pose with differentiable point clouds. In: Advances in Neural Information Processing Systems

Kanazawa A, Tulsiani S, Efros AA, Malik J (2018) Learning category-specific mesh reconstruction from image collections. In: Proceedings of the European Conference on Computer Vision

Kar A, Tulsiani S, Carreira J, Malik J (2015) Category-specific object reconstruction from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

Kato H, Ushiku Y, Harada T (2018) Neural 3D mesh renderer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

Kingma DP, Ba JL (2015) Adam: A method for stochastic optimization. In: International Conference on Learning Representations

Kingma DP, Welling M (2014) Auto-Encoding Variational Bayes. In: International Conference on Learning Representations

Lambert JH (1760) Photometria. Eberhard Klett Verlag

Laurentini A (1994) The visual hull concept for silhouette-based image understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(2):150–162

Li J, Xu K, Chaudhuri S, Yumer E, Zhang H, Guibas L (2017) GRASS: Generative recursive autoencoders for shape structures. ACM Transactions on Graphics 36(4)

Liu S, Cooper DB (2010) Ray markov random fields for image-based 3D modeling: model and efficient inference. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

Loper MM, Black MJ (2014) OpenDR: An approximate differentiable renderer. In: Proceedings of the European Conference on Computer Vision, pp 154–169

Mandikal P, Murthy N, Agarwal M, Babu RV (2018) 3D-LMNet: Latent embedding matching for accurate and diverse 3D point cloud reconstruction from a single image. In: Proceedings of the British Machine Vision Conference

Mousavian A, Anguelov D, Flynn J, Kosecka J (2017) 3D bounding box estimation using deep learning and geometry. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

Nash C, Williams CKI (2017) The shape variational autoencoder: A deep generative model of part-segmented 3D objects. Computer Graphics Forum 36(5):1–12

Niu C, Li J, Xu K (2018) Im2Struct: Recovering 3D shape structure from a single RGB image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

Novotny D, Larlus D, Vedaldi A (2017) Learning 3d object categories by looking around them. In: Proceedings of the International Conference on Computer Vision, pp 5218–5227

Rezende DJ, Mohamed S, Wierstra D (2014) Stochastic backpropagation and approximate inference in deep generative models. In: International Conference on Machine Learning

Rezende DJ, Eslami SMA, Mohamed S, Battaglia P, Jaderberg M, Heess N (2016) Unsupervised learning of 3D structure from images. In: Advances in Neural Information Processing Systems

Richter SR, Roth S (2018) Matryoshka networks: Predicting 3D geometry via nested shape layers. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1936–1944

Seitz S, Curless B, Diebel J, Scharstein D, Szeliski R (2006) A comparison and evaluation of multi-view stereo reconstruction algorithms.

In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

Shin D, Fowlkes CC, Hoiem D (2018) Pixels, voxels, and views: A study of shape representations for single view 3D object shape prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

Soltani AA, Huang H, Wu J, Kulkarni TD, Tenenbaum JB (2017) Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

Tan Q, Gao L, Yu-Kun Lai SX (2018) Variational autoencoders for deforming 3d mesh models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

Tulsiani S, Su H, Guibas LJ, Efros AA, Malik J (2017a) Learning shape abstractions by assembling volumetric primitives. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

Tulsiani S, Zhou T, Efros AA, Malik J (2017b) Multi-view supervision for single-view reconstruction via differentiable ray consistency. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

Tulsiani S, Efros AA, Malik J (2018) Multi-view consistency as supervisory signal for learning shape and pose prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

Vicente S, Carreira J, Agapito L, Batista J (2014) Reconstructing PASCAL VOC. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

Wang N, Zhang Y, Li Z, Fu Y, Liu W, Jiang YG (2018) Pixel2Mesh: Generating 3D mesh models from single RGB images. In: Proceedings of the European Conference on Computer Vision

Wiles O, Zisserman A (2017) SilNet: Single- and multi-view reconstruction by learning from silhouettes. In: Proceedings of the British Machine Vision Conference

Wu J, Zhang C, Xue T, Freeman WT, Tenenbaum JB (2016) Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In: Advances in Neural Information Processing Systems

Wu Z, Song S, Khosla A, Yu F, Zhang L, Tang X, Xiao J (2015) 3D ShapeNets: A deep representation for volumetric shape modeling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

Xie J, Zheng Z, Gao R, Wang W, Zhu SC, Wu YN (2018) Learning descriptor networks for 3d shape synthesis and analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

Yan X, Yang J, Yumer E, Guo Y, Lee H (2016) Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. In: Advances in Neural Information Processing Systems

Yang G, Cui Y, Belongie S, Hariharan B (2018) Learning single-view 3D reconstruction with limited pose supervision. In: Proceedings of the European Conference on Computer Vision

Zhang R, Tsai PS, Cryer JE, Shah M (1999) Shape-from-shading: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 21(8):690–706

Zhu JY, Zhang Z, Zhang C, Wu J, Torralba A, Tenenbaum J, Freeman B (2018) Visual object networks: Image generation with disentangled 3D representations. In: Advances in Neural Information Processing Systems

Zhu R, Kiani Galoogahi H, Wang C, Lucey S (2017) Rethinking reprojection: Closing the loop for pose-aware shape reconstruction from a single image. In: Proceedings of the International Conference on Computer Vision

Zou C, Yumer E, Yang J, Ceylan D, Hoiem D (2017) 3D-PRNN: Generating shape primitives with recurrent neural networks. In: Proceedings of the International Conference on Computer Vision