Learning Detailed Radiance Manifolds for High-Fidelity and 3D-Consistent Portrait Synthesis from Monocular Image
(*Supplementary Material*)

I. More Implementation Details

I.1. Data Preparation

We align all images in FFHQ [10] and CelebA-HQ [9] using the detected facial landmarks following [5]. Specifically, we first use an off-the-shelf landmark detector [1] to extract 5 facial landmarks for each image. Then, we resize and crop the images by solving a least square problem between the detected landmarks and canonical 3D landmarks from the average shape of a 3D face model [20]. Camera poses of the images are extracted using a 3D face reconstruction model [6].

I.2. Network Structure

The structure of the detail manifolds reconstructor is shown in Fig. V. It consists of two sub-networks. A detail encoder E_{detail} and a super-resolution module $\mathcal{U}$.

Detail encoder E_{detail}. The detail encoder receives the concatenation of the input image $\hat{I}$ and the difference map $\hat{I} - I_w$, and predicts a low-resolution feature voxel V (see Fig. V (a)). It consists of several 2D downsampling blocks, followed by a 2D convolution to project the low-resolution 2D feature map to 3D voxel. A 3D U-Net structure with skip connections is then applied, followed by several 3D resblocks to obtain the final low-resolution feature voxel V.

Super-resolution module $\mathcal{U}$. The super-resolution module takes the low-resolution feature map F_i^{lr} derived from each low-resolution feature manifold as input, and produces a high-resolution feature map F_i^{hr} which will be later projected back to the surface manifolds (see Fig. V (b)). It consists of two upsampling blocks, and each block contains two 2D convolutions.

I.3. Intersection Calculation Details

The efficient GRAM requires to calculate ray-manifold intersections for manifold rendering following [5]. To accelerate the efficiency of this process, we calculate ray-manifold intersections at 1/4 resolution of the final image, as depicted in Fig. I.

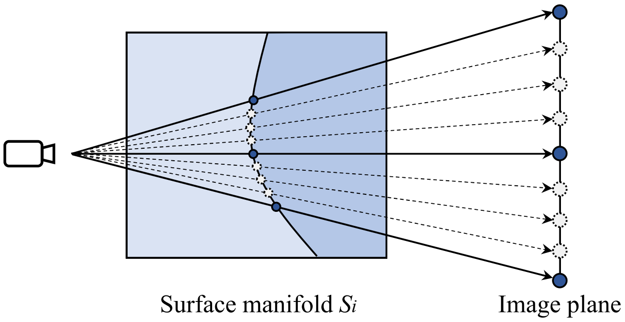


Figure I. Illustration of our intersection calculation. Ray-manifold intersections are first calculated at $1/4$ resolution of the final image (*i.e.* blue dots), and then go through bilinear upsampling to obtain dense intersections at the final resolution (*i.e.* gray dots).

Specifically, we first generate viewing rays at a resolution of 64×64, and calculate their intersections with each surface manifold produced by the manifold predictor $\mathcal{M}$ following [5]. Then, we upsample the obtained low-resolution intersection grid on each manifold via bilinear interpolation to obtain dense intersections at the final resolution (*i.e.* 256×256). In this way, only the low-resolution intersections obtained in the first step require forwarding the manifold predictor, which largely reduces the computation cost compare to directly calculating intersections at the final resolution. Since the learned surface manifolds for human faces have small curvature and are nearly planar at local regions (see illustration in [5]), the intersections obtained via the bilinear upsampling are close to the ground truth and have a minor influence on the final synthesis results.

I.4. More Training Details

Pretraining efficient GRAM. We follow [5] to train the efficient GRAM on FFHQ dataset at a resolution of 256×256. During training, we randomly sample latent code $\boldsymbol{z}$ from the normal distribution and camera pose $\boldsymbol{\theta}$ from the estimated distribution of the training data and send them to the efficient GRAM to generate corresponding images. The manifold predictor $\mathcal{M}$ is initialized following [5]. The tri-plane generator Ψ and the MLP-based decoder m are initialized following [3]. The synthesized images, together with randomly sampled real images from the training data, are sent into an extra discriminator [11] for loss computation. We adopt the non-saturating GAN loss with R1 regularization [15] to learn the efficient GRAM and the discriminator. We also enforce the pose regularization in [5] to ensure that the learned geometries are reasonable.

We use the Adam optimizer [12] with $\beta_1 = 0$ and $\beta_2 = 0.99$. The learning rates are set to $2.5e-3$ for the tri-plane generator and the MLP-based decoder, $2e-5$ for the manifold predictor, and $1e-3$ for the discriminator. The loss weights for the R1 regularization and the pose regularization are set to 10 and 30, respectively. We trained the efficient GRAM for 150K iterations with a batchsize of 32. Training took 2 days on 4 NVIDIA Tesla V100 GPUs with 32GB memory.

General inversion stage. During this stage, we fix the pre-trained efficient GRAM and learn the image inverter E_w. The image inverter is initialized following [26]. We adopt the multi-level reconstruction loss $\mathcal{L}_r$ for faithful image reconstruction (*i.e.* Eq.(6) in the main paper), and the minimal variation loss $\mathcal{L}_{d-reg}$ proposed in [26] to ensure that each $\boldsymbol{w}_i, i = 1, ..., L$ in the predicted $\boldsymbol{w}^+$ latent code are close to each other. Besides, we apply a regularization on the predicted latent code $\boldsymbol{w}^+$ to ensure that it falls in a semantically meaningful latent space:

$$\mathcal{L}_{\boldsymbol{w}+} = ||\boldsymbol{w}^+ - \bar{\boldsymbol{w}}^+||^2, \tag{I}$$

where $\bar{\boldsymbol{w}}^+$ is the average latent code of the $\mathcal{W}+$ space computed using 10K randomly sampled $\boldsymbol{z}$.

We use the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The initial learning rate for the image inverter is $3e-4$, and decreases to $6e-5$ after 100K iterations. The balancing weights for the three terms in $\mathcal{L}_r$ are set to $1e-2$, 1, and $4e-2$, respectively. The weights for $\mathcal{L}_{d-reg}$ and $\mathcal{L}_{\boldsymbol{w}+}$ are $1e-3$ and $1e-4$, respectively. The network is trained for 150K iterations with a batchsize of 32, which took 2 days on 4 NVIDIA Tesla V100 GPUs.

General inversion stage - finetuning. After the image inverter is learned, we further finetune the efficient GRAM for better image reconstruction. We only finetune the tri-plane generator Ψ and the MLP-based decoder m, and leave the manifold predictor $\mathcal{M}$ unchanged. The two networks are learned following [22]. Specifically, we adopt the multi-level reconstruction loss $\mathcal{L}_r$ in the main paper. In addition, we leverage the locality regularization $\mathcal{L}_R$ proposed in [22] to ensure that images synthesized by the finetuned efficient GRAM stay close to those of the original one at randomly sampled locations in the latent space. Different from [22], we finetune the efficient GRAM on the whole training set instead of using only a single image.

During training, the Adam optimizer is also applied with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate for the efficient GRAM is $1e-3$. The balancing weights for $\mathcal{L}_r$ are similar to the above stage, and the weight for $\mathcal{L}_R$ is set to 0.5. We use a batchsize of 16 and finetune the efficient GRAM for 100K iterations. The whole process took 1 day on 4 NVIDIA Tesla V100 GPUs.

