GIRAFFE HD: A High-Resolution 3D-aware Generative Model

Yang Xue 1  Yuheng Li 2  Krishna Kumar Singh 3  Yong Jae Lee 2
1 UC Davis  2 UW–Madison  3 Adobe Research

Figure 1. Our model, GIRAFFE HD, inherits all of GIRAFFE’s [38] 3D controllability—change in camera viewpoint, object translation, scale, rotation, appearance, shape, and background—while generating higher quality, higher resolution images. Moreover, it achieves better foreground-background disentanglement; e.g., when changing the car’s shape (fourth and fifth columns), notice how parts of the road and building in the background change in the GIRAFFE images, whereas they remain constant in ours.

Abstract

3D-aware generative models have shown that the introduction of 3D information can lead to more controllable image generation. In particular, the current state-of-the-art model GIRAFFE [38] can control each object’s rotation, translation, scale, and scene camera pose without corresponding supervision. However, GIRAFFE only operates well when the image resolution is low. We propose GIRAFFE HD, a high-resolution 3D-aware generative model that inherits all of GIRAFFE’s controllable features while generating high-quality, high-resolution images (512^2 resolution and above). The key idea is to leverage a style-based neural renderer, and to independently generate the foreground and background to force their disentanglement while imposing consistency constraints to stitch them together to composite a coherent final image. We demonstrate state-of-the-art 3D controllable high-resolution image generation on multiple natural image datasets.

1. Introduction

In image generation, two of the most important objectives are image realism and controllability. Style-based GANs (i.e., StyleGAN [25] and its variants [24, 27]) can generate high-resolution, photorealistic images. However, while their latent style code design provides a level of disentanglement and controllability in 2D space (e.g., color and shape changes), their lack of explicit 3D information makes it difficult to impose 3D-level control over the generated image content. Meanwhile, the recent NeRF [35] based GANs [5, 38, 43] have shown that explicit modeling of the scene in 3D space conditioned on camera pose can enable effective 3D-level control. However, the computationally expensive nature of 3D representations has limited current 3D-aware generative models from directly learning and rendering images in high resolutions.

GIRAFFE [38] is the current state-of-the-art 3D-aware generative model for both image realism and controllability. It models the foreground and background as two separate 3D objects, uses volume rendering to render the combined 3D features into low-resolution 2D feature maps, and finally uses a neural renderer to further render the feature maps into higher resolution images. These design choices enable GIRAFFE to change the background’s appearance independent of the foreground, translate or rotate the foreground object in 3D, and change the foreground object’s shape and color. However, the neural renderer is specifically designed to provide only spatially small refinements in order to avoid entangling global scene properties and losing controllability. Thus, it is significantly less powerful than style-based renderers, and hence the highest image resolution that GIRAFFE can generate is 256^2.
In this work, we propose a two-stage style-based 3D-aware generative model that inherits all of GIRAFFE’s controllability while generating high-quality, high-resolution images (up to 1024² resolution); see Fig. 1. Our design is motivated by three key observations when replacing GIRAFFE’s neural renderer with a style-based neural render (based on StyleGAN2 [27]): 1) Using the style renderer to upsample the volume-rendered low-res 2D feature maps leads to high-quality, high-resolution image generations while still preserving controllability over the foreground object’s 3D properties (translation, rotation). However, due to its high capacity, the style renderer 2) now gains full control over color as well as some control over shape, and 3) entangles and loses controllability over the foreground and background features (i.e., changing the foreground color/shape also changes the background color/shape).

In order to regain controllability over the foreground and background, we generate them independently using two different style-based renderers and combine them into a coherent image by imposing geometric and photometric compatibility constraints that eliminate inconceivable combinations. Furthermore, to disentangle color and shape, we exploit the well-known emergent properties of StyleGAN, namely that the early layers control coarse shape, mid layers control fine-grained shape, and later layers control color. Specifically, we inject the shape code into the 3D feature generator as well as the early layers of the style renderer to control shape, and the appearance code into the later layers of the style renderer to control color.

