BallGAN: 3D-aware Image Synthesis with a Spherical Background

Minjung Shin1* Yunji Seo1 Jeongmin Bae1 Young Sun Choi1 Hyunsu Kim2 Hyeran Byun1 Youngjung Uh1†
Yonsei University1 NAVER AI Lab2

Abstract

3D-aware GANs aim to synthesize realistic 3D scenes that can be rendered in arbitrary camera viewpoints, generating high-quality images with well-defined geometry. As 3D content creation becomes more popular, the ability to generate foreground objects separately from the background has become a crucial property. Existing methods have been developed regarding overall image quality, but they can not generate foreground objects only and often show degraded 3D geometry. In this work, we propose to represent the background as a spherical surface for multiple reasons inspired by computer graphics. Our method naturally provides foreground-only 3D synthesis facilitating easier 3D content creation. Furthermore, it improves the foreground geometry of 3D-aware GANs and the training stability on datasets with complex backgrounds. Project page: https://minjung-s.github.io/ballgan/

*Part of the work was done during an internship at NAVER AI Lab.
†Corresponding author

1. Introduction

Traditional generative adversarial networks (GANs) synthesize realistic images. Although they provide some control over the camera poses [36, 37, 15, 38], they lack explicit 3D understanding of the scenes. Recently, 3D-aware GANs [27, 6, 35, 53] reformulate the generative procedure as modeling the potential 3D scenes and rendering them to images. The state-of-the-art 3D-aware GANs [5, 14, 47] rely on neural radiance fields or their variants to represent 3D scenes. Note that they can generate 3D scenes even without 3D supervision or multi-view supervision, rendering realistic images across different viewpoints. Although the quality of images generated by 3D-aware GANs continues to improve, their practical usage has been less explored.

Solely generating foreground objects is an important element for the practical use of generative models, especially for content creation. In this context, the diffusion-based methods have grown popular for 3D object synthesis despite their lack of realism [18, 32, 24, 39, 44]. Some
2D GANs model their output images as a combination of foreground and background, replacing the need for laborious post-processing [1, 4, 54]. On the other hand, few 3D-aware GANs inadequately separate the background and suffer from broken 3D shapes [47] or training instability [14]. Objects generated by EG3D [5] are connected to unrealistic walls as shown in Figure 2.

Learning to synthesize 3D foreground objects using a single-view dataset is challenging because it lacks both depth and separation supervision.

To solve this problem, we are inspired by a popular approach for video games or movies in the graphics community: representing salient objects with detailed 3D models and approximating peripheral scenery with simple surfaces (Figure 1a) to reduce the overall complexity. Despite approximating the 3D space to 2D, the rendered image achieves a realistic appearance. We expect the 3D-aware generators with a similar approach to achieve both separation and physically reasonable foreground geometry.

Accordingly, we propose our novel 3D-aware GAN framework, named BallGAN. It approximates the background as a 2D opaque surface of a sphere and employs conventional 3D features as the foreground. It accompanies a modified volume rendering equation for the opaque background. In addition, we introduce regularizers for clear foreground geometry and separation.

We demonstrate the strength of our work as follows. By design, BallGAN provides clear foreground-background separation without extra supervision (Figure 1b). For content creation, it enables inserting generated 3D foregrounds in arbitrary viewpoints without post-processing (Figure 1c). Our background representation as a spherical surface is generally applicable to any generator architectures or foreground representations. BallGAN allows StyleNeRF [14] to be trained on a higher resolution of CompCars [48] and achieve a large FID boost, which is notable as the dataset is challenging due to its complex backgrounds. More importantly, BallGAN not only enhances multi-view consistency, pose accuracy, and depth reconstruction compared to EG3D, but it also faithfully captures fine details in 3D space that are easy to represent in 2D images but challenging to model in 3D.

2. Related work

Representations for 3D-aware GANs: Generators in 3D-aware GANs involve representing 3D scenes somehow and rendering them to 2D images so that the generator is aware of the 3D scene given only a collection of unstructured 2D images. HoloGAN [27] represents a scene with a 3D grid of voxels containing feature vectors, i.e., 4D tensor. However, as the 3D grid of voxels is limited by computational complexity, its maximum resolution is $128^2$.