Detail-specific reconstruction stage. Finally, we fix the image inverter as well as the efficient GRAM learned from the previous stages, and learn the detail manifolds reconstructor via the losses proposed in Sec. 3.4 in the main paper. We set the balancing weights for $\mathcal{L}_r$ following the above stages. The loss weights for the novel view regularization $\mathcal{L}_{nv}$ and the depth regularization $\mathcal{L}_{depth}$ are set to 4 and $2e-4$, respectively. We use the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and set the learning rate for the detail manifolds reconstructor to $3e-4$. We use a batchsize of 8 and train the whole pipeline for 60K iterations. It took 1 day on 4 NVIDIA Tesla V100 GPUs.

I.5. Baseline Implementation Details

PIRenderer. PIRenderer [21] is a face-reenactment method learned on video data. It leverages a 3D Morphable Model (3DMM) [6, 20] as guidance and learns 2D warping flow to drive a source image with target motions. It supports intuitive control of a given image by directly modifying the input 3DMM parameters to the network. We use the officially released code and model trained on VoxCeleb [18] dataset[1] in our experiments, and achieve pose editing of an image by modifying the input 3D pose parameters.

Face-vid2vid. Face-vid2vid [28] is also a face-reenactment method learned on video data. It extracts 3D keypoints of an image and derives 3D warping flows from them to transfer the 3D features of a source image to a target position. By using a single frame as both the source and the target, and applying 3D rotation to the extracted 3D keypoints of the target, it can also achieve intuitive control over the 3D pose of a given portrait image. Since the official code and model are unavailable, we use a re-implementation of it trained on VoxCeleb dataset[2] for our experiments.

e4e. e4e [26] is an encoder-based StyleGAN2 [11] inversion method. Its encoder adopts a feature-pyramid structure [13] and predicts StyleGAN2's $\mathcal{W}+$ space vector for a

[1] https://github.com/RenYurui/PIRender

[2] https://github.com/zhanglonghao1992/One-Shot_Free-View_Neural_Talking_Head_Synthesis

given image. By editing the predicted latent code towards certain direction, and sending the modified code into the pre-trained StyleGAN2, it can achieve pose control of the given image. We use the official released code and model trained on FFHQ[3] to carry out our experiments.

HFGI. HFGI [27] is also an encoder-based StyleGAN2 inversion method. It builds upon the e4e method and extracts extra feature maps from a given image as substitutions to the original feature maps within StyleGAN2. Therefore, it achieves more faithful inversion results compare to e4e. We use its officially released code and model trained on FFHQ[4] in our experiments.

InterFaceGAN. InterFaceGAN [23] is a latent space editing method for StyleGAN [10] and StyleGAN2. It learns the binary classification boundaries of multiple image attributes for latent vectors in StyleGAN's $\mathcal{W}+$ space. By modifying the latent code along the direction perpendicular to an interface, it can change the corresponding attribute of a synthesized image. It can also be combined with GAN inversion methods like e4e and HFGI for real image editing. Since the officially released model only contains interfaces for StyleGAN, we use the model provided by [27] for StyleGAN2-based pose editing.

StyleHEAT. StyleHEAT [30] is also a latent space editing method for StyleGAN2 which targets at talking head synthesis. Different from InterFaceGAN, it modifies the latent feature maps within the StyleGAN2 instead of the $\mathcal{W}+$ space latent vector. It learns 2D warping flows for the feature maps via the help of video data as well as the guidance of 3DMM, similarly as done by PIRenderer. It also supports direct 3D pose editing of a given image by modifying the 3D pose parameters for generating the warping flow. We use the officially released code and model trained on VoxCeleb[5] in our experiments.

pix2NeRF. pix2NeRF [2] is an encoder-based 3D-aware GAN inversion method based on pi-GAN [4]. It simultaneously learns an image encoder and a 3D-aware image generator to reconstruct NeRF [16] representation from a given image for novel view synthesis. We adopt its official released code[6] in our experiments. Since the official model is trained on CelebA [14] dataset that overlaps with our test set, we re-train it on the FFHQ dataset for a fair comparison.

Table I. Comparison with the full pipeline of IDE-3D which contains an extra optimization step.

Methods	PSNR ↑	LPIPS ↓	ID_{nv} ↑	PSNR_{mv} ↑	Time(s) ↓
IDE-3D (full)	**24.43**	**0.092**	0.507	37.10	100
Ours	21.57	0.123	**0.645**	**39.53**	**0.3**

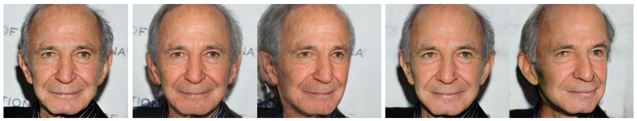


Figure II. Comparison with the full pipeline of IDE-3D.

IDE-3D. IDE-3D [25] is a 3D-aware GAN aiming for 3D-consistent portrait synthesis with interactive control. Its image generator is based on the tri-plane generator and the 2D super-resolution module proposed in [3]. It also achieves disentangled editing of real images by introducing a hybrid GAN inversion scheme, where it first learns an image encoder to map a given image into the latent space of the pre-trained generator, and then leverages instance-specific optimization [22] to further improve the reconstruction fidelity. We use its official model trained on FFHQ[7] in our experiments. Moreover, in the main paper, we only use the inversion results from its encoder instead of those from the further optimization step for a fair comparison. A comparison with its full pipeline including the optimization step is demonstrated in Tab. I and Fig. II.

I.6. Visualization Details

Visualization results in this paper are rendered with yaw angles ranging from -0.4 rad to 0.4 rad. The pitch angles are identical to those of the input images, which are estimated via the face reconstruction method of [6]. The roll angles are set to zero.

I.7. Novel View Experiment Details

We describe more details about the novel view synthesis comparison proposed in Sec. 4.2 in the main paper. Specifically, we generate novel views of the first 1K test images in the CelebA-HQ dataset using different methods to calculate the metrics (*i.e.* ID_{nv} and FID_{nv} in Tab. 2). We randomly sample the yaw angle within a range of $[-0.5, -0.4] \cup [0.4, 0.5]$, and set the pitch and roll identical to those of the original input. To ensure that the novel view images have a large pose difference with the input, we multiply the sampled yaw angle by -1 if its absolute difference with that of the input is smaller than 0.3. For all methods, we use the same 1K sampled yaw angles to generate novel view images for a fair comparison.

[3] https://github.com/omertov/encoder4editing
[4] https://github.com/Tengfei-Wang/HFGI
[5] https://github.com/FeiiYin/StyleHEAT/
[6] https://github.com/primecai/Pix2NeRF
[7] https://github.com/MrTornado24/IDE-3D

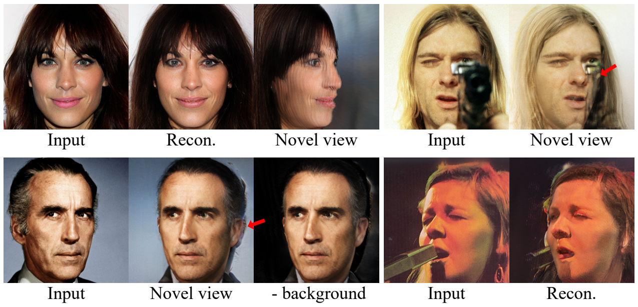


Figure III. Limitations of our method. It can produce layered artifacts under large viewing angles. We also observe a ghosting artifact for certain subjects where the background contains the appearance of ears. In addition, it cannot well handle occlusions and out-of-distribution data.

