A. Training details

We set the batch size to 32 and train the model from scratch on eight V100 GPUs for 400 epochs. We implement the model in TensorFlow 2.4 [1]. We use the Adam optimizer [17] with $(lr = 2.5e^{-3}, \beta_1 = 0.0, \beta = 0.99)$ for both the discriminator and the generator.

Perceptual Loss In Eq. 8, the perceptual loss contains two parts: feature reconstruction loss $L_{feature}$ and style reconstruction loss L_{style}. Input an image x, assume i is a convolution layer then $\phi(x)$ will be a feature map of size $C_i \times H_i \times W_i$

$$L_{feature} = \sum_i \frac{1}{C_i H_i W_i} ||\phi_i(x) - \phi_i(y)||_2^2 \tag{12}$$

$$L_{style} = \sum_i ||Gram_i(x) - Gram_i(y)||_F^2 \tag{13}$$

where $\phi(\cdot)$ is a VGG feature extractor and $Gram_i(\cdot)$ is a Gram matrix at layer i whose elements at index (c, c') are given by

$$Gram_i(x) = \frac{1}{C_i H_i W_i} \sum_{h=1}^{H_i} \sum_{w=1}^{W_i} \phi(x)_{h,w,c} \phi(x)_{h,w,c'} \tag{14}$$

A.1. Network architecture

In this section, we provide architectures of the applied model.

Encoder The encoder consists of stacked residual blocks and each residual block (ResBlock) consists of two convolution layer, where the first layer does not change the spatial size whereas the second one comes with a stride 2 for down-sampling. The kernel size is 3×3. We use the non-parameterized AttentionPooling as the last layer to aggregate global spatial information.

Layer	Output Shape
Image x	$512 \times 512 \times 3$
ResBlock	$256 \times 256 \times 32$
ResBlock	$128 \times 128 \times 64$
ResBlock	$64 \times 64 \times 128$
ResBlock	$32 \times 32 \times 256$
ResBlock	$16 \times 16 \times 512$
ResBlock	$8 \times 8 \times 512$
ResBlock	$4 \times 4 \times 512$
AttentionPooling	$1 \times 1 \times 512$

Table 4. Encoder architecture.

Decoder The decoder includes several StyleBlock, which is borrowed from the StyleGAN generator [15]. Each StyleBlock takes two inputs: previous feature map and external modulation vector.

Layer	Output Shape
Last Feature	$4 \times 4 \times 512$
StyleBlock	$8 \times 8 \times 512$
StyleBlock	$16 \times 16 \times 512$
StyleBlock	$32 \times 32 \times 512$
StyleBlock	$64 \times 64 \times 256$
StyleBlock	$128 \times 128 \times 128$
StyleBlock	$256 \times 256 \times 64$
StyleBlock	$512 \times 512 \times 32$
ToRGB	$512 \times 512 \times 3$

Table 5. Decoder architecture.

Discriminator The architecture of discriminator is mostly the same as the encoder, except that the AttentionPooling is replaced by a minibatch discrimination layer [15].

Algorithm 1 Precision and iPrecision

```
# G: generated images set. R: ground truth set.
# k: neighborhood size.
# net: pretrained feature extractor.
import numpy as np
# compute features for fake and real images.
N = len(G) # or N = len(R)
E_g = np.stack([net(g) for g in G]) # fake: Nxd
E_r = np.stack([net(r) for r in R]) # real: Nxd

# compute neighbors' distance and identity.
# we also store the info of data itself.
gn_dist, gn_id = neighbor(E_g) # Nx(k+1), Nx(k+1)
rn_dist, rn_id = neighbor(E_r) # Nx(k+1), Nx(k+1)
precision, iprecision = [], []
for e_g in E_g:
    dist = euclidean_distance(e_g, E_r) # N
    # check whether e_g in any neighborhood
    eg_in = dist[:, :, None] <= rn_dist # Nxk
    # check id(e_g) is equal to any id(e_r).
    eg_id_eq = (id(e_g) == rn_id[:, 0]) # N
    # check both condition are met.
    eg_both = np.logical_and(eg_in, eg_id_eq)
    pred = np.any(eg_dist_in, axis=0) # k
    ipred = np.any(eg_both, axis=0) # 1
    precision.append(pred)
    iprecision.append(ipred)

# Average over all fake data.
precision = np.stack(pred).mean(axis=0)
iprecision = np.stack(ipred).mean(axis=0)
```

