Appendix

A. Implementation Details

Dataset Details. We conduct our experiments on two categories of data: monocular images of human portraits and cat faces. We follow the method in EG3D [2] to extract the camera parameters of these images with off-the-shelf pose detectors [4, 5]. For human portraits, we use FFHQ [1] which contains about 70,000 images to train our model. To evaluate our model's performance of input-view reconstruction, we randomly sample 1,500 images from CelebA-HQ [6] test for quantitative evaluation. Additionally, we use a multi-view dataset MEAD [14] to evaluate our model's performance across novel views. Specifically, we use five views (left60°, left30°, front, right30°, right60°) images of 43 persons per frame. We randomly sample 5 frames for each person. For cat faces, we use AFHQv2 Cats [3] following EG3D. We split about 5,000 images into train, evaluation, and test sets by 8:1:1 ratio.

Architecture of Geometry-aware Encoder. Our encoder uses Swin-transformer as the backbone, and we further design attention modules at different scale feature layers for different level latent codes. The encoder architecture is shown in Fig. 1. We split the intermediate output of the Swin-transformer into four levels "query, coarse, mid, fine" similar to the pyramid architecture of CNN models. We use "query" to get w_0 and query $Q_{coarse}, Q_{mid}, Q_{fine}$, and leverage "coarse", "mid", "fine" to obtain keys and values $(K,V)_{coarse}, (K,V)_{mid}, (K,V)_{fine}$. Then the different level queries with corresponding keys and values are sent into cross-attention modules to yield different w_i. Finally, the final latent code w^+ is obtained by:

$$w^+ = w_{avg} + w_0 + (0, w_{1\sim3}, w_{3\sim6}, w_{7\sim13}). \quad (1)$$

Occlusion-aware Mix Tri-plane. As mentioned in Sec.4.3, we can get the visible points set $\mathcal{V}_{(x,y,z)}$. Then we perform orthogonal projection that projects these points to the three axis-aligned feature planes (F_{xy}, F_{xz}, F_{yz}) of tri-plane to get three masks separately, denotes as $tri\text{-}Mask$. The grid point in $tri\text{-}Mask$ is equal to 1 if the corresponding 3D point is in $\mathcal{V}_{(x,y,z)}$, otherwise it is equal to 0. Finally, the mix tri-plane can be obtained by:

$$\begin{aligned} tri\text{-}plane_{mix} = {} & tri\text{-}plane_{F^*} \odot tri\text{-}Mask \\ & + tri\text{-}plane_{w^+} \odot (tri\text{-}I - tri\text{-}Mask), \end{aligned} \quad (2)$$

where $tri\text{-}I$ is the concatenated result of three all-one matrices, which has the same dimension as $tri\text{-}Mask$.

Training Strategy We use a two-stage training strategy. In the first stage, we only train our encoder model based on the loss in Sec. 4.1. After the loss is converged, we freeze the encoder parameters, and train the Adaptive Feature Alignment (AFA) module with occlusion-aware mix tri-plane. When training the geometry-aware encoder, we follow e4e [13] to train different level modules and output corresponding w_i progressively. Only the latent code before and at the current stage will be added to w^+. Different from e4e, we only use 3 progressives stage (*i.e.* coarse, mid, fine), instead of 14 w_i stages (*i.e.* $0, 1, 2, ..., 13$) in e4e. We train the canonical latent discriminator $\mathcal{D}_{\mathcal{W}_c}$ at the beginning of each encoder training iteration, and fix its parameter when training the encoder.

Experiment settings. For human data, we train the encoder with a batch size of 3 on one 3090 GPU for about 1,000,000 iterations. We use a discriminator learning rate of 0.00002 with Adam optimizer and an encoder learning rate of 0.0001 with a ranger optimizer, which is a combination of Rectified Adam [8] with the Lookahead technique [15]. Then we train our AFA module with a learning rate of 0.000025 with Adam optimizer, which uses 800,000 iterations of batch size 2. We only train the encoder 200,000 iterations and 150,000 iterations for cat data. We test our method and other methods with 3090 GPUs, and test inference time with the same settings. We use images from CelebA-HQ or AFHQv2 Cats to perform inversion.

IDE-3D. IDE-3D [12] uses semantic segmentation to retrain EG3D. It learns an encoder to get a latent code and further optimize the generator parameters for inversion and editing images. We compare our encoder with its encoder both qualitatively and quantitatively.

PTI. PTI [10] is the most used optimization method in 2D GAN inversion. We follow its official settings, and only replace the 2D GAN by the EG3D generator. It learns the pivot latent code for 450 iterations, and finetunes generator parameters for 350 iterations.

Pose Opt. Pose Opt. [7] jointly optimizes camera pose, latent codes, and generator parameters for 3D GAN inversion. We follow its official settings and learn the pivot latent code and camera pose for 400 iterations at the first stage, and finetune generator parameters for 400 iterations at the second stage. We also use its pre-trained encoder for pivot latent code and camera pose initialization.

