Supplementary Material
TriPlaneNet: An Encoder for EG3D Inversion

Ananta R. Bhattarai Matthias Nießner Artem Sevastopolsky

Technical University of Munich (TUM)

A. Implementation Details

Dataset Details. Our model is trained on a combination of real images from FFHQ and generated samples from EG3D. We extract the camera pose and pre-process the FFHQ and synthetic data in the same way as in [3]. Since the pre-processing technique could not identify the camera poses of 4 images, we skipped the quantitative evaluation of 4 images for all the methods presented in the paper. We also augment the training dataset by mirroring it. As shown in the main text, adding EG3D samples makes the model robust to the input image shifts. The synthetic training samples are generated via sampling latent codes z for EG3D with no truncation ($\psi = 1$) often applied for large-scale GANs [2], thus including the hard samples. In order to match the camera pose distribution in the FFHQ, we generate EG3D samples with randomly sampled camera poses from FFHQ and their flipped versions. In Table 1, we demonstrate the dependence of the reconstruction quality on CelebA-HQ on the number of synthetic samples created by EG3D in advance added to the dataset.

Experiment settings. For training our models, we adopt the same training configuration from [15] except for some minor modifications. In particular, we train the second branch only after 20K steps and then train both branches until 500K. Afterward, we freeze the first branch and fine-tune the second branch until 1.5M steps. In each training step, we re-render the batch of input images from the same view and the mirror view. Then, we compute input view reconstruction losses using same-view rendered images and mirror-view losses $\mathcal{L}_m$ using mirror-view rendered images. We operate in the resolution of 256×256 except for the calculation of $\mathcal{L}_{id}$. The region around the face is cropped and resized to 112×112 before feeding into the face recognition network [4] to calculate $\mathcal{L}_{id}$. The models are trained with a batch size of 3. We use the Ranger optimizer that combines Rectified Adam [14] with the Lookahead technique [22] and set a learning rate to 0.0001. The models are trained using a single NVIDIA GeForce RTX A6000 GPU.

Loss functions. As outlined in the main text, we train our models using same-view reconstruction and mirror-view losses. The loss function for our first branch latent encoder $\phi(.)$ is defined as:

$$\mathcal{L}_\phi(x, x_m, \hat{y}, \hat{y}_m) = \mathcal{L}_{rec}(x, \hat{y}) + \lambda_m \mathcal{L}_m(x_m, \hat{y}_m) \quad (1)$$

where $x_m = \text{flip}(x)$, $\mathcal{L}_{rec}(x, \hat{y})$ is defined as

$$\mathcal{L}_{rec}(x, \hat{y}) = \lambda_1 \mathcal{L}_2(x, \hat{y}) + \lambda_2 \mathcal{L}_{\text{LPIPS}}(x, \hat{y}) + \lambda_3 \mathcal{L}_{\text{id}}(x, \hat{y}) \quad (2)$$

and $\mathcal{L}_m(x_m, \hat{y}_m)$ is a *probably symmetric prior* defined as

$$\mathcal{L}_m(x_m, \hat{y}_m) = \lambda_4 \mathcal{L}_{\text{symm}}(x_m, \hat{y}_m, \sigma(x_m)) + \lambda_5 \mathcal{L}_{\text{LPIPS}}(x_m, \hat{y}_m) + \lambda_6 \mathcal{L}_{\text{id}}(x_m, \hat{y}_m) \quad (3)$$

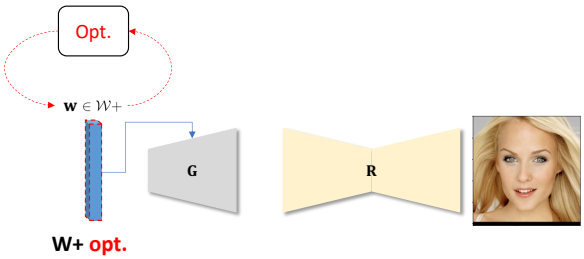
The main text shows that $\mathcal{L}_m$ significantly improves the embedding in 3D space. However, directly applying $\mathcal{L}_2$ between the mirror-view image and the surrogate mirrored image is not applicable since human faces are not perfectly symmetric. Therefore, following the practice outlined in [19], we construct $\mathcal{L}'_{\text{symm}}(x_m, \hat{y}_m, \sigma(x_m))$ as a penalty between mirrored image x_m and reconstruction for the mirror image $\hat{y}_m$ weighted by a pixel-wise uncertainty map $\sigma(x_m)$ computed for each pixel and taking an average. Mathematically,

$$\begin{aligned} &\mathcal{L}'_{\text{symm}}(x_m, \hat{y}_m, \sigma(x_m)) \\ &= -\frac{1}{|\Omega|} \sum_{uv \in \Omega} \log \frac{1}{\sqrt{2}(\sigma(x_m))_{uv}} \exp - \frac{\sqrt{2}\ell_{1,uv}}{(\sigma(x_m))_{uv}} \\ &= \log(\sqrt{2}) + \frac{1}{|\Omega|} \sum_{uv \in \Omega} \log(\sigma(x_m))_{uv} + \frac{\sqrt{2}\ell_{1,uv}}{(\sigma(x_m))_{uv}} \end{aligned} \quad (4)$$

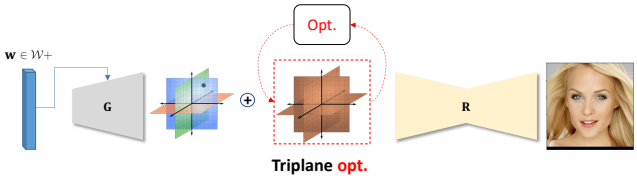
where $\ell_{1,uv}$ is the $\mathcal{L}_1$ distance between the intensity of pixels at location uv, and $\sigma(x_m)$ is estimated by the neural network for image x_m. We can interpret the loss function as the negative log-likelihood of a factorized Laplacian distribution on the reconstruction residuals. We take pre-trained network from [19] for predicting uncertainty map $\sigma(x_m)$

Table 1. Quantitative ablation study over the number of synthesized EG3D samples in the training set.

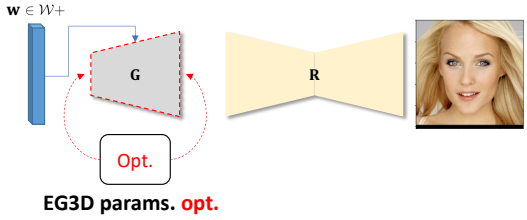
Our Method	MSE ↓	LPIPS ↓	MS-SSIM ↑	Depth ↓	ID ↑						
					Same View	Novel View (Yaw angle in radians)					
						-0.8	-0.6	-0.3	0.3	0.6	0.8
... w/ 0 EG3D samples	0.022	0.09	0.86	**0.044**	**0.70**	**0.41**	**0.48**	**0.60**	**0.60**	**0.50**	**0.42**
... w/ 10K EG3D samples	0.020	0.09	0.86	0.048	0.68	0.39	0.46	0.58	0.58	0.48	0.40
... w/ 50K EG3D samples	0.021	0.09	0.86	0.047	0.68	0.39	0.47	0.58	0.59	0.48	0.40
... w/ 100K EG3D samples	**0.019**	**0.08**	**0.87**	0.051	0.68	0.39	0.47	0.58	0.59	0.48	0.40
... w/ 150K EG3D samples	0.021	0.09	0.86	0.053	0.66	0.37	0.44	0.56	0.57	0.47	0.40



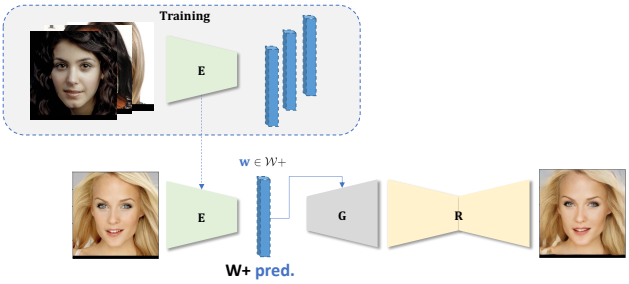
(a) $\mathcal{W}+$ opt.