B. iPrecision and iRecall

Figure 8 evaluates the metric with Inception V3. It is shown that our approach consistently outperforms the baselines as in FaceNet. Besides, compared with Figure 5,

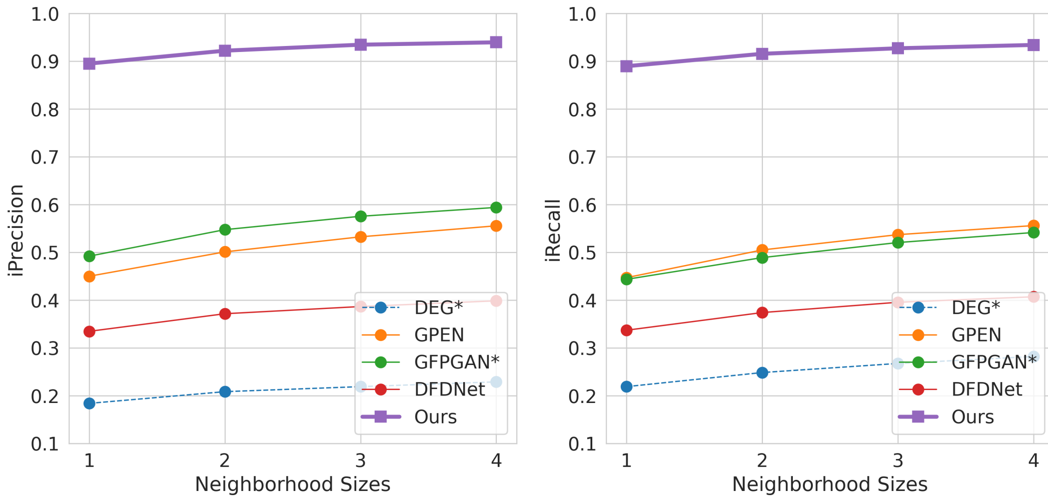

Figure 8. iPrecision and iRecall with Inception V3

FaceNet can provide more discriminative quantitative numbers.

B.1. Pseudo-code for precision

We provide the pseudoc-code for computing precidion in Algorithm 1. Recall can be computed in a similar way.

C. More experimental results

C.1. Ablations

C.1.1 The number of skip connections

The other critical factor that affects the restoration is the number of skip connections. Table 6 quantifies the restoration performances. In this paper, we use 4 skip connections at resolution nodes $(8^2, 16^2, 32^2, 64^2)$ by default. As is seen, more skip connections usually lead to better results, except for using as many as six.

NO.	PSNR↑	LPIPS↓	SSIM↑	FID↓	iPrecision ↑
0	21.16±0.45	0.3358	0.5754	24.30	0.321
1	24.75±0.12	0.3098	0.6668	20.49	0.902
2	26.15±0.04	0.2543	0.6915	**19.17**	0.945
4	**27.43**±0.03	**0.2349**	**0.7316**	19.19	**0.982**
6	27.07±0.04	0.3112	0.6707	27.17	0.931

Table 6. On the impact of the number of skip connections.

C.1.2 The impact of noises

We also evaluate how different noises affect the restoration results given the same input. It is observed that: (i) The influence of noises diminishes with more skip connections as seen in Table 6. (ii) Less number of skip connections can generate more diverse images at the cost of sacrificing face identities, as seen in Table 6 and Figure 11. (iii) Stochastic generation doesn't lead to instability issues as seen in Table 6.

C.1.3 Adversarial data augmentation

Table 7 compares the effect of adversarial data augmentation.

Adv. Aug.	PSNR↑	SSIM↑	LPIPS↓	FID↓
N	26.48	0.7021	0.2574	20.22
Y	26.89	0.7134	0.2452	19.77

Table 7. On the impact of adversarial data augmentation.

C.1.4 On the impact of α

Except for the aforementioned model design that is critical to balance reconstruction and generation, the relative weight α is obviously crucial. Overall, we find that increasing α causes very opposite results in terms of PSNR and FID. This happens because $\mathcal{L}_{\text{ADV}}$ and $\mathcal{L}_{\text{REC}}$ optimize the generator towards different directions. Larger α helps FID but harms PSNR. In contrast, smaller α can improve PSNR but generates blurry samples. In this work, we simply use $\alpha = 1.0$ by default.

Methods	PSNR↑	LPIPS↓	iPrecision↑	Preference (%)↑
DFDNet	23.68	0.434	0.462	3.2
GFPGAN	24.19	0.296	0.711	5.3
GPEN	23.91	0.331	0.773	15.1
Ours	**28.01**	**0.205**	**0.943**	**76.41**

Table 8. Metric comparison on BFR.

