

CAMPARI: Camera-Aware Decomposed Generative Neural Radiance Fields

Michael Niemeyer^{1,2} Andreas Geiger^{1,2}

¹Max Planck Institute for Intelligent Systems, Tübingen ²University of Tübingen

{firstname.lastname}@tue.mpg.de

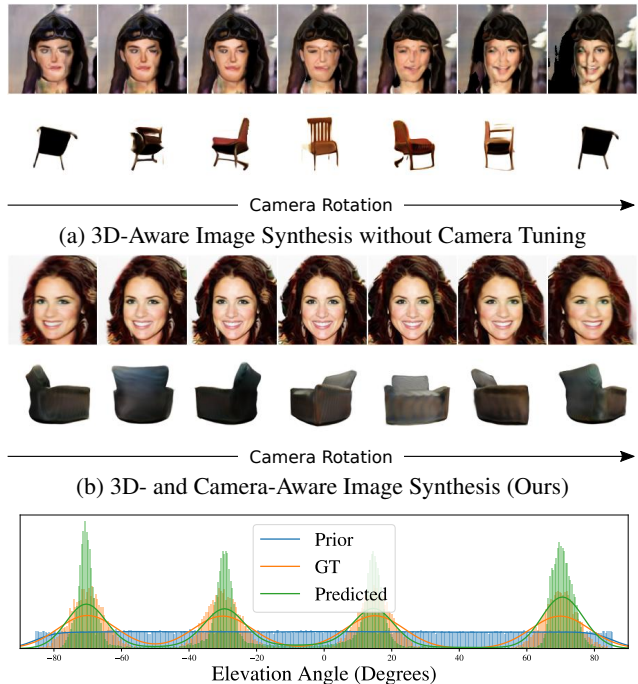
Abstract

Tremendous progress in deep generative models has led to photorealistic image synthesis. While achieving compelling results, most approaches operate in the two-dimensional image domain, ignoring the three-dimensional nature of our world. Several recent works therefore propose generative models which are 3D-aware, i.e., scenes are modeled in 3D and then rendered differentially to the image plane. While this leads to impressive 3D consistency, the camera needs to be modelled as well and we show in this work that these methods are sensitive to the choice of prior camera distributions. Current approaches assume fixed intrinsics and predefined priors over camera pose ranges, and parameter tuning is typically required for real-world data. If the data distribution is not matched, results degrade significantly. Our key hypothesis is that learning a camera generator jointly with the image generator leads to a more principled approach to 3D-aware image synthesis. Further, we propose to decompose the scene into a background and foreground model, leading to more efficient and disentangled scene representations. While training from raw, unposed image collections, we learn a 3D- and camera-aware generative model which faithfully recovers not only the image but also the camera data distribution. At test time, our model generates images with explicit control over the camera as well as the shape and appearance of the scene.

1. Introduction

Deep generative models [17, 30] are able to synthesize photorealistic images at high resolutions. While state-of-the-art models [4, 10, 11, 28, 29] produce impressive results, most approaches lack control over the generation process. Control, however, is one key aspect required in many applications where generative models can be used.

To tackle this problem, many works investigate how architectures and training regimes can be improved to achieve more controllable image synthesis [7, 18, 28, 31, 32, 35, 55, 56, 74, 75]. Most approaches, however, operate on the two-dimensional image plane and hence do not consider the



(c) Learned Camera Elevation Distribution for Chairs

Figure 1: **Overview.** We propose a 3D and camera-aware generative model for controllable image synthesis. We show that current 3D-aware models are sensitive to the chosen camera priors and results degrade if the data distribution is not matched (1a). In contrast, we learn a camera generator jointly with the image generator. While training from raw, unstructured image collections, we faithfully recover the camera distribution (1c) and generate 3D consistent representations (1b) with explicit control over the camera viewpoint as well as the shape and appearance of the scene.

three-dimensional nature of our world. While latent factors of variation representing e.g. object rotation or translation may be found, full 3D disentanglement is hard to achieve.

In contrast, recent works [5, 24, 34, 47–49, 59] incorporate three-dimensionality as an inductive bias into the generative model. This leads to impressive, 3D-aware image synthesis in which the camera viewpoint can explicitly be controlled

at test time. However, we find that these approaches are sensitive to the prior camera distributions. Most works use fixed intrinsics and the true camera distributions for synthetic data or a uniform distribution over predefined ranges for real-world image collections. As a result, these methods are either limited to simple data or typically require parameter tuning for real-world datasets. Further, as a principled approach for obtaining the pose distribution is missing, results degrade if the distribution is not matched (see Fig. 1a).

Contribution: We propose Camera-Aware Decomposed Generative Neural Radiance Fields (*CAMPARI*), a novel generative model for 3D- and camera-aware image synthesis which is trained from raw, unposed image collections. Our key idea is to learn a camera generator jointly with the image generator. This allows us to apply our method to datasets with more complex camera distributions and, in contrast to previous works, requires no tuning of camera pose ranges. We further propose to decompose the scene into a foreground and a background model, leading to a more efficient and disentangled scene representation. We find that our model is able to learn 3D consistent representations (Fig. 1b) and to faithfully recover the camera distribution (Fig. 1c). At test time, we can generate new scenes in which we have explicit control over the camera viewpoint as well the shape and appearance of the scene while training from raw, unposed image collections only.

2. Related Work

Generative Adversarial Networks: State-of-the-art Generative Adversarial Networks (GANs) [17] allow for photorealistic image generation at high resolutions [3, 4, 10, 11, 28, 29]. As many applications require control mechanisms during image synthesis, a variety of works investigate how factors of variation can be disentangled without explicit supervision, e.g., by modifying the training objective [7, 55] or network architecture [28, 29], or discovering factors of variation in latent spaces of pre-trained generative models [1, 12, 16, 19, 25, 60, 73]. While achieving impressive results, the aforementioned works build on 2D-based convolutional or coordinate-based networks and hence model the image process in the two-dimensional image domain. In this work, we advocate exploiting the fact that we know that our world is three-dimensional by combining a 3D generator with differentiable volume rendering.

