Supplementary Materials for "*S2FGAN*: Semantically Aware Interactive Sketch-to-Face Translation"

Yan Yang[1,2] Md Zakir Hossain[1,2] Tom Gedeon[3] Shafin Rahman[4]
[1] BDSI, Australian National University, Australia [2]A&F, CSIRO, Australia
[3] EECMS, Curtin University, Australia [4] ECE, North South University, Bangladesh
{Yan.Yang, zakir.hossain}@anu.edu.au Tom.Gedeon@curtin.edu.au shafin.rahman@northsouth.edu

1. Additional Implementation Details

Our *S2FGAN* has four key components: a Encoder (Image Latent Encoder $\mathbf{E}_i$, Sketch Latent Encoder $\mathbf{E}_s$), Attribute Mapping Network $\mathbf{M}$, Style Aware Decoder $\mathbf{D}$ and a Discriminator $\mathbf{F}$. Our sketch-to-image translation with attribute editing relies on encoding the sketch into latent code, and having a superior Style Aware Decoder $\mathbf{D}$ and a Discriminator $\mathbf{F}$ are essential for our work. Thus, we implement our decoder and discriminator backbone based on the recommendation of StyleGAN [6]. Note, we work on the $\mathcal{W}$ space of StyleGAN instead of $\mathcal{W}^+$ space because we target controllable sketch-to-image translation instead of inverting the image to the latent code. There are several minor differences in our decoder and discriminator compared with the original StyleGAN implementation [6]. For the decoder, the PixelNorm, Mapping Network, and Noise Layer are removed. For the discriminator, we remove the minibatch standard deviation layer. And, the same linear layers architectures (in the discriminator) are used to produce attribute classification for S2F-DEC. Here we provide the details of Encoders and Attribute Mapping Networks.

Notions. In Figure 1 and Figure 2, we use the abbreviations. C stands for the convolution 2d layer. CT stands for the transposed convolution 2d layer. CO stands for the convolution 1d layer. S stands for the stride parameter. P stands for the padding parameter. The number following the layer name denotes the output of the layer. LReLU stands for LeakyReLU, where we always use slop 0.2. AdAvg Stands for adaptive average pool. Blur stands for Blur layer, we use the same blur kernel $[1, 3, 3, 1]$ as the StyleGAN [6], and the padding is always set to 1.

Encoder. Our encoder is a $(\lfloor\sqrt{\log(HW)}\rfloor - 2)$ layers ResNet [3] architecture. We calculate the mean of features maps for each downsamples. Finally, the encoder aggregates them by a mean linear and a multi-layer perceptron. It aims to provide the statistic summarization of multi-level facial attributes. For example, pale skin and smiling are likely to desire different size of feature representation. Note, our sketch Latent Encoder $\mathbf{E}_s$ always works on 128×128 resolution because we believe larger resolution reduce the perceptible horizon of convolution layers while the most important property of sketch is facial layouts.

Attribute Mapping Network. The Attribute Mapping Network takes the final output $\mathbf{r_i}$ of the Sketch Latent Encoder $\mathbf{E}_s$ and the attribute shifting vector $\mathbf{a}$ as inputs, and produces output vector $\mathbf{r}'_i$. The Attribute Mapping Network for S2F-DEC and S2F-DIS are presented in Figure 2 (a) and Figure 2 (b), respectively.

Final objectives: The final objectives for generator are defined as follow,
For Lemma 3.1,

$$\begin{aligned}\mathcal{L}_{G_{ortho}} = \lambda_{sem}\mathcal{L}_{sem} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{percept}\mathcal{L}_{percept} \\ + \lambda_{ortho}\mathcal{L}_{ortho} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{domain}\mathcal{L}_{domain}\end{aligned}$$

For Lemma 3.2,

$$\begin{aligned}\mathcal{L}_{G_{decom}} = \lambda_{sem}\mathcal{L}_{sem} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{percept}\mathcal{L}_{percept} \\ + \lambda_{decom}\mathcal{L}_{decom} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{domain}\mathcal{L}_{domain}\end{aligned}$$

The discriminator loss is defined below,

$$\mathcal{L}_D = \lambda_{adv}\mathcal{L}_{adv} + \lambda_{R1}\mathcal{L}_{R1}$$

During training, we set λ_{sem}, λ_{rec}, $\lambda_{percept}$, λ_{ortho}, λ_{decom}, λ_{adv}, λ_{domain} and λ^l_{R1} to 2.5, 1, 2.5, 2^{-1}, 1, 1, 10^{-1} and 1, respectively. For the perceptual Loss, $\mathcal{L}_{percept}$, we use $ReLU1, ReLU2, ReLU3$, $ReLU4$ and $ReLU5$ of VGG19 [9] (which is pretrained on ImageNet) with weights $2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}$ and 1.

2. Additional Qualitative Result

Data Statistic. We present the statistic of attributes in training and testing in Table 1.

Sketch Translation. First, We present the comparison for machine extracted sketch with Pix2PixHD [10] in Figure 3. Here, input sketches used in training and testing come from

Table 1. Attribute statistic of CelebAMask-HQ. The value indicates the number of times each attribute appeared in faces.

Fold \ Attribute	Smiling	Male	No_Beard	Eyeglasses	Young	Bangs	Narrow_Eyes	Pale_Skin	Big_Lips	Big_Nose	Mustache	Chubby
Training	13446	10512	23084	1404	22194	5141	3350	1441	10367	9280	1661	1999
Testing	646	545	1244	64	1174	284	166	92	523	454	74	103

Algorithm 1: Refine badly drawn sketches.

Input: I_i, K, TT // I_i is a human drawn sketch.
Output: I_{out} // I_{out} is a refined sketch.
Data: Training data set $\mathbf{X}$ // X is set of photo-realistic images.
$\mathbf{r}_i = \mathbf{E}_s(I_i)$ // Encode input sketch to latent code.
TR = [] // Initialize a empty container.
for $\mathbf{x} \in \mathbf{X}$ **do**
  TR.append($\mathbf{E}_i(\mathbf{x})$)
end
$\mathbf{r}_i = Mean(nearest\_neighbors(\mathbf{r}_i, TR, K)) * (1 - TT) + TT * \mathbf{r}_i$ // First, get the K-nearest neighbors of $\mathbf{r}_i$ in latent space. Second, calculate the mean of the nearest neighbors. Third, apply truncation tricks.
$I_{out} = \mathbf{D}(\mathbf{r}_i)$ // Generate refined images.

the same sketch extraction method, which produces high-quality images using Pix2PixHD, S2F-NDIS, S2F-DEC, and S2F-DIS. Second, We present the example of translating all of the human drawn sketch provided by [11] in Figure 8, Figure 9 and Figure 10. We combine the k-nearest neighbors algorithm and truncation trick [1] to refine the badly drawn sketches. The details are provided in Algorithm 1. We visualize the impact of truncation ratio and nearest neighbors for S2F-DEC and S2F-DIS in Figure 6 and Figure 7, respectively. Besides, we also considering use the style-transfer to refine the badly drawn sketches. Our Image Latent Encoder $\mathbf{E}_i$ is capable of encoding the styles from photo-realistic facial images. Similar with [5], we define copying styles from resolution $4^2 - 8^2$, $8^2 - 32^2$ and $32^2 - 256^2$ as high-level refinement, medium-level refinement and low-level refinement. Examples are presented in Figure 4 and Figure 5.

Attribute Editing. We present more examples of multi-attribute editing in Figure 11 and Figure 12, while the examples of single attribute editing are in Figure 13, Figure 14, Figure 15 and Figure 16. We note the STGAN [8], AttGAN[4], S2F-NDIS and S2F-DEC have poor "beard" editing performance. They edit the attributes by following the training data distributions, while it is unlikely for a female to have a beard, which is one of their drawbacks. However, for S2F-DIS, the semantic vectors for attributes of interests are orthogonal with each other, and hence lead to strong editing behaviours (See "Female" and "beard" editing case in Figure 12).