C.1.5 Discussion on failure cases

Figure 9 shows two failed restoration cases. The unrealistic artifacts usually appear in face key points, *e.g.,* eyes and teeth. It in turn suggests that optimization on these regions could be an interesting direction to explore.

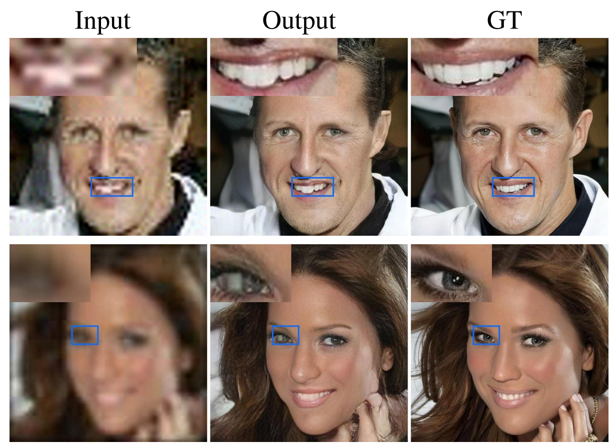

Figure 9. Failure restoration.

C.1.6 More qualitative results

Figure 13 compares the restoration results with different approaches on real low-quality images. Figure 14-Figure 16 show more qualitative results on BFR, $\times 8$ and $\times 16$.

C.2. Human evaluation

Talbe 8 shows the human evaluation results on BFR task. Similar to Table 2, we can also observe that our proposed metric is a better indicator for face restoration.

In terms of detailed human study, we randomly select 100 samples from the testing images and distribute them to 5 experts that have been devoted to the camera software development for years. In each example, we place input degraded image, ground truth and four restored images from different approaches, as shown in Figure 12. The four restored images are places in random order for each example. People were asked to select the best restored face image following standards:

- It shows less color shift, *e.g.*, the eyeball, hair color and skin tone should be consistent with ground truth.