(b) Tri-plane opt.



(c) EG3D params opt.



(d) $\mathcal{W}+$ pred. (pSp [15])

Figure 2. Overview of our baseline approaches. $\boldsymbol{G}(\cdot)$ and $\mathcal{R}(\cdot)$ stand for EG3D generator and renderer blocks respectively. Hybrid approaches described in the main text constitute the combination of the techniques shown above, applied sequentially, one after another. For instance, PTI [16] sequentially performs (a) and then (c)., and $\mathcal{W}+$ pred. + tri-plane opt. sequentially performs (d) and then (b).

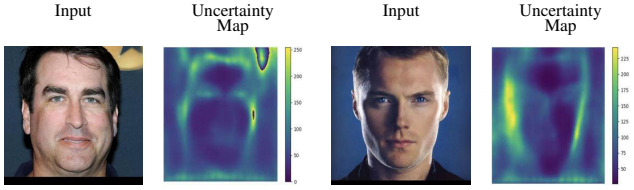


Figure 1. Uncertainty map σ predicted by the pre-trained network from [19] for the given input. The network assigns higher uncertainty to regions such as ears, hair, and background where the symmetry assumption fails.

and replace $\mathcal{L}_1$ distance with $\mathcal{L}_2$ in (4). Therefore, we reformulate $\mathcal{L}'_{\text{symm}}(x_m, \hat{y}_m, \sigma(x_m))$ as

$$\mathcal{L}_{\text{symm}}(x_m, \hat{y}_m, \sigma(x_m)) = \frac{1}{|\Omega|} \sum_{uv \in \Omega} \frac{\ell_{2,uv}}{(\sigma(x_m))_{uv}} \tag{5}$$

However, we can also train the prediction network from scratch, optimizing the likelihood in (4). $\sigma(x_m)$ assigns lower confidence to the region in the mirrored image x_m where the symmetry assumption fails (see Fig. 1). The uncertainty map predictor network is an encoder-decoder architecture that operates in the 64 × 64 resolution. Therefore, we resize the image to 64 × 64 before feeding into this network and upsample the output back to 256 × 256 for calculating $\mathcal{L}_{\text{symm}}$.

We use AlexNet [12] to extract features for the $\mathcal{L}_{\text{LPIPS}}$

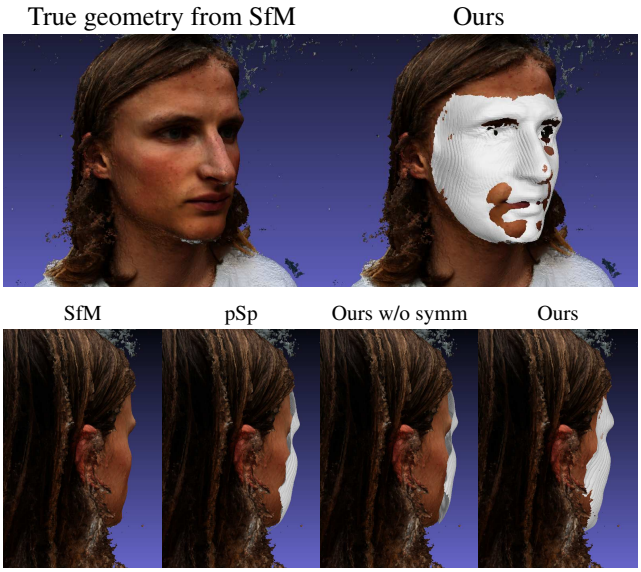


Figure 3. Overlay of the ground truth SfM mesh and predicted meshes. As observed in the bottom row (view from behind), *Ours* provides the tightest fit to the ground truth SfM mesh. PTI and SPI produce unnaturally wide meshes that mostly lie inside the true geometry, except for the nose region (the discrepancy can be better observed in Fig. 4).

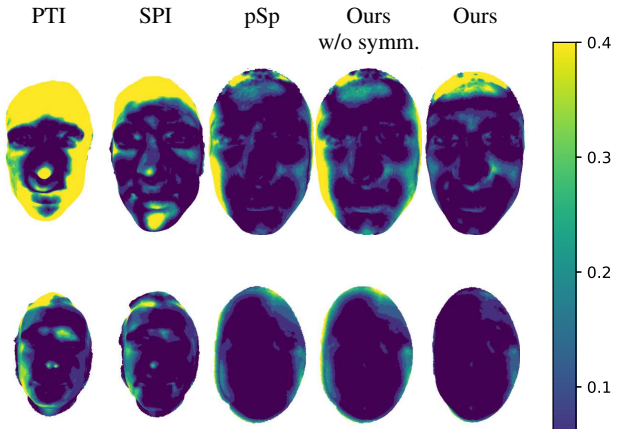


Figure 4. Absolute distance (fraction of inter-ocular dist.) from the predicted meshes points to their nearest neighbors in the SfM mesh (top row: Subject #1; bottom row: Subject #2). Lower values (dark blue) indicate that the predicted mesh is closer to the true geometry, while higher values (yellow) indicate that the predicted mesh is too far (inside or outside) from the true geometry.

loss. Similarly, $\mathcal{L}_{\text{id}}$ is computed by measuring the cosine similarity between the input image and the output with a pre-trained ArcFace [4] network. We set the weight of each component in the loss function as follows: $\lambda_m = 0.1$, $\lambda_1 = \lambda_4 = 1.0$, $\lambda_2 = \lambda_5 = 1.0$ and $\lambda_3 = \lambda_6 = 0.1$. Analogously, we construct the loss for the second branch $\mathcal{L}_\psi$ by replacing $\mathcal{L}_2$ with $\mathcal{L}_1$ smooth loss in (2), inside L_{symm} in (3) and first branch outputs $\hat{y}$ and $\hat{y}_m$ with the second branch outputs y and y_m. We use the same weight for each component.

First branch architecture. To implement the latent encoder, we adopt the design of the pSp encoder from [15]. As the EG3D generator expects 14 style vectors for the selected resolution, we modify the pSp architecture to output 14 style vectors instead of 18. We employ IR-SE-50 [4] pre-trained for face recognition for the backbone network.

Second branch architecture. The tri-plane offsets predictor consists of an encoder and a decoder network, a typical U-Net [17] architecture. The encoder backbone is an IR-SE-50 [4] pre-trained on face recognition, accelerating convergence. We adopt the design of the RUNet [8] for the decoder with some minor modifications. Instead of using ReLU as in RUNet, we use PReLU [7] with a separate α for each input channel and an initial value of 0.25. Like RUNet, every step in the decoder path consists of upsampling, concatenation, and convolution operations. Upsamping of the feature map is performed with a PyTorch nearest neighbor upsample layer (*torch.nn.Upsample*). Then, it is followed by a concatenation with the intermediate feature maps from the encoder path. The intermediate features are extracted from the encoder's 3rd, 7th, 21st, and 22nd layers. Finally, batch normalization, 3×3 convolution, PReLU, 3×3 convolution, and PReLU are applied sequentially. The final step in the decoder path takes the concatenation of first branch tri-plane features with upsampled features from the previous step as an input and outputs $256 \times 256 \times 96$ tri-plane offsets. The final step applies 3×3 convolution, PReLU, 3×3 convolution, PReLU, and 1×1 convolution operations sequentially.

B. Novel view rendering of videos

We demonstrate an application of our method to render in-the-wild videos from a novel view. In Fig. 7, frames of a video with a person talking and their rendering from a fixed novel view in the EG3D space are presented. The background in the video was removed by a matting network [10]. The encoder is capable of representing tiny details of in-the-wild portrait imagery in 3D and supports complex facial expressions.

