StyleSDF: High-Resolution 3D-Consistent Image and Geometry Generation

Roy Or-El[1] Xuan Luo[1] Mengyi Shan[1] Eli Shechtman[2]
Jeong Joon Park[3] Ira Kemelmacher-Shlizerman[1]
[1]University of Washington [2]Adobe Research [3]Stanford University

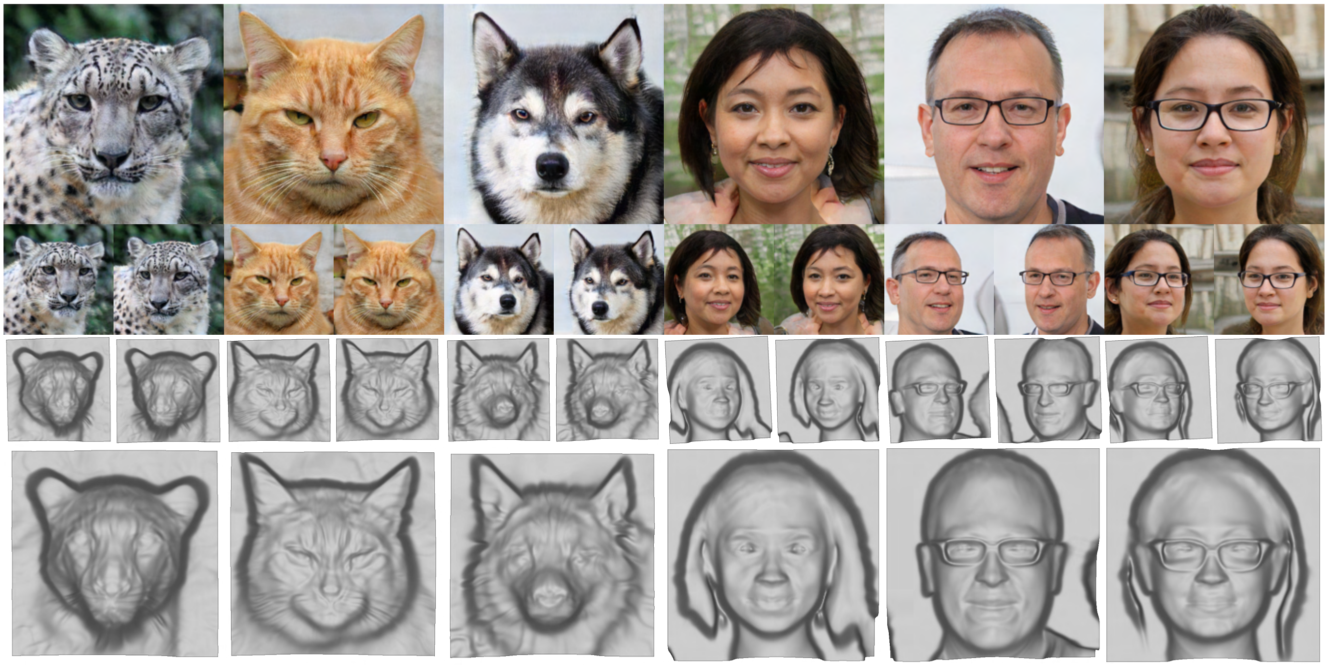


Figure 1. Our proposed framework–StyleSDF– learns to jointly generate high resolution, 3D-consistent images (top rows) along with their detailed view-consistent geometry represented with SDFs (depth maps in bottom rows), while being trained on single view RGB images.

Abstract

We introduce a high resolution, 3D-consistent image and shape generation technique which we call StyleSDF. Our method is trained on single-view RGB data only, and stands on the shoulders of StyleGAN2 for image generation, while solving two main challenges in 3D-aware GANs: 1) high-resolution, view-consistent generation of the RGB images, and 2) detailed 3D shape. We achieve this by merging a SDF-based 3D representation with a style-based 2D generator. Our 3D implicit network renders low-resolution feature maps, from which the style-based network generates view-consistent, 1024×1024 images. Notably, our SDF-based 3D modeling defines detailed 3D surfaces, leading to consistent volume rendering. Our method shows higher quality results compared to state of the art in terms of visual and geometric quality.

1. Introduction

StyleGAN architectures [35–37] have shown an unprecedented quality of RGB image generation. They are, however, designed to generate single RGB views rather than 3D content. In this paper, we introduce StyleSDF, a method for generating 3D-consistent 1024×1024 RGB images and geometry, trained only on single-view RGB images.

Related 3D generative models [9, 48, 53, 57, 62] present shape and appearance synthesis via coordinate-based multi-layer-perceptrons (MLP). These works, however, often require 3D or multi-view data for supervision, which are difficult to collect, or are limited to low-resolution rendering outputs as they rely on expensive volumetric field sampling. Without multi-view supervision, 3D-aware GANs [9,48,57] typically use opacity fields as geometric proxy, forgoing well-defined surfaces, which results in low-quality depth maps that are inconsistent across views.

Project Page: https://stylesdf.github.io/

At the core of our architecture lies the SDF-based 3D volume renderer and the 2D StyleGAN generator. We use a coordinate-based MLP to model Signed Distance Fields (SDF) and radiance fields which render low resolution feature maps. These feature maps are then efficiently transformed into high-resolution images using the StyleGAN generator. Our model is trained with an adversarial loss that encourages the networks to generate realistic images from all sampled viewpoints, and an Eikonal loss that ensures proper SDF modeling. These losses automatically induce view-consistent, detailed 3D scenes, without 3D or multi-view supervision. The proposed framework effectively addresses the resolution and the view-inconsistency issues of existing 3D-aware GAN approaches that base on volume rendering. Our system design opens the door for interesting future research in vision and graphics that involves a latent space of high quality shape and appearance.

Our approach is evaluated on the FFHQ [36] and AFHQ [13] datasets. We demonstrate through extensive experiments that our system outperforms the state-of-the-art 3D-aware methods, measured by the quality of the generated images and surfaces, and their view-consistencies.

2. Related Work

In this section, we review related approaches in 2D image synthesis, 3D generative modeling, and 3D-aware image synthesis.

