G-NeRF: Geometry-enhanced Novel View Synthesis from Single-View Images

Supplementary Material

In the appendix, we provide more details and more experimental results of the proposed G-NeRF[1]. We organize the appendix into the following sections.

- In section A, we depict the preliminary of NeRF.
- In section B, we provide more implementation details.
- In section C, we show more qualitative and quantitative results.
- In section D, we provide more discussions about the potential limitations of our method and the difference with concurrent works.

A. Preliminary of Neural Radiance Fields

NeRF [11] aims to synthesize novel views of complex scenes from sparse input views. By querying 5D coordinates along a camera ray and leveraging volume rendering technique, NeRF generates the color of an image pixel that intersects with the ray. Specifically, for a given pixel coordinate $\mathbf{x} \in \mathbb{R}^2$ and camera parameters $\mathbf{P}$ of an image, we acquire a camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ where $\mathbf{o}$ is camera center, $\mathbf{d} \in \mathbb{S}^2$ denotes view direction calculated with $\mathbf{o}$, $\mathbf{x}$ and $\mathbf{P}$. Here, the procedure of acquiring rendered image color can be defined as the following equations with near and far bounds t_n and t_f regarding $\mathbf{r}$:

$$\begin{aligned} \hat{\mathbf{C}}(\mathbf{r}) &= \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt, \\ T(t) &= \exp(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds), \end{aligned} \tag{1}$$

where the density $\sigma(x)$ is the probability that the ray terminates at a particle. $T(t)$ denotes the probability the ray $\mathbf{r}$ travels from t_n to t without hitting any particle. To numerically estimate the continuous integral (Eq. 1), NeRF samples particles along the continuous camera ray $\mathbf{r}(t)$ with a stratified sampling strategy in which $\mathbf{r}(t)$ is evenly partitioned into n bins. By querying the position and direction of each particle with a multi-layer perception (MLP), we obtain the color and density of each particle. Using Eq. 2, we accumulate the color and density of particles along a ray for the pixel color of the rendered image:

$$\begin{aligned} \hat{\mathbf{C}}(\mathbf{r}) &= \sum_{i=1}^{N_s} \tau_i \alpha_i \mathbf{c}(\mathbf{r}(t_i)), \ \ \hat{\mathbf{D}}(\mathbf{r}) = \sum_{i=1}^{N_s} \tau_i \alpha_i z_i, \\ \text{where } \tau_i &= \prod_{j=1}^{i-1}(1-\alpha_j), \ \ \alpha_i = 1 - e^{-\sigma(\mathbf{r}(t_i))\delta_i}, \end{aligned} \tag{2}$$

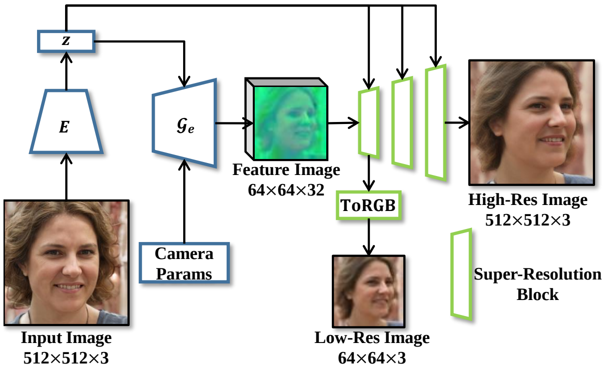

Figure A. **More details of our network architecture.** Our G-NeRF consists of a scene encoder E, an EG3D backbone $\mathcal{G}_e$ and three super-resolution blocks.

where τ_i denotes the accumulated transmittance along the ray from the t_n to t_f and $\delta_i = t_{i+1} - t_i$ is the distance of two adjacent particles. $\hat{\mathbf{D}}(\mathbf{r})$ is depth value of the rendered image and z_i denotes the depth of the i_{th} particle in the stratified $r(t)$. Since estimating $\hat{\mathbf{C}}(\mathbf{r})$ from $\mathbf{c}(\mathbf{r})$ and $\sigma(\mathbf{r}(t_i))$ is differentiable, NeRF is optimized via minimizing the MSE loss between $\hat{\mathbf{C}}(\mathbf{r})$ and the ground truth color $\mathbf{C}(\mathbf{r})$ via equation $\mathcal{L}_{nerf} = ||\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})||_2^2$.

