GRAM: Generative Radiance Manifolds for 3D-Aware Image Generation
(*Supplementary Material*)

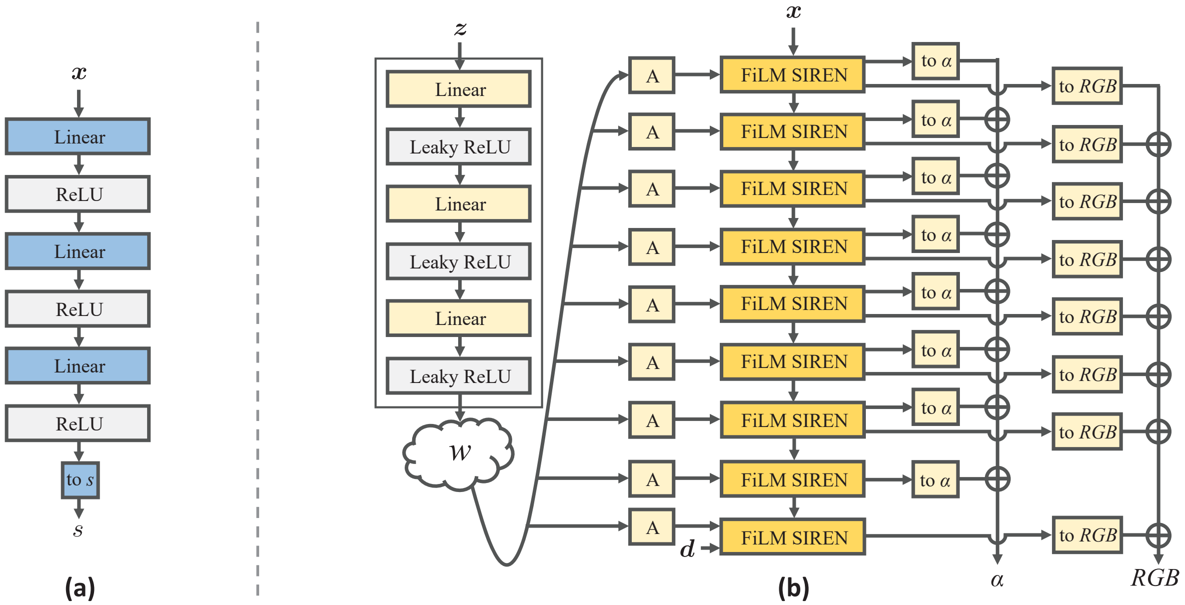

Figure I. Detailed network structures of (a) the manifold predictor $\mathcal{M}$ and (b) the radiance generator Φ.

I. More Implementation Details

I.1. Data Preparation

FFHQ [7]. 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 [2]. Then we follow [5] 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 [11]. For pose distribution estimation, the face reconstruction method of [5] 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 [14]. For the cat images, we follow a similar procedure to align and resize the images using landmarks provided by the dataset [14]. 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 [6, 12]. We directly resize the car images rendered by [12] to 128^2 resolution without any alignment. Following [3, 12], we uniformly sample camera pose from the upper hemisphere during training.

I.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 [3] of dimension 256, and one FiLM SIREN block of dimension 259 which receives an extra view direction as input.

I.3. More Training Details

During training, we randomly sample latent code $\boldsymbol{z}$ from the normal distribution and camera pose $\boldsymbol{\theta}$ from the known or estimated distributions of the training datasets. 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 [1] 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 [3].

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 [3] 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. Training took 3 to 7 days depending on the dataset and image resolution.

I.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 [4], and LPIPS$(\cdot,\cdot)$ is the perceptual loss from [13]. $\bar{\boldsymbol{\gamma}}$ 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.

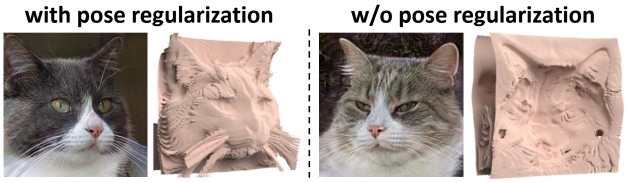

Figure II. Learned 3D geometry with and w/o pose regularization.