3. Additional Materials

Quantitative Evaluation. Because of the absence of ground truth photorealistic images for the badly drawn sketches, we perform a user study to better understand the performance of the state of art methods. There are 25 participants, where 60% of them are from non-computer science backgrounds. Similar with [11], they are asked to selecting the best translation results, which balances the sketch faithfulness with the output verisimilitude, among different the synthesis from different methods. All the sketches (which are from [11]) and translation results are presented in Figure 8, 9 and 10. To ensure the fairness of user study, we randomize the order of presenting sketch and translation pairs, while we also randomly present the order of translation results without leaking any model information of them. There are 6000 selections in total. We calculate the user preference ratio by counting the fraction of the translation results being selected, which follows [11].

The user preference ratio is presented in Table 2. S2F-DEC, S2F-DIS, and DPS [11] have significantly better scores than the rest of the methods. Our S2F-DEC has the best average user preference. The S2F-DIS has a similar result with the second-best average performed method DPS [11]. Among the 30 sketches, S2F-DEC and S2F-DIS earn the highest score for 13 individuals and 9 individuals, respectively, which are at least better than DPS [11] (that has the highest scores for 9 individuals). The user study proves the superiority of our approach in translating badly drawn human sketches. Moreover, as 60% of the participants are from a non-computer science background, it also demonstrates the potential of deploying our approach in a real-world application.

References

[1] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

ID	Pix2PixHD	DFD-0	DFD-1	DFP	DPS	S2F-NDIS	S2F-DEC	S2F-DIS
0	0.12	0.08	0.04	0.0	**0.36**	0.2	0.04	0.16
1	0.04	0.04	0.04	0.0	0.16	0.08	**0.4**	0.24
2	0.16	0.0	0.2	0.0	0.16	0.0	**0.28**	0.2
3	0.08	0.04	0.04	0.0	0.2	0.24	**0.36**	0.04
4	0.08	0.04	0.16	0.0	**0.32**	0.08	**0.32**	0.0
5	0.0	0.12	0.32	0.0	**0.44**	0.0	0.04	0.08
6	0.0	0.08	0.08	0.08	0.12	**0.36**	0.04	0.24
7	0.08	0.04	0.04	0.0	**0.32**	0.04	0.28	0.2
8	0.04	0.04	0.0	0.04	0.24	0.0	**0.32**	**0.32**
9	0.16	0.04	0.04	0.0	0.16	0.04	**0.52**	0.04
10	0.08	0.0	0.16	0.0	**0.32**	0.0	0.2	0.24
11	0.04	0.12	0.08	0.0	0.16	0.16	0.16	**0.28**
12	0.08	0.0	0.08	0.04	0.24	0.08	0.16	**0.32**
13	0.04	0.08	0.12	0.0	0.2	0.04	0.08	**0.44**
14	0.08	0.08	0.0	0.0	0.08	0.08	**0.6**	0.08
15	0.16	0.04	0.08	0.04	0.12	0.0	**0.52**	0.04
16	0.16	0.04	0.04	0.0	0.08	0.12	**0.36**	0.2
17	0.08	0.0	0.04	0.0	0.16	0.12	**0.52**	0.08
18	0.16	0.08	0.04	0.0	0.2	0.08	0.16	**0.28**
19	0.04	0.12	0.08	0.0	**0.32**	0.2	0.04	0.2
20	0.04	0.0	0.08	0.0	0.04	0.0	**0.48**	0.36
21	0.04	0.08	0.16	0.0	0.16	0.04	0.24	**0.28**
22	0.04	0.08	0.0	0.0	0.28	0.08	**0.32**	0.2
23	0.0	0.08	0.12	0.0	**0.32**	0.04	0.24	0.2
24	0.16	0.0	0.0	0.0	0.08	0.0	**0.4**	0.36
25	0.12	0.04	0.2	0.0	0.24	0.0	0.04	**0.36**
26	0.04	0.0	0.08	0.04	0.2	0.12	0.08	**0.44**
27	0.24	0.12	0.0	0.0	**0.28**	0.04	0.2	0.12
28	0.08	0.04	0.0	0.0	0.2	0.04	0.24	**0.4**
29	0.04	0.04	0.32	0.0	**0.4**	0.16	0.04	0.0
Average	0.083	0.052	0.088	0.008	0.219	0.081	**0.256**	0.213

Table 2. Quantitative evaluation of translating badly drawn sketches. We present the comparison of normalized user preference scores for the state of art methods.

[2] Shu-Yu Chen, Wanchao Su, Lin Gao, Shihong Xia, and Hongbo Fu. Deepfacedrawing: deep generation of face images from sketches. *ACM Trans. Graph.*, 39(4):72, 2020.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[4] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE Trans. Image Process.*, 28(11):5464–5478, 2019.

[5] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 4401–4410. Computer Vision Foundation / IEEE, 2019.

[6] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 8107–8116. IEEE, 2020.

[7] Yuhang Li, Xuejin Chen, Binxin Yang, Zihan Chen, Zhihua Cheng, and Zheng-Jun Zha. Deepfacepencil: Creating face images from freehand sketches. In Chang Wen Chen, Rita Cucchiara, Xian-Sheng Hua, Guo-Jun Qi, Elisa Ricci, Zhengyou Zhang, and Roger Zimmermann, editors, *MM '20: The 28th ACM International Conference on Multimedia, Virtual Event / Seattle, WA, USA, October 12-16, 2020*, pages 991–999. ACM, 2020.

[8] Ming Liu, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo, and Shilei Wen. STGAN: A unified selective transfer network for arbitrary image attribute editing. In *IEEE Conference on Computer Vision and Pattern*

