FFaceNeRF: Few-shot Face Editing in Neural Radiance Fields

Supplementary Material

Overview

In this supplementary material, we present details of model architecture in Section A. Section B covers details of the experiments. Section C presents additional details and explanation of our applications. Section D presents an experiment on dataset scaling. Section E present additional experiments. Lastly, Section F discusses about the social impact.

A. Architecture of Geometry Adapter

Our geometry adapter, Φ_{geo}, consists of a lightweight MLP that receives the viewing direction $v_d \in \mathbb{R}^3$, the Tri-plane feature $\hat{F}'_{tri} \in \mathbb{R}^{32}$, the segmentation label of the geometry decoder $Seg \in \mathbb{R}^{15}$, and the density $\sigma \in \mathbb{R}^1$ in the Figure 2 of the main paper. Therefore, the input dimension is 51, and the output dimension is the number of new mask labels, which in our experiments were 17 for the Base layout and 19 for the Nose and Eyes layouts.

To evaluate whether this simple MLP can effectively fuse different features compared to attention modules, we conducted an additional study to identify any differences. The alternative architecture replaced the first layer of our Φ_{geo} with multi-head self-attention, where the number of heads was set to 4 as shown in the Figure 1. The comparison was conducted under the same conditions as in the ablation study in Section 4.4 of the main paper. The evaluation results show that switching to the multi-head self-attention module did not have a meaningful effect (0.847 mIoU with training on 10 data, compared to 0.850 with our MLP) but did increase the training time. Therefore, we chose the MLP architecture for Φ_{geo}.

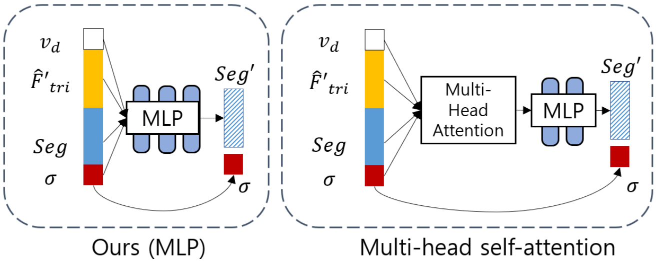

Figure 1. Visualization of our MLP network and multi-head self-attention network which are used as the geometry adapter.

B. Additional Details of Experiments

B.1 Tri-plane Mixing

We conducted tri-plane mixing experiments in Section 4.3 of the main paper and found a balance between the tri-plane mixing ratio and augmentation effectiveness. We considered two variables, mIoU and L1, each representing the retention of semantic information and the variance of information, respectively. Using these variables, we identified the top layers for each variable, as shown in Table 1. For instance, referring to Table 1, when N=2 layers were mixed for the top mIoUs, we combined layers 13 and 14. When N=5 layers were mixed for mIoU, we combined layers 10 through 14 (13, 14, 11, 12, and 10). Similarly, when N=5 layers were mixed for the top L1, we combined layers 1 through 5.

Table 1. List of the top 7 layers in mIoU and L1, respectively

mIoU	L1
13	5
14	1
11	2
12	4
10	3
9	6
8	7

B.2 Baseline Comparisons

We compared our method with two masked based face editing methods: NeRFFaceEditing [3] and IDE-3D [5]. For NeRFFaceEditing, we fine-tuned the model by training only the geometry decoder while keeping the other components fixed. This approach was necessary because training the entire model with only 10 data samples, as in our method, led to overfitting. Similarly, for IDE-3D, we trained only the semantic decoder. Note that the geometry decoder in NeRFFaceEditing and the semantic decoder in IDE-3D serve similar roles, though they are named differently in their respective papers. In addition to the comparison results included in the main paper, we present additional editing results produced by our FFaceNeRF and two baselines, in the upper part of Figure 2.