C. Facial Manipulation

To perform image editing, we first obtain the latent code $w \in \mathcal{W}$ of the input image via optimization. Since $\mathcal{W}$ space offers more editing power than $\mathcal{W}+$ [18], we select $\mathcal{W}$ space for our experiments. We then obtain the final inversion using our second branch by replacing first

branch components with components obtained using optimization. Given the latent code $w \in \mathcal{W}$, tri-plane features ($\boldsymbol{G}(w) + \Delta\boldsymbol{T}$) and edited latent code w_{edit}, we can render edited image with camera matrix π by $\mathcal{R}(\boldsymbol{T}_{edit}, \pi)$. Inspired by [20], we perform following operation to obtain $\boldsymbol{T}_{edit}$:

$$\boldsymbol{T}_{edit} = (\boldsymbol{G}(w) + \Delta\boldsymbol{T}) + \boldsymbol{G}(w_{edit}) - \boldsymbol{G}(w) \quad (6)$$

We take two editing directions, smile, and age, from the official implementation of [11], obtained using GANspace [6] and show editing results in Figs. 8 and 9. Note that our first branch latent encoder could be modified slightly to embed the input image in $\mathcal{W}$ instead of space $\mathcal{W}+$ as done in [5] and [1]. We demonstrate, however, that modification of the tri-plane features that would correspond to a certain semantic direction is possible and leave the research on the most plausible face manipulation for both input images and videos as a suggestion for future work.

D. Discussion of the baselines design

In Fig. 2, we provide a visual overview of the baseline designs used for the analysis of PTI [16] and tri-plane offsets behavior. One-stage inversion techniques can be divided into two approaches: optimization-based and based on an encoder prediction. Two-stage inversion techniques involve inference followed by fine-tuning of some of the generator parameters. Similarly, these parameters can be fine-tuned via optimization or by encoder prediction. Since our method involves the prediction of the tri-plane offsets and avoids fine-tuning the generator parameters, we also consider the baseline where the tri-plane offsets in the second stage are optimized. In the main paper text, we demonstrate the design of different hybrid two-stage inversion approaches that combine both optimization and prediction.

For $\mathcal{W}+$ opt., we optimize the latent code $w \in \mathcal{W}+$ for 1K steps following [9]. The $\mathcal{W}+$ pred. constitutes the baseline with the latent code $w \in \mathcal{W}+$ predicted by pSp encoder [15]. For EG3D params opt., we apply the second stage of PTI from [16] and optimize for 1K steps. To optimize for the tri-plane offsets (tri-plane opt.), we use L-BFGS [13] as the optimizer and employ combination of L_2 or LPIPS [23] with regularization term (L_2 and LPIPS discrepancy with the first branch prediction) as a loss objective. We run the optimization for 50 steps. As L-BFGS approximates the Hessian by calculating several estimates in a single step, 50 steps take equivalently 1K gradient evaluations.

E. Geometric evaluation

In order to evaluate how well the method embeds a head into 3D without any information about the head's geometry, we compare the prediction to the true head geometry constructed by a Structure-from-Motion method (see paragraph "**Geometry evaluation for a multi-view sequence**" in Sec. 4.2) for two subjects. The reconstruction is based on a 360° DSLR capture, while the methods make predictions for the image of the sequence with the head pose closest to the straight frontal. The sequence for subject #1 is the same as demonstrated in Fig. 7 in the main text. We rigidly align the meshes by 5 eyes, nose, and mouth landmarks to the SfM mesh and analyze the proximity of each predicted mesh to the SfM mesh in the face region (the bounding region is defined as an ellipsoid in 3D with the same location and size for all methods). We deliberately only select 5 landmarks for alignment to analyze the shape correctness of the parts not fully visible in the frontal image, such as cheeks. In Fig. 3, we demonstrate the overlay of the mesh predicted by TriPlaneNet and true SfM mesh, as well as a comparison to the meshes obtained from other methods. As shown in the view behind, parts with only partial presence in the frontal view get predicted more correctly by our method; Fig. 4, and Table 2 demonstrate that analytically via pixel-wise proximity to the SfM mesh.

	AD ↓				
	PTI	SPI	pSp	Ours w/o symm. prior	Ours
Subject #1	0.971	0.360	0.115	0.167	**0.090**
Subject #2	0.032	0.024	0.016	0.019	**0.009**

Table 2. Comparison of the average absolute distance (AD) from the mesh for various methods' predictions to the true geometry from SfM.

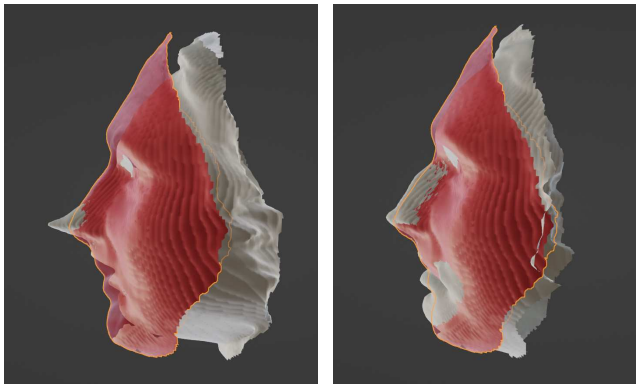


Figure 5. Overlay of the shape predicted by our method (red) and: PTI (left image, gray) and SPI (right image, gray). Here, we demonstrate that these shapes, especially for PTI, are typically wider in the side projection than for our method, which introduces a discrepancy with the true geometry (see Fig. 4). The same effect is demonstrated in Fig. 6 in the main text.

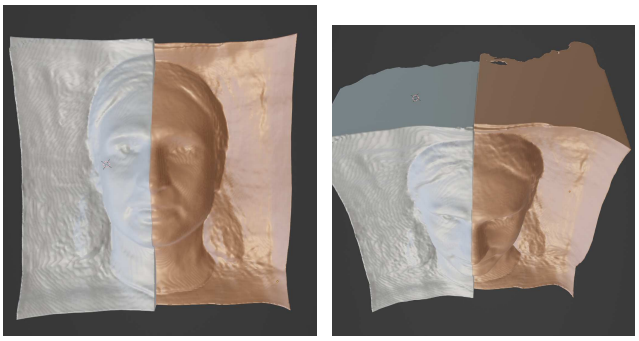


Figure 6. Side-by-side comparison of *Ours* (golden, right half) and *Ours w/o symm. prior* (gray, left half). We observe that the methods produce similar geometry in the facial region, while the background "cutting plane" of the mesh is closer to the face w/o symm. prior. This way, the symm. prior allows to extend the modeled region. *Zoom-in recommended.*

F. Additional Qualitative Results

In this section, we present additional qualitative results on same-view inversion, novel view rendering and ablation for the loss, and architecture change.

- Figs. 10, 11, 12 and 13 provide qualitative comparison of our approach with existing state-of-the-art inversion techniques on image reconstruction.
- Figs. 16, 14, 15, 17, and 18 demonstrate qualitative comparison of our approach with existing state-of-the-art inversion techniques on novel-view rendering.
- Figs. 19 and 20 reflect extensive qualitative ablation studies for the loss and architecture changes.

Figure 7. Novel view synthesis of frames extracted from a talking head video. The novel view yaw angles relative to the frontal view are as follows: second row (-0.3 radians), fourth row (0.0 radians), sixth row (-0.3 radians), eighth row (0.6 radians), and last row (0.8 radians). *Electronic zoom-in recommended.*

