Self-Supervised Geometry-Aware Encoder for Style-Based 3D GAN Inversion

Supplementary Material

A. Background

Since recent 3D-aware image generative models are all based on neural implicit representations, especially NeRF [15], here we briefly introduce the NeRF-based 3D representation and more StyleSDF details for clarification.

NeRF-based 3D Representation. NeRF [15] proposed an implicit 3D representation for novel view synthesis. Specifically, NeRF defines a scene as $\{\boldsymbol{c}, \sigma\} = F_\Phi(\boldsymbol{x}, \boldsymbol{v})$, where $\boldsymbol{x}$ is the query point, $\boldsymbol{v}$ is the viewing direction from camera origin to $\boldsymbol{x}$, $\boldsymbol{c}$ is the emitted radiance (RGB value), σ is the volume density. To query the RGB value $C(\boldsymbol{r})$ of a point on a ray $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{v}$ shoot from the 3D coordinate origin $\boldsymbol{o}$, we have the volume rendering formulation,

$$C(\boldsymbol{r}) = \int_{t_n}^{t_f} T(t)\sigma(\boldsymbol{r}(t))\boldsymbol{c}(\boldsymbol{r}(t), \boldsymbol{v})dt, \tag{1}$$

where $T(t) = \exp(-\int_{t_n}^{t} \sigma(\boldsymbol{r}(s))ds)$ is the accumulated transmittance along the ray $\boldsymbol{r}$ from t_n to t. t_n and t_f denote the near and far bounds.

More StyleSDF Details. In hybrid 3D generation [4, 10, 16], the intermediate feature map is calculated by replacing the color $\boldsymbol{c}$ with feature $\mathbf{f}$ from ϕ_f, namely $\mathbf{F}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{f}(\mathbf{r}(t), \boldsymbol{v})dt$. In StyleSDF, the Sigmoid activation function σ is replaced by $\sigma(\boldsymbol{x}) = K_\alpha(d(\boldsymbol{x})) = \operatorname{Sigmoid}(-d(\boldsymbol{x})/\alpha)/\alpha$, where α is a learned parameter that controls the tightness of the density around the surface boundary.

Notation Table. For clarity, we include the notations used in the proposed method in Tab. 1.

B. Implementation Details

B.1. More Methods Details

Surface Point Sampling in Self-supervised Inversion Learning. In Sec. 4.1 of the main paper, to extract the 3D shape information $\mathcal{S}$ of each synthetic shape, we first sample a point set $\mathcal{P} = \{\mathcal{P}_\mathcal{O}, \mathcal{P}_\mathcal{F}\}$ where $\mathcal{P}_\mathcal{O}$ and $\mathcal{P}_\mathcal{F}$ contain points sampled from the surface and around the surface, respectively. To get points over the surface $\mathcal{P}_\mathcal{O}$ for training, for efficiency, we directly reuse the intermediate results to render $\mathbf{I_0}$ to calculate the surface. Specially, to sample point set $\mathcal{O}$ we replace the color $\mathbf{c}$ as the coordinates $\boldsymbol{x}$ of points along a ray in Eq. (1) and approximate the 3D coordinates of surface, namely $\boldsymbol{t}_s(\mathbf{w}, \boldsymbol{\xi}) = \int_{t_n}^{t_f} T(t, \mathbf{w})\sigma(\boldsymbol{r}(t), \mathbf{w})t \, dt$. In this way, we get $B \times H \times W$ surface points for training in each iteration, where B stands for batch size and $H \times W$ stands for the resolution to render 3D consistent images, *e.g.*, 64×64. To sample point set $\mathcal{F}$, we add Gaussian offset to each of the calculated surface points $\mathcal{O}$. Specifically, we adopt Gaussian distribution $\mathcal{N}(0, (r/4)^2)$ where r is the radius of the scene. In this way, points falling within 4 standard deviations would cover 95.44% of the whole 3D space. Following PIFu [22], we also uniformly sample $0.5 \times B \times H \times W$ points within the whole 3D space defined. The overall quantity of the point set surface is $|\mathcal{F}| = 1.5 \times B \times H \times W$. We find this sampling strategy avoids overfitting and yields better performance.

Training Details of High-Fidelity Inversion With Local Features. In Sec. 4.2 of the main paper, we train a local encoder $E_\mathbf{1}$ to extract pixel-aligned features to enrich texture details for high-fidelity inversion. The network architecture of $E_\mathbf{1}$ is identical to that of PIFu [22], which is a stacked hourglass network with residual connections. The input residual map resolution is 256×256, and the output 64×64 resolution feature map. $\mathbf{f}_\text{L} \in \mathcal{R}^{256}$ is bilinearly interpolated from feature map $\mathbf{F}_\text{L}$ at the projected position $\pi(\boldsymbol{x})$. As shown in Fig. 1, we implement the FiLM layer [17] with two MLP residual blocks [33], which outputs α and β for modulation, respectively. We use the identical learning rate and optimizer to train $E_\mathbf{1}$.

Novel-View Training Details. For novel-view training for coherent view synthesis in Sec. 4.3 of the main paper, in each training iteration with batch size n, rather than sampling n different latent codes $\{\mathbf{z}_i\}_{i=1}^{n}$, we halve the number of identical latent codes $\{\mathbf{z}_i\}_{i=1}^{n/2}$ while double the rendered images for each latent code $\{\mathbf{I}_i^{\boldsymbol{\xi}_1}, \mathbf{I}_i^{\boldsymbol{\xi}_2}\}_{i=1}^{n/2}$ where n is even. Thus, we train the models to reconstruct plausible *novel views*, *i.e.*, $G(E(\mathbf{I}_i^{\boldsymbol{\xi}_1}), \boldsymbol{\xi}_2) \approx \mathbf{I}_i^{\boldsymbol{\xi}_2}$ and $G(E(\mathbf{I}_i^{\boldsymbol{\xi}_2}), \boldsymbol{\xi}_1) \approx \mathbf{I}_i^{\boldsymbol{\xi}_1}$. Since the paired-sampled images could serve as both inputs and ground truths, the effective batch size and training cost maintains the same. To train 2D alignment model E_ADA, we further regularize the predicted residual map $\hat{\Delta}^{\boldsymbol{\xi}_1} \approx \mathbf{I}^{\boldsymbol{\xi}_1} - \mathbf{I}_\mathbf{0}^{\boldsymbol{\xi}_1}$ with $\mathcal{L}_1$ loss, where $\mathbf{I}_\mathbf{0}^{\boldsymbol{\xi}_1}$ is corresponding renderer output low-resolution image and $\lambda_1 = 0.1$. Note that we finetune pre-trained E_{ADA} from HFGI [28] with novel-view training and no edited images are involved in the training time.

