Supplementary for:
StyleRes: Transforming the Residuals for Real Image Editing with StyleGAN

Hamza Pehlivan Yusuf Dalva Aysegul Dundar
Bilkent University
{hamza.pehlivan,yusuf.dalva}@bilkent.edu.tr
adundar@cs.bilkent.edu.tr

Table 1. Inference time of the competing methods are given.

Method	Runtime (sec)
pSp [8]	0.088
e4e [11]	0.089
ReStyle [1]	0.365
HyperStyle [2]	0.437
HFGI [12]	0.130
StyleTransformer [5]	0.063
FeatureStyle [13]	0.526
PTI [9]	97.94
StyleRes (Ours)	0.125

Table 2. Quantitative results of reconstruction and editing on CelebA-HQ dataset. We compare with PTI. For reconstruction, we report FID, SSIM, and LPIPS scores. For editing, we report FID metrics for smile addition (+) and removal (-).

	Reconstruction			Editing - FIDs	
Method	FID	SSIM	LPIPS	Smile(+)	Smile(-)
PTI [9]	10.64	**0.92**	**0.07**	30.29	30.11
StyleRes (Ours)	**7.04**	0.90	0.09	**23.52**	**21.80**

In Supplementary material, we provide

- Run-time comparisons of different models in Table 1.
- Evaluation results of different models on age and pose edits in Table 3.
- Training, architecture and evaluation details.
- Discussion on limitations of our work.
- Visual comparisons with PTI, FeatureStyle, and StyleTransformer.

1. Training Details

We use e4e [11] as the basic encoder and invert StyleGAN2 [6] generator. The high level features F_0 are the input of the first map2style layer used in e4e. F_0 has the spatial dimension of $128 \times 64 \times 64$. The intermediate StyleGAN features are extracted as $G_W = G_{0 \to 8}(W)$, where the arrow operator indicates the indices of convolution layers used. G_W has the spatial dimension of $512 \times 64 \times 64$.

When we choose the *no editing path*, we set $\lambda_{r1} = 1.0$, $\lambda_{r2} = 0.001$, $\lambda_{r3} = 0.1$ for the face and $\lambda_{r3} = 0.5$ for the car dataset. When we choose the *cycle translation path*, we set $\lambda_{r1} = 0.0$, meaning we do not use cycle consistency at the pixel level, $\lambda_{r2} = 0.0001$, $\lambda_{r3} = 0.01$ for the face and $\lambda_{r3} = 0.05$ for the car dataset. At both paths, we set $\lambda_a = 0.1$. The regularizer coefficient is set to $\lambda_f = 5.0$ for the face dataset and $\lambda_f = 3.0$ for the car dataset. The network is trained with Adam optimizer, with a learning rate equal to 0.0001. We halved the learning rate at iterations 5000, 10000 and 15000.

Table 3. Age and pose editing FIDs of competing methods are given.

Method	Pose(+)	Pose(-)	Age(+)	Age(-)
pSp	22.95	22.13	35.89	28.52
e4e	26.92	26.83	42.21	40.36
ReStyle	21.65	22.43	25.82	26.62
HyperStyle	15.59	15.83	16.19	22.81
HFGI	15.37	15.97	19.28	18.19
StyleTransformer	20.26	21.06	42.22	35.21
FeatureStyle	30.00	24.24	14.80	23.59
StyleRes (Ours)	**11.31**	**10.73**	**12.94**	**14.91**

2. Model Architecture

Our model consists of residual layers, which are visualized in Fig. 2. Encoder E_1 takes the concatenated features F_0 and G_w, which has a spatial dimension of 640x64x64. It then forwards the concatenated features to E_2 using an encoder-decoder architecture. The first convolution layer sets the channel size as 512, and it is not changed in the rest of the layers. *DownResBlk* reduces the resolution to half, and *Interpolate* layer doubles the resolution, by using linear

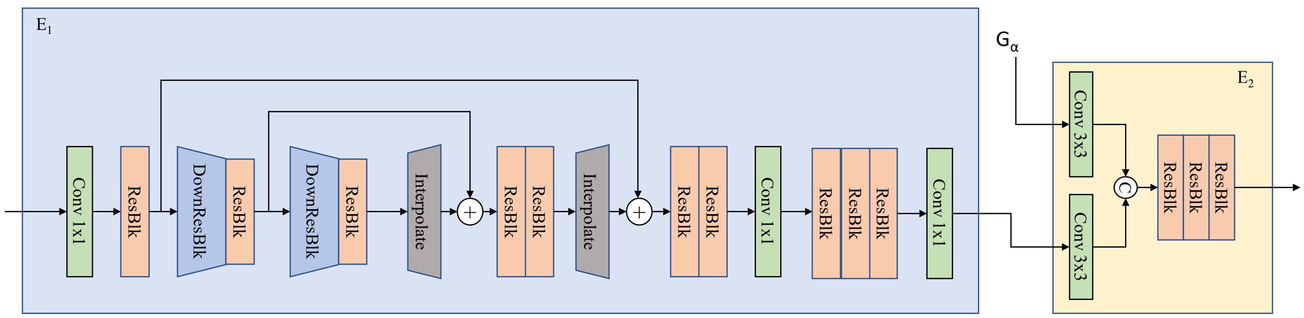

Figure 1. Detailed architecture of our encoders. More information regarding dimensions are given in the corresponding text.

