

# HoloGAN: Unsupervised Learning of 3D Representations From Natural Images

Thu Nguyen-Phuoc <sup>1</sup> Chuan Li <sup>2</sup> Lucas Theis <sup>3</sup> Christian Richardt <sup>1</sup> Yong-Liang Yang <sup>1</sup>

<sup>1</sup> University of Bath <sup>2</sup> Lambda Labs <sup>3</sup> Twitter



Figure 1. HoloGAN learns to separate pose from identity (shape and appearance) only from unlabelled 2D images without sacrificing the visual fidelity of the generated images. All results shown here are sampled from HoloGAN for the same identities in each row but in different poses.

## Abstract

We propose a novel generative adversarial network (GAN) for the task of unsupervised learning of 3D representations from natural images. Most generative models rely on 2D kernels to generate images and make few assumptions about the 3D world. These models therefore tend to create blurry images or artefacts in tasks that require a strong 3D understanding, such as novel-view synthesis. HoloGAN instead learns a 3D representation of the world, and to render this representation in a realistic manner. Unlike other GANs, HoloGAN provides explicit control over the pose of generated objects through rigid-body transformations of the learnt 3D features. Our experiments show that using explicit 3D features enables HoloGAN to disentangle 3D pose and identity, which is further decomposed into shape and appearance, while still being able to generate images with similar or higher visual quality than other generative models. HoloGAN can be trained end-to-end from unlabelled 2D images only. In particular, we do not require pose labels, 3D shapes, or multiple views of the same objects. This shows that HoloGAN is the first generative model that learns 3D representations from natural images in an entirely unsupervised manner.

## 1. Introduction

Learning to understand the relationship between 3D objects and 2D images is an important topic in computer vision and computer graphics. In computer vision, it has applications in fields such as robotics, autonomous vehicles or security. In computer graphics, it benefits applications in both content generation and manipulation.

This ranges from photorealistic rendering of 3D scenes or sketch-based 3D modelling, to novel-view synthesis or relighting.

Recent generative image models, in particular, generative adversarial networks (GANs), have achieved impressive results in generating images of high resolution and visual quality [1, 5, 27, 28, 62], while their conditional versions have achieved great progress in image-to-image translation [23, 50], image editing [11, 12, 60] or motion transfer [6, 30]. However, GANs are still fairly limited in their applications, since they do not allow explicit control over attributes in the generated images, while conditional GANs need labels during training (Figure 2 left), which are not always available.

Even when endowed with labels like pose information, current generative image models still struggle in tasks that require a fundamental understanding of 3D structures, such as novel-view synthesis from a single image. For example, using 2D kernels to perform 3D operations, such as out-of-plane rotation to generate novel views, is very difficult. Current methods either require a lot of labelled training data, such as multi-view images or segmentation masks [41, 52], or produce blurry results [13, 14, 33, 54, 57]. Although recent work has made efforts to address this problem by using 3D data [40, 64], 3D ground-truth data are very expensive to capture and reconstruct. Therefore, there is also a practical motivation to directly learn 3D representations from unlabelled 2D images.

Motivated by these observations, we focus on designing a novel architecture that allows unsupervised learning of 3D representations from images, enabling direct manipulation of view, shape and appearance in generative image models (Figure 1). The key insight of our network design is the combination of a strong inductive bias about the 3D world with deep generative models to

learn better representations for downstream tasks. Conventional representations in computer graphics, such as voxels and meshes, are *explicit* in 3D and *easy to manipulate* via, for example, rigid-body transformations. However, they come at the cost of memory inefficiency or ambiguity in how to discretise complex objects. As a result, it is non-trivial to build generative models with such representations directly [43, 46, 49]. *Implicit* representations, such as high-dimensional latent vectors or deep features, are favoured by generative models for being *spatially compact* and *semantically expressive*. However, these features are not designed to work with explicit 3D transformations [9, 14, 20, 25, 45], which leads to visual artefacts and blurriness in tasks such as view manipulation.

We propose HoloGAN, an unsupervised generative image model that learns representations of 3D objects that are not only explicit in 3D but also semantically expressive. Such representations can be learnt directly from unlabelled natural images. Unlike other GAN models, HoloGAN employs both 3D and 2D features for generating images. HoloGAN first learns a 3D representation, which is then transformed to a target pose, projected to 2D features, and rendered to generate the final images (Figure 2 right). Different from recent work that employs hand-crafted differentiable renderers [18, 22, 29, 34, 36, 51, 64], HoloGAN learns perspective projection and rendering of 3D features from scratch using a projection unit [40]. This novel architecture enables HoloGAN to learn 3D representations directly from natural images for which there are no good hand-crafted differentiable renderers. To generate new views of the same scene, we directly apply 3D rigid-body transformations to the learnt 3D features, and visualise the results using the neural renderer that is jointly trained. This has been shown to produce sharper results than performing 3D transformations in high-dimensional latent vector space [40].

HoloGAN can be trained end-to-end in an unsupervised manner using only unlabelled 2D images, without any supervision of poses, 3D shapes, multiple views of objects, or geometry priors such as symmetry and smoothness over the 3D representation that are common in this line of work [3, 26, 45]. To the best of our knowledge, HoloGAN is the first generative model that can learn 3D representations directly from *natural* images in a purely unsupervised manner. In summary, our main technical contributions are:

- A novel architecture that combines a strong inductive bias about the 3D world with deep generative models to learn disentangled representations (pose, shape, and appearance) of 3D objects from images. The representation is explicit in 3D and expressive in semantics.
- An unconditional GAN that, for the first time, allows native support for view manipulation without sacrificing visual image fidelity.
- An unsupervised training approach that enables disentangled representation learning without using labels.

## 2. Related work

HoloGAN is at the intersection of GANs, structure-aware image synthesis and disentangled representation learning. In this section, we review related work in these areas.

