# Towards Unsupervised Learning of Generative Models for 3D Controllable Image Synthesis

Yiyi Liao[1,2,*] Katja Schwarz[1,2,*] Lars Mescheder[1,2,3,†] Andreas Geiger[1,2]
[1]Max Planck Institute for Intelligent Systems, Tübingen [2]University of Tübingen [3]Amazon, Tübingen
{firstname.lastname}@tue.mpg.de

## Abstract

*In recent years, Generative Adversarial Networks have achieved impressive results in photorealistic image synthesis. This progress nurtures hopes that one day the classical rendering pipeline can be replaced by efficient models that are learned directly from images. However, current image synthesis models operate in the 2D domain where disentangling 3D properties such as camera viewpoint or object pose is challenging. Furthermore, they lack an interpretable and controllable representation. Our key hypothesis is that the image generation process should be modeled in 3D space as the physical world surrounding us is intrinsically three-dimensional. We define the new task of 3D controllable image synthesis and propose an approach for solving it by reasoning both in 3D space and in the 2D image domain. We demonstrate that our model is able to disentangle latent 3D factors of simple multi-object scenes in an unsupervised fashion from raw images. Compared to pure 2D baselines, it allows for synthesizing scenes that are consistent wrt. changes in viewpoint or object pose. We further evaluate various 3D representations in terms of their usefulness for this challenging task.*

## 1. Introduction

Realistic image synthesis is a fundamental task in computer vision and graphics with many applications including gaming, simulation, virtual reality and data augmentation. In all these applications, it is essential that the image synthesis algorithm allows for controlling the 3D content and produces coherent images when changing the camera position and object locations. Imagine exploring a virtual reality room. When walking around and manipulating objects in the room, it is essential to observe coherent images, i.e. manipulating the pose of one object should not alter properties (such as the color) of any other object in the scene.

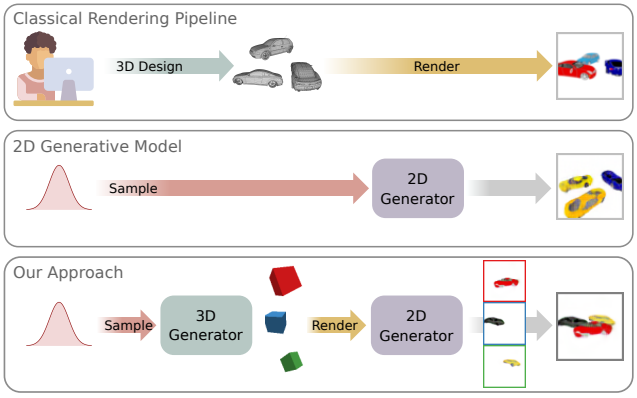



Figure 1: **Motivation.** While classical rendering algorithms require a holistic scene representation, image-based generative neural networks are able to synthesize images from a latent code and can hence be trained directly from images. We propose to consider the generative process in both 3D and 2D space, yielding a 3D controllable image synthesis model which learns both the 3D content creation process as well as the rendering process jointly from raw images.

Currently, image synthesis in these applications is realized using rendering engines (e.g., OpenGL) as illustrated in Fig. 1 (top). This approach provides full control over the 3D content since the input to the rendering algorithm is a holistic description of the 3D scene. However, creating photorealistic content and 3D assets for a movie or video game is an extremely expensive and time-consuming process and requires the concerted effort of many 3D artists. In this paper, we therefore ask the following question:

*Is it possible to learn the simulation pipeline including 3D content creation from raw 2D image observations?*

Recently, Generative Adversarial Networks (GANs) have achieved impressive results for photorealistic image synthesis [21, 22, 29] and hence emerged as a promising alternative to classical rendering algorithms, see Fig. 1 (middle). However, a major drawback of image-based GANs is that the learned latent representation is typically "3D entan-

* Joint first authors with equal contribution.
† This work was done prior to joining Amazon.

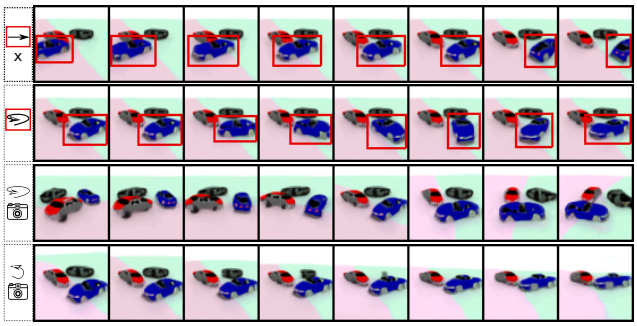


Figure 2: **3D Controllable Image Synthesis.** We propose a model for 3D controllable image synthesis which allows for manipulating the viewpoint and 3D pose of individual objects. Here, we illustrate consistent translation in 3D space, object rotation and viewpoint changes.

gled". That is, the latent dimensions do not automatically expose physically meaningful 3D properties such as camera viewpoint, object pose or object entities. As a result, 2D GANs fall short in terms of controllability compared to classical rendering algorithms. This limits their utility for many applications including virtual reality, data augmentation and simulation.

**Contribution:** In this work, we propose the new task of *3D Controllable Image Synthesis*. We define this task as an unsupervised learning problem, where a 3D controllable generative image synthesis model that allows for manipulating 3D scene properties is learned without 3D supervision. The 3D controllable properties include the 3D pose, shape and appearance of *multiple* objects as well as the viewpoint of the camera. Learning such a model from 2D supervision alone is very challenging as the model must reason about the modular composition of the scene as well as physical 3D properties of the world such as light transport.

