GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields

Michael Niemeyer[1,2] Andreas Geiger[1,2]
[1]Max Planck Institute for Intelligent Systems, Tübingen [2]University of Tübingen
{firstname.lastname}@tue.mpg.de

Abstract

Deep generative models allow for photorealistic image synthesis at high resolutions. But for many applications, this is not enough: content creation also needs to be controllable. While several recent works investigate how to disentangle underlying factors of variation in the data, most of them operate in 2D and hence ignore that our world is three-dimensional. Further, only few works consider the compositional nature of scenes. Our key hypothesis is that incorporating a compositional 3D scene representation into the generative model leads to more controllable image synthesis. Representing scenes as compositional generative neural feature fields allows us to disentangle one or multiple objects from the background as well as individual objects' shapes and appearances while learning from unstructured and unposed image collections without any additional supervision. Combining this scene representation with a neural rendering pipeline yields a fast and realistic image synthesis model. As evidenced by our experiments, our model is able to disentangle individual objects and allows for translating and rotating them in the scene as well as changing the camera pose.

1. Introduction

The ability to generate and manipulate photorealistic image content is a long-standing goal of computer vision and graphics. Modern computer graphics techniques achieve impressive results and are industry standard in gaming and movie productions. However, they are very hardware expensive and require substantial human labor for 3D content creation and arrangement.

In recent years, the computer vision community has made great strides towards highly-realistic image generation. In particular, Generative Adversarial Networks (GANs) [24] emerged as a powerful class of generative models. They are able to synthesize photorealistic images at resolutions of 1024^2 pixels and beyond [6, 14, 15, 39, 40].

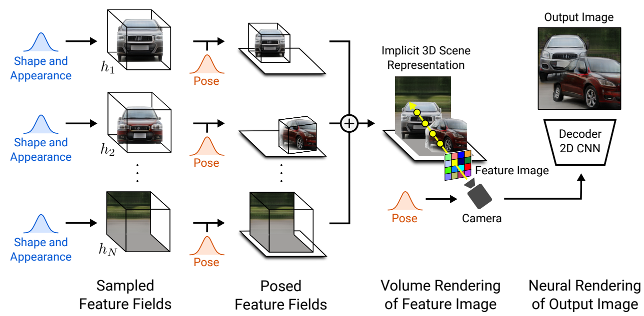
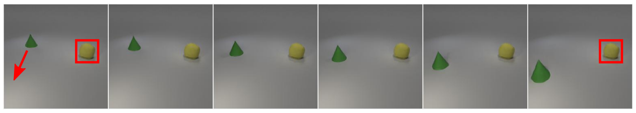


Figure 1: **Overview.** We represent scenes as compositional generative neural feature fields. For a randomly sampled camera, we volume render a feature image of the scene based on individual feature fields. A 2D neural rendering network converts the feature image into an RGB image. While training only on raw image collections, at test time we are able to control the image formation process wrt. camera pose, object poses, as well as the objects' shapes and appearances. Further, our model generalizes beyond the training data, e.g. we can synthesize scenes with more objects than were present in the training images. Note that for clarity we visualize volumes in color instead of features.

Despite these successes, synthesizing realistic 2D images is not the only aspect required in applications of generative models. The generation process should also be controllable in a simple and consistent manner. To this end, many works [9, 25, 39, 43, 44, 48, 54, 71, 74, 97, 98] investigate how disentangled representations can be learned from data without explicit supervision. Definitions of disentanglement vary [5, 53], but commonly refer to being able to control an attribute of interest, e.g. object shape, size, or pose, without changing other attributes. Most approaches, however, do not consider the compositional nature of scenes and operate in the 2D domain, ignoring that our world is three-dimensional. This often leads to entangled representations (Fig. 2) and control mechanisms are not built-in, but need to be discovered in the latent space a posteriori. These properties, however, are crucial for successful applications, e.g. a movie production where complex object trajectories

(a) Translation of Left Object (2D-based Method [71])



(b) Translation of Left Object (Ours)

(c) Circular Translation (Ours) (d) Add Objects (Ours)

Figure 2: **Controllable Image Generation.** While most generative models operate in 2D, we incorporate a compositional 3D scene representation into the generative model. This leads to more consistent image synthesis results, e.g. note how, in contrast to our method, translating one object might change the other when operating in 2D (Fig. 2a and 2b). It further allows us to perform complex operations like circular translations (Fig. 2c) or adding more objects at test time (Fig. 2d). Both methods are trained unsupervised on raw unposed image collections of two-object scenes.

need to be generated in a consistent manner.

Several recent works therefore investigate how to incorporate 3D representations, such as voxels [32,63,64], primitives [46], or radiance fields [77], directly into generative models. While these methods allow for impressive results with built-in control, they are mostly restricted to single-object scenes and results are less consistent for higher resolutions and more complex and realistic imagery (e.g. scenes with objects not in the center or cluttered backgrounds).

Contribution: In this work, we introduce *GIRAFFE*, a novel method for generating scenes in a controllable and photorealistic manner while training from raw unstructured image collections. Our key insight is twofold: First, incorporating a compositional 3D scene representation directly into the generative model leads to more controllable image synthesis. Second, combining this explicit 3D representation with a neural rendering pipeline results in faster inference and more realistic images. To this end, we represent scenes as compositional generative neural feature fields (Fig. 1). We volume render the scene to a feature image of relatively low resolution to save time and computation. A neural renderer processes these feature images and outputs the final renderings. This way, our approach achieves high-quality images and scales to real-world scenes. We find that our method allows for controllable image synthesis of single-object as well as multi-object scenes when trained on raw unstructured image collections. Code and data is available at https://github.com/autonomousvision/giraffe.

2. Related Work

GAN-based Image Synthesis: Generative Adversarial Networks (GANs) [24] have been shown to allow for photorealistic image synthesis at resolutions of 1024^2 pixels and beyond [6, 14, 15, 39, 40]. To gain better control over the synthesis process, many works investigate how factors of variation can be disentangled without explicit supervision. They either modify the training objective [9, 40, 71] or network architecture [39], or investigate latent spaces of well-engineered and pre-trained generative models [1, 16, 23, 27, 34, 78, 96]. All of these works, however, do not explicitly model the compositional nature of scenes. Recent works therefore investigate how the synthesis process can be controlled at the object-level [3,4,7,18,19,26,45,86,90]. While achieving photorealistic results, all aforementioned works model the image formation process in 2D, ignoring the three-dimensional structure of our world. In this work, we advocate to model the formation process directly in 3D for better disentanglement and more controllable synthesis.