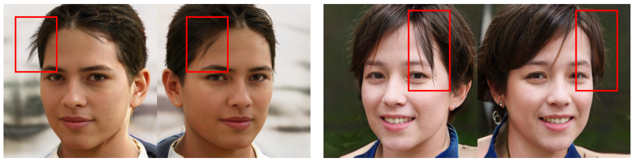

Figure III. Exaggerated parallax artifacts on generated subjects.

II. More Results

II.1. Qualitative Results

Figure IV, V, and VI 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*.

II.2. Comparisons

More comparisons with previous methods. Figure VII shows more visual comparisons between GRAM and the previous 3D-aware image generation methods [3, 10, 12]. 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 VIII shows the visual comparisons between our manifold sampling strategy and the original NeRF-H [3,9] 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.

II.3. Failure Cases

Concave geometry. We empirically found that for cats, dropping pose regularization sometimes led to unstable training and yielded wrong pose and geometry (which is known as the "hollow-face illusion"; see Fig. II). Training on faces and cars were quite stable no matter pose regularizations were used or not.

Exaggerated parallax artifacts. When varying camera poses, some contents (*e.g.* hair fringes) on certain generated subjects could float away from their expected positions, as shown in Fig. III. This is due to that the fixed and limited number of surface manifolds across the whole category cannot provide accurate depth for all structures on every single subject. The problem could be alleviated when using instance-specific surfaces, which we will explore in future works.

II.4. Camera Zoom

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

II.5. Latent Space Interpolation

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

II.6. Style Mixing

Figure XI shows the style mixing results between source subjects and target subjects. Similar to [7, 8], 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.

II.7. 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.