Generative Adversarial Networks: State-of-the-art Generative Adversarial Networks [21] (GANs) can synthesize high-resolution RGB images that are practically indistinguishable from real images [34–37]. Substantial work has been done in order to manipulate the generated images, by exploring meaningful latent space directions [1–3, 14, 26, 29, 58, 59, 63, 64], introducing contrastive learning [60], inverse graphics [73], examplar images [32] or multiple input views [38]. While 2D latent space manipulation produces realistic results, these methods tend to lack explicit camera control, have no 3D understanding, require shape priors from 3DMM models [63, 64], or reconstruct the surface as a preprocessing step [38].

Coordinate-based 3D Models: While multiple 3D representations have been proposed for generative modeling [24, 67, 69], recent coordinate-based neural implicit models [10, 42, 53] stand out as an efficient, expressive, and differentiable representation.

Neural implicit representations (NIR) have been widely adopted for learning shape and appearance of objects [4, 11, 15, 22, 43, 49, 51, 55, 56], local parts [19, 20], and full 3D scenes [7, 12, 30, 54] from explicit 3D supervisions. Moreover, NIR approaches have been shown to be a powerful tool for reconstructing 3D structure from multi-view 2D supervision via fitting their 3D models to the multi-view images using differentiable rendering [44, 50, 62, 71].

Two recent seminal breakthroughs are NeRF [44] and SIREN [61]. NeRF introduced the use of volume rendering [33] for reconstructing a 3D scene as a combination of neural radiance and density fields to synthesize novel views. SIREN replaced the popular ReLU activation function with sine functions with modulated frequencies, showing great single scene fitting results. We refer readers to [65] for more comprehensive review.

Single-View Supervised 3D-Aware GANs: Rather than relying on 3D or multi-view supervisions, recent approaches aim at learning a 3D generative model from a set of unconstrained single-view images. These methods [9, 18, 27, 31, 40, 45–48, 57] typically optimize their 3D representations to render realistic 2D images from all randomly sampled viewpoints using adversarial loss.

Most inline with our work are methods that use implicit neural radiance fields for 3D-aware image and geometry generation (GRAF [57] and Pi-GAN [9]). However, these methods are limited to low-resolution outputs due to the high computational costs of the volume rendering. In addition, the use of density fields as proxy for geometry provides ample amount of leeway for the networks to produce realistic images while violating 3D consistency, leading to inconsistent volume rendering w.r.t. the camera viewpoints (the rendered RGB or depth images are not 3D-consistent).

To improve the surface quality, ShadeGAN [52] introduces a shading-guided pipeline, and GOF [68] gradually shrink the sampling region of each camera ray. However, the image output resolution (128×128) is still bounded by the computational burden of the volume rendering. GIRAFFE [48] proposed a dual stage rendering process. A backbone volume renderer generates low resolution feature maps (16×16) that are passed to a 2D CNN to generate outputs at 256×256 resolution. Despite improved image quality, GIRAFFE outputs lack view consistency. The hairstyle, facial expression, and sometimes the object's identity, are entangled with the camera viewpoint inputs, likely because 3D outputs at 16×16 are not descriptive enough.

Concurrent works [8, 17, 25, 74] adopt two-stage rendering process or smart sampling procedures for high-resolution image generation, yet these works still do not model well-defined, view-consistent 3D geometry.

3. Algorithm

3.1. Overview

Our framework consists of two main components. A backbone conditional SDF volume renderer, and a 2D style-based generator [37]. Each component also has an accompanied mapping network [36] to map the input latent vector into modulation signals for each layer. An overview of our architecture can be seen in Figure 2.