Implicit Functions: Using implicit functions to represent 3D geometry has gained popularity in learning-based 3D reconstruction [11, 12, 22, 59, 60, 65, 67, 69, 76] and has been extended to scene-level reconstruction [8, 13, 35, 72, 79]. To overcome the need of 3D supervision, several works [50, 51, 66, 81, 92] propose differentiable rendering techniques. Mildenhall et al. [61] propose Neural Radiance Fields (NeRFs) in which they combine an implicit neural model with volume rendering for novel view synthesis of complex scenes. Due to their expressiveness, we use a generative variant of NeRFs as our object-level representation. In contrast to our method, the discussed works require multi-view images with camera poses as supervision, train a single network per scene, and are not able to generate novel scenes. Instead, we learn a generative model from unstructured image collections which allows for controllable, photorealistic image synthesis of generated scenes.

3D-Aware Image Synthesis: Several works investigate how 3D representations can be incorporated as inductive bias into generative models [21,29–32,46,55,63,64,75,77]. While many approaches use additional supervision [2, 10, 87,88,99], we focus on works which are trained on raw image collections like our approach.

Henzler et al. [32] learn voxel-based representations using differentiable rendering. The results are 3D controllable, but show artifacts due to the limited voxel resolutions caused by their cubic memory growth. Nguyen-Phuoc et al. [63, 64] propose voxelized feature-grid representations which are rendered to 2D via a reshaping operation. While achieving impressive results, training becomes less stable and results less consistent for higher resolutions. Liao et al. [46] use abstract features in combination with primitives and differentiable rendering. While handling multi-object scenes, they require additional supervision in the form of pure background images which are hard to obtain for real-world scenes. Schwarz et al. [77] propose Generative Neural Radiances Fields (GRAF). While achieving controllable image synthesis at high resolutions, this representation is restricted to single-object scenes and results degrade on more complex, real-world imagery. In contrast, we incorporate compositional 3D scene structure into the generative model such that it naturally handles multi-object scenes. Further, by integrating a neural rendering pipeline [20, 41, 42, 49, 62, 80, 81, 83, 84], our model scales to more complex, real-world data.

3. Method

Our goal is a controllable image synthesis pipeline which can be trained from raw image collections without additional supervision. In the following, we discuss the main components of our method. First, we model individual objects as neural feature fields (Sec. 3.1). Next, we exploit the additive property of feature fields to composite scenes from multiple individual objects (Sec. 3.2). For rendering, we explore an efficient combination of volume and neural rendering techniques (Sec. 3.3). Finally, we discuss how we train our model from raw image collections (Sec. 3.4). Fig. 3 contains an overview of our method.

3.1. Objects as Neural Feature Fields

Neural Radiance Fields: A radiance field is a continuous function f which maps a 3D point $\mathbf{x} \in \mathbb{R}^3$ and a viewing direction $\mathbf{d} \in \mathbb{S}^2$ to a volume density $\sigma \in \mathbb{R}^+$ and an RGB color value $\mathbf{c} \in \mathbb{R}^3$. A key observation in [61, 82] is that the low dimensional input $\mathbf{x}$ and $\mathbf{d}$ needs to be mapped to higher-dimensional features to be able to represent complex signals when f is parameterized with a neural network. More specifically, a pre-defined positional encoding is applied element-wise to each component of $\mathbf{x}$ and $\mathbf{d}$:

$$\begin{aligned}&\gamma(t, L) = \\ &(\sin(2^0 t\pi), \cos(2^0 t\pi), \ldots, \sin(2^L t\pi), \cos(2^L t\pi))\end{aligned} \tag{1}$$

where t is a scalar input, e.g. a component of $\mathbf{x}$ or $\mathbf{d}$, and L the number of frequency octaves. In the context of generative models, we observe an additional benefit of this representation: It introduces an inductive bias to learn 3D shape representations in canonical orientations which otherwise would be arbitrary (see Fig. 11).

Following implicit shape representations [12, 59, 69], Mildenhall et al. [61] propose to learn Neural Radiance Fields (NeRFs) by parameterizing f with a multi-layer perceptron (MLP):

$$\begin{aligned}f_\theta : \mathbb{R}^{L_\mathbf{x}} \times \mathbb{R}^{L_\mathbf{d}} &\to \mathbb{R}^+ \times \mathbb{R}^3 \\ (\gamma(\mathbf{x}), \gamma(\mathbf{d})) &\mapsto (\sigma, \mathbf{c})\end{aligned} \tag{2}$$

where θ indicate the network parameters and $L_\mathbf{x}, L_\mathbf{d}$ the output dimensionalities of the positional encodings.

Generative Neural Feature Fields: While [61] fits θ to multiple posed images of a single scene, Schwarz et al. [77] propose a generative model for Neural Radiance Fields (GRAF) that is trained from unposed image collections. To learn a latent space of NeRFs, they condition the MLP on shape and appearance codes $\mathbf{z}_s, \mathbf{z}_a \sim \mathcal{N}(\mathbf{0}, I)$:

$$\begin{aligned}g_\theta : \mathbb{R}^{L_\mathbf{x}} \times \mathbb{R}^{L_\mathbf{d}} \times \mathbb{R}^{M_s} \times \mathbb{R}^{M_a} &\to \mathbb{R}^+ \times \mathbb{R}^3 \\ (\gamma(\mathbf{x}), \gamma(\mathbf{d}), \mathbf{z}_s, \mathbf{z}_a) &\mapsto (\sigma, \mathbf{c})\end{aligned} \tag{3}$$

where M_s, M_a are the dimensionalities of the latent codes.

In this work we explore a more efficient combination of volume and neural rendering. We replace GRAF's formulation for the three-dimensional color output $\mathbf{c}$ with a more generic M_f-dimensional feature $\mathbf{f}$ and represent objects as Generative Neural Feature Fields:

$$\begin{aligned}h_\theta : \mathbb{R}^{L_\mathbf{x}} \times \mathbb{R}^{L_\mathbf{d}} \times \mathbb{R}^{M_s} \times \mathbb{R}^{M_a} &\to \mathbb{R}^+ \times \mathbb{R}^{M_f} \\ (\gamma(\mathbf{x}), \gamma(\mathbf{d}), \mathbf{z}_s, \mathbf{z}_a) &\mapsto (\sigma, \mathbf{f})\end{aligned} \tag{4}$$