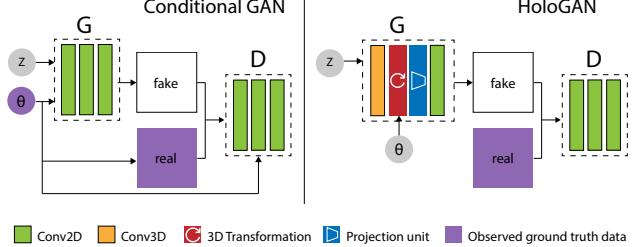


Figure 2. Comparison of generative image models. Data given to the discriminator are coloured purple. **Left:** In conditional GANs, the pose is observed and the discriminator is given access to this information. **Right:** HoloGAN does not require pose labels during training and the discriminator is not given access to pose information.

### 2.1. Generative adversarial networks

GANs learn to map samples from an arbitrary latent distribution to data that fool a discriminator network into categorising them as real data [16]. Most recent work on GAN architectures has focused on improving training stability or visual fidelity of the generated images, such as multi-resolution GANs [27, 61], or self-attention generators [1, 62]. However, there is far less work on designing GAN architectures that enable unsupervised disentangled representation learning which allows control over attributes of the generated images. By injecting random noise and adjusting the “style” of the image at each convolution, StyleGAN [28] can separate fine-grained variation (e.g., hair, freckles) from high-level features (e.g., pose, identity), but does not provide explicit control over these elements. A similar approach proposed by Chen et al. [8] shows that this network design also achieves more training stability. The success of these approaches indicates that network architecture can be more important for training stability and image fidelity than the specific choice of GAN loss. Therefore, we also focus on architecture design for HoloGAN, but with the goal of learning to separate pose, shape and appearance, and to enable direct manipulation of these elements.

### 2.2. 3D-aware neural image synthesis

Recent work in neural image synthesis and novel-view synthesis has found success in improving the fidelity of the generated images with 3D-aware networks. Work that uses geometry templates achieves great improvements in image fidelity [15, 32], but does not generalise well to complex datasets that cannot be described by a template. RenderNet [40] introduces a differentiable renderer using a convolutional neural network (CNN) that learns to render 2D images directly from 3D shapes. However, RenderNet requires 3D shapes and their corresponding rendered images during training. Other approaches learn 3D embeddings that can be used to generate new views of the same scene without any 3D supervision [48, 51]. However, while Sitzmann et al. [51] require multiple views and pose information as input, Rhodin et al. [48] require supervision from paired images, background segmentation and pose information. Aiming to separate geometry from texture, visual object networks (VONs) [64] first sample 3D objects from a 3D generative model, render these objects using a hand-crafted differentiable layer to normal, depth and silhouette maps, and finally apply a trained image-to-image translation network. How-

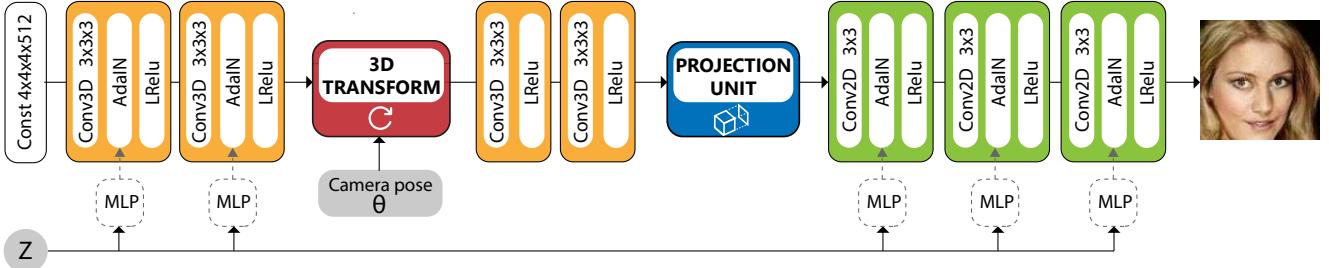


Figure 3. HoloGAN’s generator network: we employ 3D convolutions, a 3D rigid-body transformation, the projection unit and 2D convolutions. We also remove the traditional input layer from  $\mathbf{z}$ , and start from a learnt constant 4D tensor. The latent vector  $\mathbf{z}$  is instead fed through multilayer perceptrons (MLPs) to map to the affine transformation parameters for adaptive instance normalisation (AdaIN). Inputs are coloured gray.

ever, VONs need explicit 3D data for training, and only work for single-object images with a simple white background. Our HoloGAN also learns a 3D representation and renders it to produce a 2D image but without seeing any 3D shapes, and works with real images containing complex backgrounds and multi-object scenes.

The work closest to ours is Pix2Scene [45], which learns an implicit 3D scene representation from images, also in an unsupervised manner. However, this method maps the implicit representation to a surfel representation for rendering, while HoloGAN uses an explicit 3D representation with deep voxels. Additionally, using a hand-crafted differentiable renderer, Pix2Scene can only deal with simple synthetic images (uniform material and lighting conditions). HoloGAN, on the other hand, learns to render from scratch, and thus works for more complex natural images.

### 2.3. Disentangled representation learning

The aim of disentangled representation learning is to learn a factorised representation, in which changes in one factor only affect the corresponding elements in the generated images, while being invariant to other factors. Most work in disentangled learning leverages labels provided by the dataset [2, 47, 55] or benefits from set supervision (e.g., videos or multiple images of the same scene; more than two domains with the same attributes) [10, 14, 33].

Recent efforts in unsupervised disentangled representation learning, such as  $\beta$ -VAE [20] or InfoGAN [19, 24], have focused mostly on designing loss functions. However, these models are sensitive to the choice of priors, provide no control over what factors are learned, and do not guarantee that the learnt disentangled factors are semantically meaningful. Moreover,  $\beta$ -VAE comes with a trade-off between the quality of the generated images and the level of disentanglement. Finally, these two methods struggle with more complicated datasets (natural images with complex backgrounds and lighting). In contrast, by redesigning the architecture of the generator network, HoloGAN learns to successfully separate pose, shape and appearance, as well as providing explicit pose control and enables shape/appearance editing even for more complex natural image datasets.