Towards a solution to this problem, we propose a novel formulation that combines the advantages of deep generative models with those of traditional rendering pipelines. Our approach enables 3D controllability when synthesizing new content while learning meaningful representations from images alone. Our key idea is to learn the image generation process jointly in 3D and 2D space by combining a 3D generator with a differentiable renderer and a 2D image synthesis model as illustrated in Fig. 1 (bottom). This allows our model to learn abstract 3D representations which conform to the physical image formation process, thereby retaining interpretability and controllability. We demonstrate that our model is able to disentangle the latent 3D factors of simple scenes composed of multiple objects and allows for synthesizing images that are consistent wrt. changes in viewpoint or object pose as illustrated in Fig. 2. Our code and data are provided at https://github.com/autonomousvision/controllable_image_synthesis.

## 2. Related Work

**Image Decomposition:** Several works have considered the problem of decomposing a scene, conditioned on an input image [4, 9, 11–13, 24, 39, 43]. One line of work uses RNNs to sequentially decompose the scene [4, 9, 24, 39], whereas other approaches iteratively refine the image segmentation [11–13, 43]. While early works consider very simple scenes [9, 24] or binarized images [12, 13, 43], recent works [4, 11] also handle scenes with occlusions.

In contrast to the problem of learning controllable generative models, the aforementioned approaches are purely discriminative and are not able to synthesize novel scenes. Furthermore, they operate purely in the 2D image domain and thus lack an understanding of the physical 3D world.

**Image Synthesis:** Unconditional Generative Adversarial Networks (GANs) [10, 30, 34] have greatly advanced photo-realistic image synthesis by learning deep networks that are able to sample from the natural image manifold. More recent works condition these models on additional input information (e.g., semantic segmentations) to guide the image generation process [2, 6, 19, 20, 26, 36, 42, 46, 50].

Another line of work learns interpretable, disentangled features directly from the input data for manipulating the generated images [7, 22, 35, 49]. More recent work aims at controlling the image generation process at the object level [8, 40, 48]. The key insight is that this splits the complex task of image generation into easier subproblems which benefits the quality of the generated images.

However, all of these methods operate based on a 2D understanding of the scene. While some of these works are able to disentangle 3D pose in the latent representation [7], full 3D control in terms of object translations and rotations as well as novel viewpoints remains a challenging task. In this work, we investigate the problem of learning an interpretable intermediate representation at the object-level that is able to provide *full 3D control* over all objects in the scene. We also demonstrate that reasoning in 3D improves consistency wrt. pose and viewpoint changes.

**3D-Aware Image Synthesis:** Several works focus on 3D controllable image synthesis with 3D supervision [45, 52] or using 3D information as input [1]. In this work, we instead focus on learning from 2D images alone. The problem of learning discriminative 3D models from 2D images is often addressed using differentiable rendering [15, 23, 25, 27]. Our work is related to these works in that we also exploit a differentiable renderer. However, unlike the aforementioned methods, we combine differentiable rendering with generative models in 3D and 2D space. We learn an abstract 3D representation that can be transformed into photo-realistic images via a neural network. Previous works in this direction can be categorized as either implicit or explicit,

depending on whether the learned features have a physical meaning.

Implicit methods [37, 47] apply rotation to a 3D latent feature vector to generate transformed images through a multilayer perceptron. This feature vector is a global representation as it influences the entire image.

In contrast, explicit methods [15, 32, 38] exploit feature volumes that can be differentiably projected into 2D image space. DeepVoxels [38] proposes a novel view synthesis approach which unprojects multiple images into a 3D volume and subsequently generates new images by projecting the 3D feature volume to novel viewpoints. Albeit generating high quality images, their model is not generative and requires hundreds of images from the same object for training. In contrast, HoloGAN [32] and PLATONICGAN [15] are generative models that are able to synthesize 3D consistent images of single objects by generating volumetric 3D representations based on a differentiable mapping.

However, all these approaches are only able to learn models for single objects or static scenes. Therefore, their controllability is limited to object-centric rotations or camera viewpoint changes. In contrast, we propose a model for learning disentangled 3D representations which allows for manipulating multiple objects in the scene individually.

# 3. Method

Our goal is to learn a controllable generative model for image synthesis from raw image data. We advocate that image synthesis should benefit from an interpretable 3D representation as this allows to explicitly take into account the image formation process (i.e., perspective projection, occlusion, light transport), rather than trying to directly learn the distribution of 2D images. We thus model the image synthesis process jointly in 3D and 2D space by first generating an abstract 3D representation that is subsequently projected to and refined in the 2D image domain.

Our model is illustrated in Fig. 3. We first sample a latent code that represents all properties of the generated scene. The latent code is passed to a 3D generative model which generates a set of 3D abstract object representations in the form of 3D primitives as well as a background representation. The 3D representations are projected onto the image plane where a 2D generator transforms them into object appearances and composites them into a coherent image. As supervision at the primitive level is hard to acquire, we propose an approach based on adversarial training which does not require 3D supervision.

We now specify our model components in detail. One contribution of our paper is to analyze and compare various 3D representations regarding their suitability for this task. We thus first define the 3D primitive representations that we consider in this work. In the next section, we describe our model for generating the primitive parameters as well as the rendering step that projects the 3D primitives to 2D feature maps in the image plane. Finally, we describe the 2D generator which synthesizes and composites the image as well as the loss functions that we use for training our model in an end-to-end fashion from unlabeled images.

## 3.1. 3D Representations

We aim for a disentangled 3D object representation that is suitable for end-to-end learning while at the same time allowing full control over the geometric properties of each individual scene element, including the object's scale and pose. We refer to this abstract object representation using the term "primitive" in the following.

Further, we use the term "disentangled representation" to describe a set of distinct, informative factors of variations in the observations [3, 28]. We model these factors with a set of 3D primitives $\mathcal{O} = \{\mathbf{o}_{bg}, \mathbf{o}_1, \dots, \mathbf{o}_N\}$ to disentangle the foreground objects and the scene background. Each *foreground object* is described by a set of attributes $\mathbf{o}_i = (\mathbf{s}_i, \mathbf{R}_i, \mathbf{t}_i, \boldsymbol{\phi}_i)$ which represent the properties of each object instance. Here, $\mathbf{s}_i \in \mathbb{R}^3$ denotes the 3D object scale, $\mathbf{R}_i \in SO(3)$ and $\mathbf{t}_i \in \mathbb{R}^3$ are the primitive's 3D pose parameters, and $\boldsymbol{\phi}_i$ is a feature vector determining its appearance. To model the *scene background*, we introduce an additional background primitive $\mathbf{o}_{bg}$ defined accordingly.