Object Representation: A key limitation of NeRF and GRAF is that the entire scene is represented by a single model. As we are interested in disentangling different entities in the scene, we need control over the pose, shape and appearance of *individual* objects (we consider the background as an object as well). We therefore represent each object using a separate feature field in combination with an affine transformation

$$\mathbf{T} = \{\mathbf{s}, \mathbf{t}, \mathbf{R}\} \tag{5}$$

where $\mathbf{s}, \mathbf{t} \in \mathbb{R}^3$ indicate scale and translation parameters, and $\mathbf{R} \in SO(3)$ a rotation matrix. Using this representation, we transform points from object to scene space as follows:

$$k(\mathbf{x}) = \mathbf{R} \cdot \begin{bmatrix} s_1 & & \\ & s_2 & \\ & & s_3 \end{bmatrix} \cdot \mathbf{x} + \mathbf{t} \tag{6}$$

In practice, we volume render in scene space and evaluate the feature field in its canonical object space (see Fig. 1):

$$(\sigma, \mathbf{f}) = h_\theta(\gamma(k^{-1}(\mathbf{x})), \gamma(k^{-1}(\mathbf{d})), \mathbf{z}_s, \mathbf{z}_a) \tag{7}$$

This allows us to arrange multiple objects in a scene. All object feature fields share their weights and $\mathbf{T}$ is sampled from a dataset-dependent distribution (see Sec. 3.4).

3.2. Scene Compositions

As discussed above, we describe scenes as compositions of N entities where the first $N - 1$ are the objects in the scene and the last represents the background. We consider

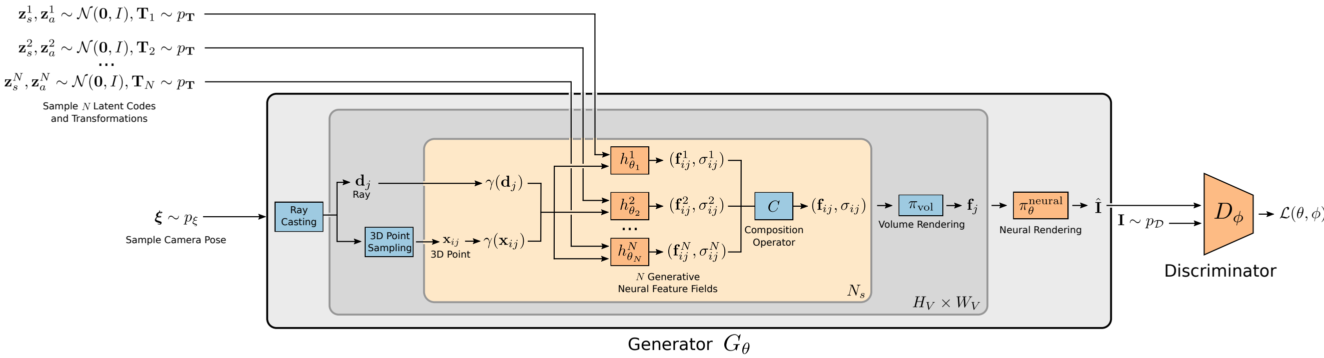


Figure 3: **GIRAFFE.** Our generator G_θ takes a camera pose $\boldsymbol{\xi}$ and N shape and appearance codes $\mathbf{z}_s^i, \mathbf{z}_a^i$ and affine transformations $\mathbf{T}_i$ as input and synthesizes an image of the generated scene which consists of $N-1$ objects and a background. The discriminator D_ϕ takes the generated image $\hat{\mathbf{I}}$ and the real image $\mathbf{I}$ as input and our full model is trained with an adversarial loss. At test time, we can control the camera pose, the shape and appearance codes of the objects, and the objects' poses in the scene. Orange indicates learnable and blue non-learnable operations.

two cases: First, N is fixed across the dataset such that the images always contain $N-1$ objects plus the background. Second, N is varied across the dataset. In practice, we use the same representation for the background as for objects except that we fix the scale and translation parameters $\mathbf{s}_N, \mathbf{t}_N$ to span the entire scene, and to be centered at the scene space origin.

Composition Operator: To define the composition operator C, let's recall that a feature field of a single entity $h_{\theta_i}^i$ predicts a density $\sigma_i \in \mathbb{R}^+$ and a feature vector $\mathbf{f}_i \in \mathbb{R}^{M_f}$ for a given point $\mathbf{x}$ and viewing direction $\mathbf{d}$. When combining non-solid objects, a natural choice [17] for the overall density at $\mathbf{x}$ is to sum up the individual densities and to use the density-weighted mean to combine all features at $(\mathbf{x}, \mathbf{d})$:

$$C(\mathbf{x}, \mathbf{d}) = \left(\sigma, \frac{1}{\sigma}\sum_{i=1}^{N}\sigma_i \mathbf{f}_i\right), \text{ where } \quad \sigma = \sum_{i=1}^{N}\sigma_i \tag{8}$$

While being simple and intuitive, this choice for C has an additional benefit: We ensure gradient flow to all entities with a density greater than 0.

3.3. Scene Rendering

3D Volume Rendering: While previous works [47, 57, 61, 77] volume render an RGB color value, we extend this formulation to rendering an M_f-dimensional feature vector $\mathbf{f}$.

For given camera extrinsics $\boldsymbol{\xi}$, let $\{\mathbf{x}_j\}_{j=1}^{N_s}$ be sample points along the camera ray $\mathbf{d}$ for a given pixel, and $(\sigma_j, \mathbf{f}_j) = C(\mathbf{x}_j, \mathbf{d})$ the corresponding densities and feature vectors of the field. The volume rendering operator π_{vol} [37] maps these evaluations to the pixel's final feature vector $\mathbf{f}$:

$$\pi_{\text{vol}} : (\mathbb{R}^+ \times \mathbb{R}^{M_f})^{N_s} \to \mathbb{R}^{M_f}, \quad \{\sigma_j, \mathbf{f}_j\}_{j=1}^{N_s} \mapsto \mathbf{f} \tag{9}$$

Using numerical integration as in [61], $\mathbf{f}$ is obtained as

$$\mathbf{f} = \sum_{j=1}^{N_s}\tau_j\alpha_j\mathbf{f}_j \quad \tau_j = \prod_{k=1}^{j-1}(1-\alpha_k) \quad \alpha_j = 1 - e^{-\sigma_j\delta_j} \tag{10}$$

where τ_i is the transmittance, α_j the alpha value for $\mathbf{x}_j$, and $\delta_j = ||\mathbf{x}_{j+1} - \mathbf{x}_j||_2$ the distance between neighboring sample points. The entire feature image is obtained by evaluating π_{vol} at every pixel. For efficiency, we render feature images at resolution 16^2 which is lower than the output resolution of 64^2 or 256^2 pixels. We then upsample the low-resolution feature maps to higher-resolution RGB images using 2D neural rendering. As evidenced by our experiments, this has two advantages: increased rendering speed and improved image quality.