B. More implementation Details

B.1. More details of our network architecture

The network architecture of G-NeRF is depicted in Fig. A. The structure of scene encoder E is ResNeXt [15] and borrow from [16]. The NeRF-based generator $\mathcal{G}_n$ consists of an EG3D backbone $\mathcal{G}_e$ and a super-resolution module. The structure of the EG3D-backbone $\mathcal{G}_e$ and depth-aware discriminator $\mathcal{D}_g$ are borrowed from [3]. Differently, to capture more information, we increase the latent code dimension of $\mathcal{G}_e$ to 5120. The super-resolution module includes three super-resolution blocks, which are the same blocks used in $\mathcal{G}_e$. All these modules are trained from scratch together.

B.2. More experimental details

All experiments are conducted on PyTorch [12] with 2 80GB RTX A800 GPUs. We use Adam [9] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ for $\mathcal{E}$ and $\mathcal{G}_n$, and $\beta_1 = 0$, $\beta_2 = 0.99$ for $\mathcal{D}_g$. We set the learning rate as 1e-03 for the generator and 8e-06 for the discriminator. For the hyperparameter λ_g, we empirically set it to 1.2. We train our model with FFHQ [8] for 4000k images with batch size 24 and for 2000k images with AFHQv2-Cats [5].

In comparisons on ShapeNet datasets, since Pix2NeRF [2] does not include an evaluation on the

---

[1] We suggest checking the video demo synthesized by our G-NeRF in the supplementary.

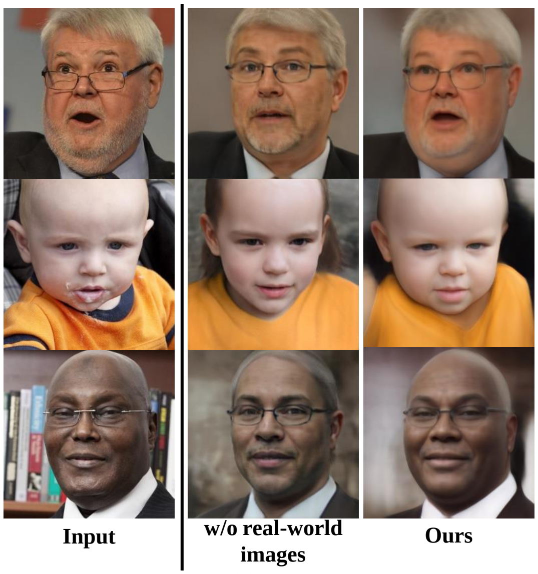

Figure B. **Ablation study of incorporating real-world images.** Without incorporating real-world images, our model generates results with poor similarity to the input images.

ShapeNet Cars [4, 13], we first train a Pix2NeRF model using the official code. We generate 62,000 synthetic images for ShapeNet Cars and 140,000 for ShapeNet Chairs. Subsequently, we incorporate the synthetic images with the ShapeNet datasets to train our model. During the evaluation phase, for each category, we randomly select one image as the input and ten images as ground truths.

C. Additional Results

C.1. Training without real-world images

Since we trade off the diversity and geometry quality in synthetic multi-view data through a truncation ratio of 0.5, it is important to incorporate real-world images to provide diverse appearance priors. We train our model without incorporating real-world images to verify its effectiveness. As seen from Fig. B and Tab. A, our model produces results with poor similarity to the reference images and all the evaluation metrics decrease compared to our full model.

C.2. More comparison with HeadNeRF

We compare our method with another single-shot novel view synthesis method named HeadNeRF [7], which introduces a NeRF-based parametric head model to synthesize controllable 3D faces from single-view images. As shown in Fig. C, HeadNeRF [7] is incapable of generating ID-preserving results but inherits some artifacts due to the limitation of the 3D Morphable Models (3DMMs). Moreover, HeadNeRF [7] also requires additional training processes to fit a single image. In contrast, our G-NeRF achieves high-fidelity novel view synthesis without any test-time optimization.