Neural Scene Representations: Using coordinate-based neural representations to represent 3D geometry gained popularity in learning-based 3D reconstruction [8, 9, 15, 43, 44, 50, 52, 53, 58] and several works [37, 38, 51, 61, 68] propose differentiable rendering techniques for them. Mildenhall et al. [45] propose Neural Radiance Fields (NeRFs) in which they combine a coordinate-based neural model with volume rendering for novel view synthesis. We use radiance fields as 3D representation in our generative model due to

their expressiveness and suitability for gradient-based learning. While discussed methods require camera poses as input, recent works [41, 62, 65, 70] propose to estimate them instead. However, all aforementioned approaches fit network weights to a single scene based on multi-view images of that scene, and do not have generation capabilities. Our model, in contrast, allows for controllable image synthesis of generated scenes and is trained from unstructured image collections with only a single, unposed image per scene.

3D-Aware Image Synthesis: Many recent approaches investigate how a 3D representation can be incorporated into the generator model [5, 14, 21–24, 34, 40, 46, 48, 49, 57, 59]. While some use additional supervision [2, 6, 66, 67, 76], in the following, we focus on methods that are trained from unposed image collections similar to our approach. While voxel-based representations [24] in combination with differentiable rendering lead to 3D controllable image synthesis, the visual quality is limited due to voxels’ cubic memory growth. Voxelized feature grids [34, 46, 48] with neural 2D rendering lead to impressive results, but training is less stable and results less multi-view consistent due to the learnable projection. Most related to our approach are [5, 34, 49, 59]. CIS [34] requires pure background images as additional supervision which are hard to obtain for real-world scenes. GRAF [59], GIRAFFE [49], and π -GAN [5] use fixed camera intrinsics and pose distributions over predefined rotation, elevation, and translation ranges. While this allows for 3D consistent image synthesis on synthetic data, it requires a significant amount of tuning for real-world scenes and results degrade if the data distribution is not matched (see Fig. 1a). In contrast, we learn a 3D- and camera-aware generative model by jointly estimating 3D representations and camera distributions.

3. Method

Our goal is a 3D- and camera-aware generative model that is trained from raw image collections. We first discuss our 3D-aware image generator model in Section 3.1. Next, we describe how a camera generator can be learned jointly to train a 3D- and camera-aware generative model in Section 3.2. Finally, we describe our training procedure and implementation details in Section 3.3 and 3.4. Fig. 2 contains an overview of our method.

3.1. 3D-Aware Image Generator

Scene Representation: Our goal is to learn a generative model of single object scenes with background. This allows us to incorporate prior knowledge into the scene representation for more efficient and disentangled representations. We represent scenes using a foreground and a background model. More specifically, we partition 3D space \mathbb{R}^3 into a foreground $\mathcal{M}_{fg} \subset \mathbb{R}^3$ and a background $\mathcal{M}_{bg} \subset \mathbb{R}^3$,

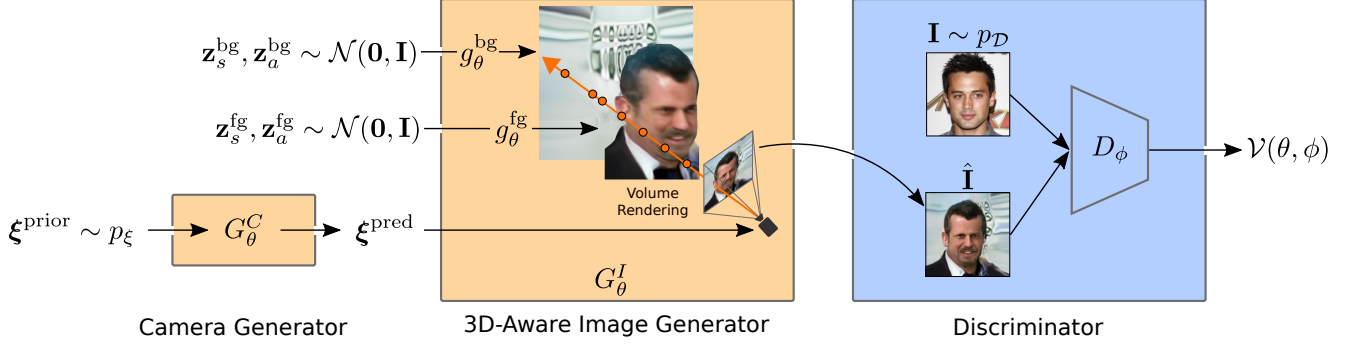


Figure 2: **CAMPARI**. During training, we sample a prior camera $\xi^{\text{prior}} \sim p_\xi$ and pass it to our camera generator G_θ^C which predicts camera ξ^{pred} . Next, we pass the predicted camera together with latent shape and appearance codes $\mathbf{z}_s^{\text{fg}}, \mathbf{z}_a^{\text{fg}}, \mathbf{z}_s^{\text{bg}}, \mathbf{z}_a^{\text{bg}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to our 3D-aware image generator G_θ^I . We then differentially render the image $\hat{\mathbf{I}}$ of the scene for camera ξ^{pred} using volume rendering. Finally, our discriminator D_ϕ takes as input the generated image $\hat{\mathbf{I}}$ and real image \mathbf{I} drawn from the data distribution $p_{\mathcal{D}}$ and predicts whether they are real or fake. While training from raw image collections, at test time, we have explicit control over the camera and latent shape and appearance codes allowing for 3D- and camera-aware image synthesis.

where the foreground is encapsulated by a sphere¹ of radius $r_{\text{fg}} < 1$

$$\mathcal{M}_{\text{fg}} = \{\mathbf{x} \in \mathbb{R}^3 \mid \|\mathbf{x}\|_2 \leq r_{\text{fg}}\} \quad (1)$$

and the background is everything outside the unit sphere

$$\mathcal{M}_{\text{bg}} = \{\mathbf{x} \in \mathbb{R}^3 \mid 1 \leq \|\mathbf{x}\|_2\} \quad (2)$$

We define the space of possible camera locations to be the space between fore- and background

$$\mathcal{M}_{\text{cam}} = \{\mathbf{x} \in \mathbb{R}^3 \mid r_{\text{fg}} < \|\mathbf{x}\|_2 < 1\} \quad (3)$$

As a result, the foreground is in front, and the background behind every possible camera. In contrast to [71] where the foreground is assumed to be within the unit sphere, our representation exhibits a stronger bias for single-object scenes while not being limited to specific scenarios as we do not enforce hard constraints.