References

[1] 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. 2

[2] 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. 1

[3] 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, 8

[4] 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. 2

[5] 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. 1

[6] 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. 1

[7] 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, 3

[8] 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. 3

[9] 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. 2, 8

[10] 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. 2

[11] 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. 1

[12] 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

[13] 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. 2

[14] 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

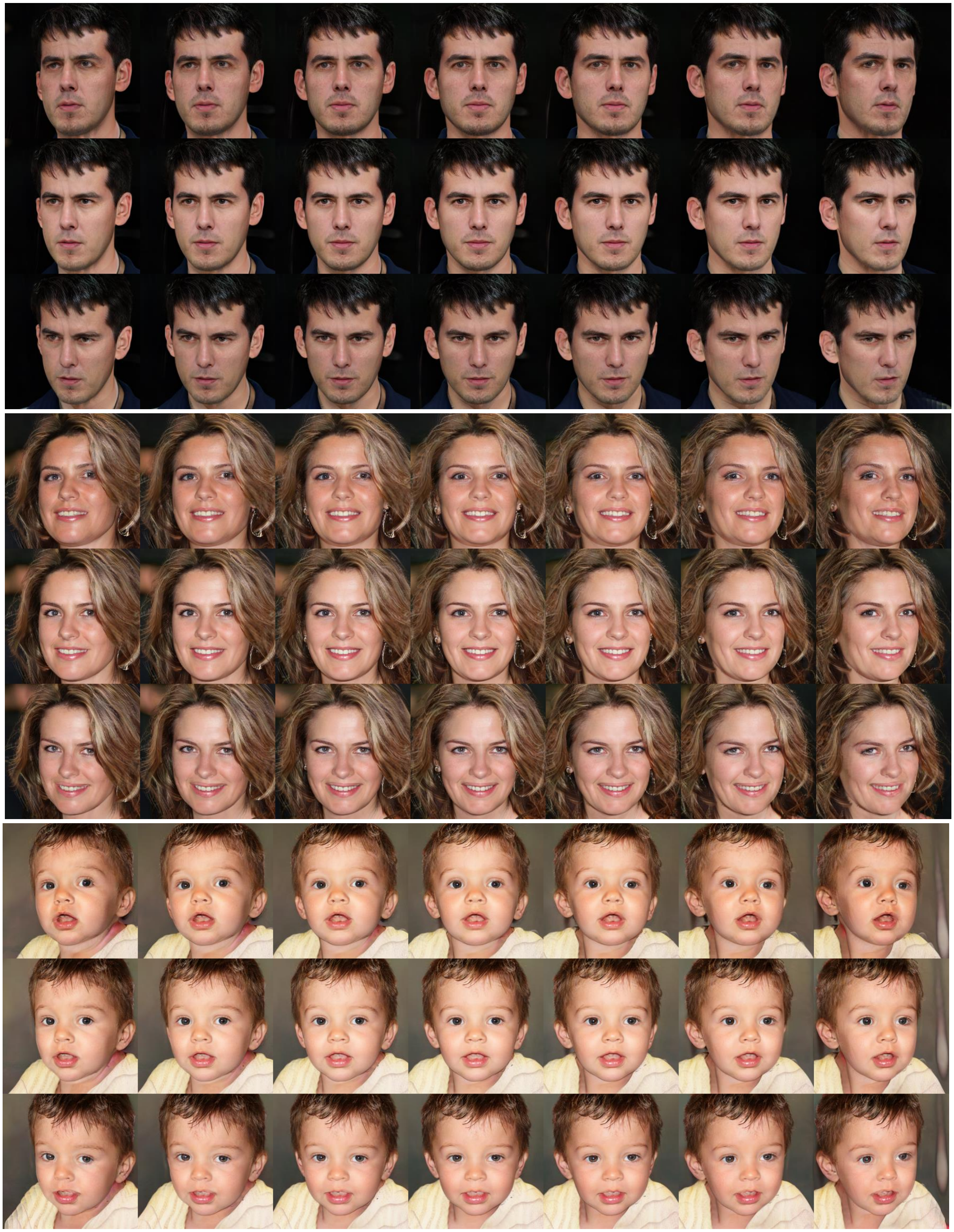

Figure IV. Multiview generation results of GRAM on FFHQ.