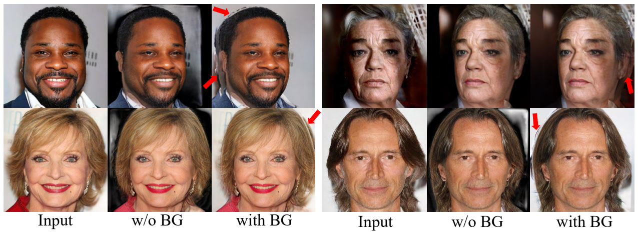


Figure IV. More comparisons between rendering with or without the background plane. **Best viewed with zoom-in.**

II. More Results

II.1. Novel View Synthesis Results

Figure VI, VII and VIII shows more novel view synthesis results by our method on CelebA-HQ test data. Figure IX further shows novel view synthesis results on in-the-wild images. Our method can generate realistic novel views with high fidelity and strong 3D consistency for diverse subjects. **Please see the project page for animations.**

II.2. Comparisons with the Prior Art

Figure X and XI show more comparisons between our method and the previous methods. Our method can well preserve fine details in the original images and produces their novel views with more strict 3D consistency compared to the others. **Please see the project page for animations.**

We further compare with the full inversion pipeline of IDE-3D, which adopts the EG3D structure and leverages optimization-based inversion (*i.e.* encoder-based initialization + pivot tuning). The results and a visual example are shown in Tab. I and Fig. II. The PSNR, LPIPS, and ID_{nv} are calculated on the first 100 instances in the CelebA-HQ, and the PSNR_{mv} on 50 instances. Our method performs slightly worse than the state-of-the-art optimization-based method on image reconstruction quality, but shows better novel view results and 3D consistency, and has dramatically faster inference speed.

II.3. More Applications

Dolly zoom effect. Since our method is based on GRAM [5] that leverages the radiance manifolds representation, we can explicitly move the camera towards or away from a subject, and adjust the camera fov accordingly to ensure that the size of a portrait in the synthesized image stays a constant. In this way, we can generate a sequence of images under different levels of camera distortions, which is known as the dolly zoom effect[8]. It can hardly be achieved by 2D-GAN based face editing methods without explicit camera modeling. Examples of this effect generated by our method are shown in Fig. XII. **Animations can be found in the project page.**

3D-consistent editing. Our method can also be applied to 3D-consistent interactive portrait editing thanks to its ability to preserve fine image details. Specifically, given a real portrait image, we can draw some arbitrary patterns on it and send the edited result to our GRAMinverter for reconstruction and novel view synthesis. As shown in Fig. XIII, our method can well preserve the drawn patterns on the input images and generate their 3D-consistent novel views bearing these patterns. **Corresponding animations are in the project page.**

III. Limitations and Future Works

We thoroughly discuss the limitations of our method and possible future solutions to improve it.

Our method adopts the radiance manifolds representation. Although it helps us to synthesize novel views with strong 3D consistency, it can produce layered artifacts at large viewing angles as shown in Fig. III. This artifact could be alleviated to some extent by using more profile images as training data. In addition, it could also be reduced by leveraging alternative 3D representations, such as some recently proposed efficient NeRF representations [8, 17, 24]. However, it is still unclear how to effectively incorporate these representations for high-quality and efficient novel view synthesis of monocular portraits.

We also observed ghosting artifacts in some cases where the background contains the appearance of ears. The major cause is that the background plane and the foreground subject share the same tri-plane generator so they might have similar appearance patterns in some regions. Some floating points can also be observed around the silhouette, which are mainly due to the wrong parallax provided by inaccurate coarse depth (geometry) estimated from the general inversion stage. These problems can be alleviated by only rendering the foreground subject as shown in Fig. III, or using

[8] https://en.wikipedia.org/wiki/Dolly_zoom

an extra image generator to synthesize the background. We show more comparisons with or without the background in Fig. IV. Clearly, removing the background largely reduces the layered artifacts and the floating points.

Besides, our method cannot well handle occlusions and tends to interpret them as textures clinging to the face as shown in Fig. III. One possible solution is to leverage an extra face segmentation network to mask out the occluded regions and let the model only focus on reconstructing the portrait region. Our method can also produce inferior results for out-of-distribution input with large poses and abnormal lighting. The synthesized images may also have a global color shift compared to the input in certain cases. We believe these problems can be mitigated by training on larger-scale datasets with carefully tuned loss weights. In addition, our method cannot well handle complex lighting effect when varying the camera pose, such as specular reflectance. More dedicated 3D representations [7] are required to tackle this problem.

Finally, our method does not support editing of attributes like expression, due to the learned details being aligned with the original input image. This problem can be tackled by introducing a distortion-aware detail reconstructor similarly as done by some recent 2D GAN inversion methods [27], or leveraging a 3D representation that handles dynamic changes [19, 29]. We leave these explorations as future works.

IV. Ethics Consideration

The goal of this paper is efficient large-scale virtual avatar creation. It does not intend to create misleading or deceptive content. However, it could still be potentially misused for impersonating humans. In particular, the 3D-consistent synthesized portraits might be used to fool the 3D face recognition system that relies on multiview consistency. We condemn any behavior to create such harmful content. Currently, the synthesized portraits by our method contain certain visual artifacts that can be identified by humans and some deepfake detection methods. We encourage to apply this method for learning more advanced forgery detection approaches to avoid potential misusage.

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] Shengqu Cai, Anton Obukhov, Dengxin Dai, and Luc Van Gool. Pix2nerf: Unsupervised conditional p-gan for single image to neural radiance fields translation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3981–3990, 2022. 3

[3] 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 Conference on Computer Vision and Pattern Recognition*, 2022. 2, 3

[4] 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. 3

[5] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. Gram: Generative radiance manifolds for 3d-aware image generation. In *IEEE Computer Vision and Pattern Recognition*, 2022. 1, 2, 4

[6] 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, 2, 3

[7] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. Nerfren: Neural radiance fields with reflections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18409–18418, 2022. 5

[8] Fangzhou Hong, Zhaoxi Chen, Yushi Lan, Liang Pan, and Ziwei Liu. Eva3d: Compositional 3d human generation from 2d image collections. *arXiv preprint arXiv:2210.04888*, 2022. 4

[9] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 1

[10] 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

[11] 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. 2

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 2

[13] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 2

[14] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015. 3

[15] 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. 2

[16] 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. 3

[17] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*, 2022. 4

[18] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*, 2017. 2

[19] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 5

[20] 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, 2

[21] Yurui Ren, Ge Li, Yuanqi Chen, Thomas H Li, and Shan Liu. Pirenderer: Controllable portrait image generation via semantic neural rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13759–13768, 2021. 2

[22] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *arXiv preprint arXiv:2106.05744*, 2021. 2, 3

[23] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *IEEE/CVF conference on computer vision and pattern recognition*, pages 9243–9252, 2020. 3

[24] Ivan Skorokhodov, Sergey Tulyakov, Yiqun Wang, and Peter Wonka. Epigraf: Rethinking training of 3d gans. *arXiv preprint arXiv:2206.10535*, 2022. 4