Object Representation: To represent the fore- and background, we use conditional neural radiance fields [45, 59] which are multilayer perceptrons (MLPs) mapping a 3D point $\mathbf{x} \in \mathbb{R}^3$ and viewing direction $\mathbf{d} \in \mathbb{S}^2$ together with latent shape and appearance codes $\mathbf{z}_s, \mathbf{z}_a \in \mathbb{R}^{L_z}$ to a density $\sigma \in \mathbb{R}^+$ and RGB color $\mathbf{c} \in \mathbb{R}^3$:

$$g_\theta : \mathbb{R}^{L_x} \times \mathbb{R}^{L_d} \times \mathbb{R}^{L_z} \times \mathbb{R}^{L_z} \rightarrow \mathbb{R}^+ \times \mathbb{R}^3 \quad (4)$$

$$(\gamma(\mathbf{x}), \gamma(\mathbf{d}), \mathbf{z}_s, \mathbf{z}_a) \mapsto (\sigma, \mathbf{c})$$

where θ indicates the network parameters, γ the positional encoding [45, 63] applied element-wise to \mathbf{x} and \mathbf{d} , L_x, L_d the output dimensions of the positional encodings, and L_z

¹In practice, we enforce the foreground to only be roughly inside the sphere of radius r_{fg} as we sample within the same near and far bounds for all pixels of the same image.

the latent code dimension. We use two separate networks for the fore- and background, and define our 3D scene representation as

$$g_\theta(\mathbf{x}, \mathbf{d}, \mathbf{z}_s, \mathbf{z}_a) = \begin{cases} g_\theta^{\text{fg}}(\mathbf{x}, \mathbf{d}, \mathbf{z}_s, \mathbf{z}_a), & \text{for } \mathbf{x} \in \mathcal{M}_{\text{fg}} \\ g_\theta^{\text{bg}}(\mathbf{x}, \mathbf{d}, \mathbf{z}_s, \mathbf{z}_a), & \text{for } \mathbf{x} \in \mathcal{M}_{\text{bg}} \end{cases} \quad (5)$$

To avoid cluttered notation, we always use the same θ to indicate network parameters.

Scene Rendering: To render images of our scene representation, classic volume rendering techniques can be used [26, 45] which are trivially differentiable. More specifically, for given camera intrinsics \mathbf{K} and extrinsics $[\mathbf{R}|\mathbf{t}]$, a pixel's color value $\mathbf{c}_{\text{final}}$ is calculated by integrating over the camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ within near and far bounds t_n, t_f

$$\mathbf{c}_{\text{final}} = \int_{t_n}^{t_f} T(t) \sigma(\mathbf{r}(t)) \mathbf{c}(\mathbf{r}(t), \mathbf{d}) dt \quad (6)$$

$$\text{where } T(t) = \exp\left(-\int_{t_n}^t \sigma(\mathbf{r}(s)) ds\right)$$

Scene Spacing Sampling: As no analytical solution exists for integral 6, it is commonly approximated using stratified sampling [45]. To faithfully render the scene, however, it is crucial that the numerical integration approximates the true integral well. As we decompose the scene into fore- and background, we are able to inject prior knowledge into the sampling process, thereby saving computational cost and encouraging disentanglement.

More specifically, for camera pose $[\mathbf{R}|\mathbf{t}]$, we sample points uniformly for the foreground within bounds

$$(t_n^{\text{fg}}, t_f^{\text{fg}}) = (\|\mathbf{t}\|_2 - r_{\text{fg}}, \|\mathbf{t}\|_2 + r_{\text{fg}}) \quad (7)$$

For the background, we adopt the inverted sphere parametrization from [71] where a 3D point \mathbf{x} outside the

unit sphere is described as

$$\mathbf{x}' = \left(\frac{\mathbf{x}}{\|\mathbf{x}\|_2}, \frac{1}{\|\mathbf{x}\|_2} \right) \in [-1, 1]^4 \quad (8)$$

We then sample points uniformly between 0 and 1 in *inverse depth* for the background [71]. This way, we do not need to assume the background to be within a predefined range, but sample space denser if it is nearer to the foreground.

3D-Aware Image Generator: We define our image generator G_θ^I as a function which renders an image of g_θ for given camera $\xi = (\mathbf{K}, [\mathbf{R}|\mathbf{t}])$ and fore-and background shape and appearance codes $\mathbf{z} = \{\mathbf{z}_s^{\text{fg}}, \mathbf{z}_s^{\text{bg}}, \mathbf{z}_a^{\text{fg}}, \mathbf{z}_a^{\text{bg}}\}$

$$\hat{\mathbf{I}} = G_\theta^I(\xi, \mathbf{z}) \quad (9)$$

where $\hat{\mathbf{I}}$ denotes the generated image.

3.2. Camera Generator

While incorporating a 3D representation into the generator leads to more controllable image synthesis, it also comes at a price: the camera and its pose distribution needs to be modeled as well. A key limitation of state-of-the-art 3D-aware image synthesis models [5, 24, 47–49, 59] is that camera intrinsics and the pose distribution are predefined. This requires tuning of the range parameters and leads to degraded results if the camera distribution is wrong. In the following, we describe how a camera generator can be learned jointly with the image generator to avoid tuning and to improve results if the camera distribution is unknown.