As it is unclear which 3D representation is best suited for our task, we compare and discuss the following feature representations in our experimental evaluation:

**Point Clouds:** Point clouds are one of the simplest forms to represent 3D information. We represent each primitive with a set of sparse 3D points. More specifically, each $\boldsymbol{\phi}_i = (\boldsymbol{\phi}_i^l, \boldsymbol{\phi}_i^f)$ represents the location $\boldsymbol{\phi}_i^l \in \mathbb{R}^{M\times 3}$ and feature vectors $\boldsymbol{\phi}_i^f \in \mathbb{R}^{M\times K}$ of $M$ points with latent feature dimension $K$. We apply scaling $\mathbf{s}_i$, rotation $\mathbf{R}_i$ and translation $\mathbf{t}_i$ to the point location $\boldsymbol{\phi}_i^l$.

**Cuboids and Spheres:** We further consider cuboids and spheres as representations which are more closely aligned with classical mesh-based models used in computer graphics. In this case, the feature vector $\boldsymbol{\phi}_i$ represents a texture map that is attached to the surface of the cuboid or sphere, respectively. The geometry is determined by scaling, rotating and translating the primitive via $\mathbf{s}_i$, $\mathbf{R}_i$ and $\mathbf{t}_i$.

**Background:** In order to allow for changes in the camera viewpoint, we do not use a vanilla 2D GAN for generating a background image. Instead, we represent the background using a spherical environment map. More specifically, we attach the background feature map $\boldsymbol{\phi}_{bg}$ as a texture map to the inside of a sphere located at the origin, i.e., $\mathbf{R}_{bg} = \mathbf{I}$ and $\mathbf{t}_{bg} = \mathbf{0}$. The scale $\mathbf{s}_{bg}$ of the background sphere is fixed to a value that is large enough to inclose the entire scene including the camera.

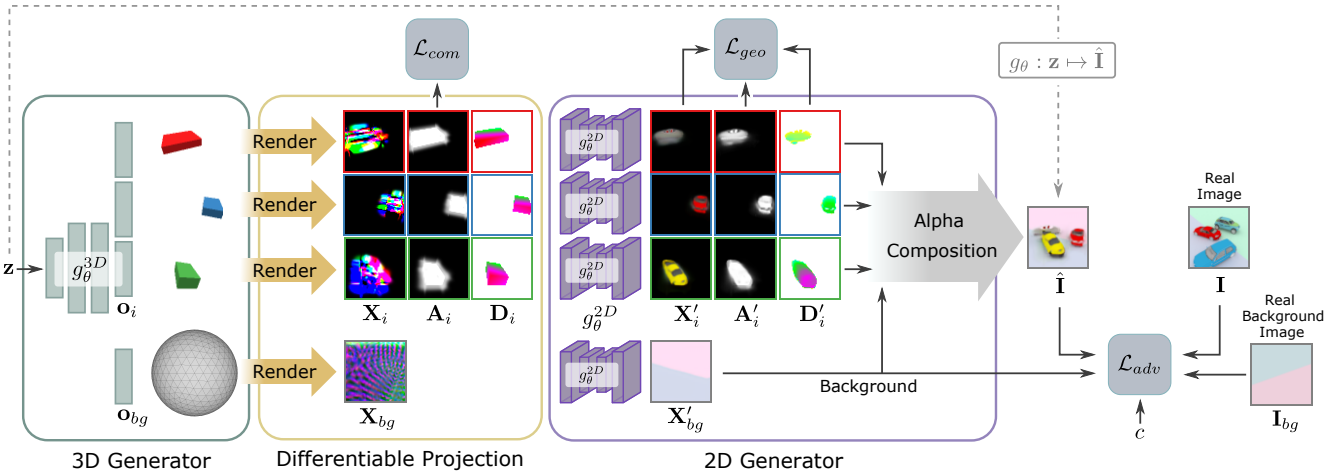



Figure 3: **Method Overview.** Our model comprises three main parts: A *3D Generator* which maps a latent code $\mathbf{z}$ drawn from a Gaussian distribution into a set of abstract 3D primitives $\{\mathbf{o}_i\}$, a *Differentiable Projection* layer which takes each 3D primitive $\mathbf{o}_i$ as input and outputs a feature map $\mathbf{X}_i$, an alpha map $\mathbf{A}_i$ and a depth map $\mathbf{D}_i$, and a *2D Generator* which refines them and produces the final composite image $\hat{\mathbf{I}}$. We refer to the entire model as $g_\theta : \mathbf{z} \mapsto \hat{\mathbf{I}}$ and train it by minimizing a compactness loss $\mathcal{L}_{com}$, a geometric consistency loss $\mathcal{L}_{geo}$ and an adversarial loss $\mathcal{L}_{adv}$ which compares the predicted image $\hat{\mathbf{I}}$ to images $\mathbf{I}$ from the training set. The last primitive $\mathbf{o}_{bg}$ generates the image background $\mathbf{X}'_{bg}$. The flag $c$ determines if the adversarial loss compares full composite images ($c = 1$) or only the background ($c = 0$) to the training set.

### 3.2. 3D Generator

The first part of our neural network is an implicit generative 3D model which generates the set of latent 3D primitives $\{\mathbf{o}_{bg}, \mathbf{o}_1, \ldots, \mathbf{o}_N\}$ from a noise vector $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We use a fully connected model architecture with a fixed number of $N + 1$ heads to generate the attributes for the $N$ foreground primitives and the background. Note, that early layers share weights such that all primitives are generated jointly and share information. More formally we have:

$$g_\theta^{3D} : \mathbf{z} \mapsto \{\mathbf{o}_{bg}, \mathbf{o}_1, \ldots, \mathbf{o}_N\} \tag{1}$$

where $\theta$ denote the parameters of the 3D generation layer. Fig. 3 (left) illustrates our 3D generator. Details about the network architecture can be found in the supplementary.

