GRAM: Generative Radiance Manifolds for 3D-Aware Image Generation

Yu Deng*[1,2] Jiaolong Yang[2] Jianfeng Xiang*[3,2] Xin Tong[2]
[1]Tsinghua University [2]Microsoft Research Asia [3]USTC
{t-yudeng,jiaoyan,v-jxiang,xtong}@microsoft.com

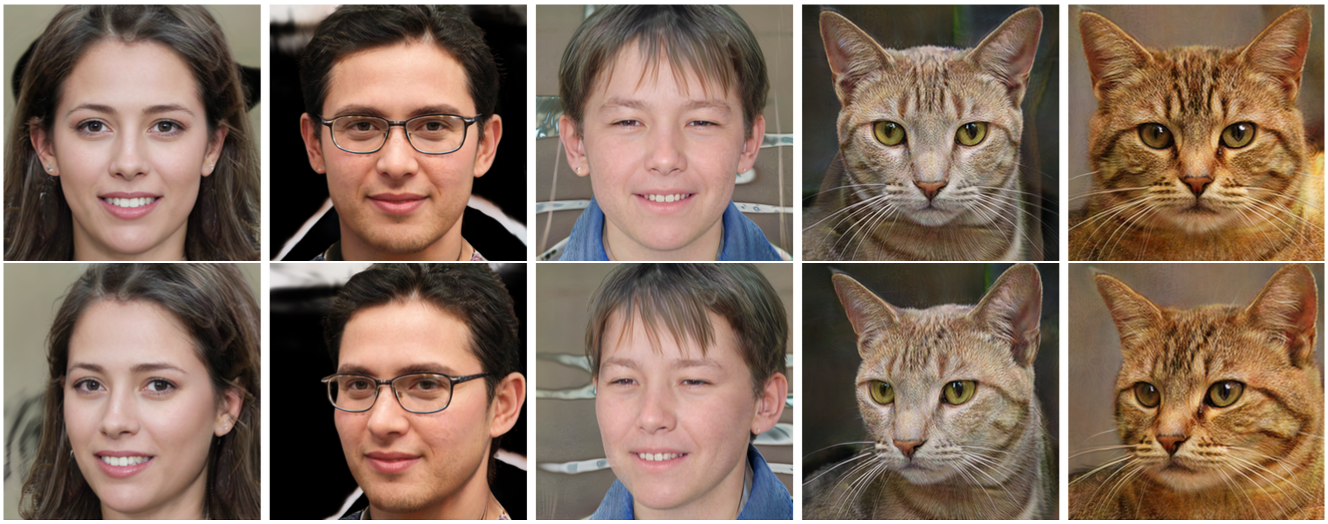

Figure 1. Image samples randomly generated by our method (256×256 resolution). Trained on unstructured image collections (FFHQ [28] and Cats [70] in this figure), our method can generate view-controllable images that are of high quality (*e.g.*, see the fine details) and strong 3D consistency (*e.g.*, see the correct parallax when view changes). (The second row contains ***animations*** best viewed in Adobe Reader; more can be found on the ***project page***)

Abstract

3D-aware image generative modeling aims to generate 3D-consistent images with explicitly controllable camera poses. Recent works have shown promising results by training neural radiance field (NeRF) generators on unstructured 2D images, but still can not generate highly-realistic images with fine details. A critical reason is that the high memory and computation cost of volumetric representation learning greatly restricts the number of point samples for radiance integration during training. Deficient sampling not only limits the expressive power of the generator to handle fine details but also impedes effective GAN training due to the noise caused by unstable Monte Carlo sampling. We propose a novel approach that regulates point sampling and radiance field learning on 2D manifolds, embodied as a set of learned implicit surfaces in the 3D volume. For each viewing ray, we calculate ray-surface intersections and accumulate their radiance generated by the network. By training and rendering such radiance manifolds, our generator can produce high quality images with realistic fine details and strong visual 3D consistency. Code available.

*Work done when YD and JX were interns at MSRA.

1. Introduction

Learning 3D-aware image generation with Generative Adversarial Networks (GAN) [20] has attracted a surge of attention in recent years [11, 13, 15, 24, 34, 44–46, 57]. Given an unstructured 2D image collection, GANs are trained to synthesize geometrically-consistent multiview imagery of novel instances. In particular, methods [11, 24, 57] that use the volumetric rendering paradigm [18, 27] to composite an output image have demonstrated impressive results with more "strict" 3D consistency by virtue of an explicit, physics-based rendering process.

Notwithstanding the promising results shown by these methods, the image quality still lags far behind traditional 2D image synthesis, for which state-of-the-art GAN models [28, 29] can generate high-resolution and photorealistic images. One prominent hurdle is the high computation and memory requirements for training a volumetric representation. Methods [11, 57] that use neural radiance field (NeRF) [42] generators can greatly reduce the complexity of voxel-based approaches [24], but the volume integrations approximated by sampling points along viewing rays are still costly for both training and inference.

This problem becomes even more pronounced in GAN training where a full image (rather than sparse pixels) needs to be rendered to train the discriminator. One workaround is to render patches during training [57], but using a patch discriminator may lead to inferior image generation quality. With an image discriminator, the state-of-the art method [11] can only afford training on smaller image resolution and with significantly reduced number of sampling points per ray (typically a few dozens) compared to standard NeRF [42]. However, we observed that radiance integration using Monte Carlo sampling becomes unstable with insufficient samples. The integrated colors among adjacent pixels suffer from intractable noise patterns that are detrimental to GAN training. An even worse issue is that optimizing a full radiance volume requires the sampling to cover both low-frequency regions and high-frequency details, leading to even less sample budget for the latter. Consequently, it is extremely difficult to generate fine details as they simply can be missed by the sampling.

This paper presents a novel method named Generative Radiance Manifolds (GRAM). Different from the previous methods, we constrain our point sampling and radiance field learning on 2D manifolds, embodied as a set of implicit surfaces. These implicit surfaces are shared for the trained object category, jointly learned with GAN training, and fixed at inference time. To generate an image, we accumulate the radiance along each ray using ray-surface intersections as point samples.

There are several advantages of our GRAM method. First, by confining sampling and radiance learning in a reduced space rather than anywhere in the volume, it greatly facilitates fine detail learning. The network can easily learn to generate thin structures and texture details on the surface manifolds which are guaranteed to have projections on the image and receive supervision during GAN training. Besides, our generated images are free from the noise pattern caused by inadequate Monte Carlo sampling, as the ray-surface intersections are deterministically calculated and smoothly varying across rays. Even with very few point samples (*i.e.*, learning very few surfaces), our method can still learn to generate quality results. As a byproduct, at inference time we can render a generated instance in real time by pre-extracting the surfaces with their radiance.

Our implicit surfaces are defined as a set of isosurfaces in a scalar field predicted by a light-weight MLP network. Another MLP for radiance generation is employed, for which we use a structure similar to [11]. We extract ray-surface intersections in a differentiable manner, and the whole framework is trained end-to-end using adversarial learning. Orthogonal to our novel radiance manifold design, we also explore network architecture and training method enhancements. In particular, we modify the network structure of [11] inspired by [29] and remove the progressive growing strategy used therein. Progressive growing not only introduces additional hyperparameters to tune but may also lead to degraded image quality shown in traditional 2D GAN [29]. We also empirically find that our method generates better results by removing it.

Our method is evaluated on multiple datasets including FFHQ [28], Cats [70], and CARLA [16, 57]. We show that our 3D-aware generation method significantly outperforms the prior art. It can synthesize highly realistic images with geometrically-consistent fine details, which are unseen in previous results. We believe our method makes a significant step towards diminishing the quality gap between 3D-aware generation and traditional 2D image generation.

2. Related Work

Neural scene representation and rendering. For scene representation and synthesis, a large volume of works [5, 8, 17, 19, 26, 30, 32, 36, 43, 52, 61, 62, 64, 65, 74, 75] adopt neural networks as a new type of rendering tool due to their ability to synthesize high-quality images without requiring excessive human labor. Among them, earlier works employ convolutional networks for a variety of applications such as novel view synthesis [23, 41, 60, 66], image-to-image translation [7, 51, 52, 67], and controllable image manipulation [1, 4, 55, 71].

More recently, plenty of works [10, 40, 42, 47, 49, 56, 59, 61, 68] leverage implicit neural representations to model 3D scenes using Multi-Layer Perceptrons (MLP). The continuous representation of MLPs brings them the superiority at 3D-level control of image synthesis compared to conventional CNN-based methods. Among these approaches, NeRF [3, 42] shows promising results in capturing complex scene structures and synthesizing 3D-consistent images with fine details. Most of the NeRF-based methods [35, 38, 48, 50, 54] focus on scene-specific learning tasks where a network is trained to fit a set of posed images of a certain scene. Only a few recent methods [11, 22, 46, 57] work on the image generation task using unconstrained 2D images for supervision. This paper proposes a new generative model for improving the image generation quality while maintaining the 3D consistency of generated contents.