Camera Intrinsics: Assuming a pinhole camera model, we can express the intrinsics as

$$\mathbf{K} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (10)$$

where f_x, f_y indicate the focal lengths and $(c_x, c_y)^T$ the principal point. In this work, we assume the principal point to lie in the center of the image plane, and hence $(c_x, c_y) = (\frac{W}{2}, \frac{H}{2})$ where $H \times W$ indicates the image resolution. As a result, the camera intrinsics are reduced to

$$\xi^{\text{intr}} = (f_x, f_y) \in \mathbb{R}^2 \quad (11)$$

Camera Pose: We parameterize the camera pose as a location on a sphere with radius r_{cam} looking at the world-space origin $(0, 0, 0)^T \in \mathbb{R}^3$ and fixing the up-right position. The camera pose can hence be easily described using radius r_{cam} , a rotation angle $\alpha_r \in [-\pi, \pi]$ and an elevation angle $\alpha_e \in [-\frac{\pi}{2}, \frac{\pi}{2}]$:

$$\xi^{\text{pose}} = (r_{\text{cam}}, \alpha_r, \alpha_e) \in \mathbb{R}^3 \quad (12)$$



(a) Camera Generator without Residual Design



(b) Camera Generator with Residual Design (Ours)

Figure 3: Residual Camera Generator. We show camera rotation in a fixed range for our model using a camera generator with and without a residual connection to the sampled input camera. The residual design encourages exploration of a larger pose range from the beginning of training leading to more 3D consistent results.

We obtain $[\mathbf{R}|\mathbf{t}]$ from ξ^{pose} as the composition of the Euler rotation matrices resulting from α_r and α_e , respectively, and the translation vector resulting from r_{cam} . When operating on 360° rotation scenes, we represent α_r by a 2×2 matrix and project it to the special orthogonal group $SO(2)$ avoiding periodic boundary issues and ensuring well-behaved gradients [33].

Camera Generator: Our goal is to learn a camera generator G_θ^C in addition to the image generator to obtain a 3D- and camera-aware generative model. We define a camera as

$$\xi = (\xi^{\text{intr}}, \xi^{\text{pose}}) \quad (13)$$

We implement G_θ^C as an MLP which maps a *prior* camera $\xi^{\text{prior}} \sim p_\xi$ to a predicted camera $\xi^{\text{pred}} \in \mathbb{R}^3$. While the input to our camera generator could be of any dimension, using the same space as input and output allows us to incorporate prior knowledge and to design G_θ^C as a residual function [13, 20]. More specifically, we define

$$G_\theta^C(\xi^{\text{prior}}) = \xi^{\text{prior}} + \Delta\xi_\theta \quad \text{where} \quad \xi^{\text{prior}} \sim p_\xi \quad (14)$$

and θ indicates the network parameters. This way, we are able to encode prior knowledge into the prior distribution p_ξ which encourages our model to explore a wider range of camera poses directly from the start of training. It is important to note that as we train from raw, unposed image collections, our generator is not forced to learn 3D consistent representations and could therefore predict a static camera. We find that the residual design is key to avoid trivial solutions, i.e., not 3D consistent (see Fig. 3). In practice, we set p_ξ to a Gaussian or uniform prior, however, more complex prior distributions can be incorporated as well.

3.3. Training

Progressive Growing: To avoid excessive memory requirements and to improve training stability, we train our models

using progressive growing [5,27]. We start training on a low image resolution which allows the generator and discriminator to focus on coarse structures during early iterations and to train with higher batch sizes. As training progresses, we increase the image resolution until we reach the final resolution (128² pixels). Due to GPU memory limitations, we reduce the batch size in each progressive growing step.

Discriminator: We implement the discriminator using residual blocks [20] of CoordConv layers [36] similar to [5]. We progressively add new residual blocks when a new progressive growing step is reached. We follow [5,27] and fade in newly added layers to allow for a smooth transition.

Training: During each iteration, we first sample a camera from the prior $\xi_{\text{prior}} \sim p_{\xi}$. Next, we pass the sampled prior camera to our camera generator G_{θ}^C and obtain the predicted camera ξ_{pred} . Finally, we volume render the predicted image from camera ξ_{pred} and for sampled latent shape and appearance codes $\mathbf{z} = \{\mathbf{z}_s^{\text{fg}}, \mathbf{z}_s^{\text{bg}}, \mathbf{z}_a^{\text{fg}}, \mathbf{z}_a^{\text{bg}}\}$ which are all drawn from a unit Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$. We train our model with the non-saturating GAN objective [17] and R_1 gradient penalty [42]:

$$\begin{aligned} \mathcal{V}(\theta, \phi) = & \\ & \mathbb{E}_{\xi_{\text{prior}} \sim p_{\xi}, \mathbf{z} \sim p_{\mathbf{z}}} [f(D_{\phi}(G_{\theta}^I(G_{\theta}^C(\xi_{\text{prior}}), \mathbf{z})))] + \quad (15) \\ & \mathbb{E}_{\mathbf{I} \sim p_{\mathcal{D}}} [f(-D_{\phi}(\mathbf{I})) - \lambda \|\Delta D_{\phi}(\mathbf{I})\|^2] \end{aligned}$$

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

3.4. Implementation Details

Network Parametrization: We parameterize our radiance fields g_{θ}^{fg} and g_{θ}^{bg} as MLPs with 8 hidden layers of dimension 128, ReLU activation, and a skip connection to the fourth layer [53]. We concatenate the latent shape and appearance codes to the encoded input point and viewing direction, respectively [59]. We implement our camera generator G_{θ}^C as an MLP with 4 hidden layers of dimension 64 and ReLU activation, and we initialize the last layer’s biases as zeros and the weights from $\mathcal{N}(0, 0.05)$. After adding the learned offset to the prior pose (14), we clamp the output to be within a valid range (see sup. mat.). We parameterize our discriminator using 5 residual blocks [20] of CoordConv layers [36] with leaky ReLU activation similar to [5].