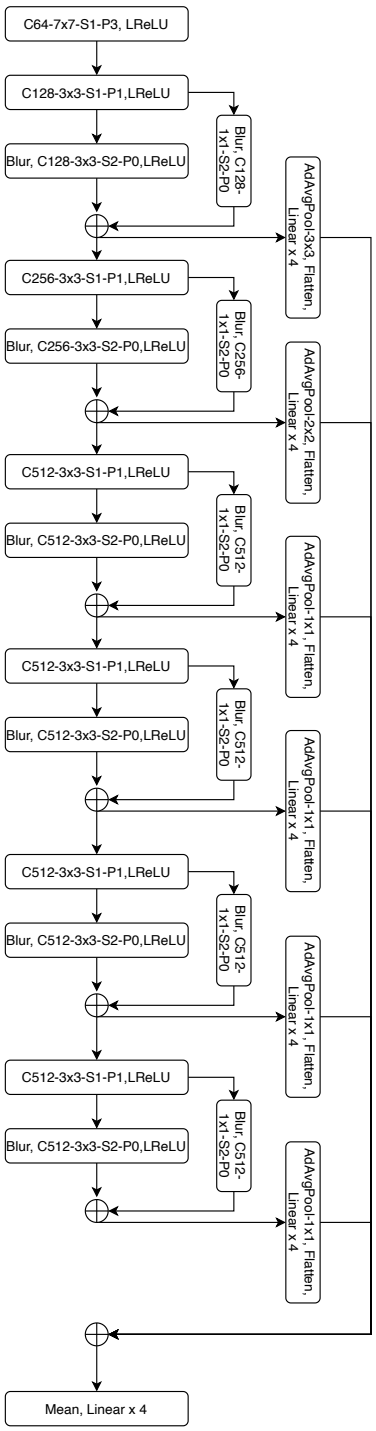

Figure 1. Architecture of Image Latent Encoder $\mathbf{E}_i$

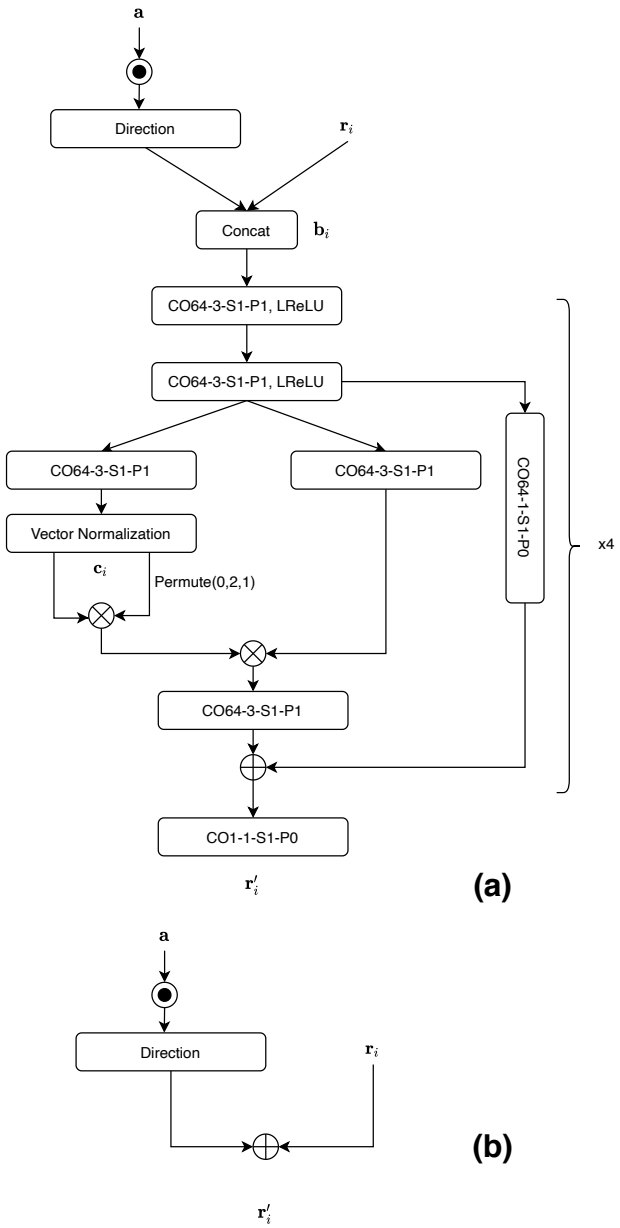

Figure 2. Architecture of Attribute Mapping Network for S2F-DEC **(a)** and S2F-DIS **(b)**.

Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pages 3673–3682. Computer Vision Foundation / IEEE, 2019.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[10] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8798–8807. IEEE Computer Society, 2018.

[11] Shuai Yang, Zhangyang Wang, Jiaying Liu, and Zongming Guo. Deep plastic surgery: Robust and controllable image

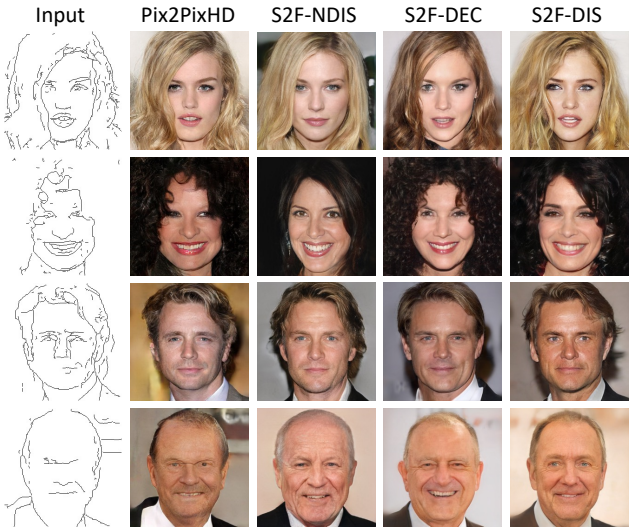

Figure 3. Comparison of sketch-to-image translation.

editing with human-drawn sketches. *CoRR*, abs/2001.02890, 2020.

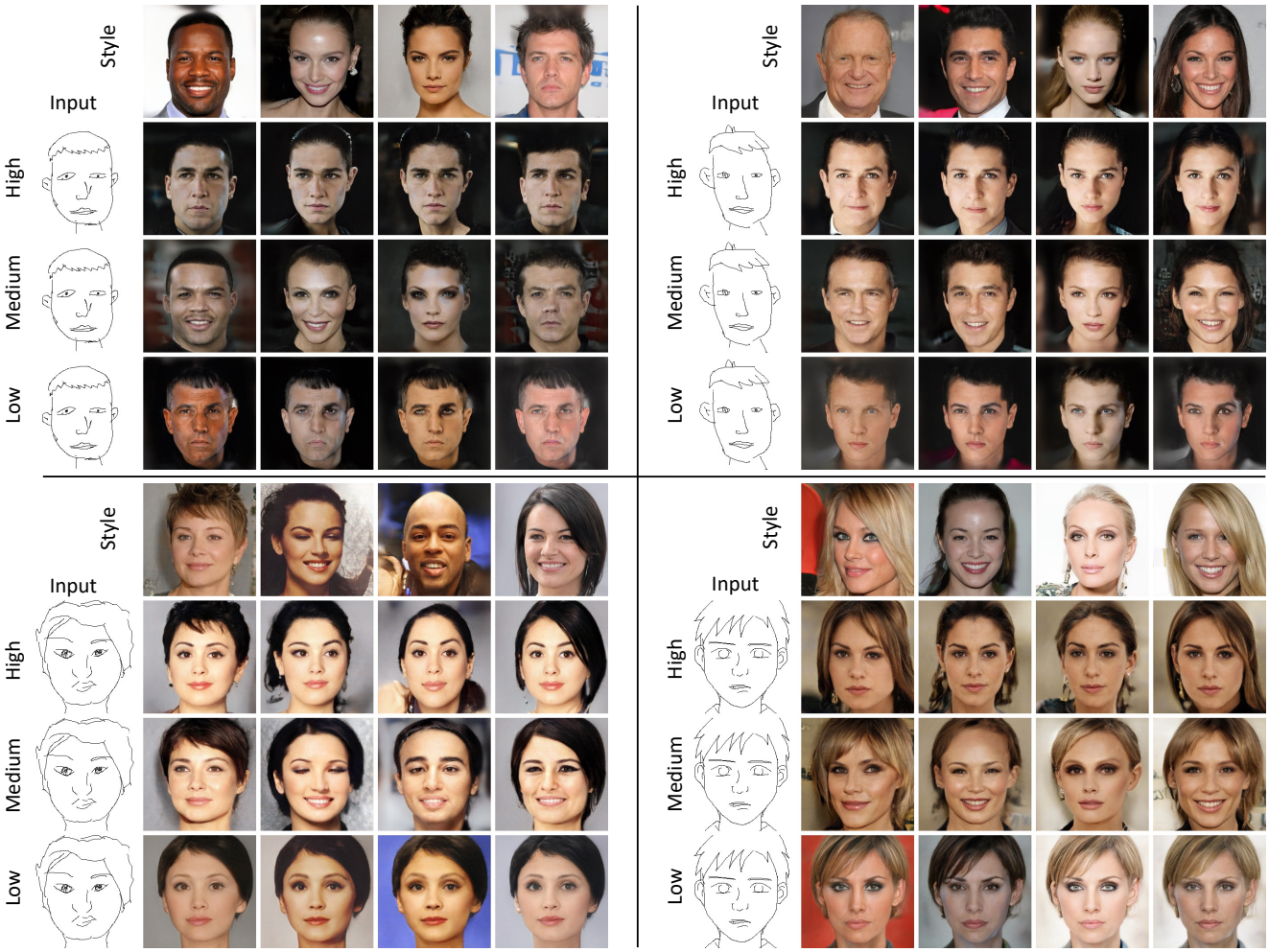

Figure 4. Style transfer for S2F-DEC. In high-level refinement, the face shape is borrowed from the reference image. In medium level refinement, the high style, facial layouts, and pose are inherited. In low-level refinement, the color schema of the reference image is preserved. Note, we do not use Algorithm 1 to refine the input sketches here.