Curriculum Pose Sampling. At the beginning of the training of the hybrid alignment in Sec. 4.3 of the main paper, large view changes will make the prediction of residual features and the inpainting of occlusion regions extremely difficult. As a result, our model is prone to blurry results. We attribute the reason to the ill-posed nature of rendering novel views given partial observations since the inpainted image is not unique. To facilitate novel-view training, we design a curriculum learning strategy [7] based on *pose sampling difficulty*. Implementation wise, given the camera pose distribution $\boldsymbol{\xi} \sim p_{\boldsymbol{\xi}}$ with mean μ and standard variance σ, we fix the μ and scale the σ with a weight α which is initially set to 0 and gradually increases to 1 as the training goes. Intuitively, when $\alpha = 0$ the source view $\boldsymbol{\xi}$ is identical to the query view $\boldsymbol{\xi}'$, the training degrades to a regression task where the model shall reconstruct all the texture details to minimize the loss. As the variance $\alpha \cdot \sigma$ increases, the training becomes a conditional generation task to inpaint plausible and photo-realistic areas.

B.2. More Experiments Details

Training Details. In this work, we directly use the officially released pre-trained GAN models from StyleSDF. In self-supervised shape inversion learning (Sec. 4.1), due to GPU memory restriction, we sample 4 shapes per GPU each iteration for training. After $E_{\mathbf{0}}$ converged, we fix the network weights and only train the $E_{\mathbf{1}}$ for high-fidelity inversion. We train each stage for $50,000$ iterations, which costs 2 days on 4 Tesla V100 GPUs.

Network Architecture Details. For $E_{\mathbf{0}}$, a modified version of the pSp encoder [20] is deployed here for a fair comparison with existing work. Since G_0 and G_1 of StyleSDF have 9 and 10 latent codes, respectively, we introduce $9 + 10$ extra prediction heads to the pSp for the latent code prediction. We observe that early layers of $G_{\mathbf{0}}$ control the geometry of generated samples, and later $G_{\mathbf{0}}$ layers as well as decoder generator $G_{\mathbf{1}}$ control the texture and high-frequency details. Thus, we adopt the early pSp feature map of resolution 32×32 to predict latent code of G_0 for geometry control, and pSp feature map of resolution 64×64 to predict latent code of G_0 for texture control. We use the highest resolution feature map of pSp with resolution 128×128 to predict the latent code for G_1. We show our FiLM layer implementation in Fig. 1, where the input features are modulated by the input conditions with predicted γ, and β. The MLP is implemented with the MLP residual block [33].

Editing. For attribute editing, following previous works, we adopt vector-arithmetic [19] based editing. Specifically, a searched latent code vector paired with a certain attribute

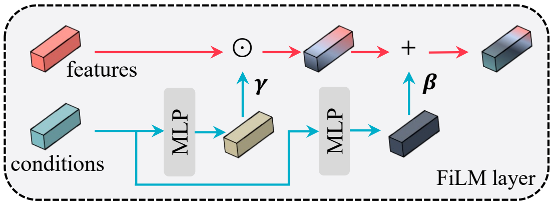

Figure 1. FiLM Layer Architecture.

is weighted and added to the predicted code $\hat{\mathbf{w}}$. To search for the meaningful editing directions on the 3D GAN used, we first sample $10,000$ images with paired latent codes from StyleSDF, and then apply the face attribute predictor from Talk-to-Edit [11] to predict the corresponding attributes score. Based on the prediction, we apply SVM classifier from InterfaceGAN [25] to search for the decision boundary. As in previous works [20, 26], we search for the editing latent code in the $\mathcal{W}$ space.

3D Face Reconstruction Evaluation Details. We evaluate the reconstructed 3D meshes and compare them with the performance of several model-based reconstruction methods on NoW benchmark [23]. NoW benchmark [23], provides a test set of $1,702$ images of 80 subjects and a ground-truth 3D scan per subject. These images are captured with a higher variety in facial expression, occlusion, and lighting and shall validate the generality of single-view reconstruction methods under real-world conditions.

To extract meshes for evaluation, we detect faces and crop the images using RetinaFace [24] implemented by [29] and obtain 3D mesh reconstructions from the depth maps predicted by our method trained on FFHQ pre-trained generator. We then use the evaluation protocol provided by the benchmark, which aligns the predicted meshes with the ground-truth meshes with a rigid transformation based on seven pre-defined keypoints and computes the scan-to-mesh distances. We obtain keypoints on our predicted meshes by applying a facial keypoint detector [30] on the reconstructed canonical images. Following Unsup3D [32], the average keypoints are used when the keypoint detector fails.

Video Trajectory Evaluation Details. We sample 500 trajectory videos with pre-trained FFHQ StyleSDF generator with an ellipsoid trajectory of size 250 from official StyleSDF code, making a dataset of size $12,5000$. The evaluation code and dataset will be released.

Computational cost. We include the computational cost of each component in the table below.

Component	$E_{\mathbf{0}}$ (pSp)	$E_{\mathbf{1}}$	E_{ADA}
Parameters(M)	219.71	14.06	0.60
MACs(G)	62.95	26.07	4.03

Comparisons with Optimization-based Methods. In this paper, we include the comparisons with two canonical optimization-based methods here, namely SG2 [1, 12] which is initially proposed in StyleGAN [12] paper to project input image to the $\mathcal{W}$ space of the paired generator, and PTI [21] which further finetune the generator weights to achieve high-fidelity inversion. We implement SG2 and PTI following the official implementations and tune the corresponding parameters for StyleSDF generator. For SG2, we optimize 450 steps with learning rate $5e-3$, and for the pivotal tuning stage, we optimize 100 steps with learning rate $5e-5$. We adopt an open source EG3D projection implementation [31] for all EG3D inversion experiments.