Training Procedure: We schedule the number of sample points along the ray [50] and, depending on the scene type, sample between 20 and 52 points on each ray at the final stage. We use the RMSprop optimizer [64] with learning rates of 5×10^{-4} and 1×10^{-4} for our generators and discriminator, respectively. We use exponential learning rate decay with a rate of 0.1 after 1.5×10^5 iterations [45]. For the generator weights, we use an exponential moving average [69] with decay 0.999. To ensure training stability,

we fix our camera generator for the first iterations. We start progressive growing at 32² pixels and double the resolution after 2×10^4 and 7×10^4 iterations. We use batch sizes of [64, 24, 20] for 180° and [64, 20, 15] for 360° rotation scenes. We train on a single NVIDIA V100 GPU.

4. Experiments

Datasets: We run experiments on the commonly-used datasets *Cats* [72], *CelebA* [39], and *Cars* [59]. In contrast to previous works on 3D-aware image synthesis [47,59] we use a center crop of the entire image for CelebA instead of a close-up region. Note that learning a consistent 3D representation becomes more challenging as the data variety is larger and ideally the model should disentangle fore- and background. We further create synthetic datasets *Chairs1* and *Chairs2* which consist of photorealistic renderings of the Photoshape chairs [54]. To test our method on complex camera pose distributions, we sample rotation and elevation angles from mixtures of Gaussians (see sup. mat. for details). For Cats and the synthetic datasets we only use a foreground model as they do not contain any background.

Baselines: We compare against the state-of-the-art 3D-aware methods HoloGAN [46] and GRAF [59] which are both suited for single-object scenes like our approach. In HoloGAN, scenes are represented as voxelized feature grids which are differentially rendered via a reshaping operation and learnable convolutional filters. Similar to us, GRAF uses radiance fields as 3D representation. Note that while for both methods, the goal is 3D-aware image synthesis, we also decompose the scene into fore- and background. Further, in contrast to us, both methods require hand-tuned cameras. We therefore report results for them in tuned and non-tuned settings where for the first, we use the ranges reported by the authors, and for the latter, we use the ranges of our prior distribution (see sup. mat. for details).

Camera Distributions: For Cats and CelebA, we use a Gaussian prior for both rotation and elevation, and for Carla and Chairs, we use a uniform distribution over the entire rotation and elevation range. For the camera radius and focal lengths, we use a Gaussian prior except for Chairs where we fix the focal length. Note that jointly optimizing camera intrinsics and extrinsics has many different valid solutions, but as we are interested in comparing our results against the ground truth, we fix the intrinsics.

Metrics: We adhere to common practice and report the Frechet Inception Distance (FID) to quantify image quality.

4.1. Results

How does our approach compare to baseline methods?

In Tab. 1 and Fig. 4 we show quantitative and qualitative comparisons of our method to baselines. Although our model does not require tuning of camera parameters, we

	Tuned?	Cats	CelebA	Carla	Chairs1	Chairs2
HGAN [46]	yes	34	67	153	-	-
	no	42	83	169	124	116
GRAF [59]	yes	21	38	28	-	-
	no	72	74	90	55	53
Ours	no	23	28	39	31	33

Table 1: **Quantitative Comparison.** We report FID (\downarrow) for baselines and our method at 128^2 pixel resolution.



(a) Rotation for GRAF [59] (Camera Parameters Tuned)



(b) Rotation for GRAF [59] (Camera Parameters not Tuned)



(c) Rotation for Ours (No Tuning Required)

Figure 4: **Qualitative Comparison.** State-of-the-art 3D-aware models typically require hand-tuned camera parameters (Fig. 4a), and results degrade if the data distribution is not matched (Fig. 4b). In contrast, we learn a camera generator jointly with the image generator leading to more 3D consistent results (Fig. 4c) while no tuning is required.

achieve similar or better performance on datasets for which baseline methods are tuned. We find that the results of baselines indeed depend on tuned camera parameters, and their performances drop if the data’s camera distribution is not matched. We further observe that the performance drop is more significant for GRAF [59] than for HoloGAN [46] which we attribute to HoloGAN’s learnable projection. It introduces multi-view inconsistencies [5, 59], but the model becomes more robust against wrong cameras [49]. In contrast, our model is able to learn the camera distribution and achieves 3D consistent results without the need of a learnable projection. We conclude that learning a camera generator jointly with the image generator is indeed beneficial for obtaining high-quality 3D-aware image synthesis, in particular if the camera distribution is unknown.

What does the camera generator learn? In Fig. 5 we visualize learned camera distributions. We can see that our camera generator learns to deform the prior distribution in a meaningful way. For example, while starting with a uniform prior over the entire rotation and elevation range, our model adapts the camera elevation to lie on the upper hemisphere for the Car dataset as it only contains positive camera eleva-

Ours	-Cam. Gen.	-BG	-Intr.	+Patch Dis.
28	54	41	32	40

Table 2: **Ablation Study.** We find that removing the camera generator (-Cam. Gen.), the background model (-BG), and fixing the intrinsics distribution (-Intr.) worsens the results in terms of FID on CelebA. FID also drops when using a patch discriminator [59] instead of progressive growing.

Method	π -GAN	π -GAN	Ours + π -GAN
Tuned?	yes	no	no
FID	32	101	33

Table 3: **π -GAN as Image Generator.** Our camera generator is agnostic to the image generator parameterization and can be used with other image generators, e.g. π -GAN [5].

tion angles. Further, our model correctly approximates the more complicated marginals of the Chairs datasets. We observe that for elevation, the predicted distribution is closer to the ground truth than for rotation. First, arbitrarily shifted rotation distributions are equally-valid solutions as we learn them unsupervised. Further, we hypothesize that object symmetry causes structural changes to be more dominant along the upward direction, and as a result, the model is enforced more strongly to match the correct elevation distribution. It is important to note that as we train our model from raw, unstructured data collections, inferring the correct camera distribution is performed completely unsupervised.