Editing. We perform InterfaceGAN [11] to get a semantic latent direction for editing. First, we sample 500000 front faces whose corresponding latent codes are conditioned by canonical poses, and sort them by attribution classifiers. We choose the top 10000 and bottom 10000 samples according to their score of classification, then we use SVM to get the direction. Finally, We use the method in Sec. 4.4 to get 3D-consistent editing results.

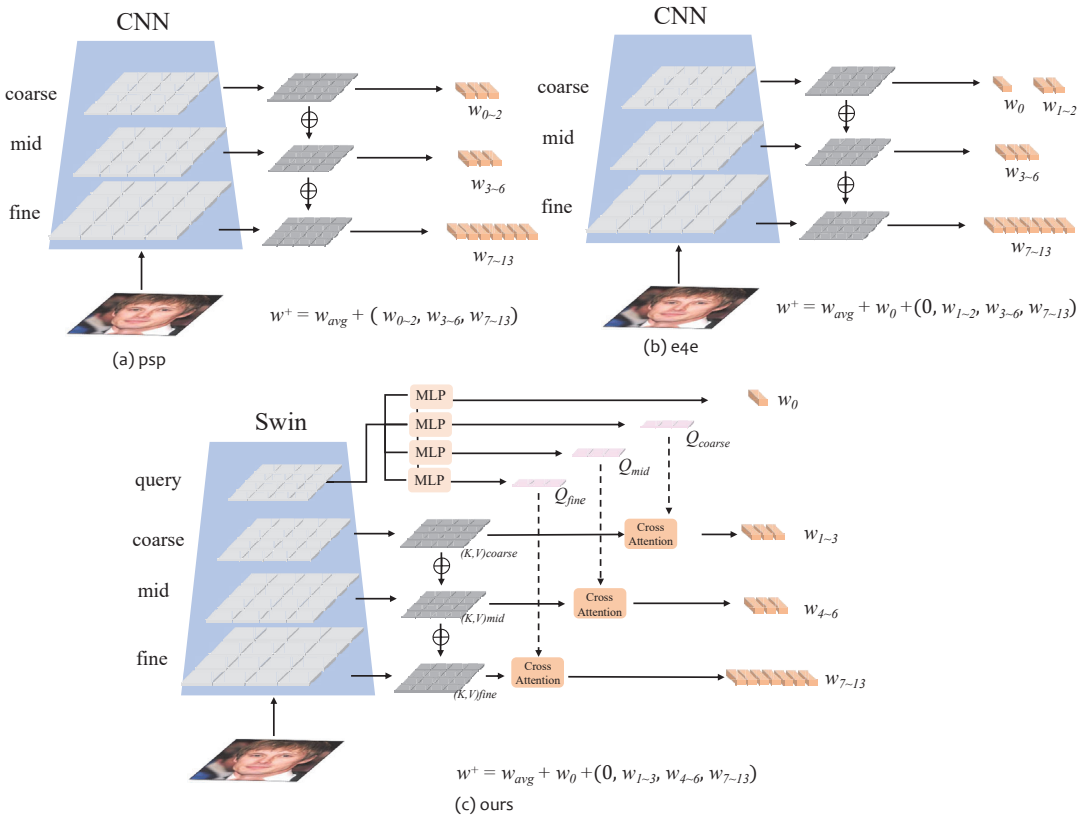

Figure 1: **Encoder architecture comparison.** We refer to pSp [9], e4e [13]'s model structure, refine and build up EG3D inverters respectively (see in (a),(b)). Our designed Swin-transformer based encoder is shown in (c).

Method	Novel view quality												
	yaw= ±60°							yaw = 0°					
	PSNR ↑	SSIM ↑	MSE ↓	LPIPS ↓	FID ↓	ID ↑	Geo. Err. ↓	PSNR ↑	SSIM ↑	MSE ↓	LPIPS ↓	FID ↓	ID ↑	Geo. Err. ↓
w/o $\mathcal{D}_{\mathcal{W}_c}$, w/o $\mathcal{L}_{BG}$	19.82	0.5730	0.0565	0.3069	62.1	0.6206	0.1422	19.83	0.5382	0.0540	0.2966	68.9	0.6842	0.1775
w/o $\mathcal{D}_{\mathcal{W}_c}$, w/ $\mathcal{L}_{BG}$	19.41	0.5576	0.0585	0.2883	59.0	0.6176	0.1016	20.76	0.5829	0.0417	0.2572	60.6	**0.6864**	0.1082
w/ $\mathcal{D}_{\mathcal{W}_c}$, w/o $\mathcal{L}_{BG}$	20.20	0.5999	0.0495	0.2599	55.5	0.6459	0.1414	20.67	0.5757	0.0428	0.2527	55.1	0.6844	0.1491
w/ $\mathcal{D}_{\mathcal{W}_c}$, w/ $\mathcal{L}_{BG}$	**20.66**	**0.6211**	**0.0473**	**0.2203**	**52.2**	**0.6552**	**0.0943**	**21.12**	**0.5944**	**0.0383**	**0.2372**	**54.1**	0.6848	**0.0955**
w/o occlusion-aware	20.69	0.6176	0.0489	0.2327	52.5	0.6832	0.0983	20.26	0.6374	**0.03202**	0.2230	50.3	**0.744**	0.0954
Ours	**20.87**	**0.6299**	**0.0424**	**0.2192**	**50.9**	**0.700**	**0.0950**	**22.13**	**0.6502**	0.0329	**0.2204**	**51.7**	0.743	**0.0944**

Table 1: **Ablation studies on Geometry-aware Encoder and Occlusion-aware Mix Tri-plane.** We evaluate the proposed canonical space $\mathcal{W}_c$, the background regularization, and the occlusion-aware mix tri-plane on MEAD dataset. The best performance on $\mathcal{W}$ space inversion and $\mathcal{F}$ space inversion are in bold.