## 3. Method

To learn 3D representations from 2D images without labels, HoloGAN extends traditional unconditional GANs by introducing a strong inductive bias about the 3D world into the generator

network. Specifically, HoloGAN generates images by learning a 3D representation of the world and to render it realistically such that it fools the discriminator. View manipulation therefore can be achieved by directly applying 3D rigid-body transformations to the learnt 3D features. In other words, the images created by the generator are a *view-dependent* mapping from a learnt 3D representation to the 2D image space. This is different from other GANs which learn to map a noise vector  $\mathbf{z}$  directly to 2D features to generate images.

Figure 3 illustrates the generator architecture of HoloGAN: HoloGAN first learns a 3D representation (assumed to be in a canonical pose) using 3D convolutions (Section 3.1), transforms this representation to a certain pose, projects and computes visibility using the projection unit (Section 3.2), and computes shaded colour values for each pixel in the final images with 2D convolutions. HoloGAN shares many rendering insights with RenderNet [40], but works with natural images, and needs neither pre-training of the neural renderer nor paired 3D shape–2D image training data.

During training, we sample random poses from a uniform distribution and transform the 3D features using these poses before rendering them to images. We assume every image has a corresponding single global pose, and show that this assumption still works with images of multiple objects. This random pose perturbation pushes the generator network to learn a disentangled representation that is suitable for both 3D transformation and generating images that can fool the discriminator. While pose transformation could be learnt from data, we provide this operation, which is differentiable and straightforward to implement, explicitly to HoloGAN. Using explicit rigid-body transformations for novel-view synthesis has been shown to produce sharper images with fewer artefacts [40]. More importantly, this provides an inductive bias towards representations that are compatible with *explicit* 3D rigid-body transformations. As a result, the learnt representations are explicit in 3D and disentangled between pose and identity.

Kulkarni et al. [33] categorize the learnt disentangled representation into *intrinsic* and *extrinsic* elements. While *intrinsic elements* describe shape, appearance, etc., *extrinsic elements* describe pose (elevation, azimuth) and lighting (location, intensity). The design of HoloGAN naturally lends itself to this separation by using more inductive biases about the 3D world: the adoption of a native 3D transformation, which controls the pose (shown as  $\theta$  in Figure 3) directly, to the learnt 3D features, which control the identity (shown as  $\mathbf{z}$  in Figure 3).

### 3.1. Learning 3D representations

HoloGAN generates 3D representations from a learnt constant tensor (see Figure 3). The random noise vector  $\mathbf{z}$  instead is treated as a “style” controller, and mapped to affine parameters for adaptive instance normalization (AdaIN) [21] after each convolution using a multilayer perceptron (MLP)  $f: \mathbf{z} \rightarrow \gamma(\mathbf{z}), \sigma(\mathbf{z})$ .

Given some features  $\Phi_l$  at layer  $l$  of an image  $\mathbf{x}$  and the noise “style” vector  $\mathbf{z}$ , AdaIN is defined as:

$$\text{AdaIN}(\Phi_l(\mathbf{x}), \mathbf{z}) = \sigma(\mathbf{z}) \left( \frac{\Phi_l(\mathbf{x}) - \mu(\Phi_l(\mathbf{x}))}{\sigma(\Phi_l(\mathbf{x}))} \right) + \gamma(\mathbf{z}). \quad (1)$$

This can be viewed as generating images by transformation of a template (the learnt constant tensor) using AdaIN to match the mean and standard deviation of the features at different levels  $l$  (which are believed to describe the image “style”) of the training images. Empirically, we find this network architecture can disentangle pose and identity much better than those that feed the noise vector  $\mathbf{z}$  directly to the first layer of the generator.

HoloGAN inherits this style-based strategy from StyleGAN [28] but is different in two important aspects. Firstly, HoloGAN learns 3D features from a learnt 4D constant tensor (size  $4 \times 4 \times 4 \times 512$ , where the last dimension is the feature channel) before projecting them to 2D features to generate images, while StyleGAN only learns 2D features. Secondly, HoloGAN learns a disentangled representation by combining 3D features with rigid-body transformations during training, while StyleGAN injects independent random noise into each convolution. StyleGAN, as a result, learns to separate 2D features into different levels of detail, depending on the feature resolution, from coarse (e.g., pose, identity) to more fine-grained details (e.g., hair, freckles). We observe a similar separation in HoloGAN. However, HoloGAN further separates pose (controlled by the 3D transformation), shape (controlled by 3D features), and appearance (controlled by 2D features).

It is worth highlighting that to generate images at  $128 \times 128$  (same as VON), we used a deep 3D representation of size up to  $16 \times 16 \times 16 \times 64$ . Even with such limited resolution, HoloGAN can still generate images with competitive quality and more complex backgrounds than other methods that use full 3D geometry such as VON’s voxel grid of resolution  $128 \times 128 \times 128 \times 1$  [64].

### 3.2. Learning with view-dependent mappings

In addition to adopting 3D convolutions to learn 3D features, during training, we introduce more bias about the 3D world by transforming these learnt features to random poses before projecting them to 2D images. This random pose transformation is crucial to guarantee that HoloGAN learns a 3D representation that is disentangled and can be rendered from all possible views, as also observed by Tran et al. [55] in DR-GAN. However, HoloGAN performs explicit 3D rigid-body transformation, while DR-GAN performs this using an implicit vector representation.

**Rigid-body transformation** We assume a virtual pinhole camera that is in the canonical pose (axis-aligned and placed along the negative z-axis) relative to the 3D features being rendered. We parameterise the rigid-body transformation by 3D rotation, scaling followed by trilinear resampling. Although translation is

inherently supported by our framework, we did not use it in this work. Assuming the up-vector of the object coordinate system is the global y-axis, rotation comprises rotation around the y-axis (azimuth) and the x-axis (elevation). Details on ranges for pose sampling are included in the supplemental document.

**Projection unit** In order to learn meaningful 3D representations from just 2D images, HoloGAN learns a differentiable projection unit [40] that reasons over occlusion. In particular, the projection unit receives a 4D tensor (3D features), and returns a 3D tensor (2D features).