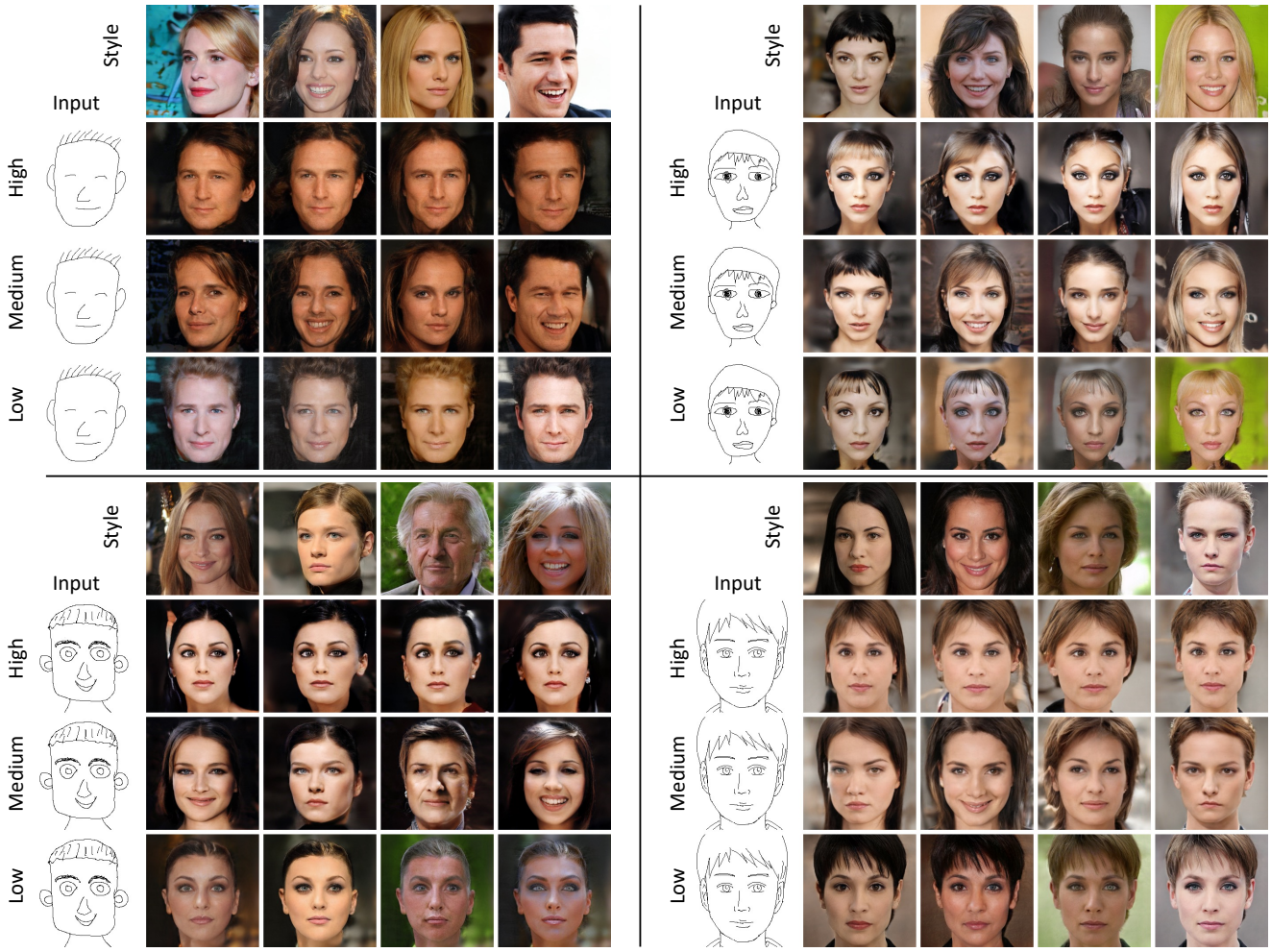

Figure 5. Style transfer for S2F-DIS. In high-level refinement, the face shape is borrowed from the reference image. In medium level refinement, the high style, facial layouts, and pose are inherited. In low-level refinement, the color schema of the reference image is preserved. Note, we do not use Algorithm 1 to refine the input sketches here.

Figure 6. The impact of truncation ratio (TT) and nearest neighbors (K) for S2F-DEC.

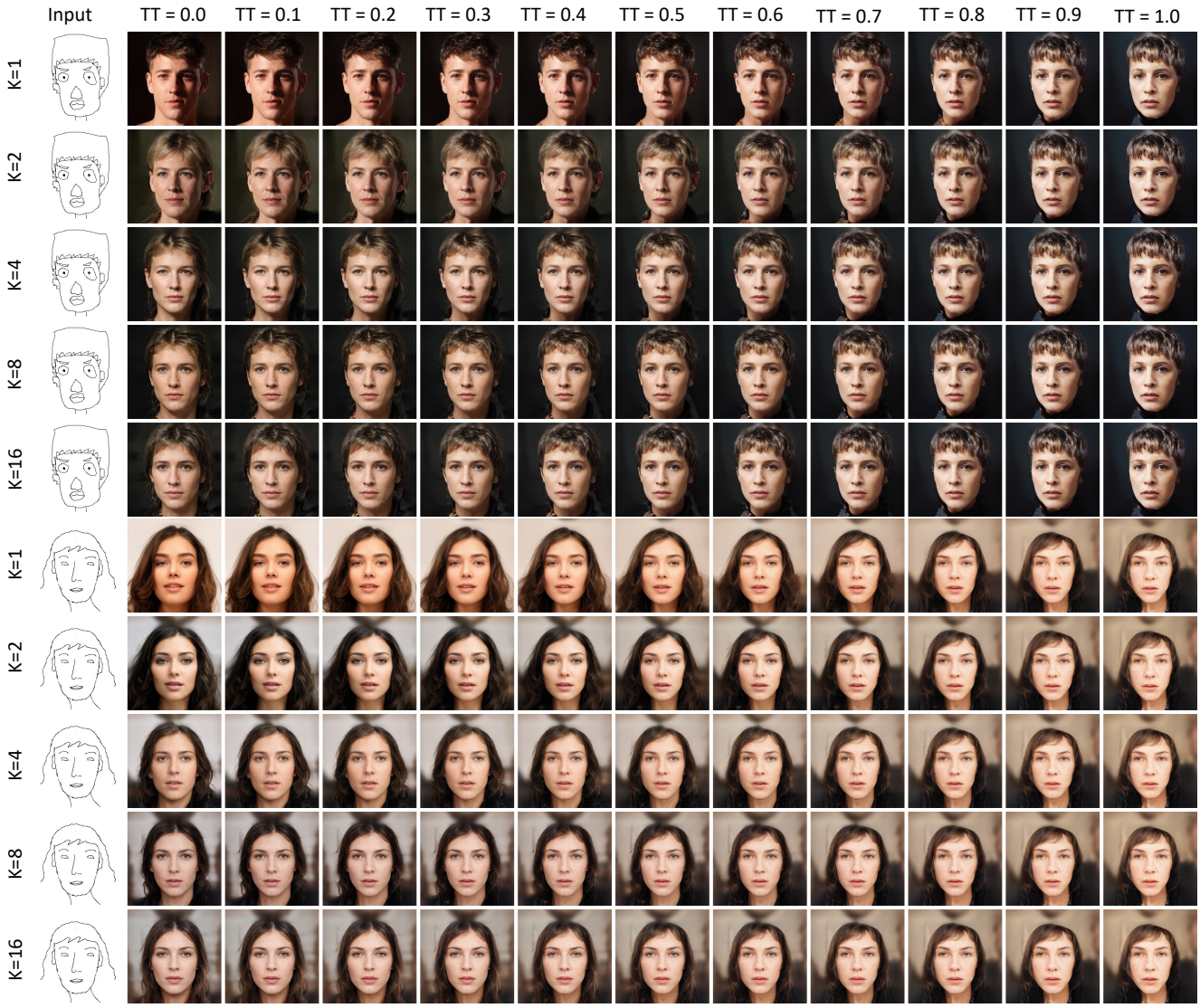

Figure 7. The impact of truncation ratio (TT) and nearest neighbors (K) for S2F-DIS.