2D Neural Rendering: The neural rendering operator

$$\pi_\theta^{\text{neural}} : \mathbb{R}^{H_V \times W_V \times M_f} \to \mathbb{R}^{H \times W \times 3} \tag{11}$$

with weights θ maps the feature image $\mathbf{I}_V \in \mathbb{R}^{H_V \times W_V \times M_f}$ to the final synthesized image $\hat{\mathbf{I}} \in \mathbb{R}^{H \times W \times 3}$. We parameterize $\pi_\theta^{\text{neural}}$ as a 2D convolutional neural network (CNN) with leaky ReLU [56, 89] activation (Fig. 4) and combine nearest neighbor upsampling with 3×3 convolutions to increase the spatial resolution. We choose small kernel sizes and no intermediate layers to only allow for spatially small refinements to avoid entangling global scene properties during image synthesis while at the same time allowing for increased output resolutions. Inspired by [40], we map the feature image to an RGB image at every spatial resolution, and add the previous output to the next via bilinear upsampling. These skip connections ensure a strong gradient flow to the feature fields. We obtain our final image prediction $\hat{\mathbf{I}}$ by applying a sigmoid activation to the last RGB layer. We validate our design choices in an ablation study (Tab. 4).

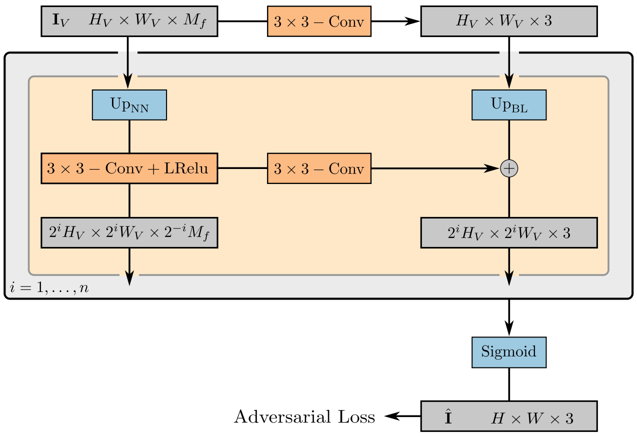


Figure 4: **Neural Rendering Operator.** The feature image $\mathbf{I}_V$ is processed by n blocks of nearest neighbor upsampling and 3×3 convolutions with leaky ReLU activations. At every resolution, we map the feature image to an RGB image with a 3×3 convolution and add it to the previous output via bilinear upsampling. We apply a sigmoid activation to obtain the final image $\hat{\mathbf{I}}$. Gray color indicates outputs, orange learnable, and blue non-learnable operations.

3.4. Training

Generator: We denote the full generative process formally as

$$
\begin{aligned}
&G_\theta(\{\mathbf{z}_s^i, \mathbf{z}_a^i, \mathbf{T}_i\}_{i=1}^N, \boldsymbol{\xi}) = \pi_\theta^{\text{neural}}(\mathbf{I}_V) \\
&\text{where} \quad \mathbf{I}_V = \{\pi_{\text{vol}}(\{C(\mathbf{x}_{jk}, \mathbf{d}_k)\}_{j=1}^{N_s})\}_{k=1}^{H_V \times W_V}
\end{aligned}
\tag{12}
$$

and N is the number of entities in the scene, N_s the number of sample points along each ray, $\mathbf{d}_k$ is the ray for the k-th pixel, and $\mathbf{x}_{jk}$ the j-th sample point for the k-th pixel / ray.

Discriminator: We parameterize the discriminator D_ϕ as a CNN [73] with leaky ReLU activation.

Training: During training, we sample the the number of entities in the scene $N \sim p_N$, the latent codes $\mathbf{z}_s^i, \mathbf{z}_a^i \sim \mathcal{N}(\mathbf{0}, I)$, as well as a camera pose $\boldsymbol{\xi} \sim p_\xi$ and object-level transformations $\mathbf{T}_i \sim p_T$. In practice, we define p_ξ and p_T as uniform distributions over dataset-dependent camera elevation angles and valid object transformations, respectively.[1] The motivation for this choice is that in most real-world scenes, objects are arbitrarily rotated, but not tilted due to gravity. The observer (the camera in our case), in contrast, can freely change its elevation angle wrt. the scene.

We train our model with the non-saturating GAN objective [24] and R_1 gradient penalty [58]

$$
\begin{aligned}
&\mathcal{V}(\theta, \phi) = \\
&\mathbb{E}_{\mathbf{z}_s^i, \mathbf{z}_a^i \sim \mathcal{N},\, \boldsymbol{\xi} \sim p_\xi,\, \mathbf{T}_i \sim p_T} \left[f(D_\phi(G_\theta(\{\mathbf{z}_s^i, \mathbf{z}_a^i, \mathbf{T}_i\}_i, \boldsymbol{\xi}))\right] \\
&+ \mathbb{E}_{\mathbf{I} \sim p_{\mathcal{D}}} \left[f(-D_\phi(\mathbf{I})) - \lambda \|\nabla D_\phi(\mathbf{I})\|^2 \right]
\end{aligned}
\tag{13}
$$

where $f(t) = -\log(1 + \exp(-t))$, $\lambda = 10$, and $p_{\mathcal{D}}$ indicates the data distribution.

3.5. Implementation Details

All object feature fields $\{h_{\theta_i}^i\}_{i=1}^{N-1}$ share their weights and we parametrize them as MLPs with ReLU activations. We use 8 layers with a hidden dimension of 128 and a density and a feature head of dimensionality 1 and $M_f = 128$, respectively. For the background feature field $h_{\theta_N}^N$, we use half the layers and hidden dimension. We use $L_{\mathbf{x}} = 2 \cdot 3 \cdot 10$ and $L_{\mathbf{d}} = 2 \cdot 3 \cdot 4$ for the positional encodings. We sample $M_s = 64$ points along each ray and render the feature image $\mathbf{I}_V$ at 16^2 pixels. We use an exponential moving average [93] with decay 0.999 for the weights of the generator. We use the RMSprop optimizer [85] with a batch size of 32 and learning rates of 1×10^{-4} and 5×10^{-4} for the discriminator and generator, respectively. For experiments at 256^2 pixels, we set $M_f = 256$ and half the generator learning rate to 2.5×10^{-4}.