[25] Jingxiang Sun, Xuan Wang, Yichun Shi, Lizhen Wang, Jue Wang, and Yebin Liu. Ide-3d: Interactive disentangled editing for high-resolution 3d-aware portrait synthesis. *arXiv preprint arXiv:2205.15517*, 2022. 3

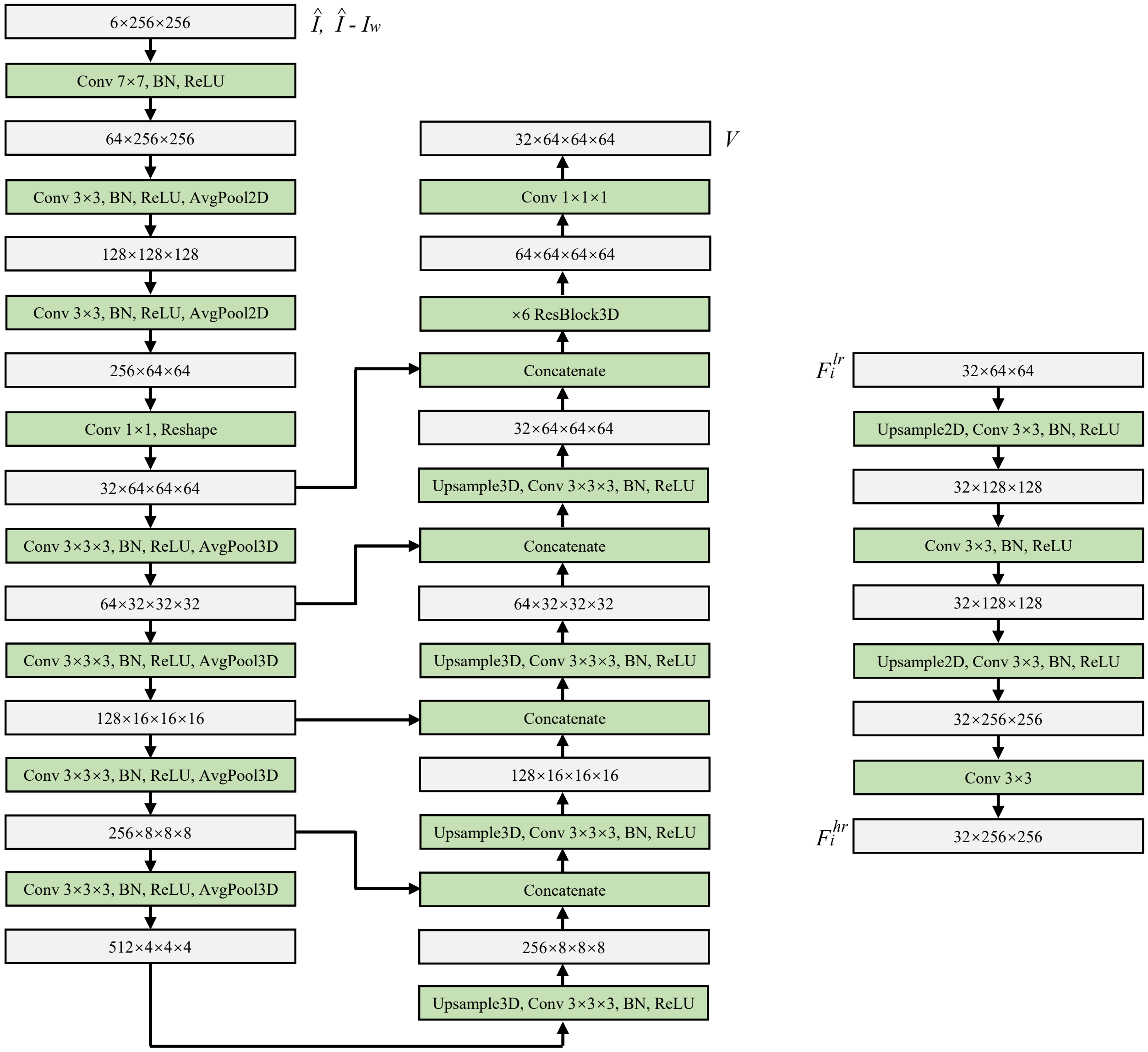
[26] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021. 2

[27] Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. High-fidelity gan inversion for image attribute editing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11379–11388, 2022. 3, 5

[28] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10039–10049, 2021. 2

[29] Yue Wu, Yu Deng, Jiaolong Yang, Fangyun Wei, Qifeng Chen, and Xin Tong. Anifacegan: Animatable 3d-aware face image generation for video avatars. *arXiv preprint arXiv:2210.06465*, 2022. 5

[30] Fei Yin, Yong Zhang, Xiaodong Cun, Mingdeng Cao, Yanbo Fan, Xuan Wang, Qingyan Bai, Baoyuan Wu, Jue Wang, and Yujiu Yang. Styleheat: One-shot high-resolution editable talking face generation via pretrained stylegan. *arXiv preprint arXiv:2203.04036*, 2022. 3


6×256×256
$\hat{I}$, $\hat{I}$ - I_w
Conv 7×7, BN, ReLU
64×256×256
Conv 3×3, BN, ReLU, AvgPool2D
128×128×128
Conv 3×3, BN, ReLU, AvgPool2D
256×64×64
Conv 1×1, Reshape
32×64×64×64
Conv 3×3×3, BN, ReLU, AvgPool3D
64×32×32×32
Conv 3×3×3, BN, ReLU, AvgPool3D
128×16×16×16
Conv 3×3×3, BN, ReLU, AvgPool3D
256×8×8×8
Conv 3×3×3, BN, ReLU, AvgPool3D
512×4×4×4
32×64×64×64
V
Conv 1×1×1
64×64×64×64
×6 ResBlock3D
Concatenate
32×64×64×64
Upsample3D, Conv 3×3×3, BN, ReLU
Concatenate
64×32×32×32
Upsample3D, Conv 3×3×3, BN, ReLU
Concatenate
128×16×16×16
Upsample3D, Conv 3×3×3, BN, ReLU
Concatenate
256×8×8×8
Upsample3D, Conv 3×3×3, BN, ReLU
F_i^{lr}
32×64×64
Upsample2D, Conv 3×3, BN, ReLU
32×128×128
Conv 3×3, BN, ReLU
32×128×128
Upsample2D, Conv 3×3, BN, ReLU
32×256×256
Conv 3×3
F_i^{hr}
32×256×256

(a) Detail encoder E_{detail}

(b) Superres. module $\mathcal{U}$

Figure V. Network structure of the detail manifolds reconstructor. It consists of a detail encoder E_{detail} and a super-resolution module $\mathcal{U}$.