To generate an image, we sample a latent vector $\mathbf{z}$ from

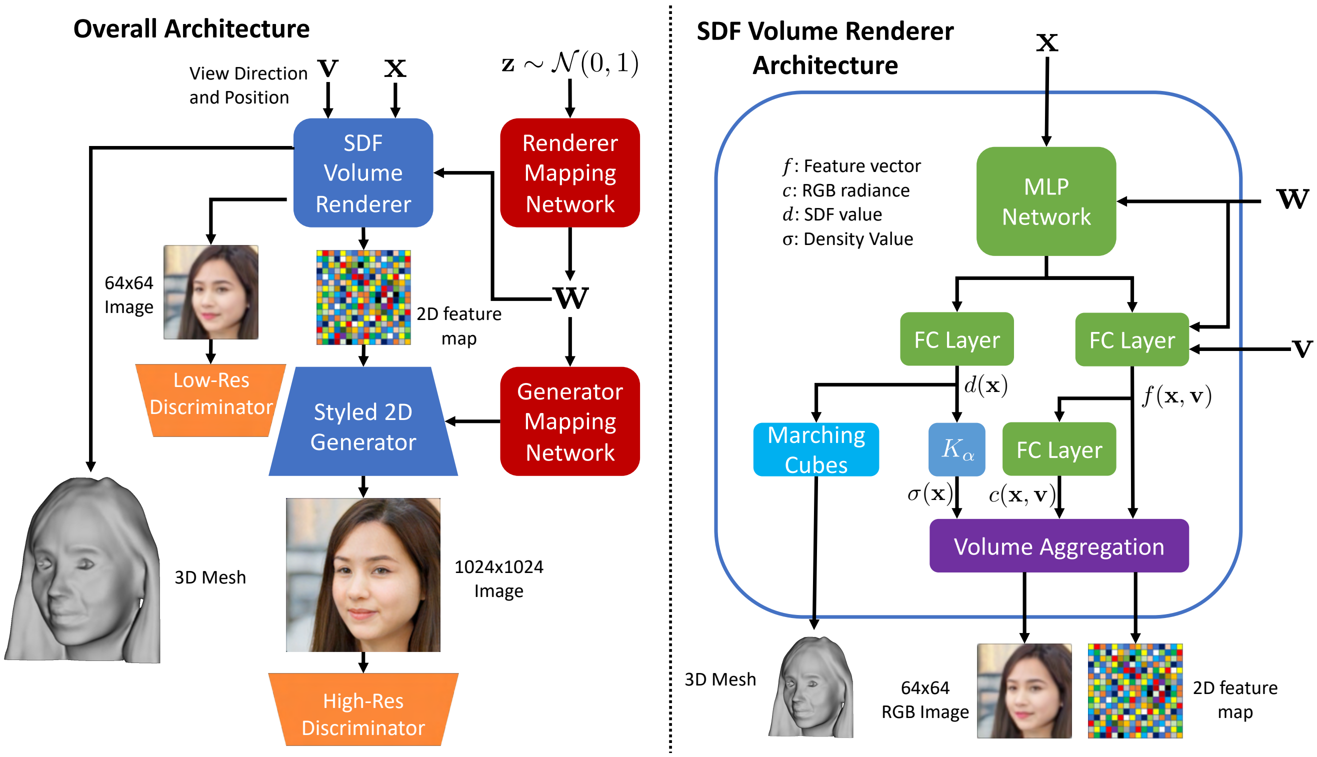


Figure 2. StyleSDF Architecture: (Left) Overall architecture: SDF volume renderer takes in a latent code and camera parameters, queries points and view directions in the volume, and projects the 3D surface features into the 2D view. The projected features are fed to the Styled 2D generator that creates the high resolution image. (Right) our SDF volume renderer jointly models volumetric SDF and radiance field, providing a well defined and view consistent geometry.

the unit normal distribution, and camera azimuth and elevation angles (ϕ, θ) from the dataset's estimated object pose distribution. For simplicity, we assume that the camera is positioned on the unit sphere and directed towards the origin. Next, our volume renderer outputs the signed distance value, RGB color, and a 256 element feature vector for all the sampled volume points along the camera rays. We calculate the surface density for each sampled point from its SDF value and apply volume rendering [44] to project the 3D surface features into 2D feature map. The 2D generator then takes the feature map and generates the output image from the desired viewpoint. The 3D surface can be visualized with volume-rendered depths or with the mesh from marching-cubes algorithm [39].

3.2. SDF-based Volume Rendering

Our backbone volume renderer takes a 3D query point, $\mathbf{x}$ and a viewing direction $\mathbf{v}$. Conditioned by the latent vector $\mathbf{z}$, it outputs an SDF value $d(\mathbf{x}, \mathbf{z})$, a view dependent color value $\mathbf{c}(\mathbf{x}, \mathbf{v}, \mathbf{z})$, and feature vector $\mathbf{f}(\mathbf{x}, \mathbf{v}, \mathbf{z})$. For clarity, we omit $\mathbf{z}$ from hereon forward.

The SDF value indicates the distance of the queried point from the surface boundary, and the sign indicates whether the point is inside or outside of a watertight surface. As shown in VolSDF [70], the SDF can be serve as a proxy for the density function used for the traditional volume rendering [44]. Assuming a non-hollow surface, we convert the SDF value into the 3D density fields σ,

$$\sigma(\mathbf{x}) = K_\alpha\left(d(x)\right) = \frac{1}{\alpha} \cdot \text{Sigmoid}\left(\frac{-d(\mathbf{x})}{\alpha}\right), \quad (1)$$

where α is a learned parameter that controls the tightness of the density around the surface boundary. α values that approach 0 represent a solid, sharp, object boundary, whereas larger α values indicate a more "fluffy" object boundary. A large positive SDF value would drive the sigmoid function towards 0, meaning no density outside of the surface, and a high-magnitude negative SDF value would push the sigmoid towards 1, which means maximal density inside the surface.

We render low resolution 64×64 feature maps and color images with volume rendering. For each pixel, we query points on a ray that originates at the camera position $\mathbf{o}$, and points at the camera direction $\mathbf{r}(t) = \mathbf{o} + t\mathbf{v}$. and calculate the RGB color and feature map as follows:

$$\begin{aligned}
\mathbf{C}(\mathbf{r}) &= \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{v})dt, \\
\mathbf{F}(\mathbf{r}) &= \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{f}(\mathbf{r}(t), \mathbf{v})dt, \qquad (2) \\
\text{where} \quad & T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right),
\end{aligned}$$

which we approximate with discrete sampling along rays.

Unlike NeRF [44] and other 3D-aware GANs such as Pi-GAN [9] and StyleNeRF [25] we do not use stratified sampling. Instead, we split $[t_n, t_f]$ into N evenly-sized bins, draw a single offset term uniformly $\delta \sim \mathcal{U}[0, \frac{t_f - t_n}{N}]$, and sample N evenly-spaced points,

$$t_i = \frac{t_f - t_n}{N} \cdot i + \delta, \quad \text{where} \quad i \in \{0, \dots, N-1\}. \tag{3}$$

In addition, we forgo hierarchical sampling altogether, thereby reducing the number of samples by 50%. We discuss the merits of our sampling strategy in the supplementary material.

The incorporation of SDFs provides clear definition of the surface, allowing us to extract the mesh via Marching Cubes [39]. Moreover, the use of SDFs along with the related losses (Sec. 3.4.1) leads to higher quality geometry in terms of expressiveness and view-consistency (as shown in Sec. 4.4), even with a simplified volume sampling strategy.

The architecture of our volume renderer mostly matches that of Pi-GAN [9]. The mapping network consists of a 3 layer MLP with LeakyReLU activation and maps an input latent code $\mathbf{z}$ into $\mathbf{w}$ space and then generates frequecny modulation, γ_i, and phase shift, β_i, for each layer of the volume renderer. The volume rendering network contains eight shared modulated FC layers with SIREN [61] activation:

$$\phi_i(x) = \sin\left(\gamma_i(W_i \cdot x + b_i) + \beta_i\right), \quad i \in \{0, \dots, 7\} \tag{4}$$

where W_i and b_i are the weight matrix and bias vector of the fully connected layers. The volume renderer then splits into two paths, the SDF path and the color path. The SDF path is implemented using a single FC layer denoted ϕ_d. In the color path, the output of the last shared layer ϕ_7 is concatenated with the view direction input and passed into one additional FiLM siren layer [9] ϕ_f followed by a single FC layer ϕ_c that generates the color output. To summarize:

$$\begin{aligned} \sigma(\mathbf{x}) &= K_\alpha \circ \phi_d \circ \phi_7 \circ \dots \circ \phi_0(\mathbf{x}), \\ f(\mathbf{x}, \mathbf{v}) &= \phi_f(\phi_7 \circ \dots \circ \phi_0(\mathbf{x}), \mathbf{v}) \\ c(\mathbf{x}, \mathbf{v}) &= \phi_c \circ \phi_f. \end{aligned} \tag{5}$$

The output features of ϕ_f are passed to the 2D style-based generator, and the generated low resolution color image is fed to a discriminator for supervision. The discriminator is identical to the Pi-GAN [9] discriminator.