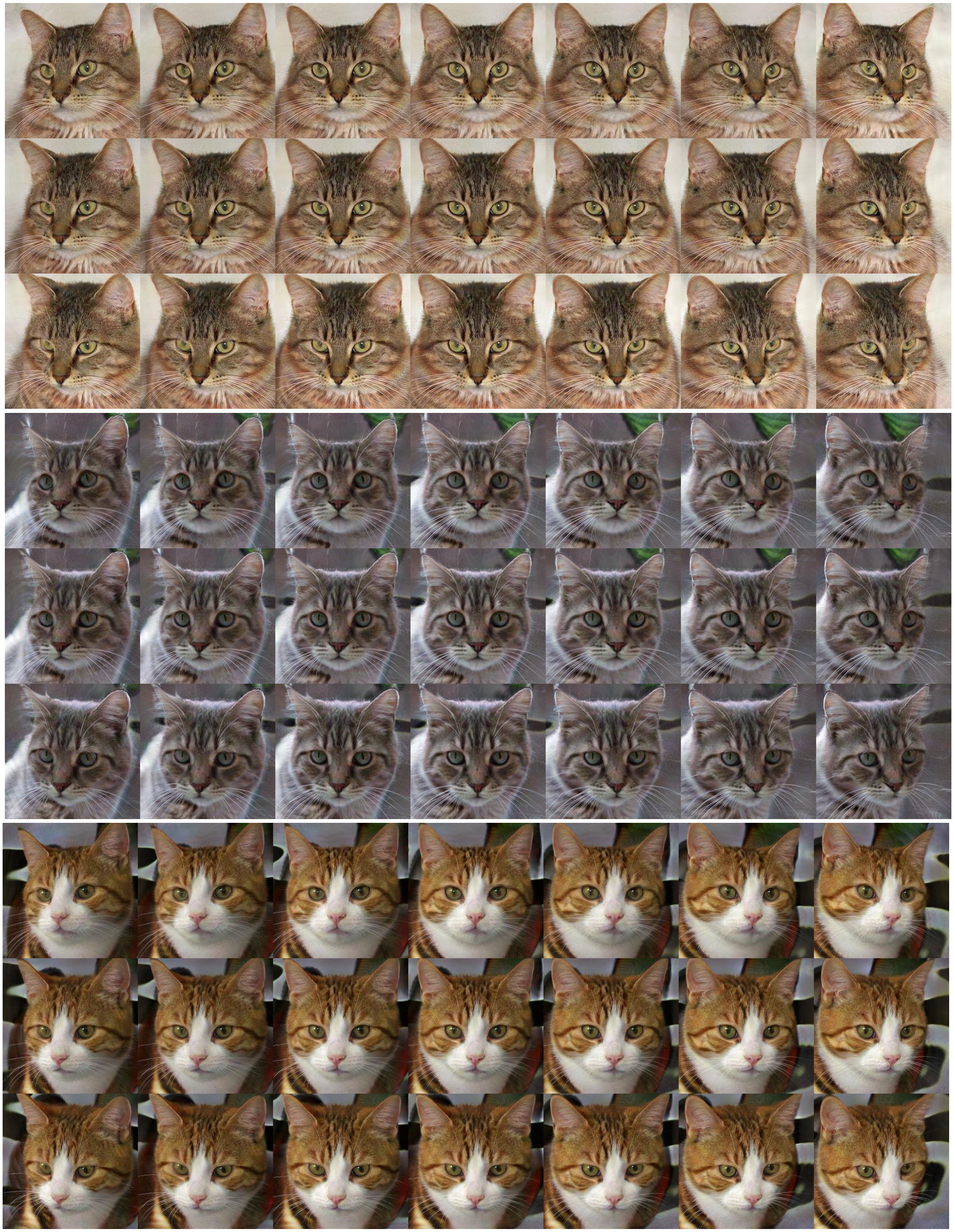

Figure V. Multiview generation results of GRAM on Cats.