### 3.3. Differentiable Projection

The differentiable projection layer takes each of the generated primitives $\mathbf{o}_i \in \mathcal{O}$ as input and converts each of them separately into a feature map $\mathbf{X}_i \in \mathbb{R}^{W \times H \times F}$ where $W \times H$ are the dimensions of the output image and $F$ denotes the number of feature channels. In addition, it generates a coarse alpha map $\mathbf{A}_i \in \mathbb{R}^{W \times H}$ and an initial depth map $\mathbf{D}_i \in \mathbb{R}^{W \times H}$ for each primitive which are further refined by the 2D generator that is described in the next section. We implement the differentiable projection layer as follows, assuming (without loss of generality) a fixed calibrated camera with intrinsics $\mathbf{K} \in \mathbb{R}^{3 \times 3}$.

**Cuboids, Spheres and Background:** As cuboids, spheres and the background are represented in terms of a surface mesh, we exploit a differentiable mesh renderer to project them onto the image domain. More specifically, we use the Soft Rasterizer of Liu et al. [27] and adapt it to our purpose. In addition to the projected features $\mathbf{X}_i$, we obtain an alpha map $\mathbf{A}_i$ by smoothing the silhouette of the projected object using a Gaussian kernel. We further output the depth map $\mathbf{D}_i$ of the projected mesh.

**Point Clouds:** For point clouds we follow prior works [18] and represent them in a smooth fashion using isotropic Gaussians. More specifically, we project all features onto an initial empty feature map $\mathbf{X}_i$ and smooth the result using a Gaussian blur kernel. This allows us to learn both the locations $\phi_i^l$ as well as the features $\phi_i^f$ while also back-propagating gradients to the pose parameters $\{\mathbf{s}_i, \mathbf{R}_i, \mathbf{t}_i\}$ of the $i$'th primitive. Since point clouds are sparse, we determine the initial alpha map $\mathbf{A}_i$ and depth map $\mathbf{D}_i$ by additionally projecting a cuboid with the same scale, rotation and translation onto the image plane as described above.

### 3.4. 2D Generator

Learning a 3D primitive for each object avoids explicitly reconstructing their exact 3D models, while the projected features are abstract. We learn to transform this abstract representation into a photorealistic image using a 2D generative model. More specifically, for each primitive, we use a fully convolutional network which takes the features, the initial alpha map and the initial depth map as input and refines it, yielding a color rendering, a refined alpha map and

a refined depth map of the corresponding object

$$g_\theta^{2D} : \mathbf{X}_i, \mathbf{A}_i, \mathbf{D}_i \mapsto \mathbf{X}'_i, \mathbf{A}'_i, \mathbf{D}'_i \tag{2}$$

where $\theta$ denote the parameters of the network that are shared across the objects and the background[1]. We use a standard encoder-decoder structure based on ResNet [14] as our 2D generator. See supplementary material for details.

It is important to note that we perform amodal object prediction, i.e., all primitives are predicted separately without taking into account occlusion. Thus, our 2D generator can learn to generate the entire object for each primitive, even if it is partially occluded in some of the images.

The final step in our 2D generation layer is to combine the individual predictions into a composite image $\hat{\mathbf{I}}$. Towards this goal, we fuse all predictions using alpha composition in ascending order of depth at each pixel. Implementation details are provided in the supplementary.

### 3.5. Loss Functions

We train the entire model end-to-end using adversarial training. Importantly, we do not rely on supervision in the form of labeled 3D primitives, instance segmentations or pose annotations. The only input to our method are images of scenes with various number of objects in different poses, from varying viewpoints and with varying backgrounds.

Learning such a model without supervision is a challenging task with many ambiguities. The model could for instance learn to explain two objects with a single primitive or even to generate all foreground objects with the background model, setting all alpha maps $\mathbf{A}'_i$ to zero. We therefore introduce multiple loss functions that encourage a disentangled and interpretable 3D representation while at the same time synthesizing images from the training data distribution. Our loss $\mathcal{L}$ is composed of three terms:

$$\mathcal{L}(\theta, \psi) = \sum_{c \in \{0,1\}} \mathcal{L}_{adv}(\theta, \psi, c) + \mathcal{L}_{com}(\theta) + \mathcal{L}_{geo}(\theta) \tag{3}$$

**Adversarial Loss:** We use a standard adversarial loss [10] to encourage that the images generated by our model follow the data distribution. Let $\mathbf{I}$ denote an image sampled from the data distribution $p_{\mathcal{D}}(\mathbf{I})$ and let $g_\theta : \mathbf{z} \mapsto \hat{\mathbf{I}}$ denote the entire generative model. Let further $d_\psi(\mathbf{I})$ denote the discriminator. Our adversarial loss is formulated as follows

$$\begin{aligned}\mathcal{L}_{adv}(\theta, \psi, c) = \mathbb{E}_{p(\mathbf{z})}[f(d_\psi(g_\theta(\mathbf{z}, c), c))] \\ + \mathbb{E}_{p_{\mathcal{D}}(\mathbf{I}|c)}[f(-d_\psi(\mathbf{I}, c))]\end{aligned} \tag{4}$$

with $f(t) = -\log(1 + \exp(-t))$. Note that we condition both the generator $g_\theta(\mathbf{z}, c)$ as well as the discriminator $d_\psi(\mathbf{I}, c)$ on an additional variable $c \in \{0, 1\}$ which determines if the generated/observed image is a full composite image ($c = 1$) or a background image ($c = 0$). This condition is helpful to disentangle foreground objects from the background as evidenced by our experiments. For training our model, we thus collect two datasets: one that includes foreground objects and one with empty background scenes.