3D-Aware Image Generation. Given uncontrolled 2D image collections, 3D-aware image generation methods aim to learn a generative model that can explicitly control the camera viewpoint of the generated content. To achieve this goal, the literature mainly follows two directions. The first line of works [21, 34, 44, 46, 72] utilize 3D-aware features to represent a scene, and apply a neural renderer, typically a CNN, on top of them for realistic image synthesis. For example, HoloGAN [44] and BlockGAN [45] learn low-resolution voxel features for objects, project them onto 2D image plane, and apply a StyleGAN-like [28] CNN to gen-

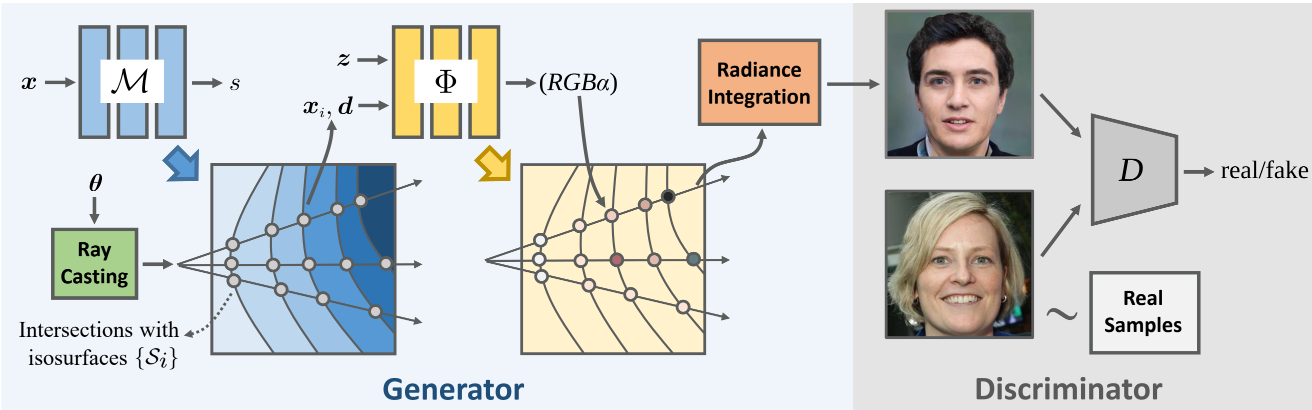

Figure 2. Overview of the GRAM method. The generator G consists of a manifold predictor $\mathcal{M}$ and a radiance generator Φ. $\mathcal{M}$ predicts multiple isosurfaces which define the input domain of Φ. The intersections between camera rays and the isosurfaces are sent to Φ for color and occupancy prediction. Images are then generated by compositing the color of the points along the ray.

erate higher-resolution images. Liao *et al.* [34] first generate 3D primitives using a 3D generator and then apply a 2D generator with an encoder-decoder structure on the projected features. Giraffe [46] and GANcraft [22] instead use 3D volumetric rendering to generate 2D feature maps for the subsequent image generation. Following a similar idea, some works concurrent to ours [21, 72] focus on designing better rendering networks to enable 3D-aware image generation at very high resolution. Nevertheless, an inevitable problem of these methods is the sacrifice of exact multi-view consistency due to the learned black-box rendering.

Another group of works [11, 15, 57, 58, 63] seek to learn direct 3D representation of scenes and synthesize images under physical-based rendering process to achieve more strict 3D consistency. [63] and [58] adopt a mesh-based representation and generate images via rasterization. However, they cannot well handle complicated structures with non-Lambertian reflectance such as hair and fur. Recent methods [11, 15, 57] use the NeRF representation to synthesize images with high 3D consistency. Still, the expensive computational cost of volumetric representation learning prevents them from generating images with adequate details. In this work, we propose a novel approach to learn a generative radiance field on 2D manifolds, and we achieve more realistic image generation with finer details significantly outperforming the previous methods.

3. Approach

Given a collection of real images, we learn a 3D-aware image generator G which takes a random noise $\boldsymbol{z} \in \mathbb{R}^d \sim p_z$ and a camera pose $\boldsymbol{\theta} \in \mathbb{R}^3 \sim p_\theta$ as input, and outputs an image I of a synthetic instance under pose $\boldsymbol{\theta}$:

$$G : (\boldsymbol{z}, \boldsymbol{\theta}) \in \mathbb{R}^{d+3} \to I \in \mathbb{R}^{H\times W\times 3}. \quad (1)$$

Figure 2 shows the overall structure of G, which consists of a manifold predictor $\mathcal{M}$ and a radiance generator Φ. The manifold predictor $\mathcal{M}$ defines a scalar field which derives a reduced domain for radiance generation, which is composed of multiple implicit isosurfaces (Sec. 3.1). Given a latent code $\boldsymbol{z}$, the radiance generator Φ generates the occupancy and color for points on the manifolds (Sec. 3.2). Images are then generated by integrating the color of the manifold points along each viewing ray (Sec. 3.3). The whole method is trained end-to-end in an adversarial learning framework (Sec. 3.4). After training, GRAM can render high-quality and 3D-consistent images from different viewpoints.

3.1. Manifold Predictor

Our manifold predictor $\mathcal{M}$ predicts a reduced space for point sampling and radiance field learning, which is shared across all generated instances. We implement it as an scalar field function which determines a set of isosurfaces with specific scalar levels. Specifically, $\mathcal{M}$ is a light-weight MLP which takes a point $\boldsymbol{x}$ as input and predicts a scalar value s:

$$\mathcal{M} : \boldsymbol{x} \in \mathbb{R}^3 \to s \in \mathbb{R}. \quad (2)$$

Given the predicted scalar field, we obtain N isosurfaces $\{\mathcal{S}_i\}$ with different levels $\{l_i\}$:

$$\mathcal{S}_i = \{\boldsymbol{x} | \mathcal{M}(\boldsymbol{x}) = l_i\}. \quad (3)$$

These levels are predefined constant values. Note that although the scalar field is defined in the 3D volume of the scene to be rendered, the scalar values per se have no physical meaning and the levels $\{l_i\}$ can be trivially chosen.

We define the input domain of the radiance generator to be on these surfaces. Let $\{\boldsymbol{x}_i\}$ be the N intersections between a camera ray $\boldsymbol{r} = \{\boldsymbol{o} + t\boldsymbol{d}, t \in [t_n, t_f]\}$ and $\{\mathcal{S}_i\}$, *i.e.*,

$$\{\boldsymbol{x}_i\} = \{\boldsymbol{x} | \boldsymbol{x} = \boldsymbol{o} + t\boldsymbol{d}, \boldsymbol{x} \in \{\mathcal{S}_i\}, t \in [t_n, t_f]\}, \quad (4)$$

we only pass these points to the radiance generator Φ for radiance generation and the final rendering, as shown in

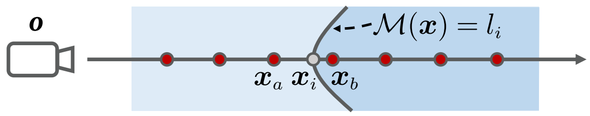

Figure 3. Our differentiable ray-isosurface intersection computation, achieved by linear interpolation between two endpoints of a small interval.

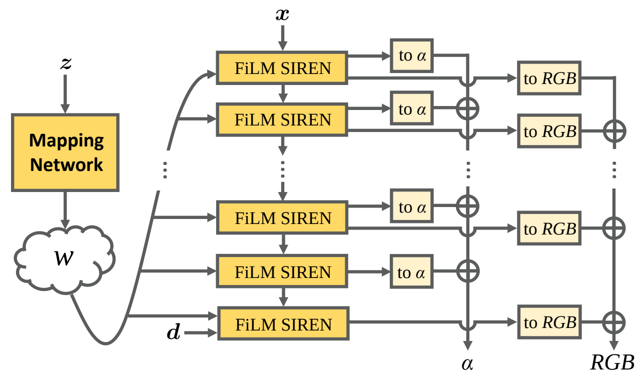

Figure 4. The network structure of radiance generator Φ.

Fig. 2. Since there is no prior knowledge for the optimal solution of the isosurfaces $\{\mathcal{S}_i\}$, we learn them automatically in the generative adversarial training process.

Training the manifold predictor $\mathcal{M}$ with GAN necessitates a differentiable scheme for ray-surface intersection computation in order to backpropagate the adversarial loss. To this end, we follow Michael *et al*. [47]'s strategy to calculate the intersections. As shown in Fig. 3, we evenly sample points along a ray between the near and far planes and feed them to $\mathcal{M}$ to obtain their values s. Then we search for the first interval that a certain scalar level l_i falls in, and calculate the intersection using linear interpolation between the two endpoints of the interval via:

$$\boldsymbol{x}_i = \frac{l_i - s_a}{s_b - s_a}\boldsymbol{x}_b + \frac{s_b - l_i}{s_b - s_a}\boldsymbol{x}_a. \tag{5}$$

We implement $\mathcal{M}$ as a light-weight MLP with 3 hidden layers, and thus dense points along each ray can be sampled to get accurate intersections using Eq. (5).