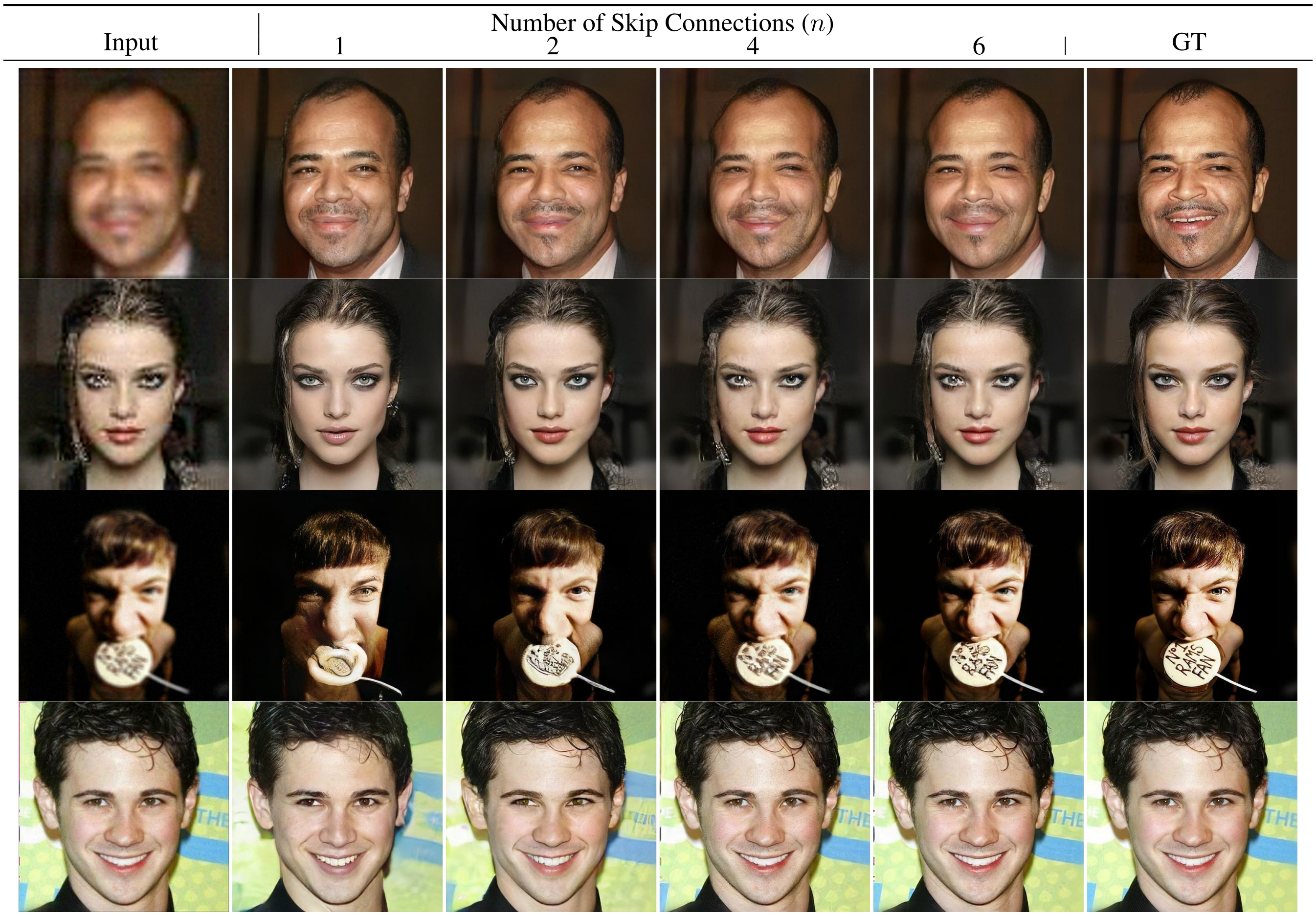

Figure 10. Qualitative comparison by varying the number of skip connections. We count from the layer with feature resolution 8×8, *i.e.,* there exist possible skip connections at resolution nodes $\{2^{n+2} \times 2^{n+2}\}_{n=1}^{6}$ when we set the maximum input resolution at 512×512.

- It has sharp and defined features
- It looks realistic and shows no or less artifacts.
- No excessive features are observed, *e.g.*, the facial features shouldn't be too bright or crispy to look realistic, the appearance of eyelid and eyelash should be consistent, etc.