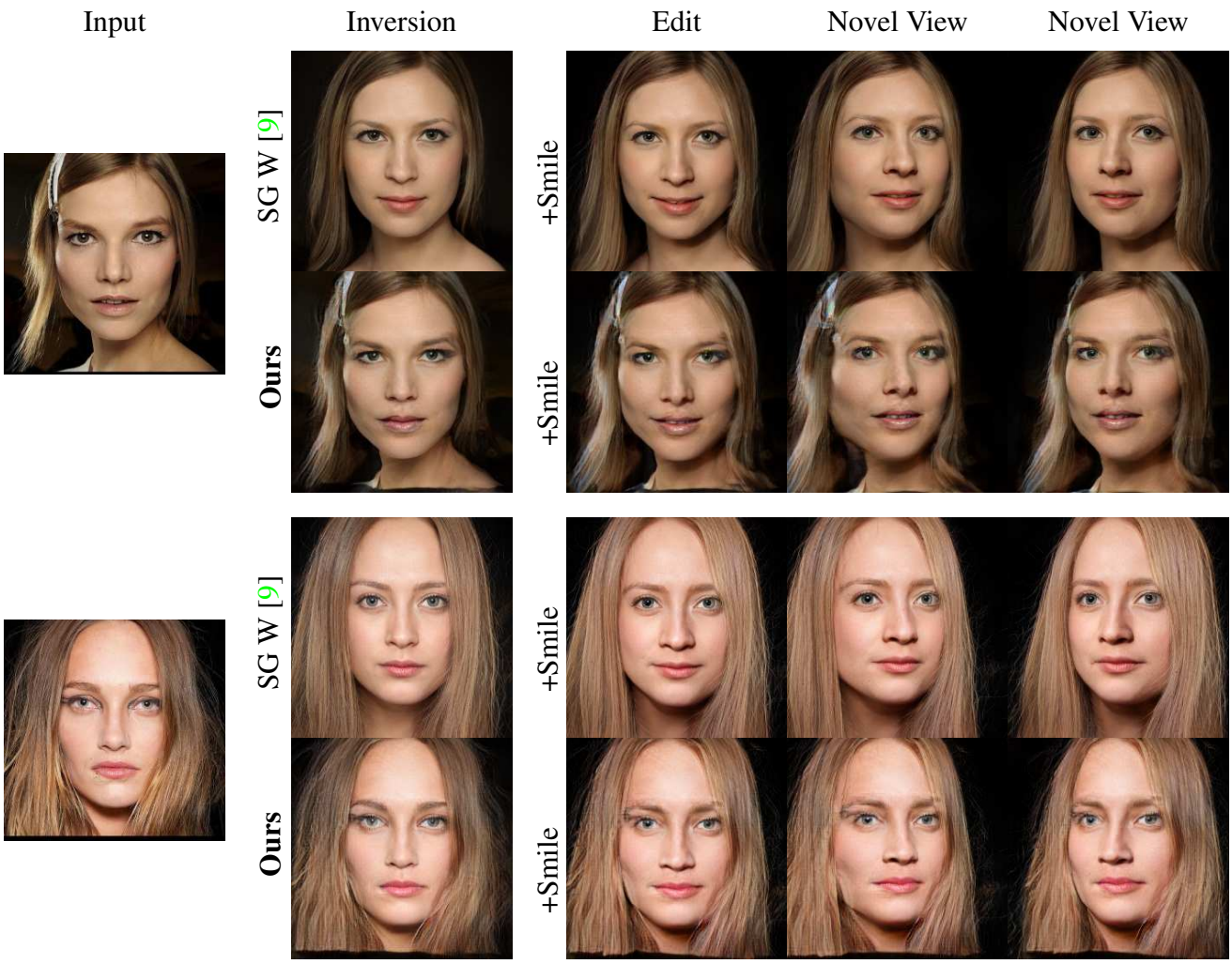


Figure 8. Editing result for smile attribute. By utilizing tri-plane features, our method preserves identity more and also generates both realistic and view-consistent editing.

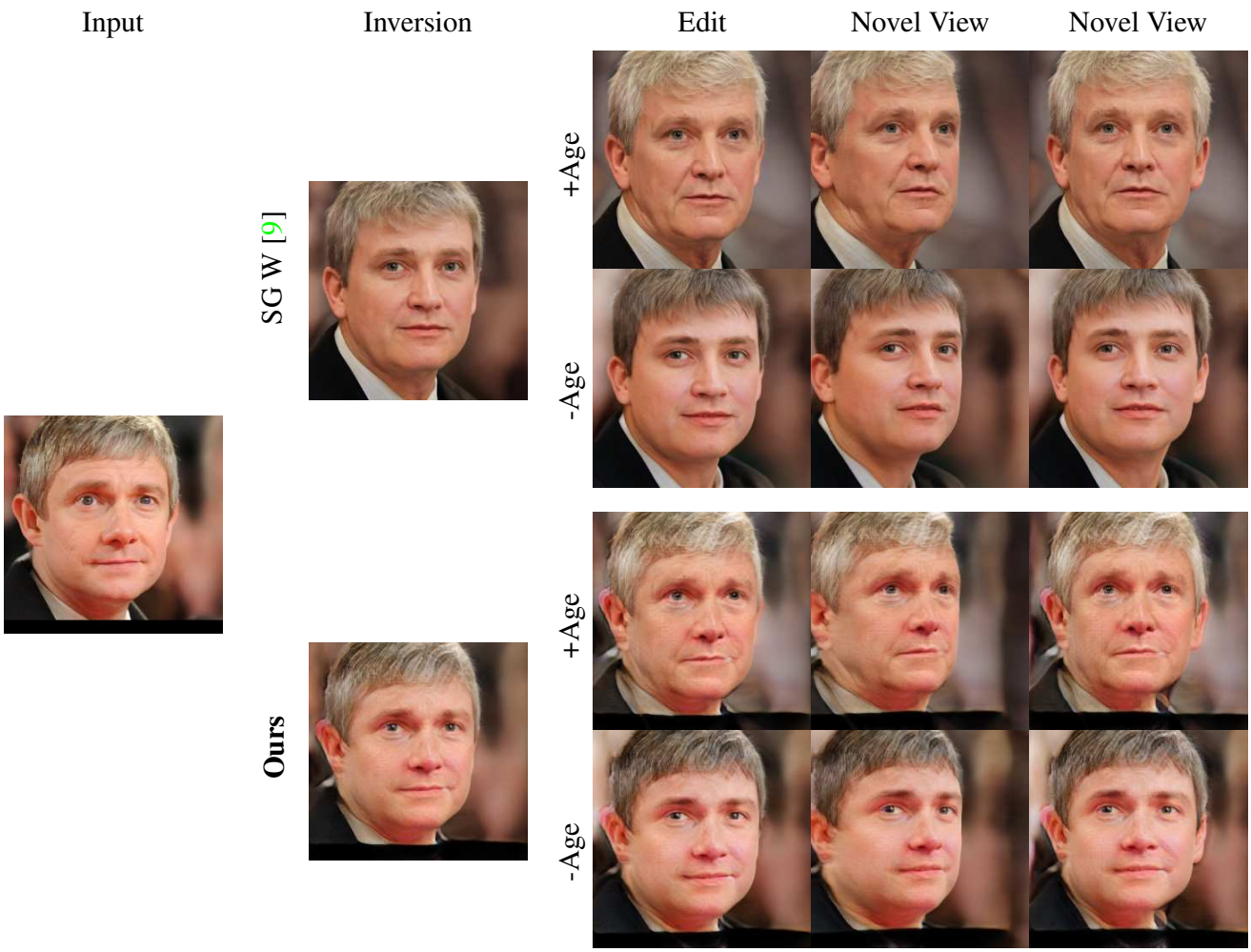


Figure 9. Editing result for age attribute. By utilizing tri-plane features, our method preserves identity more and also generates both realistic and view-consistent editing.