Random initialization of $\mathcal{M}$ may give rise to highly irregular isosurfaces which is unfavourable for the training process. In this work, we adopt the geometric initialization strategy proposed by Atzmon *et al*. [2] with which the initial isosurfaces are close to spheres.

3.2. Radiance Generator

Given a latent code $\boldsymbol{z}$, our radiance generator Φ generates the radiance for points lying on the learned manifolds. Specifically, Φ is parameterized by an MLP which produces the occupancy α and color $\boldsymbol{c} = (R, G, B)$ for a point $\boldsymbol{x} \in \mathbb{R}^3$ with view direction $\boldsymbol{d}$:

$$\Phi : (\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{d}) \in \mathbb{R}^{d+6} \rightarrow (\boldsymbol{c}, \alpha) \in \mathbb{R}^4. \tag{6}$$

Since radiance is defined on surface manifolds instead of the whole volume in our method, we generate occupancy α instead of volume density σ in the original NeRF, following [48, 73].

The network structure of Φ is adapted from the FiLM SIREN backbone of [11] with some modifications, as presented in Fig. 4. Inspired by StyleGAN2 [29], we use skip connections between output layers at different levels instead of only predicting occupancy and color at the final layer done in previous methods [11, 42]. In this way, different levels of details are now predicted by different output layers and combined together to form the final results. This change not only removes the necessity of the progressive growing strategy used in previous methods, but also yields better results in our method as shown in the experiments.

3.3. Manifold Rendering

For a camera ray $\boldsymbol{r}$ which intersects the surface manifolds at points $\{\boldsymbol{x}_i\}$ sorted from near to far following Eq. (4), the rendering equation can be written as [48, 73]:

$$\begin{aligned} C(\boldsymbol{r}) &= \sum_{i=1}^{N} T(\boldsymbol{x}_i)\alpha(\boldsymbol{x}_i)c(\boldsymbol{x}_i, \boldsymbol{d}) \\ &= \sum_{i=1}^{N} \prod_{j<i} (1 - \alpha(\boldsymbol{x}_j))\alpha(\boldsymbol{x}_i)c(\boldsymbol{x}_i, \boldsymbol{d}). \end{aligned} \tag{7}$$

Our rendering scheme is clearly different from the original volume rendering in NeRF which applies a hierarchical random sampling strategy. We only use intersections between camera rays and surface manifolds which are deterministically calculated, instead of selecting points in the whole volume space in a Monte Carlo fashion. This helps us eliminate the randomness in image generation and enable training a generator with fewer point samples per ray. Moreover, it greatly facilitate fine detail learning as high-frequency structures and textures can be easily generated on the surface manifolds.

3.4. Training Strategy

At training stage, we randomly sample latent code $\boldsymbol{z}$ and camera pose $\boldsymbol{\theta}$ from prior distributions p_z and p_θ. The generator G synthesizes images with corresponding latent codes and poses as input. We also sample real images from the training data with prior distribution p_{real}. As in standard GAN [20], a discriminator D receives the generated images as well as real images and judge if they are fake or real, for which we use the same CNN structure as in [11]. We train all the networks, including the manifold predictor $\mathcal{M}$, the radiance generator Φ and the discriminator D, using

Figure 5. Uncurated 256 × 256 image samples of human face and cat generated by our method (first 13 random seeds).

non-saturating GAN loss with R1 regularization [39]:

$$\mathcal{L}(D,G) = \mathbb{E}_{\boldsymbol{z}\sim p_z,\boldsymbol{\theta}\sim p_\theta}[f(D(G(\boldsymbol{z},\boldsymbol{\theta})))] + \mathbb{E}_{I\sim p_{real}}[f(-D(I)) + \lambda||\nabla D(I)||^2], \tag{8}$$

where $f(u) = -\log(1+\exp(-u))$ is the Softplus function.

In addition, we find that for certain objects, the training process with only adversarial loss is sometimes sensitive to random initialization. In a few occasions, the learned 3D geometry of convex objects could become concave. To tackle this issue, we can optionally add a pose regularization term to enforce the generator to generate images under correct pose:

$$\mathcal{L}_{pose} = \mathbb{E}_{\boldsymbol{z}\sim p_z,\boldsymbol{\theta}\sim p_\theta}||D_p(G(\boldsymbol{z},\boldsymbol{\theta})) - \boldsymbol{\theta}||^2 + \mathbb{E}_{I\sim p_{real}}||D_p(I) - \hat{\boldsymbol{\theta}}||^2, \tag{9}$$

where D_p is an additional branch of the discriminator D that predicts the camera pose of a given image, and $\hat{\boldsymbol{\theta}}$ is the pose label of a real image. We find that this loss can also slightly improve the image generation quality for objects without the concave geometry issue observed.

4. Experiments

Implementation details. We evaluate our GRAM method on three datasets: FFHQ [28], Cats [70], and CARLA [16, 57], which contain 70K high-resolution face images, 10K cat images with various resolutions, and 10K synthetic car images of 16 car models, respectively. During training, we randomly sample latent code $\boldsymbol{z}$ from the normal distribution and camera pose $\boldsymbol{\theta}$ from the known or approximated distributions of the training datasets. For all experiments, we use the Adam optimizer [31] with $\beta_1 = 0$ and $\beta_2 = 0.9$. The initial learning rates are set to 2×10^{-5} for the generator and 2×10^{-4} for the discriminator. The models are trained on 8 Nvdia Tesla V100 GPUs with 32GB memory using a batchsize of 32 for 128^2 resolution and 16 for 256^2 resolution. Training took 3 to 7 days depending on the dataset and image resolution. More implementation details can be found in Sec. A.

4.1. Generation Results

Some random image samples generated by our method are shown in Fig. 1, 5, and 6. For face and cat, the model is trained with 256^2 resolution and 24 manifold surfaces (*i.e.*,

Figure 6. Uncurated 128×128 image samples of car generated by our method (first 8 random seeds).

24 point samples per ray). For the car images, we train on 128^2 resolution and use 48 manifold surfaces. As we can see, our method is able to generate high-quality images with fine details. Moreover, it allows an explicit control of camera viewpoint and achieves highly consistent results across different views. It even maintains strong visual 3D consistency for very thin structures such as bangs of hair, eyeglass, and whiskers of cat, which show correct parallax corresponding to realistic 3D geometry. Note that *3D consistency is best viewed with animations*, for which we refer the readers to Fig. 1 and our *project page*.

Visualization of surface manifolds. Figure 7 shows the learned surface manifolds on the three datasets. Initially, the surfaces have near-spherical shapes and are positioned across the whole volume. After training, the surfaces for face and cat are tightened and exhibit small curvatures. The surfaces for car are also tightened but maintain a curving structure that covers the car geometry. The face and cat images from FFHQ [28] and Cats [70] only have small angle variations; most of them are nearly frontal. In this case, near-planar surfaces are enough to render a generated instance. In contrast, the camera viewpoints of the car images from CARLA [57] are uniformly distributed on the upper hemisphere (*i.e.*, 360° azimuth and 90° elevation angles). Such a wide viewpoint range necessities curved surfaces to ensure good rendering results from different views.

Figure 8 shows the radiance predicted on the manifolds with two examples. We evenly sample surfaces from front to back and render the color patterns on them with their contribution to the final image as opacity. As shown in the figures, the network is able to learn high-frequency details and thin structures (*e.g.* whiskers) on the manifolds.

Visualization of 3D geometry. Although our method confines the input domain of the radiance field on 2D manifolds, we can still extract proxy 3D shapes of the generated objects using the volume-based marching cubes algorithm [37]. Figure 9 shows the proxy 3D shapes of several generated instances. It can be observed that our method produces high-quality geometry with detailed structures well depicted, which is the key to achieve strong visual 3D consistency across different views for not only low-frequency regions but also fine details.

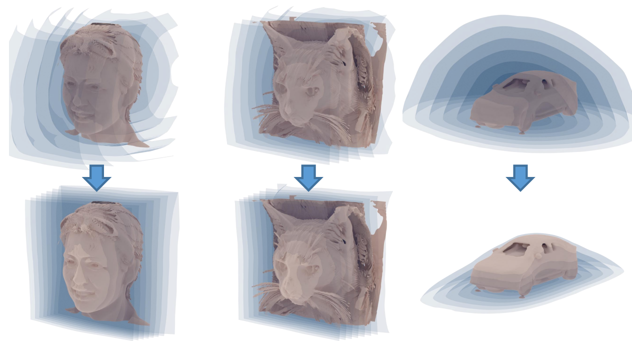

Figure 7. Initial (top) and final (bottom) surface manifolds learned on three datasets. Eight evenly-sampled surfaces are visualized here. To show the relative position of the surfaces in the 3D object space, we also visualize an extracted 3D shape for reference.