**Compactness Loss:** To bias solutions towards compact representations and to encourage the 3D primitives to tightly encase the objects, we minimize the projected shape of each object. We formulate this constraint as a penalty on the $l_1$-norm of the alpha maps $\mathbf{A}_i$ of each object:

$$\mathcal{L}_{com}(\theta) = \mathbb{E}_{p(\mathbf{z})}\left[\sum_{i=1}^{N} \max\left(\tau, \frac{\|\mathbf{A}_i\|_1}{H \times W}\right)\right] \tag{5}$$

Here, $\tau = 0.1$ is a truncation threshold which avoids shrinkage below a fixed minimum size and $\mathbf{A}_i$ depends[2] on the model parameters $\theta$ and the latent code $\mathbf{z}$.

**Geometric Consistency Loss:** To favor solutions that are consistent across camera viewpoints and 3D object poses, we follow [33] and encourage the learned generative model to conform to multi-view geometry constraints. For instance, a change in the pose parameters $(\mathbf{R}_i, \mathbf{t}_i)$ should change the pose of the object but it should not alter neither its color nor its identity. We formulate this constraint as follows:

$$\begin{aligned}\mathcal{L}_{geo}(\theta) = \mathbb{E}_{p(\mathbf{z})}\left[\sum_{i=1}^{N} \|\mathbf{A}'_i \odot (\mathbf{X}'_i - \tilde{\mathbf{X}}'_i)\|_1\right] \\ + \mathbb{E}_{p(\mathbf{z})}\left[\sum_{i=1}^{N} \|\mathbf{A}'_i \odot (\mathbf{D}'_i - \tilde{\mathbf{D}}'_i)\|_1\right]\end{aligned} \tag{6}$$

Here, $(\mathbf{X}'_i, \mathbf{D}'_i)$ are the outputs of the 2D generator for latent code $\mathbf{z}$. $(\tilde{\mathbf{X}}'_i, \tilde{\mathbf{D}}'_i)$ correspond to the outputs when using the same latent code $\mathbf{z}$, but adding random noise to the pose parameters of each primitive and warping the results back to the original viewpoint. The warping function is determined by the predicted depth and the relative transformation between the two views. The operator $\odot$ denotes elementwise multiplication and makes sure that geometric consistency is only enforced inside the foreground region, i.e., where $\mathbf{A}'_i = 1$. Intuitively, the geometric consistency loss encourages objects with the same appearance but which are observed from a different view to agree both in terms of appearance and depth. At the same time it serves as an unsupervised loss for training the depth prediction model. For details on the warping process, we refer the reader to the supplemental material.

[1] We slightly abuse notation and use the same symbol $\theta$ to describe the parameters of the 2D and the 3D generative models

[2] We omit this dependency here for clarity.

Car w/o BG Car with BG Indoor Fruit

Figure 4: **Datasets.** Random samples from each dataset.

### 3.6. Training

We train our model using RMSprop [41] and a learning rate of $10^{-4}$. To stabilize GAN training, we use gradient penalties [29] and spectral normalization [31] for the discriminator. Following [22] we use adaptive instance normalization [17] for the 2D generator. We randomly sample the camera viewpoint during training from an upper-half sphere around the origin.

## 4. Experiments

In this section, we first compare our approach to several baselines on the task of 3D controllable image generation, both on synthetic and real data. Next, we conduct a thorough ablation study to better understand the influence of different representations and architecture components.

**Dataset:** We render synthetic datasets using objects from ShapeNet [5], considering three datasets with varying difficulty. Two datasets contain cars, one with and the other without background. For both datasets, we randomly sample 1 to 3 cars from a total of 10 different car models. Our third dataset is the most challenging of these three. It comprises indoor scenes containing objects of different categories, including chairs, tables and sofas. As background we use empty room images from Structured3D [51], a synthetic dataset with photo-realistic 2D images. For each dataset we render 48k real images $\mathbf{I}$ and 32k unpaired background images $\mathbf{I}_{bg}$ for training, as well as 9.6k images for validation and testing. The image resolution is $64 \times 64$ for all datasets. In addition to the synthetic datasets, we apply our method to a real world dataset containing 800 images of 5 fruits with 5 different backgrounds. In order to train our model, we also collect a set of unpaired background images. Sample images from our datasets are shown in Fig. 4.

**Baselines:** We first compare our method to several baselines on the car dataset and the indoor dataset: **Vanilla GAN** [29]: a state-of-the-art 2D GAN with gradient penalties. **Layout2Im** [50]: a generative 2D model that generates images conditioned on 2D bounding boxes. **2D baseline**: To investigate the advantage of our 3D representation, we implement a 2D version of our method by replacing the 3D generator by a 2D feature generator, i.e. learning a set of 2D feature maps instead of 3D primitives. **Ours w/o** $c$: We train our method without background images $\mathbf{I}_{bg}$ to investigate if the method is able to disentangle foreground and background without extra supervision. All implementation details are provided in the supplementary.

**Metrics:** We measure the quality of the generated images using the Fréchet Inception Distance (FID) [16]. More specifically, we compute the FID score between the distribution of real images $\mathbf{I}$ and the distribution of generated images $\hat{\mathbf{I}}$, excluding the background images $\mathbf{I}_{bg}$. In our ablation study, we also report $\text{FID}_i$ to quantify the disentanglement of object instances. $\text{FID}_i$ measures the distance between rendered single-object images and our images for each primitive before alpha composition. Besides evaluating on the generated samples, we also apply additional random rotation and translation to the 3D primitives and report FID scores on these transformed samples ($\text{FID}_{\mathbf{t}}$, $\text{FID}_{\mathbf{R}}$). Similarly, we apply random translation to Layout2Im and the 2D baseline. We investigate how well our model captures the underlying 3D distribution in the supplementary.