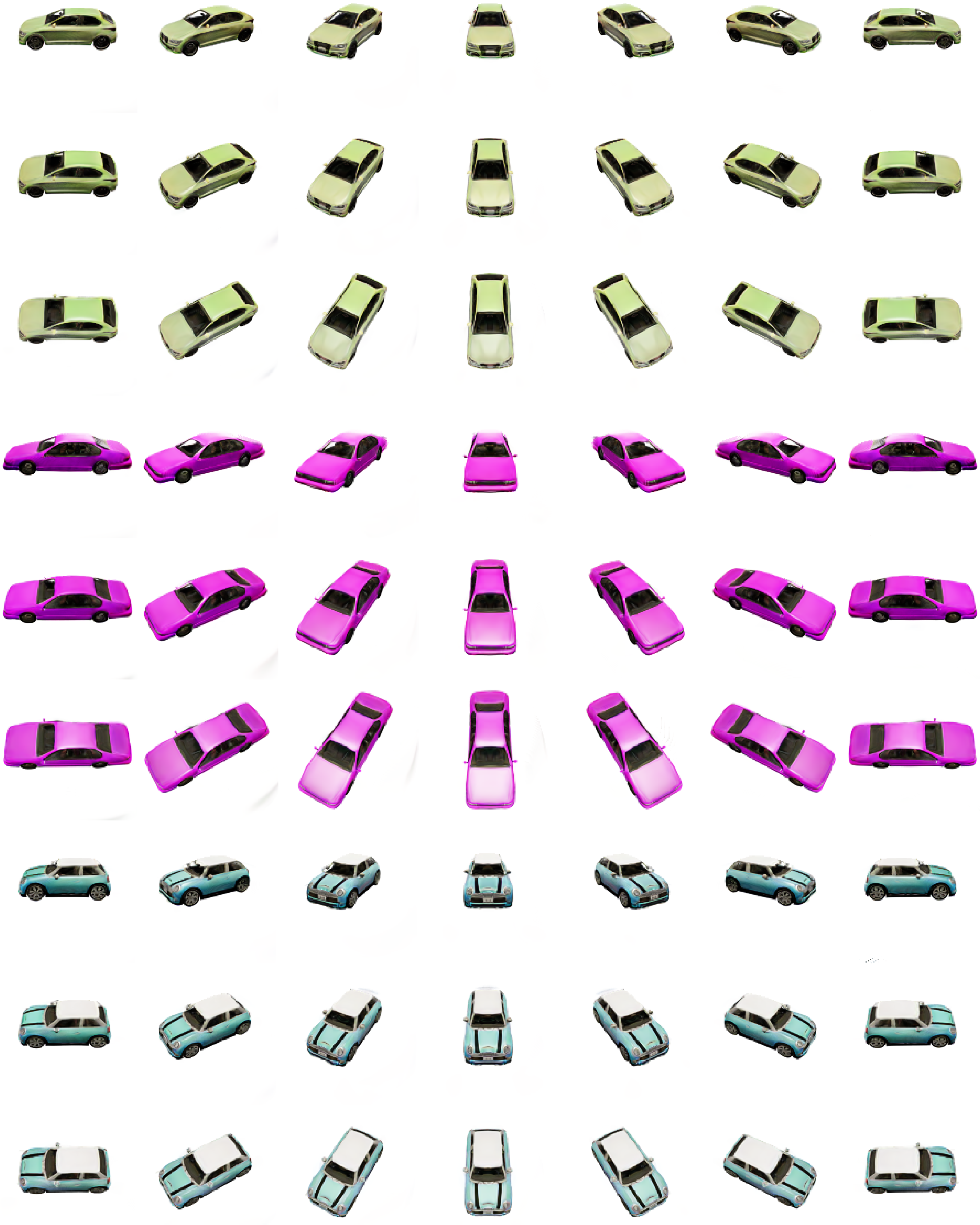

Figure VI. Multiview generation results of GRAM on CARLA.