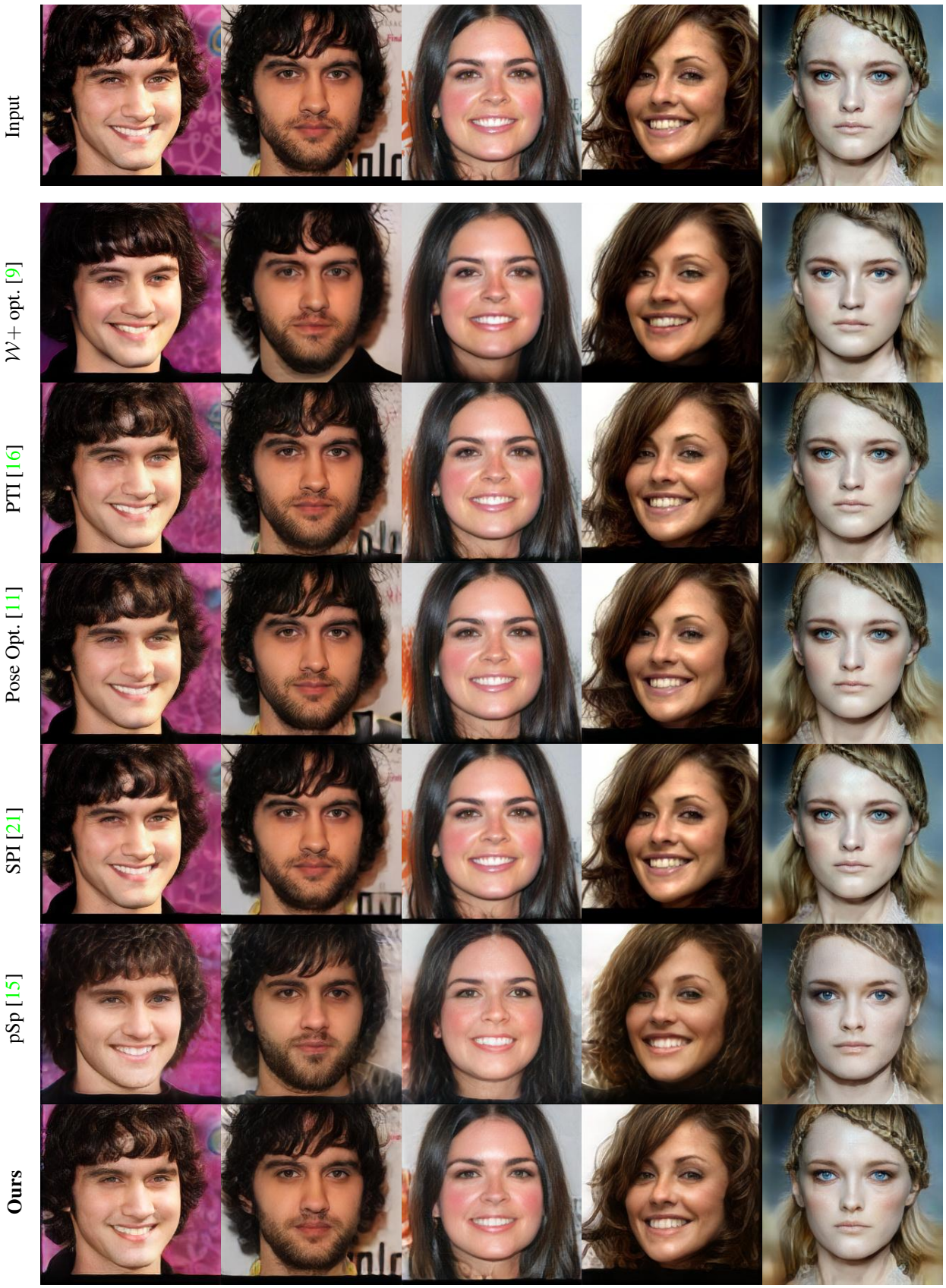


Figure 10. Additional qualitative comparison on image reconstruction.

Figure 11. Additional qualitative comparison on image reconstruction.

Figure 12. Additional qualitative comparison on image reconstruction.

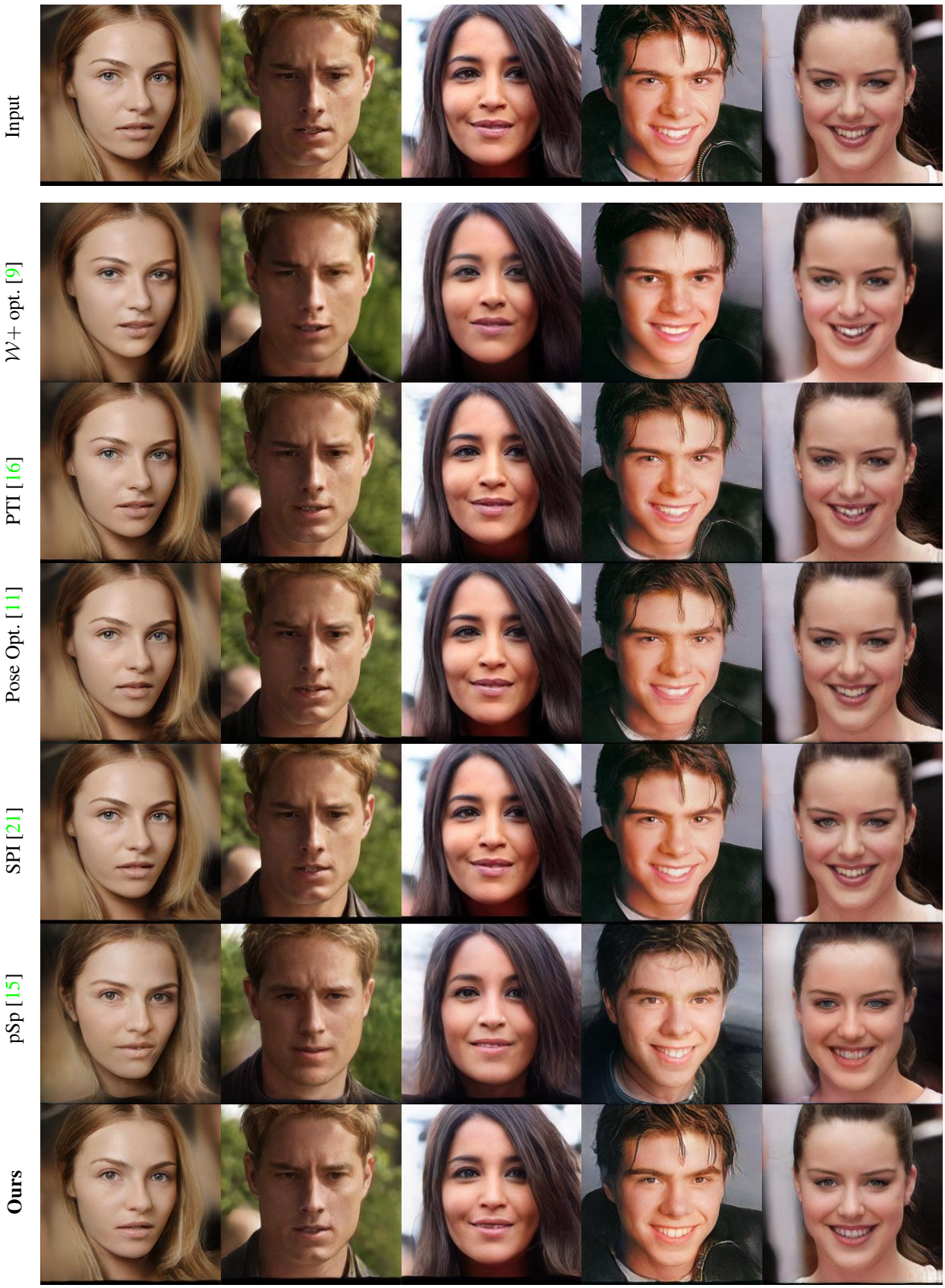


Figure 13. Additional qualitative comparison on image reconstruction.

Figure 14. Additional qualitative evaluation on novel view rendering of yaw angle -0.6, -0.3, 0.3, and 0.6 radians.