Since the training images are captured with different perspectives, HoloGAN needs to learn perspective projection. However, as we have no knowledge of the camera intrinsics, we employ two layers of 3D convolutions (without AdaIN) to morph the 3D representation into a perspective frustum (see Figure 3) before their projection to 2D features.

The projection unit is composed of a reshaping layer that concatenates the channel dimension with the depth dimension, thus reducing the tensor dimension from 4D ( $W \times H \times D \times C$ ) to 3D ( $W \times H \times (D \cdot C)$ ), and an MLP with a non-linear activation function (leakyReLU [37] in our experiments) to learn occlusion.

### 3.3. Loss functions

**Identity regulariser** To generate images at higher resolution ( $128 \times 128$  pixels), we find it beneficial to add an identity regulariser  $L_{\text{identity}}$  that ensures a vector reconstructed from a generated image matches the latent vector  $\mathbf{z}$  used in the generator  $G$ . We find that this encourages HoloGAN to only use  $\mathbf{z}$  for the identity to maintain the object’s identity when poses are varied, helping the model learn the full variation of poses in the dataset. We introduce an encoder network  $F$  that shares the majority of the convolution layers of the discriminator, but uses an additional fully-connected layer to predict the reconstructed latent vector. The identity loss is:

$$L_{\text{identity}}(G) = \mathbb{E}_{\mathbf{z}} \|\mathbf{z} - F(G(\mathbf{z}))\|^2. \quad (2)$$

**Style discriminator** Our generator is designed to match the “style” of the training images at different levels, which effectively controls image attributes at different scales. Therefore, in addition to the image discriminator that classifies images as real or fake, we propose multi-scale *style discriminators* that perform the same task but at the feature level. In particular, the style discriminator tries to classify the mean  $\mu(\Phi_l)$  and standard deviation  $\sigma(\Phi_l)$ , which describe the image “style” [21]. Empirically, the multi-scale style discriminator helps prevent mode collapse and enables longer training. Given a style discriminator  $D_l(\mathbf{x}) = \tilde{D}_l(\mu(\Phi_l(\mathbf{x})), \sigma(\Phi_l(\mathbf{x})))$  for layer  $l$ , the style loss is defined as:

$$L_{\text{style}}^l(G) = \mathbb{E}_{\mathbf{z}} [-\log D_l(G(\mathbf{z}))]. \quad (3)$$

The total loss can be written as:

$$L_{\text{total}}(G) = L_{\text{GAN}}(G) + \lambda_i \cdot L_{\text{identity}}(G) + \lambda_s \cdot \sum_l L_{\text{style}}^l(G). \quad (4)$$

We use  $\lambda_i = \lambda_s = 1.0$  for all experiments. We use the GAN loss from DC-GAN [44] for  $L_{\text{GAN}}$ .

## 4. Experiment settings

**Data** We train HoloGAN using a variety of datasets: Basel Face [42], CelebA [35], Cats [63], Chairs [7], Cars [58], and LSUN bedroom [59]. We train HoloGAN on resolutions of  $64 \times 64$  pixels for Cats and Chairs, and  $128 \times 128$  pixels for Basel Face, CelebA, Cars and LSUN bedroom. More details on the datasets and network architecture can be found in the supplemental document.

Note that only the Chairs dataset contains multiple views of the same object; all other datasets only contain unique single views. For this dataset, because of the limited number of ShapeNet [7] 3D chair models (6778 shapes), we render images from 60 randomly sampled views for each chair. During training, we ensure that each batch contains completely different types of chairs to prevent the network from using set supervision, i.e., looking at the same chair from different viewpoints in the same batch, to cheat.

**Implementation details** We use adaptive instance normalization [21] for the generator, and a combination of instance normalization [56] and spectral normalization [39] for the discriminator. See our supplemental document for details.

We train HoloGAN from scratch using the Adam solver [31]. To generate images during training, we sample  $\mathbf{z} \sim \mathcal{U}(-1, 1)$ , and also sample random poses from a uniform distribution (more details on pose sampling can be found in the supplemental document). We use  $|\mathbf{z}| = 128$  for all datasets, except for Cars at  $128 \times 128$ , where we use  $|\mathbf{z}| = 200$ . Our code is available at <https://github.com/thunguyenphuoc/HoloGAN>.

## 5. Results

We first show qualitative results of HoloGAN on datasets with increasing complexity (Section 5.1). Secondly, we provide quantitative evidence that shows HoloGAN can generate images with comparable or higher visual fidelity than other 2D-based GAN models (Section 5.2). We also show the effectiveness of using our learnt 3D representation compared to explicit 3D geometries (binary voxel grids) for image generation (Section 5.3). We then show how HoloGAN learns to disentangle shape and appearance (Section 5.4). Finally, we perform an ablation study to demonstrate the effectiveness of our network design and training approach (Section 5.5).

### 5.1. Qualitative evaluation

Figures 1, 4, 6 and 7b show that HoloGAN can smoothly vary the pose along azimuth and elevation while keeping the same identities for multiple different datasets. Note that the LSUN dataset contains a variety of complex layouts of multiple objects. This makes it a very challenging dataset for learning to disentangle pose from object identity.

In the supplemental document, we show results for the Basel Face dataset. We also perform linear interpolation with the noise vectors while keeping the poses the same, and show that HoloGAN can smoothly interpolate the identities between two samples. This demonstrates that HoloGAN correctly learns an explicit deep 3D representation that disentangles pose from identity, despite not seeing any pose labels or 3D shapes during training.