Recent 3D-aware GANs integrate neural radiance fields (NeRFs) [26]. NeRF represents a 3D scene using a coordinate-based function that produces RGB color and density at that coordinates. This 3D scene can be projected onto a 2D image from arbitrary camera poses via volume rendering integral. GRAF [35] introduces a patch-based discriminator, which dramatically reduces memory usage in high-resolution 3D-aware image synthesis. Its successors improve image quality and 3D awareness by 1) enhancing the function for NeRF [6, 14], 2) volume rendering feature field followed by neural rendering with upsampling blocks [14, 29, 45, 5, 47], or 3) designing voxel-based [43, 12, 16, 28, 45] or hybrid [5] representations. Going further, our method introduces a separate NeRF for modeling spherical background, which encloses the foreground of EG3D [5] or StyleNeRF [14].

Scene decomposition: Some methods decompose the 3D scenes into multiple components. GIRAFFE and its variant [29, 47] separate scenes into objects and the background, enabling them to control objects independently with the background fixed. However, their background representation lives in the same ray points with the foregrounds, and the 3D geometry does not benefit from the separation. StyleNeRF [14] and EpiGRAF [40] separate the background outside a sphere following NeRF++ [50] where the background region goes through the same volume rendering with multiple ray points at variable depth. On the contrary, we remove the depth ambiguity of the background by modeling it with an opaque representation on a 2D spherical surface enclosing the foreground.

Reducing dimensions: has been a viable option for reducing space and time complexity. TensoRF [7] uses a sum of vector-matrix outer products to represent a 3D feature field.

Figure 2: Comparison of the 3D geometry extracted by marching cubes. (a) GIRAFFE-HD exhibits broken 3D shapes, (b) StyleNeRF has jaggy surfaces, and (c) EG3D has hair sticking to the wall. Unlike other models, (d) our model produces high-quality foreground geometry that is separated from the background.
EG3D [5] represents a 3D feature field with three 2D planes to adopt StyleGAN architecture. K-Planes [11] represents a d-dimensional scene using \((d/2)\) planes. While these methods decompose 3D feature fields into low-dimensional feature representations to reduce the memory usage of NeRFs, BallGAN squeezes the background space into a surface to provide an easier task for 3D-aware GANs.

3. BallGAN

In this section, we provide an overview of our framework and describe its key components and intuitions.

**Overview** We suppose that generating unbounded 3D scenes is too complex to learn relying on a limited guide for producing realistic 2D images. To resolve this challenge, BallGAN bounds the scene in a ball and approximates the background as an opaque spherical surface. We expect it to alleviate the burden of producing correct shapes of the backgrounds because the shape is fixed on a ball.

As shown in Figure 3, our generator consists of two backbone networks for foreground and background (§3.1). Representations from these networks are rendered by our modified volume rendering equation to synthesize images (§3.2) and trained with GAN objectives and auxiliary regularizations (§3.3).

3.1. Bounding the 3D space

While traditional 2D GANs learn to produce arrays of RGB pixels in fixed dimensions, 3D-aware GANs aim to produce realistic images by synthesizing 3D scenes and rendering them into 2D images. In contrast to training NeRFs with multi-view observations of a single scene, the only objective for the 3D-aware GANs is producing realistic 2D images. In other words, the datasets and the objective functions do not provide any clues for the 3D geometry. To reformulate 3D-aware generation as an easier constrained problem, we approximate the backgrounds on an opaque spherical surface.

**Background model** We model the background as a neural feature field defined on a sphere with a fixed radius. Given a ray \(r = o + td\) (\(t\) is the distance from the camera center \(o\)), we find the 3D background point on the sphere with radius \(R_{bg}\) by simply computing the ray’s intersection on the sphere surface:

\[
x_{bg} = o + \frac{-2[d \cdot o] + \sqrt{(2[d \cdot o])^2 - 4\|d\|^2(o - R_{bg})}}{2\|d\|^2}d
\]

(1)

Since the background points are on a sphere surface of fixed radius \(R_{bg}\), we further reparameterize the 3D coordinates \(x\) as 2D spherical coordinates \(s = (\theta, \phi)\) to further reduce the complexity.

Then we represent the feature field \(F_{bg}\) using a StyleGAN2-like architecture:

\[
F_{bg}(s, z_{bg}) = g^0_w \circ ... g^1_w \circ \zeta(s),
\]

(2)

where \(w = f(z_{bg})\) is the style vector produced by a mapping network \(f\) given a noise vector \(z_{bg}\), and \(\zeta\) is the positional encoding [42] of \(s\), and \(g_w\) denotes \(1 \times 1\) convolutions whose weights are modulated by \(w\). Note that there is no mapping for density from the background feature field because our background is an opaque surface.