### 4.1. Controllable Image Generation

We now report our results for the challenging task of 3D controllable image generation on synthetic and real data.

**Synthetic Data:** Table 1 compares the FID scores on the Car and Indoor datasets. Comparing the quantitative results, we see that our method can achieve competitive FID score compared to Layout2Im. However, Layout2Im requires 2D bounding boxes as supervision while our method operates in unsupervised manner.

To test our hypothesis if 3D representations are helpful for controllable image synthesis, we measure FID scores $\text{FID}_{\mathbf{t}}$ and $\text{FID}_{\mathbf{R}}$ on transformed samples. Our results show that the FID scores are relatively stable wrt. random rotations and translations, demonstrating that the perturbed images follow the same distribution. Without background supervision (w/o $c$), our method can disentangle foreground from background if the background appearance is simple (e.g., Car dataset). However, in the case of more complex background appearance (e.g., Indoor dataset) the foreground primitives vanish while the background generates the entire image, resulting in a higher FID score. In contrast, with unpaired background images as supervision, our method is able to disentangle foreground from background even in the presence of complex backgrounds.

In Fig. 5 and Fig. 6, we show qualitative results when transforming the objects. By translating the 2D bounding boxes, Layout2Im achieves 2D controllability that can even handle occlusion. However, the images lack consistency: moving an object changes its identity, indicating a failure of the model to disentangle the latent factors. Moreover, Layout2Im fuses the objects with a recurrent module. Therefore, manipulating one object also affects other objects in the scene as well as the background. Note for example how

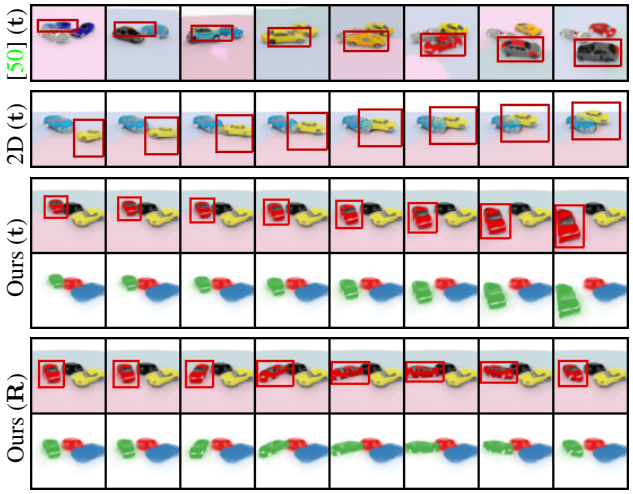



Figure 5: **Car Dataset.** We translate one object for all methods. Additionally, we rotate one object with our method which cannot be achieved with the baselines.

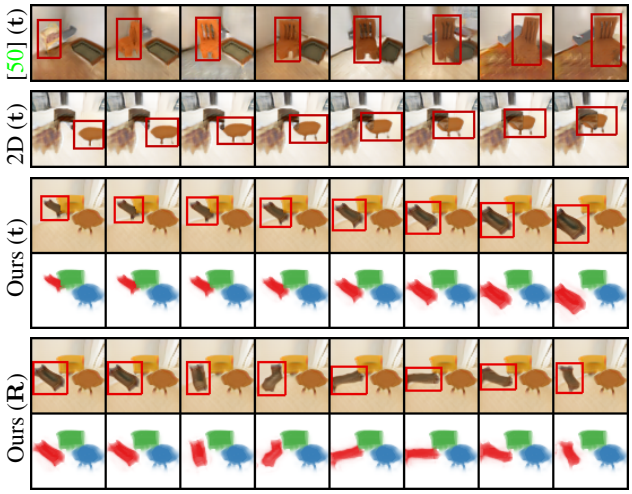
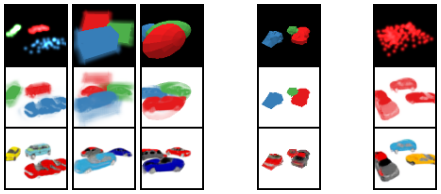



Figure 6: **Indoor Dataset.** We translate one object for all methods. Additionally, we rotate one object with our method which cannot be achieved with the baselines.

| | Car | | | Indoor | | |
|---|---|---|---|---|---|---|
| | FID | $\text{FID}_\mathbf{t}$ | $\text{FID}_\mathbf{R}$ | FID | $\text{FID}_\mathbf{t}$ | $\text{FID}_\mathbf{R}$ |
| Vanilla GAN [29] | **43** | – | – | 89 | – | – |
| Layout2Im [50] | **43** | 56 | – | **84** | 93 | – |
| 2D Baseline | 80 | 79 | – | 107 | 102 | – |
| Ours (w/o $c$) | 65 | 71 | 75 | 120 | 120 | 120 |
| Ours | 44 | **54** | **66** | 88 | **90** | **100** |

Table 1: **FID on Car dataset and Indoor dataset.**


Figure 7: **Fruit Dataset.** We translate one fruit in each row.



(a) (b) (c) (d) (e)

Figure 8: **Ablation Study.** (a) point cloud, (b) cuboid, (c) sphere, (d) deformable sphere w/o $g_\theta^{2D}$, (e) single primitive. The first two rows show the projected primitives and the composite $\mathbf{A}'_i$ (colors denote instances). Note that we visualize all primitives in one image while during training they are rendered and fed into the 2D generator individually.

the black car in the leftmost image in Fig. 5 becomes yellow during translation or how the background in both Fig. 5 and Fig. 6 varies. Our 2D baseline is able to better disentangle objects, while it struggles to learn the correct occlusion relationship. In contrast, our model is able to correctly disentangle the latent factors and produces images that are consistent wrt. object rotation and translation.

**Real Data:** Qualitative results of our method on the real-world dataset are shown in Fig. 7. We see that our method is able to synthesize plausible images even from real data. Making use of our disentangled 3D representation, we are able to change the arrangement of fruits while our model properly handles occlusions and shadows.