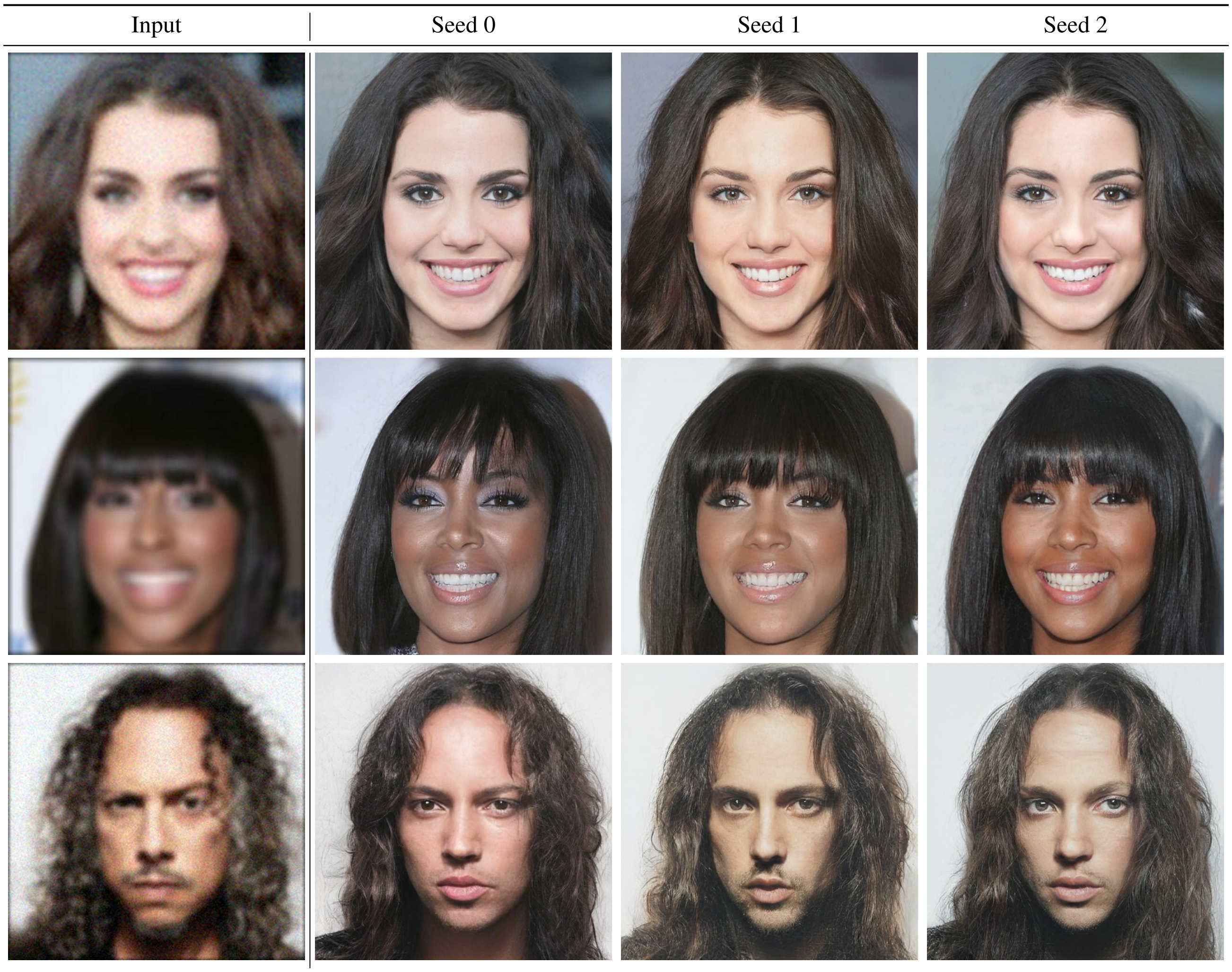

Figure 11. Restored samples with different random seeds when no skip connections are used.