How important is the camera generator? In Tab. 2 we report results for our method with and without a camera generator. We observe that learning a camera generator jointly with the image generator leads to improved results. Qualitatively, we find that the results of our full model are more 3D consistent compared to using a camera generator without the residual design. The residual design encourages exploration of wider pose ranges and avoids local minima of small rotation and elevation ranges (see Fig. 3).

How sensitive is the camera generator? In Tab. 3 we report results on CelebA for our camera generator with π -GAN [5] as the image generator. While using non-tuned camera distributions leads to a significant performance drop similar to the other baselines, adding our camera generator leads to similar performance as π -GAN (tuned) while no careful tuning is required (see Tab. 3). Our method is hence agnostic to the image generator parameterization. Similarly, in Fig. 6 we investigate the camera generator’s sensitivity to the prior distribution. We find that we obtain similar predicted distributions for different priors, i.e. the camera generator is not sensitive to the prior choice.

Does our model disentangle factors of variation?

In Fig. 7 we show examples for how our model disentangles camera viewpoint, fore- from background and the shape and

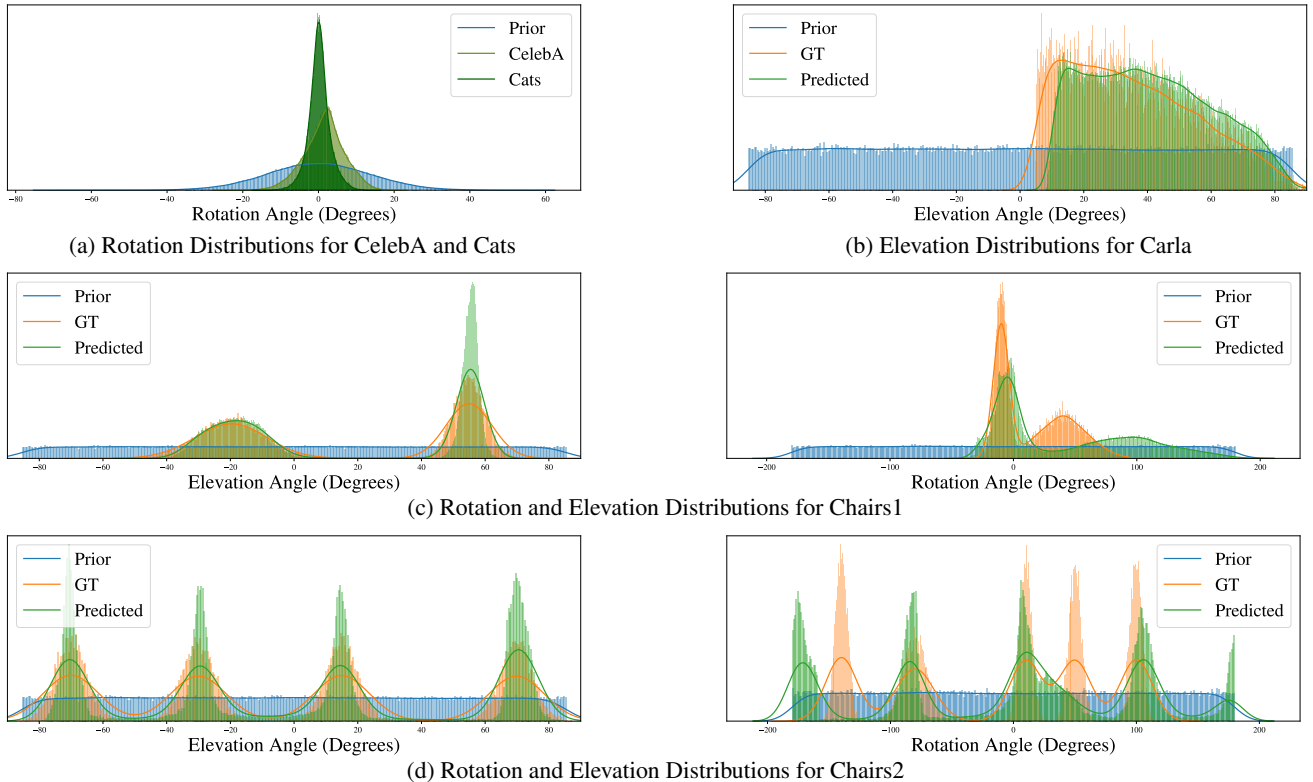


Figure 5: **Learned Cameras.** We show the prior, ground truth (if existing), and predicted camera elevation and rotation distributions. Although only training from raw image collections without any annotation, our camera generator learns to deform the prior to match the data distributions. Note that as the camera distributions are learned fully unsupervised, arbitrarily-shifted rotation distributions are equally-valid solutions, and here we therefore manually aligned them for this visualization.

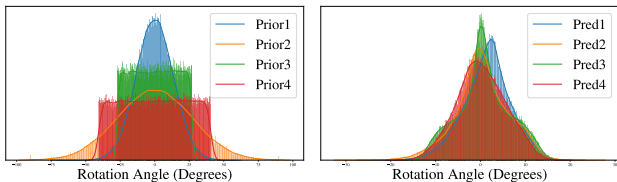


Figure 6: **Prior Sensitivity.** Different camera priors (left) result in similar predicted distributions (right) on CelebA.

appearance of the foreground object. Our model disentangles shape and appearance as well as fore- from background (see Fig. 7) and learns correct 3D representations for them (see Fig. 8). At test time, these factors can explicitly be controlled, facilitating controllable image synthesis. Although not being the primary goal, we further find that using separate fore- and background models also improves results quantitatively (see Tab. 2).