B.3 Comparisons with Other Editing Methods

Mask-based editing offers distinct advantages over other face editing methods, particularly in its ability to achieve detailed and precise modifications. Unlike point-based,

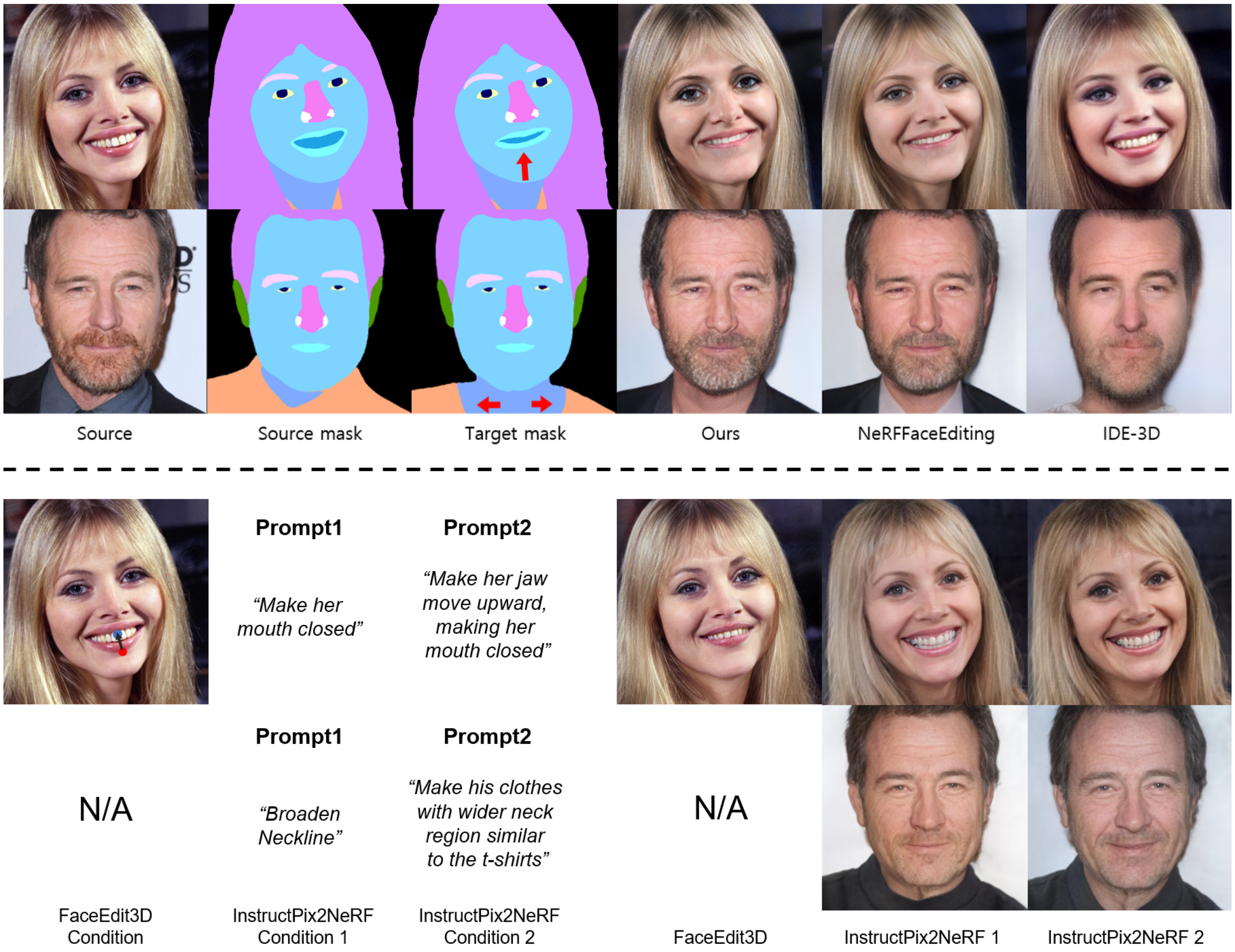

Figure 2. Additional comparison results between our method, two baseline methods, and two additional methods. NeRFFaceEditing showed degraded quality with texture shifts and incomplete editing, particularly on elements like clothing. IDE-3D failed to reconstruct identity, FaceEdit3D failed to edit undefined regions, and InstructPix2NeRF exhibited limited control. In contrast, our method faithfully edited the source image in accordance with the target mask.

text-based, or sketch-based approaches, mask-based methods enable fine-grained control by explicitly defining regions for edits. To demonstrate this, we conducted additional experiments comparing our approach with text-based and point-based editing methods [1, 4].

FaceEdit3D [1] is a point-based editing method that warps the tri-plane to determine the editing direction in the latent space. When using this method, use of excessive control points may lead to distortions. To mitigate this, FaceEdit3D allows to use predefined landmarks only, making edits outside these points impossible. InstructPix2NeRF [4], a state-of-the-art text-based 3D face editing method, can perform edits using natural language prompts. Due to the nature of text prompts, controlling the range or specificity of edits is challenging with this method. SketchFaceNeRF [2] is a sketch based 3D face editing method, that can either generate face from sketch and edit randomly generated face using sketch. However this model does not provide method to edit face from arbitrary face, therefore we randomly generated a face and edit the images to visualize.