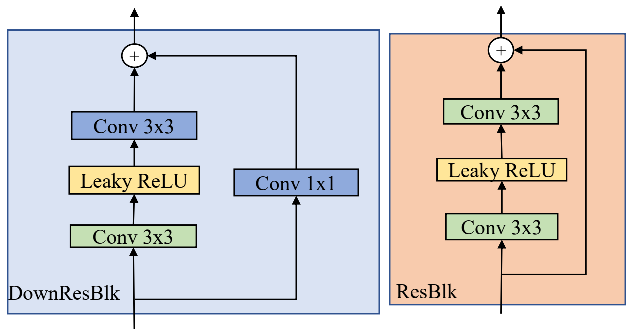

Figure 2. Building blocks of our encoders. On the left, we demonstrate the downsampling layers used in E_1. On the right, we show standard residual layers used in both E_1 and E_2.

interpolation.

E_2 takes both F_a and G_α as inputs, and learns how to adapt encoded features to the generator features. Before concatenating the input features, their channel sizes are reduced to 256 using 3x3 convolution layers. After the concatenation, the channel size becomes 512, and it does not change throughout the E_2. The detailed architecture of our encoders is given in Fig. 1.

3. Evaluation Details

For the face images, the reconstruction evaluations are performed on the last 2000 images of the CelebA-HQ dataset, same as the most of the inversion methods. For smile addition, we first find the ground truth smiling and non-smiling images in the CelebA test set. We add a smile to non-smiling ground truth images, and obtain fake smiling images. Finally, we compute the FID between the ground-truth smiling and the fake smiling images. Similar setup is used for smile removal as well. To add or remove a smile, we use the boundary obtained by InterfaceGAN, with a factor of 3. For age and pose edits, we obtain FID score between the original and edited images, which is calculated using the entire evaluation dataset. Pose and age edits are obtained by InterfaceGAN, with a factor of 2.

For the car images, the reconstruction and editing evaluations are performed on the first 2000 images of the Stanford Cars test set. Because we do not have ground-truth labels, we directly calculate FID between the original and edited images. The edited images are obtained using GanSpace directions.

For HyperStyle and Restyle, we use 5 iterative iterations in the reconstruction and editing, which is consistent with their training scheme. For PTI, we deploy locality regularization, as introduced in their work, for both reconstruction and editing.

We provide visuals obtained with various editing methods, namely InterfaceGan [10], GanSpace [4], StyleClip [7] and GradCtrl [3]. Because our network is trained with editing in mind, we can directly apply boundaries found by different editing methods.

4. Limitations

A common problem with high-rate inversion models is they may not be able to adapt to viewpoint or structure changes. Although our model is trained to adapt better to edits, it can still produce unpleasurable results for edits with large misalignments like large pose changes. Examples of such cases are given in Fig. 3. We believe that a better way of training would also consider geometric consistency, which is not explicitly modeled with StyleRes when applying the edits. We leave this as future work.

5. Additional Results

In Table 2, we provide comparisons with PTI. PTI runs optimization for each image and achieves better reconstruction scores in terms of SSIM and LPIPS. However, our method achieves significantly better edit FIDs and reconstruction FIDs. Furthermore, running evaluation with PTI method takes more than 2 days on a single GPU, whereas our method finishes within 5 minutes. The running times (Table 1) are obtained by averaging the reconstruction times

Figure 3. High rate inversion models struggle with pose edits. Our model is also affected with this limitation.

of 2000 samples with batch size equal to 1 on a single NVIDIA GeForce GTX 1080 Ti GPU.

We provide visual comparisons on:

1. smile editing with StyleTransformer, PTI, and FeatureStyle in Figs. 4 and 5, by using InterfaceGAN.
2. smile editing with e4e, HFGI and HyperStyle in Figs. 6 and 7, by using InterfaceGAN.
3. age editing with e4e, HFGI and Hyperstyle in Figs. 8 and 9, by using InterfaceGAN.
4. pose change with e4e, HFGI and HyperStyle in Figs. 10 and 11, by using InterfaceGAN.
5. beard addition with e4e, HFGI and HyperStyle in Fig. 12, by using GanSpace.
6. eye openness with e4e, HFGI and HyperStyle in Fig. 13, by using GanSpace.
7. lipstick addition with e4e, HFGI and HyperStyle in Fig. 14, by using GanSpace.
8. eyeglasses addition with e4e, HFGI and HyperStyle in Fig. 15, by using StyleClip.
9. bangs addition with e4e, HFGI and HyperStyle in Fig. 16, by using StyleClip.
10. bob cut hairstyle with e4e, HFGI and HyperStyle in Fig. 17, by using StyleClip.
11. car color change with e4e, HFGI and HyperStyle in Fig. 18, by using GanSpace.
12. grass addition with e4e, HFGI and HyperStyle in Fig. 19, by using GanSpace.

References

[1] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6711–6720, 2021. 1

[2] 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. 1, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20

[3] Zikun Chen, Ruowei Jiang, Brendan Duke, Han Zhao, and Parham Aarabi. Exploring gradient-based multi-directional controls in gans. In *European Conference on Computer Vision*, pages 104–119. Springer, 2022. 2

[4] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[5] Xueqi Hu, Qiusheng Huang, Zhengyi Shi, Siyuan Li, Changxin Gao, Li Sun, and Qingli Li. Style transformer for image inversion and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11337–11346, June 2022. 1, 5, 6

[6] 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. 1

[7] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094, 2021. 2

[8] 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. 1

[9] 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, 5, 6

[10] 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. 2

[11] 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, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20

[12] Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. High-fidelity gan inversion for image attribute editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11379–11388, 2022. 1, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20

[13] 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. 1, 5, 6

Figure 4. Qualitative results of inversion and editing. For each method, first column shows inversion, and second shows editing.