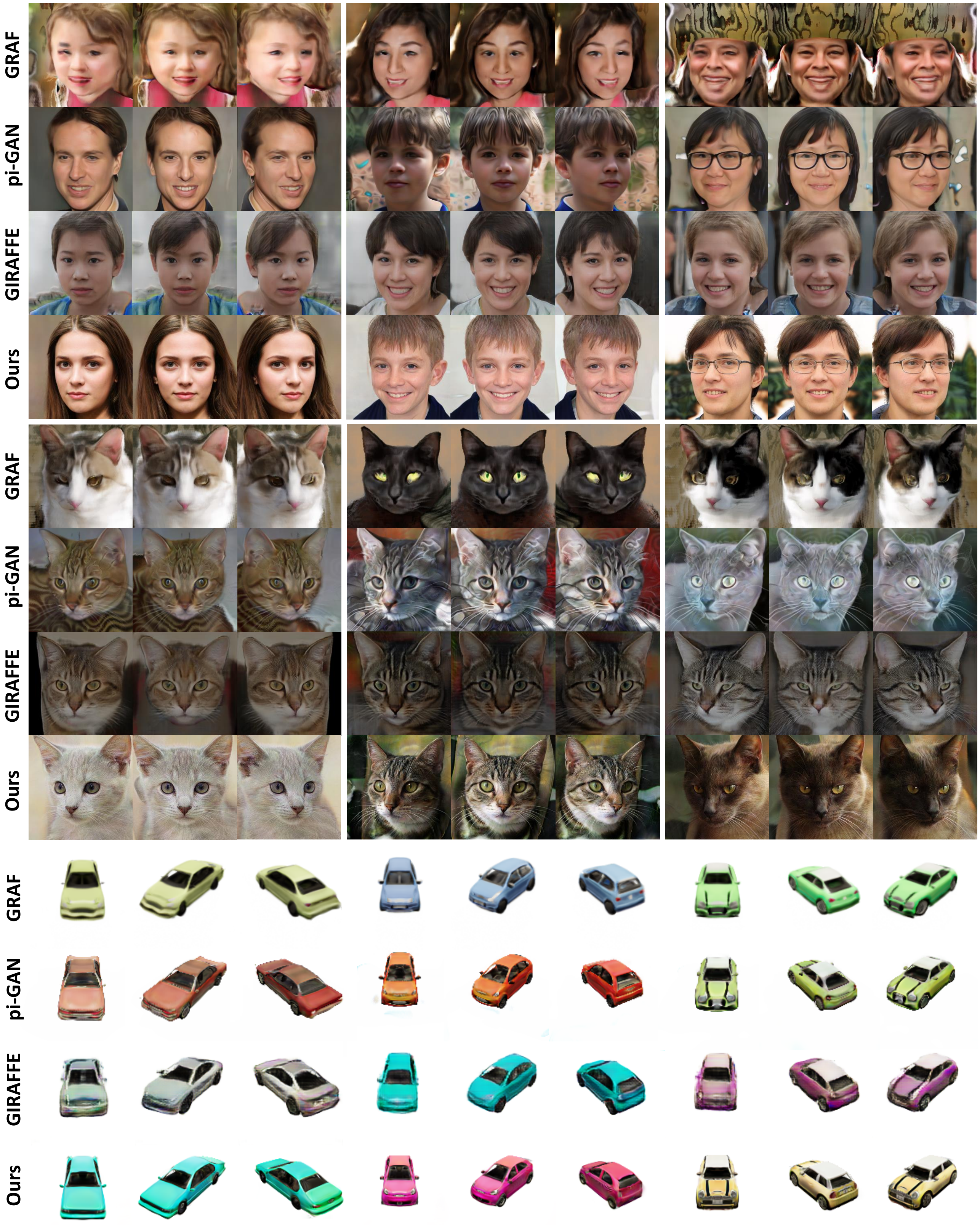

Figure VII. More qualitative comparisons with previous 3D-aware image generation methods on three datasets.

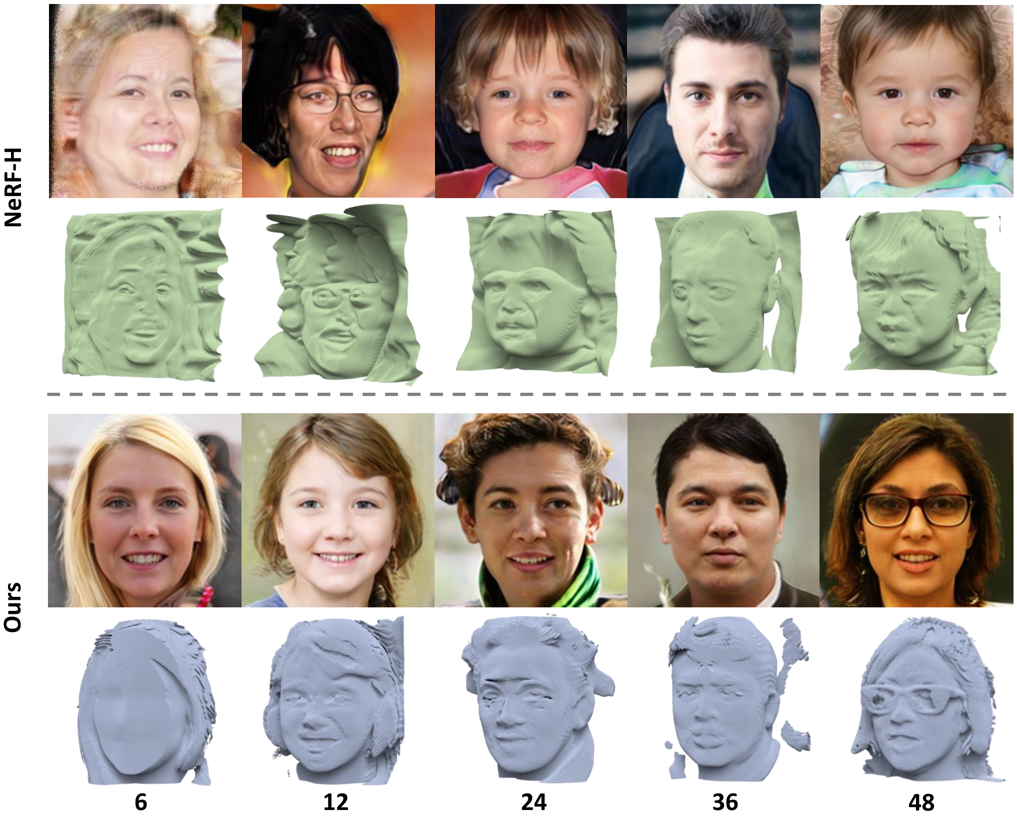

Figure VIII. Comparison between our manifold sampling and NeRF-H [3, 9] sampling strategy.