Our background representation drastically reduces the number of points to be fed to the model, i.e., only one intersection of our sphere background and the ray \(r\). Therefore, we do not use hierarchical sampling for the background.

EG3D [5] represents a 3D feature field with three 2D planes to adopt StyleGAN architecture. K-Planes [11] represents a d-dimensional scene using \((d/2)\) planes. While these methods decompose 3D feature fields into low-dimensional feature representations to reduce the memory usage of NeRFs, BallGAN squeezes the background space into a surface to provide an easier task for 3D-aware GANs.

**Background model** We model the background as a neural feature field defined on a sphere with a fixed radius. Given a ray \(r = o + td\) (\(t\) is the distance from the camera center \(o\)), we find the 3D background point on the sphere with radius \(R_{bg}\) by simply computing the ray’s intersection on the sphere surface:

\[
x_{bg} = o + \frac{-2[d \cdot o] + \sqrt{(2[d \cdot o])^2 - 4\|d\|^2(o - R_{bg})}}{2\|d\|^2}d
\]

(1)

Since the background points are on a sphere surface of fixed radius \(R_{bg}\), we further reparameterize the 3D coordinates \(x\) as 2D spherical coordinates \(s = (\theta, \phi)\) to further reduce the complexity.

Then we represent the feature field \(F_{bg}\) using a StyleGAN2-like architecture:

\[
F_{bg}(s, z_{bg}) = g^0_w \circ ... g^1_w \circ \zeta(s),
\]

(2)

where \(w = f(z_{bg})\) is the style vector produced by a mapping network \(f\) given a noise vector \(z_{bg}\), and \(\zeta\) is the positional encoding [42] of \(s\), and \(g_w\) denotes \(1 \times 1\) convolutions whose weights are modulated by \(w\). Note that there is no mapping for density from the background feature field because our background is an opaque surface.

Our background representation drastically reduces the number of points to be fed to the model, i.e., only one intersection of our sphere background and the ray \(r\). Therefore, we do not use hierarchical sampling for the background.
Figure 3b visualizes the difference in space for each method with ray points. GIRAFFE-HD does not separate the background coordinate space from the foreground. StyleNeRF keeps multiple point candidates for the unbounded continuous depth. On the other hand, our method separates the foreground and background and bounds the background to lie on the surface. This effectively constrains the solution space and improves training stability and output quality.

**Design choice for background** One may wonder why we chose the sphere among many alternatives. First, the background should enclose the scene entirely to cover all viewing directions. Thus, an open plane is not available in wide-angle scenes. Second, the background should be identical when observed from all directions to make it easier for the generator to perform consistently well. Therefore, the spherical surface is the only reasonable choice. Appendix A provides empirical comparison.

**Foreground model** We adopt StyleNeRF [14] or EG3D [5] for foreground modeling, where a random foreground code \( z_{fg} \) is fed to StyleGAN2 [22] network to produce implicit or hybrid representation, respectively. Formally:

\[
(\Phi^f, \sigma) = F_{fg}(x, z_{fg}).
\]  

Note that our simple and effective background modeling is applicable to arbitrary 3D scene representations other than StyleNeRF and EG3D.

### 3.2. Volume rendering

Volume rendering aggregates the neural feature field along the rays through individual pixels to produce feature maps for a given camera pose. The conventional volume rendering computes the contribution of all points \{\( x_i \)\} sampled on a ray using the same equation \( T(x_i)(1 - \exp(-\sigma(x_i)\delta_i(x_i))) \), where \( T \) denotes transmittance, \( \sigma \) denotes density.

We modify the volume rendering equation to reflect our background design, a single point with full density:

\[
\phi(r) = \sum_{i=1}^{N_{fg}} T_i(1 - \exp(-\sigma_i\delta_i))\Phi_i^f + T_{bg}\Phi_{bg},
\]

where \( \phi(r) \) is an aggregated pixel feature along the ray \( r \), \( T_i = \exp(\sum_{j=1}^{i-1} -\sigma_j\delta_j) \) denotes accumulated transmittance at \( i \)-th point \( x_i \), \( \Phi_i \) and \( \sigma_i \) are the feature and the density at \( x_i \), and \( \delta_i = t_{i+1} - t_i \) is the distance between adjacent points. Since the background point is considered opaque and proceeded by all foreground points, we define its contribution using only the transmittance \( T_{bg} = \exp(\sum_{j=1}^{N_{bg}} -\sigma_j\delta_j) \). It is equivalent to placing an opaque background behind the scene in computer graphics techniques.