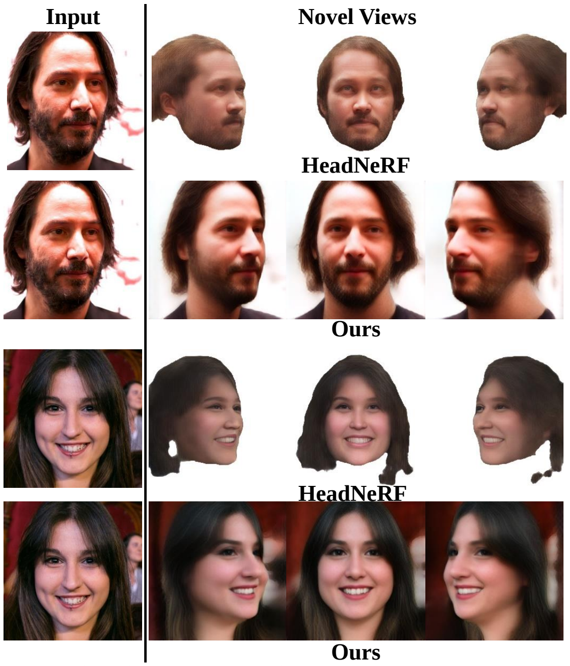

Figure C. **Qualitative comparison with HeadNeRF [7].** Compared to HeadNeRF [7], our method synthesizes more ID-preserving and realistic novel views.

Table A. **Impact of incorporating real-world images.** The **bold** numbers highlight the best results.

Trunc. Ratio	real-world images	$\mathcal{D}_g$	SSIM (↑)	Depth (↓)	ID (↑)
0.5	✗	✓	0.55	0.37	0.16
0.5	✓	✓	0.63	**0.35**	0.35

C.3. More qualitative results

We present additional qualitative results using various reference images from three datasets, including FFHQ [8], CelebAMask-HQ [10] and AFHQv2-Cats [5]. Specifically, for each reference image, we generate a front view and estimate its geometry following the approach described in [3]. To visualize the geometry results, we use ChimeraX [6]. As shown in Fig. D, G-NeRF successfully synthesizes novel views with accurate geometry for a wide range of input images. Notably, our approach can handle inputs with glasses, complex lighting environments, different ages, and varying viewpoints. Although the geometry of AFHQv2-Cats [5] exhibits some hole artifacts due to the limited poses available, we are still able to generate novel views for these cat faces.

C.4. Impact of latent code dimension

The dimension of the latent code has a substantial impact on the overall performance. To evaluate the effectiveness of a larger latent code dimension, we conducted experiments using different dimension sizes. As illustrated in Tab. B, increasing the latent code dimension enhances the model's