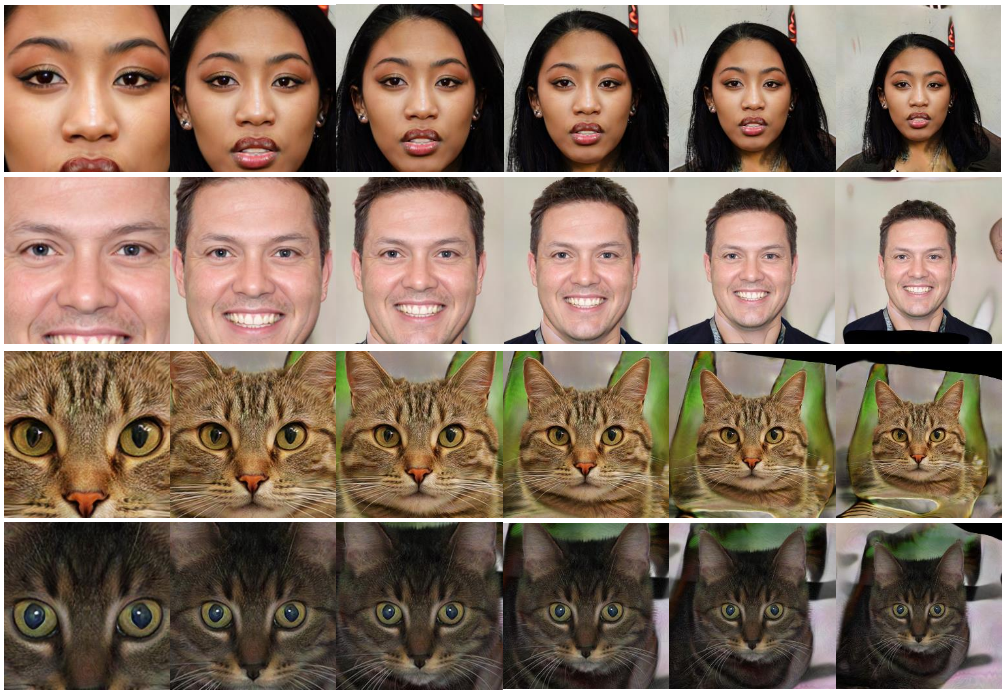

Figure IX. Generation results under camera zoom-in and zoom-out.