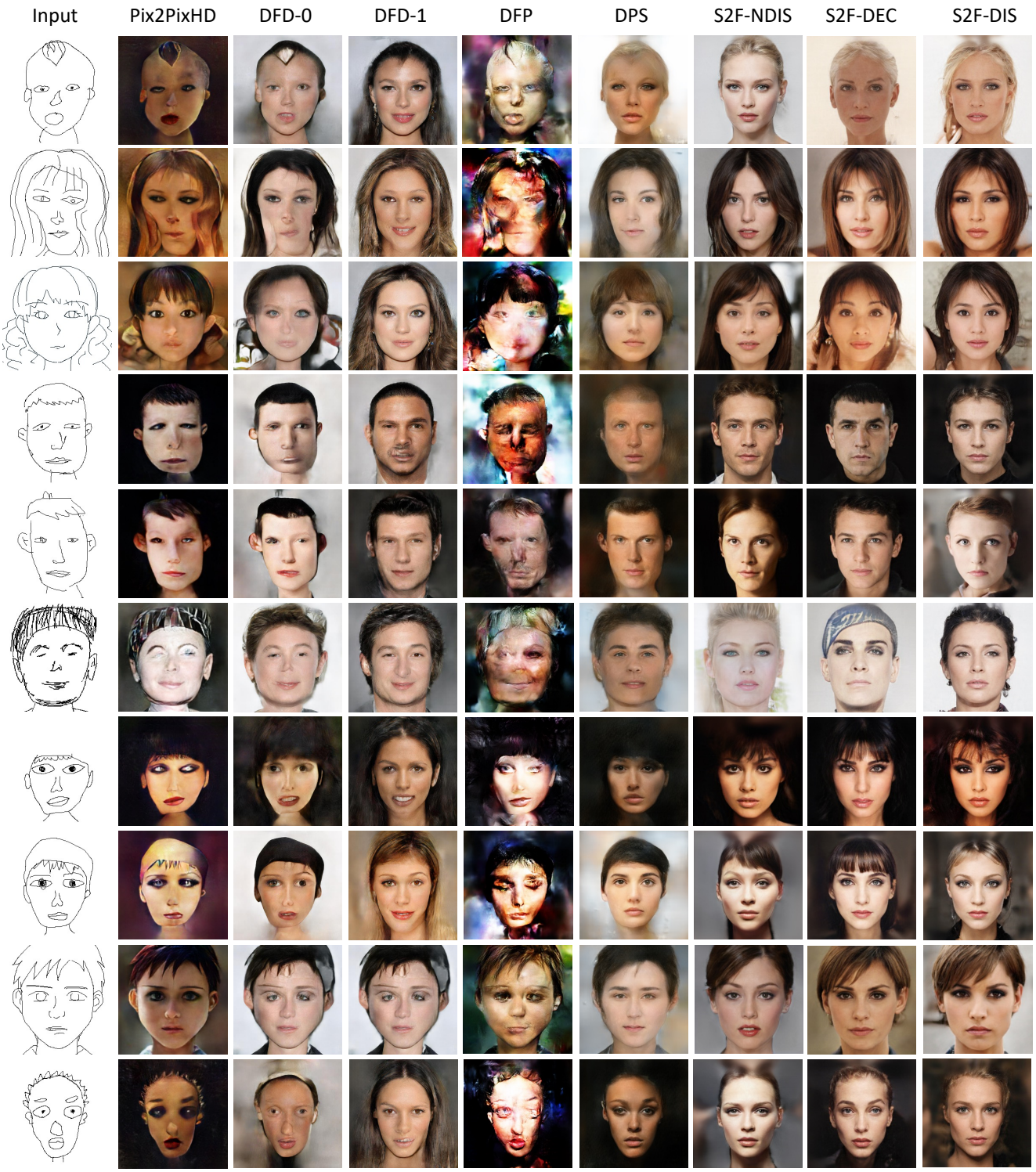

Figure 8. Comparison of translating human drawn sketches (provided by [11]) with Pix2PixHD [10], DeepFaceDrawing (DFD) [2], DeepFacePencil (DFP) [7] and Deep Plastic Surgery (DPS) [11]. We use DFD-0 and DFD-1 to represent the medium refinement and fully refinement of DFD.

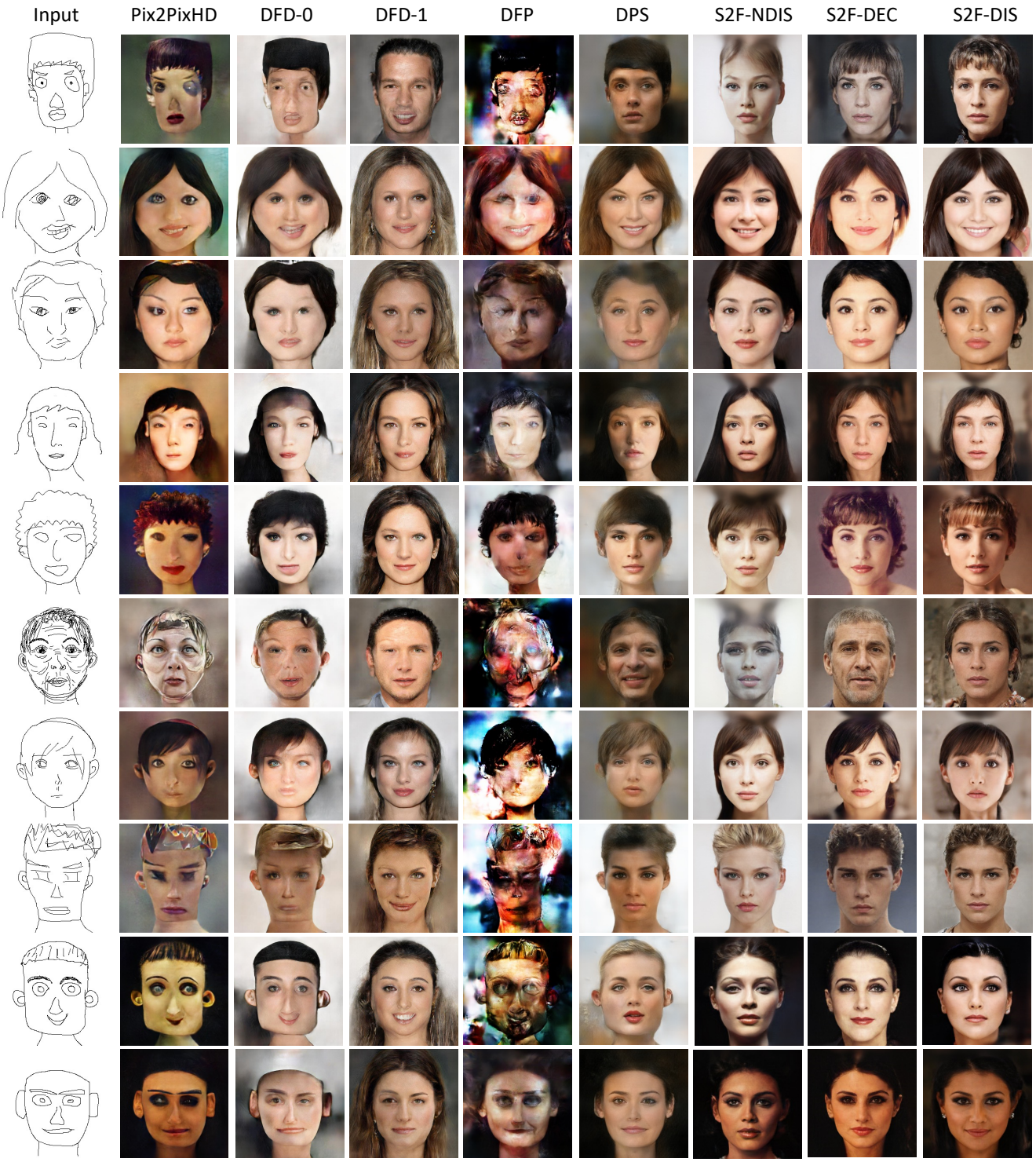

Figure 9. Comparison of translating human drawn sketches (provided by [11]) with Pix2PixHD [10], DeepFaceDrawing (DFD) [2], DeepFacePencil (DFP) [7] and Deep Plastic Surgery (DPS) [11]. We use DFD-0 and DFD-1 to represent the medium refinement and fully refinement of DFD..

