GRAM-HD: 3D-Consistent Image Generation at High Resolution with Generative Radiance Manifolds
(Supplementary Material)

A. Network architecture

Radiance manifold generator As shown in Figure 1 (left), the network architecture of our radiance manifold generator is the same as GRAM [6], except that we additional apply several fully-connected layers with skip connections to extract intermediate features. These layers project the 256-dimension hidden features of the FiLM SIREN MLP [3] into 32-dimension intermediate features which are used as the input for the super-resolution module.

Super-resolution module Figure 1 (right) shows the detailed structure of our super-resolution module. For LR feature processing, we apply 8 RRDB blocks, each having 64 channels. We then apply 4 sub-pixel convolution layers with 64, 64, 32, and 16 channels respectively for upsampling to 1024^2 resolution. Finally, a 16-channel convolution layer and a 4-channel projection layer are applied to produce the color and occupancy. Besides, a mapping MLP with three 256-dimension hidden layers maps the latent code to the style code. Each conv layer after the RRDBs is modulated by an affine-transformed style code.

B. More implementation details

B.1. Data preparation

We align the images in FFHQ [9] and AFHQv2-CATS [4] using detected landmarks. Specifically, we first detect landmarks of the images (5 landmarks for FFHQ and 9 for AFHQv2-CATS) using of-the-shelf landmark detectors [1, 11]. We then resize and crop the images by solving a least-square fitting problem between the detected landmarks and a set of predefined 3D keypoints, following the strategy of [7]. The 3D keypoints for human face are derived from the mean face of a 3D parametric model [14], while for cat they are some manually-selected vertices on a cat head mesh downloaded from the Internet.

We extract camera poses for the images in the datasets, which are used to estimate the prior camera pose distributions and serve as the pseudo labels for the pose loss term following [6]. For FFHQ, the face reconstruction method of [7] is employed for pose estimation. For AFHQv2-CATS, the estimated angles are obtained via solving the aforementioned least-square fitting. Then, we fit Gaussian distributions on yaw and pitch angles as prior pose distributions. The standard deviations for yaw and pitch angles are $(0.3, 0.15)$ and $(0.18, 0.15)$ for FFHQ and AFHQv2-CATS, respectively.

B.2. More training details

During training, we randomly sample latent code $\boldsymbol{z}$ from the normal distribution and camera pose $\boldsymbol{\theta}$ from the estimated prior distributions of the datasets. A two-stage training strategy is applied as mentioned in the main paper.

At the first training stage, we initialize the radiance manifold generator following [6]: 23 evenly distributed sphere manifolds centered at $(0, 0, -1.5)$ covering 3D objects inside of the $[-1, 1]^3$ cube and an additional background plane at $z = -1$ are initialized. The learning rates are set to $2\mathrm{e}-5$ for the radiance manifold generator and $2\mathrm{e}-4$ for the LR discriminator.

At the second stage, we freeze the radiance manifold generator and only optimize the super-resolution module (and the newly-added transformation layers for intermediate features). The HR discriminator is trained from scratch without progressive growing. The learning rates are set to $2\mathrm{e}-4$ for both the super-resolution module and the HR discriminator.

For all the training processes, we use the Adam [10] optimizer with $\beta_1 = 0$ and $\beta_2 = 0.9$. Batch size is set to 32 in the first training stage and 32, 16, 16 in second stage for 256^2, 512^2, 1024^2, respectively. We train our model for 100K iterations on FFHQ and 40K iterations on AFHQv2-CATS since it is a relatively small dataset. Training took 3 to 7 days depending on the dataset and resolution.

Background super-resolution details For the last surface manifold, *i.e.,* the background plane, we use a larger projection view during manifold gridding, as it spans a much wider region for covering the background under extreme views. A nonlinear mapping is applied to do the sam-