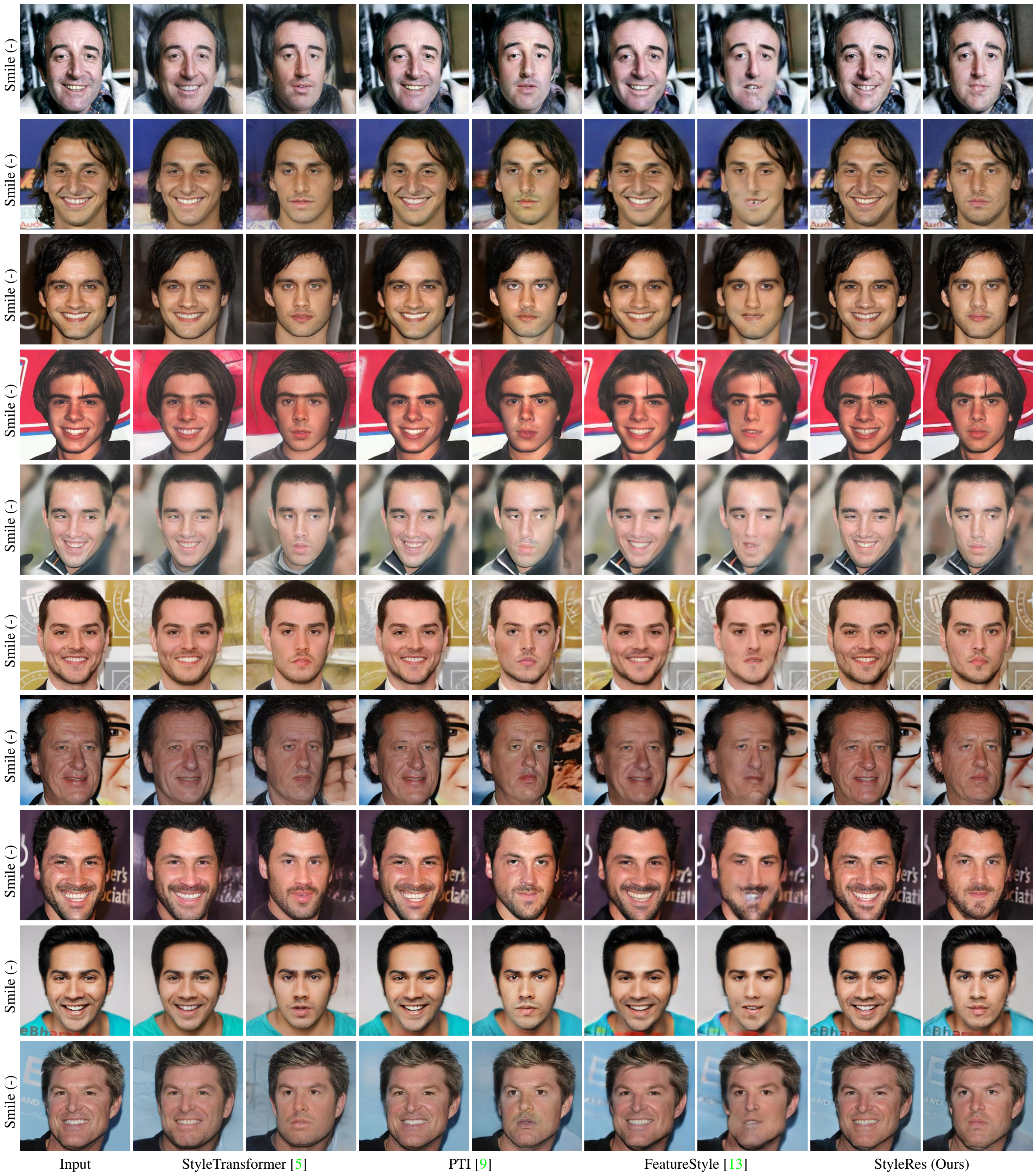

Figure 5. Qualitative results of inversion and editing. For each method, first column shows inversion, and second shows editing.