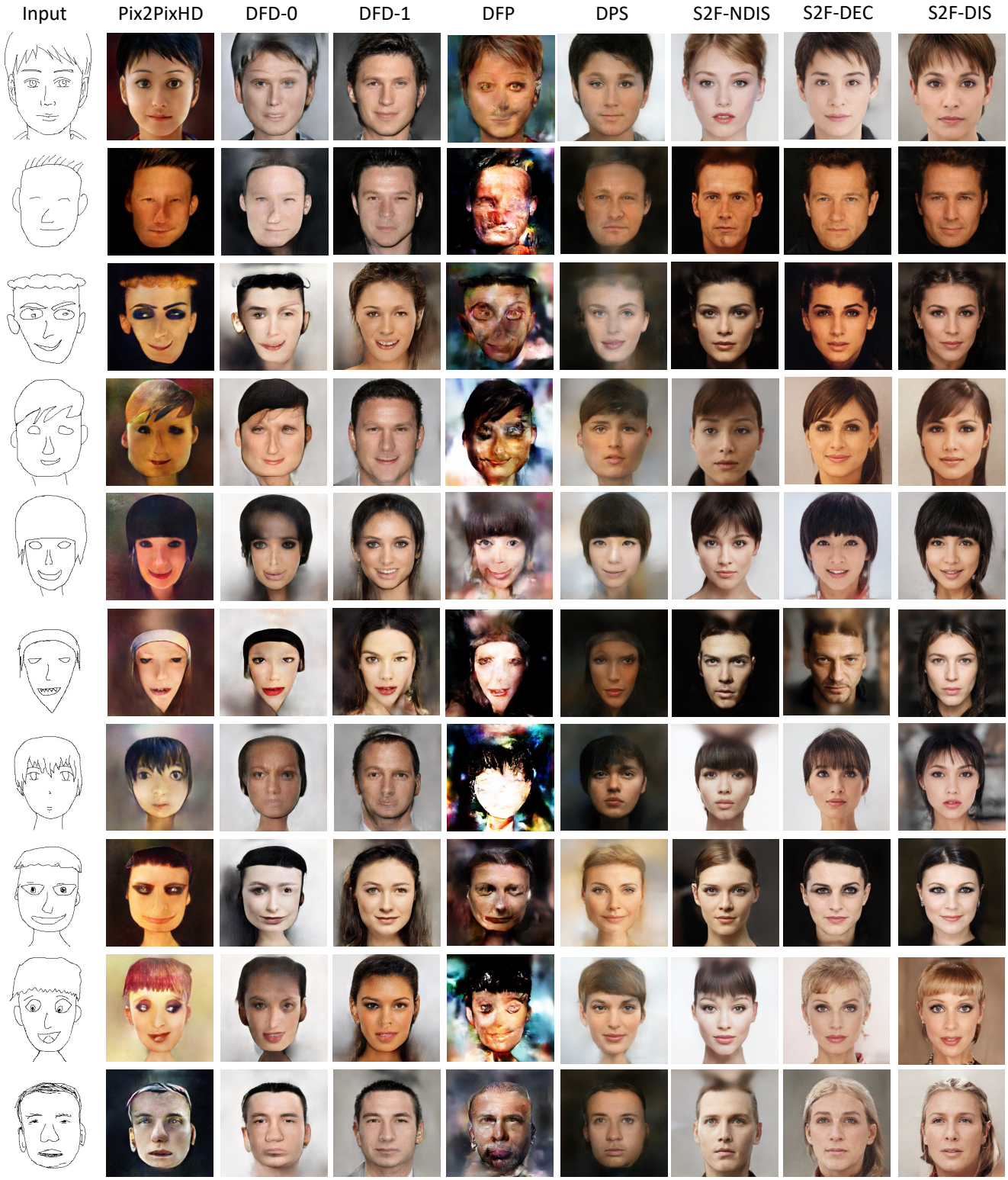

Figure 10. Comparison of translating human drawn sketches (provided by [11]) with Pix2PixHD [10], DeepFaceDrawing (DFD) [2], DeepFacePencil (DFP) [7] and Deep Plastic Surgery (DPS) [11]. We use DFD-0 and DFD-1 to represent the medium refinement and fully refinement of DFD.

Figure 11. Comparison of multi-attribute editing with STGAN [8], AttGAN [4] and S2F-NDIS.

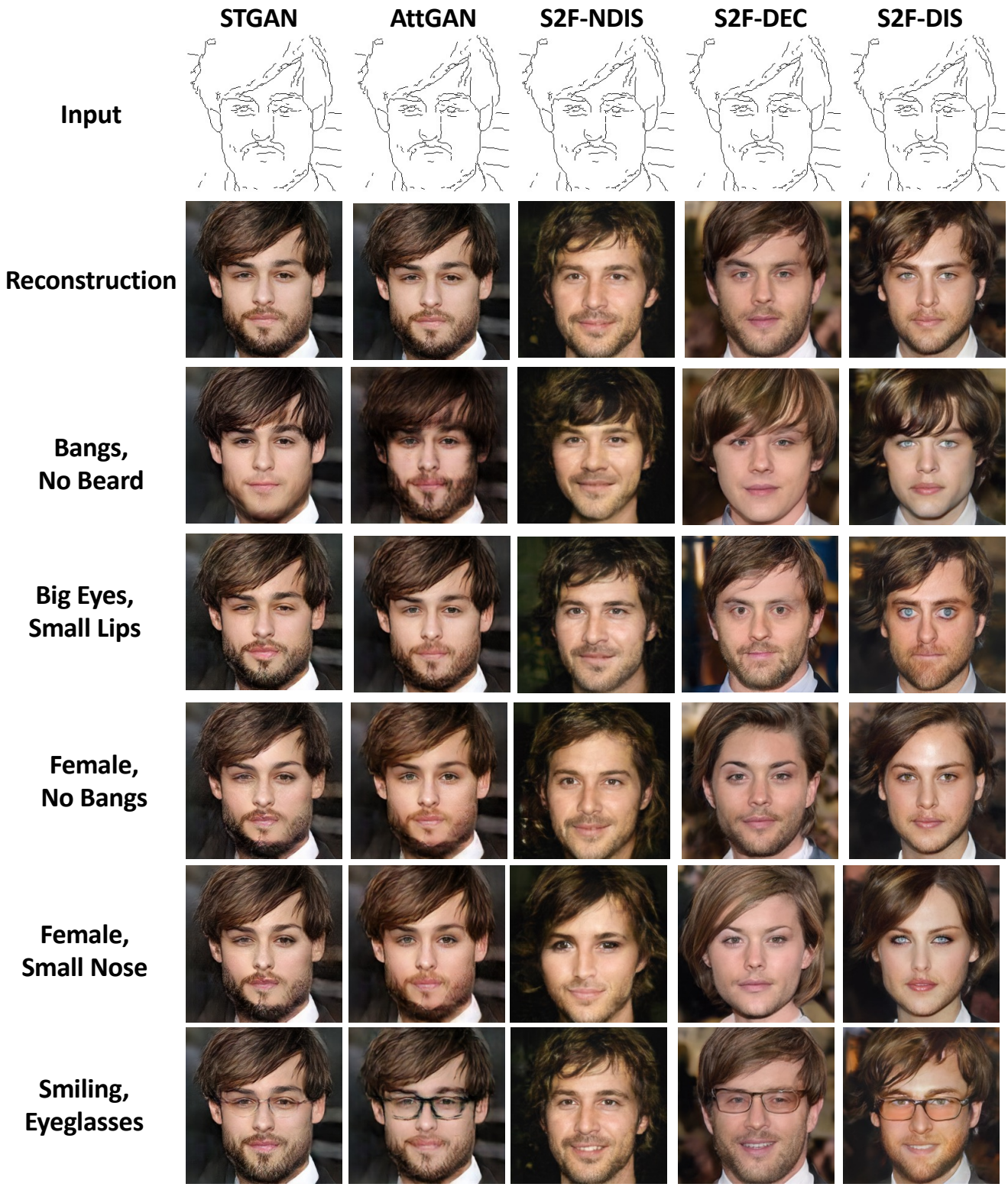

Figure 12. Comparison of multi-attribute editing with STGAN [8], AttGAN [4] and S2F-NDIS.