Method	PSNR ↑	SSIM ↑	MSE ↓	LPIPS ↓	FID ↓	ID ↑	Geo. Err. ↓
w/ conv	19.67	0.5962	0.0455	0.1766	25.1	0.8033	0.1094
w/ AFA	**21.84**	**0.7079**	**0.301**	**0.1242**	**18.1**	**0.8797**	**0.0984**

Table 2: **Ablation study on AFA module.** We replace the proposed adaptive alignment module in AFA with CNN network and evaluate on CelebA-HQ dataset.

B. More Experimental Results

Ablation of Geometry-aware Encoder Designs. We test different ablation settings of our encoder on MEAD, for novel view geometry and texture evaluation. As shown in Table 1, the design of the canonical discriminator ($\mathcal{D}_{\mathcal{W}_c}$) and background depth regularization ($\mathcal{L}_{BG}$) is necessary for a good geometry inversion.

Ablation of Adaptive Feature Alignment. We evaluate Adaptive Feature Alignment (AFA) module ablation on source view reconstruction performance on CelebA-HQ, as shown in Table 2. Modified feature maps generated by only convolution modules are hard to align to the facial region in canonical feature space $\mathcal{F}_c$, whose reconstruction quality is inferior to our method.

Ablation of Occlusion-aware Mix Tri-plane. We evaluate occlusion-aware mix tri-plane design on MEAD for novel view evaluation. As the distortion exists in the occlusion part, our full models will perform better on novel view im-

age synthesis, as shown in Table 1.

Comparison with Encoder-based Methods. We show more qualitative comparisons with encoder-based methods in Fig. 3. Our method significantly surpasses others.

Comparison with Optimization-based Methods. We present more qualitative results of comparison with optimization-based methods in Fig. 4. It is worth noting that when the input image is a side face, optimization-based methods tend to overfit the input image and are hard to synthesize novel-view images. In some cases, Pose Opt. fails to converge to an accurate pose using optimization and generates degradation results.

More Human Faces Results in extreme condition. As shown in Fig 2, we present our method compared with PTI [10] on different extreme conditions: pose, appearance (heavy make-up), expression, and show the failure cases.

More Novel View Results with Our Method. More novel view results with our method can be found in Fig. 5, 6, 7. Our method can achieve high-quality 3D-consistent multi-view image synthesis.

More Results of Editing with Our Method. We present more 3D-consistent results of human faces and cat faces in Fig. 8, 9, 10. Our method shows powerful editing ability which can be used in real-world applications.

C. More Disscussion

Without off-the-shelf pose estimator. IDE-3D uses the same off-the-shelf pose estimator as ours, and we use the estimated poses when testing IDE-3D. While Pose Opt. uses an encoder to get a pose as an initialization of their optimization process, we can easily equip our approach with camera pose estimation. By employing a simple MLP to map the inverted latent code to the pose, we achieve favorable results. Table 3 lists the performance of our camera pose estimator on human faces and car datasets.

	Human faces			cars		
Pose	C-MSE ↓	MSE ↓	LPIPS ↓	C-MSE ↓	MSE ↓	LPIPS ↓
GT	0.0031	0.0301	0.1257	0.0218	0.0367	0.1745
ours-pred		0.0354	0.1286		0.0452	0.1802

Table 3: **Camera pose estimation results. C-MSE denotes the MSE between our predicted camera pose and ground truth. MSE and LPIPS are calculated between input images and corresponding inversion results.**

Without Transformer-based Encoder. As shown in Table 4, we replace the backbone with the encoder of e4e (e4e+ours). It shows better performance than e4e-3D in Table.1 in the main paper, which indicates the effectiveness of $\mathcal{D}_{\mathcal{W}_c}, L_{BG}$. Meanwhile, the Swin-transformer brings better results by comparing the e4e+ours and Ours-w^+ in the main paper.

	yaw=±60°				yaw=0°			
	LPIPS ↓	FID ↓	ID ↑	Geo. Err ↓	LPIPS ↓	FID ↓	ID ↑	Geo. Err ↓
e4e+ours	0.2453	53.6	0.629	0.1178	0.2416	52.2	0.671	0.1211

Table 4: **Results of e4e backbone with our $\mathcal{D}_{\mathcal{W}_c}, L_{BG}$.**