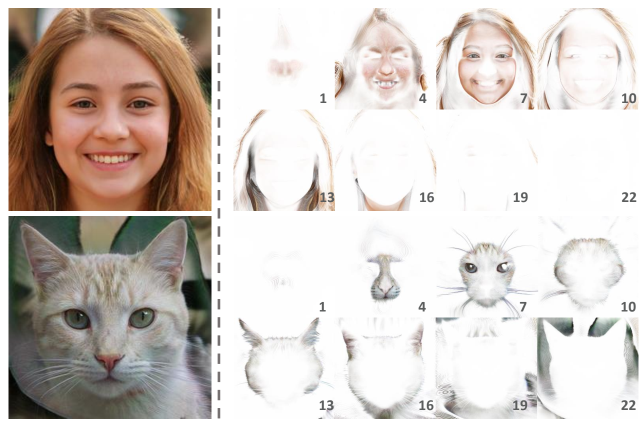

Figure 8. Visualization of generated radiance on the surface manifolds. Eight evenly sampled surfaces from front to back are shown.

Figure 9. Extracted proxy 3D shapes of the generated instances.

4.2. Comparison with Previous Methods

We compare GRAM with three state-of-the-art 3D-aware image generation approaches: GRAF [57], pi-GAN [11], and GIRAFFE [46]. Experiments are conducted using the official implementation provided by the authors. For GRAF and GIRAFFE, we modify the camera pose distribution according to different datasets, and leave other configurations unchanged. For pi-GAN, we follow the authors' settings that use 24, 48, and 96 sampling points for

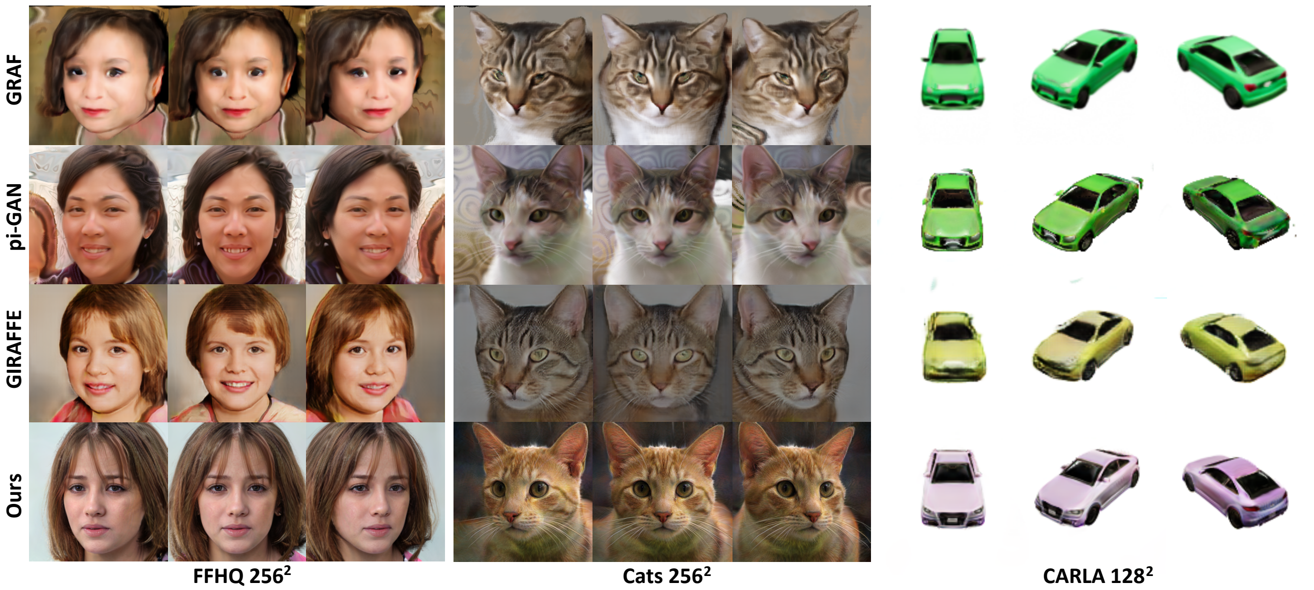

Figure 10. Qualitative comparison with previous 3D-aware image generation methods on three datasets. (**Best viewed with zoom-in**)

Figure 11. Qualitative comparison with a controllable face image generation method DiscofaceGAN. (**Best viewed with zoom-in**)

Table 1. Quantitative comparisons on three datasets using FID and KID×100 between 5K generated images and 5K real images. Results of StyleGAN2 [29] are included for reference. †: Evaluated using pre-trained models provided by the authors.

	FFHQ 256^2		Cats 256^2		CARLA 128^2	
Methods	FID	KID	FID	KID	FID	KID
StyleGAN2	12.2	0.18	10.5	0.30	12.6	0.45
GRAF	78.5	5.92	61.4	4.60	34.2†	1.81†
pi-GAN	61.3	4.23	55.4[1]	4.33	38.7†	2.11†
GIRAFFE	38.6†	2.25†	22.3	1.12	109[2]	7.29
Ours	**29.8**	**1.16**	**16.7**	**0.75**	**28.5**	**1.09**

FFHQ, Cats, and CARLA respectively, for both training and testing. Note that for our method, we use 24 surfaces for FFHQ and Cats, and 48 surfaces for CARLA.

We further compare GRAM with a face-specific controllable image generation approach: DiscofaceGAN [13], which uses a 2D CNN as the generator and achieves pose control with the guidance of a prior 3D face model [53].

[1]Due to memory limitation, we train pi-GAN on Cats up to 128^2 resolution and render at 256^2 resolution for evaluation.

[2]We tried our best to train GIRAFFE on CARLA using multiple different settings and report the best result we obtained.

Qualitative comparison. Figure 10 shows the visual comparison between GRAM and other methods. As we can see, GRAF and pi-GAN struggle to generate high-frequency details such as the texture of hair and fur. GIRAFFE produces images with finer details, but it suffers from 3D inconsistency (*e.g.*, see hair region of the woman) due to the use of a CNN renderer . Our method achieves the best visual quality with realistic details and remarkable 3D consistency. See Fig. V for more results.

Figure 11 shows the qualitative comparison between GRAM and DiscofaceGAN. While DiscofaceGAN can generate realistic face images and explicitly control their camera poses, it cannot well maintain the 3D consistency (*e.g.*, see the bangs). By contrast, GRAM achieves strong 3D consistency under comparable generation quality without requiring extra 3D face priors.

Quantitative comparison. We quantitatively evaluate the image quality using the Fréchet Inception Distances (FID) [25] and Kernel Inception Distances (KID) [6] between 5K randomly generated images and 5K sampled real images. Table 1 shows that we significantly improve the two metrics compared to GRAF and pi-GAN, which also use NeRF generators. We even achieve lower FID and KID compare to GIRAFFE which applies a refinement CNN after the NeRF rendering to achieve better image quality.

4.3. Ablation Study

We further conduct ablation study to validate the efficacy of our method designs. For efficiency, all experiments are conducted on FFHQ with 128^2 resolution. Unless otherwise specified, we use 24 points per ray for these experiments.

Table 2. Ablation study on different point sampling strategies (24 points used for each ray)

	NeRF-H [11, 42]	Planes	Spherical (init)	Ours
FID 5K	35.4	28.3	27.8	**25.8**

Table 3. Ablation study on number of sampling points per ray.

Number of points		6	12	24	36	48
FID 5K	NeRF-H [11, 42]	117	62.6	35.4	32.9	30.0
	Ours	**27.4**	**27.0**	**25.8**	**25.8**	**25.2**

Table 4. Ablation study on pose regularization.

	Real pose	NeRF-H [11, 42]	Ours
FID 5K	✗	44.4	26.4
	✓	35.4	25.8

Table 5. Ablation study on training strategy and network structure with our GRAM method.

	Base	- PG	+ Skip (Ours)
FID 5K	30.6	28.8	**25.8**

Sampling methods. We compare our manifold sampling strategy with several baseline methods as shown in Table 2. *NeRF-H* is the original hierarchical sampling strategy used in NeRF [42] and pi-GAN [11]. *Planes* denotes using intersections between camera rays and multiple parallel planes placed across the volume. *Spherical (init)* denotes sphere-like surfaces obtained from the geometric initialization [2] and fixed during training. Compare to the alternatives, our learnable manifolds yield the best image quality in terms of FID metrics. *NeRF-H* has a large performance gap with the others, indicating its deficiency under limited sample points. Our method outperforms *Planes* and *Spherical (init)*, which demonstrates the advantage of using learnable surfaces that can better fit the trained object category.

Number of surface manifolds. We further evaluate the generation quality of GRAM when training with different number of surfaces. For a reference, we also train models using the hierarchical sampling strategy *NeRF-H* with same number of sampling points for each ray. Table 3 shows that our method can generate high quality results using as few as 6 surfaces, and adding more gradually improves the quality. In contrast, training with *NeRF-H* largely fails with less than 12 points as indicated by the high FIDs, due to the difficulty to handle high-frequency details as well as the noise brought by inadequate sampling (Fig. 12). Even using 48 points, its generation quality is still worse than ours with 6 surfaces. In addition, it tends to learn unreasonable geometry with concave human foreheads, which rarely happens in our case (see Fig. VI for visual results).

Influence of pose regularization. Table 4 shows the effect of using pose labels of real images in Eq. (9) during training. For human face, our method produces slightly better results using the real pose regularization. In contrast, the hierarchical sampling strategy is unstable without real pose as guidance, leading to much worse results.