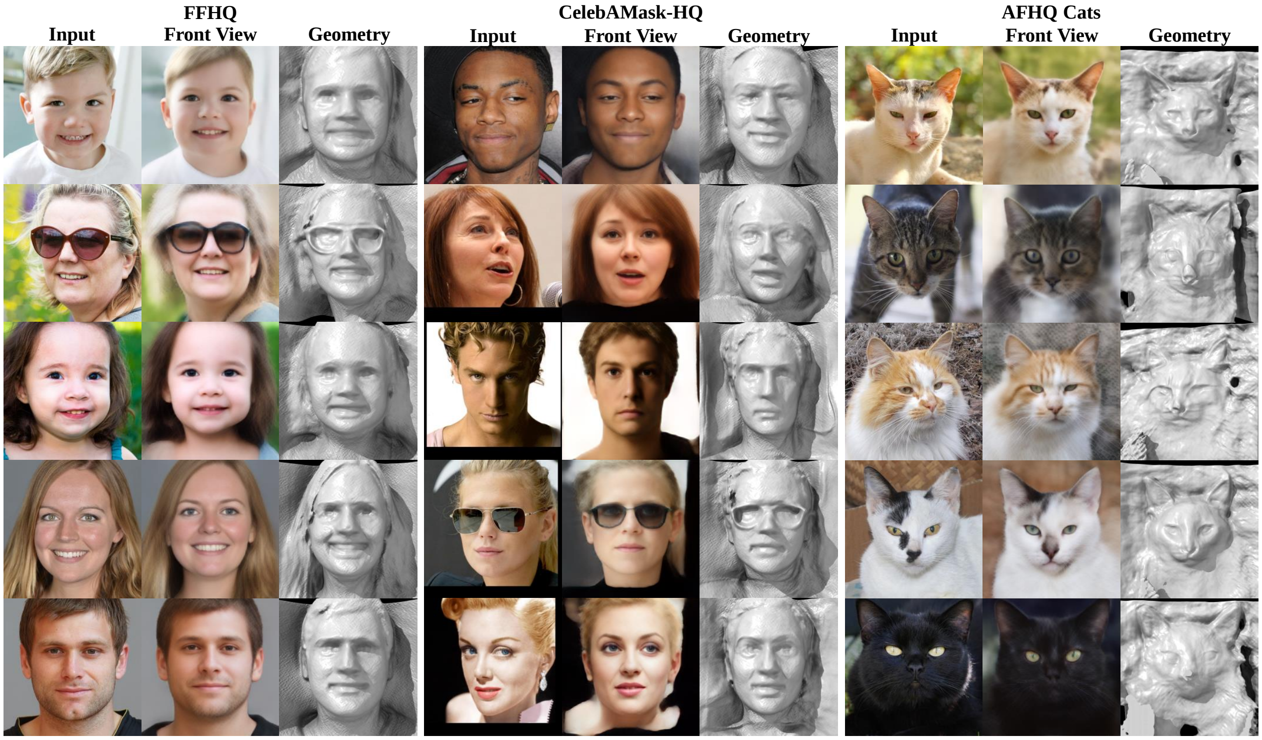

Figure D. **More qualitative results.** We provide front views and geometry generated with various reference images. Our method is capable of synthesizing novel views for diverse input on FFHQ [8], CelebAMask-HQ [10], and AFHQv2-Cats [5].

Table B. Quantitative results of different latent code dimensions. A larger latent dimension can offer increased capacity for models to capture more information, thereby leading to better performance.

Latent code dim.	LPIPS (↓)	Depth (↓)	FID (↓)	KID (↓)
512	0.36	0.36	51.68	3.68
1024	0.35	0.36	49.20	3.66
3072	0.34	0.37	43.10	3.07
5120 (ours)	**0.33**	**0.35**	40.24	2.72
7168	**0.33**	0.36	**39.27**	**2.63**

capability to capture finer details. Nevertheless, once the dimension reaches 5120, the incremental benefits of further expansion become negligible. Taking into account GPU memory consumption, we ultimately decided to set the latent code dimension to 5120.

D. More discussions

D.1. More discussion with concurrent work

In a concurrent study [14], a pre-trained EG3D model is adopted to synthesize a collection of synthetic images for single-shot novel view synthesis of human faces and cat faces. Although this work demonstrates the effectiveness of synthetic face images, it overlooks the fact that models trained solely on synthetic data are susceptible to a gradual decline in either quality (precision) or diversity (recall) [1]. In contrast, our method tackles this issue by incorporating real-world images into training our model. Meanwhile, no experimental results were provided on 360° datasets such as ShapeNet Cars [4, 13]. Therefore, the performance of the proposed method, when trained on this particular dataset, remains uncertain.

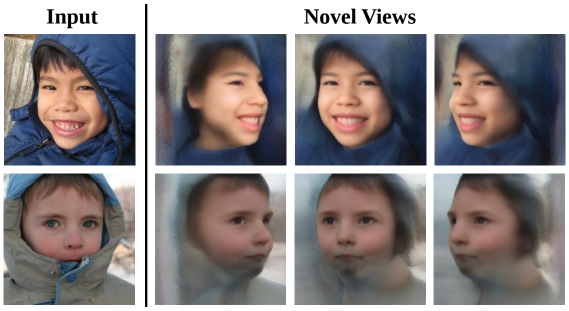

Figure E. **Failure Cases.** Our method may encounter failures when faces are occluded by items such as clothing.

D.2. Potential limitations

In this section, we present an analysis of the failure cases encountered during our experiments. Our method may encounter failures when faces are occluded by items such as clothing, as illustrated in the first two rows of Fig. E. Unlike faces which often have a similar shape, these irregular occlusions present a challenge for our model to capture a common geometry prior. As a result, this occlusion leads to a blurred area around the faces.