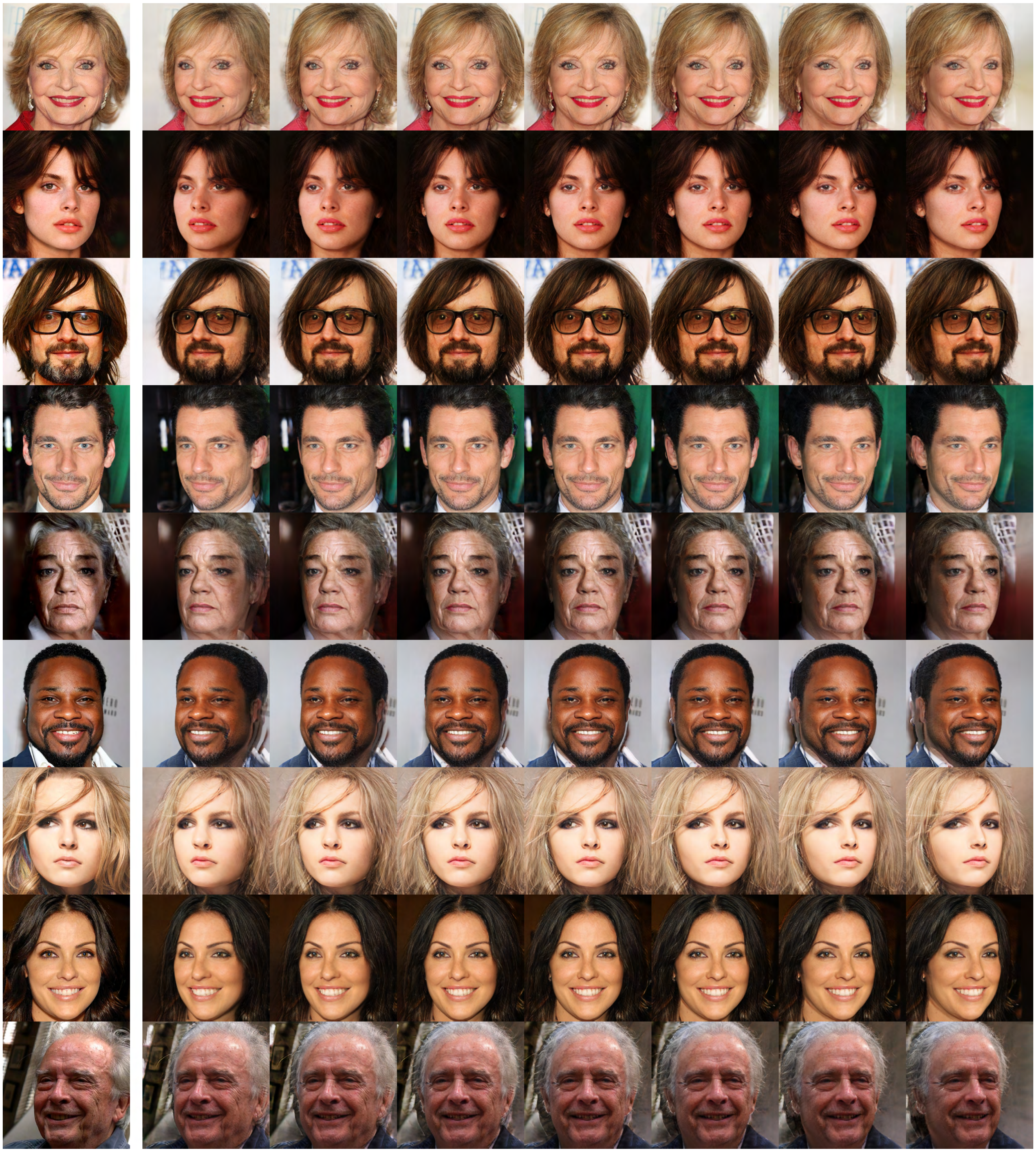


Figure VI. More novel view synthesis results on CelebA-HQ by our method. **Best viewed with zoom-in.**

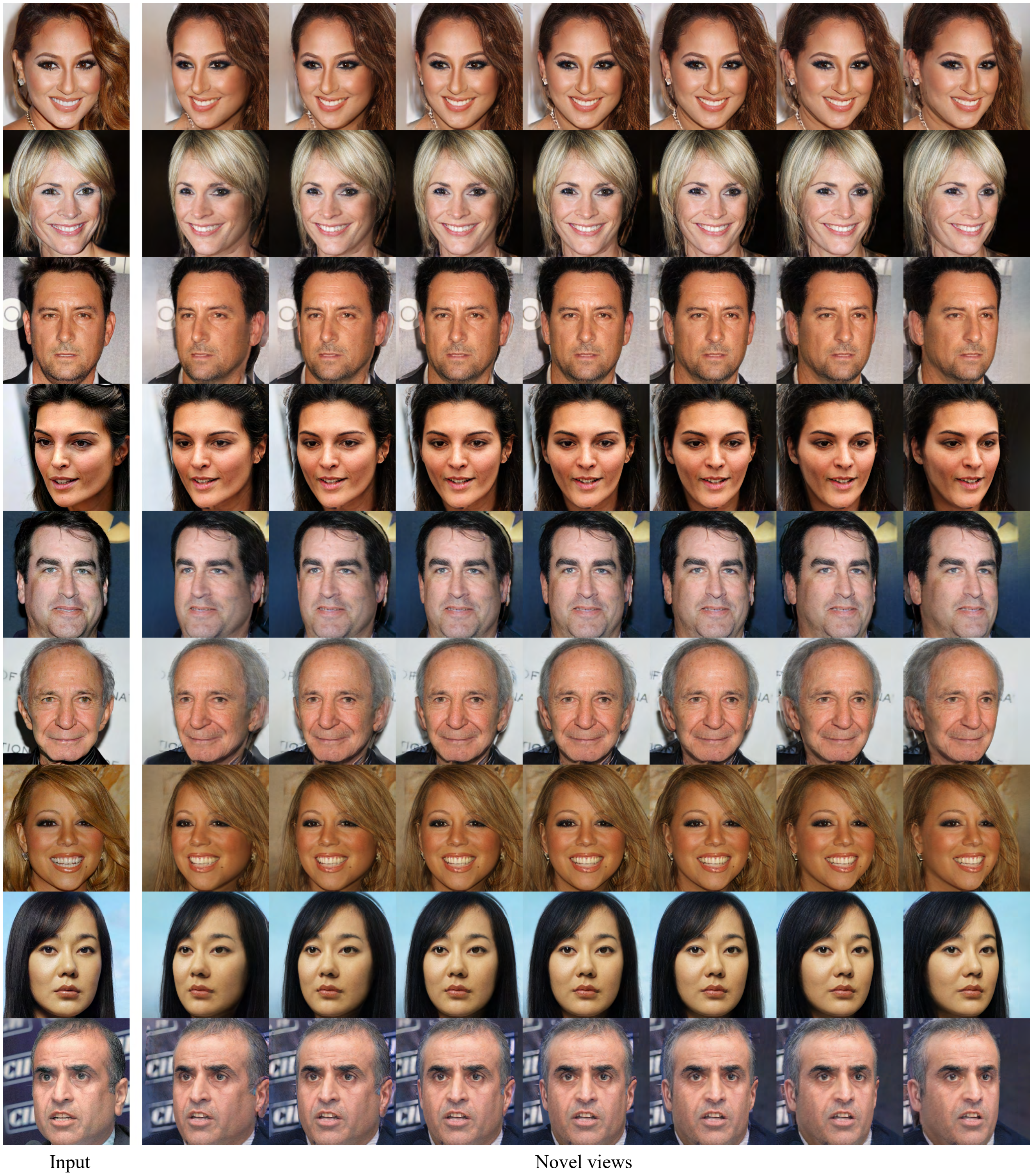


Figure VII. More novel view synthesis results on CelebA-HQ by our method. **Best viewed with zoom-in.**