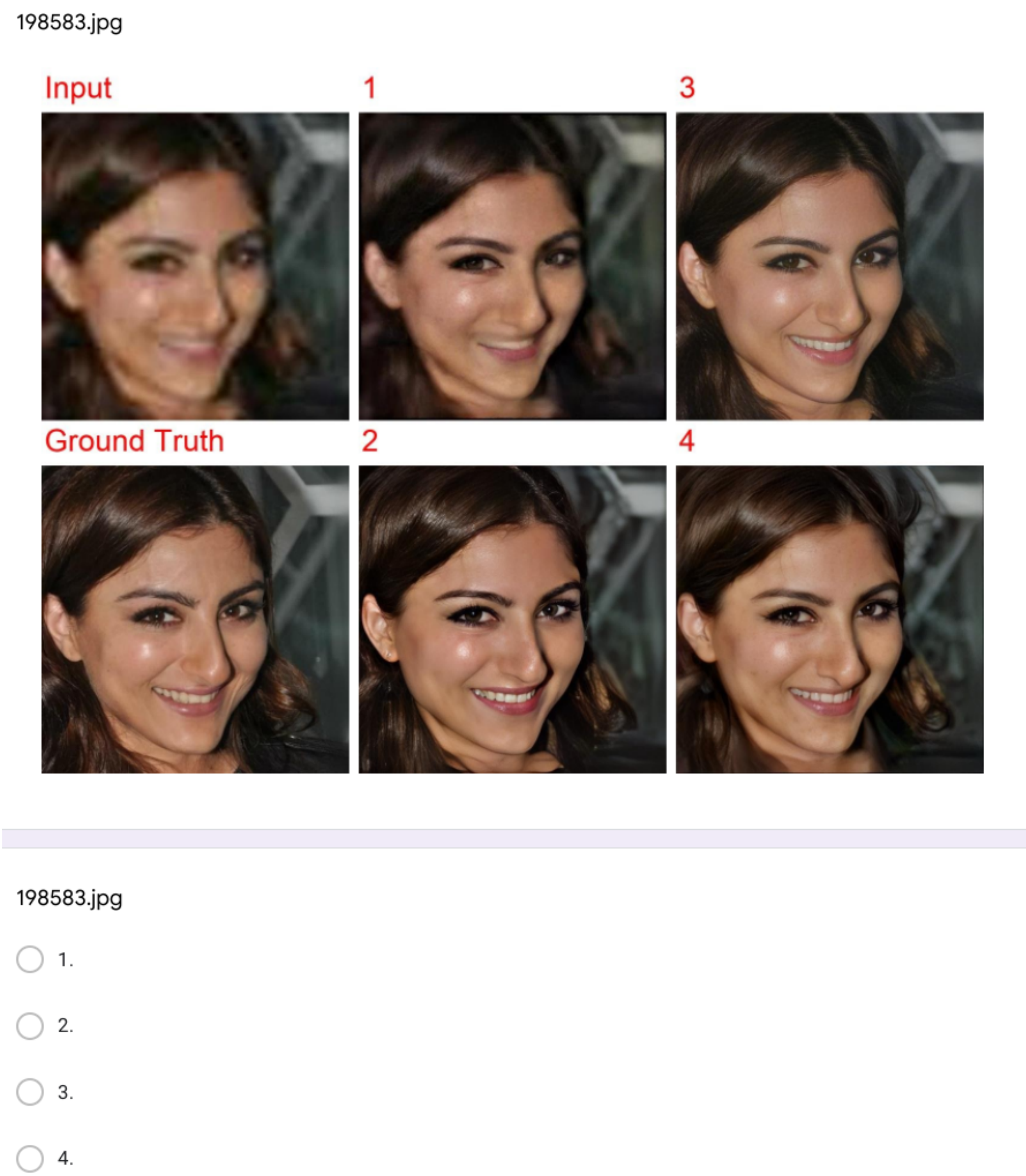

Figure 12. An example of human evaluation

Figure 13. Restoration comparison on real images. The real low-quality images are available in DFDNet [20] public repository. Note that the above real-world degraded images are usually contaminated by unnatural color distortion, which is not synthesized in the standard protocol as in Eq. 11. Our approach is able to correct the color shift and produce natural faces.