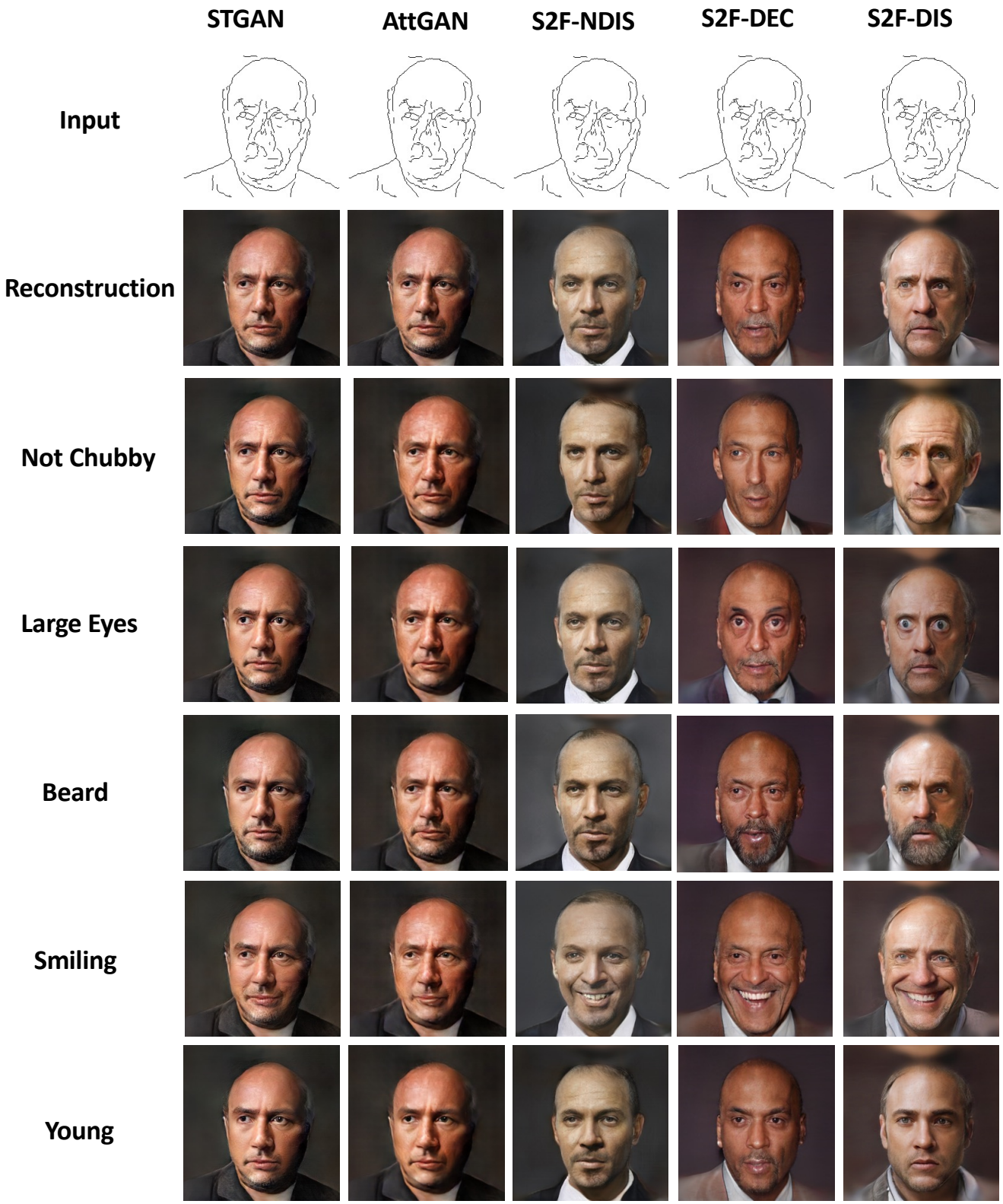

Figure 13. Comparison of single-attribute editing with STGAN [8], AttGAN [4] and S2F-NDIS.

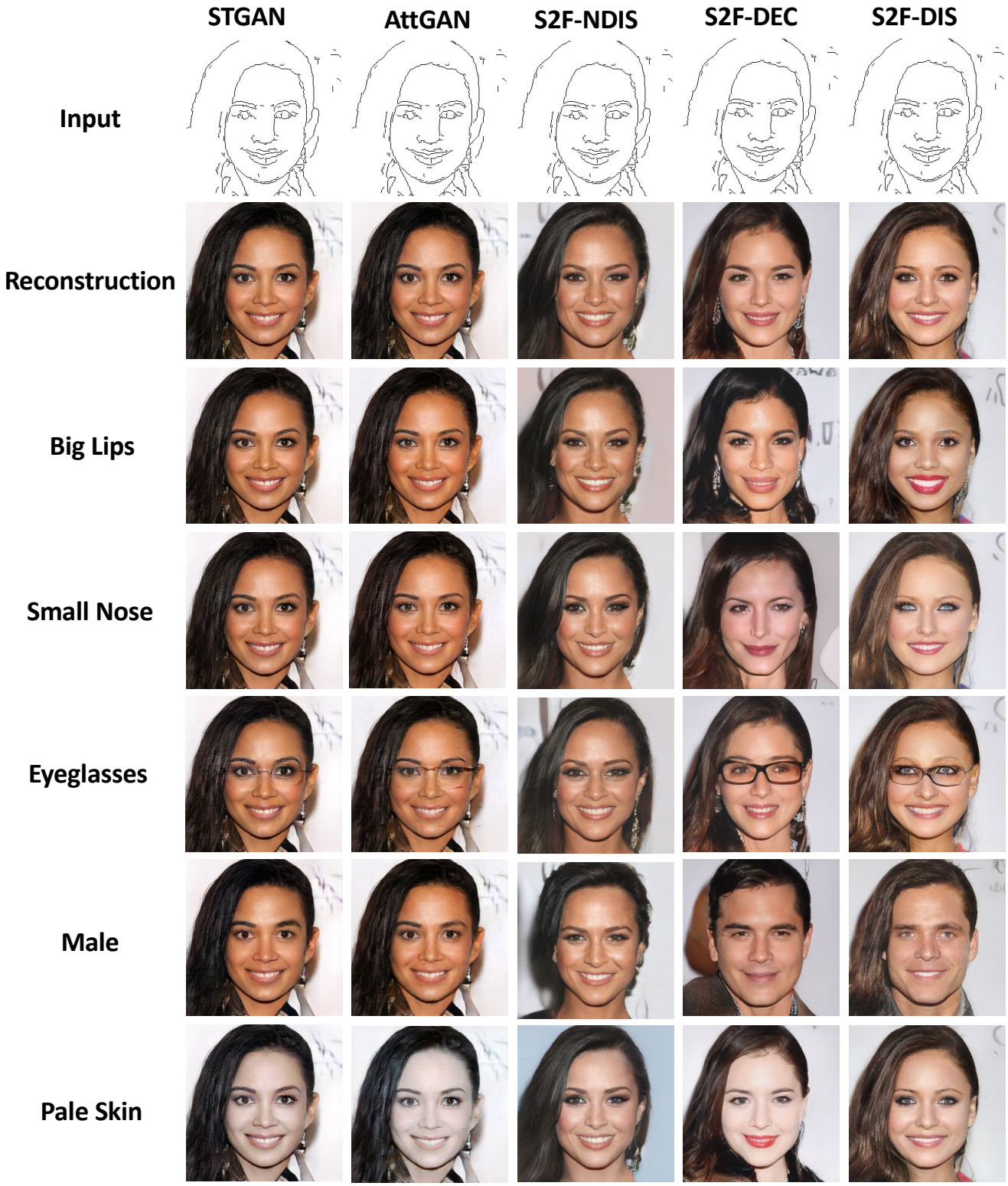

Figure 14. Comparison of single-attribute editing with STGAN [8], AttGAN [4] and S2F-NDIS.