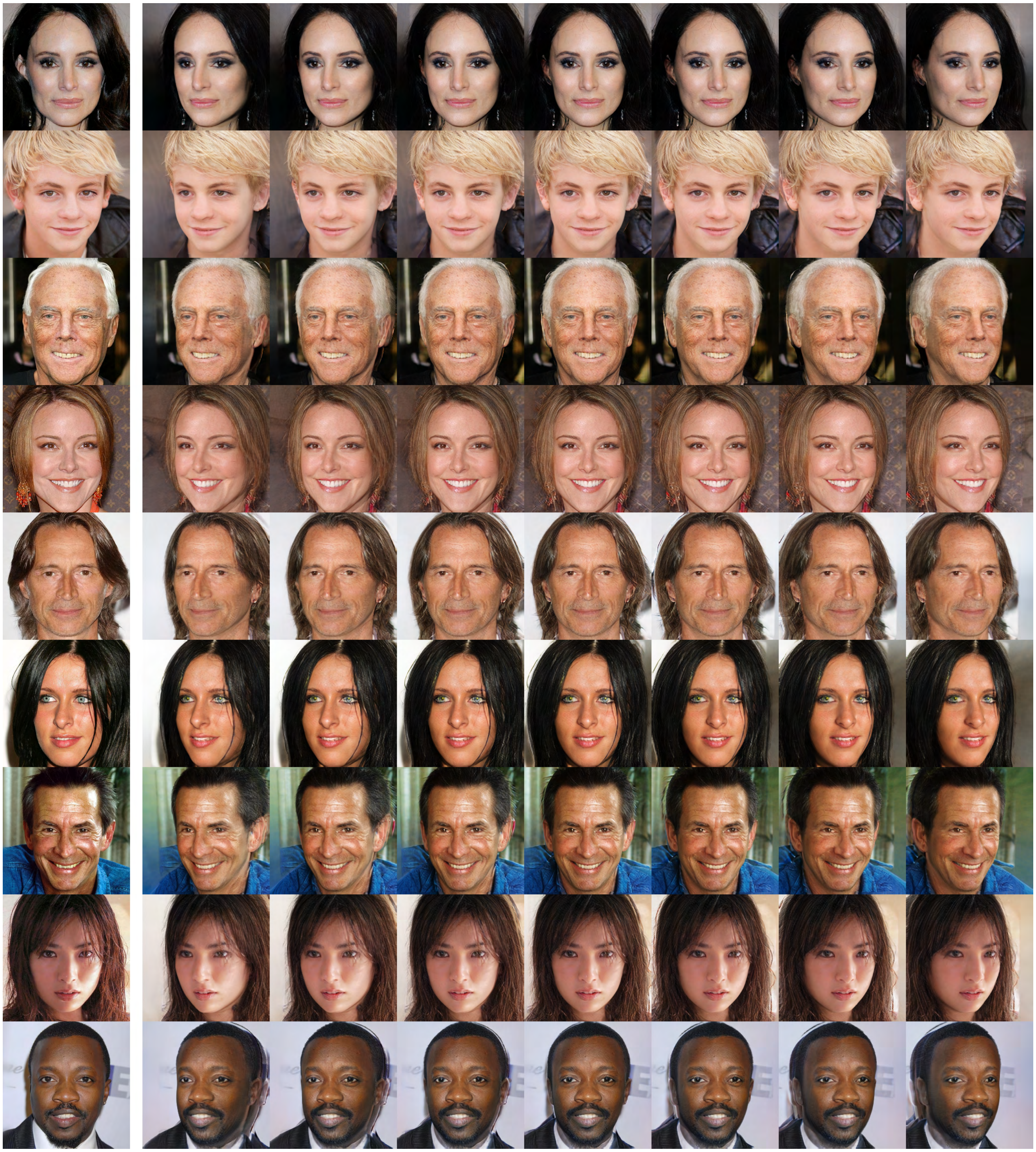


Figure VIII. More novel view synthesis results on CelebA-HQ by our method. **Best viewed with zoom-in.**

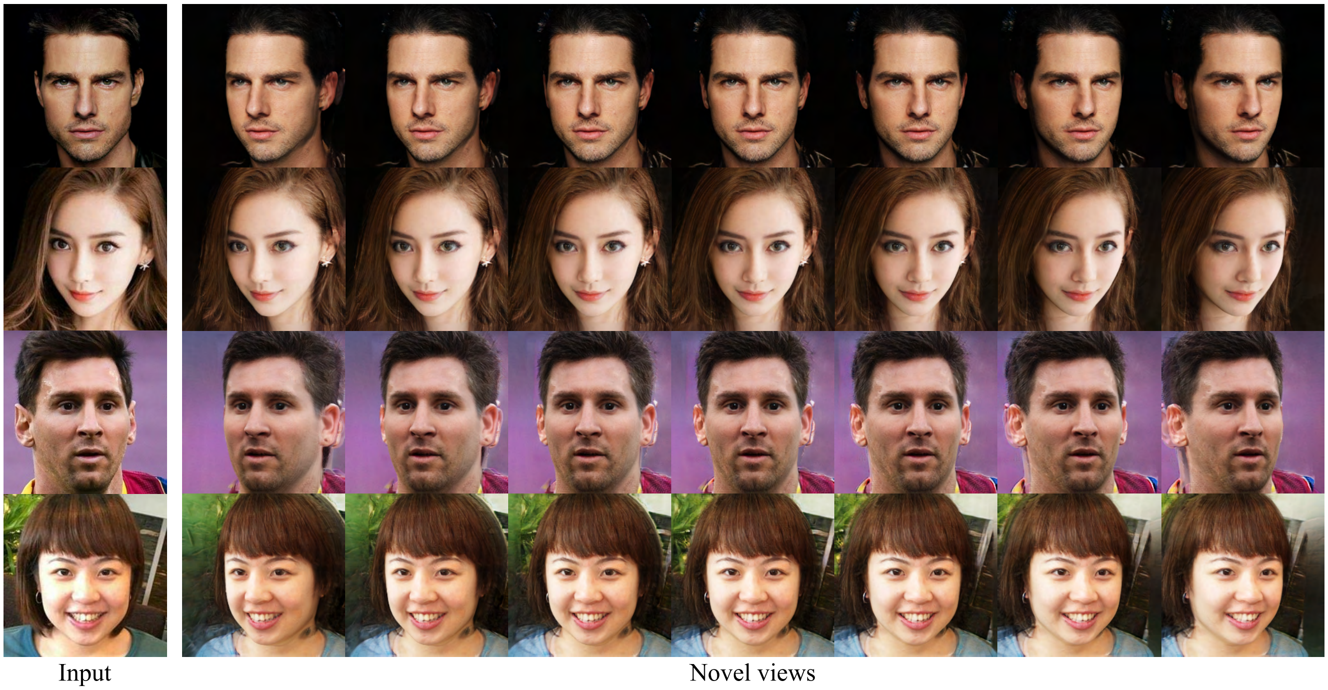
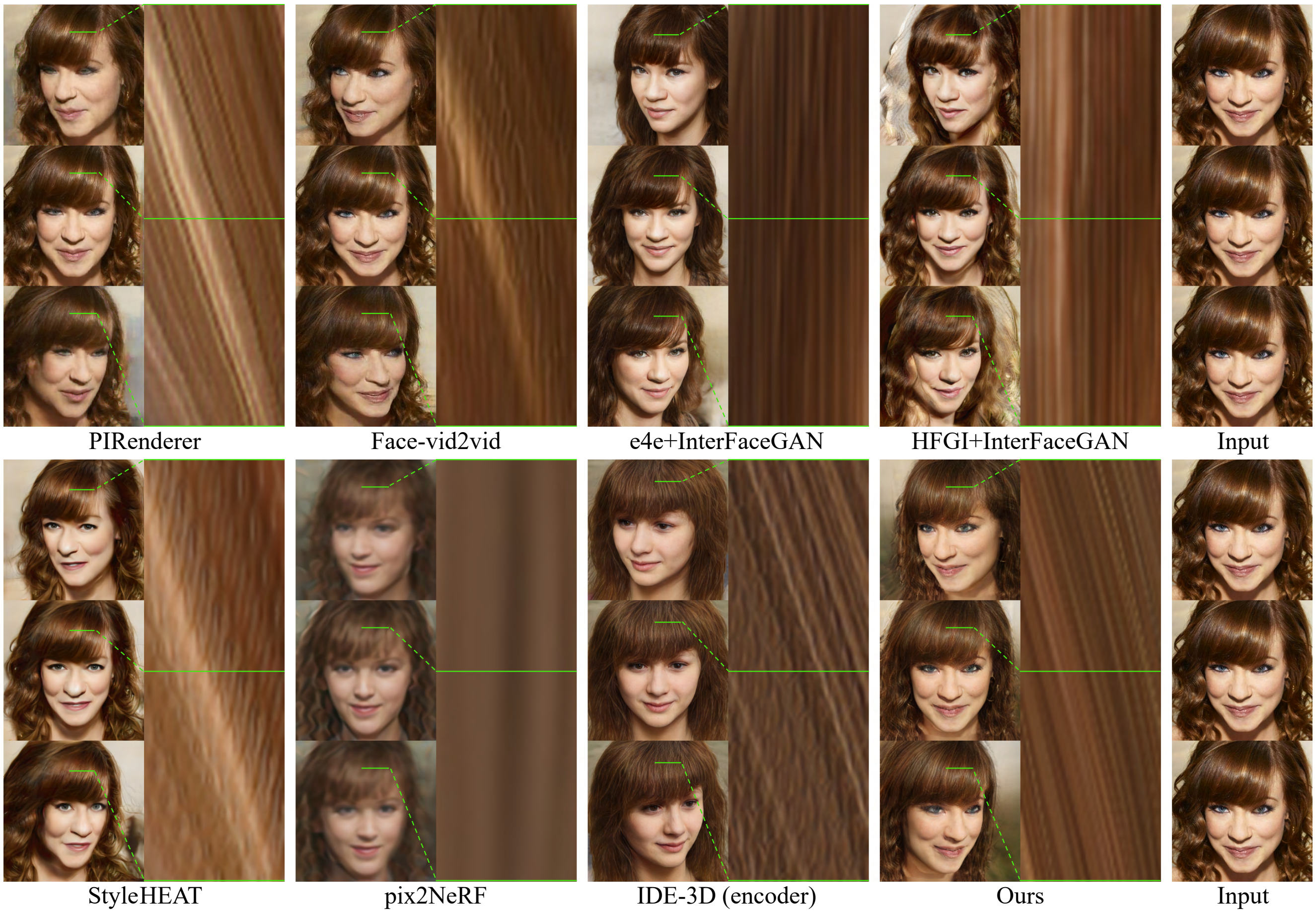