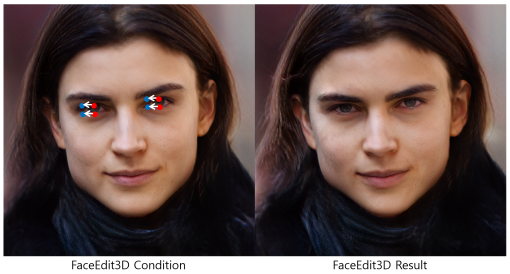

Figure 3. Gaze direction editing results of FaceEdit3D.

Visual comparison results are shown in the lower part of Figure 2. While FaceEdit3D [1] successfully edited the mouth, it failed to edit clothing because there were no control points on the neck or clothing. Similarly, gaze direction

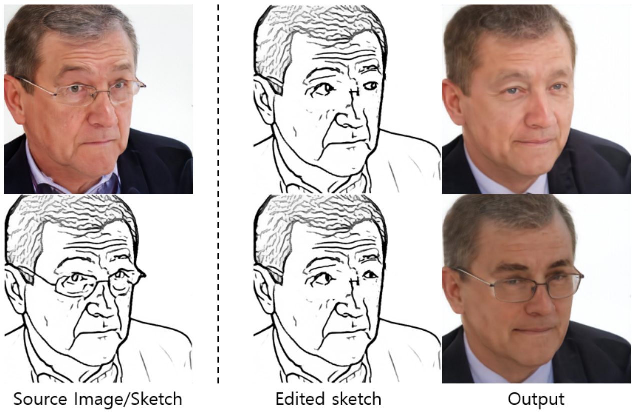

Figure 4. Gaze direction editing results of SketchFaceNeRF. Left shows source image and corresponding sketch, while right shows edited sketch and corresponding output face.

could not be edited, as shown in Figure 3. On the other hand, InstructPix2NeRF [4] was able to generate both samples, but the generated images did not consistently follow the given prompts. Additionally, similar contexts with different prompts often produced inconsistent outputs. Visual results of SketchFaceNeRF are presented in Figure 4. Starting from the randomly generated image (top left), we edited the image using two sketches (middle) to modify the eye gaze while removing the glasses. Although the results on the top right successfully removed the glasses, both trials failed to modify the eye gaze because the pupil is a very small region in the overall sketch, limiting to change the eye gaze using the sketch based model.

B.4 Statistical Analysis

We conducted statistical analysis on user study to verify significance compared to baselines. We conducted a Chi-Square Goodness-of-Fit test followed by a two-proportion z-test as a post-hoc analysis. Null hypotheses for all evaluation metrics were rejected in Chi-Square Goodness-of-Fit test (all p<0.001). The post-hoc analysis was performed only to compare our method with the competitors'. As shown in Figure 5, all differences in choices were statistically significant (p<0.001), denoted by ***.

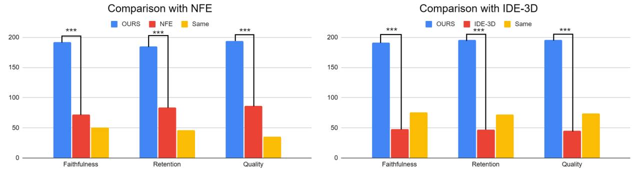

Figure 5. User study analysis results.

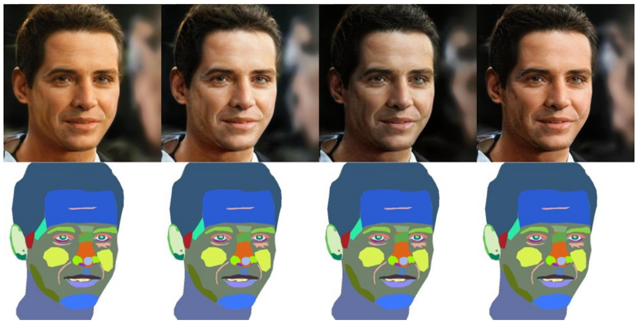

Figure 6. Visualization of augmentation results, which show color changes but preservation of the semantics of the source image.