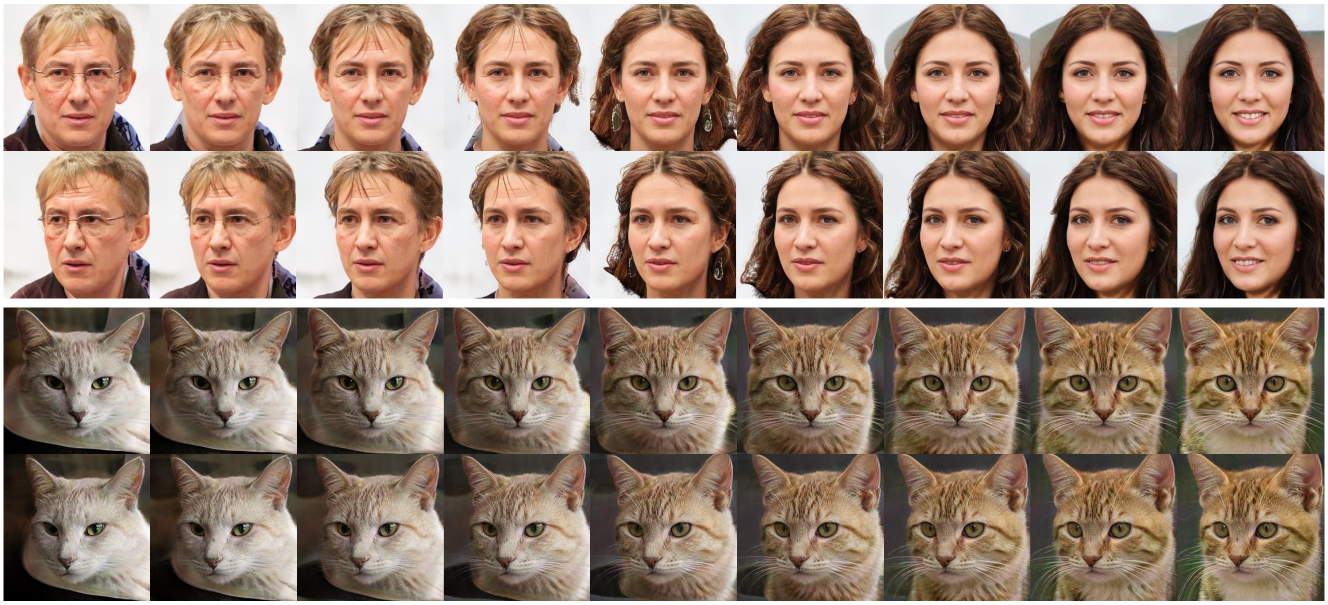

Figure X. Latent space interpolation results.

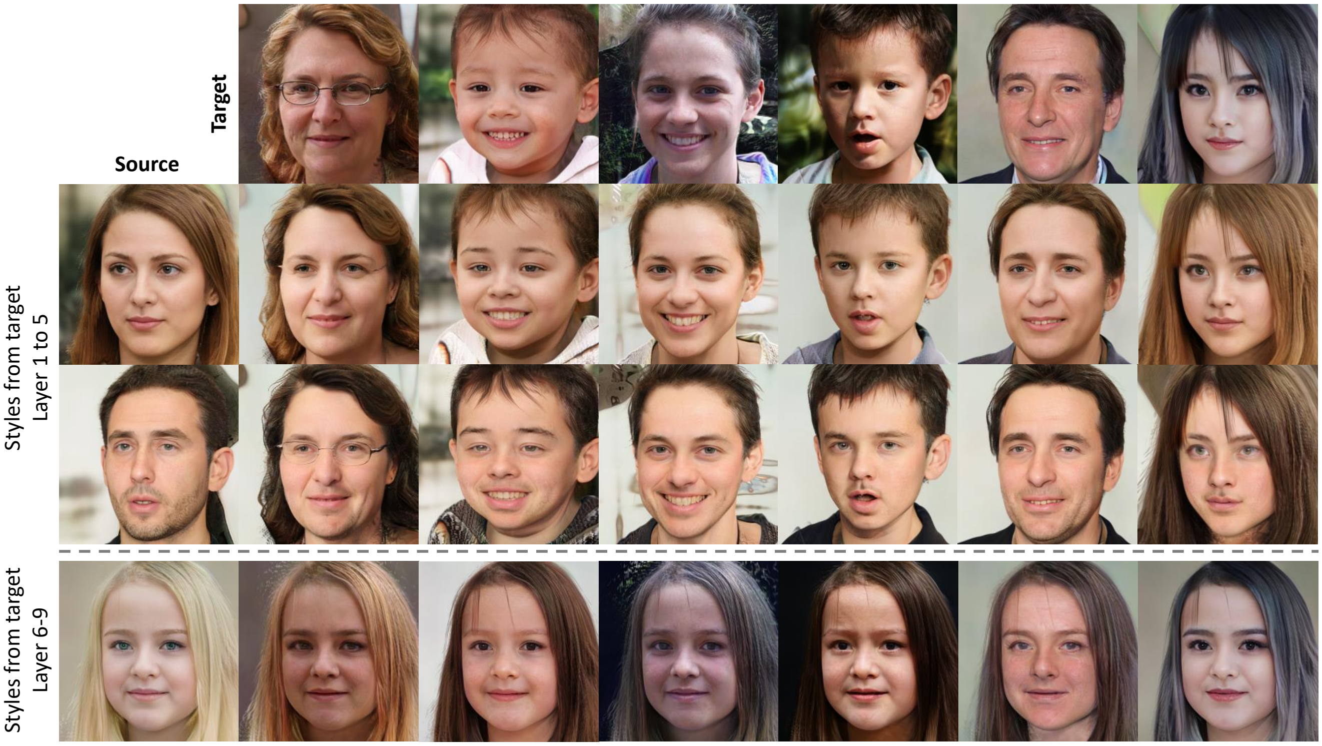

Figure XI. Style mixing between different generated subjects. Note that our method is not trained with the style mixing strategy.