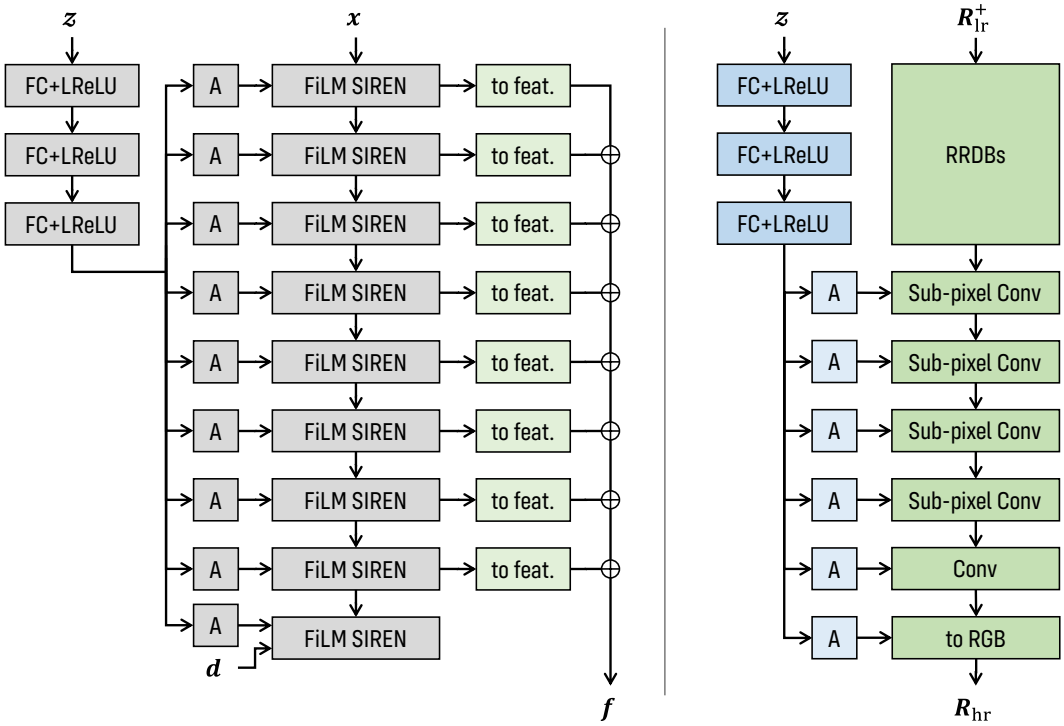

Figure 1: Network architecture. The components in gray color are from GRAM [6] and others are newly introduced in our GRAM-HD. **Left:** The architecture of the radiance manifold generator. The intermediate features are transformed by simply fully connected layers and accumulated to be the input for the super-resolution CNN. The RGBα prediction branches and the manifold predictor network are the identical to GRAM and omitted here for brevity. **Right:** The architecture of the super-resolution module with several Residual-in-Residual Dense Blocks (RRDBs) [18] and sub-pixel convolutional layers [15] as backbone.

pling. Specifically, we first uniformly sample xy coordinates within $[-1, 1]^2$ and then apply the following nonlinear mapping function:

$$\text{BGTrans}(x) = \begin{cases} 2\tan(x + 0.5) - 1 & x < -0.5 \\ 2x & -0.5 \le x \le 0.5. \\ 2\tan(x - 0.5) + 1 & x > 0.5 \end{cases} \tag{1}$$

The purpose of this transformation is to enlarge the sampling region and sample denser points around center. Radiance are calculated at these sampled and transformed coordinates and form the radiance map for super-resolution. A small independent super-resolution CNN is applied since the radiance distribution on the background significantly differs from the foreground.

B.3. 3D-consistency metric details

To quantitatively evaluate 3D consistency, we use the reconstruction quality of a recent surface-based multiview reconstruction method - NeuS [17]. Specifically, for each method, we first randomly generate 50 instances. For each instance, we render 30 images with yaw angle evenly sampled from -0.4 radian to 0.4 radian, and train a NeuS model with these images as input. The mean PSNR and SSIM scores of the reconstructed images by NeuS are used as the quantitative metrics. In theory, the more consistent the input mutlview images are, the higher the reconstruction quality will be. For NeuS training, we use the official implementation with default settings.

B.4. Shape extraction details

We employ a multiview depth fusion method to extract shapes at high resolution. For a given view, the depth map can be calculated by:

$$\begin{aligned} d(\boldsymbol{r}) &= \sum_{i=1}^{N} T(\boldsymbol{x}_i)\alpha(\boldsymbol{x}_i)z(\boldsymbol{x}_i) \\ &= \sum_{i=1}^{N} \prod_{j<i} (1 - \alpha(\boldsymbol{x}_j))\alpha(\boldsymbol{x}_i)z(\boldsymbol{x}_i), \end{aligned} \tag{2}$$

where $\boldsymbol{r}$ is a viewing ray, $\boldsymbol{x}_i$ are the point samples, *i.e.*, ray-manifold intersections, and $z(\cdot)$ denotes the projected depth. We then calculate a discrete occupancy field on a 3D sampling grid. Specifically, for each point $\boldsymbol{x}_\mathrm{s}$ on the sampling grid, we project it to the depth map and calculate its occupancy as $\alpha = \mathrm{Sigmoid}\big(k\big(z(\boldsymbol{x}_\mathrm{s}) - d\big)\big)$ where k is a scaling factor we set to 10. We average the occupancy from 15 different views, and run MarchingCube [12] to extract the shape.