Contributions. Our approach, GIRAFFE HD, preserves the 3D controllability of GIRAFFE, including independent control over foreground and background, while generating much higher-resolution and higher-quality images (up to 1024² vs. GIRAFFE’s 256²). We validate our approach on multiple natural image datasets (CompCar [51], FFHQ [26], AFHQ Cat [12], CelebA-HQ [23], LSUN Church [52]) and demonstrate better foreground-background disentanglement and image realism compared to GIRAFFE in higher resolution domains. Finally, we perform ablation studies to justify the different design choices for our model.

2. Related Work

3D-aware image synthesis. In recent years, using implicit neural representations to represent 3D scenes and volume render into 2D images has shown great potential [9–11, 14, 22, 34, 42, 46]. For example, NeRF [35] can effectively learn the 3D geometry using multiple images of a scene from different viewpoints and generate new images from new viewpoints. NeRF-based 3D-aware GANs [5, 6, 38, 43] condition the neural representations on sample noise or appearance/shape codes to represent different 3D scenes with a single network. This improvement also enables these models to be trained on unstructured image collections, as opposed to images from a single scene. Our GIRAFFE HD builds upon this line of work; in particular, it extends GIRAFFE [38] to higher-resolution image domains while retaining all its 3D understanding capabilities.

High-resolution image synthesis. Generative Adversarial Networks (GANs) [4, 15, 41] can generate photorealistic images, and the state-of-the-art for high-resolution image synthesis are style-based GANs [24, 25, 27]. By injecting style codes [21] to the network, these models achieve not only high-resolution outputs but also some level of feature disentanglement (e.g., pose, shape, lighting). Some recent work [29, 30] have demonstrated that using StyleGAN2 as a neural renderer can effectively upsample the low-resolution feature maps produced by another network into high-resolution images. We leverage the StyleGAN2 architecture as our model’s neural renderer to generate high-resolution 2D images (512² resolution and higher).

Disentanglement and controllability. Generative models that learn disentangled representations [2, 8, 13, 18–20, 25, 28, 32, 40] provide extra control in their generations, for example, the ability to control different factors in the scene (e.g., object pose, shape, appearance). However, most methods only operate in the 2D domain without considering the 3D structure of the objects/scenes.

Among generative models that learn 3D disentanglement [16, 36–38, 44], GIRAFFE [38] is the current state-of-the-art. It represents a 3D scene as a composition of foreground and background 3D objects, which enables it to disentangle object shape, appearance, position, camera viewpoint, as well as the foreground and background during image synthesis. However, we observe that this disentanglement comes at a trade-off to image quality – replacing GIRAFFE’s low capacity neural renderer with a style-based renderer leads to high-resolution synthesis but at the loss of foreground-background disentanglement. Several supervised methods exploit StyleGAN’s style-based disentanglement properties to control the generation process [1, 39, 49]. However, they have yet to demonstrate accurate foreground-background disentanglement, which suggests that the vanilla StyleGAN architecture has a limitation when it comes to foreground-background disentanglement. The most intuitive and reliable foreground-background disentanglement methods are two-stage image generators, which generate the foreground and background independently and use 2D composition to form the final output image [3, 31, 45, 50]. However, these methods fall short in terms of image quality, presumably due to a lack of explicit foreground and background information sharing and compatibility constraints. Our approach also generates the foreground and background in two separate stages, but it imposes explicit geometric and photometric compatibility...
constraints. This leads to accurate foreground-background disentanglement in the high-resolution image domain.

3. Approach
Given an image collection containing a single object category (e.g., cars), our goal is to learn a 3D-aware image generation model that generates photo-realistic, high-resolution images while also providing 3D-level control without human supervision. To this end, our architecture builds upon GIRAFFE [38], but replaces its low-capacity neural render with a StyleGAN2 [27] based renderer, and has two separate parallel streams to generate separate foreground and background images to enforce their disentanglement. We create the final output by combining the foreground and background while imposing compatibility constraints to ensure a coherent image; see Fig. 2.