### 4.2. Ablation Study

We now compare different representations and analyze multiple variants of our framework in an ablation study. As we focus on the foreground objects in this study, we use the Car dataset without background for all experiments.

**Point cloud, sphere or cuboid?** We first compare the performance of different 3D representations for the task of 3D controllable image synthesis. Table 2 compares different representations in terms of their 3D controllability ($\text{FID}_\mathbf{t}$ and $\text{FID}_\mathbf{R}$) and disentanglement capability ($\text{FID}_i$). We observe that different representations achieve very similar FID scores, suggesting that the precise form of the 3D representation is less relevant than the general concept of a joint 3D-2D representation. This can also be observed from Fig. 8 which illustrates the different representations and the corresponding generated images. The sphere representation performs best overall. We hypothesize that it is superior to the cuboid representation as it does not suffer as much from surface discontinuities in 3D space.

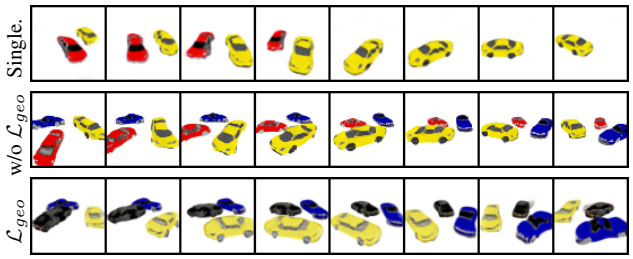



Figure 9: **Evaluation of Rotation Consistency.** Images sampled from 180° of camera rotation. Top row: single primitive representation. The number of objects varies wrt. camera rotation. Middle and bottom rows: multi-primitive model w/o and with the geometric consistency loss.

**Can we learn the 3D model directly?** Inspired by previous works [23, 27] that learn a 3D mesh directly from 2D images in the discriminative model, we also explore whether we can directly learn an accurate 3D model in our generative setting. In other words, we investigate if the 2D generator is helpful or not. To answer this question, we remove the 2D generator from the pipeline and directly add the adversarial loss on the rendered image $\mathbf{X}_i$. This tasks the generative 3D model to predict accurate and textured 3D reconstructions. To allow the generator to modify the shape, we learn to deform a sphere similar to recent mesh-based reconstruction techniques [44]. Fig. 8 (d) shows that it is indeed possible to learn a reasonable shape this way. However, the image fidelity is inferior compared to images generated from our 2D generator as shown in Table 2.

**What is the advantage of disentanglement?** To investigate the advantage of our object-wise generation pipeline, we create a variant of our method using only a single primitive to represent the entire scene. This is illustrated in Fig. 8 (e) and evaluated quantitatively in Table 2. More specifically, we render a point cloud with $N \times M$ points to ensure the model has the same capacity as our multi-primitive model. We do not evaluate $\text{FID}_i$ as the model is not able to generate single object images. Not surprisingly, the single-primitive model is also able to generate images with low FID scores. However, this technique does not allow full control over the individual 3D objects. Moreover, the number of objects changes when camera viewpoint changes as shown in Fig. 9 (top row).

**Is the learned representation consistent in 3D?** We aim for a model that can generate consistent scenes across camera viewpoints and 3D object poses. To test this ability, we rotate the camera and generate new samples as shown in Fig. 9. Our method generates locally coherent images due to its architecture design. In contrast, the single primitive baseline fails to preserve the object identity. In addition, we evaluate if adding the geometric consistency loss can further improve performance in Fig. 9. While we do not observe significant performance gains, geometric consistency cues allow for learning to synthesize depth alongside object appearance. See supplementary for qualitative results.

| | FID | $\text{FID}_\mathbf{t}$ | $\text{FID}_\mathbf{R}$ | $\text{FID}_i$ |
|---|---|---|---|---|
| Vanilla GAN [29] | 50 | – | – | 41 |
| Point cloud | 38 | 43 | 44 | 66 |
| Cuboid | 38 | 45 | 45 | 60 |
| Sphere | 33 | 45 | 45 | 53 |
| Deformable primitive w/o $g_\theta^{2D}$ | 69 | 71 | 74 | 69 |
| Single primitive | 30 | 38 | 44 | – |

Table 2: **Ablation Study.** FID on Car dataset without background wrt. different primitive representations and architecture components.

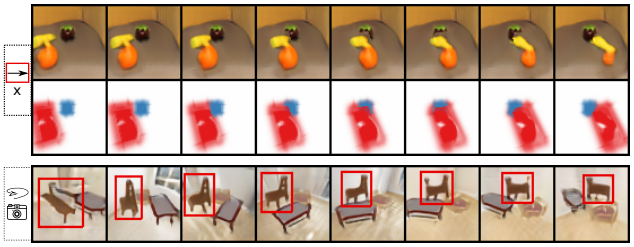


Figure 10: **Failure Cases.** Two fruits are generated from the same primitive (red in the alpha map) in the first example. The object identity changes in the second example.

### 4.3. Failure Cases

We show several failure cases of our method in Fig. 10. Occasionally, a single primitive generates multiple objects (top) or the object's identity changes when the viewpoint change is very large (bottom). We believe that stronger inductive biases are required to tackle these problems.

## 5. Conclusion

We consider this paper as a first step towards unsupervised learning of 3D controllable image synthesis. We demonstrate that modeling both in 3D and 2D space is crucial for accurate and view-consistent results. In addition, we show that our method is able to successfully disentangle scenes with multiple objects while providing controllability in terms of camera viewpoint and object poses. We believe that incorporating stronger inductive biases about object shape or scene geometry will be key for tackling more challenging scenarios. We hope that our results on this new task inspire further research in this exciting area.

## Acknowledgments

We acknowledge support from the BMBF through the Tübingen AI Center (FKZ: 01IS18039B).