4. Experiments

Datasets: We report results on commonly-used single-object datasets *Chairs* [68], *Cats* [95], *CelebA* [52], and *CelebA-HQ* [38]. The first consists of synthetic renderings of Photoshape chairs [70], and the others are image collections of cat and human faces, respectively. The data complexity is limited as the background is purely white or only takes up a small part of the image. We further report results on the more challenging single-object, real-world datasets *CompCars* [91], *LSUN Churches* [94], and *FFHQ* [39]. For *CompCars*, we randomly crop the images to achieve more variety of the object's position in the image.[2] For these datasets, disentangling objects is more complex as the object is not always in the center and the background is more cluttered and takes up a larger part of the image. To test our model on multi-object scenes, we use the script from [36] to render scenes with 2, 3, 4, or 5 random primitives (*Clevr-N*). To test our model on scenes with a varying number of objects, we also run our model on the union of them (*Clevr-2345*).

Baselines: We compare against voxel-based PlatonicGAN [32], BlockGAN [64], and HoloGAN [63], and radiance field-based GRAF [77] (see Sec. 2 for a discussion

[1] Details can be found in the supplementary material.

[2] We do not apply random cropping for [32] and [77] as we find that they cannot handle scenes with non-centered objects (see supplementary).

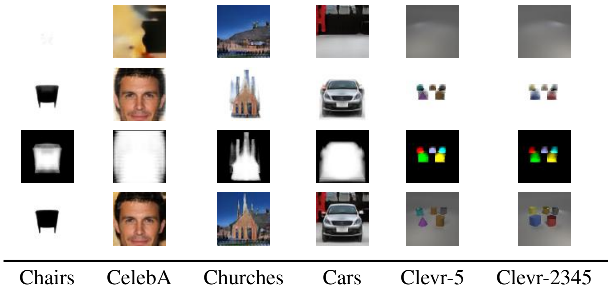


Figure 5: **Scene Disentanglement.** From top to bottom, we show only backgrounds, only objects, color-coded object alpha maps, and the final synthesized images at 64^2 pixel resolution. Disentanglement emerges without supervision, and the model learns to generate plausible backgrounds although the training data only contains images with objects.

Figure 6: **Training Progression.** We show renderings of our model on *Clevr-2345* at 256^2 pixels after 0, 1, 2, 3, 10, and 100-thousand iterations. Unsupervised disentanglement emerges already at the very beginning of training.

	Cats	CelebA	Cars	Chairs	Churches
2D GAN [58]	18	15	**16**	59	19
Plat. GAN [32]	318	321	299	199	242
BlockGAN [64]	47	69	41	41	28
HoloGAN [63]	27	25	17	59	31
GRAF [77]	26	25	39	34	38
Ours	**8**	**6**	**16**	**20**	**17**

Table 1: **Quantitative Comparison.** We report the FID score (↓) at 64^2 pixels for baselines and our method.

	CelebA-HQ	FFHQ	Cars	Churches	Clevr-2
HoloGAN [63]	61	192	34	58	241
w/o 3D Conv	33	70	49	66	273
GRAF [77]	49	59	95	87	106
Ours	**21**	**32**	**26**	**30**	**31**

Table 2: **Quantitative Comparison.** We report the FID score (↓) at 256^2 pixels for the strongest 3D-aware baselines and our method.

2D GAN	Plat. GAN	BlockGAN	HoloGAN	GRAF	Ours
1.69	381.56	4.44	7.80	0.68	0.41

Table 3: **Network Parameter Comparison.** We report the number of generator network parameters in million.

of the methods). We further compare against HoloGAN w/o 3D Conv, a variant of [63] proposed in [77] for higher resolutions. We additionally report a ResNet-based [28] 2D GAN [58] for reference.

Metrics: We report the Frechet Inception Distance (FID) score [33] to quantify image quality. We use 20,000 real and fake samples to calculate the FID score.

4.1. Controllable Scene Generation

Disentangled Scene Generation: We first analyze to which degree our model learns to generate disentangled scene representations. In particular, we are interested if objects are disentangled from the background. Towards this goal, we exploit the fact that our composition operator is a simple addition operation (Eq. 8) and render individual components and object alpha maps (Eq. 10). Note that while we always render the feature image at 16^2 during training, we can choose arbitrary resolutions at test time.

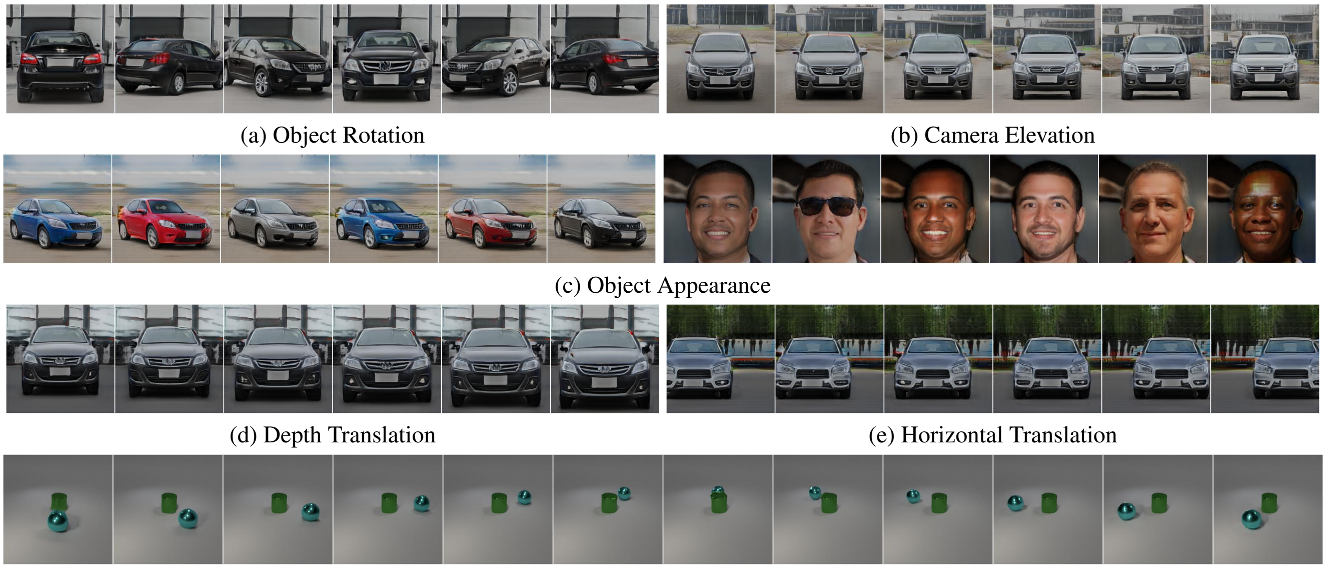
Fig. 5 suggests that our method disentangles objects from the background. Note that this disentanglement emerges without any supervision, and the model learns to generate plausible backgrounds without ever having seen a pure background image, implicitly solving an inpainting task. We further observe that our model correctly disentangles individual objects when trained on multi-object scenes with fixed or varying number of objects. We further find that unsupervised disentanglement is a property of our model which emerges already at the very beginning of training (Fig. 6). Note how our model synthesizes individual objects before spending capacity on representing the background.