B.3. Losses

Reconstruction Loss. We briefly introduce the supervisions we adopt in image reconstructions in both training stages. First, we utilize the pixel-wise $\mathcal{L}_2$ loss,

$$\mathcal{L}_2(\mathbf{I}) = ||\mathbf{I} - \hat{\mathbf{I}}||_2. \tag{2}$$

In addition, to learn perceptual similarities, we use the LPIPS [34] loss, which has been shown to better preserve image quality compared to the more standard perceptual loss:

$$\mathcal{L}_{\text{LPIPS}}(\mathbf{I}) = ||F(\mathbf{I}) - F(\hat{\mathbf{I}})||_2, \tag{3}$$

where $F(\cdot)$ denotes the perceptual feature extractor.

Finally, a common challenge when handling the specific task of encoding facial images is the preservation of the input identity. To tackle this, we incorporate a dedicated recognition loss measuring the cosine similarity between the output image and its source,

$$\mathcal{L}_{\text{Id}}(\mathbf{I}) = 1 - \langle R(\mathbf{I}), R(E_g(\mathbf{I}))\rangle, \tag{4}$$

where R is the pretrained ArcFace [5] network.

In summary, the total loss function is defined as

$$\mathcal{L}_{rec}(\mathbf{I}) = \lambda_1\mathcal{L}_2(\mathbf{I}) + \lambda_2\mathcal{L}_{\text{LPIPS}}(\mathbf{I}) + \lambda_3\mathcal{L}_{\text{Id}}(\mathbf{I}),$$

where we set $\lambda_1 = 1$, $\lambda_2 = 0.8$, $\lambda_3 = 0.1$ as the defined loss weights. In $E_\mathbf{0}$ training, we supervise images $\hat{\mathbf{I}}_\mathbf{0}, \hat{\mathbf{I}}_\mathbf{1}$ of both resolutions. In $E_\mathbf{1}$ training, we only supervise the reconstruction of high-resolution images since the network weights to render $\hat{\mathbf{I}}_\mathbf{0}$ is fixed. Here, we also impose the non-saturating adversarial loss with R1 regularization [14] to improve the naturalness of reconstructed images, which is defined as:

$$\mathcal{L}_{adv} = -\mathbb{E}[log(D(\hat{\mathbf{I}}))], \tag{5}$$

$$\mathcal{L}_D = \mathbb{E}[log(D(\hat{\mathbf{I}}))] + \mathbb{E}[log(1 - D(\mathbf{I}))], \tag{6}$$

$$\mathcal{L}_{R1} = \lambda\|\nabla D(\hat{\mathbf{I}}; \theta_D)\|_2, \tag{7}$$

where D is initialized with the pre-trained discriminator paired with the generator and θ_D is the corresponding parameters to optimize. In summary, the overall loss is the weighted summation of of the loss functions described above:

$$\mathcal{L} = \mathcal{L}_{geo} + \mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_D\mathcal{L}_D + \lambda_{R1}\mathcal{L}_{R1}, \tag{8}$$

where we set $\lambda_D = \lambda_{adv} = 0.01$ and $\lambda_{R1} = 10$ in the experiments.

C. More Results

E3DGE on Other GANs and Categories. Besides FFHQ-StyleSDF in the paper, we show the performance of our method on FFHQ-EG3D, AFHQ-EG3D(Fig. 2), and ShapeNet-StyleSDF (Fig. 3). As can be seen, our method achieves high-quality shape and texture inversion on both SoTA radiance-based (EG3D) and sdf-based (StyleSDF) NeRF GANs, demonstrating the generalizability of our method. We also attempted cat faces apart from human faces.

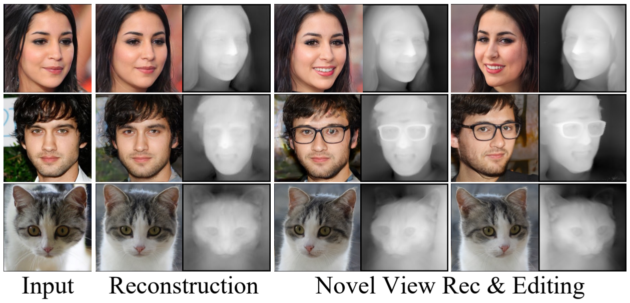

Figure 2. **E3DGE qualitative performance on EG3D Base Model.** Rows 1-2: the inversion result on the CelebA-HQ test set, with +Smiling and +Eyeglass attributes editing, respectively. Row 3: the inversion and view synthesis results of AFHQ cat.

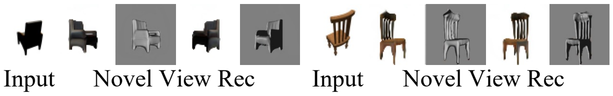

Figure 3. **E3DGE qualitative performance on ShapeNet Chair.**

We also include the preliminary quantitative benchmark of our method on EG3D in the Tab. 2, which demonstrates the generality of our method on high-fidelity inversion.

Table 2. **Quantitative performance on EG3D (FFHQ).**

Methods	MAE ↓	SSIM ↑	LPIPS ↓	Similarity ↑
pSp-EG3D	0.251±.03	0.633±.02	0.41±.04	0.377±.05
Ours-EG3D	0.143±.03	0.688±.02	0.30±.03	0.650±.04

3D Reconstruction. We report the 3D face reconstruction performance on NoW benchmark test set in Tab. 3.

Table 1. Notations used in the proposed method.