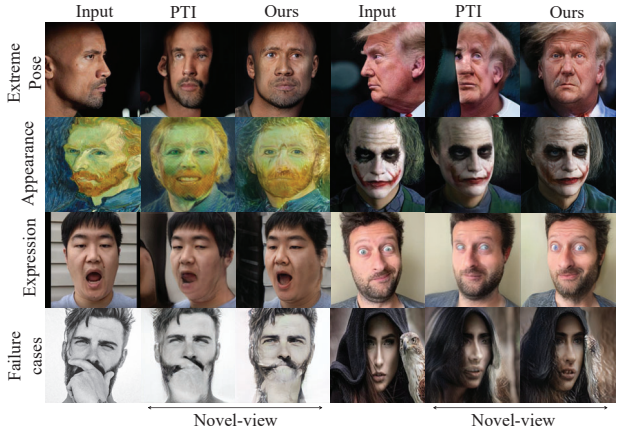

Figure 2: **More extreme results (Row. 1-3) and failure cases (Row. 4).**

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019. 1

[2] 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. 1

[3] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8188–8197, 2020. 1

[4] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1

[5] kairess. Cat hipsterizer. https://github.com/kairess/cat_hipsterizer/, 2018. 1

[6] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference On Learning Representations*, 2018. 1

[7] Jaehoon Ko, Kyusun Cho, Daewon Choi, Kwangrok Ryoo, and Seungryong Kim. 3d gan inversion with pose optimization. *arXiv preprint arXiv:2210.07301*, 2022. 1

[8] 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 preprint arXiv:1908.03265*, 2019. 1

[9] 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. 2

[10] 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. 1, 3

[11] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9243–9252, 2020. 1

[12] 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. 1

[13] 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. 1, 2

[14] Kaisiyuan Wang, Qianyi Wu, Linsen Song, Zhuoqian Yang, Wayne Wu, Chen Qian, Ran He, Yu Qiao, and Chen Change Loy. Mead: A large-scale audio-visual dataset for emotional talking-face generation. In *European Conference on Computer Vision*, pages 700–717. Springer, 2020. 1

[15] Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. Lookahead optimizer: k steps forward, 1 step back. *Advances in Neural Information Processing Systems*, 32, 2019. 1