C. Additional Details of Applications

C.1 Partial Style Transfer

As stated in the main paper, our method can be utilized for a partial style transfer application. We extracted the mean and variance of the tri-plane from style images, and denormalized the source tri-plane for the full style transfer. Here, I indicates the source image and $I_{stylized}$ indicates the fully stylized image. By passing the normalized source tri-plane to Φ_{geo} followed by Ψ_{geo}, we can obtain the resulting mask. A certain mask M_{part} (such as eyes, iris, hair, or mixture of them) can be directly applied to $I_{stylized}$ for partial stylization. This can be written as follows:

$$M_{add} = \sum^{parts} M_{parts}, \; M_{apply} = \text{smooth}_w(M_{add})$$
$$I_{partial} = \sum^{parts} M_{apply} * I_{stylized} + (1 - M_{apply}) * I \tag{1}$$

where M_{add} is simply added masks across parts, smooth_w is linear smoothing applied to the edge of the mask with the width of w, where it was set to 11 in our application. Additional results are presented in Figure 9.

C.2 FFaceGAN

To demonstrate the effectiveness of our geometry adapter, feature injection, and LMTA, we conducted experiments by incorporating an adapter module and LMTA to the DatasetGAN [7], resulting FFaceGAN. For the adapter, we used the same 15 BiseNet [6] labels as our FFaceNeRF for the pretraining of DatasetGAN. After training DatasetGAN with the pretrained network, we froze all the parameters and added a trainable adapter, which receives the output of the original segmentation labels and the StyleGAN features as input. Because DatasetGAN uses a StyleGAN backbone instead of EG3D, there are no view direction or density values. Therefore, we only used the output label of the original segmentation network and the StyleGAN features. We

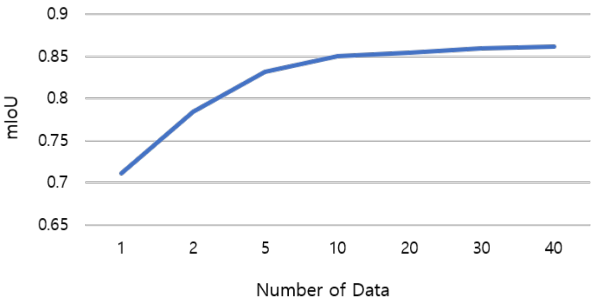

Figure 7. Number of data used for training and resulting mIoU values. The mIoU value increases as the number of training data used becomes larger.

trained our adapter in the same way as the original DatasetGAN.

For the augmentation strategy LMTA, we randomly sampled latent w'^{+} and mixed with w^{+} from training set at the last five layers, as performed for the original FFaceNeRF. To assess whether mixing the last five layers still provides correct augmentation, we added visualizations in Figure 6. The results indicate that the mixing strategy of FFaceNeRF can still be applied to DatasetGAN, changing the color without altering the semantics. The results of FFaceGAN are shown in Figure 10, demonstrating that our adapter and augmentation strategy can effectively enhance performance, even with different architectures.

D. Dataset Scaling

We conducted an additional study to investigate how the number of data used to train our model affects the performance. We varied the number of training data $n \in [1, 2, 5, 10, 20, 30, 40]$ from the Base dataset. As shown in Figure 7, the mIoU value increased as the number of training data increased up to 40. Even with a small dataset of 5 or 10 samples, the performance was already acceptable for use, and it improved incrementally as the number of training samples increased beyond 20.

E. Additioanl experiments

E.1 Additional Data

To further validate our model, we built a new layout called Mouth by adding teeth, jaw, and chin components. The current FFaceNeRF model was pretrained on 15 layouts and supports four different layouts for training and testing: Base, Eyes, Nose, and Mouth. While Base has 17 layouts, Eyes and Nose have 19 layouts, and Mouth has 20 layouts. Results produced using our Mouth layout for accessory adoption and teeth exposure are shown in Figure 8.

Furthermore, our original Base dataset consisted of a training set of 40 examples and a test set of 22 examples. We built an expanded dataset by adding an additional 20 test examples to ensure robust testing. In addition to the original dataset, we made this additional dataset publicly available.