Notation	Meaning
$\hat{*}$	Final predictions
$\tilde{*}$	Intermediate results
$*'$	Abbreviation of target view camera pose
G	Generator
$G_{\mathbf{0}}$	Renderer Generator
$G_{\mathbf{1}}$	SR Generator
D	Discriminator
E	Encoder
$E_{\mathbf{0}}$	Encoder to predict global latent code
$E_{\mathbf{1}}$	Hourglass encoder to predict pixel-aligned local features.
E_{ADA}	ADA (Adaptive Distortion Alignment) module
$\mathcal{W}$	W space for style-based GAN
$\mathbf{w}$	Latent code sampled from W space
$\mathbf{I}$	Input image
$\mathbf{I_0}$	Rendered image from renderer generator
$\mathbf{I}_{edit}$	Edited image
$\hat{\mathbf{w}}$	Predicted latent code from $E_{\mathbf{0}}$
λ	Loss weights
$\boldsymbol{x}$	3D point
$\mathcal{P}$	Point set
$\mathcal{P}_{\mathcal{O}}$	Point set sampled from object surface
$\mathcal{P}_{\mathcal{F}}$	Point set sampled near the surface or uniformly in the defined 3D space.
d	Signed distance function
$\boldsymbol{n}$	Normal for a point
ϕ_g	MLP to predict geometry
ϕ_f	MLP to predict view-dependent feature
ϕ_c	MLP to predict color
$\boldsymbol{v}$	View direction
$\mathcal{X}$	A synthetic data sample for training
$\boldsymbol{\xi}$	Source view camera pose
$\boldsymbol{\xi}'$	Target view camera pose
Δ	Residual of predicted image and input image
Δ_{edit}	Residual paired with an edited image
Δ'_{edit}	Residual paired with an edited image rendered from target camera pose.
$\pi(\boldsymbol{x})$	Projection of 3D point $\boldsymbol{x}$ to source view
$\oplus$	Concatenation
PE	Positional Encoding
β, γ	Modulation signals for FiLM
$\mathbf{t}_s(\mathbf{w}, \boldsymbol{\xi})$	Depth map for code $\mathbf{w}$ rendered from pose $\boldsymbol{\xi}$
$\mathbf{F}$	Feature map
$\mathbf{F}_{\text{L}}$	Local feature map output from $E_{\mathbf{1}}$
$\hat{\mathbf{F}}$	Modulated feature map for final prediction
$\mathbf{F}_{\text{ADA}}$	Local feature map output from $E_{\mathbf{1}}$ with E_{ADA} aligned residual
$\mathbf{f}_{\text{G}}$	Global feature output from the generator.
$\mathbf{f}_{\text{L}}$	Local feature interpolated from $\mathbf{F}_{\text{L}}$
$\mathbf{f}_{\text{ADA}}$	Aligned feature interpolated from $\mathbf{F}_{\text{L}}$
$\hat{\mathbf{f}}_{\text{L}}$	Predicted local feature for final prediction

As can be seen, our method surpasses purely model-free method [32] and shows competitive performance compared with methods designed for 3D face reconstruction using basic models, *e.g.*, 3DMM [2] and FLAME [13]. Note that as discussed in Wu *et al*. [32], NoW benchmark is designed for model-based reconstruction methods and inherently put model-free approaches at a disadvantage. Moreover, since the backbones (3DMM/FLAME and StyleSDF) and research problems are different, these methods are not directly comparable to our method Therefore, we include the comparisons just for reference and intend to demonstrate our method yields high-quality geometry inversion and even better results against the 3D reconstruction method [32] that does not rely on 3DMM or FLAME. Our comparisons could serve as a reference for fair quantitative evaluation comparisons of future model-free methods.

Table 3. Performance of 3D face reconstruction on NoW [23].

Methods	Prior Type	Median↓	Mean↓	Std
3DMM-CNN [27]	3DMM	1.84	2.33	2.05
PRNet [9]	3DMM	1.50	1.98	1.88
RingNet [23]	FLAME	1.21	1.54	1.31
3DDFA-V2	3DMM	1.23	1.57	1.39
DECA [8]	FLAME	1.09	1.38	1.18
Wu et al. [32]	Model Free	2.64	3.29	2.86
SG2 [12]	3D GAN	1.89	2.23	1.82
PTI [21]	3D GAN	2.86	3.54	3.01
$\text{pSp}_{\text{StyleSDF}}$	3D GAN	1.97	2.43	2.05
$\text{e4e}_{\text{StyleSDF}}$	3D GAN	2.83	3.40	2.67
Ours	3D GAN	**1.70**	**2.08**	**1.67**

More Comparisons with Encoder-based Methods. Here, we include more comparisons with encoder-based methods in Fig. 4. Our method achieves consistently better performance compared to the baselines in terms of reconstruction fidelity and editing visual quality.

More Editing Results. We show more editing results on changing 4 semantic attributes of our proposed method, namely smile (Fig. 5), hair/beard (Fig. 6), age (Fig. 7) and bangs (Fig. 8). Our method shows promising performance with shape-texture consistent editing. Note that since StyleSDF is still built on an MLP-based generator [3] and InterfaceGAN [25] is also not designed for 3D GANs, the editing performance is hindered to some extent and cannot achieve comparable performance compared with 2D StyleGAN. However, we believe this limitation could be alleviated in the future by adopting better-designed 3D GAN architecture, *e.g.*, tri-plane [4] and vision transformer [6]. Our results unleash the potential of this field and show that 3D consistency and high-fidelity reconstruction with high-quality editing are also achievable in recently developed 3D GAN. We hope our method could inspire later work in this field.

More Toonify Results. We show 3D toonify-stylized results over real-world faces using our proposed method in Fig. 9. Following [18], we finetune the pre-trained generator G for 400 iterations with 317 cartoon face images and use our pre-trained encoder E for inference. Visually inspected, the toonified results holds the cartoon style and also preserve identity of the input image, which demonstrates the potential of applying our method over downstream tasks.