3.1. Background on GIRAFFE

Generative neural feature fields. GIRAFFE [38] represents a 3D scene with a neural feature field [35, 38], which is a continuous function \( F \) that maps a 3D location \( x \in \mathbb{R}^3 \) and a 2D camera viewing direction \( d \in \mathbb{S}^2 \) to a density \( \sigma \in \mathbb{R}^+ \) and an appearance feature \( f \in \mathbb{R}^{M_f} \). It uses an MLP to learn \( F \), and conditions it on \( z \sim \mathcal{N}(0, I) \) so that each \( z \) corresponds to a different 3D scene:

\[
F_\theta : (\gamma^{L_c}(x), \gamma^{L_d}(d), z) \rightarrow (\sigma, f)
\]  

(1)

where \( \theta \) indicates the network parameters, \( \gamma \) is a positional encoding [47] which maps the 5D input \( (x, d) \) into a higher dimensional space, and \( L_c \) and \( L_d \) are the positional encoding dimensions of \( x \) and \( d \), respectively.

3D object representation. GIRAFFE represents the foreground and background objects using two separate MLPs associated with separate affine transformations \( T^{fg} \) and \( T^{bg} \), respectively, sampled from a dataset-dependent distribution \( T = \{s, t, R\} \), where \( s, t \in \mathbb{R}^3 \) are scale and translation parameters, and \( R \in SO(3) \) is a rotation matrix. The affine transformation \( T \) transforms the scene’s world location to the object’s local location for each object:

\[
k(x) = R \cdot sE \cdot x + t
\]  

(2)

where \( E \) is the \( 3 \times 3 \) identity matrix. This representation enables 3D object-level control.

Volume rendering. For a given camera pose \( \xi \), let \( \{x_j\}_{j=1}^{N_s} \) be \( N_s \) sample points along camera ray \( d \) for a given pixel. Then

\[
(\sigma_j, f_j) = F_\theta(\gamma^{L_c}(k^{-1}(x_j)), \gamma^{L_d}(d), z).
\]  

(3)

Let \( \delta_j = \|x_{j+1} - x_j\|^2 \) denote the distance between neighboring sampled points, \( \alpha_j = 1 - e^{-\sigma_j \delta_j} \) denote the alpha value for \( x_j \), and \( \tau_j = \prod_{i=1}^{j-1} 1 - \alpha_j \) denote the transmittance along the ray. Pixel feature vector \( f \) can then be computed using numerical integration:

\[
f = \sum_{j=1}^{N_s} \tau_j \alpha_j f_j
\]  

(4)

For efficiency, the rendered feature images are at \( 16^2 \) resolution. The volume-rendered feature map \( f_{\text{vol}} \) can then be processed by a neural renderer (i.e., a convnet) to output the final RGB image.

Note that in GIRAFFE, the foreground and background’s 3D object representations are composed into a single 3D

![Figure 2. GIRAFFE HD Architecture.](image-url)
scene representation and volume-rendered into a single 2D feature representation. However, in our approach, we will independently volume render the foreground and background's 3D representations, as explained in detail next.

3.2. Neural style rendering

In GIRAFFE, the neural renderer is purposely designed to be simple and provide only spatially small refinements to the volume-rendered feature maps, in order to avoid entangling global scene properties and losing controllability. With its default renderer, the highest resolution it can generate is $256^2$.

In order to generate higher-resolution ($\geq 512^2$) outputs, we first replace GIRAFFE’s default neural renderer with one based on StyleGAN2 [27]. Specifically, we take all the blocks of StyleGAN2 starting from $16^2$ resolution to convert the volume-rendered $16^2$ resolution 2D feature maps $f_{vol}$ into a higher resolution image $I$. As in StyleGAN2, we also use a mapping network to map $z \sim \mathcal{N}(0, I)$ to latent code $w$:

$$
\psi_{\theta}(z) \rightarrow w \quad (5)
$$

$$
\pi^r_{\theta}(I_{vol}, w) \rightarrow I \quad (6)
$$

While our style renderer leads to higher-resolution outputs, we observe several behavioral distinctions compared to GIRAFFE’s default renderer. First, the model now loses its ability to independently control the foreground and background. Second, the 3D representation no longer fully controls the object’s shape. Though it still determines overall coarse shape, the earlier stages of the style renderer gain finer level control over the shapes since the 3D representation is volume-rendered to 2D feature maps at a much lower resolution than the final image. Third, the 3D representation almost does not control color at all. Instead, the control of color is transferred to the later stages of the style renderer. These behaviors resemble those of vanilla StyleGAN2.