To synthesize high-resolution images, we employ a 2D-CNN-based super-resolution module to upsample and refine the feature maps to an RGB image as commonly done in recent methods [29, 47, 14, 5].

### 3.3. Training objectives

We use the non-saturating GAN loss \( L_{adv} \) [13] and R1 regularization \( L_{R_1} \) [25]. Additionally, we use two regularizations.

**Background transmittance loss** To ensure clear separation between foreground and background, we introduce new regularization on \( T_{bg} \). The ray through the foreground region in the image should have a high foreground density that makes \( T_{bg} \) close to 0, and thus the background feature should not affect the aggregated pixel. In contrast, foreground density should be small enough to make \( T_{bg} \) close to 1 when the ray corresponds to the background, so the aggregated pixel feature should be the same as the background feature. Therefore, we induce the transmittance of the background to be binarized:

\[
L_{bg} = \sum \min(T_{bg}, 1 - T_{bg}).
\]

**Foreground density loss** To encourage clear shape, we use foreground regularization to prevent foreground density from diffusing. Similar to Mip-NeRF 360 [3], our foreground loss penalizes the entropy of the aggregation weights on the ray to locate foreground points in the area where the actual geometry is located:

\[
L_{fg} = \sum_r \left( \sum_{i,j} w_i^r w_j^r |t_i^r - t_j^r| + \frac{1}{3} \sum_i w_i^r \delta_i^r \right),
\]

where \( i \) and \( j \) are the indices of the weight, \( r \) is the index of the ray, \( \delta_i = t_{i+1} - t_i \) is the distance between adjacent points and \( w \) is the aggregation weights after sigmoid function. This regularization is the integral of the weighted distance between all pairs of points on each ray.

The total loss function is then

\[
L_{total} = L_{adv} + \lambda_{R_1} L_{R_1} + \lambda_{fg} L_{fg} + \lambda_{bg} L_{bg},
\]

where \( \lambda_{R_1}, \lambda_{fg} \) and \( \lambda_{bg} \) are hyperparameters.

### 4. Experiments

In this section, we evaluate the effectiveness of BallGAN compared to the baselines regarding the faithfulness of foreground-background separation in §4.1, effectiveness on complex backgrounds in §4.2, the faithfulness of underlying 3D geometry in §4.3, and image quality in §4.4. Implementation details are in Appendix D.
Datasets We validate our method on two front-facing datasets, FFHQ [21] and AFHQv2-Cats [8, 20], and one 360° dataset, CompCars [48]. FFHQ has 70K images of real human faces, and AFHQv2-Cats contains 5,558 images of cat faces. We resize the resolutions of these datasets to $512^2$. CompCars contains 136K images of cars with various resolutions and aspect ratios. In CompCars, we use a center cropping for each image and resize it to $256^2$.

Competitors For our main comparisons we use EG3D [5], StyleNeRF [14] and GIRAFFE-HD [47]. We include EpiGRAF [40] 1, MVCGAN [52], VolumeGAN [46] and StyleSDF [30] for quantitative comparisons.

1 By incorporating NeRF++’s inverse sphere parameterization, EpiGRAF can separate foreground and background, same as StyleNeRF. However, the reported performance in the paper is based on a setting without the utilization of background representation. The official repository indicates a performance drop of approximately 10% to 15% when background representation is employed. Therefore, we employ the official version of EpiGRAF that doesn’t use the background representation as a competitor. Refer to the Appendix G for a detailed ablation study using EpiGRAF, which adopts NeRF++ as the background representation.

4.1. Foreground separation

To achieve reasonable 3D perception and applicability, accurately separating foreground and background is an important evaluation factor. As the background on a spherical surface is one of the key components of our method, we evaluate the separability and geometry of foregrounds against GIRAFFE-HD and StyleNeRF. EG3D is excluded because it does not provide separation.

Comparison Figure 4 shows rendered images of foreground and background, respectively. GIRAFFE-HD uses an alpha mask for detailed foreground separation, but it relies on 2D feature maps instead of understanding the 3D scene. Therefore, the foreground partly includes the background. StyleNeRF shows some ability to separate the foreground on FFHQ, but fails to do so for all cases of AFHQ-cats, which contain a significant amount of fine-grained details. By contrast, our results demonstrate fine-grained foreground separation, including intricate details like cat whiskers. Please refer to Appendix E for quantitative evaluation (User study).