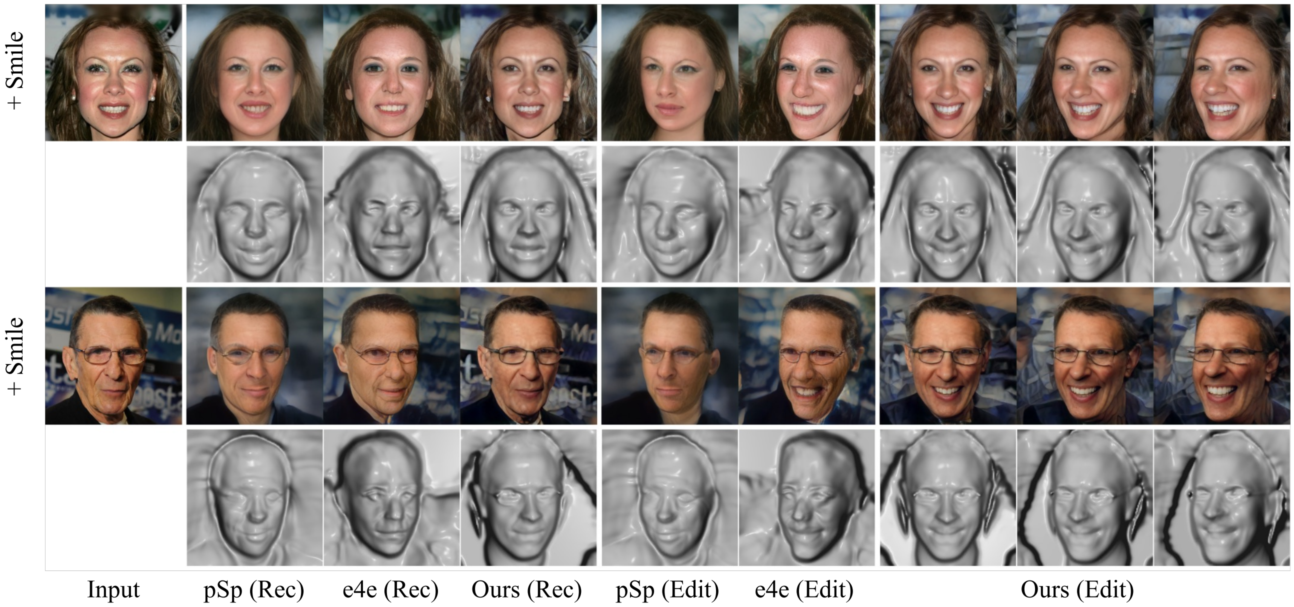

Figure 4. Visual comparisons on encoder-based methods. ’Rec’ and ’Edit’ represent reconstruction and editing, respectively.

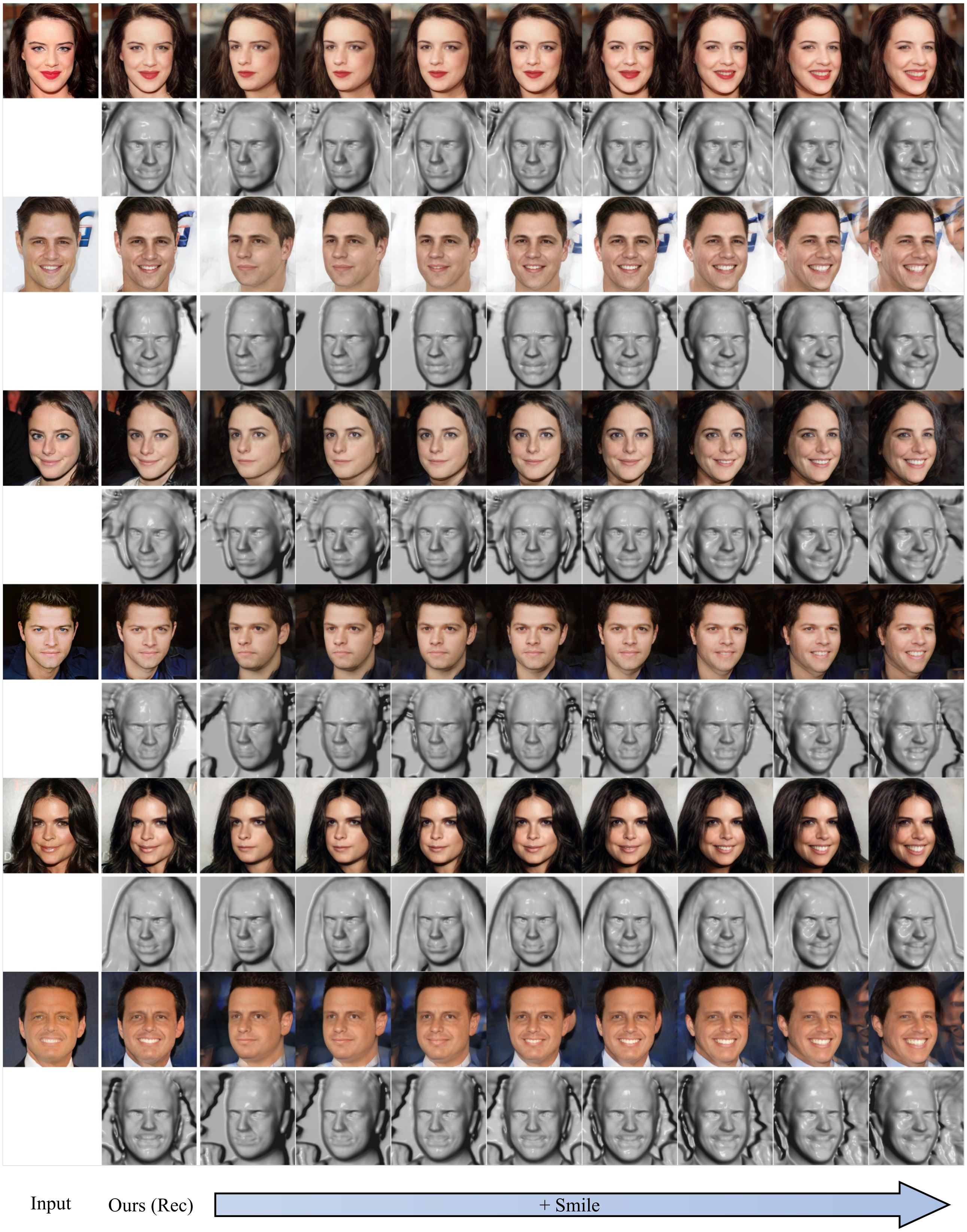

Figure 5. Visual comparisons on face editing (Smile).