To regain independent control over foreground and background, and to better disentangle object color and shape, we make the following design choices. First, instead of compositing the scene at the 3D level and then rendering it into a single final 2D image, we first render the foreground and background independently into two 2D images and then perform 2D composition to get the final image. Second, unlike GIRAFFE, which conditions the 3D representation on the object’s shape code $z_a$ as well as its appearance code $z_a$, we remove the dependency of each point’s feature $f$ on $z_a$. Instead, during training, we perform style mixing in the style renderer (as described in StyleGAN2) with $w_a = \psi_{\theta}(z)$ and $w_a = \psi_{\theta}(z_a)$. During evaluation, we inject $w_a$ into the earlier stages and $w_a$ into the later stages of the style renderer (we vary the injection index depending on final image resolution). This way of injecting codes enables our model to disentangle color and shape finely.

3.3. Enforcing foreground-background consistency

In order to combine the separately generated foreground and background images into a coherent final image, we need to impose geometric and photometric consistency between the foreground and background objects. Geometric consistency requires the foreground and background objects to obey physical world rules; for example, objects in the same image have to share the same viewing perspective, or a car cannot be floating in the air. Photometric consistency requires the foreground and background objects to appear to reside in the same environment by sharing the same lighting, hue or saturation, etc. To this end, we devise two mechanisms to satisfy the two consistency requirements: position sharing and environment sharing.

Position sharing. We activate position sharing when the foreground object’s $z$-translation and view perspective to the background object’s $z$-translation and view perspective. In this way, the generated background can actively accommodate all foreground objects.

For datasets where the foreground object is not expected to rest on a ground surface (e.g., frontal human faces), we only synchronize the viewing perspective without sharing the $z$-translation between foreground and background.

Environment sharing. Besides enforcing geometric consistency between the foreground and background objects, we also need to ensure photometric consistency; i.e., the foreground object should naturally be immersed into the environment created by the background. To this end, we designate the background appearance latent code $w_{bg}^a$ to encode the scene environment configuration. Our refinement network consists of several layers of style-based convolutions. It takes as input the foreground feature maps $I_{fg}^{\text{out}}$, which are also used to render the initial foreground image, and $w_{bg}^a$ as the style code, and outputs the foreground image residual $I_{fg}^{\text{residual}}$ and foreground object mask $I_{mask}$:

$$
\pi^r_{\theta}(I_{fg}^{\text{out}}, w_{bg}^a) \rightarrow (I_{fg}^{\text{residual}}, I_{mask}) \quad (7)
$$

We add the foreground image residual to the initial foreground image to get the final foreground image:

$$
I_{fg} = I_{fg}^{\text{initial}} + I_{fg}^{\text{residual}} \quad (8)
$$

We observe that the initial foreground image already determines the foreground object’s true appearance. The re-
finement operation only adjusts the shading/shine of the foreground without altering its true appearance; see Fig. 3.

### 3.4. Compositing the final image

Finally, we use the foreground object mask \( I_{mask} \) generated by the refinement network to perform alpha composition of the foreground and background images:

\[
I_{final} = (1 - I_{mask}) \cdot I^{bg} + I_{mask} \cdot I^{fg} \tag{9}
\]

where \( I_{final} \) is our model’s final generated image.

Like GIRAFFE, we can generalize our model to generate multiple foreground objects. To do this, we first render the background and the foregrounds as described previously. We then compute occlusion relations between the foreground objects (by ordering based on their depth i.e., \( x \)-translation). Finally, we recursively perform 2D composition (Eqn. 9) from the furthest to the nearest foreground object, where in each recursive iteration, the 2D composition result becomes the new background image.