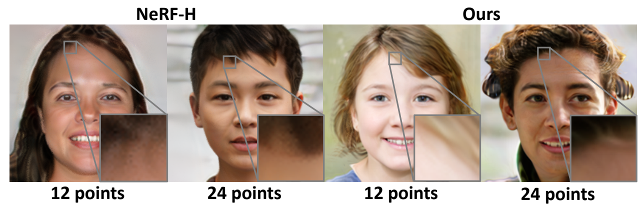

Figure 12. Images generated using NeRF-H [11, 42] sampling contain noise patterns under limited point samples whereas ours are noise-free. **(Best viewed with zoom-in)**

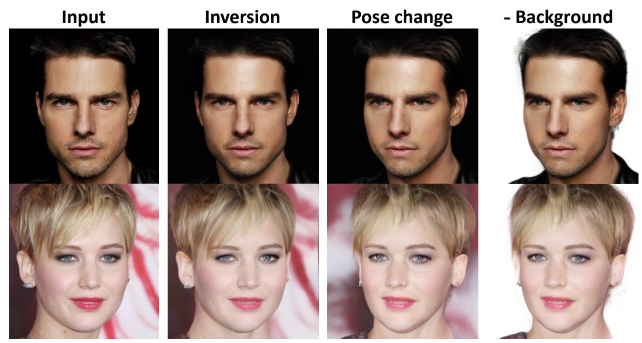

Figure 13. Image embedding and editing results.

Training strategy and network structure. As shown in Table 5, we first train our GRAM model with the network structure proposed in [11] and the progressive growing strategy from 32^2 resolution following [11], which is the *Base* setting. Then we switch to the non-progressive growing strategy by training a model from scratch using 128^2 resolution. Finally, we add skip connections in the network structure as depicted in Fig. 4. The improvements on FID clearly demonstrate the advantages of our design.

4.4. Applications

Image embedding and editing. GAN inversion is naturally supported by our GRAM method. Given an input image, we can first embed it into the learned latent space and then freely move the camera viewpoint to synthesize images at novel views. As shown in Fig. 13, we achieve 3D-consistent view manipulation of the embedded images. Thin structures such as hair look natural under camera movements, which has not been shown in the previous methods. More details and results can be found in Sec. A.4 and on the *project page*.

Real-time view synthesis. For objects generated by GRAM, we can achieve real-time free-view rendering thanks to our radiance manifold design. By pre-extracting the surface manifolds using marching cubes [37] and storing the radiance on them, we can generate images under any view using mesh rendering. With an efficient mesh rasterizer [33], we achieve 180FPS free-view rendering of 256^2 images on a Nvidia Tesla V100 GPU.

5. Conclusions

We presented a novel approach for 3D-aware image generation. The core idea is to regulate point sampling and radiance learning on 2D manifolds for the radiance generator. Extensive experiments have shown its superiority over previous methods on both generation quality and 3D consistency. We believe our method takes a large step towards generating 3D-aware virtual contents for real applications.

Ethics consideration. Our goal in this paper is to generate contents of virtual objects. We condemn any behavior to create misleading or harmful contents of real person.

Limitations and future works. Currently we mainly deals with object categories sharing similar geometry. The method may not well handle complex 3D scenes consisting of multiple subjects with diverse structures, for which we will explore in our future work.

References

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

[2] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2565–2574, 2020. 4, 8, 14

[3] 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 *IEEE/CVF International Conference on Computer Vision*, 2021. 2

[4] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *ACM Transactions on Graphics*, 38(4):1–11, 2019. 2

[5] Sai Bi, Kalyan Sunkavalli, Federico Perazzi, Eli Shechtman, Vladimir G Kim, and Ravi Ramamoorthi. Deep cg2real: Synthetic-to-real translation via image disentanglement. In *International Conference on Computer Vision*, pages 2730–2739, 2019. 2

[6] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *International Conference on Learning Representations*, 2018. 7

[7] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3722–3731, 2017. 2

[8] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. 2

[9] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *IEEE International Conference on Computer Vision*, pages 1021–1030, 2017. 13

[10] 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 *European Conference on Computer Vision*, pages 608–625, 2020. 2

[11] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5799–5809, 2021. 1, 2, 3, 4, 6, 8, 13, 14, 19

[12] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 14

[13] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3D imitative-contrastive learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5154–5163, 2020. 1, 7

[14] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 13

[15] Terrance DeVries, Miguel Angel Bautista, Nitish Srivastava, Graham W Taylor, and Joshua M Susskind. Unconstrained scene generation with locally conditioned radiance fields. In *IEEE/CVF International Conference on Computer Vision*, 2021. 1, 3

[16] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on Robot Learning*, pages 1–16, 2017. 2, 5, 13

[17] Alexey Dosovitskiy, Jost Tobias Springenberg, Maxim Tatarchenko, and Thomas Brox. Learning to generate chairs, tables and cars with convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):692–705, 2016. 2

[18] Robert A Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. *ACM SIGGRAPH*, 22(4):65–74, 1988. 1

[19] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. 2

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014. 1, 4

[21] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*, 2021. 2, 3

[22] Zekun Hao, Arun Mallya, Serge Belongie, and Ming-Yu Liu. Gancraft: Unsupervised 3d neural rendering of minecraft worlds. In *IEEE/CVF International Conference on Computer Vision*, 2021. 2, 3

[23] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics*, 37(6):1–15, 2018. 2

[24] Philipp Henzler, Niloy J Mitra, and Tobias Ritschel. Escaping plato's cave: 3d shape from adversarial rendering. In *IEEE/CVF International Conference on Computer Vision*, pages 9984–9993, 2019. 1

[25] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. 7

[26] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017. 2

[27] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH*, 18(3):165–174, 1984. 1

[28] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 1, 2, 5, 6, 13, 14

[29] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 1, 2, 4, 7, 14

[30] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *ACM Transactions on Graphics*, 37(4):1–14, 2018. 2

[31] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 5

[32] Tejas D Kulkarni, Will Whitney, Pushmeet Kohli, and Joshua B Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, 2015. 2

[33] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics*, 39(6):1–14, 2020. 9

[34] Yiyi Liao, Katja Schwarz, Lars Mescheder, and Andreas Geiger. Towards unsupervised learning of generative models for 3d controllable image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5871–5880, 2020. 1, 2, 3

[35] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *Advances in Neural Information Processing Systems*, 2020. 2

[36] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: learning dynamic renderable volumes from images. *ACM Transactions on Graphics*, 38(4):1–14, 2019. 2

[37] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH*, 21(4):163–169, 1987. 6, 8

[38] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021. 2

[39] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International Conference on Machine Learning*, pages 3481–3490, 2018. 5

[40] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 2

[41] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics*, 38(4):1–14, 2019. 2

[42] 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 *European Conference on Computer Vision*, pages 405–421. Springer, 2020. 1, 2, 4, 8, 14, 19

[43] Thu Nguyen-Phuoc, Chuan Li, Stephen Balaban, and Yong-Liang Yang. Rendernet: A deep convolutional network for differentiable rendering from 3d shapes. In *Advances in Neural Information Processing Systems*, 2018. 2

[44] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3D representations from natural images. In *IEEE/CVF International Conference on Computer Vision*, pages 7588–7597, 2019. 1, 2

[45] Thu Nguyen-Phuoc, Christian Richardt, Long Mai, Yong-Liang Yang, and Niloy Mitra. BlockGAN: Learning 3d object-aware scene representations from unlabelled images. In *Advances in Neural Information Processing Systems*, 2020. 1, 2

[46] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 1, 2, 3, 6, 14

[47] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020. 2, 4

[48] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *IEEE/CVF International Conference on Computer Vision*, 2021. 2, 4

[49] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 2

[50] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Deformable neural radiance fields. In *IEEE/CVF International Conference on Computer Vision*, 2021. 2

[51] Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*, pages 319–345. Springer, 2020. 2

[52] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 2

[53] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3D face model for pose and illumination invariant face recognition. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 296–301, 2009. 7, 13

[54] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9054–9063, 2021. 2

[55] Tiziano Portenier, Qiyang Hu, Attila Szabo, Siavash Arjomand, Paolo Favaro, and Matthias Zwicker. Faceshop: Deep sketch-based image editing. *ACM Transactions on Graphics*, 37(4):1–13, 2018. 2

[56] 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 *IEEE/CVF International Conference on Computer Vision*, pages 2304–2314, 2019. 2

[57] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *Advances in Neural Information Processing Systems*, 2020. 1, 2, 3, 5, 6, 13, 14

[58] Yichun Shi, Divyansh Aggarwal, and Anil K Jain. Lifting 2d stylegan for 3d-aware face generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6258–6266, 2021. 3

[59] 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

[60] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2437–2446, 2019. 2

[61] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, pages 1121–1132, 2019. 2

[62] Tiancheng Sun, Jonathan T Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul E Debevec, and Ravi Ramamoorthi. Single image portrait relighting. *ACM Transactions on Graphics*, 38(4):79–1, 2019. 2