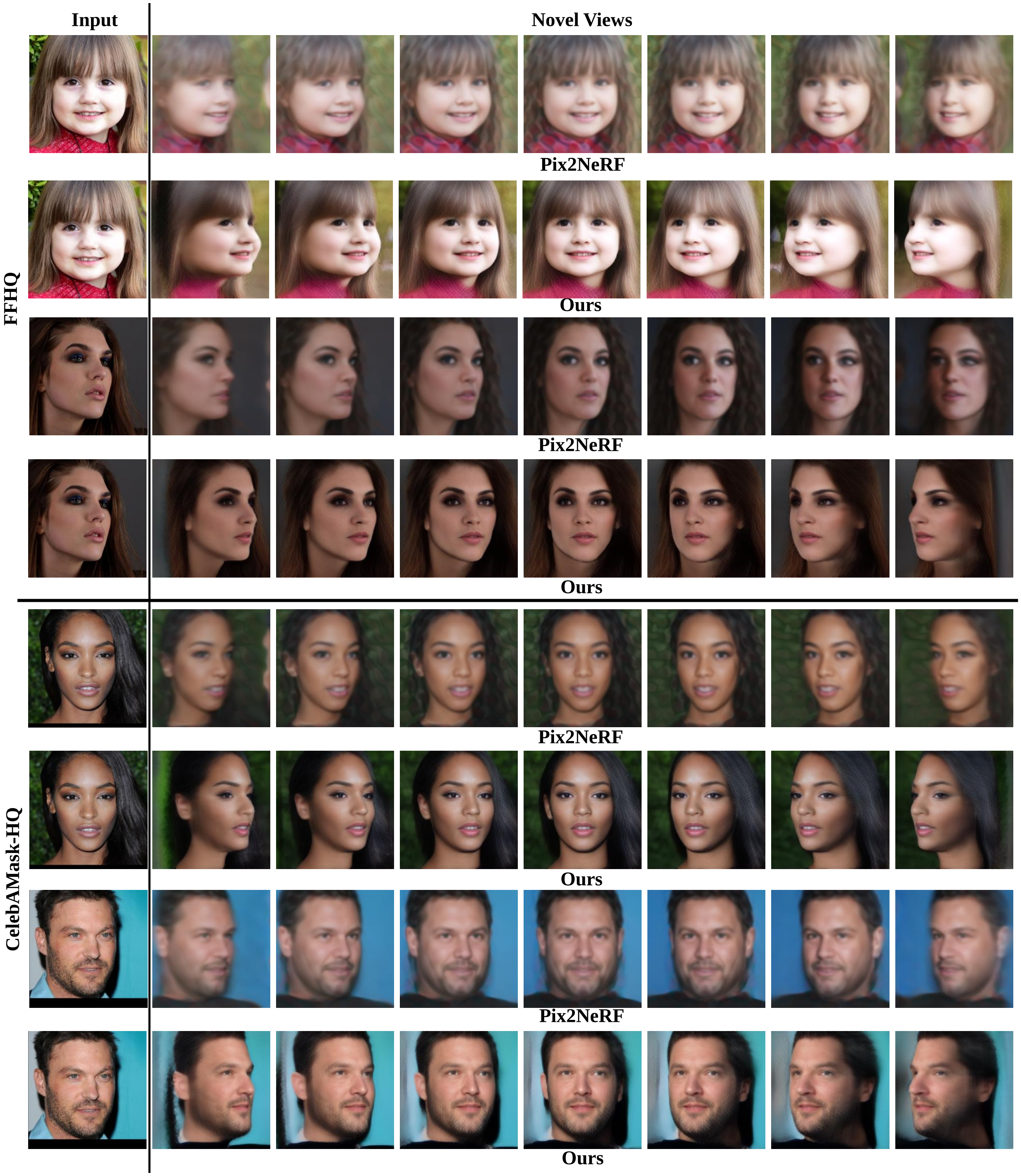

Figure F. More qualitative comparisons with Pix2NeRF [2] on FFHQ [8] and CelebAMask-HQ [10].