B.5. Image embedding details

Given a target image I_t, we freeze the weights of the generator and optimize the style code w_i for each modulated layer to generate an image I_g that best matches the target image. The following objective function is used:

$$\begin{aligned}\mathcal{L}_\mathrm{emb} = &\|I_\mathrm{g} - I_\mathrm{t}\|^2 + \mathrm{LPIPS}(I_\mathrm{g}, I_\mathrm{t}) \\ &+ (1 - \langle f_\mathrm{id}(I_\mathrm{g}), f_\mathrm{id}(I_\mathrm{t})\rangle) \\ &+ \sum_i \|w_i - \bar{w}\|^2 + \sum_j \|\sigma_j^\mathrm{d}\|^2,\end{aligned} \tag{3}$$

where f_id is an identity feature extractor [5], LPIPS is a perceptual loss from [19], $\bar{w}$ is the precomputed mean style code, and $\sigma_j^\mathrm{d} = \mathrm{sqrt}\big(\sum_{i=1}^N T(\boldsymbol{x}_i)\alpha(\boldsymbol{x}_i)z^2(\boldsymbol{x}_i) - d^2(\boldsymbol{r}_j)\big)$ is the standard deviation of depth along each viewing ray $\boldsymbol{r}_j$. The style and depth regularizations are added to avoid overfitting. With the Adam [10] optimizer, we first run the optimization on low resolution for 200 steps and then switch to the high resolution for another 5000 steps.

C. More experimental results

C.1. More Qualitative results

Figure 2 and 3 present the uncurated generation results of GRAM-HD. Figure 4 and 5 further show the multiview images of some generated instances. Our method can generate realistic images at high resolution with strong 3D consistency.

C.2. More Comparisons

In this section, we provide more comparisons of geometry details and 3D consistency between our method and StyleNeRF [8], StyleSDF [13], EG3D [2], EpiGRAF [16] and GMPI [20].

Visual comparison of geometry details In Figure 6, we show more samples from different methods with some thin geometry structure highlighted. As we can see from the figures, almost all other methods generate some artifacts around eyeglass for human face and whiskers for cats. In particular, all these method failed to generate reasonable whiskers of cats: the generated cat whiskers are stuck onto the cat faces instead of floating naturally in the front. In contrast, our method can generate highly-realistic results for such thin structures.

EPI comparison In Figure 7, we show more EPI-like texture images to demonstrate GRAM-HD's superiority on 3D consistency. The textures of StyleNeRF, StyleSDF and EG3D are either distorted or stuck to image coordinates, indicating different types of inconsistency. Although EpiGRAF uses a pure 3D representation without image-space upsampler, there are still some noise on its generated textures as the Monte-Carlo volume rendering is not noise-free. The textures from our method and GMPI are smooth and natural, demonstrating their superior 3D consistency.

C.3. Latent space interpolation

Figure 8 shows the results of latent space interpolation with GRAM-HD. We select generated instances of different gender, skin color, age, *etc.*, and then show the results by linearly interpolating their latent codes. The meaningful intermediate results and smooth changes demonstrate the reasonable latent space learned by GRAM-HD.

C.4. Style mixing

We further tested style mixing [9] with GRAM-HD, and the results are shown in Figure 10. By combining styles from the source and target instances in different layers, it is found that styles in shallower layers (layer 1 to 5 in radiance manifold generator) mainly control geometry, while those in deeper layers mainly control appearance. Note that our method is not trained with the style mixing strategy.

C.5. Image embedding and editing

Figure 9 shows more synthesized novel-view images obtained by embedding the given single images. We achieve high-resolution image embedding and pose manipulation with well-maintained 3D consistency even for fine details.

C.6. Failure cases of generated results

Floaters On some randomly generated results, there could be unwanted floaters in the front, as shown in Figure 11 (left). These floaters are not produced by super-resolution module, but already exist on the LR radiance manifolds. The reason may be that the supervision in LR image-space cannot eliminate such floaters for they look fine at low resolution. Jointly training the whole model in one stage may solve the problem, which we leave as our future work.

Exaggerated parallax artifacts When rotating the camera, some contents (e.g. hair fringes) on certain generated instances could be floating at unexpected positions, as shown in Figure 11 (right). We reckon that this is due to the shared surface manifolds across the whole category cannot

provide accurate position for all structures of all instances. It could be alleviated by using more shared surface manifolds or learning instance-specific manifolds, which we will also explore in future.