Controllable Scene Generation: As individual components of the scene are correctly disentangled, we analyze how well they can be controlled. More specifically, we are interested if individual objects can be rotated and translated, but also how well shape and appearance can be controlled. In Fig. 7, we show examples in which we control the scene during image synthesis. We rotate individual objects, translate them in 3D space, or change the camera elevation. By modeling shape and appearance for each entity with a different latent code, we are further able to change the objects' appearances without altering their shape.

Generalization Beyond Training Data: The learned compositional scene representations allow us to generalize outside the training distribution. For example, we can increase the translation ranges of objects or add more objects than there were present in the training data (Fig. 8).

4.2. Comparison to Baseline Methods

Comparing to baseline methods, our method achieves similar or better FID scores at both 64^2 (Tab. 1) and 256^2 (Tab. 2) pixel resolutions. Qualitatively, we observe that while all approaches allow for controllable image synthesis on datasets of limited complexity, results are less consistent for the baseline methods on more complex scenes



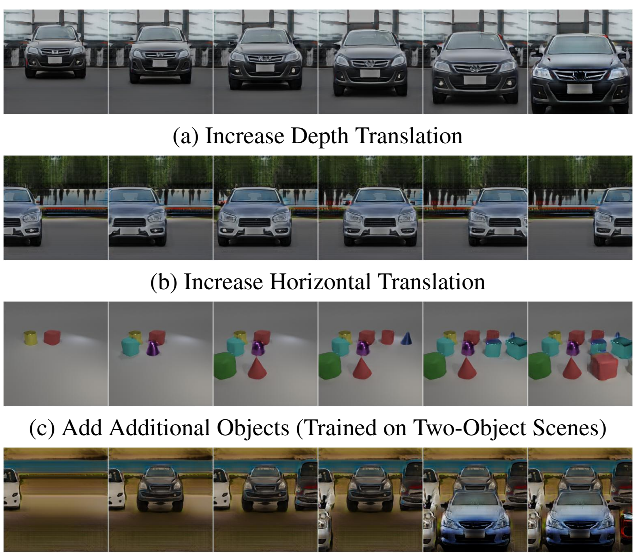
(a) Object Rotation (b) Camera Elevation

(c) Object Appearance

(d) Depth Translation (e) Horizontal Translation

(f) Circular Translation of One Object Around Another Object

Figure 7: **Controllable Scene Generation at 256^2 Pixel Resolution.** Controlling the generated scenes during image synthesis: Here we rotate or translate objects, change their appearances, and perform complex operations like circular translations.



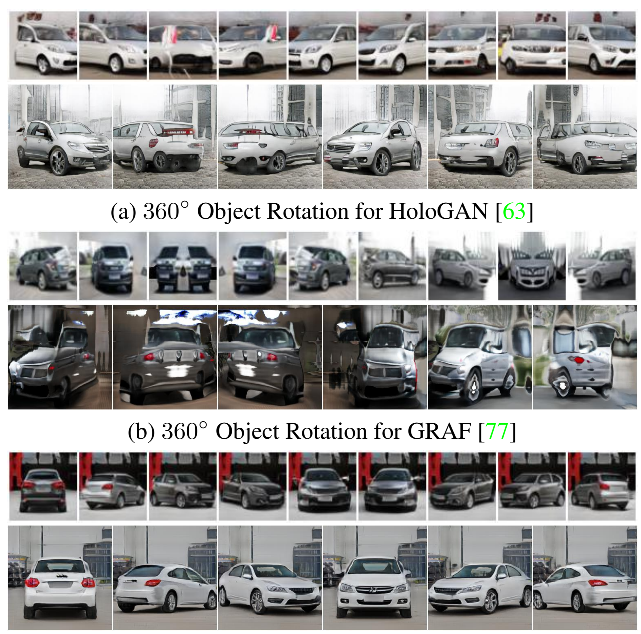
(a) Increase Depth Translation

(b) Increase Horizontal Translation

(c) Add Additional Objects (Trained on Two-Object Scenes)

(d) Add Additional Objects (Trained on Single-Object Scenes)

Figure 8: **Generalization Beyond Training Data.** As individual objects are correctly disentangled, our model allows for generating out of distribution samples at test time. For example, we can increase the translation ranges or add more objects than there were present in the training data.



(a) 360° Object Rotation for HoloGAN [63]

(b) 360° Object Rotation for GRAF [77]

(c) 360° Object Rotation for Our Method

Figure 9: **Qualitative Comparison.** Compared to baseline methods, we achieve more consistent image synthesis for complex scenes with cluttered background at 64^2 (top rows) and 256^2 (bottom rows) pixel resolutions. Note that we disentangle the object from the background and are able to rotate only the object while keeping the background fixed.

with cluttered backgrounds. Further, our model disentangles the object from the background, such that we are able to control the object independent of the background (Fig. 9).

We further note that our model achieves similar or better FID scores than the ResNet-based 2D GAN [58] despite fewer network parameters (0.41m compared to 1.69m). This confirms our initial hypothesis that using a 3D representation as inductive bias results in better outputs. Note that for fair comparison, we only report methods which are

Full	-Skip	-Act.	+NN. RGB Up.	+Bi. Feat. Up.
16.16	16.66	21.61	17.28	20.68

Table 4: **Ablation Study.** We report FID (↓) on *CompCars* without RGB skip connections (-Skip), without final activation (-Act.), with nearest neighbor instead of bilinear image upsampling (+ NN. RGB Up.), and with bilinear instead of nearest neighbor feature upsampling (+ Bi. Feat. Up.).

Figure 10: **Neural Renderer.** We change the background while keeping the foreground object fixed for our method at 256^2 pixel resolution. Note how the neural renderer realistically adapts the objects' appearances to the background.

(a) 0° Rotation for Axis-Aligned Positional Encoding [61]

(b) 0° Rotation for Random Fourier Features [82]

Figure 11: **Canonical Pose.** In contrast to random Fourier features [82], axis-aligned positional encoding (1) encourages the model to learn objects in a canonical pose.

similar wrt. network size and training time (see Tab. 3).