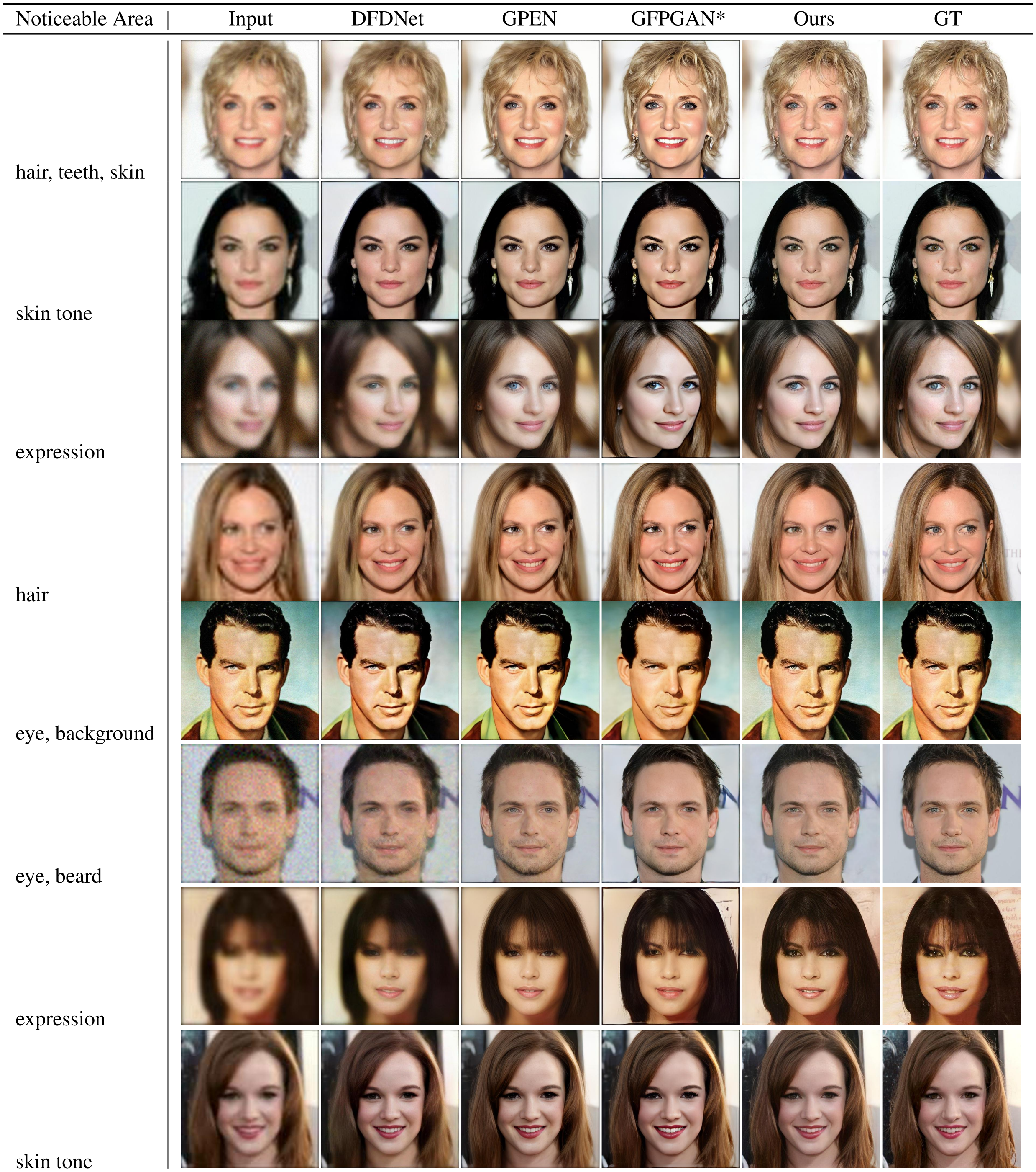

Figure 14. More qualitative comparison across various BFR methods.

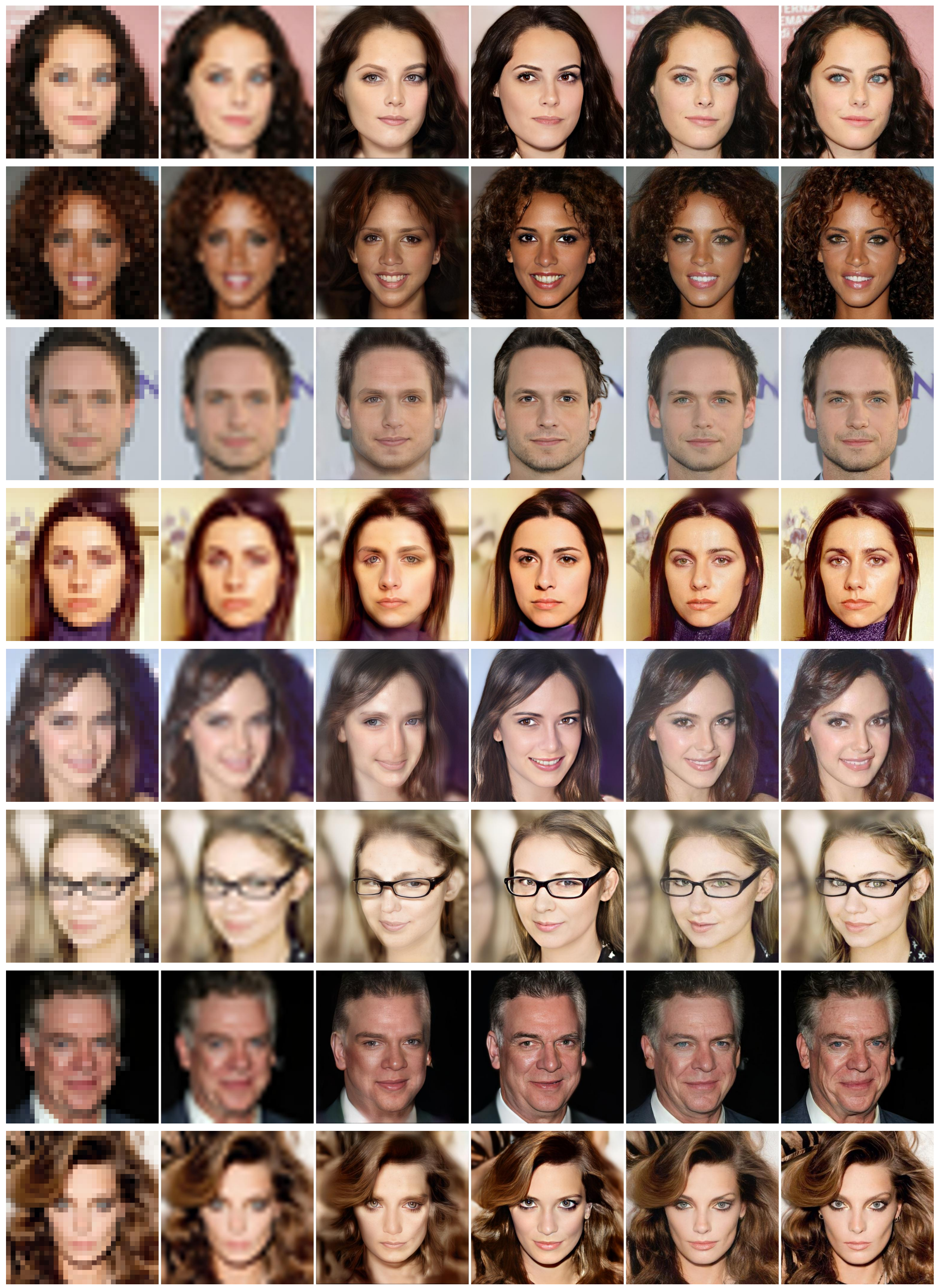

Figure 15. More qualitative comparison across various $16\times$: $32 \times 32 \rightarrow 512 \times 512$ SR methods.