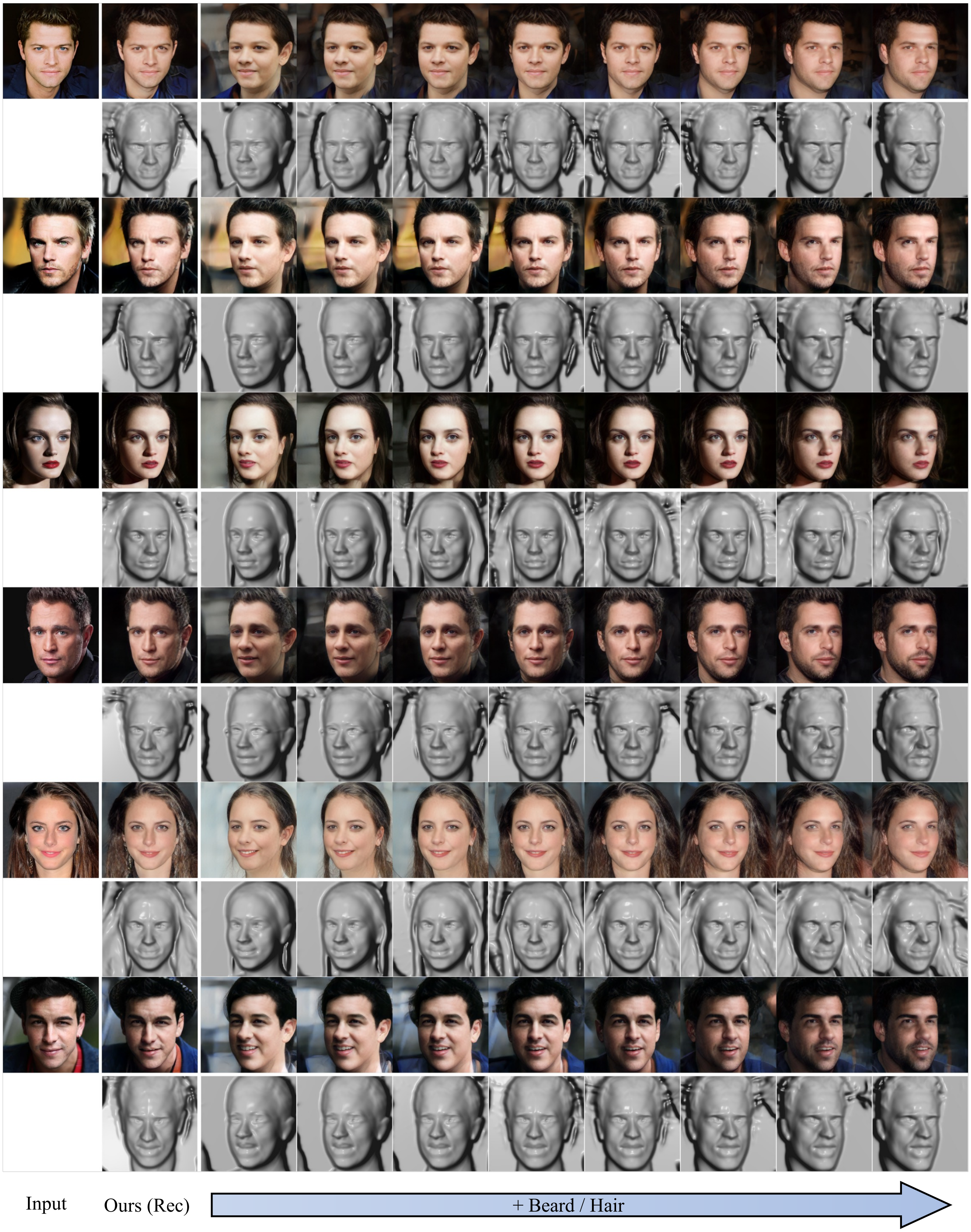

Figure 6. Visual comparisons on face editing (Beard / Hair).