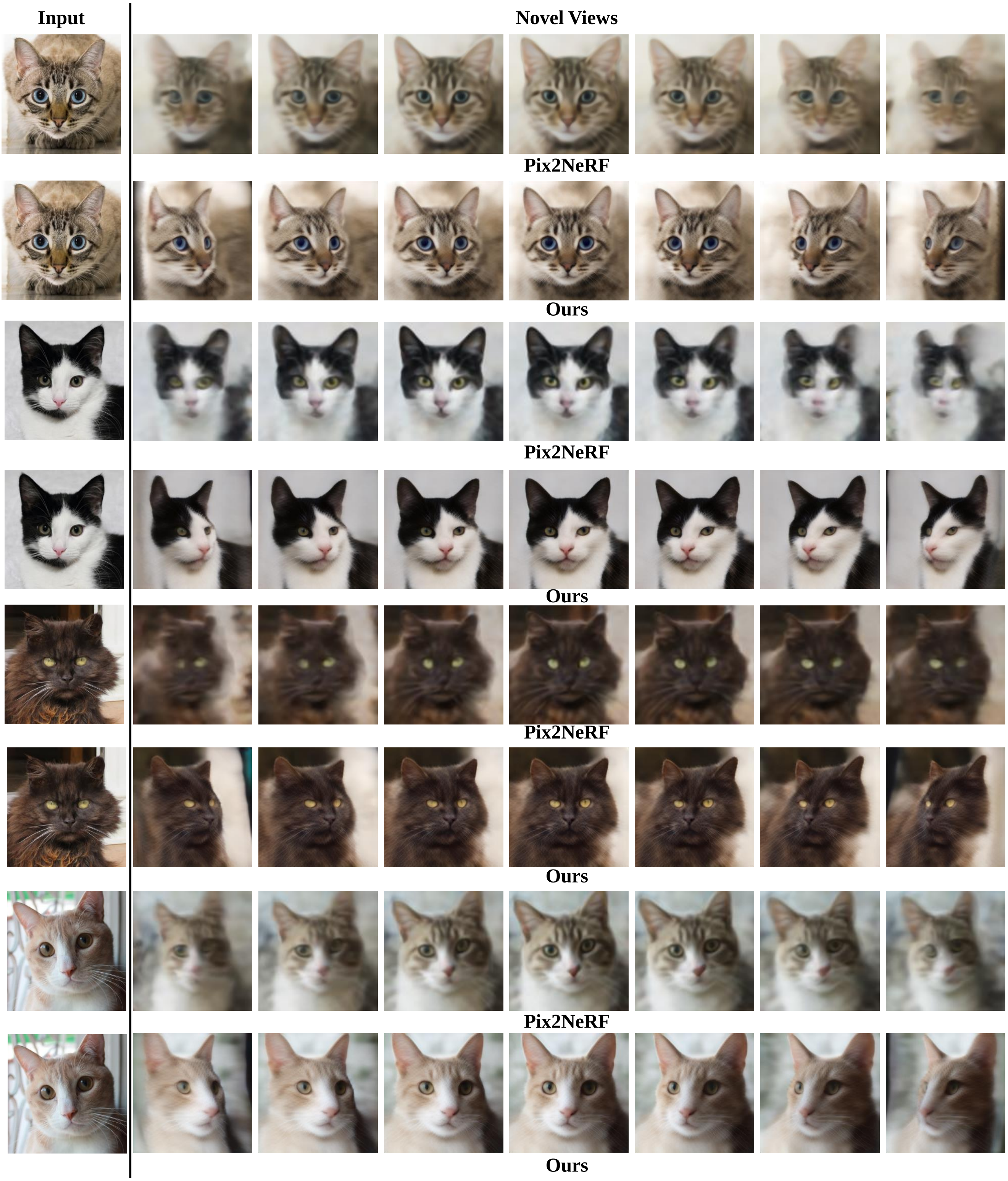

Figure G. More qualitative comparisons with Pix2NeRF [2] on AFHQv2-Cats [5].