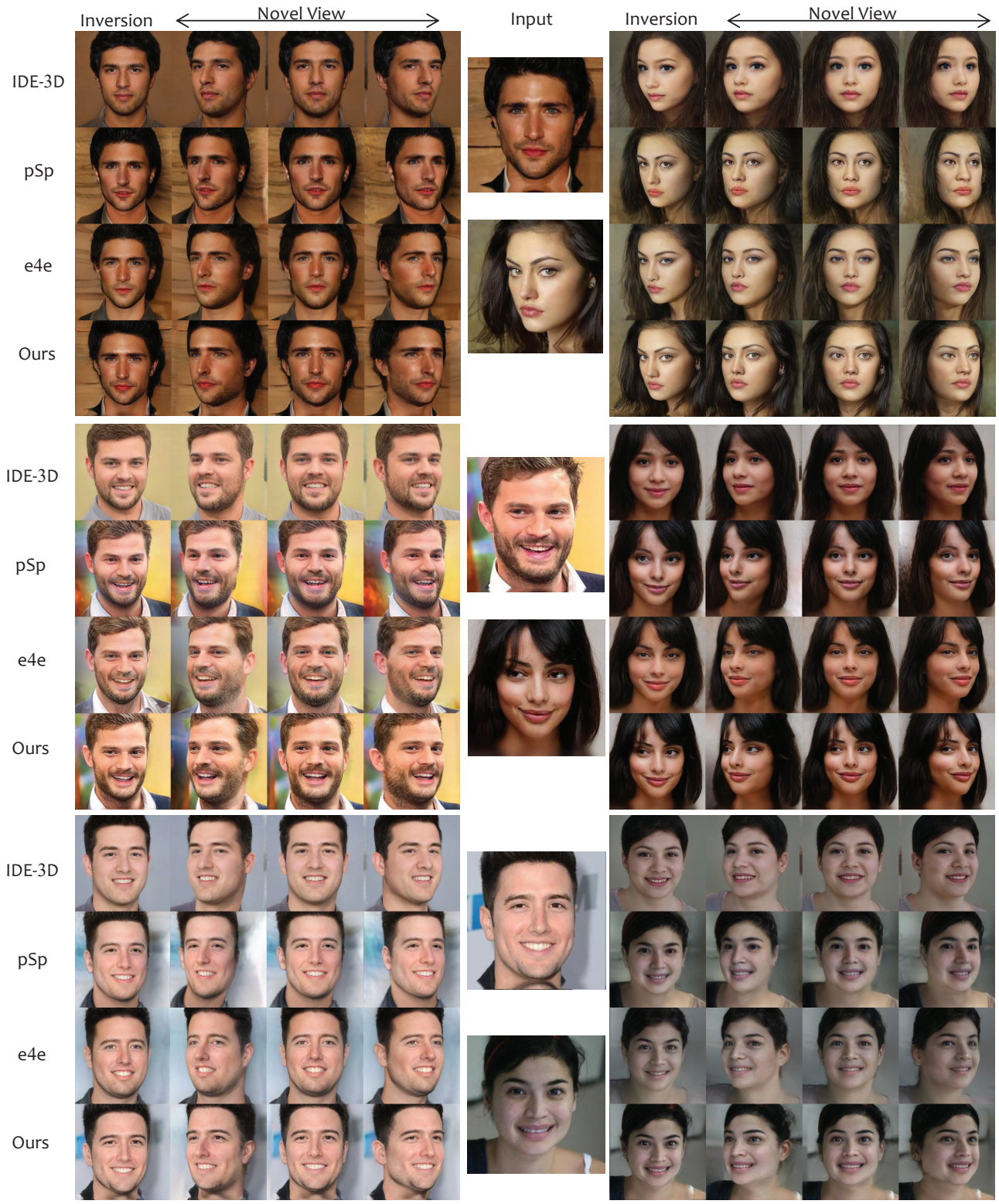

Figure 3: **Comparison with encoder-based methods.**

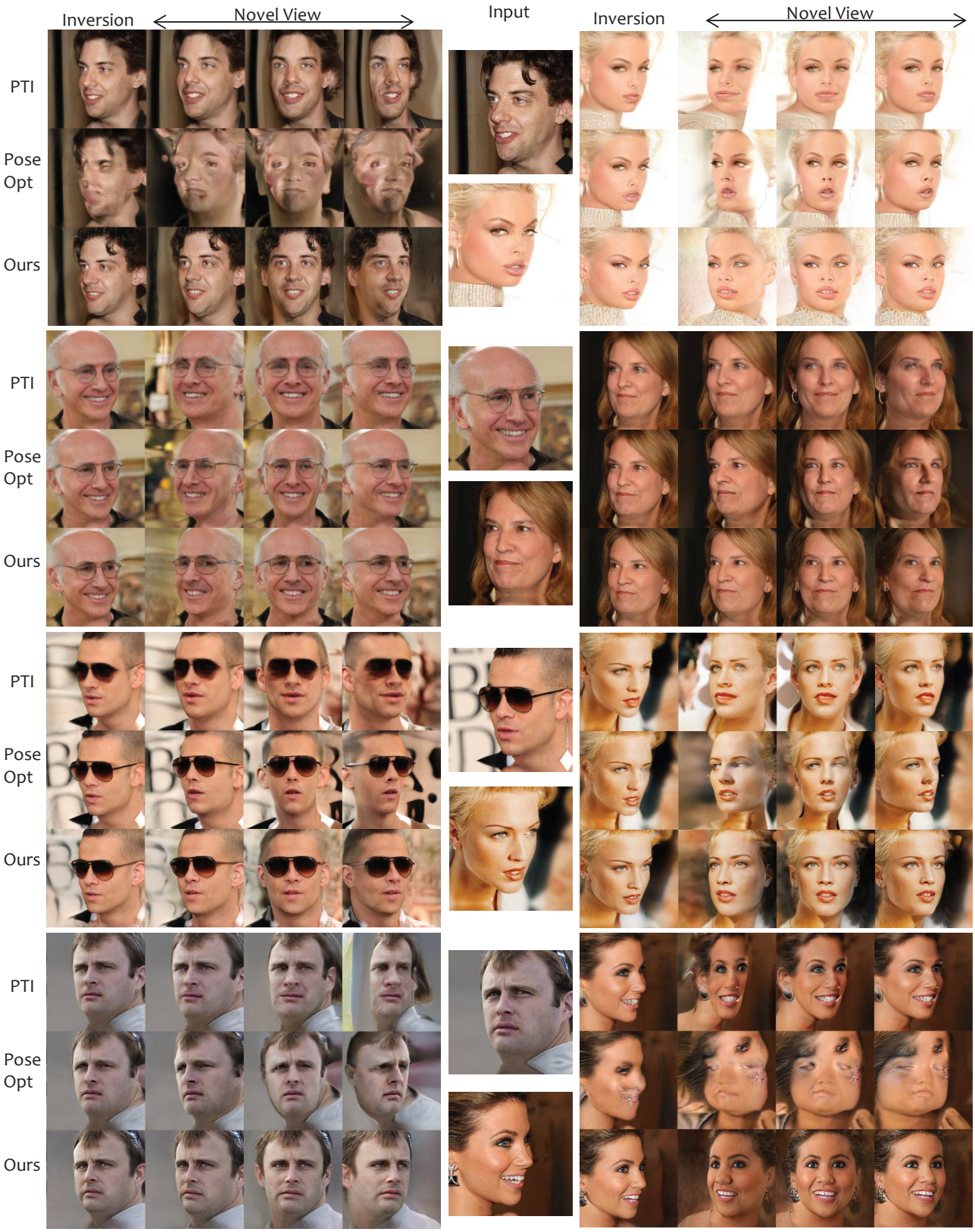

Figure 4: **Comparison with optimization-based methods.**