### 3.5. Training

**Discriminator.** We use the same residual discriminator as StyleGAN2 [27].

**Training.** During training, we follow [38] and sample latent codes \( z^k \sim \mathcal{N}, T^k \sim p_T \) and \( \xi \sim p_{\xi} \), where \( k \in \{fg, bg\} \), \( p_T \) and \( p_{\xi} \) are uniform distributions over dataset-dependent camera elevation angles and valid object transformations, respectively.

**Objectives.** Our overall objective function is:

\[
L = L_{GAN} + \frac{\lambda}{2} L_{R1} + \beta_1 L_{bbox} + \beta_2 L_{cvg} + \beta_3 L_{bin} \tag{10}
\]

where \( \lambda = 10 \), and \( \beta_1, \beta_2, \beta_3 \) are dataset specific. To enforce image realism, we use the non-saturating GAN objective \( L_{GAN} \) with R1 regularization \( L_{R1} \) [33].

In addition, we employ three auxiliary losses to guide 2D foreground-background disentanglement: bounding box containment loss \( L_{bbox} \), foreground coverage loss \( L_{cvg} \), and mask binarization loss \( L_{bin} \). \( L_{cvg} \) and \( L_{bbox} \) are adapted from [3]. Since the style neural renderer is very powerful on its own, the three auxiliary losses are necessary to prevent either the foreground or background renderer from generating the entire image by itself.

See supp. for the full expression of the loss functions, including how the sampled appearance, shape, and camera/transformation latent codes are used.

**Bounding box containment loss.** Each randomly sampled foreground affine transformation \( T \) determines a 3D bounding box within which the foreground object should reside. After projecting both the 3D foreground object and the 3D bounding box to 2D, the 2D foreground object should still reside in the 2D bounding box. Our bounding box containment loss minimizes the mean of the foreground object mask values that fall outside the 2D bounding box:

\[
L_{bbox} = \frac{1}{|S|} \sum_{i \in S} I_{mask}[i] \cdot (1 - I_{2Dbbox}[i]) \tag{11}
\]

where \( S \) is the set of all pixels in final image. This loss prevents the foreground renderer from generating background features.

**Foreground coverage loss.** This is a hinge loss on the mean mask value to ensure that the foreground is not empty:

\[
L_{cvg} = \max(0, \eta - \frac{1}{|S|} \sum_{i \in S} I_{mask}[i]) \tag{12}
\]

where \( \eta \) is the minimum coverage threshold. This prevents the background renderer from generating the entire image.

**Mask binarization loss.** This loss encourages binarization (i.e., 0 or 1 values) of the mask:

\[
L_{bin} = \frac{1}{|S|} \sum_{i \in S} \min(I_{mask}[i] - 0, 1 - I_{mask}[i]). \tag{13}
\]

### 4. Experiments

We evaluate GIRAFFE HD’s 3D controllability, with a focus on foreground and background disentanglement and their geometric/photometric consistency. We also evaluate how well it generates high-quality, high-resolution images. Finally, we perform ablation studies to evaluate its different components and losses.

**Imp. details.** Foreground and background generative neural radiance fields are MLPs with ReLU activations. We use 8 layers with hidden dimension of 256/64 (foreground/background respectively), density of 1, and feature head of dimensionality \( M_f = 256 \) for the MLPs. We sample \( N_s = 64 \) points along each ray and render 2D feature maps at 16\(^2\) pixels. Both foreground and background’s shape and appearance codes are 256 dimensions. We use 4
Figure 4. **3D Controllability.** GIRAFFE HD preserves all of GIRAFFE’s 3D controllable features while generating images at significantly higher quality and resolution. Importantly, compared to GIRAFFE, our background remains more consistent when the foreground changes.

layer MLPs to map noise vectors to style renderers’ latent codes. Refine renderer has 4 style-based convolution layers. We use a minimum coverage threshold of 0.2. Foreground and background images go through a tanh activation before 2D composition. Foreground mask goes through a sigmoid activation before 2D composition. We use Adam optimizer with a learning rate of 0.0005 and batch size of 16.

**Baselines.** We compare to GIRAFFE [38], pi-GAN [6], GRAF [44], HoloGAN [36], and HoloGAN w/o 3D Conv, a HoloGAN variant proposed in [44] for higher resolutions.