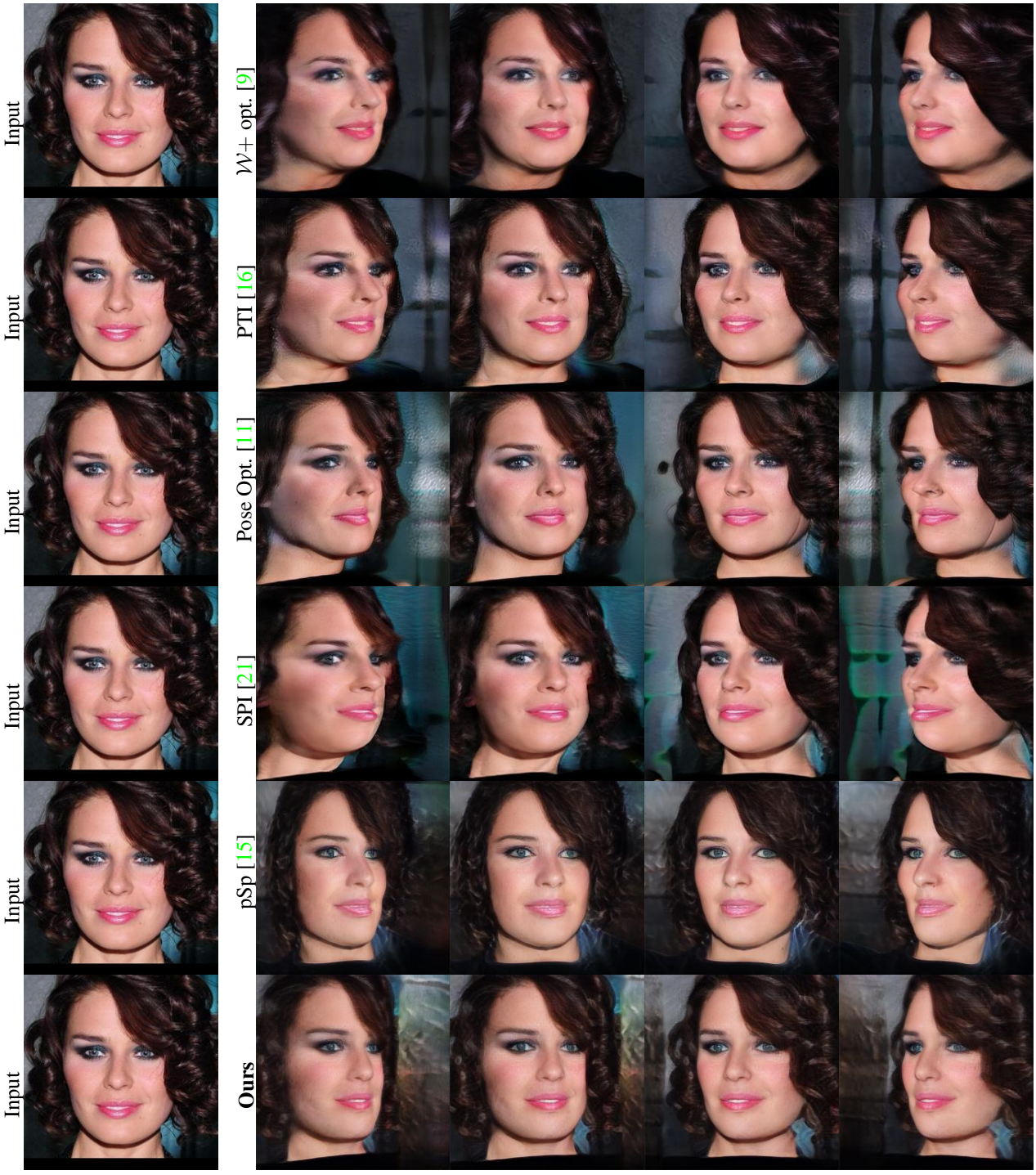


Figure 15. Additional qualitative evaluation on novel view rendering of yaw angle -0.6, -0.3, 0.3, and 0.6 radians.

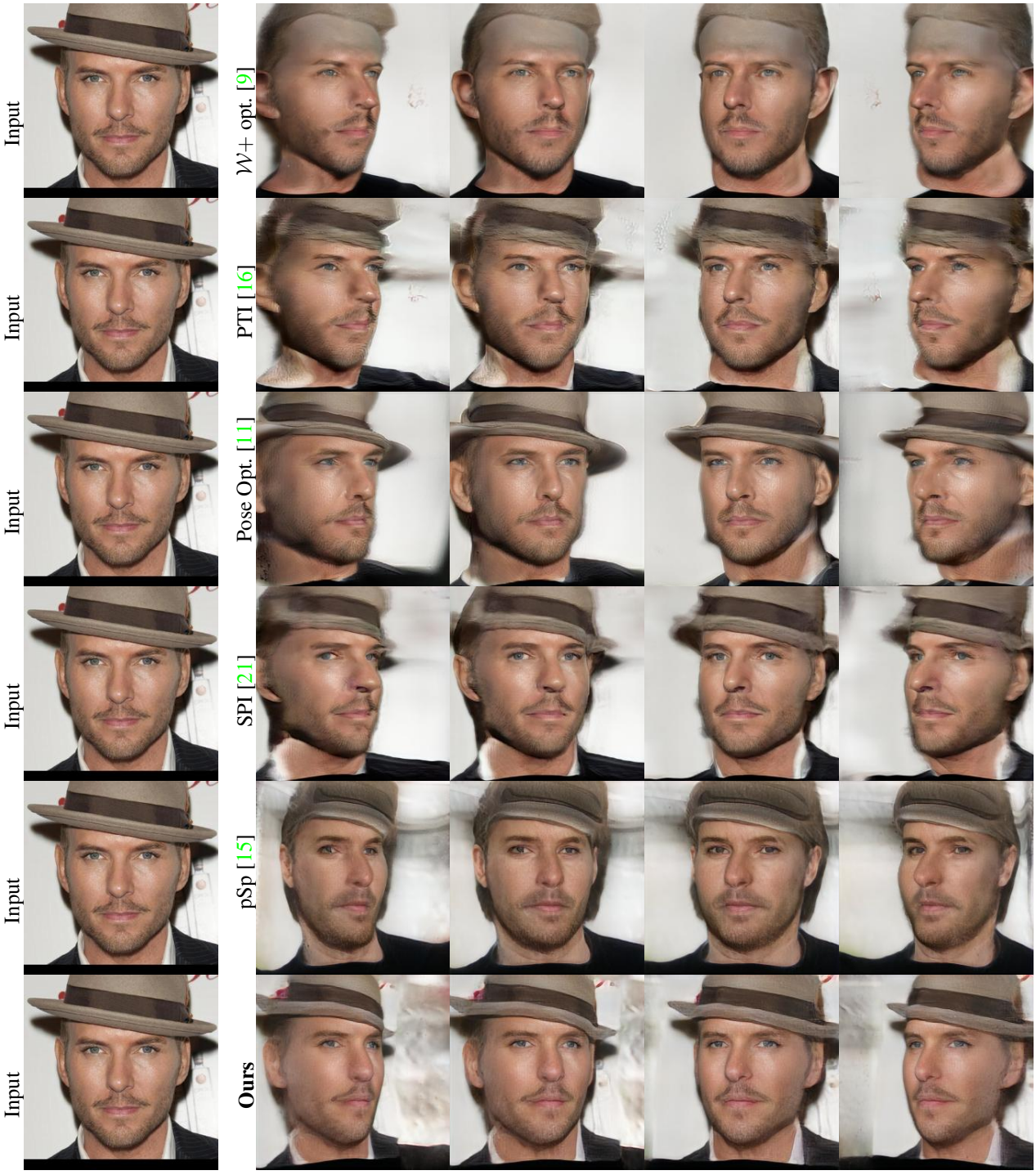


Figure 16. Additional qualitative evaluation on novel view rendering of yaw angle -0.6, -0.3, 0.3, and 0.6 radians.