Content creation Figure 5 demonstrates the content creation capabilities achievable with BallGAN. Given a real image, its inversion on BallGAN provides 3D foreground that can be rendered in novel views and combined with different backgrounds. The alpha channel for the background is computed from the background transmittance in the volume rendering step, i.e., the last term in (4). Even the facial
regions that are not seen in the original images are realistic in the rendered images, such as parts of hair or chin. Note that Figure 5 has a wider field-of-view than the standard to produce more diverse results.

Figure 6 demonstrates the potential of BallGAN to 3D content creation. We can synthesize novel views of the edited foregrounds by inverting images to the latent space and using text-guided latent editing [31]. Note that the 3D shapes are properly changed by the editing, e.g., hair. Therefore, BallGAN is useful for 3D content creation thanks to its foreground-background separation.

4.2. Effectiveness on complex backgrounds

Here, we demonstrate the effectiveness of our idea on complex backgrounds and wide camera angles, i.e. CompCars dataset. To use CompCars dataset where EG3D is not applicable due to the absence of a camera pose estimator, we apply a sphere background to StyleNeRF, namely BallGAN-S.

Training stability Figure 7 compares image quality of BallGAN-S and StyleNeRF using Fréchet Inception Distance (FID) [17] over iterations. While StyleNeRF diverges as the image resolution grows from 128 to 256, BallGAN-S smoothly converges below the reported FID of StyleNeRF. It implies that our method is generally beneficial to different foreground backbones and greatly improves training stability.

Comparisons In Figure 8, we present qualitative results of BallGAN-S, which showcase the robustness of our design on CompCars. Figure 8a shows that both GIRAFFE-HD and StyleNeRF exhibit a deficiency in fidelity in their modeled 3D compared to the quality of the generated images. On the other hand, ours maintains a high level of fidelity for both images and 3D models. In Figure 8b, we demonstrate that our simple yet effective idea ensures successful separation of foreground and background, even for datasets with complex backgrounds and wide camera angles. Quantitative comparisons will be addressed in §4.4.

4.3. Faithfulness of the underlying 3D geometry

It is essential for 3D-aware GANs to model the correct 3D geometry of the scenes so that their rendered images on arbitrary camera poses are convincing views of the real 3D scenes. Quantitative comparisons are followed by qualitative comparisons.

Quantitative results We quantitatively compare the underlying 3D model following the protocols in EG3D [5]. In Table 1, ID measures multi-view facial identity consistency, Depth indicates MSE of the expected depth maps from density against estimated depth-maps in frontal view, and Pose implies controllability by MSE between the estimated pose of synthesized image and the input (target) pose. Appendix F describes further details of the protocol. BallGAN outperforms the baselines in all metrics evaluating 3D geometry.

---

2This phenomenon is also reported in the official repository.

3The mean Arccosine [9] cosine similarity

4Estimations for Depth and Pose are from [10]
Table 1: Quantitative evaluation on 3D geometry. We report identity consistency (ID), pose accuracy, and depth errors for FFHQ. Our method outperforms baselines in all metrics of 3D-awareness.

<table>
<thead>
<tr>
<th>Method</th>
<th>ID</th>
<th>Pose</th>
<th>Depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVCGAN</td>
<td>0.58</td>
<td>0.014</td>
<td>0.123</td>
</tr>
<tr>
<td>VolumeGAN</td>
<td>0.63</td>
<td>0.025</td>
<td>0.020</td>
</tr>
<tr>
<td>StyleSDF</td>
<td>0.50</td>
<td>0.010</td>
<td>0.016</td>
</tr>
<tr>
<td>EpiGRAF</td>
<td>0.71</td>
<td>0.013</td>
<td>0.143</td>
</tr>
<tr>
<td>EG3D</td>
<td>0.71</td>
<td>0.007</td>
<td>0.011</td>
</tr>
<tr>
<td>GIRAFFE-HD</td>
<td>0.69</td>
<td>0.064</td>
<td>0.058</td>
</tr>
<tr>
<td>StyleNeRF</td>
<td>0.64</td>
<td>0.018</td>
<td>0.013</td>
</tr>
<tr>
<td>Ours</td>
<td>0.75</td>
<td>0.005</td>
<td>0.008</td>
</tr>
</tbody>
</table>

Table 2: **COLMAP point cloud reconstruction** is performed using 128 views in $[\pi/2, -\pi/2]$ from the generated scene for each model. A higher number of reconstructed points indicates better multi-view consistency.