References

[1] 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

[2] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *IEEE/CVF International Conference on Computer Vision*, 2022. 3

[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

[4] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020. 1

[5] 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. 3

[6] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. Gram: Generative radiance manifolds for 3d-aware image generation. In *IEEE/CVF International Conference on Computer Vision*, 2022. 1, 2

[7] 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

[8] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. In *International Conference on Learning Representations*, 2021. 3

[9] 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

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 1, 3

[11] Taehee Brad Lee. Cat hipsterizer. https://github.com/kairess/cat_hipsterizer, 2018. 1

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

[13] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. Stylesdf: High-resolution 3d-consistent image and geometry generation. In *IEEE/CVF International Conference on Computer Vision*, 2022. 3

[14] 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

[15] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 2

[16] Ivan Skorokhodov, Sergey Tulyakov, Yiqun Wang, and Peter Wonka. EpiGRAF: Rethinking training of 3d GANs. In *Advances in Neural Information Processing Systems*, 2022. 3

[17] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 2021. 2

[18] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018. 2

[19] 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. 3

[20] Xiaoming Zhao, Fangchang Ma, David Güera, Zhile Ren, Alexander G. Schwing, and Alex Colburn. Generative multiplane images: Making a 2d gan 3d-aware. In *Proc. ECCV*, 2022. 3