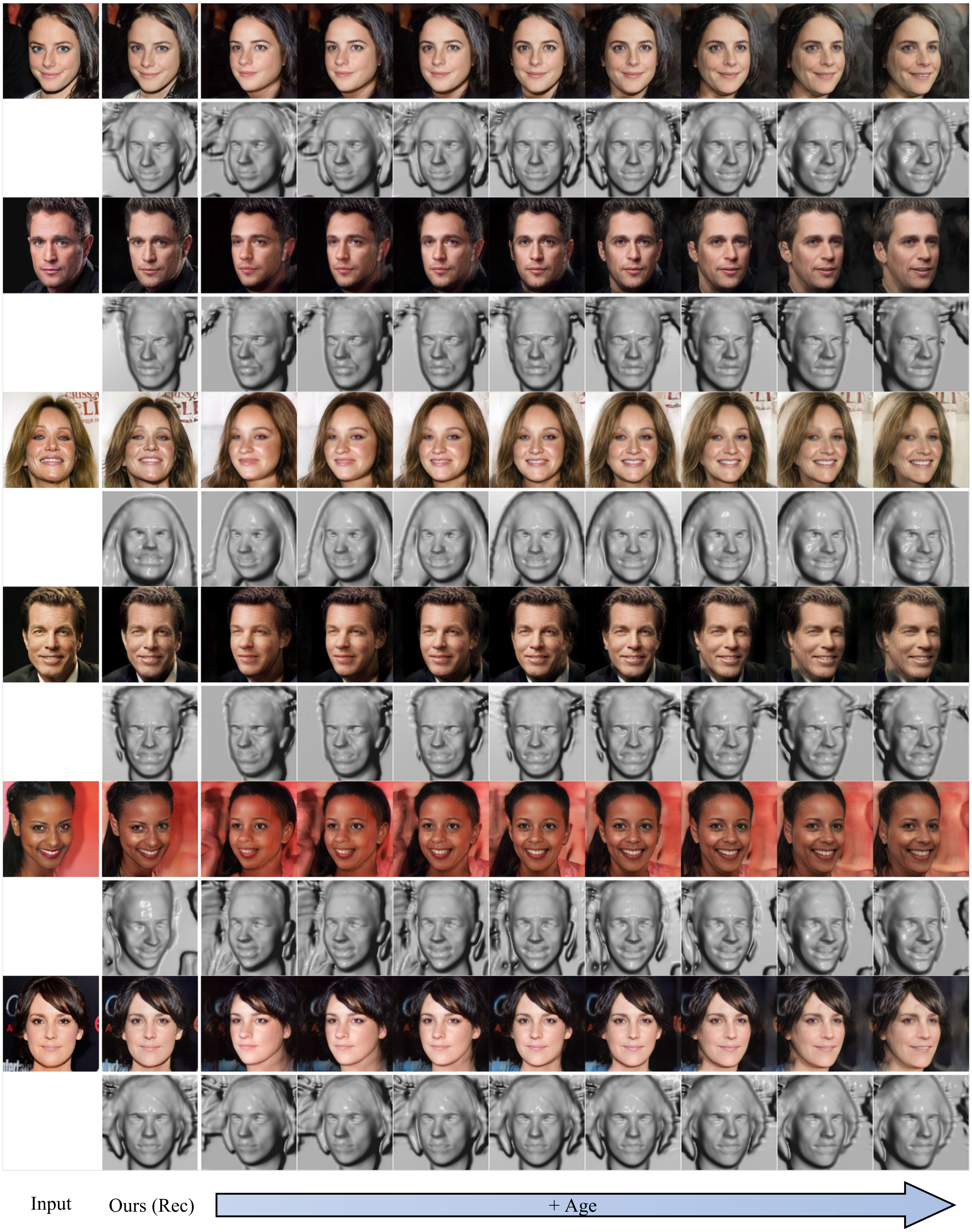

Figure 7. Visual comparisons on face editing (Age).

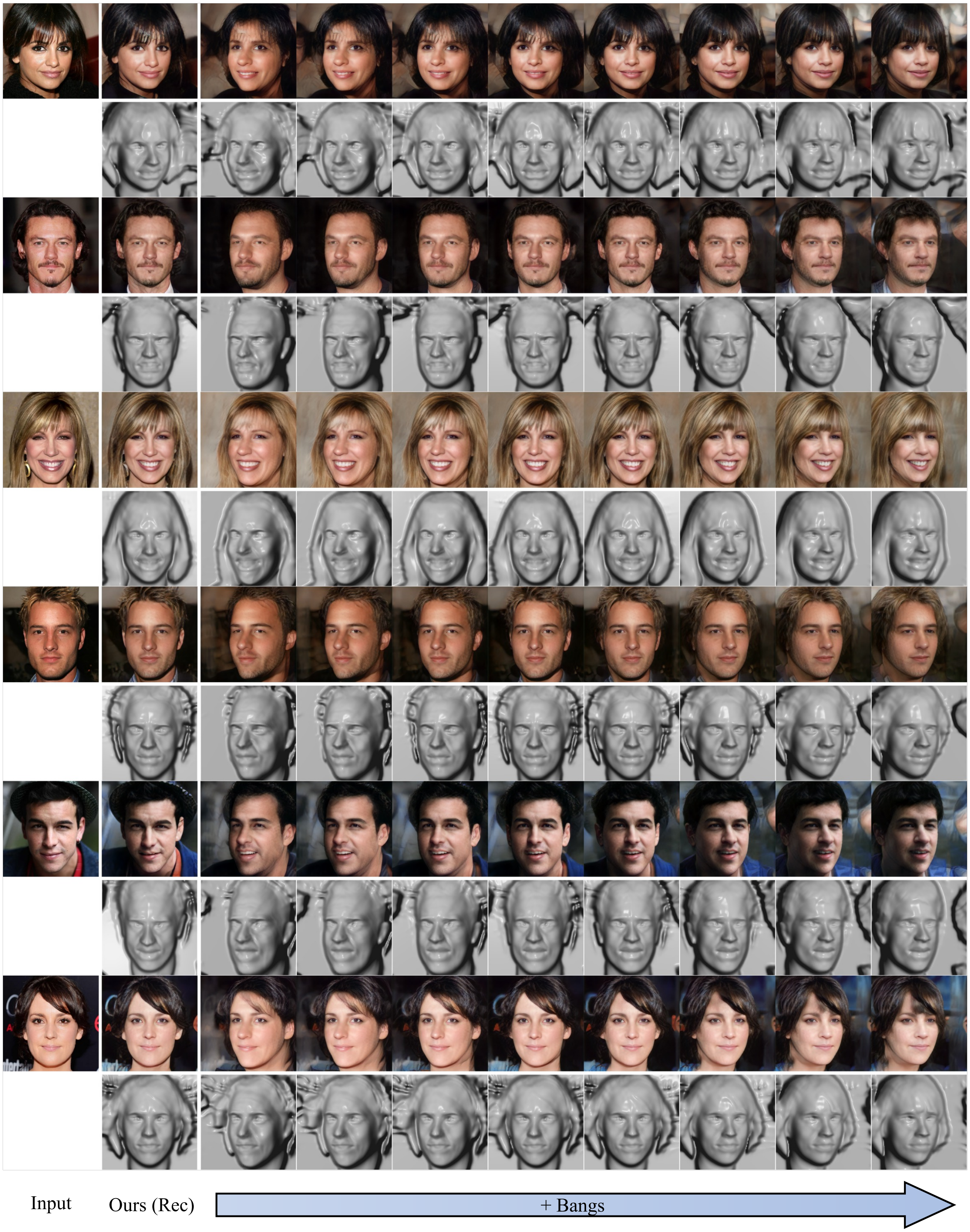

Figure 8. Visual comparisons on face editing (Bangs).