<table>
<thead>
<tr>
<th>Method</th>
<th>GIRAFFE-HD</th>
<th>StyleNeRF</th>
<th>EG3D</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td># of rec. ($10^4$)</td>
<td>17 ± 2.3</td>
<td>53 ± 8.4</td>
<td>78 ± 5.5</td>
<td>79 ± 5.0</td>
</tr>
</tbody>
</table>

We further push the evaluation: the number of reconstructed points from 128 views by COLMAP [34] in five inverted samples of FFHQ training set. Table 2 provides the numbers and example point clouds of the methods. Since COLMAP reconstructs the points with high photometric consistency, the larger number of points indicates higher multi-view consistency. BallGAN demonstrates superior performance in terms of multi-view consistency, especially in the face and hair region where the number of reconstructed points is substantially higher than other methods. While EG3D also achieves a similar number of reconstructed points as BallGAN, a large portion of these points lies on the background walls rather than the face. As the comparison results show, our sphere background induces the synthesis of accurate foreground geometry, thereby improving multi-view consistency.

**Qualitative comparison: generated scenes** Figure 9 compares how each method renders generated scenes on different perspectives, expecting the images to have multi-view consistency and realism. The leftmost column provides meshes of the scene for reference. We notice severe distortions in GIRAFFE-HD and StyleNeRF when the camera rotates more than $\pm 60^\circ$ implying their spurious 3D geometry (red box in Figure 9). This problem is evident in the marching cube results of GIRAFFE-HD, which separately models foreground and background but without their separate ranges. StyleNeRF produces rough geometry and camouflages detailed shapes with color. Discussion on the missing backgrounds is deferred to Appendix G.

Similarly, the rendered images of EG3D show distortions from $\pm 60^\circ$ angles, e.g., the ears are truncated first and then the cheeks at $\pm 90^\circ$ angles (blue box in Figure 9). The mesh explains that the faces are engraved to a concave wall expanding from the ridge of the faces. Furthermore, although the meshes show greater detail compared to StyleNeRF, there are areas of disagreement between the underlying geometry and its rendered images, e.g., the boundary between hair and forehead is fuzzy in the geometry, whereas it becomes clear after color rendering.

On the other hand, BallGAN synthesizes realistic images that maintain consistency across multiple views, even when rendered in extreme side views. It implies that the separate background on a sphere removes the depth ambiguity and does not interfere with the foreground object. Notably, we observe a significant enhancement in fine details, such as hair and whiskers. For a more detailed multi-view comparison with all baseline models, please refer to Appendix I.
Figure 10: Renderings and marching cubes of the same samples. Given real image omitted as all models faithfully reconstruct it. Although all methods render the target image close by inversion, the underlying 3D geometries of previous methods are all different. We adjusted the threshold for each mesh at the line where the pupils do not break.

**Qualitative comparison: inversion of real images** Figure 10 compares renderings and meshes of the same scenes through pivotal tuning inversion (PTI) [33] of real images from the training set. Although the image reconstructions of all methods are similar in target pose, the differences become more visible in different viewpoints and in their underlying 3D geometries. GIRAFFE-HD apparently produces geometry that least fits the rendered image and thus renders inconsistent images in different views. StyleNeRF captures only rough outlines and placements in the geometry so that color makes the rendered scene realistic. Especially, the mesh does not reveal the beard and the boundary between hair and forehead. While EG3D can recover realistic geometry that mostly fits the given image, it has limitations such as faces being stuck to a wall. Moreover, it fails to accurately represent details such as eyebrows or accessories, which are evident in the input image. In contrast, BallGAN excels at accurately modeling the foreground in 3D space, and even faithfully represents the details shown in the images, such as wavy hair, earrings, and eyebrows.

### 4.4. Image quality

We evaluate generated image quality on the FFHQ 512$^2$, AFHQv2-Cats 512$^2$, CompCars 256$^2$ datasets. Images for FFHQ 512$^2$, AFHQv2-Cats 512$^2$ are generated by BallGAN and images for CompCars 256$^2$ are generated by BallGAN-S.