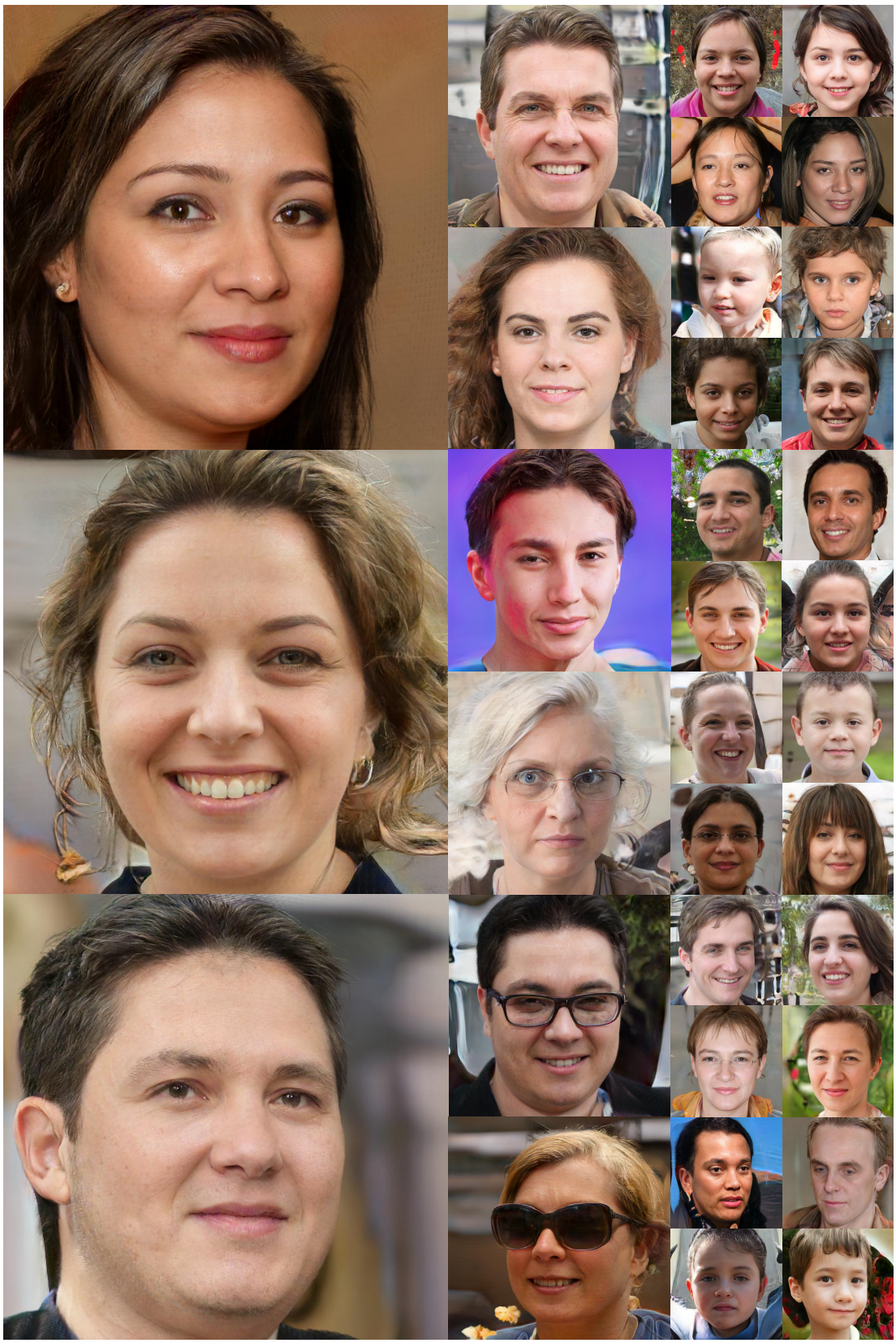

Figure 2: Uncurated 1024^2 results of GRAM-HD on FFHQ.

Figure 3: Uncurated 512^2 results of GRAM-HD on AFHQv2-CATS.

Figure 4: Multiview generation results of GRAM-HD on FFHQ.

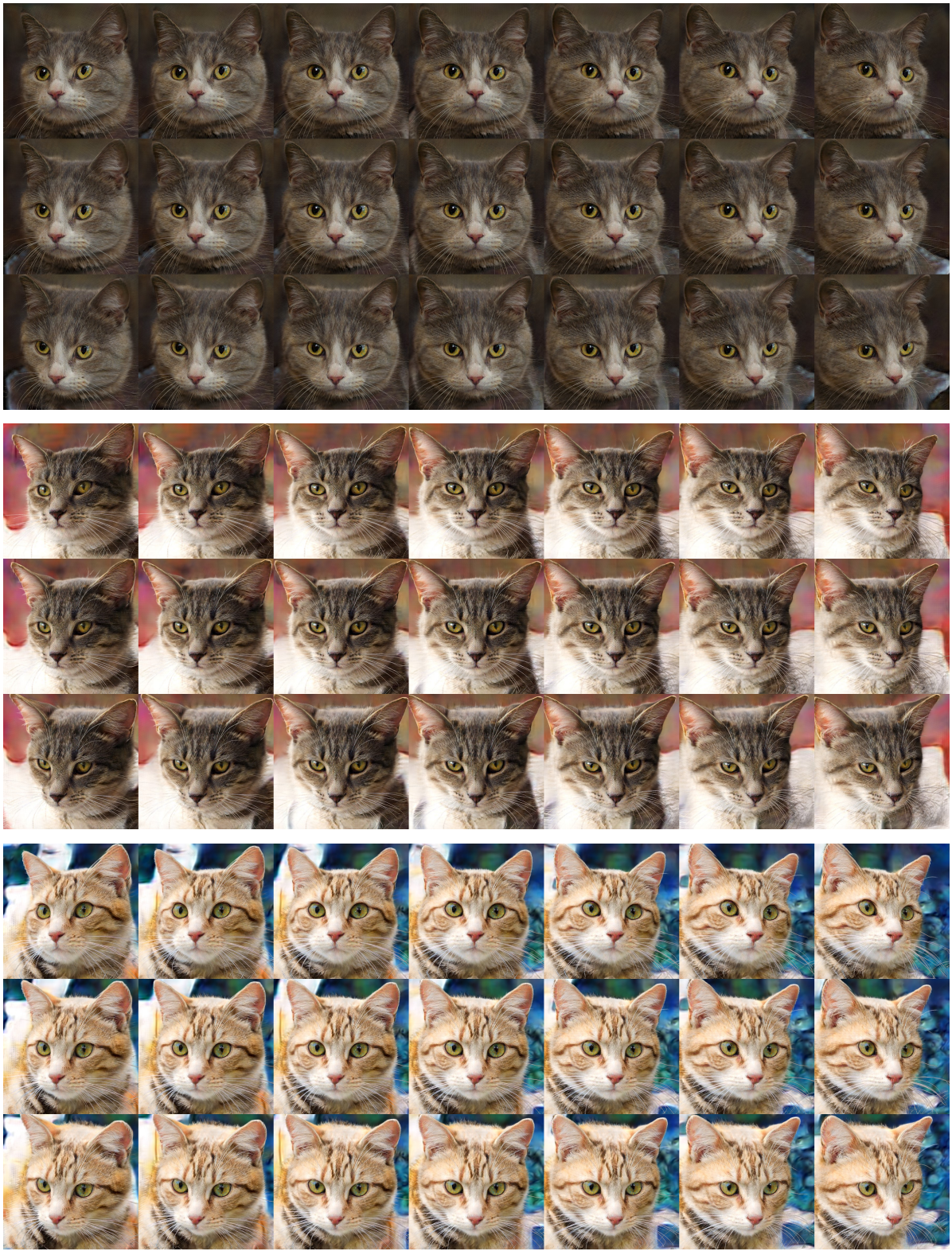

Figure 5: Multiview generation results of GRAM-HD on AFHQv2-CATS.