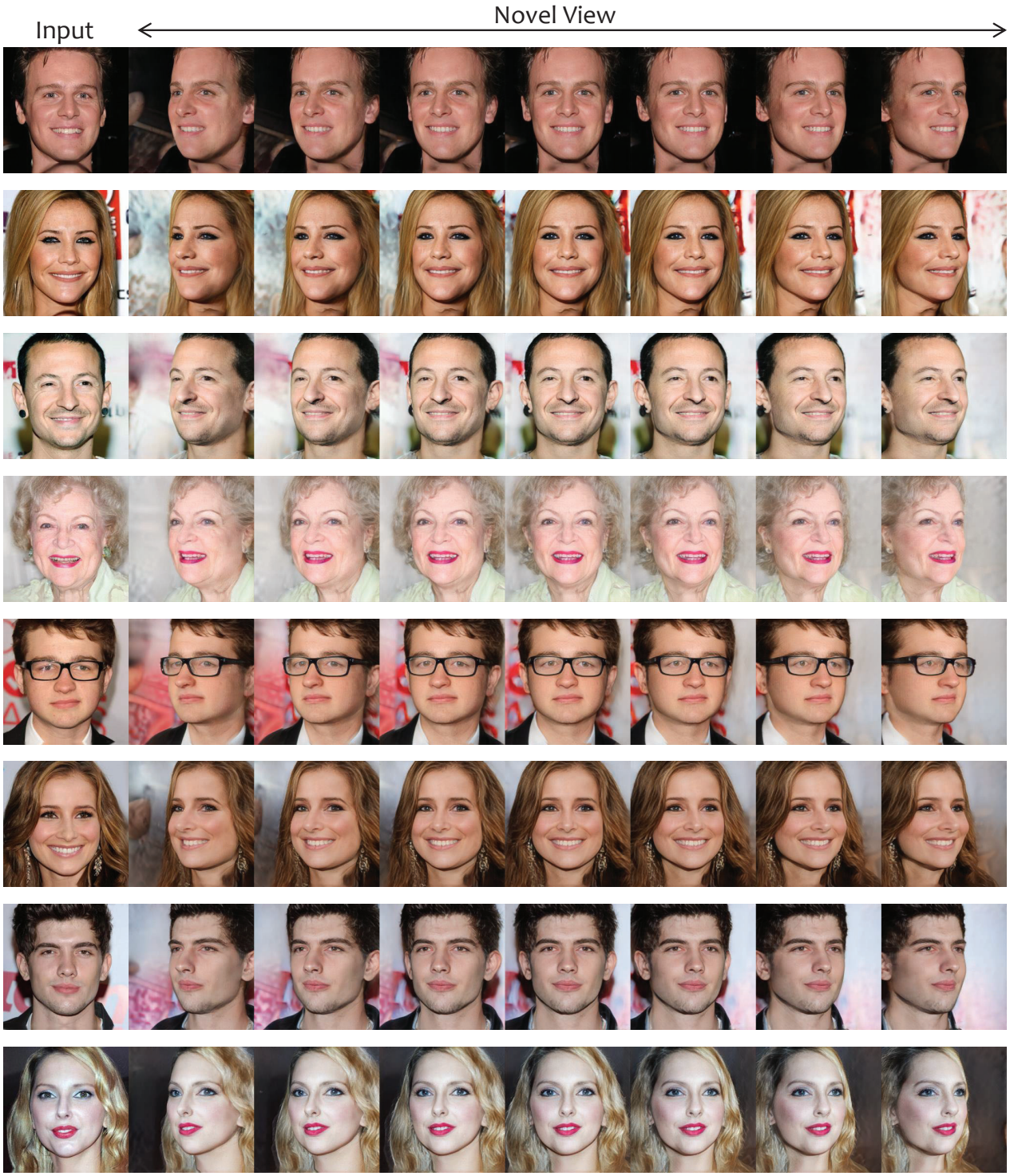

Figure 5: **Multi-view human faces inversion results of our method (part 1/2).**

Figure 6: **Multi-view human faces inversion results of our method (part 2/2).**