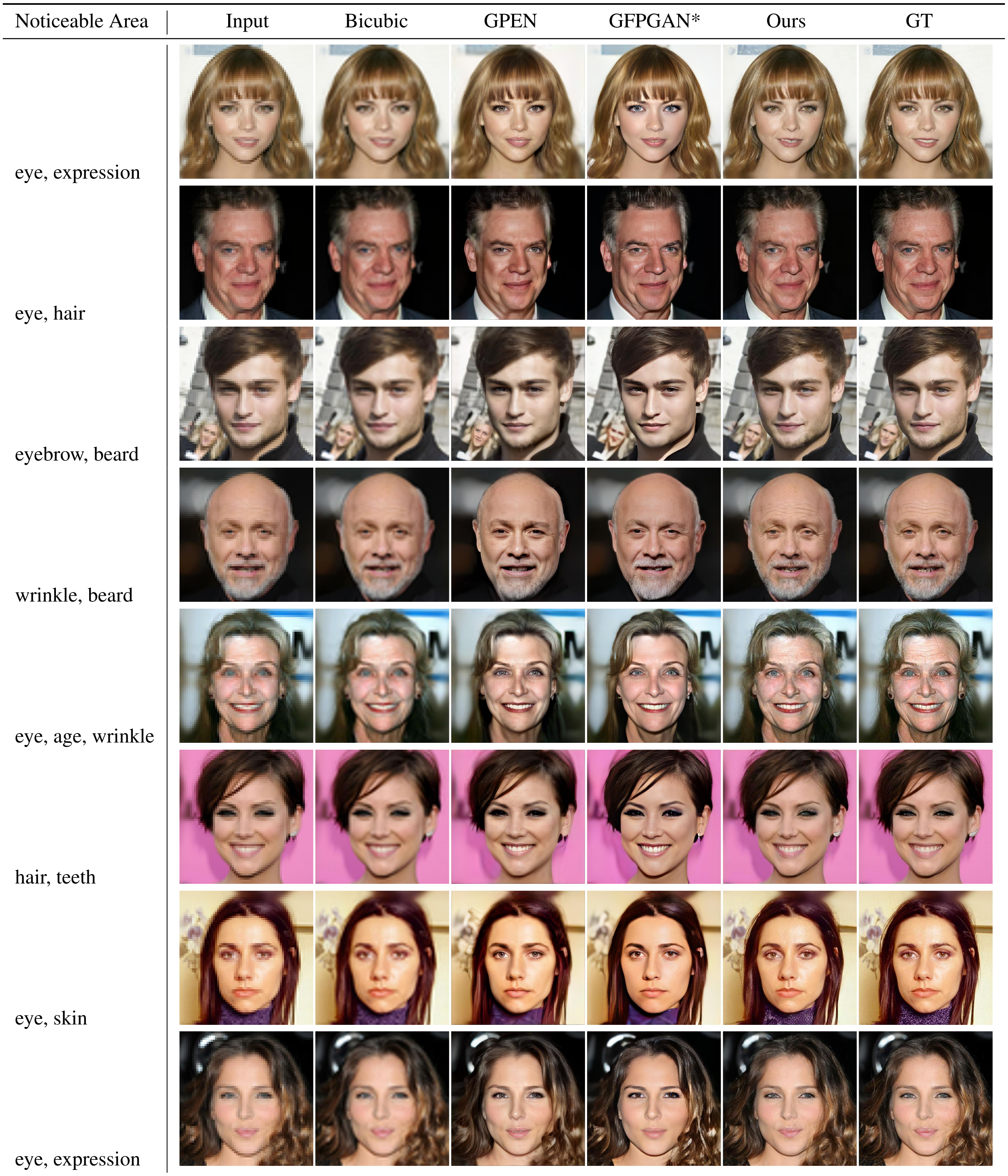

Figure 16. More qualitative comparison across various $8\times : 64 \times 64 \rightarrow 512 \times 512$ SR methods.