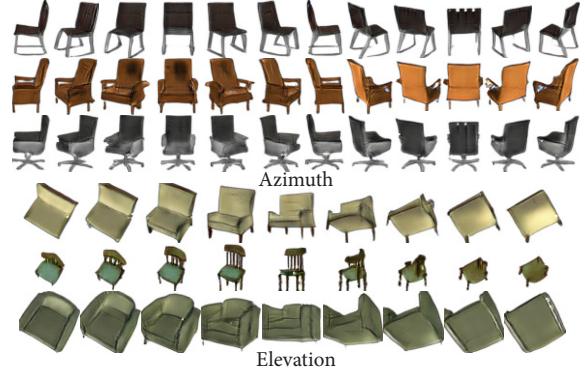


Figure 4. For the Chairs dataset with high intra-class variation, HoloGAN can still disentangle pose ( $360^\circ$  azimuth,  $160^\circ$  elevation) and identity.



Figure 5. We compare HoloGAN to InfoGAN (images adapted from Chen et al. [9]) on CelebA ( $64 \times 64$ ) in the task of separating identity and azimuth. Note that we cannot control what can be learnt by InfoGAN.

**Comparison to InfoGAN [9]** We compare our approach to InfoGAN on the task of learning to disentangle identity from pose on the CelebA dataset [35] at a resolution of  $64 \times 64$  pixels. Due to the lack of publicly available code and hyper-parameters for this dataset<sup>1</sup>, we use the CelebA figure from the published paper. We also tried the official InfoGAN implementation with the Cars dataset, but were unable to train the model successfully as InfoGAN appears to be highly sensitive to the choice of prior distributions and the number of latent variables to recover.

Figure 5 shows that HoloGAN successfully recovers and provides much better control over the azimuth while still maintaining the identity of objects in the generated images. HoloGAN can also recover elevation (Figure 6b, right) despite the limited variation in elevation in the CelebA dataset, while InfoGAN cannot. Most importantly, there is no guarantee that InfoGAN always recovers factors that control object pose, while HoloGAN explicitly controls this via rigid-body transformation.

### 5.2. Quantitative results

To evaluate the visual fidelity of generated images, we use the Kernel Inception Distance (KID) by Bińkowski et al. [4]<sup>2</sup>. KID computes the squared maximum mean discrepancy between feature representations (computed from the Inception model [53]) of the real and generated images. In contrast to FID [19], KID

<sup>1</sup>The official code repository at <https://github.com/openai/InfoGAN> only works with the MNIST dataset.

<sup>2</sup><https://github.com/mrbinkowski/MMD-GAN>

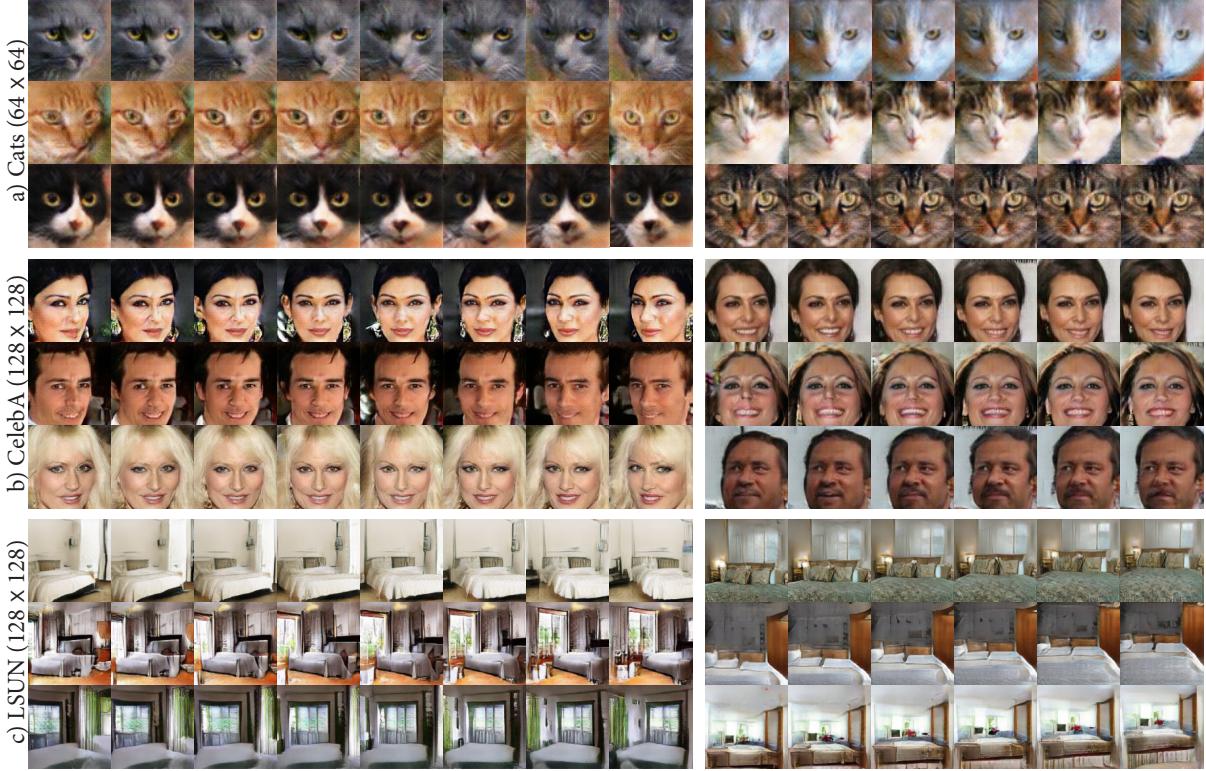


Figure 6. HoloGAN supports changes in both azimuth (range:  $100^\circ$ ) and elevation (range:  $35^\circ$ ). However, the available range depends on the dataset. For CelebA, for example, few photos in the dataset were taken from above or below.

has an unbiased estimator. The lower the KID score, the better the visual quality of generated images. We compare HoloGAN with other recent GAN models: DCGAN [44], LSGAN [38], and WGAN-GP [17], on 3 datasets in Table 1. Note that KID does not take into account feature disentanglement, which is one of the main contributions of HoloGAN.

We use a publicly available implementation<sup>3</sup> and use the same hyper-parameters (that were tuned for CelebA) provided with this implementation for all three datasets. Similarly, for HoloGAN, we use the same network architecture and hyper-parameters<sup>4</sup> for all three datasets. We sample 20,000 images from each model to calculate the KID scores shown below.