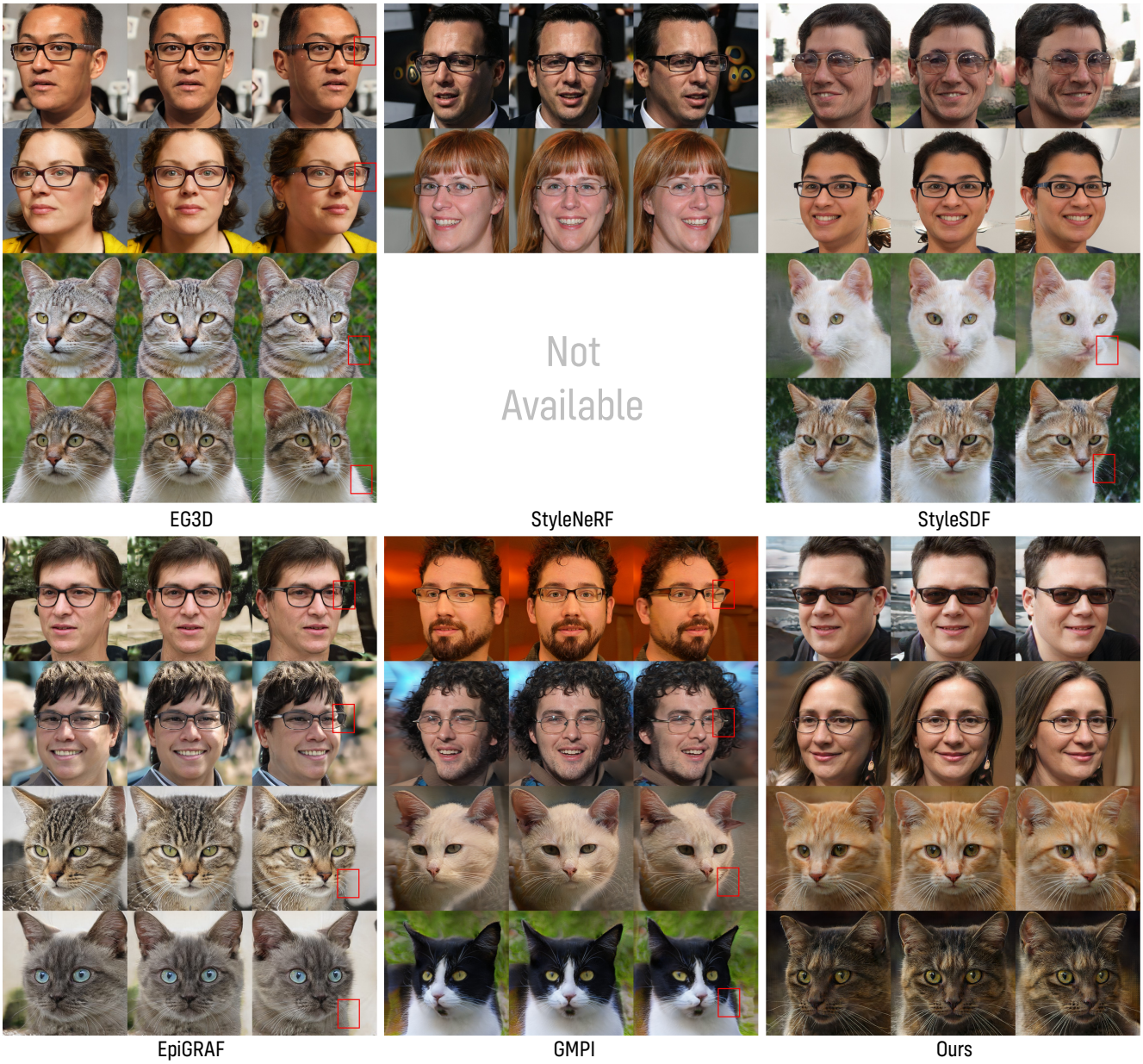

Figure 6: More comparison of geometry details. Our method can generate thin geometry structures such as glasses and whiskers while other methods either suffer some distortion (marked with box) or are not 3D-consistent. (*Best viewed with zoom-in; see also the accompanying video for better visualization.*)