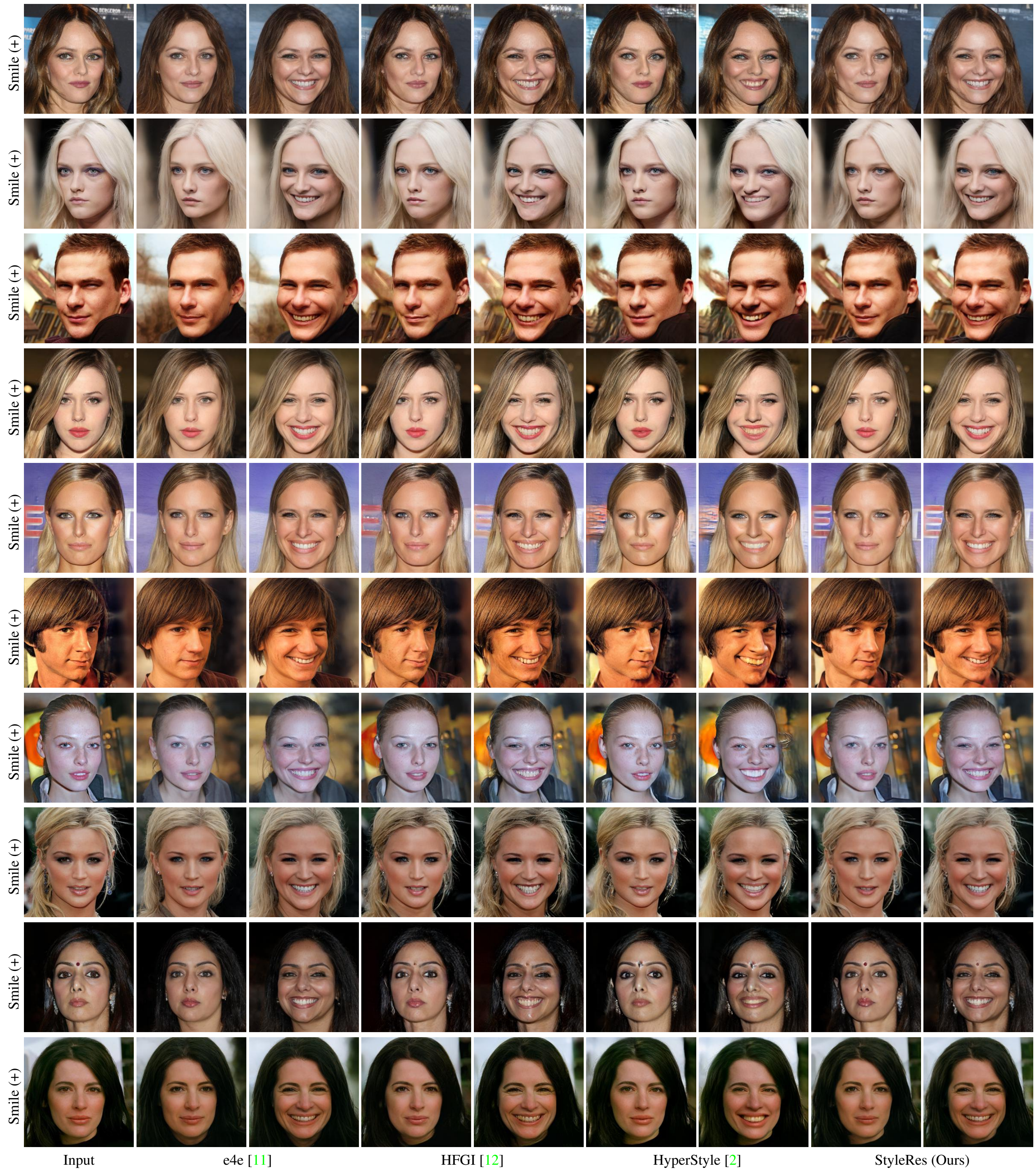

Figure 6. Qualitative results of inversion and editing. For each method, first column shows inversion, and second shows editing.