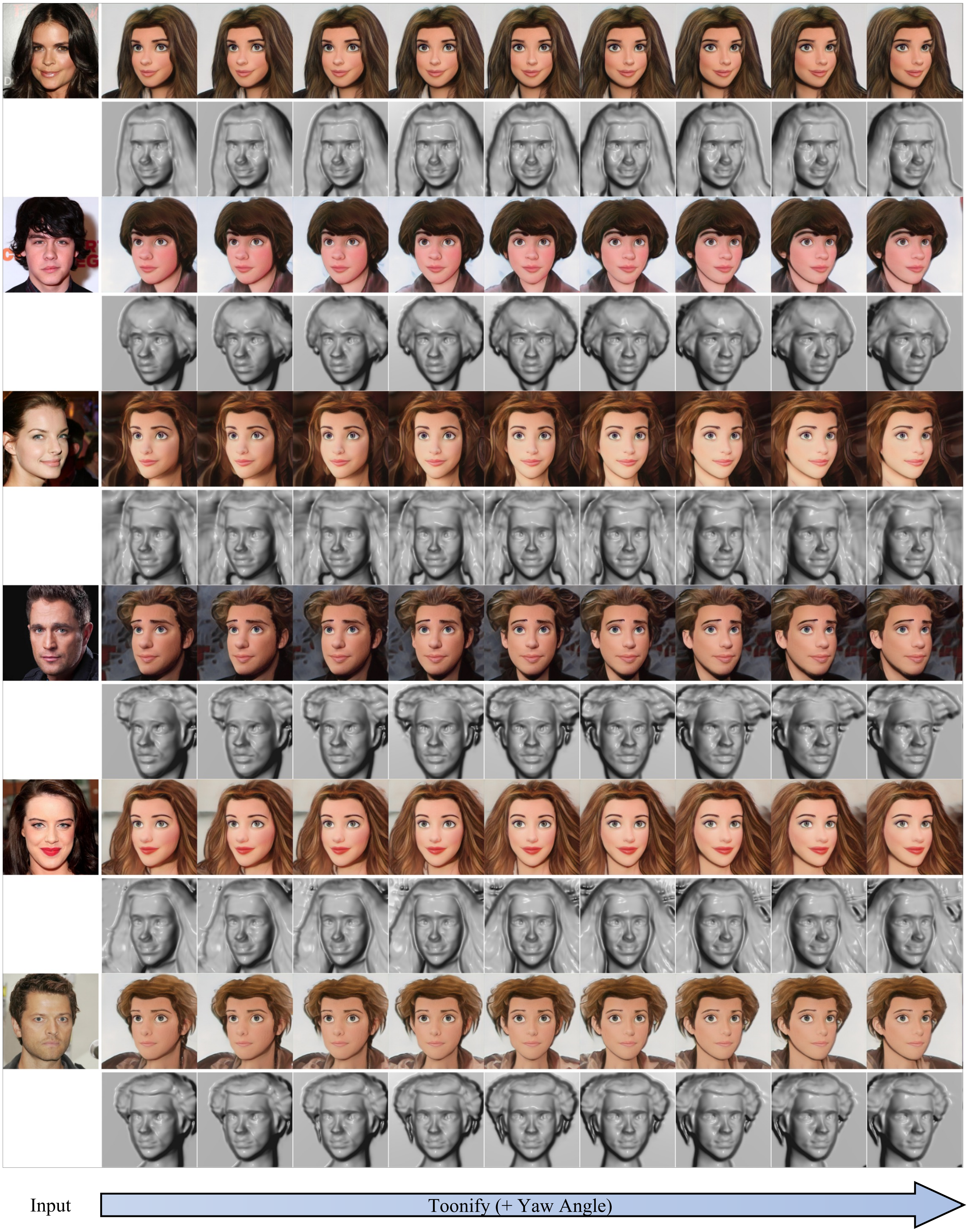

Figure 9. Toonify results on faces.

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*, 2019. 3

[2] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, 1999. 5

[3] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and G. Wetzstein. Pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis. In *CVPR*, 2021. 5

[4] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022. 1, 5

[5] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 3

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 5

[7] Yueqi Duan, Haidong Zhu, He Wang, Li Yi, Ram Nevatia, and Leonidas J. Guibas. Curriculum DeepSDF, 2020. 2

[8] Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. Learning an animatable detailed 3D face model from in-the-wild images. In *SIGGRAPH*, volume 40, 2021. 5

[9] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *ECCV*, 2018. 5

[10] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. StyleNeRF: A style-based 3D-aware generator for high-resolution image synthesis. In *ICLR*, 2021. 1

[11] Yuming Jiang, Ziqi Huang, Xingang Pan, Chen Change Loy, and Ziwei Liu. Talk-to-Edit: Fine-grained facial editing via dialog. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 2

[12] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 3, 5

[13] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *TOG*, 36(6), 2017. 5

[14] Lars M. Mescheder, Andreas Geiger, and S. Nowozin. Which training methods for GANs do actually converge? In *ICML*, 2018. 3

[15] 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 *ECCV*. Springer, 2020. 1

[16] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. StyleSDF: High-Resolution 3D-Consistent Image and Geometry Generation. In *CVPR*, 2021. 1

[17] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *AAAI*, volume 32, 2018. 1

[18] Justin NM Pinkney and Doron Adler. Resolution dependent gan interpolation for controllable image synthesis between domains. *arXiv preprint arXiv:2010.05334*, 2020. 5

[19] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *ICLR*, 2016. 2

[20] 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 *CVPR*, 2021. 2

[21] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Trans. Graph.*, 2021. 3, 5

[22] Shunsuke Saito, , Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *ICCV*, 2019. 1

[23] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael Black. Learning to regress 3D face shape and expression from an image without 3D supervision. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 5

[24] Sefik Ilkin Serengil and Alper Ozpinar. Hyperextended lightface: A facial attribute analysis framework. In *2021 International Conference on Engineering and Emerging Technologies (ICEET)*, pages 1–4. IEEE, 2021. 2

[25] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. InterFaceGAN: Interpreting the disentangled face representation learned by GANs. *PAMI*, PP, 2020. 2, 5

[26] 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. 2

[27] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gerard Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[28] Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. High-Fidelity GAN inversion for image attribute editing. In *CVPR*, 2022. 2

[29] Xintao Wang. facexlib. `https://github.com/xinntao/facexlib`, 2020. 2

[30] Xinyao Wang, Liefeng Bo, and Li Fuxin. Adaptive wing loss for robust face alignment via heatmap regression. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 2

[31] Qianyi Wu. EG3D-projector, 2022. 3

[32] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised Learning of Probably Symmetric Deformable 3D Objects from Images in the Wild. In *CVPR*, 2020. 2, 5

[33] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. PixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021. 1, 2

[34] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 3