Table 1 shows that HoloGAN can generate images with competitive (for CelebA) or even better KID scores on more challenging datasets: Chairs, which has high intra-class variability, and Cars, which has complex backgrounds and lighting conditions. This also shows that the HoloGAN architecture is more robust and can consistently produce images with high visual fidelity across different datasets with the same set of hyper-parameters (except for azimuth ranges). We include visual samples for these models in the supplemental document. More importantly, HoloGAN learns a disentangled representation that allows manipulation of the generated images. This is a great advantage compared to methods such as  $\beta$ -VAE [20], which has to compromise between the image

<sup>3</sup><https://github.com/LynnHo/DCGAN-LSGAN-WGAN-WGAN-GP-Tensorflow>

<sup>4</sup>Except for ranges for sampling the azimuth:  $100^\circ$  for CelebA since face images are only taken from frontal views, and  $360^\circ$  for Chairs and Cars.

Method	CelebA 64x64	Chairs 64x64	Cars 64x64
DCGAN [44]	$1.81 \pm 0.09$	$6.36 \pm 0.16$	$4.78 \pm 0.11$
LSGAN [38]	$1.77 \pm 0.06$	$6.72 \pm 0.19$	$4.99 \pm 0.13$
WGAN-GP [17]	<b><math>1.63 \pm 0.09</math></b>	$9.43 \pm 0.24$	$15.57 \pm 0.29$
HoloGAN (ours)	$2.87 \pm 0.09$	<b><math>1.54 \pm 0.07</math></b>	<b><math>2.16 \pm 0.09</math></b>

Table 1. KID [19] between real images and images generated by HoloGAN and other 2D-based GANs (lower is better). We report KID mean  $\times 100 \pm \text{std.} \times 100$ . The table shows that HoloGAN can achieve competitive or higher KID score with other methods, while providing explicit control of objects in the generated images (not measured by KID).

quality and the level of disentanglement of the learnt features.

### 5.3. Deep 3D representation vs. 3D geometry

Here we compare our method to the state-of-the-art visual object networks (VON) [64] on the task of generating car images. We use the trained model and code provided by the authors. Although VON also takes a disentangling approach to generating images, it relies on 3D shapes and silhouette masks during training, while HoloGAN does not. Figure 7b shows that our approach can generate images of cars with complex backgrounds, realistic shadows and competitive visual fidelity. Note that to generate images at  $128 \times 128$ , VON uses the full binary voxel geometry at  $128 \times 128 \times 128 \times 1$  resolution, while HoloGAN uses a deep voxel representation of up to  $16 \times 16 \times 16 \times 64$  resolution, which is more spatially compact and expressive since HoloGAN also generates complex backgrounds and shadows. As highlighted in Figure



Figure 7. **a)** Car images generated by VON for an azimuth range of 360°. Despite using 3D shapes and silhouette masks during training, VON can only generate images with a simple white background, and struggles for certain frontal views (highlighted in red) and rear views (highlighted in blue). **b)** HoloGAN generates car images with complex backgrounds and varying azimuth (range: 360°) and elevation (range: 35°) from unlabelled images.

**7a,** VON also tends to change the identity of the car such as changing colours or shapes at certain views (highlighted), while HoloGAN maintains the identity of the car in all views. Moreover, HoloGAN can generate car images in full 360° views (Figure 7), while VON struggles to generate images from the back views.

Traditional voxel grids can be very memory intensive. HoloGAN hints at the great potential of using explicit deep voxel representations for image generation, as opposed to using the full 3D geometry in the traditional rendering pipeline. For example, in Figure 6c, we generate images of the entire bedroom scene using a 3D representation of only  $16 \times 16 \times 16 \times 64$  resolution.

#### 5.4. Disentangling shape and appearance

Here we show that in addition to pose, HoloGAN also learns to further divide identity into shape and appearance. We sample two latent codes,  $\mathbf{z}_1$  and  $\mathbf{z}_2$ , and feed them through HoloGAN. While  $\mathbf{z}_1$  controls the 3D features (before perspective morphing and projection),  $\mathbf{z}_2$  controls the 2D features (after projection). Figure 8 shows the generated images with the same pose, same  $\mathbf{z}_1$ , but with a different  $\mathbf{z}_2$  at each row. As can be seen, while the 3D features control objects' shapes, the 2D features control appearance (texture and lighting). This shows that by using 3D convolutions to learn 3D representations and 2D convolutions to learn shading, HoloGAN learns to separate shape from appearance directly from unlabelled

images, allowing separate manipulation of these factors. In the supplemental document, we provide further results, in which we use different latent codes for 3D features at different resolutions, and show the separation between features that control the overall shapes and more fine-grained details such as gender or makeup.

#### 5.5. Ablation studies

We now conduct a series of studies to demonstrate the effectiveness of our network design and training approach.

**Training without random 3D transformations** Randomly rotating the 3D features during training is crucial for HoloGAN, as it encourages the generator to learn a disentangled representation between pose and identity. In Figure 9, we show results with and without 3D transformation during training. For the model trained without 3D transformation, we generate images of rotated objects by manually rotating the learnt 3D features after the model is trained. As can be seen, this model can still generate images with good visual fidelity, but when the pose is changed, it completely fails to generate meaningful images, while HoloGAN can easily generate images of the same objects in different poses. We believe that the random transformation during training forces the generator to learn features that meaningfully undergo geometric transformations, while still being able to generate images that can fool the discriminator. As a result, our training strategy encourages Holo-