<table>
<thead>
<tr>
<th>Sep. FG/BG</th>
<th>FFHQ 512$^2$</th>
<th>AFHQv2-Cats 512$^2$</th>
<th>CompCars 256$^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVCGAN</td>
<td>13.4(\dagger)</td>
<td>26.57(\dagger)</td>
<td>-</td>
</tr>
<tr>
<td>VolumeGAN</td>
<td>15.74</td>
<td>44.55</td>
<td>12.9(\dagger)</td>
</tr>
<tr>
<td>StyleSDF</td>
<td>19.56</td>
<td>19.44</td>
<td>-</td>
</tr>
<tr>
<td>EpiGRAF</td>
<td>9.92(\dagger)</td>
<td>6.46</td>
<td>-</td>
</tr>
<tr>
<td>EG3D</td>
<td>4.7(\dagger)</td>
<td>2.77(\dagger)</td>
<td>N/A</td>
</tr>
<tr>
<td>GIRAFFE-HD</td>
<td>6.47</td>
<td>7.33</td>
<td>7.1(\dagger)</td>
</tr>
<tr>
<td>StyleNeRF</td>
<td>10.51(\dagger)</td>
<td>21.56</td>
<td>8() (284±96)</td>
</tr>
<tr>
<td>Ours</td>
<td>5.67</td>
<td>4.72</td>
<td>4.26</td>
</tr>
</tbody>
</table>

Table 3: Quantitative comparison using FID [17] on three datasets. \(\dagger\) denotes the reported FID, and \(\dagger\) denotes the FID calculated by the official checkpoint. In other cases, we train each baseline using their official codes. In the case of StyleNeRF on CompCars, we report FID of diverged models over 3 experiments in the parenthesis. N/A denotes the model can not be trained. Bold and underline indicate the best and second-best performance. Our method shows the best score in CompCars and comparable scores with EG3D.

**Quantitative results** Table 3 compares image quality in FID. For FFHQ, AFHQv2-Cats, BallGAN outperforms all the baselines except EG3D. Although EG3D achieves the best FID, it does not support foreground-background separation and suffers in generating 3D geometry (§4.3). Furthermore, EG3D requires camera poses of real images,
Figure 11: Set of images generated by BallGAN. We sample images of 512² resolution from BallGAN on FFHQ 512² and AFHQv2-Cats 512², as well as 256² resolution images from BallGAN-S on CompCars 256². Each image is rendered with randomly sampled camera pose.

which are not always available, e.g., CompCars. On the other hand, we achieve the state-of-the-art FID on CompCars with BallGAN-S and the second-best FID on FFHQ and AFHQv2-Cats closely following EG3D. We note that CompCars has more complex backgrounds and 360° camera poses.

Qualitative results Figure 11 provides example images generated by BallGAN and BallGAN-S. Our models faithfully generate diverse samples in multiple views. More examples can be found in Appendix J.

4.5. Ablation of the losses

We conduct ablation studies to evaluate the effect of the regularizers. Figure 12 shows the effects of our foreground and background regularization. Without \( L_{fg} \), BallGAN on FFHQ occasionally generates small floating objects behind faces. \( L_{fg} \) mitigates scene diffusion, thus inhibiting the formation of subtle shape artifacts such as floating objects behind the object. Additionally, using the background regularization \( L_{bg} \), we get clearer foreground-background separation. Figure 12b shows that removing \( L_{bg} \) allows the background to participate in synthesizing the foreground. For the result without \( L_{bg} \), the beard is not entirely black, indicating partial influence from the background (red box in Figure 12b). In other words, the foreground is not fully opaque. This is because the background transmittance loss \( L_{bg} \) encourages the foreground density to either completely block or leave the space empty before the rays hit the background.

5. Conclusion

We propose a 3D-aware GAN framework named BallGAN, which represents a scene as a 3D volume within a spherical surface, enabling the background representation to lie on a 2D coordinate system. This approach resolves the challenges of training a generator to learn a 3D scene from only 2D images. Our proposed framework successfully separates the foreground in a 3D-aware manner, which enables useful applications such as rendering foregrounds from arbitrary viewpoints on top of given backgrounds. BallGAN also achieves superior performance in 3D awareness, including multi-view consistency, pose accuracy, and depth reconstruction. Additionally, our approach shows significant improvement in capturing fine image details in 3D space, compared to existing methods.

Acknowledgements This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1F1A1076241). The part of experiments was conducted on NAVER Smart Machine Learning (NSML) platform [23, 41].
References