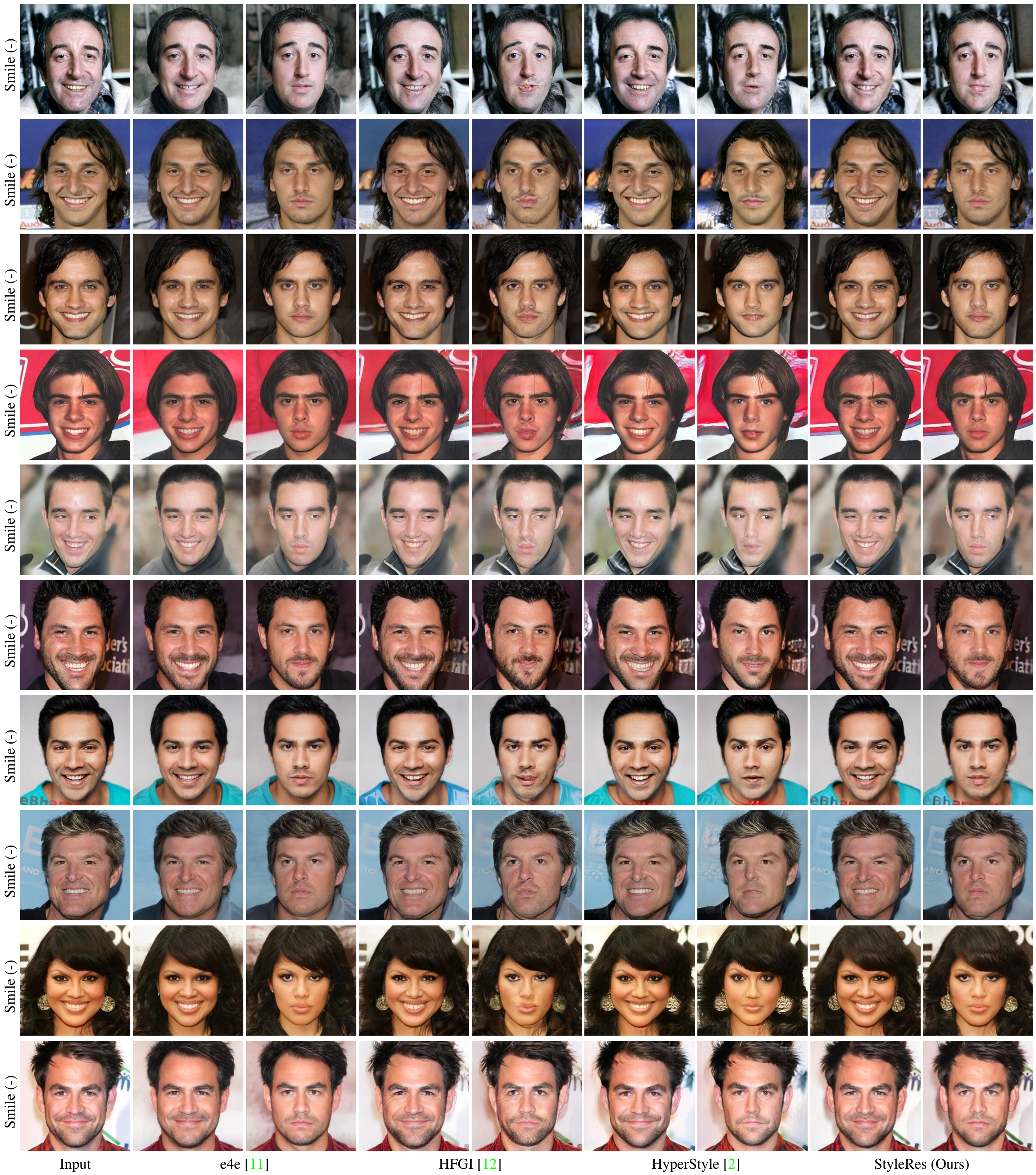

Figure 7. Qualitative results of inversion and editing. For each method, first column shows inversion, and second shows editing.

Figure 8. Qualitative results of inversion and editing. For each method, first column shows inversion, and second shows editing.

Figure 9. Qualitative results of inversion and editing. For each method, first column shows inversion, and second shows editing.

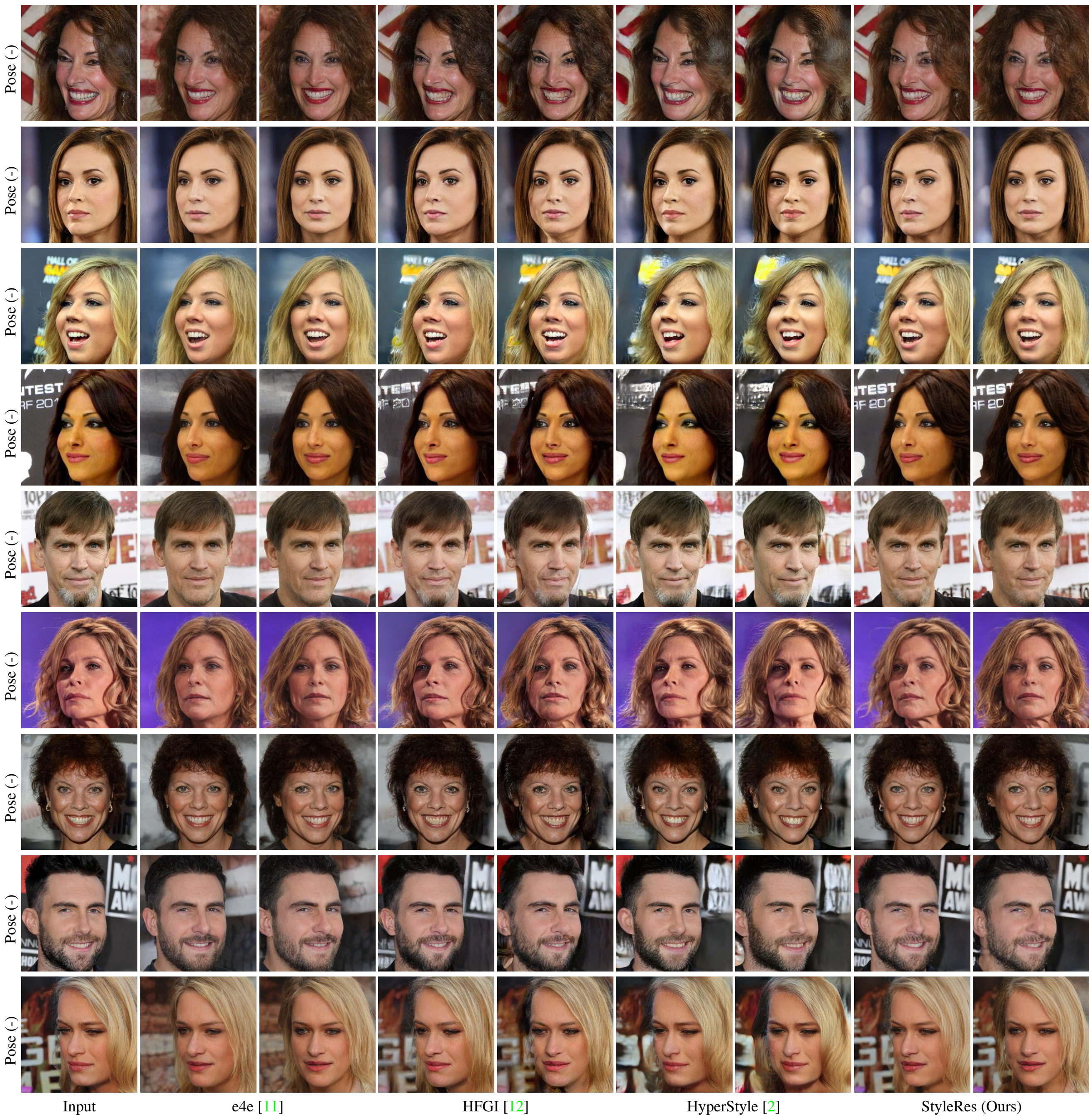

Figure 10. Qualitative results of inversion and editing. For each method, first column shows inversion, and second shows editing.

Figure 11. Qualitative results of inversion and editing. For each method, first column shows inversion, and second shows editing.