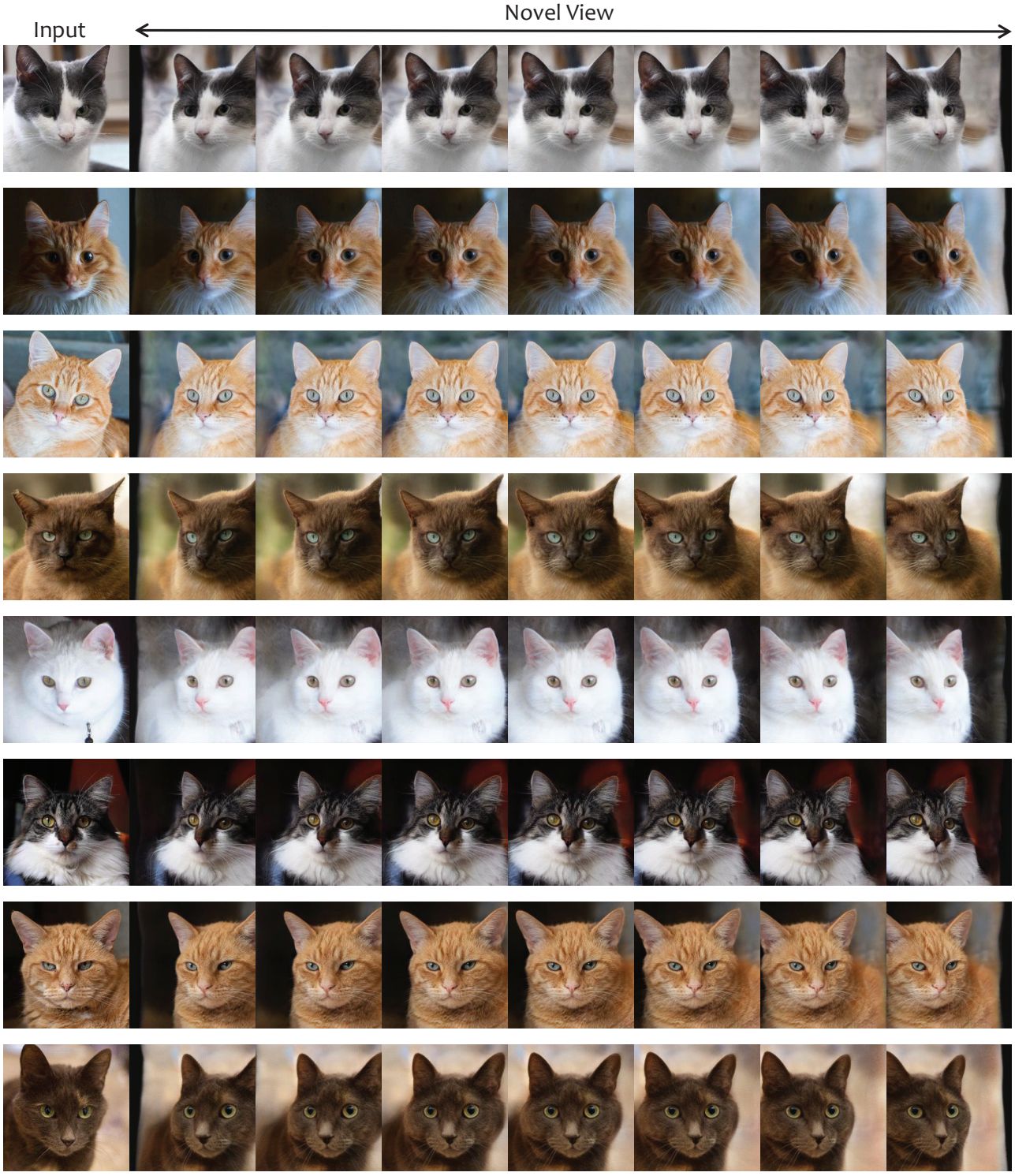

Figure 7: **Multi-view cat faces inversion results of our method.**

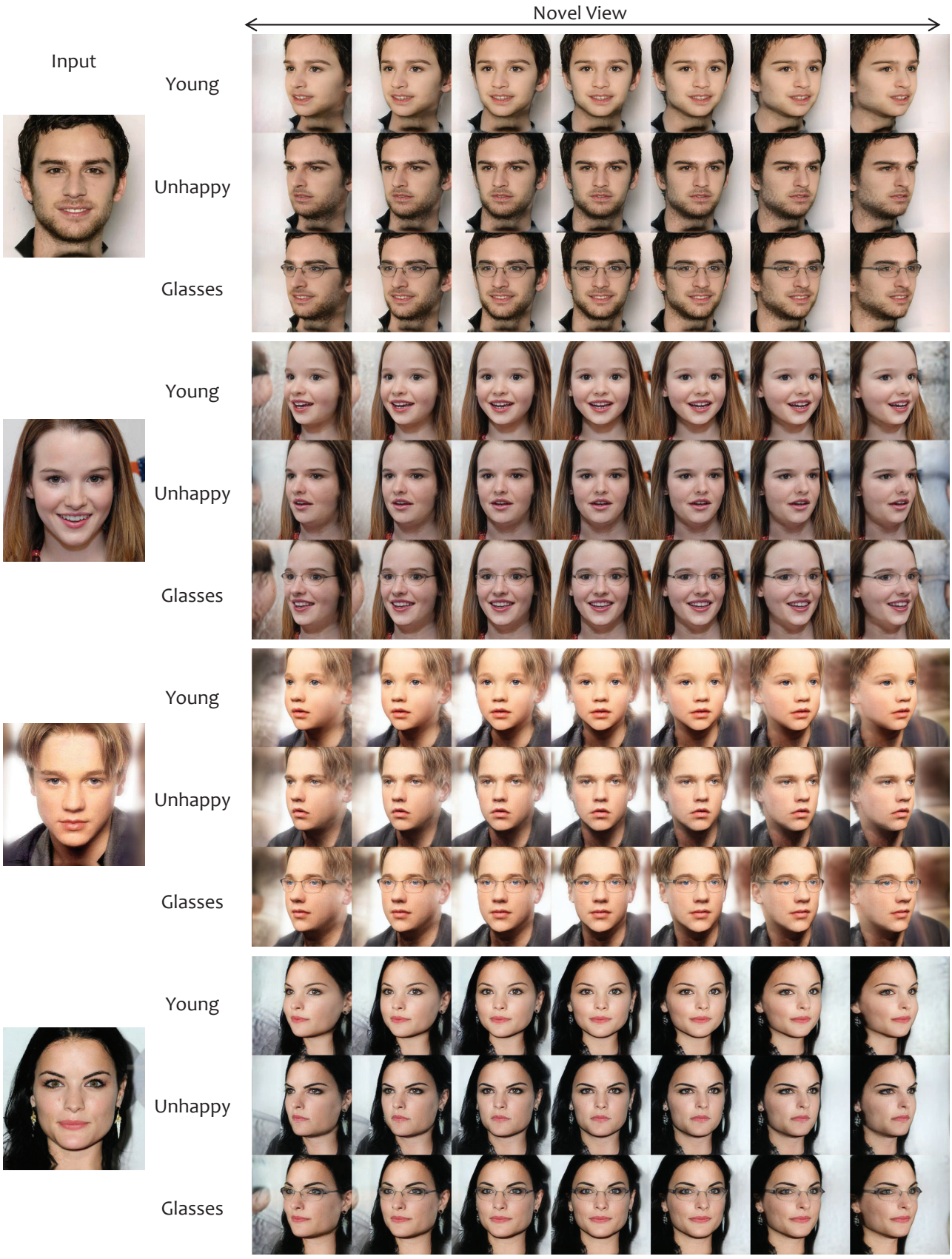