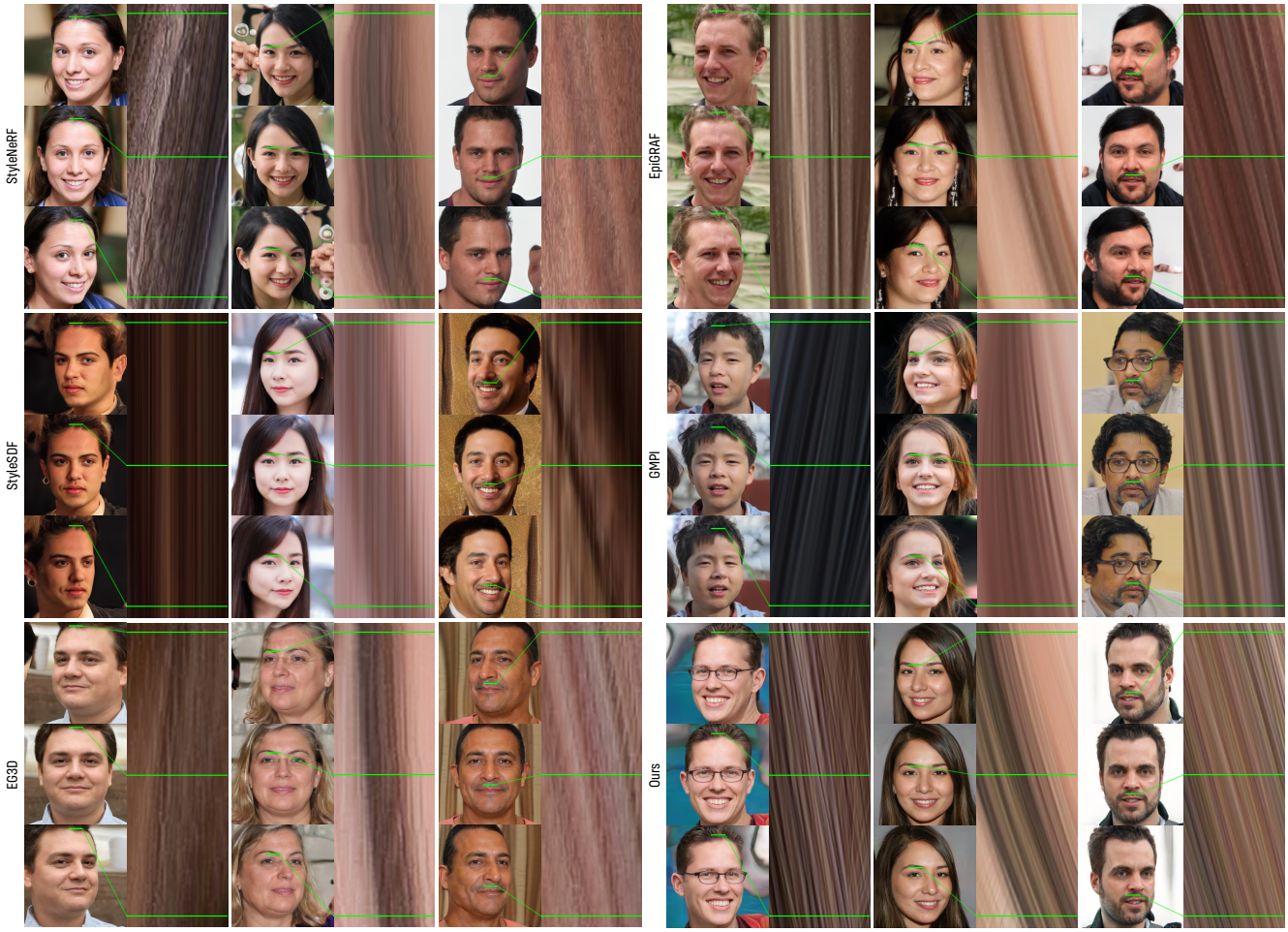

Figure 7: More comparison of 3D consistency using spatiotemporal texture image.

Figure 8: Latent space interpolation results.