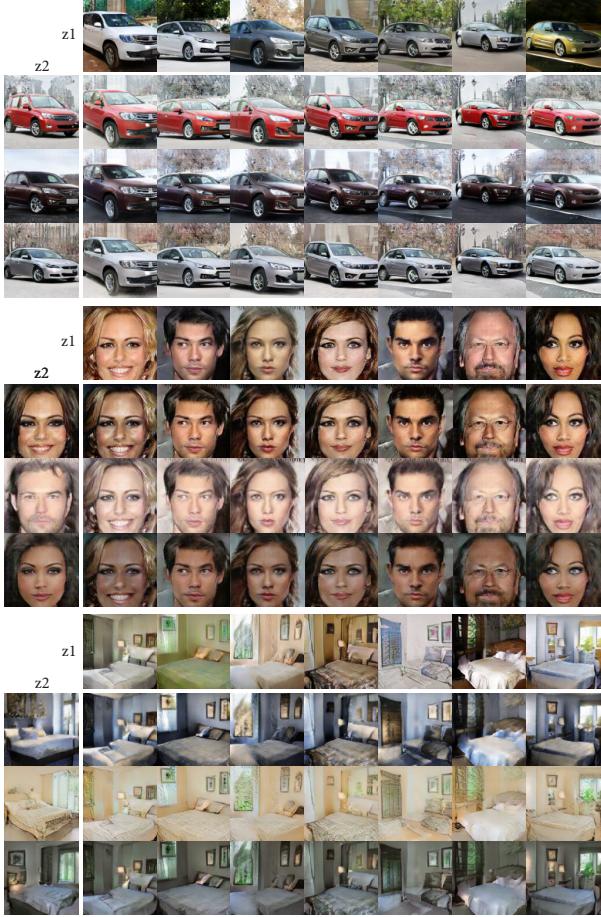


Figure 8. Combinations of different latent vectors  $z_1$  (for 3D features) and  $z_2$  (for 2D features). While  $z_1$  influences objects’ shapes,  $z_2$  determines appearance (texture and lighting). Best viewed in colour.

GAN to learn a disentangled representation of identity and pose.

**Training with traditional  $z$  input** By starting from a learnt constant tensor and using the noise vector  $z$  as a “style” controller at different levels, HoloGAN can better disentangle pose from identity. Here we perform another experiment, in which we feed  $z$  to the first layer of the generator network, like other GAN models. Figure 9 shows that the model trained with this traditional input is confused between pose and identity. As a result, the model also changes the object’s identity when it is being rotated, while HoloGAN can smoothly vary the pose along the azimuth and keep the identity unchanged.

An additional ablation study showing the effectiveness of the identity regulariser is included in the supplemental document.

## 6. Discussion and conclusion

While HoloGAN can successfully learn to separate pose from identity, its performance depends on the variety and distribution of poses included in the training dataset. For example, for the CelebA and Cats dataset, the model cannot recover elevation as well as azimuth (see Figure 6a,b), since most face images are taken at eye level and thus contain limited variation in elevation. Using the



Figure 9. Ablation study showing images with changing azimuths (from left to right). **Top:** Our approach. **Middle:** Our approach without using random 3D transformations during training fails to rotate objects. **Bottom:** Our approach with a traditional input layer mapped from  $z$  instead of a learnt constant tensor fails to disentangle object pose and identity.

wrong pose distribution might also lead to angles being mapped incorrectly. Currently, during training, we sample random poses from a uniform distribution. Future work therefore can explore learning the distribution of poses from the training data in an unsupervised manner to account for uneven pose distributions. Other directions to explore include further disentanglement of objects’ appearances, such as texture and illumination. Finally, it will be interesting to combine HoloGAN with training techniques such as progressive GANs [27] to generate higher-resolution images.

In this work, we presented HoloGAN, a generative image model that learns 3D representation from natural images in an unsupervised manner by adopting strong inductive biases about the 3D world. HoloGAN can be trained end-to-end with only unlabelled 2D images, and learns to disentangle challenging factors such as 3D pose, shape and appearance. This disentanglement provides control over these factors, while being able to generate images with similar or higher visual quality than 2D-based GANs. Our experiments show that HoloGAN successfully learns meaningful 3D representations across multiple datasets with varying complexity. We are therefore convinced that explicit deep 3D representations are a crucial step forward for both the interpretability and controllability of GAN models, compared to existing explicit (meshes, voxels) or implicit [14, 45] 3D representations.

**Acknowledgements** We received support from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 665992, the EPSRC Centre for Doctoral Training in Digital Entertainment (EP/L016540/1), RCUK grant CAMERA (EP/M023281/1), an EPSRC-UKRI Innovation Fellowship (EP/S001050/1), and an NVIDIA Corporation GPU Grant. We thank Lambda Labs for GPU cloud credits and a travel grant.