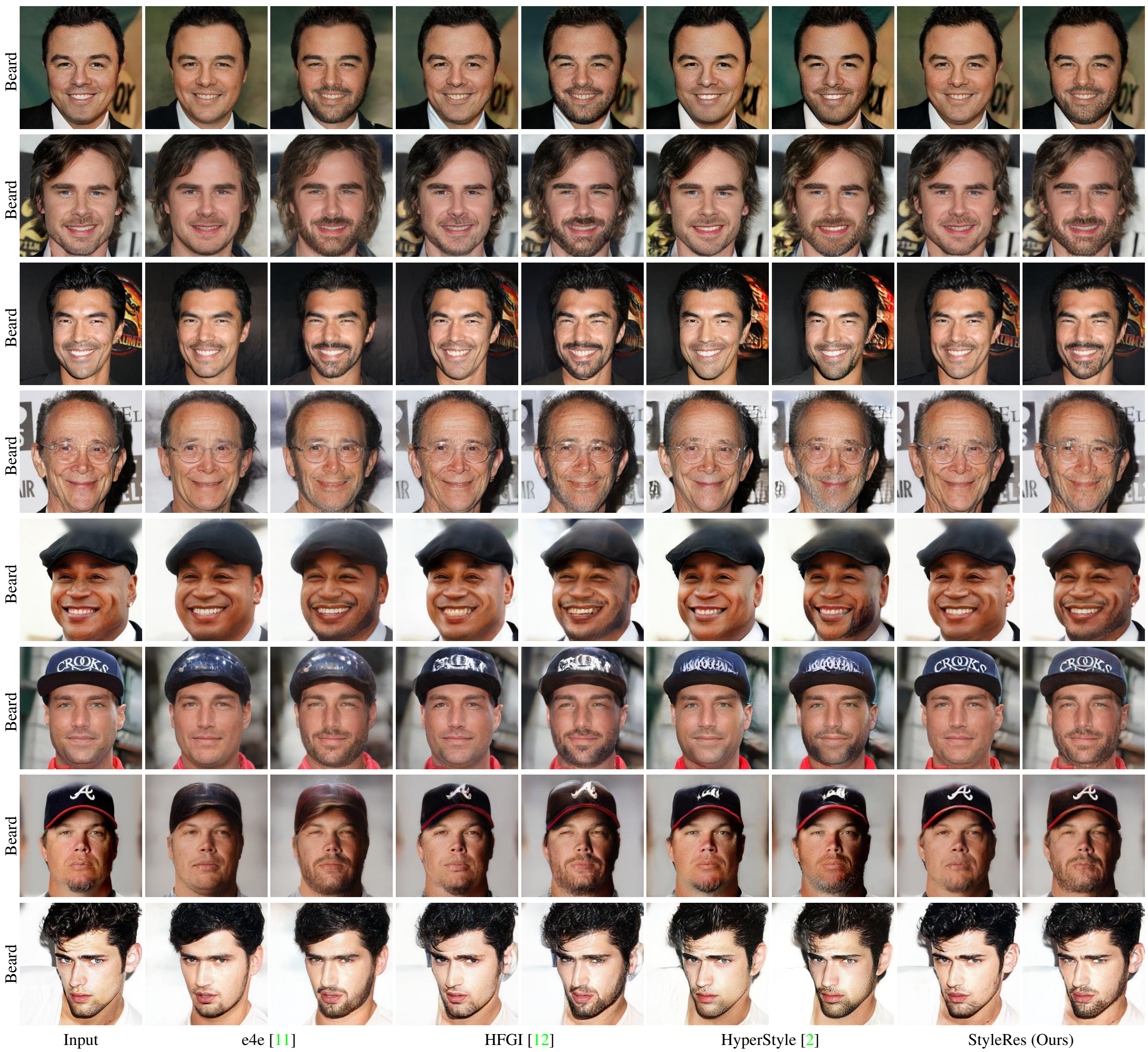

Figure 12. Qualitative results of inversion and editing. For each method, first column shows inversion, and second shows editing.

Figure 13. Qualitative results of inversion and editing. For each method, first column shows inversion, and second shows editing.

Figure 14. Qualitative results of inversion and editing. For each method, first column shows inversion, and second shows editing.

Figure 15. Qualitative results of inversion and editing. For each method, first column shows inversion, and second shows editing.

Figure 16. Qualitative results of inversion and editing. For each method, first column shows inversion, and second shows editing.