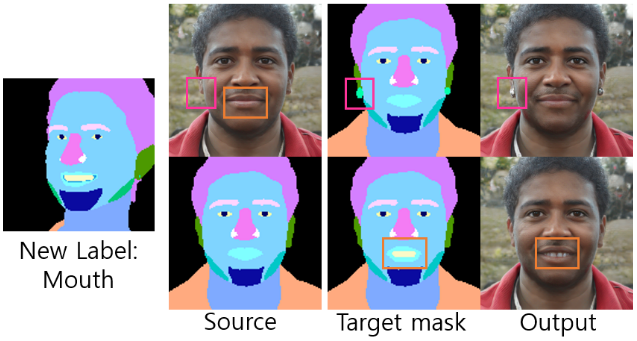

Figure 8. Editing results trained on new label ”Mouth”. Top: Accessories adoption, Bottom: Teeth.

Table 2. Quantitative results of ablation study on mask generation with a different number of training data. The highest scores are denoted in bold.

Number of Data	1	5	10	20	30
Ours	0.711	**0.832**	**0.850**	**0.855**	**0.860**
w/o injection	**0.741**	0.806	0.835	0.844	0.847

F. Social Impact

The introduction of FFaceNeRF presents a substantial positive impact on fields such as personalized medical imaging, virtual reality, and the creative arts by allowing precise and customizable 3D face editing with minimal training data. However, the potential for misuse, such as in the creation of identity manipulation, underscores the need for careful and responsible usage. While the overall societal impact of FFaceNeRF is positive, we will include appropriate ethical guidelines and safeguards to our released code. This will ensure that the technology is used to drive innovation and enhance user experiences while minimizing potential risks and protecting individuals’ privacy and security.