## References


[1] H. Alhaija, S. Mustikovela, A. Geiger, and C. Rother. Geometric image synthesis. In *Proc. of the Asian Conf. on Computer Vision (ACCV)*, 2018. 2

[2] O. Ashual and L. Wolf. Specifying object attributes and relations in interactive scene generation. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2

[3] Y. Bengio, A. C. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 35(8):1798–1828, 2013. 3

[4] C. P. Burgess, L. Matthey, N. Watters, R. Kabra, I. Higgins, M. M. Botvinick, and A. Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv.org*, 1901.11390, 2019. 2

[5] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. Shapenet: An information-rich 3d model repository. *arXiv.org*, 1512.03012, 2015. 6

[6] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017. 2

[7] X. Chen, X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2

[8] M. Engelcke, A. R. Kosiorek, O. P. Jones, and I. Posner. GENESIS: generative scene inference and sampling with object-centric latent representations. *arXiv.org*, 1907.13052, 2019. 2

[9] S. M. A. Eslami, N. Heess, T. Weber, Y. Tassa, D. Szepesvari, K. Kavukcuoglu, and G. E. Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2

[10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 2, 5

[11] K. Greff, R. L. Kaufmann, R. Kabra, N. Watters, C. Burgess, D. Zoran, L. Matthey, M. Botvinick, and A. Lerchner. Multi-object representation learning with iterative variational inference. In *Proc. of the International Conf. on Machine learning (ICML)*, 2019. 2

[12] K. Greff, A. Rasmus, M. Berglund, T. H. Hao, H. Valpola, and J. Schmidhuber. Tagger: Deep unsupervised perceptual grouping. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2

[13] K. Greff, S. van Steenkiste, and J. Schmidhuber. Neural expectation maximization. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 2

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[15] P. Henzler, N. J. Mitra, and T. Ritschel. Escaping plato's cave: 3d shape from adversarial rendering. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2, 3

[16] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 6

[17] X. Huang and S. J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017. 6

[18] E. Insafutdinov and A. Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 4

[19] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[20] J. Johnson, A. Gupta, and L. Fei-Fei. Image generation from scene graphs. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[21] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2018. 1

[22] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 6

[23] H. Kato, Y. Ushiku, and T. Harada. Neural 3d mesh renderer. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 8

[24] A. R. Kosiorek, H. Kim, Y. W. Teh, and I. Posner. Sequential attend, infer, repeat: Generative modelling of moving objects. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 2

[25] N. K. L., P. Mandikal, V. Jampani, and R. V. Babu. DIFFER: moving beyond 3d reconstruction with differentiable feature rendering. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019. 2

[26] J. Li, J. Yang, A. Hertzmann, J. Zhang, and T. Xu. Layoutgan: Generating graphic layouts with wireframe discriminators. *arXiv.org*, 1901.06767, 2019. 2

[27] S. Liu, T. Li, W. Chen, and H. Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2, 4, 8

[28] F. Locatello, S. Bauer, M. Lucic, G. Rätsch, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proc. of the International Conf. on Machine learning (ICML)*, 2019. 3

[29] L. Mescheder, A. Geiger, and S. Nowozin. Which training methods for gans do actually converge? In *Proc. of the International Conf. on Machine learning (ICML)*, 2018. 1, 6, 7, 8

[30] L. Mescheder, S. Nowozin, and A. Geiger. The numerics of GANs. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 2

[31] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2018. 6

[32] T. Nguyen-Phuoc, C. Li, L. Theis, C. Richardt, and Y.-L. Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 3

[33] A. Noguchi and T. Harada. RGBD-GAN: unsupervised 3d representation learning from natural image datasets via RGBD image synthesis. *arXiv.org*, 1909.12573, 2019. 5

[34] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv.org*, 1511.06434, 2015. 2

[35] S. Reed, K. Sohn, Y. Zhang, and H. Lee. Learning to disentangle factors of variation with manifold interaction. In *Proc. of the International Conf. on Machine learning (ICML)*, 2014. 2

[36] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee. Learning what and where to draw. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2

[37] H. Rhodin, M. Salzmann, and P. Fua. Unsupervised geometry-aware representation for 3d human pose estimation. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. 3

[38] V. Sitzmann, J. Thies, F. Heide, M. Nießner, G. Wetzstein, and M. Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[39] A. Stanic and J. Schmidhuber. R-SQAIR: relational sequential attend, infer, repeat. In *Advances in Neural Information Processing Systems (NIPS) Workshops*, 2019. 2

[40] T. D. Tamar Rott Shaham and T. Michaeli. Singan: Learning a generative model from a single natural image. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2

[41] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. 2012. 6

[42] M. O. Turkoglu, W. Thong, L. J. Spreeuwers, and B. Kicanaoglu. A layer-based sequential framework for scene generation with gans. In *Proc. of the Conf. on Artificial Intelligence (AAAI)*, 2019. 2

[43] S. van Steenkiste, M. Chang, K. Greff, and J. Schmidhuber. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2018. 2

[44] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. 8

[45] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016. 2

[46] Y. Wang, Y. Yang, Z. Yang, L. Zhao, P. Wang, and W. Xu. Occlusion aware unsupervised learning of optical flow. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[47] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow. Interpretable transformations with encoder-decoder networks. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017. 3

[48] J. Yang, A. Kannan, D. Batra, and D. Parikh. LR-GAN: layered recursive generative adversarial networks for image generation. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2017. 2

[49] B. Zhao, B. Chang, Z. Jie, and L. Sigal. Modular generative adversarial networks. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. 2

[50] B. Zhao, L. Meng, W. Yin, and L. Sigal. Image generation from layout. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 6, 7

[51] J. Zheng, J. Zhang, J. Li, R. Tang, S. Gao, and Z. Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. *arXiv.org*, 1908.00222, 2019. 6

[52] J. Zhu, Z. Zhang, C. Zhang, J. Wu, A. Torralba, J. Tenenbaum, and B. Freeman. Visual object networks: Image generation with disentangled 3d representations. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 2