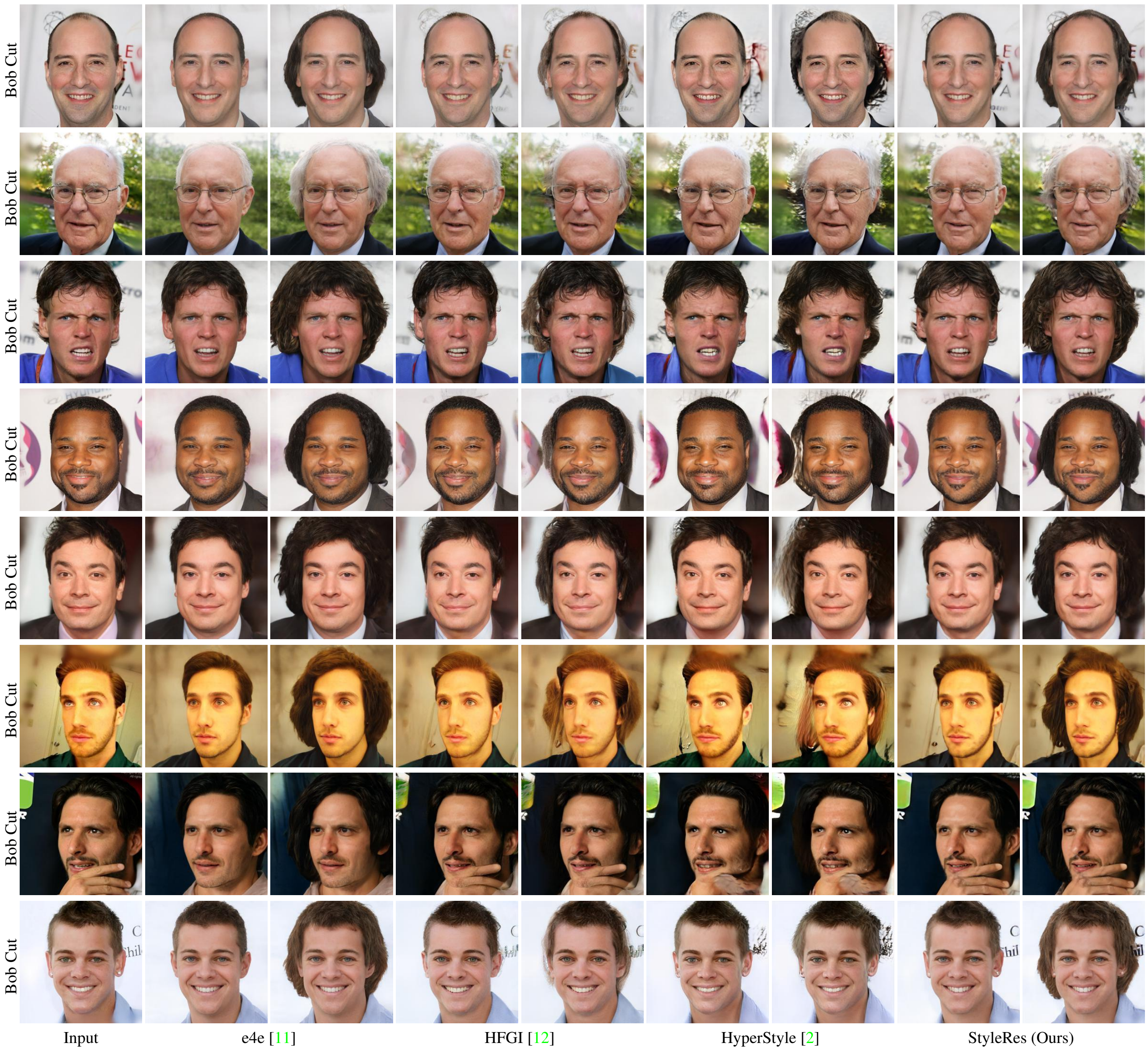

Figure 17. Qualitative results of inversion and editing. For each method, first column shows inversion, and second shows editing.