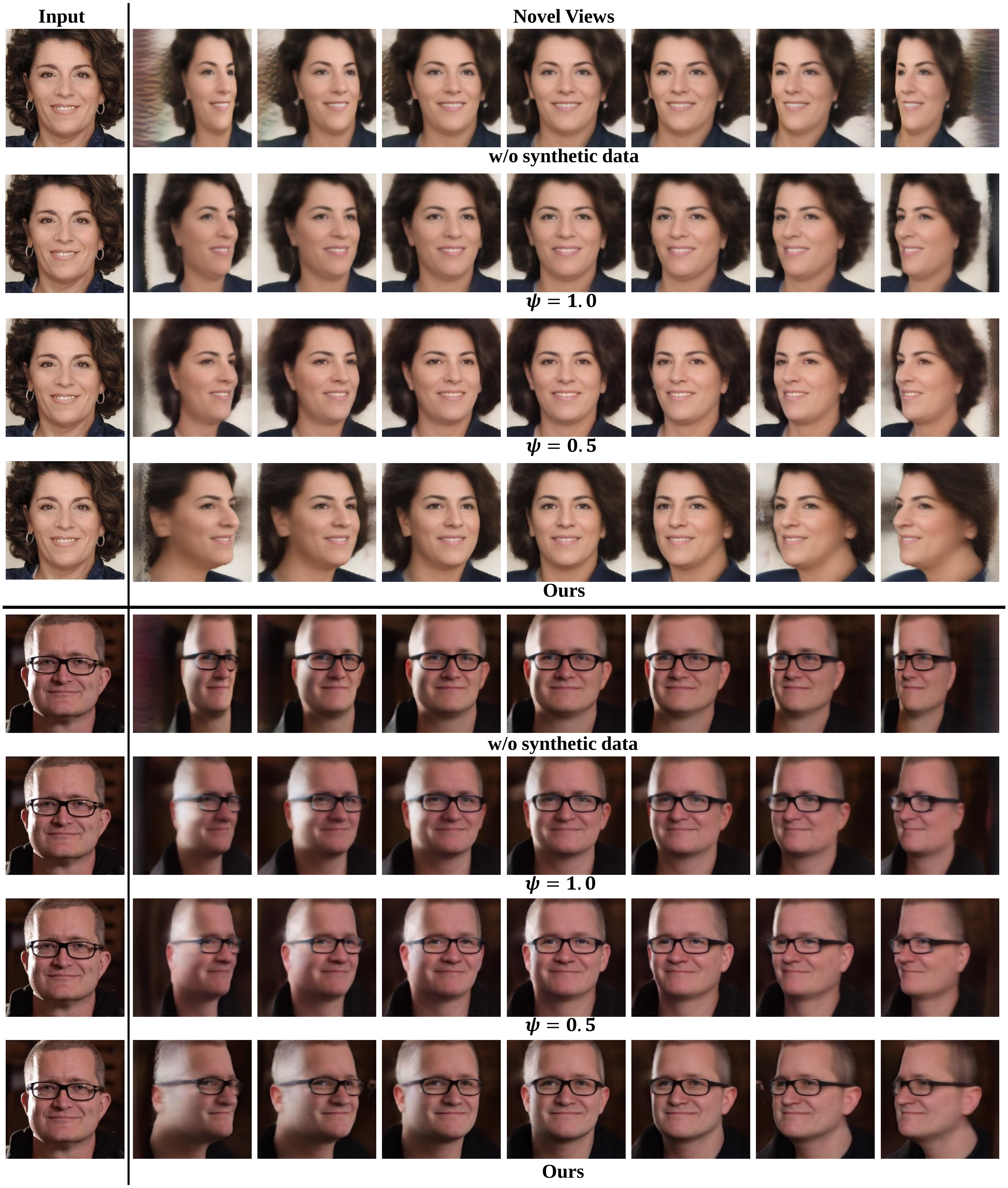

Figure H. More qualitative results of ablation studies.

References

[1] Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G Baraniuk. Self-consuming generative models go mad. *arXiv preprint arXiv:2307.01850*, 2023. 3

[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 Conf. Comput. Vis. Pattern Recog.*, pages 3981–3990, 2022. 1, 4, 5

[3] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 16123–16133, 2022. 1, 2

[4] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2, 3

[5] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8188–8197, 2020. 1, 2, 3, 5

[6] Thomas D Goddard, Conrad C Huang, Elaine C Meng, Eric F Pettersen, Gregory S Couch, John H Morris, and Thomas E Ferrin. Ucsf chimerax: Meeting modern challenges in visualization and analysis. *Protein Science*, 27(1):14–25, 2018. 2

[7] Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. Headnerf: A real-time nerf-based parametric head model. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 2

[8] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4401–4410, 2019. 1, 2, 3, 4

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[10] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5549–5558, 2020. 2, 3, 4

[11] 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 *Eur. Conf. Comput. Vis.*, pages 405–421, 2020. 1

[12] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Adv. Neural Inform. Process. Syst.*, 2019. 1

[13] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Adv. Neural Inform. Process. Syst.*, 2019. 2, 3

[14] Alex Trevithick, Matthew Chan, Michael Stengel, Eric R. Chan, Chao Liu, Zhiding Yu, Sameh Khamis, Manmohan Chandraker, Ravi Ramamoorthi, and Koki Nagano. Real-time radiance fields for single-image portrait view synthesis. In *ACM Trans. Graph.*, 2023. 3

[15] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1492–1500, 2017. 1

[16] Hang Zhou, Yasheng Sun, Wayne Wu, Chen Change Loy, Xiaogang Wang, and Ziwei Liu. Pose-controllable talking face generation by implicitly modularized audio-visual representation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4176–4186, 2021. 1