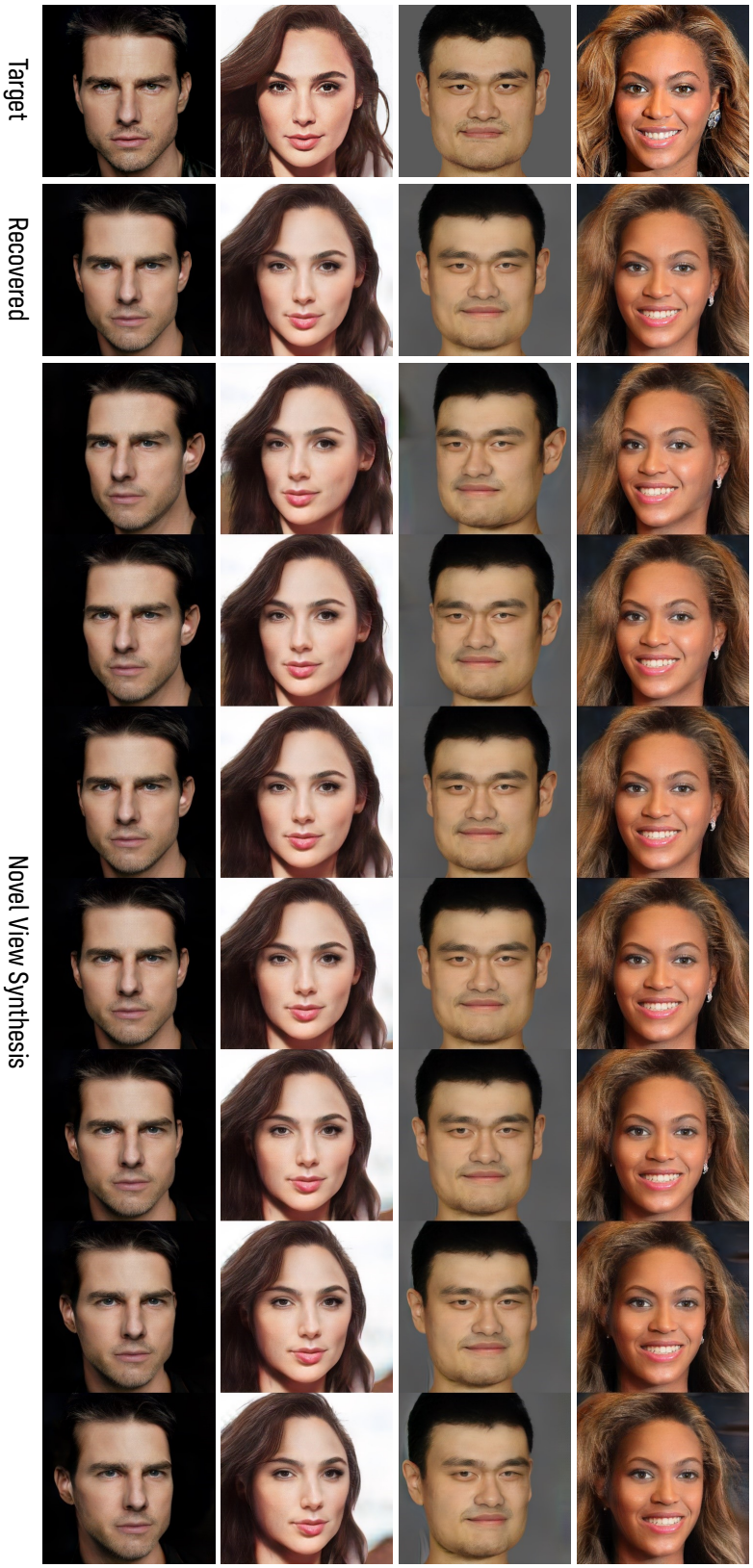

Figure 9: More high-resolution image embedding and editing results.

Figure 10: Style mixing between different generated subjects.

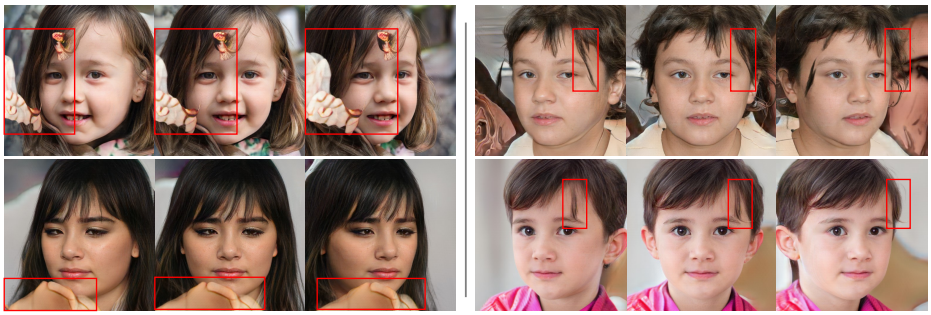

Figure 11: Failure cases. **Left:** Unwanted floaters on the generation results caused by LR generation (not the super-resolution module). **Right:** Exaggerated parallax on some generated instances.