**Datasets.** We evaluate on five high-resolution single-object real-world datasets used in GIRAFFE [38]: CompCar [51], FFHQ [26], AFHQ Cat [12], CelebA-HQ [23], LSUN Church [52].

**Metrics.** We use FID [17] to quantify image quality. We use 20,000 real and fake samples to calculate the FID score in order for a direct comparison to [38].

To quantify foreground-background disentanglement, we propose the mutual background similarity (MBS) metric. It measures the consistency in background between
Table 1. 2562 Resolution Image Quality. We report the FID score (↓) for all methods. † scores (except Cat) taken from [38].

<table>
<thead>
<tr>
<th>Method</th>
<th>Cat CelebA-HQ FFHQ CompCar Church</th>
</tr>
</thead>
<tbody>
<tr>
<td>HoloGAN [36]†</td>
<td>- 61 192 34 58</td>
</tr>
<tr>
<td>w/o 3D Conv [44]†</td>
<td>- 33 70 49 66</td>
</tr>
<tr>
<td>GRAF [44]†</td>
<td>- 49 59 95 87</td>
</tr>
<tr>
<td>GIRAFFE [38]†</td>
<td>33.39 21 32 26 30</td>
</tr>
<tr>
<td>pi-GAN [6]</td>
<td>38.92 36.27 43.19 64.01 56.80</td>
</tr>
<tr>
<td>Ours</td>
<td>12.36 8.09 11.93 7.22 10.28</td>
</tr>
</tbody>
</table>

Table 2. 5122 and 10242 Resolution Image Quality. We report the FID score (↓) for GIRAFFE and GIRAFFE HD.

<table>
<thead>
<tr>
<th>Method</th>
<th>CompCar 5122 FFHQ 10242</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIRAFFE [38]</td>
<td>40.81 70.08</td>
</tr>
<tr>
<td>Ours</td>
<td>8.36 10.13</td>
</tr>
</tbody>
</table>

Table 3. Foreground-Background Disentanglement. We report the MBS score (↓) for all methods on FFHQ and CompCar at 2562.

<table>
<thead>
<tr>
<th>Method</th>
<th>FFHQ</th>
<th>CompCar</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIRAFFE [38]</td>
<td>99.15</td>
<td>88.89</td>
</tr>
<tr>
<td>Ours</td>
<td>15.02</td>
<td>22.88</td>
</tr>
</tbody>
</table>

Table 4. Ablation: Removing Position/Environment Sharing. ‘full’ denotes the full GIRAFFE HD model. We report FID (↓).

<table>
<thead>
<tr>
<th>Method</th>
<th>FFHQ</th>
<th>CompCar</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o position sharing</td>
<td>10.89</td>
<td>11.55</td>
</tr>
<tr>
<td>w/o environment sharing</td>
<td>11.55</td>
<td>11.55</td>
</tr>
<tr>
<td>full</td>
<td>7.22</td>
<td>7.22</td>
</tr>
</tbody>
</table>

Figure 5. GIRAFFE [38] image generations. We show these to enable direct comparison. (More comparisons in the supp.)

4.1. Image generation quality

We first evaluate the quality of GIRAFFE HD’s generated images. Since it is demonstrated in [38] that GIRAFFE can reliably operate at 2562 resolution, we start our comparison at 2562 against all baselines (Table 1). We then compare to GIRAFFE at the higher resolutions of 5122 for CompCar dataset and 10242 for FFHQ dataset (Table 2). Our method outperforms the baselines in terms of image quality by a large margin. This can be attributed largely to our style-based neural renderer, which is able to model finer details than GIRAFFE’s low-capacity neural renderer.