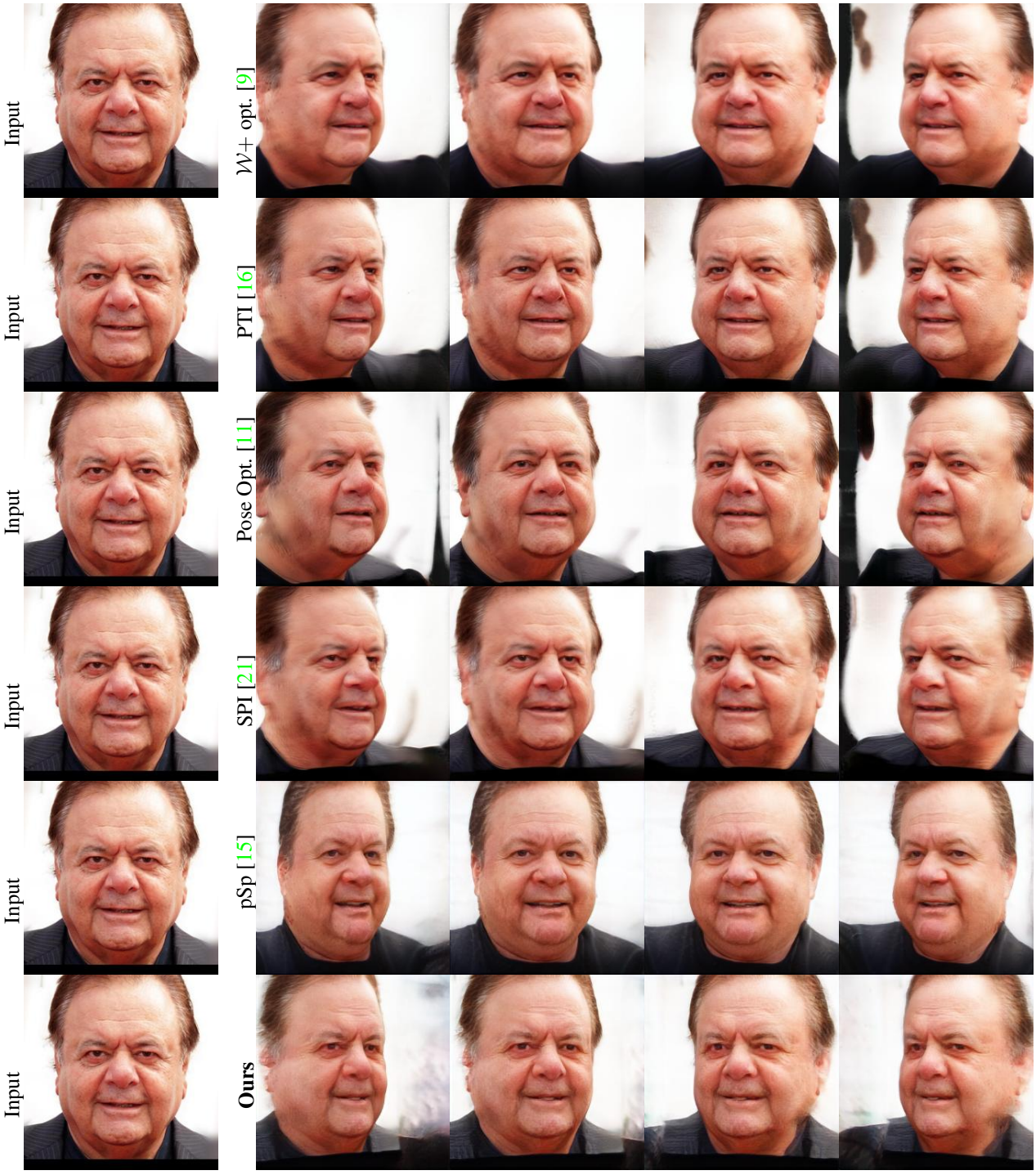


Figure 17. Additional qualitative evaluation on novel view rendering of yaw angle -0.6, -0.3, 0.3, and 0.6 radians.

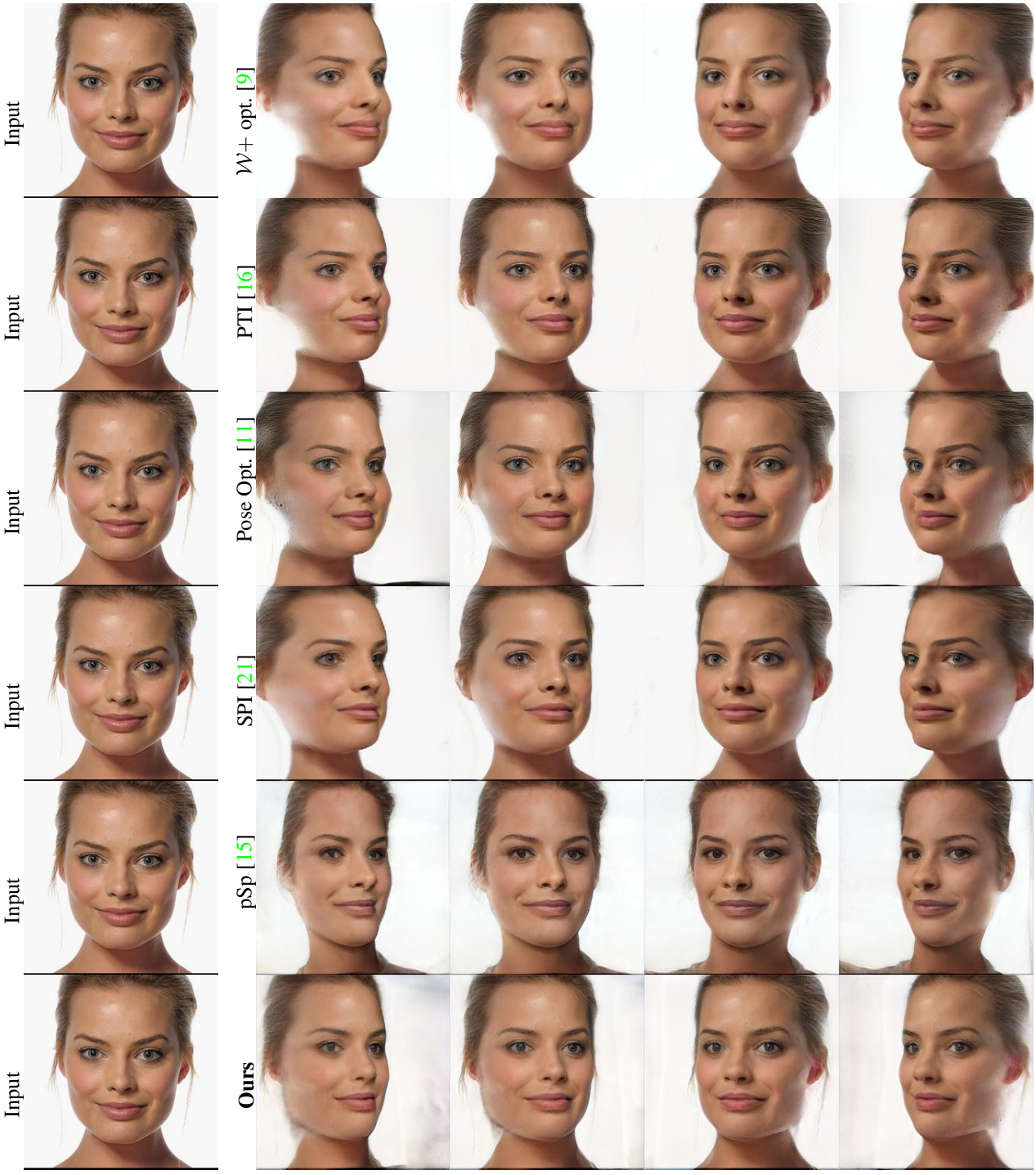


Figure 18. Additional qualitative evaluation on novel view rendering of yaw angle -0.6, -0.3, 0.3, and 0.6 radians.