Figure IX. More novel view synthesis results on in-the-wild images. **Best viewed with zoom-in.**



PIRenderer

Face-vid2vid

e4e+InterFaceGAN

HFGI+InterFaceGAN

Input

StyleHEAT

pix2NeRF

IDE-3D (encoder)

Ours

Input

Figure X. More pose editing comparisons. **Best viewed with zoom-in and see the project page for animations.**

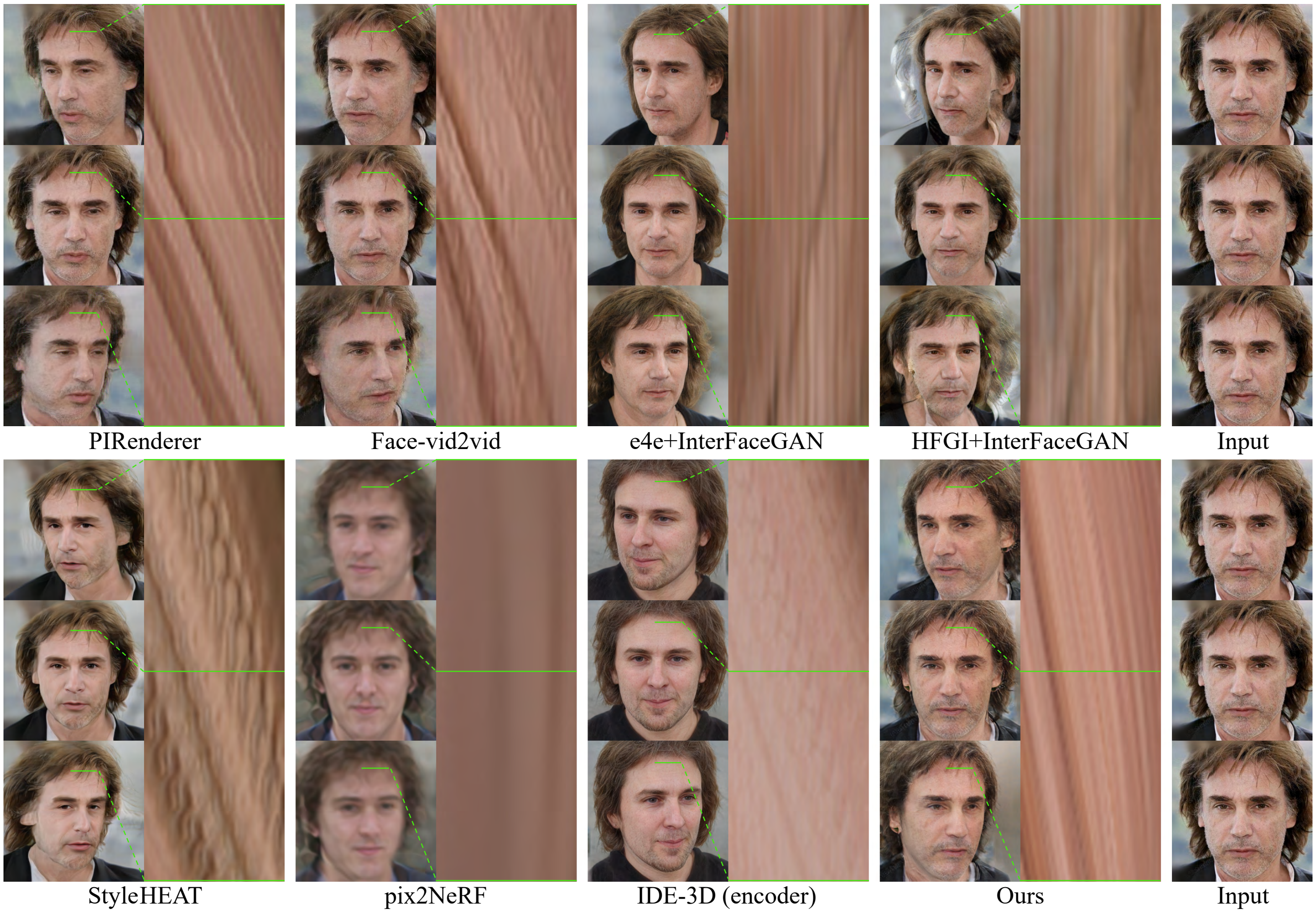


Figure XI. More pose editing comparisons. **Best viewed with zoom-in and see the project page for animations.**