4.2. Controllable scene generation

In Fig. 4, we qualitatively demonstrate that our method preserves all of GIRAFFE’s controllable features. For vertical translation, note how our position sharing enables the ground to move up with the car. Also, compared to GIRAFFE, our background remains more consistent when the foreground changes, as shown in our lower MBS in Table 3 (qualitative comparisons are in Figs. 1 and 5 and supp). This is due to our explicit separation of foreground and background generation. For objects that rest on a ground surface (e.g., cars), our model will also include the object’s shadow as part of the foreground (see Fig. 8 for examples), which is the ideal behavior. However, the DeepLabV3 model [7] that is used to compute MBS does not segment the shadow as being part of the object, which is why our MBS is higher on CompCar than on FFHQ. In Fig. 8, we show comprehensive intermediate and final image generations.

4.3. Ablation studies

Importance of two-stage. The most naive method for improving GIRAFFE’s image quality is to simply replace GIRAFFE’s neural renderer with a single style-based renderer. For this baseline, during training, we use style mixing of zfg and zfg as latent codes to the renderer. During evaluation, we still inject zfg into the earlier levels and zf into the later levels to ensure disentanglement between fine-grained shape and appearance. However, we observe that this single renderer baseline loses foreground-background disentanglement (Fig. 6). Even though the car’s...
Figure 7. Ablation: Removing Auxiliary Losses. Even after just 4000 training iterations, the baselines that lack bounding box containment or foreground coverage loss generate all 1 or 0 masks.

color remains the same when changing its shape, the background’s shape changes as well. Similarly, even though the car’s shape remains the same when changing its color, the background’s color also changes. Even though the foreground and background are disentangled at the 3D feature level, since a single style-based renderer cannot separately control foreground and background, the foreground-background disentanglement is lost in the final 2D image.

Importance of foreground-background consistency enforcement. Table 4 shows that removing either position or environment sharing hurts the model’s FID, as final images whose foreground-background combinations are incompatible in geometry/photometry can be generated.

Importance of auxiliary losses. In Fig. 7, we show our model’s 642 renderings on FFHQ after 4000 training iterations, in three configurations. Without the bounding box containment loss the foreground branch generates the entire image, and without the foreground coverage loss the background generates the entire image. Hence these two losses are critical for foreground-background disentanglement.

5. Discussion and Conclusion

We proposed GIRAFFE HD, a high-resolution 3D-aware generative model that inherits all of GIRAFFE’s [38] 3D controllable features while generating high-quality, high-resolution images.

Limitations. First, we notice our model sometimes lacks 3D consistency. For example, when trained from scratch on the CompCar dataset, our model struggles to perform full 360 rotation. Instead, some shape codes correspond to front facing cars while others correspond to back facing cars, and each can only perform 180 rotation, even though the underlying 3D model has rotated 360 degrees. However, when we initialize the 3D feature generator with the weights of a pretrained GIRAFFE 3D feature generator and continue training, the model is then able to perform full 360 rotation.

Second, our current model (as well as GIRAFFE) cannot handle cross-domain foreground-background correlations when training a single model on multiple categories (e.g., cats, dogs, wildlife). Although the generated images are still high quality, there can be incompatible foreground and background combinations. Training a single model that works well on multiple datasets would be an interesting avenue for future work.

Third, our current architecture cannot handle “interlocking” object relationships, i.e., there exist some rays along which an object appears in front of another object and also some rays behind. Our model works by composing an image with 2D layers, and the layer masks (object masks) for the composition are generated by considering only the object itself and not other objects in the scene. This can generally hold for most real-world domains. However, when objects “interlock”, the objects’ masks need to take into account other objects’ 3D geometries. We think that a module that renders object masks based on 3D occlusion reasoning could be a possible extension to address this problem.

Broader Impact. There are many possible applications for controllable image generation including those in the entertainment and design industry. For example, it could enhance the productivity of designers by allowing them to use such tools to control each object in the scene independently when creating new visual content. Since our approach does not require supervision apart from having a collection of images of the same object, it is easily scalable to many different categories. However, there could also be potential misuses, such as creating fake content to fool law enforcement or to spread misinformation on social media. Recent work on models that can detect fake images (e.g., [48]) could potentially be useful to prevent such unethical applications.

Acknowledgements. This work was supported in part by a Sony Focused Research Award and NSF CAREER IIS-2150012. We thank the anonymous reviewers for their constructive comments.
References