How important is our training regime? In Tab. 2, we compare our progressive growing regime against the patch-based training from [59]. We find that our training regime leads to better quantitative results. Qualitatively, we observe that fore- and background disentanglement is less consistent and the camera generator less stable for patch-based training due to the reduced receptive field.

4.2. Limitations and Future Work

Image Quality vs. Camera Exploration: We tackle the problem of learning a generative model of 3D representations solely from raw, unposed image collection using an adversarial loss. Note that the discriminator loss is based on 2D renderings of our model, and hence 3D consistency is purely a result of the incorporated bias. We find that at later stages of training on CelebA, our model tends to reduce the camera pose range in favor of only increasing image quality. We ascribe this to the large data complexity and resulting training dynamics at higher resolutions. In practice, we avoid this tradeoff by keeping the camera generator fixed for later stages of training on CelebA. We identify exploring how the model can be encouraged to explore the largest possible camera ranges as promising future work.

Multi-View vs. 3D Consistency: Similar to [5, 59], we find that our model sometimes generates 3D representations which are multi-view consistent from the learned pose ranges, but not as expected, e.g., we observe “inverted faces” which is also known as hollow face illusion (see Fig. 9). We plan to investigate how stronger 3D shape biases can be incorporated into the generator model.

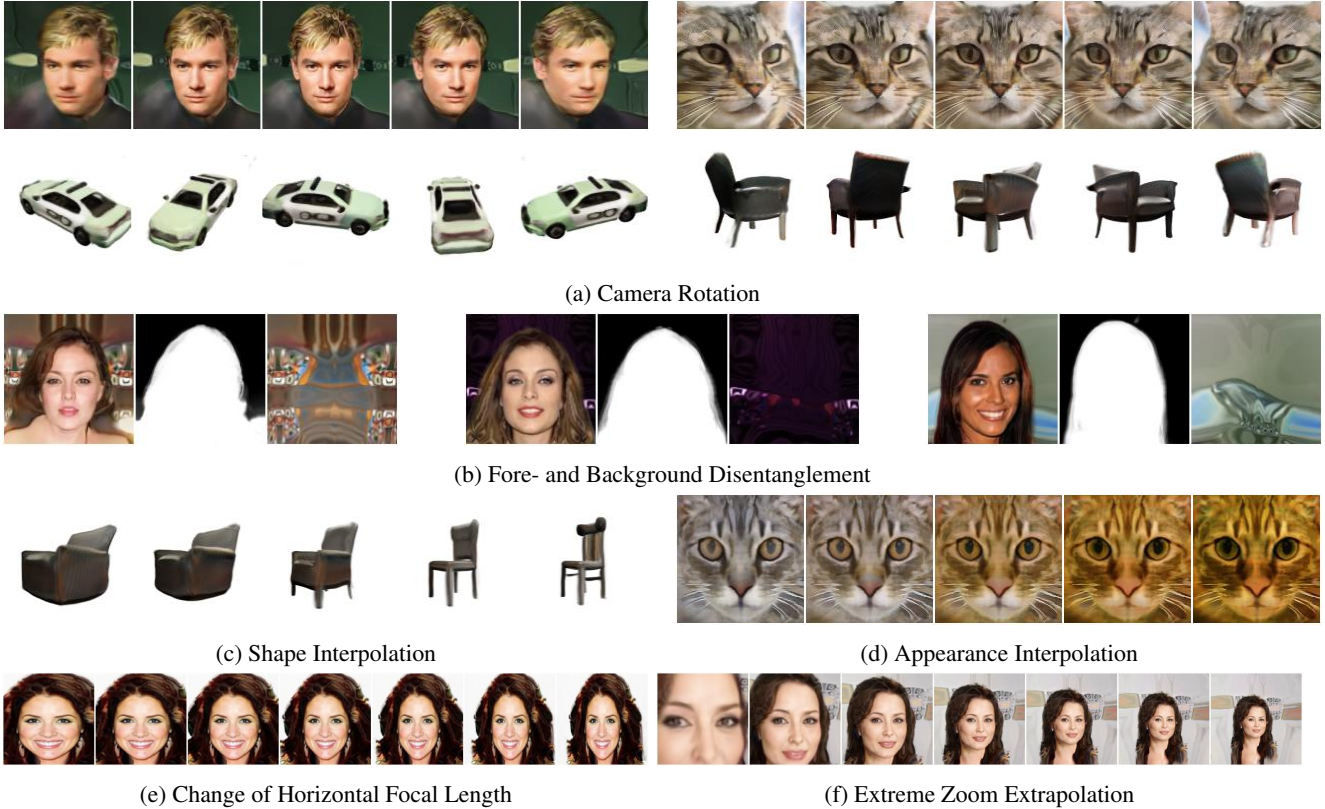


Figure 7: **Controllable Image Synthesis.** While training from raw image collections, we learn a 3D- and camera-aware generative model which allows for controllable image synthesis. We can control the camera viewpoint (7a), disentangle fore- and background (7b), i.e. we only render the fore- and background model, respectively, manipulate the shape (7c) and appearance (7d), change the focal length (7e), and perform extreme zoom extrapolation (7f).

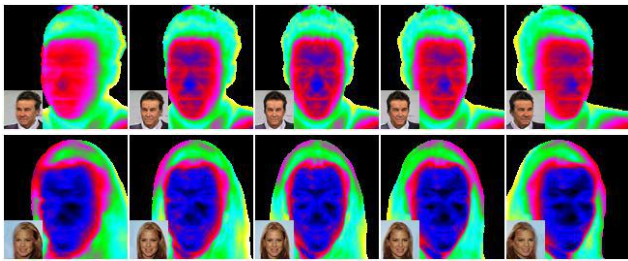


Figure 8: **Foreground Depth.** We show RGB (bottom left) and foreground depth. While training from raw unposed images, our model learns to disentangle fore- from background and associates correct 3D representations for them.