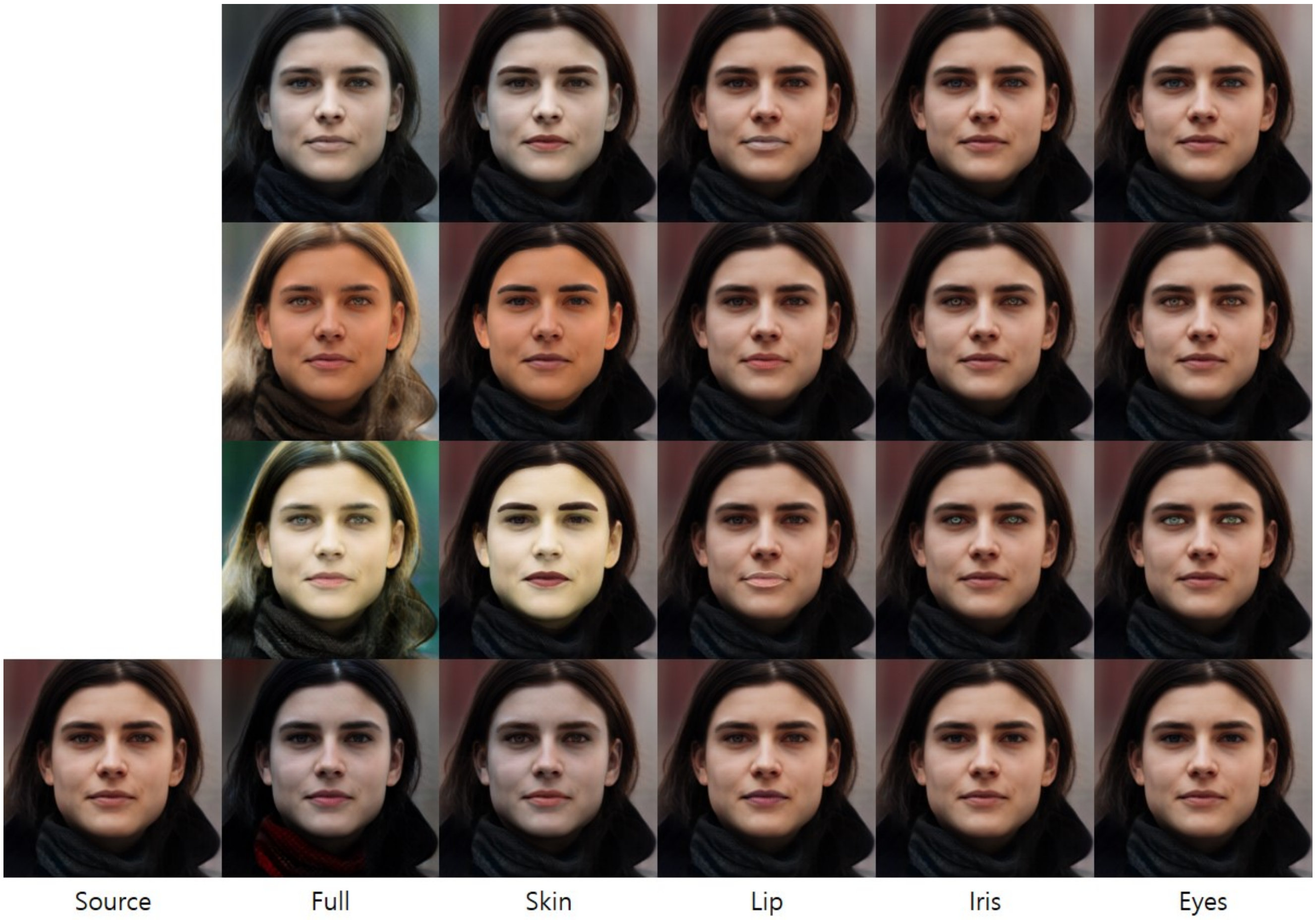

Figure 9. Visualization of partial style transfer.

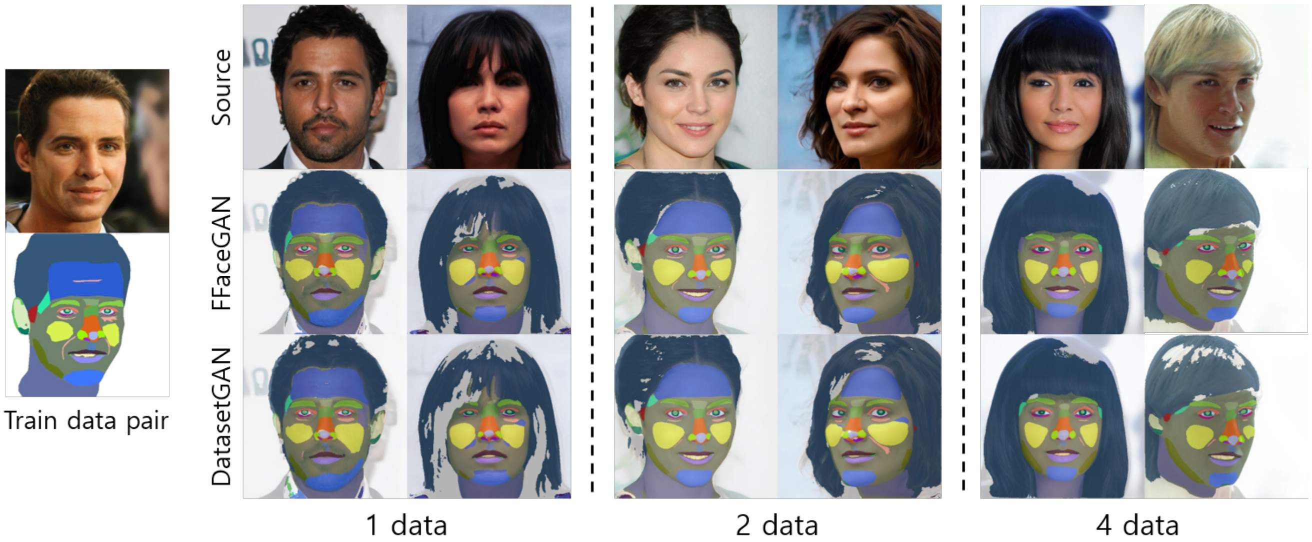

Figure 10. Comparison of results produced by our FFaceGAN and the original DatasetGAN. This indicates effectiveness of our adapter and augmentation strategy.