## References

- [1] Youssef Alami Mejjati, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim. Unsupervised attention-guided image-to-image translation. In *NeurIPS*, pages 3693–3703, 2018. [1](#) [2](#)
- [2] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Towards open-set identity preserving face synthesis. In *CVPR*, 2018. [3](#)
- [3] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8):1670–1687, 2015. [2](#)
- [4] Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *ICLR*, 2018. [5](#)
- [5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019. [1](#)
- [6] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. *arXiv preprint arXiv:1808.07371*, 2018. [1](#)
- [7] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing 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:1512.03012*, 2015. [5](#)
- [8] Ting Chen, Mario Lucic, Neil Houlsby, and Sylvain Gelly. On self modulation for generative adversarial networks. In *ICLR*, 2019. [2](#)
- [9] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, pages 2172–2180, 2016. [2](#) [3](#) [5](#)
- [10] Emily L Denton and Vignesh Birodkar. Unsupervised learning of disentangled representations from video. In *NIPS*, pages 4414–4423, 2017. [3](#)
- [11] Hui Ding, Kumar Sricharan, and Rama Chellappa. ExprGAN: Facial expression editing with controllable expression intensity. In *AAAI*, 2018. [1](#)
- [12] Garoe Dorta, Sara Vicente, Neill D. F. Campbell, and Ivor Simpson. The GAN that warped: Semantic attribute editing with unpaired data. *arXiv:1811.12784*, 2018. [1](#)
- [13] Alexey Dosovitskiy, Jost Springenberg, Maxim Tatarchenko, and Thomas Brox. Learning to generate chairs, tables and cars with convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):692–705, 2017. [1](#)
- [14] S. M. Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S. Morcos, Marta Garnelo, Avraham Ruderman, Andrei A. Rusu, Ivo Danihelka, Karol Gregor, David P. Reichert, Lars Buesing, Theophane Weber, Oriol Vinyals, Dan Rosenbaum, Neil Rabinowitz, Helen King, Chloe Hillier, Matt Botvinick, Daan Wierstra, Koray Kavukcuoglu, and Demis Hassabis. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. [1](#) [2](#) [3](#) [8](#)
- [15] Jiahao Geng, Tianjia Shao, Youyi Zheng, Yanlin Weng, and Kun Zhou. Warp-guided GANs for single-photo facial animation. *ACM Transactions on Graphics*, 37(6):231:1–12, 2018. [2](#)
- [16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014. [2](#)
- [17] Ishaaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of Wasserstein GANs. In *NIPS*, pages 5767–5777, 2017. [6](#)
- [18] Paul Henderson and Vittorio Ferrari. Learning to generate and reconstruct 3D meshes with only 2D supervision. In *BMVC*, 2018. [2](#)
- [19] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *NIPS*, pages 6626–6637, 2017. [5](#) [6](#)
- [20] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner.  $\beta$ -VAE: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017. [2](#) [3](#) [6](#)
- [21] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. [4](#) [5](#)
- [22] Eldar Insafutdinov and Alexey Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. In *NeurIPS*, 2018. [2](#)
- [23] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 5967–5976, 2017. [1](#)
- [24] Insu Jeon, Wonkwang Lee, and Gunhee Kim. IB-GAN: Disentangled representation learning with information bottleneck GAN. <https://openreview.net/forum?id=ryljV2A5KX>, 2019. [3](#)
- [25] Danilo Jimenez Rezende, S. M. Ali Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3D structure from images. In *NIPS*, pages 4996–5004, 2016. [2](#)
- [26] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018. [2](#)
- [27] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018. [1](#) [2](#) [8](#)
- [28] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. [1](#) [2](#) [4](#)
- [29] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D mesh renderer. In *CVPR*, 2018. [2](#)
- [30] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *ACM Transactions on Graphics*, 37(4):163:1–14, 2018. [1](#)
- [31] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. [5](#)
- [32] Jean Kossaifi, Linh Tran, Yannis Panagakis, and Maja Pantic. GAGAN: Geometry-aware generative adversarial networks. In *CVPR*, 2018. [2](#)
- [33] Tejas D Kulkarni, William F. Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *NIPS*, pages 2539–2547, 2015. [1](#) [3](#)
- [34] Tzu-Mao Li, Miika Aittala, Fredo Duran, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Transactions on Graphic*, 37(6):162:1–11, 2018. [2](#)
- [35] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. [5](#)
- [36] Matthew M. Loper and Michael J. Black. OpenDR: An approximate differentiable renderer. In *ECCV*, pages 154–169, 2014. [2](#)
- [37] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshops*, 2013. [4](#)
- [38] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, 2017. [6](#)

- [39] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018. 5
- [40] Thu Nguyen-Phuoc, Chuan Li, Stephen Balaban, and Yongliang Yang. RenderNet: A deep convolutional network for differentiable rendering from 3D shapes. In *NeurIPS*, pages 7902–7912, 2018. 1, 2, 3, 4
- [41] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C. Berg. Transformation-grounded image generation network for novel 3D view synthesis. In *CVPR*, pages 702–711, 2017. 1
- [42] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3D face model for pose and illumination invariant face recognition. In *International Conference on Advanced Video and Signal Based Surveillance*, pages 296–301, 2009. 5
- [43] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, pages 5099–5108, 2017. 2
- [44] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 4, 6
- [45] Sai Rajeswar, Fahim Mannan, Florian Golemo, David Vazquez, Derek Nowrouzezahrai, and Aaron Courville. Pix2Scene: Learning implicit 3D representations from images. <https://openreview.net/forum?id=BJeem3C9F7>, 2019. 2, 3, 8
- [46] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3D faces using convolutional mesh autoencoders. In *ECCV*, pages 725–741, 2018. 2
- [47] Scott Reed, Kihyuk Sohn, Yuting Zhang, and Honglak Lee. Learning to disentangle factors of variation with manifold interaction. In *ICML*, 2014. 3
- [48] Helge Rhodin, Mathieu Salzmann, and Pascal Fua. Unsupervised geometry-aware representation for 3D human pose estimation. In *ECCV*, 2018. 2
- [49] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. OctNet: Learning deep 3D representations at high resolutions. In *CVPR*, 2017. 2
- [50] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *CVPR*, 2017. 1
- [51] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. DeepVoxels: Learning persistent 3D feature embeddings. In *CVPR*, 2019. 2
- [52] Shao-Hua Sun, Minyoung Huh, Yuan-Hong Liao, Ning Zhang, and Joseph J. Lim. Multi-view to novel view: Synthesizing novel views with self-learned confidence. In *ECCV*, pages 162–178, 2018. 1
- [53] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 5
- [54] Maxim Tatarchenko, Alexei Dosovitskiy, and Thomas Brox. Multi-view 3D models from single images with a convolutional network. In *ECCV*, 2016. 1
- [55] Luan Tran, Xi Yin, and Xiaoming Liu. Representation learning by rotating your faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 3, 4
- [56] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *CVPR*, 2017. 5
- [57] 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*, pages 1696–1704, 2016. 1
- [58] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *CVPR*, pages 3973–3981, 2015. 5
- [59] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv:1506.03365*, 2015. 5
- [60] Gang Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Generative adversarial network with spatial attention for face attribute editing. In *ECCV*, 2018. 1
- [61] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017. 2
- [62] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, pages 7354–7363, 2019. 1, 2
- [63] Weiwei Zhang, Jian Sun, and Xiaoou Tang. Cat head detection – how to effectively exploit shape and texture features. In *ECCV*, 2008. 5
- [64] Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Josh Tenenbaum, and Bill Freeman. Visual object networks: Image generation with disentangled 3D representations. In *NeurIPS*, pages 118–129, 2018. 1, 2, 4, 6