[63] Attila Szabó, Givi Meishvili, and Paolo Favaro. Unsupervised generative 3d shape learning from natural images. *arXiv preprint arXiv:1910.00287*, 2019. 3

[64] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Multi-view 3d models from single images with a convolutional network. In *European Conference on Computer Vision*, pages 322–337, 2016. 2

[65] Justus Thies, Michael Zollhöfer, Christian Theobalt, Marc Stamminger, and Matthias Nießner. Image-guided neural object rendering. In *International Conference on Learning Representations*, 2020. 2

[66] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 551–560, 2020. 2

[67] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018. 2

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

[69] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 14

[70] Weiwei Zhang, Jian Sun, and Xiaoou Tang. Cat head detection-how to effectively exploit shape and texture features. In *European Conference on Computer Vision*, pages 802–816, 2008. 1, 2, 5, 6, 13

[71] Bolei Zhou. Interpreting generative adversarial networks for interactive image generation. *arXiv preprint arXiv:2108.04896*, 2021. 2

[72] 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, 3

[73] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Transactions on Graphics*, 37(4):1–12, 2018. 4

[74] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision*, pages 2223–2232, 2017. 2

[75] Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Josh Tenenbaum, and Bill Freeman. Visual object networks: Image generation with disentangled 3d representations. In *Advances in Neural Information Processing Systems*, volume 31, pages 118–129, 2018. 2

Supplementary Materials

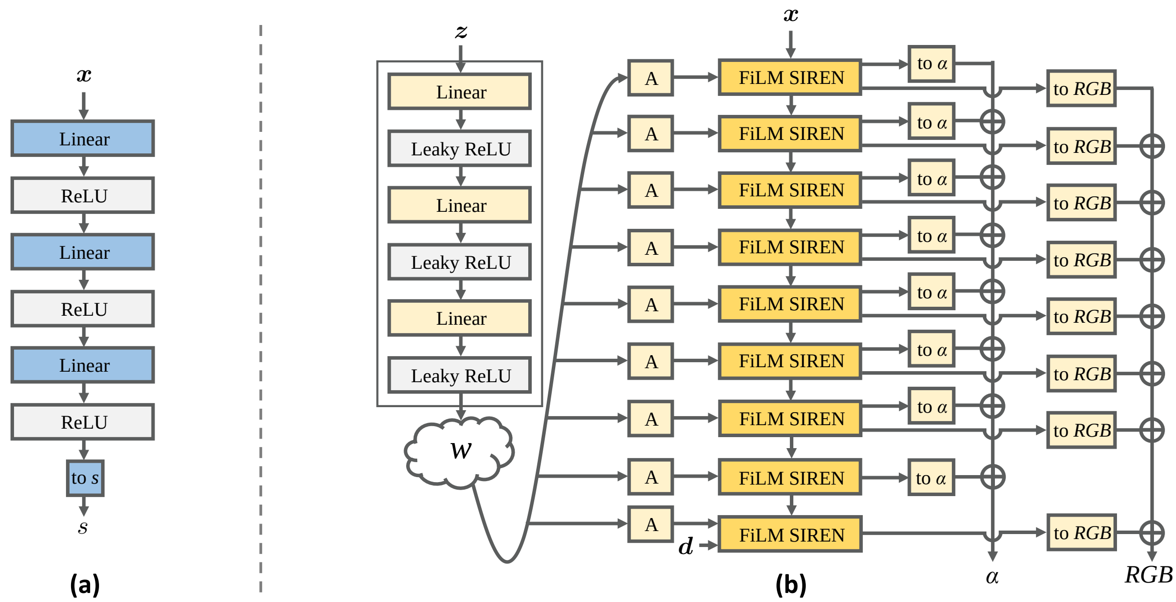

Figure I. Detailed network structures of (a) the manifold predictor $\mathcal{M}$ and (b) the radiance generator Φ.

A. More Implementation Details

A.1. Data Preparation

FFHQ [28]. We align the face images in FFHQ using 5 facial landmarks to centralize the faces and normalize their scales. Specifically, we first detected 5 facial landmarks of the images using an off-the-shelf landmark detector [9]. Then we follow [14] to resize and crop the images by solving a least square problem between the detected keypoints and corresponding 3D keypoints derived from a 3D face model [53]. For pose distribution estimation, the face reconstruction method of [14] is applied to extract the face poses for all the training images. Gaussian distributions are then fitted on the extracted poses, which are defined by the yaw and pitch angles (standard deviation 0.3 radians and 0.15 radians, respectively). During GAN training, we sample camera pose from the distributions and generate images accordingly. The extracted poses also serve as the pseudo labels for the pose regularization term defined in Eq. (9) of the main paper.

Cats [70]. For the cat images, we follow a similar procedure to align and resize the images using landmarks provided by the dataset [70]. We also estimate the camera pose by solving the least square problem between the provided 2D landmarks and a set of manually-selected 3D landmarks on a 3D cat mesh. We found the pose distribution is very close to face images in FFHQ, and thus we simply use the same Gaussian to sample poses during training.

CARLA [16, 57]. We directly resize the car images rendered by [57] to 128^2 resolution without any alignment. Following [11, 57], we uniformly sample camera pose from the upper hemisphere during training.

A.2. Network Structure

Manifold predictor $\mathcal{M}$. Figure I (a) shows the structure of the manifold predictor, which is an MLP with three hidden layers and an output layer. We set the channel dimension of the hidden layers to 128, 64, and 256 for FFHQ, Cats, and CARLA, respectively. These channel dimensions are empirically chosen without careful tuning.

Radiance generator Φ. Figure I (b) shows the detailed structure of the radiance generator, which consists of a mapping network and a synthesis network. The mapping network is an MLP with three hidden layers of dimension 256. The synthesis network consists of 8 FiLM SIREN blocks [11] of dimension 256, and one FiLM SIREN block of dimension 259 which receives an extra view direction as input.

A.3. More Training Details

We jointly learn the manifold predictor $\mathcal{M}$, the radiance generator Φ, and the discriminator D using the losses described in the main paper. Geometric initialization [2] is applied for the weights of $\mathcal{M}$ to obtain sphere-like initial isosurfaces. For FFHQ and Cats, we set the sphere center to $(0, 0, -1.5)$ for human face and cat centered in the $[-1, 1]^3$ cube. For CARLA, we set the center to $(0, 0, 0)$ to obtain hemispherical manifolds, as shown in Fig. 7 of the main paper. The $\{l_i\}$ are set to generate initial isosurfaces evenly positioned across the whole 3D volume. In addition, for FFHQ and Cats, we set the farmost surface to be a fixed plane to represent background. To calculate ray-surface intersections, we uniformly sample 64 points along each ray and calculate the intersections via Eq. (5) in the main paper. The weights of the radiance generator Φ and the discriminator D are initialized following [11].

To enable training at 256^2 resolution, we use PyTorch's Automatic Mixed Precision (AMP) to reduce memory cost. We also use the mini-batch aggregation strategy similar to [11] to ensure a relatively large batch size (16 for 256^2 resolution and 32 for 128^2 resolution) during training. We train GRAM for 120K iterations, 80K iterations, and 70K iterations on FFHQ, Cats, and CARLA, respectively.

A.4. Image Embedding Details

Given a real image I, we freeze the weights of the generator G, and optimize the frequencies $\boldsymbol{\gamma}$ and phase shifts $\boldsymbol{\beta}$ for each FiLM SIREN block to generate an image $I_{gen} = G_{syn}(\boldsymbol{\gamma}, \boldsymbol{\beta})$ that best matches the input image. To achieve this, we use an objective function consisting of several terms:

$$\begin{aligned}\mathcal{L}_{emb} =& ||I - I_{gen}||^2 + (1 - < f_{id}(I), f_{id}(I_{gen}) >) \\ &+ \text{LPIPS}(I, I_{gen}) + ||\boldsymbol{\gamma} - \bar{\boldsymbol{\gamma}}||^2 + ||\boldsymbol{\beta} - \bar{\boldsymbol{\beta}}||^2,\end{aligned} \tag{I}$$

where f_{id} is the identity feature extracted from a face recognition network [12], and $\text{LPIPS}(\cdot, \cdot)$ is the perceptual loss from [69]. $\bar{\boldsymbol{\gamma}}$ and $\bar{\boldsymbol{\beta}}$ are average frequencies and phase shifts calculated using 10K random samples. We also initialize $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ with the average values. We use the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is set to 4×10^{-3}, and we optimize $\bar{\boldsymbol{\gamma}}$ and $\bar{\boldsymbol{\beta}}$ for 20K iterations. After optimization, we can freely move the camera to synthesize an image at novel views.

B. More Results

B.1. Qualitative Results

Figure II, III, and IV show more visual results of GRAM. Our method can generate realistic images with strong multiview consistency. Animation results can be found on the *project page*.

B.2. Comparisons

More comparisons with previous methods. Figure V shows more visual comparisons between GRAM and the previous 3D-aware image generation methods [11, 46, 57]. Our method achieves the best result in terms of image quality and 3D consistency. Animations can be found on the *project page*.