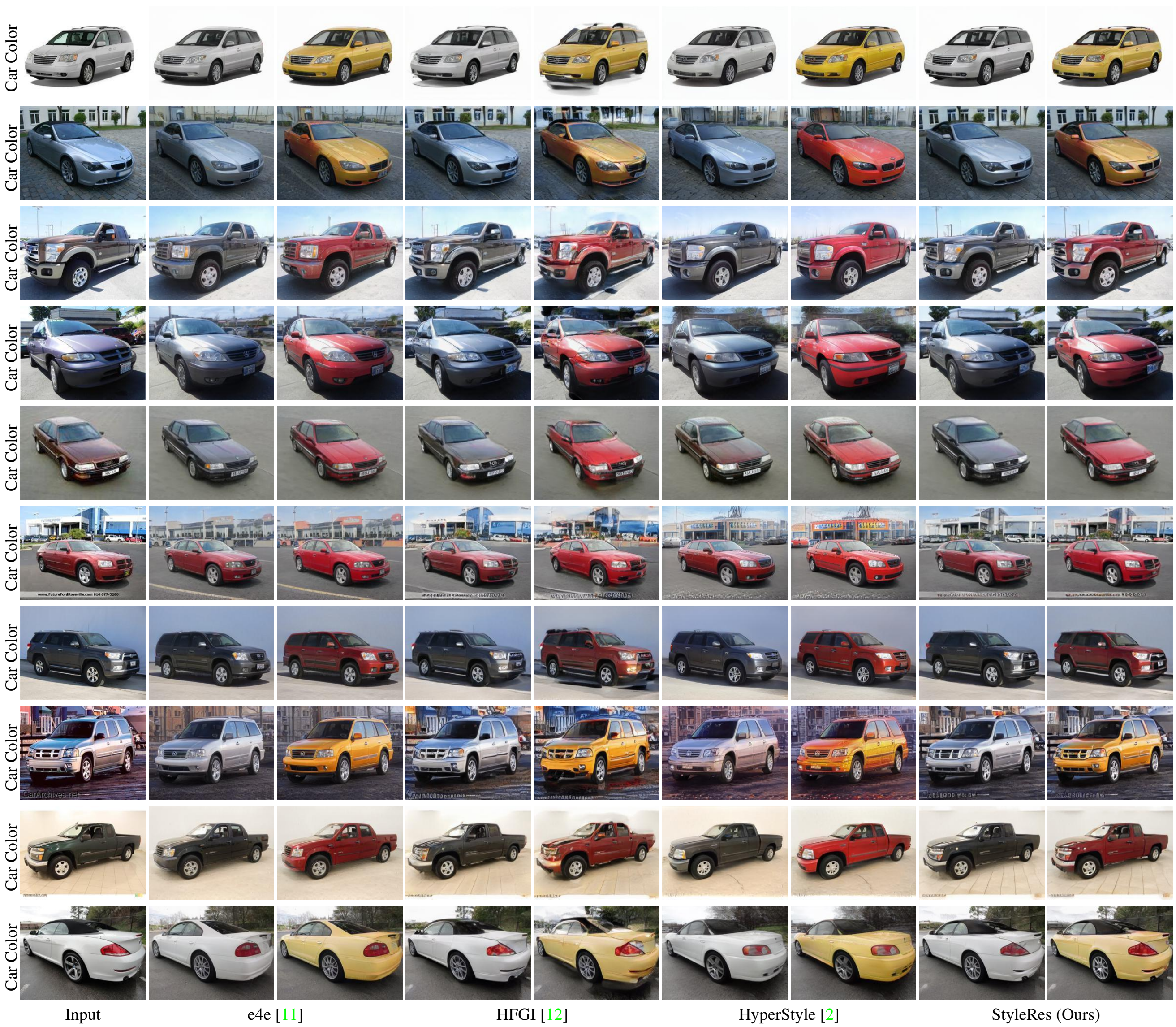

Figure 18. Qualitative results of inversion and editing. For each method, first column shows inversion, and second shows editing.

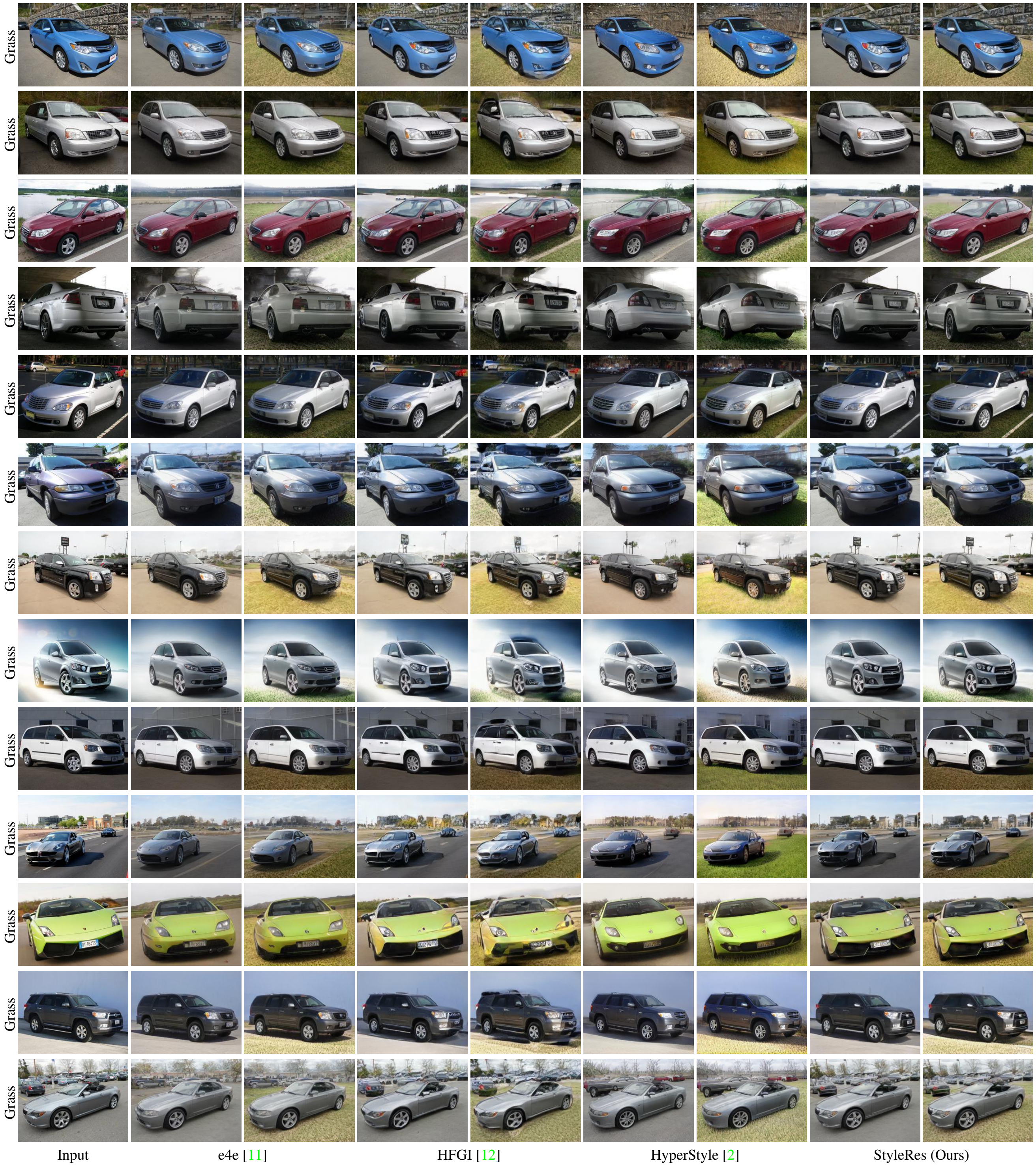

Figure 19. Qualitative results of inversion and editing. For each method, first column shows inversion, and second shows editing.