Figure 8: **Editing results on human faces (part 1/2).**

Figure 9: **Editing results on human faces (part 2/2).**

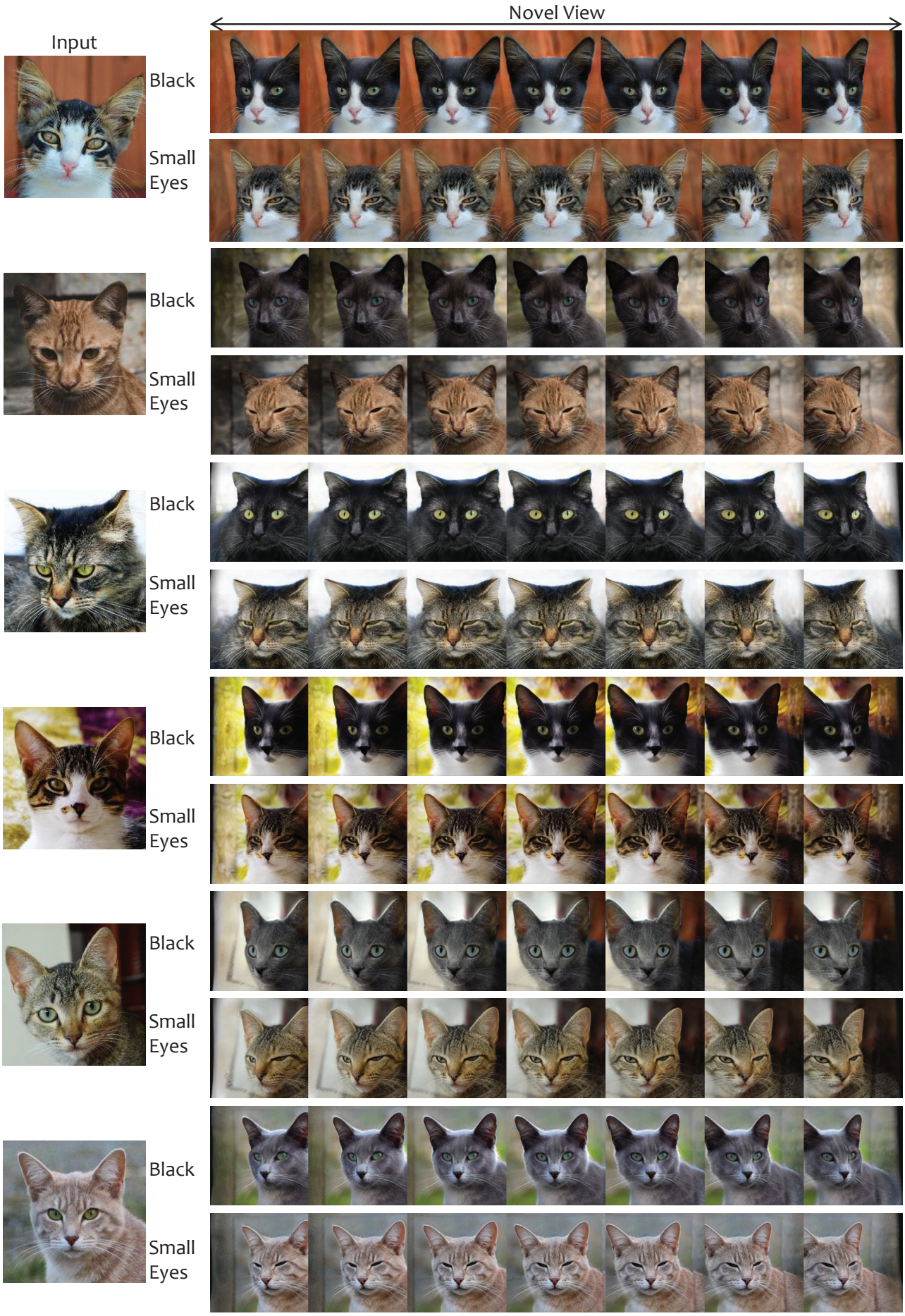

Figure 10: **Editing results on cat faces.**