We observed that using view-dependent color $\mathbf{c}(\mathbf{x}, \mathbf{v})$ tends to make the networks overfit to biases in the dataset. For instance, people in FFHQ [36] tend to smile more when facing the camera. This makes the facial expression change with the viewpoint although the geometry remains consistent. However, when we removed view-dependent color, the model did not converge. Therefore, to get view consistent images, we train our model with view dependent color, but fix the view direction $\mathbf{v}$ to the frontal view during inference.

3.3. High-Resolution Image Generation

Unlike NeRF [44], where the reconstruction loss is computed individually for each ray, adversarial training needs a full image to be present. Therefore, scaling a pure volume renderer to high-resolution quickly becomes untractable, as we need to sample over 10^7 queries to render a single 1024×1024 image. As such, we seek to fuse a volume renderer with the StyleGAN2 network that has a proven capabilities of synthesizing high-resolution 2D images.

To combine the two architectures, we truncate the early layers of the StyleGAN2 generator up until the 64×64 layer and feed the generator with the 64×64 feature maps generated by the backbone volume renderer. In addition, we cut StyleGAN2's mapping network from eight layers to five layers, and feed it with the $\mathbf{w}$ latent code from the volume renderer's mapping network, instead of the original latent vector $\mathbf{z}$. The discriminator is left unchanged.

This design choice allows us to enjoy the best of both worlds. The volume renderer learns the underline geometry, explicitly disentangles the object's pose from it's appearance, and enables full control of the camera position during inference. The StyleGAN2 generator upsamples the low resolution feature maps, adds high frequency details, and mimics complex light transport effects such as subsurface scattering and inter-reflections that are difficult to model with the low-resolution volume renderer.

3.4. Training

We employ a two-stage training procedure. First we train only the SDF-based volume renderer, then we freeze the volume renderer weights, and train the StyleGAN generator.

3.4.1 Volume Renderer training

We use the non-saturating GAN loss with R1 regularization [41], denoted $\mathcal{L}_{adv}$, to train our volume renderer. On top of that, we use 3 additional regularization terms.

Pose Alignment Loss: This loss is designed to make sure that all the generated objects are globally aligned. On top of predicting whether the image is real or fake, the discriminator also tries to predict the two input camera angles (ϕ, θ). We penalize the prediction error using a smoothed L1 loss:

$$\mathcal{L}_{view} = \begin{cases} (\hat{\theta} - \theta)^2 & \text{if } |\hat{\theta} - \theta| \leq 1 \\ |\hat{\theta} - \theta| & \text{otherwise} \end{cases}. \tag{6}$$

This loss is applied on both view angles for the generator and the discriminator, however, since we don't have ground truth pose data for the original dataset, this loss is only applied to the fake images in the discriminator pass.

Eikonal Loss: This term ensures that the learned SDF is physically valid [23]:

$$\mathcal{L}_{eik} = \mathbb{E}_{\mathbf{x}}(\|\nabla d(\mathbf{x})\|_2 - 1)^2. \tag{7}$$

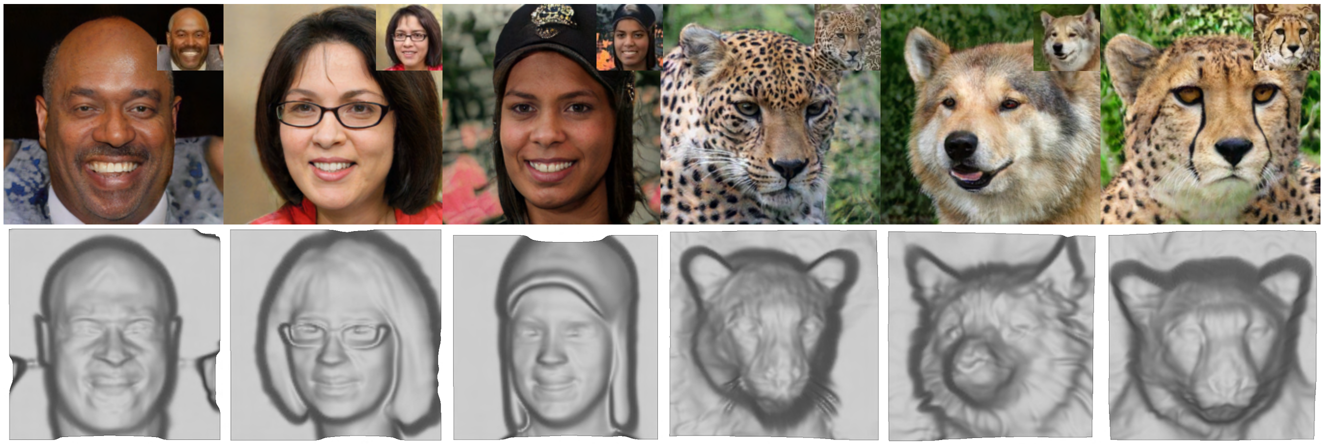


Figure 3. Generated high-res RGB images (top), low-res volume rendered images (inset) and depth maps (bottom) for the same view . The 64×64 volume rendering output features are passed to the StyleGAN generator for high-resolution RGBs. Note that the object identities and structures are preserved between the image pairs. Furthermore, as can be seen in the jaguar and cheetah examples, the StyleGAN generator occasionally corrects badly modeled background signal from the volume renderer.

Minimal Surface Loss: We encourage the 3D network to describe the scenes with minimal volume of zero-crossings to prevent spurious and non-visible surfaces from being formed within the scenes. That is, we penalize the SDF values that are close to zero:

$$\mathcal{L}_{surf} = \mathbb{E}_{\mathbf{x}}\left(\exp(-100|d(x)|)\right). \quad (8)$$

The overall loss function is then,

$$\mathcal{L}_{vol} = \mathcal{L}_{adv} + \lambda_{view}\mathcal{L}_{view} + \lambda_{eik}\mathcal{L}_{eik} + \lambda_{surf}\mathcal{L}_{surf}, \quad (9)$$

where $\lambda_{view} = 15$, $\lambda_{eik} = 0.1$, and $\lambda_{surf} = 0.05$. The weight of the R1 loss is set according to the dataset.