More comparisons with NeRF-H sampling. Figure VI shows the visual comparisons between our manifold sampling strategy and the original NeRF-H [11, 42] sampling strategy. Our method achieves better visual quality with finer details. More importantly, NeRF-H fails to learn reasonable 3D structures of the generated instances with a number of sampling points fewer than 12. It still produces undesired artifacts (*e.g.*, the concave forehead geometry which creates hollow-face illusion), even trained with 48 sampling points. In contrast, our method can learn reasonable 3D geometry with as few as 6 points (surfaces). We hardly observe the concave forehead issue for the generated instances in our cases.

B.3. Camera Zoom

As shown in Fig. VII, GRAM can generate reasonable results with camera zoom-in and zoom-out effects. Animations can be found on the *project page*.

B.4. Latent Space Interpolation

We show the results of latent code interpolation in Fig. VIII. The continuous semantic changes between adjacent images demonstrate the reasonable latent space learned by GRAM.

B.5. Style Mixing

Figure IX shows the style mixing results between source subjects and target subjects. Similar to [28, 29], styles in shallower layers (layer 1 to 5) of GRAM mainly control geometry, while styles in deeper layers (layer 6 to 9) control appearance. Note that our method is not trained with the style mixing strategy.

B.6. Image Embedding and Editing

Animations of the image editing results can be found on the *project page*. We achieve pose control of the embedded images and well maintain the 3D consistency even for fine details.

Figure II. Multiview generation results of GRAM on FFHQ.

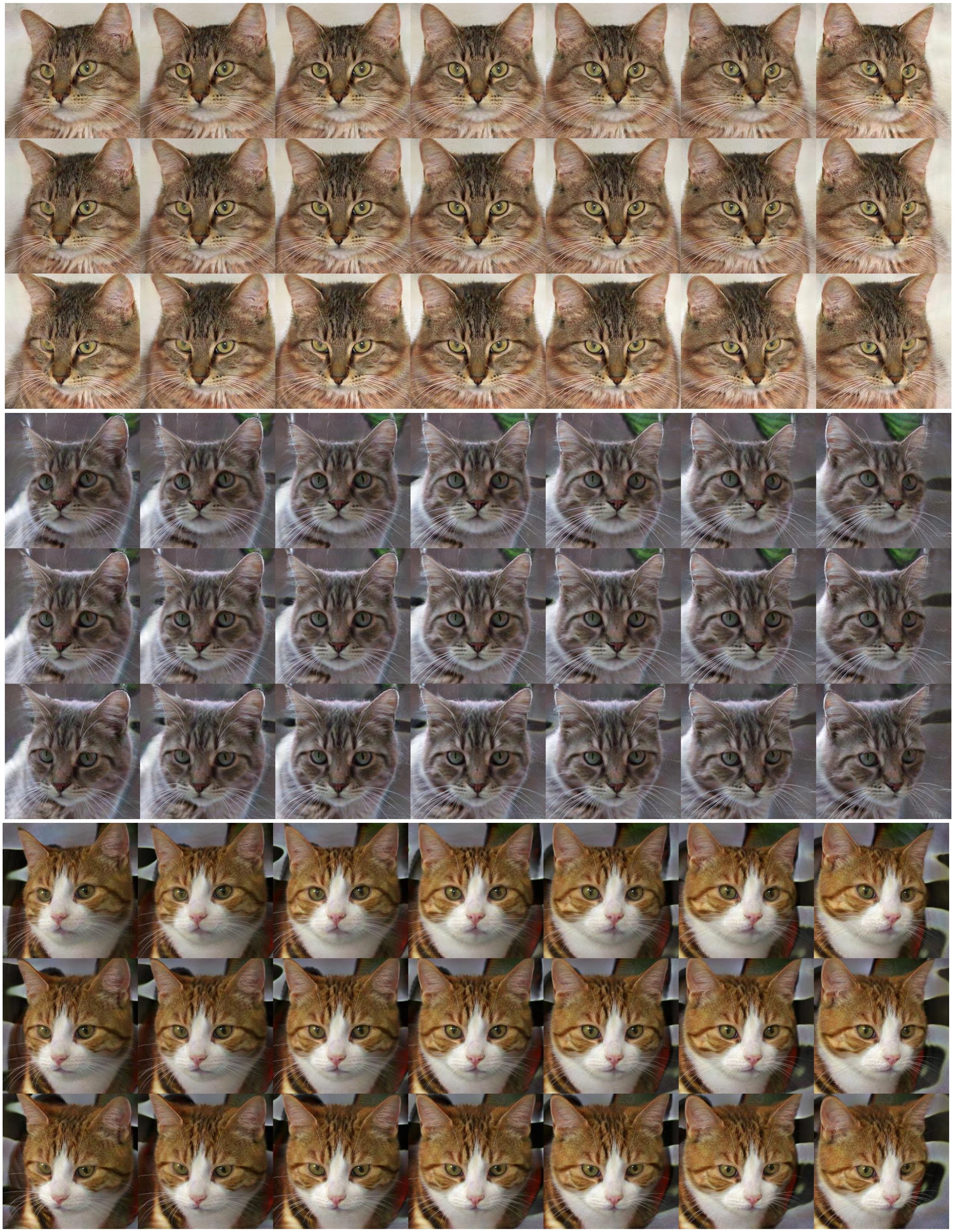

Figure III. Multiview generation results of GRAM on Cats.

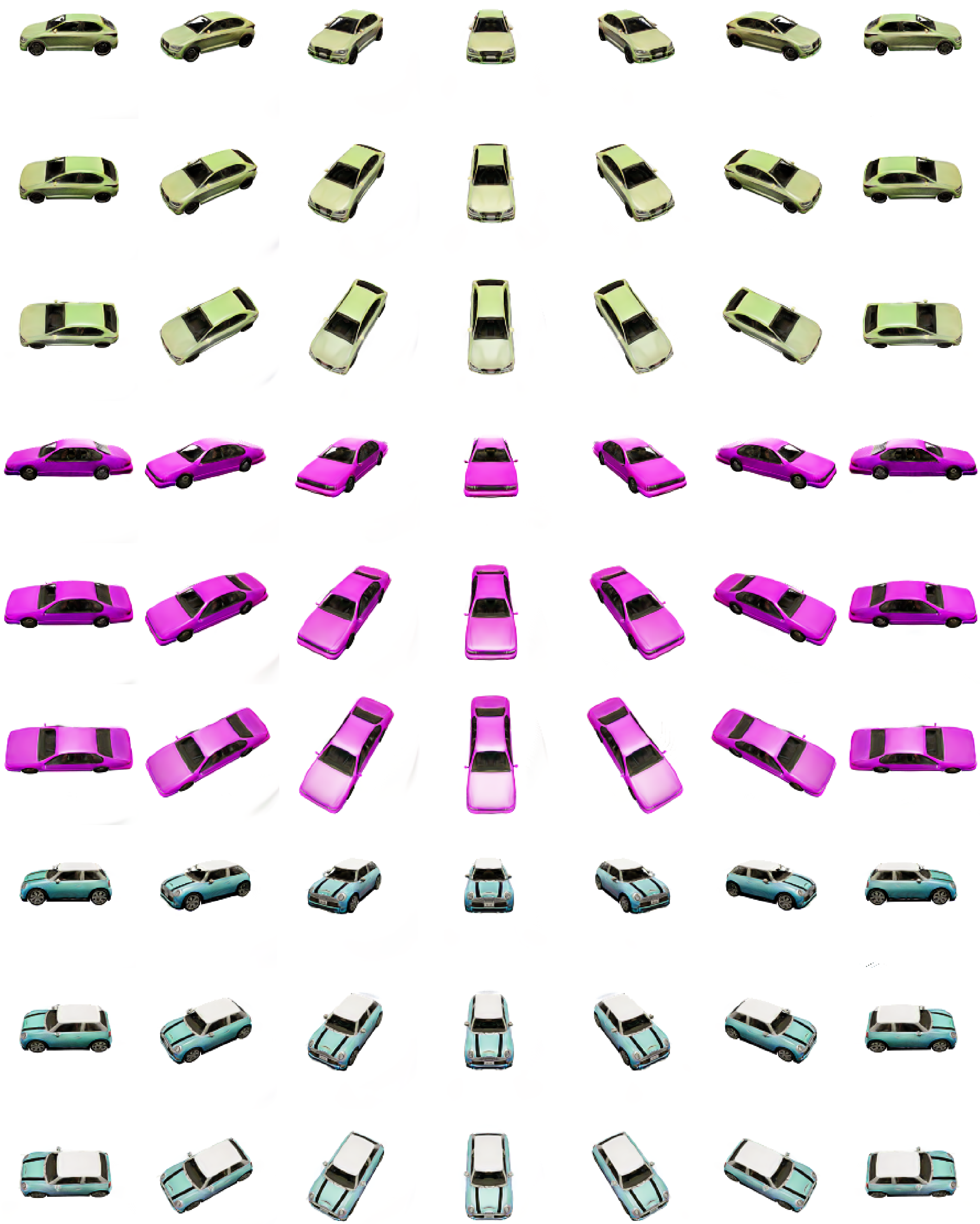

Figure IV. Multiview generation results of GRAM on CARLA.