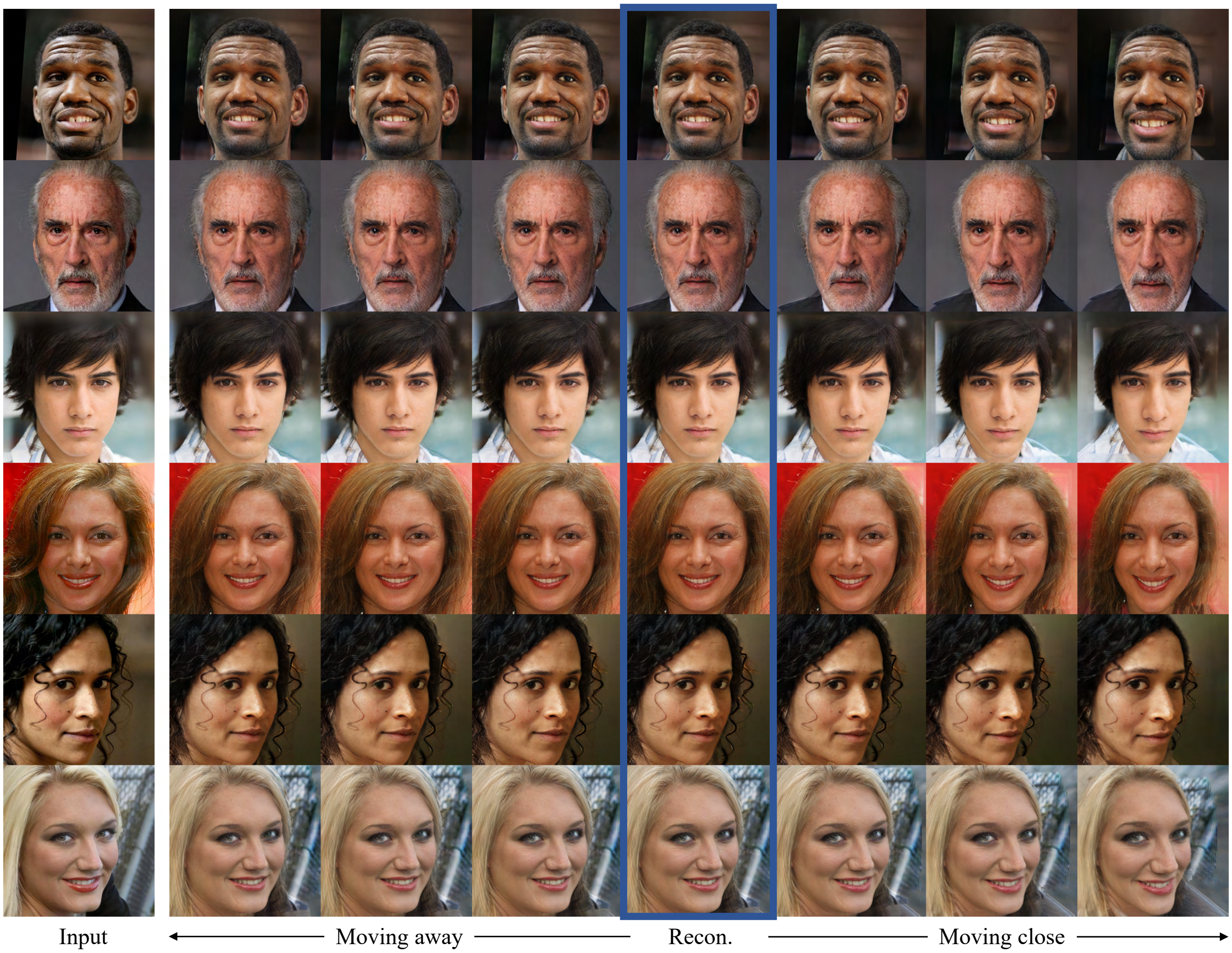


Figure XII. Dolly zoom effect of the given portraits produced by our method. **See the project page for animations.**

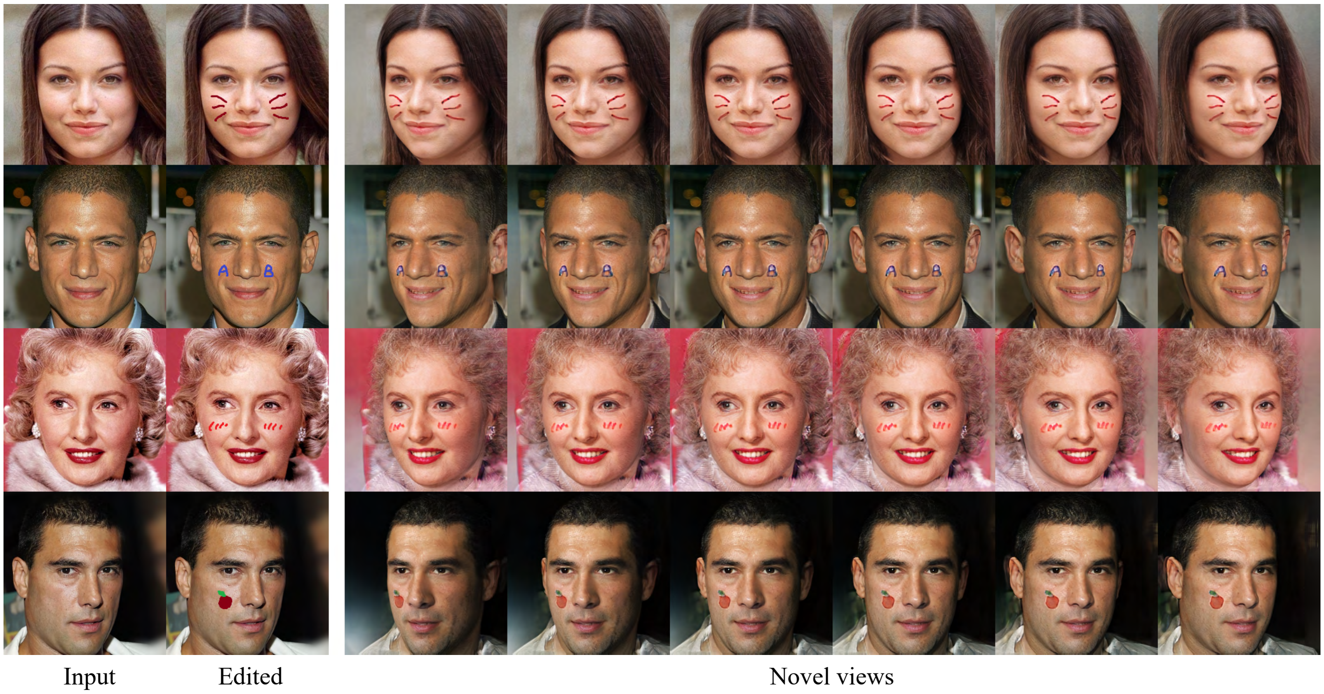


Figure XIII. 3D-consistent portrait editing results by our method. **See the project page for animations.**