3.4.2 Styled Generator Training

We train our Styled generator with the same losses and optimizer parameters as the original implementation, a non saturating adversarial loss, R1 regularization, and path regularization. As in the volume renderer training, we set the weight of the R1 regularization according to the dataset.

While it is possible to have a reconstruction loss between the low-resolution and high-resolution output images, we find that the inductive bias of the 2D convolutional architecture and the sharing of style codes is strong enough to preserve important structures and identities between the images (Fig. 3).

4. Experiments

4.1. Datasets & Baselines

We train and evaluate our model on the FFHQ [36] and AFHQ [13] datasets. FFHQ contains 70,000 images of diverse human faces at 1024 × 1024 resolution, which are centered and aligned according to the procedure introduced in Karras *et al*. [34]. The AFHQ dataset consists of 15,630 images of cats, dogs and wild animals at 512 × 512 resolution. Note that the AFHQ images are not aligned and contain diverse animal species, posing a significant challenge to StyleSDF.

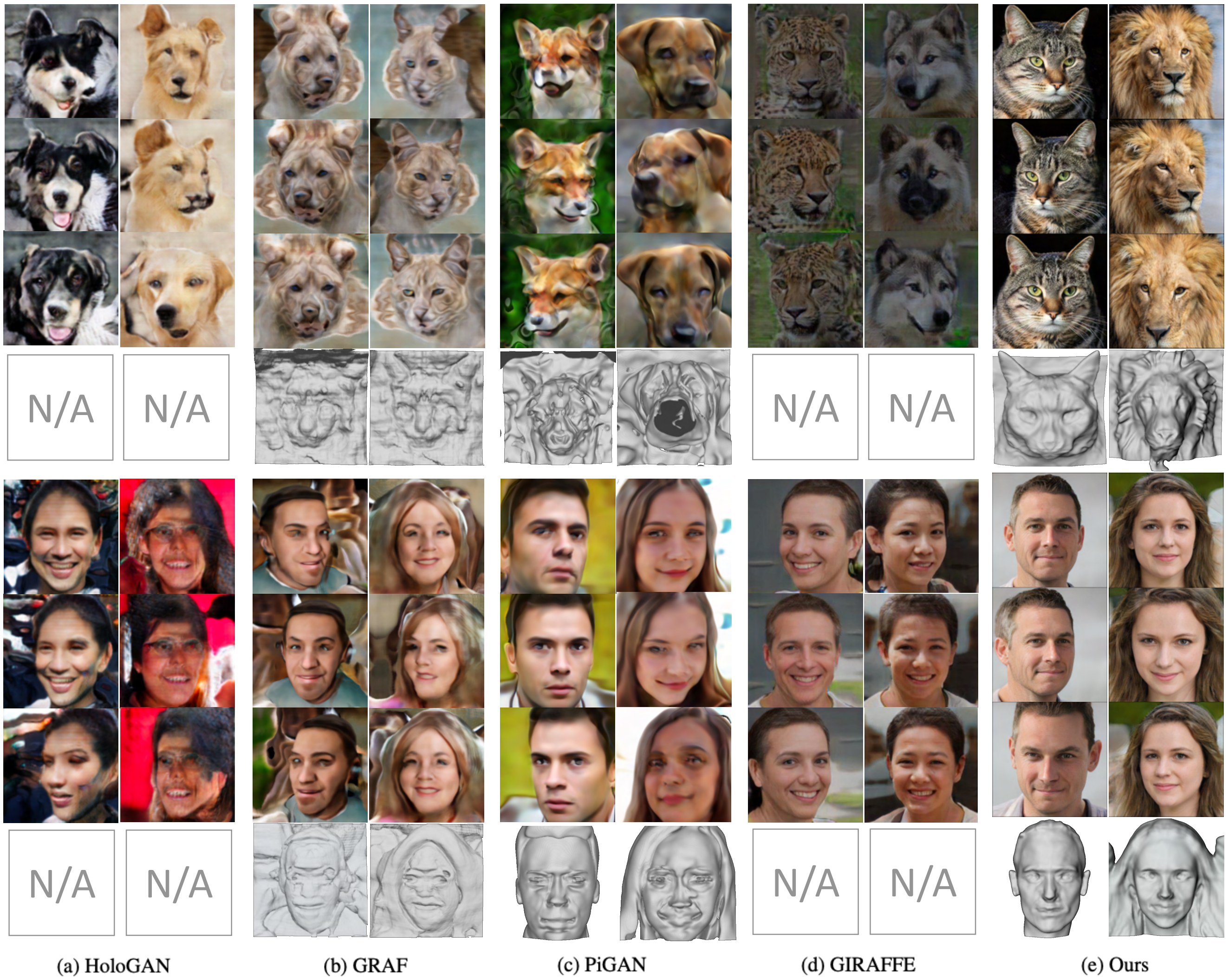
We compare our method against the state-of-the-art 3D-aware GAN baselines, GIRAFFE [48], PiGAN [9], GRAF [57] and HoloGAN [45], on the above datasets by measuring the quality of the generated images, shapes, and rendering consistency.

4.2. Qualitative Evaluations

Comparison to Baseline Methods: We compare the visual quality of our images to the baseline methods by rendering the same identity (latent code) from 4 different viewpoints, results are shown in Figure 4. To compare the quality of the underlying geometry, we also show the surfaces extracted by marching cubes from StyleSDF, Pi-GAN, and GRAF (Note that GIRRAFE and HoloGAN pipelines do not generate shapes). Our method generates superior images as well as more detailed 3D shapes. Additional generation results from our method can be seen in Figures 1 and 3.

Novel View Synthesis: Since our method learns strong 3D shape priors, it can generate images from viewpoints that are not well represented in the dataset distribution. Examples of out-of-distribution view synthesis are displayed in Figure 5.

Video Results: We urge readers to view our project's website that includes a larger set of results and videos to better appreciate the multi-view capabilities of StyleSDF.