4.3. Ablation Studies

Importance of Individual Components: The ablation study in Tab. 4 shows that our design choices of RGB skip connections, final activation function, and selected upsampling types improve results and lead to higher FID scores.

Effect of Neural Renderer: A key difference to [77] is that we combine volume with neural rendering. The quantitative (Tab. 1 and 2) and qualitative comparisons (Fig. 9) indicate that our approach leads to better results, in particular for complex, real-world data. Our model is more expressive and can better handle the complexity of real scenes, e.g. note how the neural renderer realistically adapts object appearances to the background (Fig. 10). Further, we observe a rendering speed up: compared to [77], total rendering time is reduced from 110.1ms to 4.8ms, and from 1595.0ms to 5.9ms for 64^2 and 256^2 pixels, respectively.

Positional Encoding: We use axis-aligned positional encoding for the input point and viewing direction (Eq. 1). Surprisingly, this encourages the model to learn canoncial

Figure 12: **Dataset Bias.** Eye and hair rotation are examples for dataset biases: They primarily face the camera, and our model tends to entangle them with the object rotation.

representations as it introduces a bias to align the object axes with highest symmetry with the canonical axes which allows the model to exploit object symmetry (Fig. 11).

4.4. Limitations

Dataset Bias: Our method struggles to disentangle factors of variation if there is an inherent bias in the data. We show an example in Fig. 12: In the celebA-HQ dataset, the eye and hair orientation is predominantly pointing towards the camera, regardless of the face rotation. When rotating the object, the eyes and hair in our generated images do not stay fixed but are adjusted to meet the dataset bias.

Object Transformation Distributions: We sometimes observe disentanglement failures, e.g. for *Churches* where the background contains a church, or for *CompCars* where the foreground contains background elements (see Sup. Mat.). We attribute these to mismatches between the assumed uniform distributions over camera poses and object-level transformations and their real distributions.

5. Conclusion

We present *GIRAFFE*, a novel method for controllable image synthesis. Our key idea is to incorporate a compositional 3D scene representation into the generative model. By representing scenes as compositional generative neural feature fields, we disentangle individual objects from the background as well as their shape and appearance without explicit supervision. Combining this with a neural renderer yields fast and controllable image synthesis. In the future, we plan to investigate how the distributions over object-level transformations and camera poses can be learned from data. Further, incorporating supervision which is easy to obtain, e.g. predicted object masks, is a promising approach to scale to more complex, multi-object scenes.

Acknowledgement

This work was supported by an NVIDIA research gift. We thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting MN. AG was supported by the ERC Starting Grant LEGO-3D (850533) and DFG EXC number 2064/1 - project number 390727645.

References

[1] Rameen Abdal, Peihao Zhu, Niloy J. Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *arXiv.org*, 2008.02401, 2020. 2

[2] Hassan Alhaija, Siva Mustikovela, Andreas Geiger, and Carsten Rother. Geometric image synthesis. In *Proc. of the Asian Conf. on Computer Vision (ACCV)*, 2018. 2

[3] Titas Anciukevicius, Christoph H. Lampert, and Paul Henderson. Object-centric image generation with factored depths, locations, and appearances. *arXiv.org*, 2004.00642, 2020. 2

[4] Relja Arandjelovic and Andrew Zisserman. Object discovery with a copy-pasting GAN. *arXiv.org*, 1905.11369, 2019. 2

[5] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 35(8):1798–1828, 2013. 1

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2019. 1, 2

[7] Christopher P. Burgess, Loïc Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matthew M Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv.org*, 1901.11390, 2019. 2

[8] Rohan Chabra, Jan Eric Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard A. Newcombe. Deep local shapes: Learning local SDF priors for detailed 3d reconstruction. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020. 2

[9] Xi Chen, Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 1, 2

[10] Xuelin Chen, Daniel Cohen-Or, Baoquan Chen, and Niloy J. Mitra. Neural graphics pipeline for controllable image generation. *arXiv.org*, 2006.10569, 2020. 2

[11] Zhiqin Chen, Kangxue Yin, Matthew Fisher, Siddhartha Chaudhuri, and Hao Zhang. BAE-NET: branched autoencoder for shape co-segmentation. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2

[12] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3

[13] Julian Chibane, Aymen Mir, and Gerard Pons-Moll. Neural unsigned distance fields for implicit function learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[14] Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[15] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[16] Edo Collins, Raja Bala, Bob Price, and Sabine Süsstrunk. Editing in style: Uncovering the local semantics of gans. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[17] Robert A. Drebin, Loren C. Carpenter, and Pat Hanrahan. Volume rendering. In *ACM Trans. on Graphics*, 1988. 4

[18] Sébastien Ehrhardt, Oliver Groth, Aron Monszpart, Martin Engelcke, Ingmar Posner, Niloy J. Mitra, and Andrea Vedaldi. RELATE: physically plausible multi-object scene synthesis using structured latent spaces. *arXiv.org*, 2007.01272, 2020. 2

[19] Martin Engelcke, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner. GENESIS: generative scene inference and sampling with object-centric latent representations. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2020. 2

[20] S. M. Ali Eslami, Danilo Jimenez Rezende, Frédéric 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 C. Rabinowitz, Helen King, Chloe Hillier, Matt M. Botvinick, Daan Wierstra, Koray Kavukcuoglu, and Demis Hassabis. Neural scene representation and rendering. *Science*, 360:1204–1210, 2018. 3

[21] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3d shape induction from 2d views of multiple objects. In *Proc. of the International Conf. on 3D Vision (3DV)*, 2017. 2

[22] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2

[23] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2

[24] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 1, 2, 5

[25] Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent independent mechanisms. *arXiv.org*, 1909.10893, 2019. 1

[26] Klaus Greff, Raphaël Lopez Kaufmann, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *Proc. of the International Conf. on Machine learning (ICML)*, 2019. 2

[27] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable GAN controls. *arXiv.org*, 2004.02546, 2020. 2

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE*

Conf. on Computer Vision and Pattern Recognition (CVPR), 2016. 6

[29] Paul Henderson and Vittorio Ferrari. Learning single-image 3d reconstruction by generative modelling of shape, pose and shading. *International Journal of Computer Vision (IJCV)*, 2019. 2

[30] Paul Henderson and Christoph H. Lampert. Unsupervised object-centric video generation and decomposition in 3d. *arXiv.org*, 2007.06705, 2020. 2