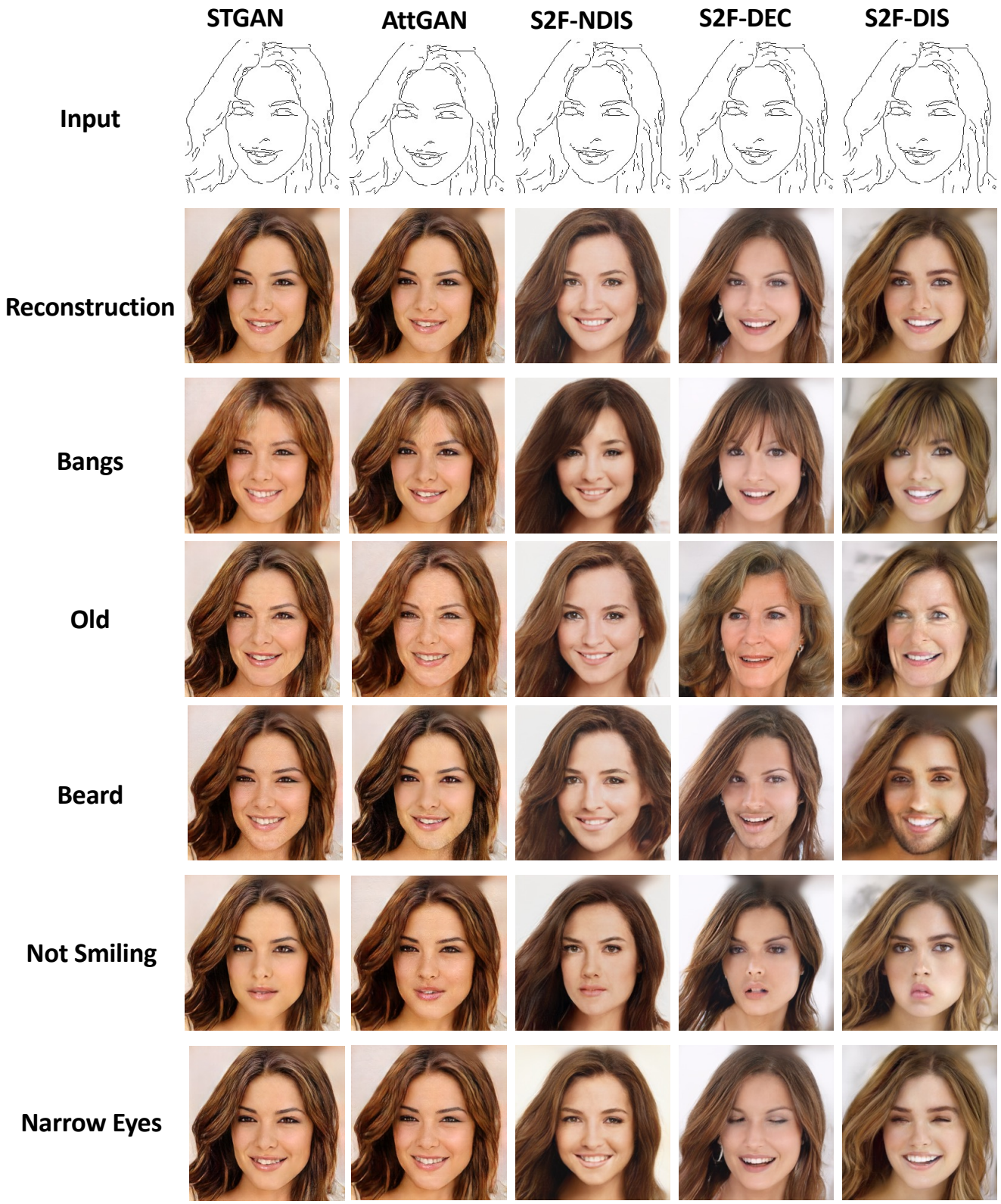

Figure 15. Comparison of single-attribute editing with STGAN [8], AttGAN [4] and S2F-NDIS.

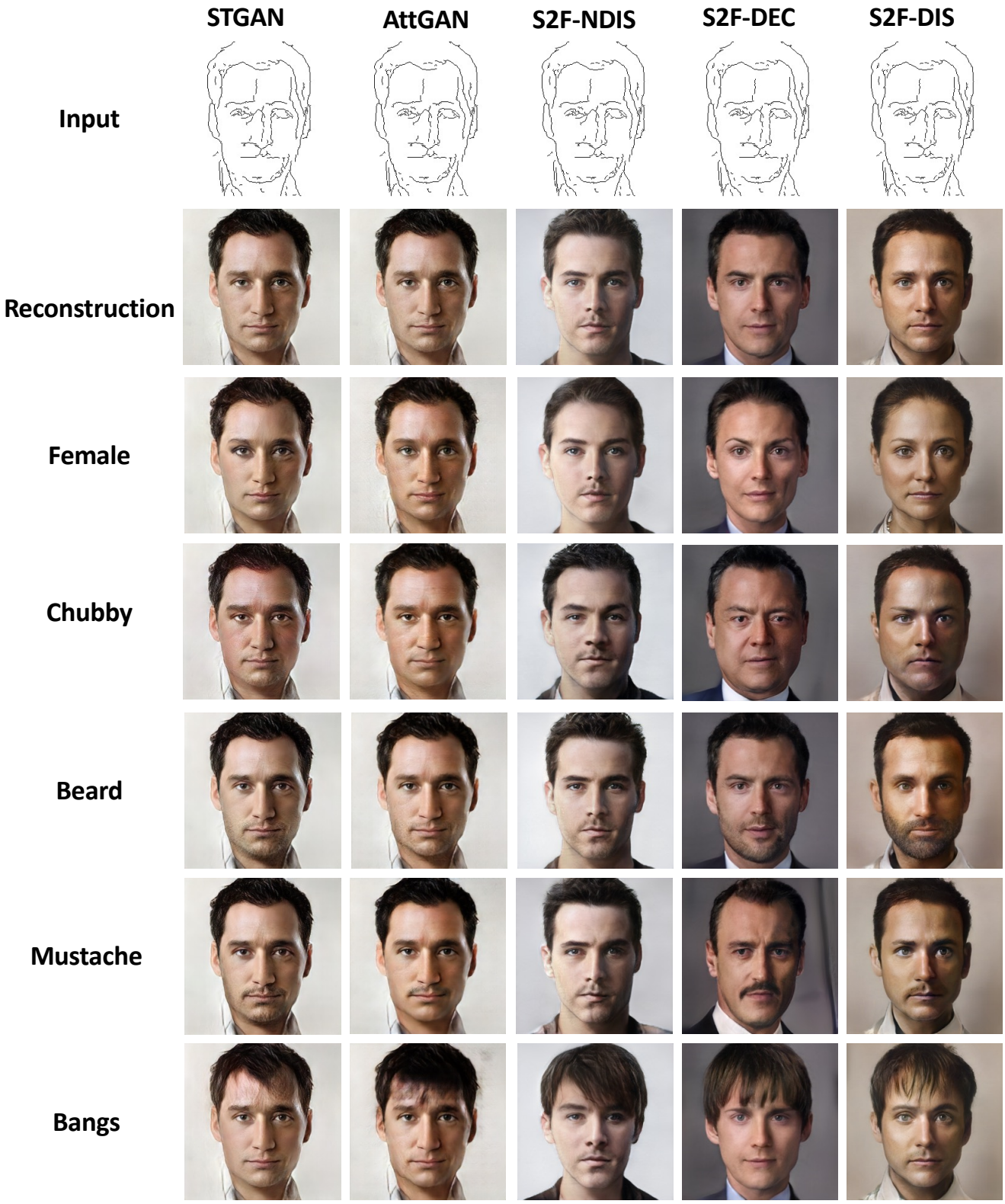

Figure 16. Comparison of single-attribute editing with STGAN [8], AttGAN [4] and S2F-NDIS.