(a) HoloGAN (b) GRAF (c) PiGAN (d) GIRAFFE (e) Ours

Figure 4. Qualitative image and geometry comparisons. We compare our sample renderings and corresponding 3D meshes against the state-of-the-art 3D-aware GAN approaches ([9, 45, 48, 57]). Note that HoloGAN and GIRAFFE are unable to create 3D mesh from their representations. Both HoloGAN (a) and GRAF (b) produce renderings that are of lower quality. The 3D mesh reconstructed from PiGAN's learned opacity fields reveal noticeable artifacts (c). While GIRAFFE (d) produces realistic low-resolution images, the identity of the person often changes with the viewpoints. StyleSDF (d) produces 1024×1024 realistic view consistent RGB, while also generating high quality 3D. Best viewed digitally.

4.3. Quantitative Image Evaluations

We evaluate the visual quality and the diversity of the generated images using the Frechet Inception Distance (FID) [28] and Kernel Inception Distance (KID) [6]. We compare our scores against the aforementioned baseline models on the FFHQ and AFHQ datasets.

All the baseline models are trained following their given pipelines to generate 256×256 images, with the exception of Pi-GAN, which is trained on 128×128 images and renders 256×256 images at inference time. The results, summarized in Table 1, show that StyleSDF performs consistently better than all the baselines in terms of visual quality. It is also on par with reported scores from concurrent works such as StyleNerf [25] and CIPS-3D [74].

4.4. Volume Rendering Consistency

Volume rendering has emerged as an essential technique to differentiably optimize a volumetric field from 2D images, as its wide-coverage point sampling leads to stable gradient-flow during training. Notably, volume rendering excels at modeling thin surfaces or transparent objects, e.g., human hairs, which are difficult to model with explicit surfaces, e.g., 3D meshes.

However, we notice that the volume rendering of existing 3D-aware GANs [9, 57] using unregularized opacity fields severely lacks view-consistency due to the absence of multi-

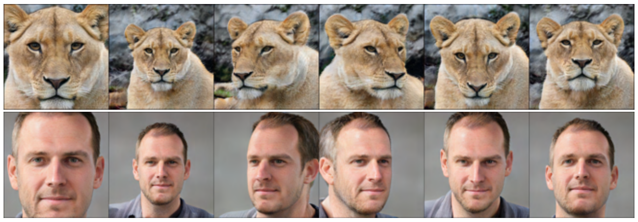


Figure 5. Out-of-distribution view synthesis (field of view and camera angles). Although StyleSDF was trained with a fixed field of view, increasing and decreasing FOV by 25% (columns 1-2) still looks realistic. Similarly with 1.5 standard deviations of the camera angles distribution used for training (columns 3-6).

Dataset:	FFHQ		AFHQ	
	FID	KID	FID	KID
HoloGAN	90.9	75.5	95.6	77.5
GRAF	79.2	55.0	129.5	85.1
PiGAN	83.0	85.8	52.4	30.7
GIRAFFE	31.2	20.1	33.5	15.1
Ours	**11.5**	**2.65**	**12.8**	**4.47**

Table 1. FID and KID evaluations. All datasets were evaluated at a resolution of 256×256. Our method demonstrates the best performance. Note that we report KID × 1000 for simplicity.

Dataset:	FFHQ	AFHQ
PiGAN	11.04	8.66
Ours	**0.40**	**0.63**

Table 2. Depth consistency results. We measure the average modified Chamfer distance (Eq. (10)) over 1,000 random pairs of depth maps for each dataset. Each pair contains one frontal view depth map and one side view depth map. Our method demonstrates significantly stronger consistency (see Fig. 6).

view supervision. That is, depth values, computed as the expected termination distance of each camera ray [16, 44], from different viewpoints do not consistently overlap in the global coordinate. This means that neural implicit features are evaluated at inconsistent 3D locations, undermining the inductive bias of the implicit 3D representation for view-consistent renderings. As such, we measure and compare the depth map consistency across viewpoints to gauge the quality of volume rendering for each system.

We sample 1,000 identities, render their 128×128 depth maps from the frontal view and a fixed side view, and compute the alignment between the two views. The depth value is defined as the expected termination distance of 128 uniformly sampled points along each ray. Note that we remove non-terminating rays whose accumulated opacity is below 0.5. We set the side viewpoint to be $1.5\times$ the standard deviation of the azimuth distribution in training. See supplementary for more experiment details.

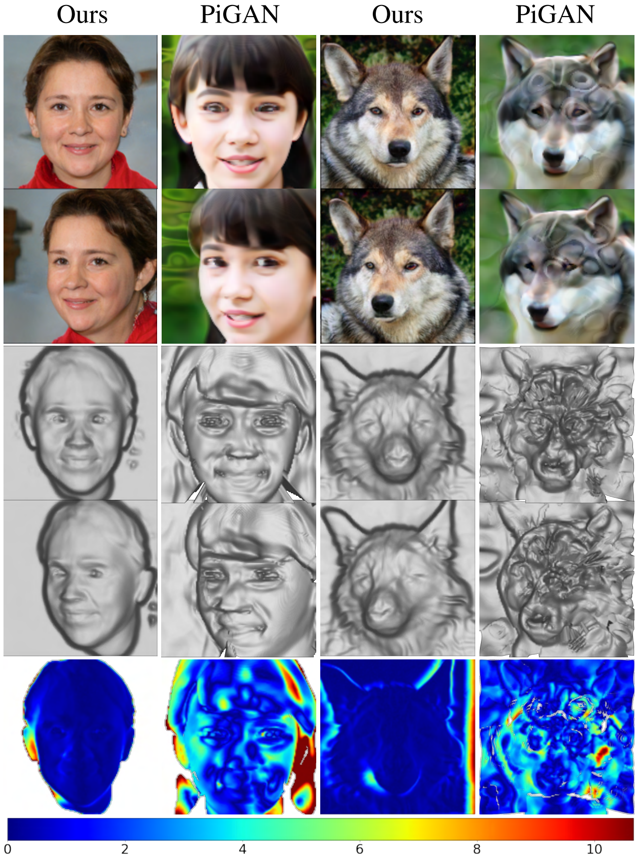


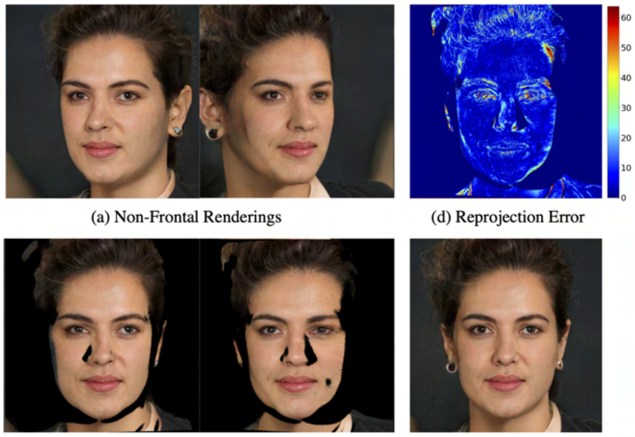
Figure 6. Visual comparison of depth consistency. We visualize the nearest neighbour distances (in sample bin units) from the frontal depth maps to side-view depth maps. Our SDF-based technique significantly improves depth consistency compared to the baseline.