[31] Paul Henderson, Vagia Tsiminaki, and Christoph H. Lampert. Leveraging 2d data to learn textured 3d mesh generation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[32] Philipp Henzler, Niloy J Mitra, , and Tobias Ritschel. Escaping plato's cave: 3d shape from adversarial rendering. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2, 5, 6

[33] 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 *Advances in Neural Information Processing Systems (NIPS)*, 2017. 6

[34] Ali Jahanian, Lucy Chai, and Phillip Isola. On the "steerability" of generative adversarial networks. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2020. 2

[35] Chiyu Max Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas A. Funkhouser. Local implicit grid representations for 3d scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[36] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[37] James T. Kajiya and Brian Von Herzen. Ray tracing volume densities. In *ACM Trans. on Graphics*, 1984. 4

[38] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2018. 5

[39] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 5

[40] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 4

[41] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey. *arXiv.org*, 2006.12057, 2020. 3

[42] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[43] Hanock Kwak and Byoung-Tak Zhang. Generating images part by part with composite generative adversarial networks. *arXiv.org*, 1607.05387, 2016. 1

[44] Wonkwang Lee, Donggyun Kim, Seunghoon Hong, and Honglak Lee. High-fidelity synthesis with disentangled representation. *arXiv.org*, 2001.04296, 2020. 1

[45] Nanbo Li, Robert Fisher, et al. Learning object-centric representations of multi-object scenes from multiple views. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[46] Yiyi Liao, Katja Schwarz, Lars Mescheder, and Andreas Geiger. Towards unsupervised learning of generative models for 3d controllable image synthesis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[47] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 4

[48] Ming-Yu Liu, Xun Huang, Jiahui Yu, Ting-Chun Wang, and Arun Mallya. Generative adversarial networks for image and video synthesis: Algorithms and applications. *arXiv.org*, 2008.02793, 2020. 1

[49] Shichen Liu, Weikai Chen, Tianye Li, and Hao Li. Soft rasterizer: Differentiable rendering for unsupervised single-view mesh reconstruction. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 3

[50] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3d supervision. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 2

[51] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. DIST: rendering deep implicit signed distance function with differentiable sphere tracing. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[52] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2015. 5

[53] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proc. of the International Conf. on Machine learning (ICML)*, 2019. 1

[54] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1

[55] Sebastian Lunz, Yingzhen Li, Andrew W. Fitzgibbon, and Nate Kushman. Inverse graphics GAN: learning to generate 3d shapes from unstructured 2d data. arXiv.org, 2020. 2

[56] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. of the International Conf. on Machine learning (ICML) Workshops*, 2013. 4

[57] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. *arXiv.org*, 2008.02268, 2020. 4

[58] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *Proc. of the International Conf. on Machine learning (ICML)*, 2018. 5, 6, 7

[59] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3

[60] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2

[61] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020. 2, 3, 4, 8

[62] Thu Nguyen-Phuoc, Chuan Li, Stephen Balaban, and Yong-Liang Yang. Rendernet: A deep convolutional network for differentiable rendering from 3d shapes. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 3

[63] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2, 5, 6, 7

[64] Thu Nguyen-Phuoc, Christian Richardt, Long Mai, Yong-Liang Yang, and Niloy Mitra. Blockgan: Learning 3d object-aware scene representations from unlabelled images. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2, 5, 6

[65] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2

[66] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[67] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2

[68] Michael Oechsle, Michael Niemeyer, Christian Reiser, Lars Mescheder, Thilo Strauss, and Andreas Geiger. Learning implicit surface light fields. In *Proc. of the International Conf. on 3D Vision (3DV)*, 2020. 5

[69] Jeong Joon Park, Peter Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 3

[70] Keunhong Park, Konstantinos Rematas, Ali Farhadi, and Steven M. Seitz. Photoshape: Photorealistic materials for large-scale shape collections. *Communications of the ACM*, 2018. 5

[71] William S. Peebles, John Peebles, Jun-Yan Zhu, Alexei A. Efros, and Antonio Torralba. The hessian penalty: A weak prior for unsupervised disentanglement. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020. 1, 2

[72] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020. 2

[73] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2016. 5

[74] Scott Reed, Kihyuk Sohn, Yuting Zhang, and Honglak Lee. Learning to disentangle factors of variation with manifold interaction. In *Proc. of the International Conf. on Machine learning (ICML)*, 2014. 1

[75] Danilo Jimenez Rezende, S. M. Ali Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3d structure from images. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2

[76] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2

[77] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2, 3, 4, 5, 6, 7, 8

[78] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[79] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[80] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[81] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems (NIPS)*, 2019. 2, 3

[82] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ra-

mamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 3, 8

[83] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason M. Saragih, Matthias Nießner, Rohit Pandey, Sean Ryan Fanello, Gordon Wetzstein, Jun-Yan Zhu, Christian Theobalt, Maneesh Agrawala, Eli Shechtman, Dan B. Goldman, and Michael Zollhöfer. State of the art on neural rendering. *Computer Graphics Forum*, 2020. 3

[84] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: image synthesis using neural textures. *ACM Trans. on Graphics*, 2019. 3

[85] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012. 5

[86] Sjoerd van Steenkiste, Karol Kurach, Jürgen Schmidhuber, and Sylvain Gelly. Investigating object compositionality in generative adversarial networks. *Neural Networks*, 2020. 2

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

[88] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2

[89] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv.org*, 1505.00853, 2015. 4

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

[91] Jiaolong Yang and Hongdong Li. Dense, accurate optical flow estimation with piecewise parametric model. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 5

[92] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[93] Yasin Yazici, Chuan-Sheng Foo, Stefan Winkler, Kim-Hui Yap, Georgios Piliouras, and Vijay Chandrasekhar. The unusual effectiveness of averaging in GAN training. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2019. 5

[94] 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.org*, 1506.03365, 2015. 5

[95] Li Zhang, Brian Curless, Aaron Hertzmann, and Steven M. Seitz. Shape and motion under varying illumination: Unifying structure from motion, photometric stereo, and multiview stereo. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2003. 5

[96] Yuxuan Zhang, Wenzheng Chen, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. Image gans meet differentiable rendering for inverse graphics and interpretable 3d neural rendering. *arXiv.org*, 2010.09125, 2020. 2

[97] Bo Zhao, Bo Chang, Zequn Jie, and Leonid Sigal. Modular generative adversarial networks. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. 1

[98] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Learning a discriminative model for the perception of realism in composite images. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2015. 1

[99] 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 *Advances in Neural Information Processing Systems (NIPS)*, 2018. 2