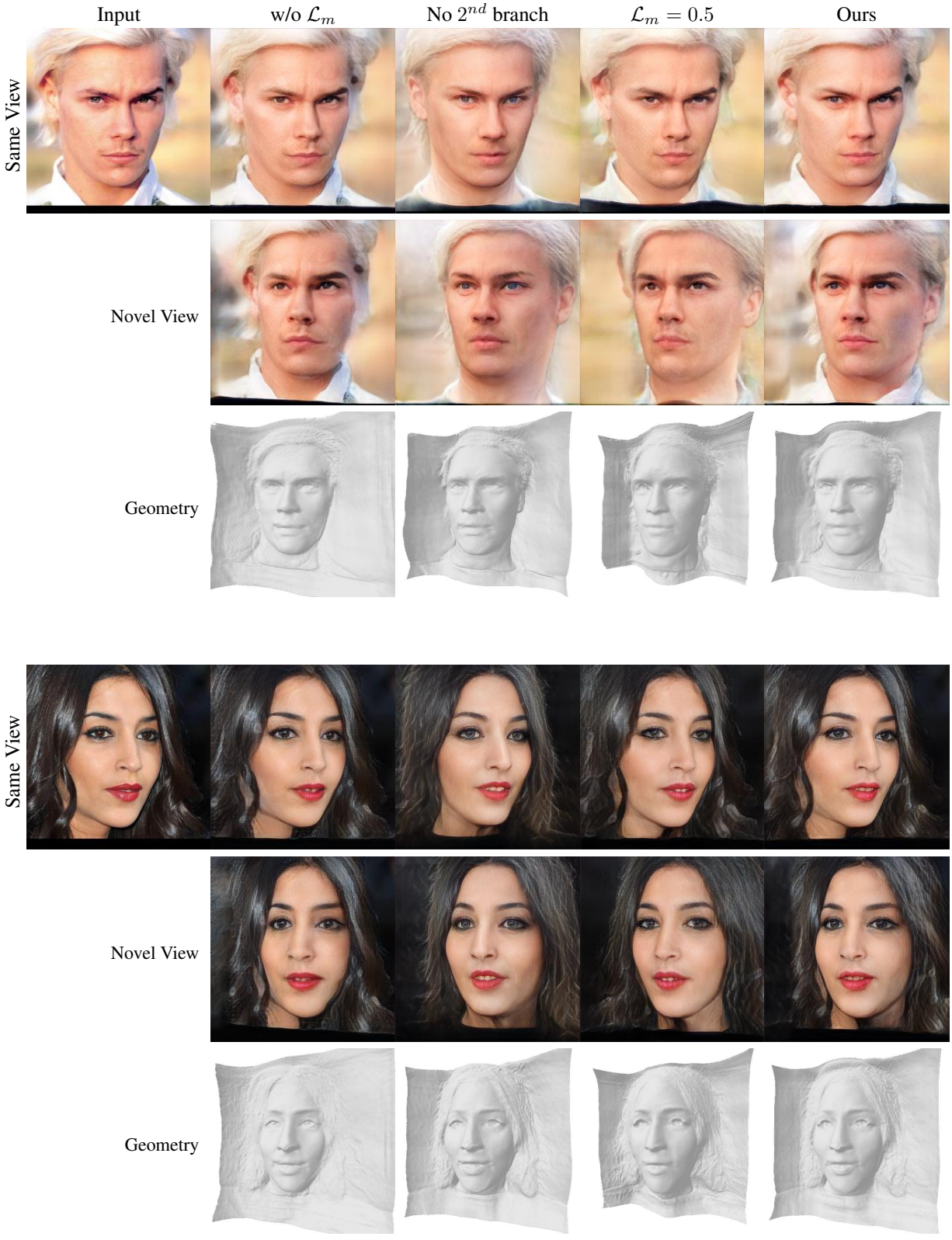


Figure 19. Additional qualitative ablation study for the loss and architecture changes.

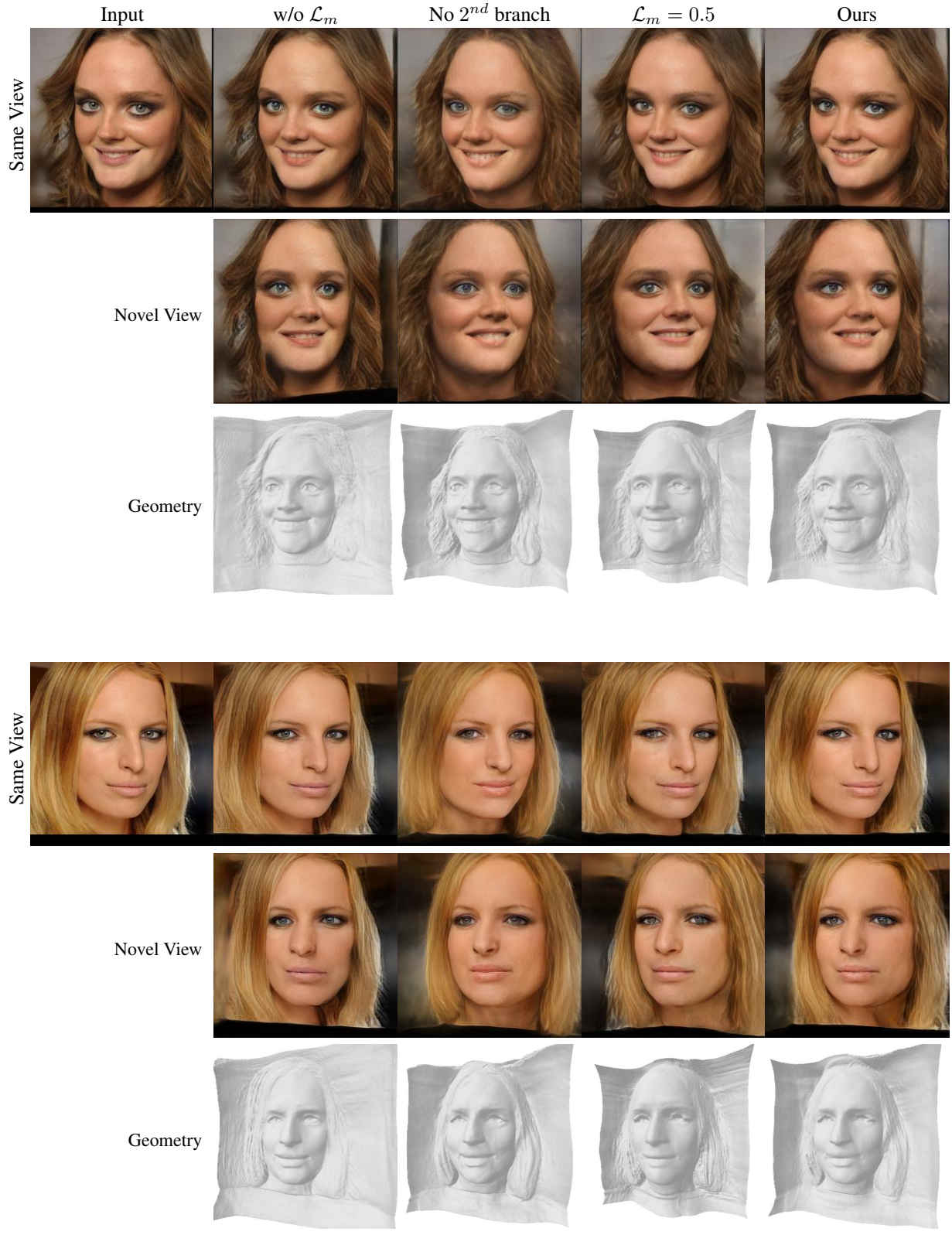


Figure 20. Additional qualitative ablation study for the loss and architecture changes.

References

[1] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *Proceedings of the IEEE/CVF conference on computer Vision and pattern recognition*, pages 18511–18521, 2022.

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022.

[4] Jiankang Deng, J. Guo, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4685–4694, 2018.

[5] Tan M Dinh, Anh Tuan Tran, Rang Nguyen, and Binh-Son Hua. Hyperinverter: Improving stylegan inversion via hypernetwork. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11389–11398, 2022.

[6] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *ArXiv*, abs/2004.02546, 2020.

[7] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[8] Xiaodan Hu, Mohamed A. Naiel, Alexander Wong, Mark Lamm, and Paul Fieguth. Runet: A robust unet architecture for image super-resolution. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 505–507, 2019.

[9] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[10] Zhanghan Ke, Jiayu Sun, Kaican Li, Qiong Yan, and Rynson WH Lau. Modnet: Real-time trimap-free portrait matting via objective decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1140–1147, 2022.

[11] Jaehoon Ko, Kyusun Cho, Daewon Choi, Kwangrok Ryoo, and Seungryong Kim. 3d gan inversion with pose optimization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2967–2976, 2023.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[13] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[14] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *ArXiv*, abs/1908.03265, 2019.

[15] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021.

[16] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Transactions on Graphics (TOG)*, 42(1):1–13, 2022.

[17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *ArXiv*, abs/1505.04597, 2015.

[18] 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.

[19] Shangzhe Wu, C. Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–10, 2019.

[20] Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hellier. A style-based gan encoder for high fidelity reconstruction of images and videos. *European conference on computer vision*, 2022.

[21] Fei Yin, Yong Zhang, Xuan Wang, Tengfei Wang, Xiaoyu Li, Yuan Gong, Yanbo Fan, Xiaodong Cun, Ying Shan, Cengiz Oztireli, et al. 3d gan inversion with facial symmetry prior. *arXiv preprint arXiv:2211.16927*, 2022.

[22] Michael Ruogu Zhang, James Lucas, Geoffrey E. Hinton, and Jimmy Ba. Lookahead optimizer: k steps forward, 1 step back. *ArXiv*, abs/1907.08610, 2019.

[23] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.