To measure the alignment errors between the depth points, we adopt a modified Chamfer distance metric. I.e., we replace the usual mean distance definition with the median of the distances to nearest points,

$$\mathrm{CD}(S_1, S_2) = \underset{x \in S_1}{\mathrm{med}} \min_{y \in S_2} \|x - y\|_2^2 + \underset{y \in S_2}{\mathrm{med}} \min_{x \in S_1} \|x - y\|_2^2, \tag{10}$$

for some point sets S_1 and S_2. This metric is more robust to outliers that come from occlusion and background mismatch that we are not interested in measuring. To put the metric at scale, we normalize the distances by the volume sampling bin size.

As shown in Table 2, our use of SDF representation dramatically improves depth consistency compared to the strongest current baseline PiGAN [9]. Figure 6 shows the sample depth map pairs used for the evaluation and the error visualizations (in terms of distance to the closest point). The color map shows that our depth maps align well except for the occluded regions and backgrounds. In contrast, PiGAN



(a) Non-Frontal Renderings

(d) Reprojection Error

(b) Reprojections to Frontal View

(c) Frontal Rendering

Figure 7. RGB rendering view-consistency. We render two side views (a) and project them to the frontal view (b) using the depth maps rendered from each views, ignoring occluded pixels. Note the high simmilarity between the reprojected images and the rendered frontal view (c), as can be seen from the error map (d). The error map shows mean absolute pixel difference for RGB channels (0-255) for the right side-view image. The errors are mostly from regions with high frequency textures and geometry (e.g., ear, hair), or occlusion boundaries (right forehead).

depth maps show significant noise and spurious concave regions (e.g., nose of the dog).

Moreover, we show that our consistent volume rendering naturally leads to high view-consistency for our RGB renderings. As shown in Fig. 7, we visualize the reprojection of side-view renderings to the frontal view, using the depth values from volume rendering. The reprojected pixels closely match those of the original frontal view, indicating that our high-res multi-view RGB renderings and depth maps are all consistent to each other. Refer to supplementary for more detailed experiments.

5. Limitations & Future Work

StyleSDF might exhibit minor aliasing and flickering, e.g., in teeth area. We leave it for future work since we expect those two to be corrected similarly to Mip-NeRF [5] and Alias-free StyleGAN [35]. See example at left two columns of Figure 8. Specularities or other strong lighting effects currently introduce depth dents since StyleSDF might find it hard to disambiguate with no multi-view data (Figure 8 third column from the left). Adjusting the losses to include those effects is left for future work. Similarly, we do not currently separate foreground from background and use a single SDF for the whole image. Figure 8 (right column) shows how the cat's face is rendered properly, but the transition to the background is too abrupt, potentially diminishing photorealism. A potential solution could be adding an additional volume renderer to model the background as suggested in NeRF++ [72].

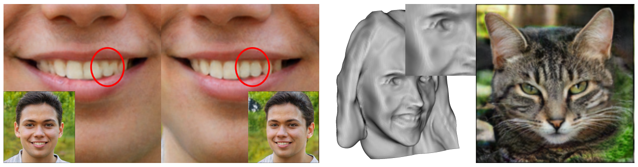


Figure 8. Limitations: potential aliasing artifacts, e.g., in teeth (left two columns). Specularities and shadows may create artifacts (3rd column from the left, cheek and eyes area), high curvatures are enhanced with radiance scaling filter [66]. Inconsistencies in background might decrease photorealism (right column).

Finally, one may consider two improvements to the algorithm. First one is training the two parts as a single end-to-end framework, instead of the current two networks. In such case the StyleGAN2 discriminator would send proper gradients back to the volume renderer to produce optimal feature maps, which might lead to even more refined geometry. However, end-to-end training poses a trade-off. The increased GPU memory consumption of this setup would require either a decreased batch size, which might hurt the overall performance, or increased training time if we keep the batch size and accumulate gradients. Second improvement could be to create a volume sampling strategy tied to SDF's surface boundary (to reduce the number of query points at each forward pass) and eliminate the need for a 2D CNN that upsamples feature maps. That would tie 3D geometry directly to the high resolution image.

6. Conclusions

We introduced StyleSDF, a method that can render 1024x1024 view-consistent images along with the detailed underlying geometry. The proposed architecture combines SDF-based volume renderer and a 2D StyleGAN network and is trained to generate realistic images for all sampled viewpoints via adversarial loss, naturally inducing view-consistent 3D scenes. StyleSDF represents and learns complex 3D shape and appearance without multi-view or 3D supervision, requiring only a dataset of single-view images, suggesting a new route ahead for neural 3D content generation, editing, and reconstruction.