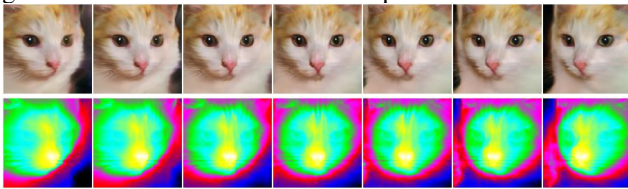


Figure 9: **Failure Analysis.** We sometimes observe inward-facing faces. We identify incorporating stronger 3D shape biases into the model as promising future work.

5. Conclusion

In this work we present *CAMPARI*, a novel method for 3D- and camera-aware image synthesis. Our key idea is to learn a camera generator jointly with a 3D-aware image generator. Further, we decompose the scene into fore- and background leading to more efficient scene representations. While training from raw, unstructured image collections, our method faithfully recovers the camera distribution and at test time, we can generate novel scenes with explicit control over the camera viewpoint as well as the shape and appearance of the scene.

Acknowledgment

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] Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzhnikov. Image generators with conditionally-independent pixel synthesis. *arXiv.org*, 2011.13775, 2020. 2
- [4] 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
- [5] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. *arXiv.org*, 2012.00926, 2020. 1, 2, 4, 5, 6, 7
- [6] Ke Chen, Ryan Oldja, Nikolai Smolyanskiy, Stan Birchfield, Alexander Popov, David Wehr, Ibrahim Eden, and Joachim Peherl. MvliDAMet: Real-time multi-class scene understanding for autonomous driving using multiple views. *arXiv.org*, 2006.05518, 2020. 2
- [7] Xi Chen, Xi Chen, Yan Duan, Rein Houthoofd, 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
- [8] Zhiqin Chen, Kangxue Yin, Matthew Fisher, Siddhartha Chaudhuri, and Hao Zhang. BAE-NET: branched auto-encoder for shape co-segmentation. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2
- [9] 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
- [10] 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
- [11] 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
- [12] Edo Collins, Raja Bala, Bob Price, and Sabine Süssstrunk. Editing in style: Uncovering the local semantics of gans. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2
- [13] 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. 4
- [14] Matheus Gadelha, Subhansu 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
- [15] 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
- [16] 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
- [17] 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
- [18] Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent independent mechanisms. *arXiv.org*, 1909.10893, 2019. 1
- [19] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable GAN controls. *arXiv.org*, 2004.02546, 2020. 2
- [20] 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. 4, 5
- [21] 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
- [22] Paul Henderson and Christoph H. Lampert. Unsupervised object-centric video generation and decomposition in 3d. *arXiv.org*, 2007.06705, 2020. 2
- [23] 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
- [24] 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. 1, 2, 4
- [25] Ali Jahani, Lucy Chai, and Phillip Isola. On the “steerability” of generative adversarial networks. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2020. 2
- [26] James T. Kajiya and Brian Von Herzen. Ray tracing volume densities. In *ACM Trans. on Graphics*, 1984. 3
- [27] 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
- [28] 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

- [29] 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
- [30] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *Proc. of the International Conf. on Learning Representations (ICLR)*, 2014. 1
- [31] Hanock Kwak and Byoung-Tak Zhang. Generating images part by part with composite generative adversarial networks. *arXiv.org*, 1607.05387, 2016. 1
- [32] Wonkwang Lee, Donggyun Kim, Seunghoon Hong, and Honglak Lee. High-fidelity synthesis with disentangled representation. *arXiv.org*, 2001.04296, 2020. 1
- [33] Jake Levinson, Carlos Esteves, Kefan Chen, Noah Snavely, Angjoo Kanazawa, Afshin Rostamizadeh, and Ameesh Makadia. An analysis of SVD for deep rotation estimation. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 4
- [34] 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. 1, 2
- [35] 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
- [36] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 5
- [37] 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
- [38] 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
- [39] 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
- [40] 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
- [41] Quan Meng, Anpei Chen, Haimin Luo, Minye Wu, Hao Su, Lan Xu, Xuming He, and Jingyi Yu. Gnerf: Gan-based neural radiance field without posed camera. *arXiv.org*, 2103.15606, 2021. 2
- [42] 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
- [43] 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
- [44] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2
- [45] 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, 5
- [46] 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
- [47] 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. 1, 4, 5
- [48] 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. 1, 2, 4
- [49] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 4, 6
- [50] 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, 5
- [51] 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
- [52] 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
- [53] 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, 5
- [54] Keunhong Park, Konstantinos Rematas, Ali Farhadi, and Steven M. Seitz. Photoshape: Photorealistic materials for large-scale shape collections. *Communications of the ACM*, 2018. 5

- [55] 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
- [56] 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
- [57] 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
- [58] 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
- [59] 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. 1, 2, 3, 4, 5, 6, 7
- [60] 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
- [61] 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
- [62] Shih-Yang Su, Frank Yu, Michael Zollhoefer, and Helge Rhodin. A-NeRF: Surface-free human 3d pose refinement via neural rendering. *arXiv.org*, 2102.06199, 2021. 2
- [63] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan 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
- [64] 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
- [65] Shaofei Wang, Marko Mihajlovic, Qianli Ma, Andreas Geiger, and Siyu Tang. MetaAvatar: Learning animatable clothed human models from few depth images. *arXiv.org*, 2004.05980, 2021. 2
- [66] 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
- [67] 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
- [68] 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
- [69] 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
- [70] Lin Yen-Chen, Pete Florence, Jonathan T. Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. iNeRF: Inverting neural radiance fields for pose estimation. *arXiv.org*, 2012.05877, 2020. 2
- [71] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv.org*, 2010.07492, 2020. 3, 4
- [72] Li Zhang, Brian Curless, Aaron Hertzmann, and Steven M. Seitz. Shape and motion under varying illumination: Unifying structure from motion, photometric stereo, and multi-view stereo. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2003. 5
- [73] 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
- [74] 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
- [75] 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
- [76] 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