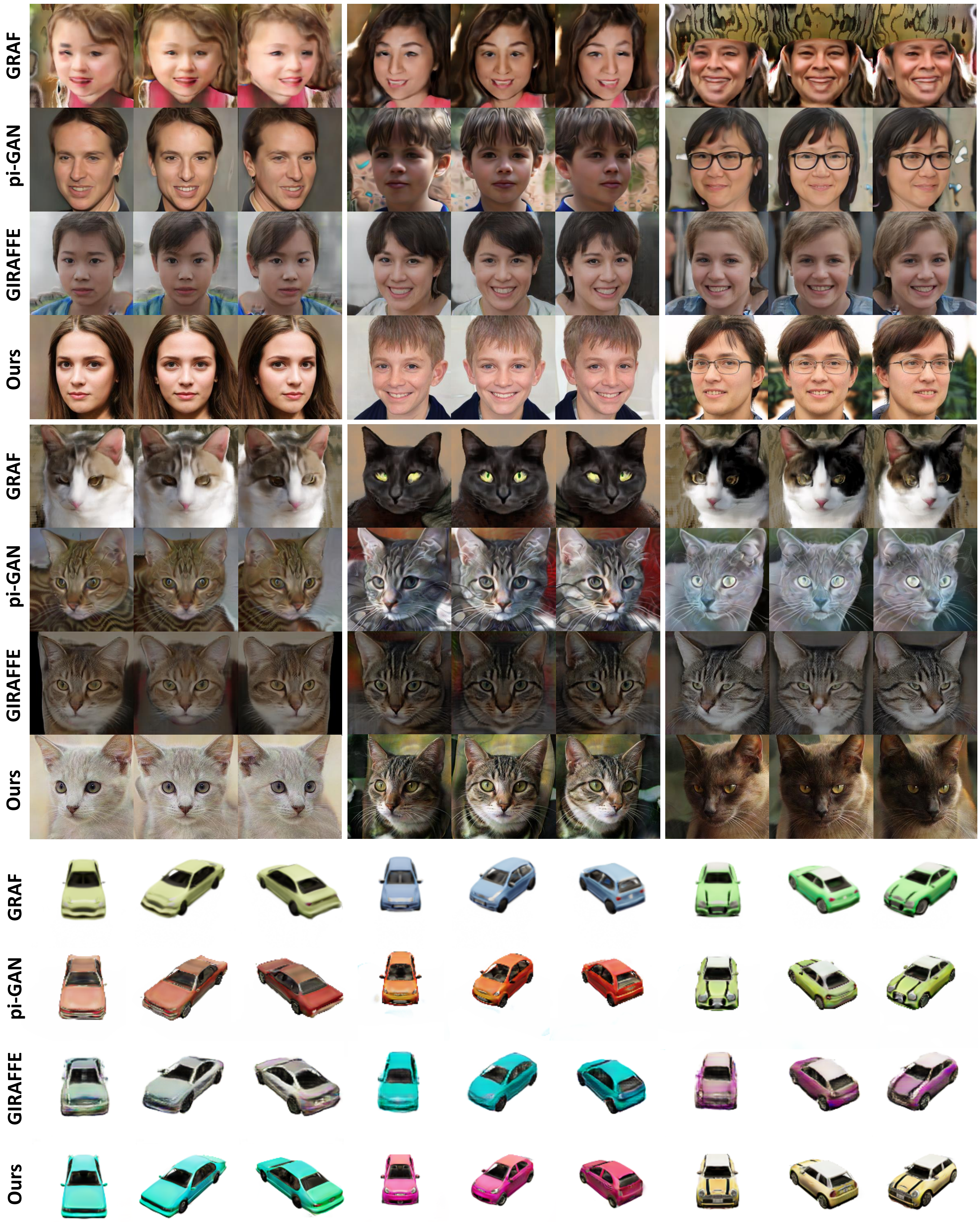

Figure V. More qualitative comparisons with previous 3D-aware image generation methods on three datasets.

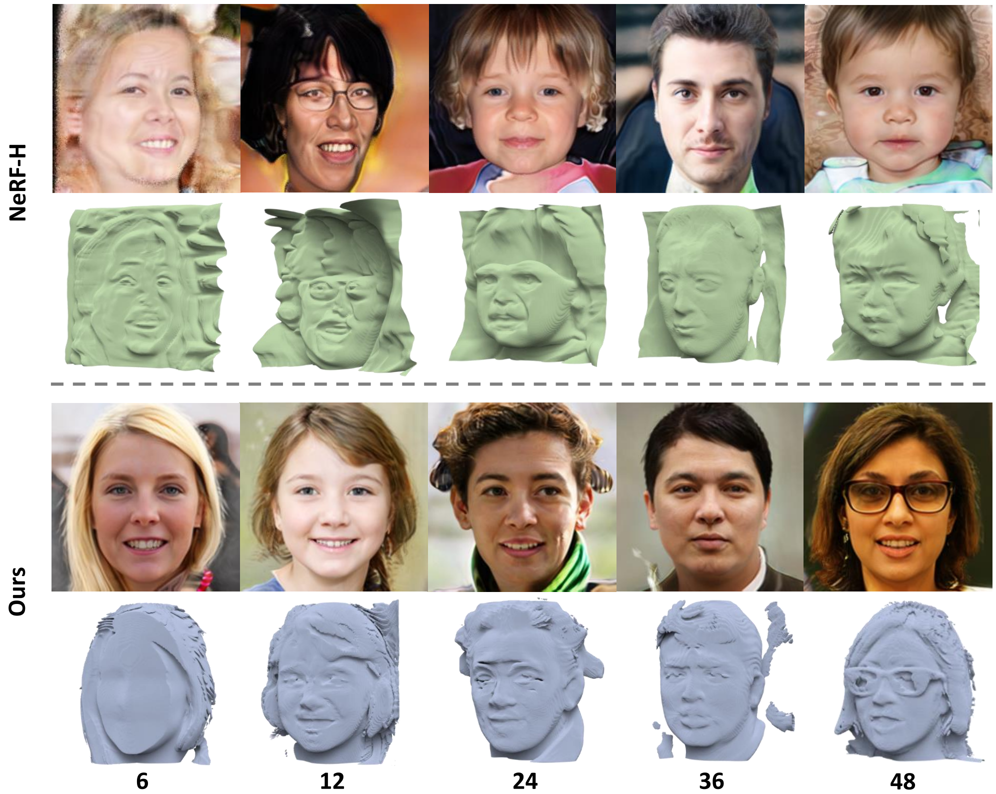

Figure VI. Comparison between our manifold sampling and NeRF-H [11, 42] sampling strategy.

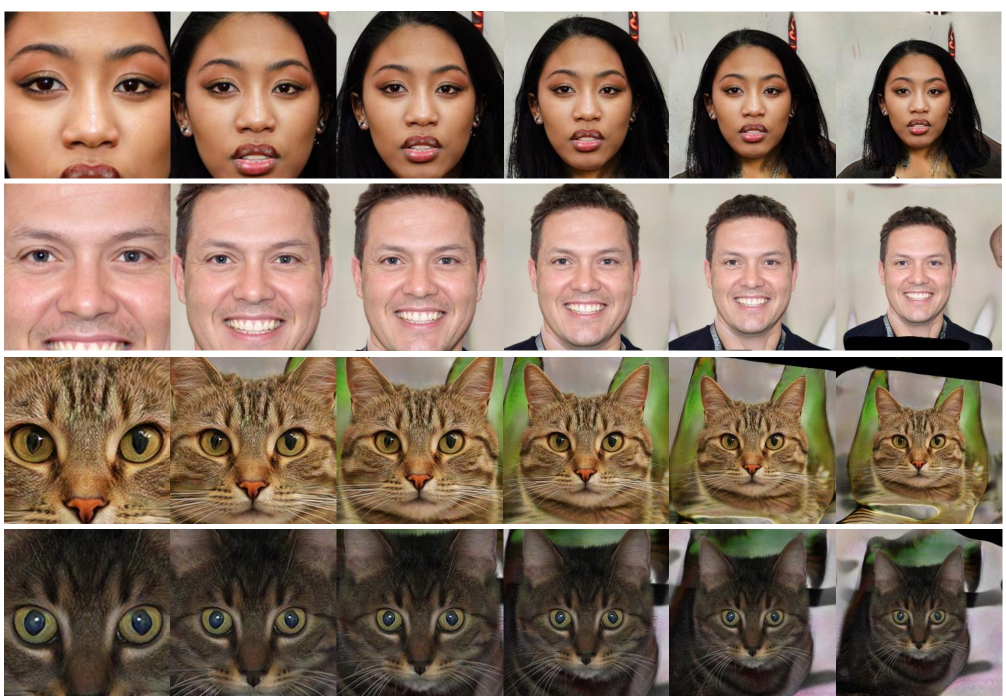
Figure VII. Generation results under camera zoom-in and zoom-out.

Figure VIII. Latent space interpolation results.

Figure IX. Style mixing between different generated subjects. Note that our method is not trained with the style mixing strategy.