Acknowledgements

We wish to thank Aleksander Holynski for his valuable advice. This work was supported by the UW Reality Lab, Meta, Google, Futurewei, Amazon and Adobe. J.J. Park was supported by the Apple fellowship.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent

space? In *ICCV*, pages 4432–4441, 2019. 2

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *CVPR*, pages 8296–8305, 2020. 2

[3] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM TOG*, 40(3):1–21, 2021. 2

[4] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *CVPR*, pages 2565–2574, 2020. 2

[5] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, pages 5855–5864, October 2021. 8

[6] Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. 6

[7] Rohan Chabra, Jan E Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *ECCV*, pages 608–625. Springer, 2020. 2

[8] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022. 2

[9] Eric R. Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *CVPR*, pages 5799–5809, 2021. 1, 2, 4, 5, 6, 7

[10] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *CVPR*, pages 5939–5948, 2019. 2

[11] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *CVPR*. IEEE, jun 2020. 2

[12] Julian Chibane, Aymen Mir, and Gerard Pons-Moll. Neural unsigned distance fields for implicit function learning. In *NeurIPS*, December 2020. 2

[13] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *CVPR*, pages 8188–8197, 2020. 2, 5

[14] Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. Editing in style: Uncovering the local semantics of gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5771–5780, 2020. 2

[15] Thomas Davies, Derek Nowrouzezahrai, and Alec Jacobson. Overfit neural networks as a compact shape representation. *arXiv preprint arXiv:2009.09808*, 2020. 2

[16] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. *arXiv preprint arXiv:2107.02791*, 2021. 7

[17] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. Gram: Generative radiance manifolds for 3d-aware image generation. In *CVPR*, 2022. 2

[18] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3d shape induction from 2d views of multiple objects. In *3DV*, pages 402–411. IEEE, 2017. 2

[19] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *CVPR*, pages 4857–4866, 2020. 2

[20] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *CVPR*, pages 7154–7164, 2019. 2

[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014. 2

[22] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *ICML*, pages 3569–3579, 2020. 2

[23] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020. 4

[24] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018. 2

[25] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *ICLR*, 2022. 2, 4, 6

[26] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In *NeurIPS*, 2020. 2

[27] Paul Henderson, Vagia Tsiminaki, and Christoph H Lampert. Leveraging 2d data to learn textured 3d mesh generation. In *CVPR*, pages 7498–7507, 2020. 2

[28] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, page 6629–6640, 2017. 6

[29] Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. In *ICLR*, 2020. 2

[30] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *CVPR*, pages 6001–6010, 2020. 2

[31] Danilo Jimenez Rezende, SM Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3d structure from images. *NeurIPS*, 29:4996–5004, 2016. 2

[32] Omer Kafri, Or Patashnik, Yuval Alaluf, and Daniel Cohen-Or. Stylefusion: A generative model for disentangling spatial segments. *arXiv preprint arXiv:2107.07437*, 2021. 2

[33] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH computer graphics*, 18(3):165–174, 1984. 2

[34] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018. 2, 5

[35] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *arXiv preprint arXiv:2106.12423*, 2021. 1, 2, 8

[36] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 1, 2, 4, 5

[37] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, pages 8110–8119, 2020. 1, 2

[38] Thomas Leimkühler and George Drettakis. Freestylegan: Free-view editable portrait rendering with the camera manifold. *arXiv preprint arXiv:2109.09378*, 2021. 2

[39] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM TOG*, 21(4):163–169, 1987. 3, 4

[40] Sebastian Lunz, Yingzhen Li, Andrew Fitzgibbon, and Nate Kushman. Inverse graphics gan: Learning to generate 3d shapes from unstructured 2d data. *arXiv preprint arXiv:2002.12674*, 2020. 2

[41] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *ICML*, pages 3481–3490, 2018. 4

[42] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, pages 4460–4470, 2019. 2

[43] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *ICCV*, pages 4743–4752, 2019. 2

[44] 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 *ECCV*, pages 405–421. Springer, 2020. 2, 3, 4, 7

[45] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *ICCV*, pages 7588–7597, 2019. 2, 5, 6

[46] Thu Nguyen-Phuoc, Christian Richardt, Long Mai, Yong-Liang Yang, and Niloy Mitra. Blockgan: Learning 3d object-aware scene representations from unlabelled images. In *NeurIPS*, 2020. 2

[47] Michael Niemeyer and Andreas Geiger. Campari: Camera-aware decomposed generative neural radiance fields. In *3DV*, pages 951–961. IEEE, 2021. 2

[48] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, pages 11453–11464, 2021. 1, 2, 5, 6

[49] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5379–5389, 2019. 2

[50] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *CVPR*, pages 3504–3515, 2020. 2

[51] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4531–4540, 2019. 2

[52] Xingang Pan, Xudong Xu, Chen Change Loy, Christian Theobalt, and Bo Dai. A shading-guided generative implicit model for shape-accurate 3d-aware image synthesis. In *NeurIPS*, 2021. 2

[53] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, pages 165–174, 2019. 1, 2

[54] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *ECCV*, pages 523–540. Springer, 2020. 2

[55] 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 *ICCV*, pages 2304–2314, 2019. 2

[56] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. PifuHD: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *CVPR*, pages 84–93, 2020. 2

[57] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *NeurIPS*, 2020. 1, 2, 5, 6

[58] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *CVPR*, pages 9243–9252, 2020. 2

[59] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *CVPR*, pages 1532–1540, 2021. 2

[60] Alon Shoshan, Nadav Bhonker, Igor Kviatkovsky, and Gérard Medioni. Gan-control: Explicitly controllable gans. In *ICCV*, pages 14083–14093, 2021. 2

[61] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 4

[62] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *NeurIPS*, 2019. 1, 2

[63] Ayush Tewari, Mohamed Elgharib, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. Pie: Portrait image embedding for semantic control. *ACM TOG*, 39(6):1–14, 2020. 2

[64] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *CVPR*, pages 6142–6151, 2020. 2

[65] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Yifan Wang, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, Tomas Simon, Christian Theobalt, Matthias Niessner, Jonathan T. Barron, Gordon Wetzstein, Michael Zollhoefer, and Vladislav Golyanik. Advances in neural rendering. *arXiv preprint arXiv:2111.05849*, 2021. 2

[66] Romain Vergne, Romain Pacanowski, Pascal Barla, Xavier Granier, and Christopher M. Schlick. Radiance scaling for versatile surface enhancement. In *I3D '10*, 2010. 8

[67] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 82–90, 2016. 2

[68] Xudong Xu, Xingang Pan, Dahua Lin, and Bo Dai. Generative occupancy fields for 3d surface-aware image synthesis. *NeurIPS*, 34, 2021. 2

[69] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4541–4550, 2019. 2

[70] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *arXiv preprint arXiv:2106.12052*, 2021. 3

[71] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[72] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 8

[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. In *ICLR*, 2020. 2

[74] Peng Zhou, Lingxi Xie, Bingbing Ni, and Qi Tian. Cips-3d: A 3d-aware generator of gans based on conditionally-independent pixel synthesis. *arXiv preprint arXiv:2110.09788*, 2021. 2, 6