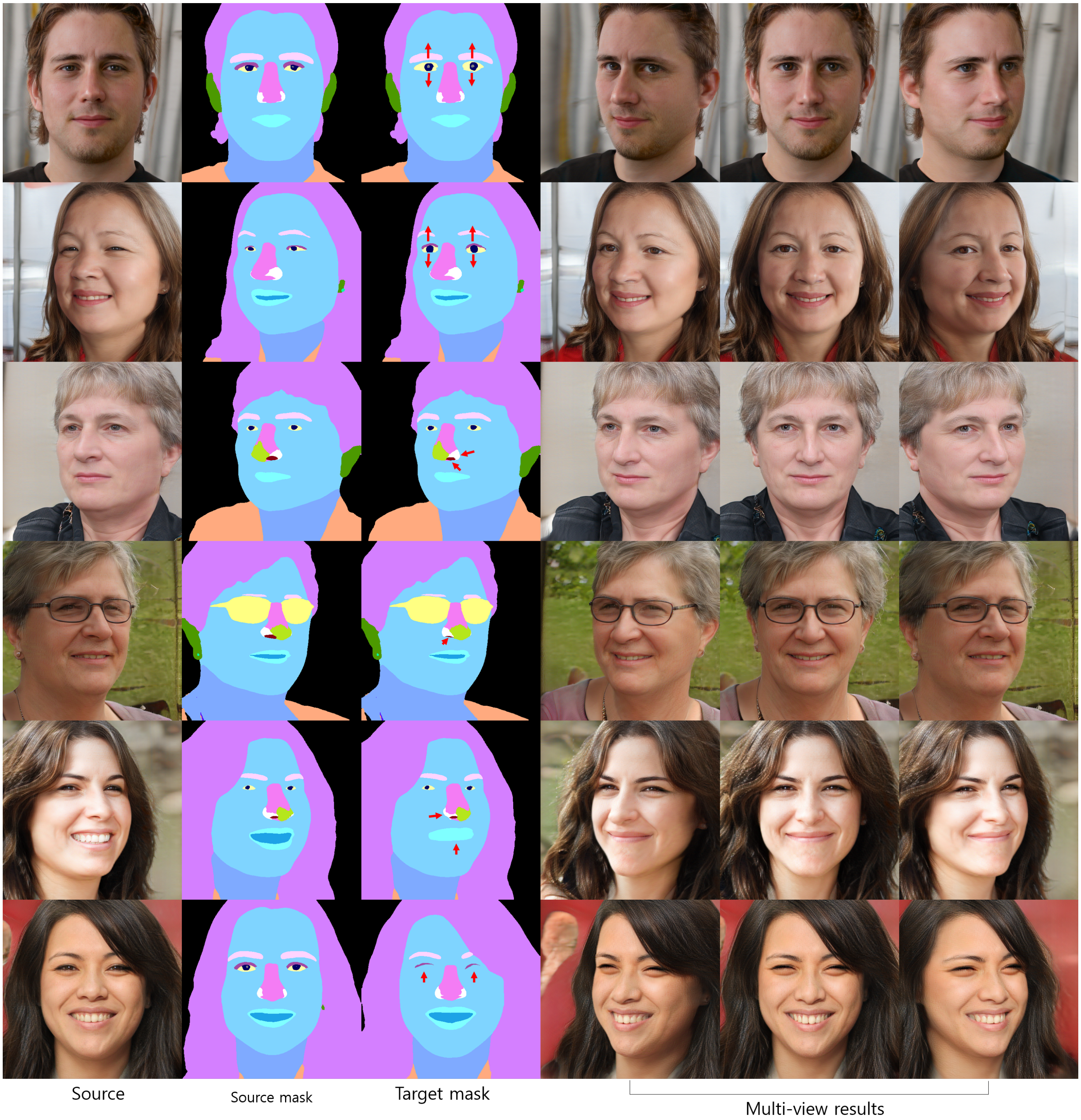

Figure 11. Additional results of FFaceNeRF in multi-view.

References

[1] Yuhao Cheng, Zhuo Chen, Xingyu Ren, Wenhan Zhu, Zhengqin Xu, Di Xu, Changpeng Yang, and Yichao Yan. 3d-aware face editing via warping-guided latent direction learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 916–926, 2024. 2

[2] Lin Gao, Feng-Lin Liu, Shu-Yu Chen, Kaiwen Jiang, Chunpeng Li, Yukun Lai, and Hongbo Fu. Sketchfacenerf: Sketch-based facial generation and editing in neural radiance fields. *ACM Transactions on Graphics*, 42(4), 2023. 2

[3] Kaiwen Jiang, Shu-Yu Chen, Feng-Lin Liu, Hongbo Fu, and Lin Gao. Nerffaceediting: Disentangled face editing in neural radiance fields. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022. 1

[4] Jianhui Li, Shilong Liu, Zidong Liu, Yikai Wang, Kaiwen Zheng, Jinghui Xu, Jianmin Li, and Jun Zhu. Instruct-pix2neRF: Instructed 3d portrait editing from a single image. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 3

[5] 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. *ACM Transactions on Graphics (ToG)*, 41(6):1–10, 2022. 1

[6] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 325–341, 2018. 3

[7] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10145–10155, 2021. 3