Supplementary Material for
Learning Oracle Attention for High-fidelity Face Completion

Tong Zhou[1] Changxing Ding[1] Shaowen Lin[1] Xinchao Wang[2] Dacheng Tao[3]
[1] South China University of Technology [2] Stevens Institute of Technology
[3] UBTECH Sydney AI Centre, School of Computer Science, Faculty of Engineering,
The University of Sydney, Darlington, NSW 2008, Australia
201821011282@mail.scut.edu.cn chxding@scut.edu.cn
eeswlin@mail.scut.edu.cn xinchao.w@gmail.com dacheng.tao@sydney.edu.au

This supplementary material includes five sections. Section A shows comparison results between our method and state-of-the-art methods on the Flickr-Faces-HQ [2] database. Section B provides more qualitative comparions on the CelebA-HQ [1] database. SectionC shows two challenging cases including faces in profile or complex illuminations. Section D shows face completion results of our method using irregular masks. Section E provides the face completion results of our approach on high resolution images (1024×1024). Finally, we introduce the details of the network architecture in Section F.

A. Comparisons on Flickr-Faces-HQ

Method	L1	PSNR	SSIM	LPIPS [7]
CA [4]	1.96%	24.30	0.8896	0.0869
PIC [8]	1.88%	24.53	0.9007	0.0982
GConv [5]	1.85%	24.93	0.8879	0.0879
Ours	**1.50%**	**26.06**	**0.9045**	**0.0693**

Table 1. Quantitative comparisons on the same test set using rectangular masks of random position. Higher SSIM and PSNR values are better; lower L1 error and LPIPS values are better.

We conduct both quantitative and qualitative comparisons between our approach and state-of-the-art methods on the Flickr-Faces-HQ [2] database. Rectangular masks of random position are adopted. All images are resized to 256×256. As the original papers of CA [4], PIC [8], and GConv [5] do not provide the performance of their models on Flickr-Faces-HQ, we use their released codes to train the three models on Flickr-Faces-HQ respectively. Quantitative comparison results are summarized in Table 1, it is shown that our approach outperforms the other methods by large margins. Qualitative comparisons are provided in Figure 1, it is clear that our method is also the best in visual effect.

B. More Comparison Results on CelebA-HQ

We show more qualitative comparisons on the CelebA-HQ [1] database in Figure 2 and Figure 3. In the two figures, rectangular masks of random position and a center mask are utilized, respectively. Intuitively, the center mask is more challenging as most facial components are occluded and it is difficult to find references from the background. It is shown that our method performs the best in visual effect.

C. Special Cases

As shown in Figure 4, we show the results of processing faces in profile or complex illuminations, which are indeed more challenging for the inpainting task due to the data imbalance problem.

D. Results with Irregular Masks

All the above experiments adopt rectangular masks. In this experiment, we show face completion results of our approach using irregular masks. As the masks are irregular now and discriminators usually require rectangular patches as input, we consistently apply the local discriminator and local subdivision discriminator to the central region (128×128) of each training image. The other settings of our approach remain unchanged. Face completion results are illustrated in Figure 5. It is shown that our approach produces face images of high-fidelity even with irregular masks.

E. Results on High Resolution Images

In this experiment, we show the effectiveness of our approach on high resolution face images (1024×1024). The experimental settings are consistent with those on low resolution images (256×256). The results are illustrated in Figure 6, Figure 7, Figure 8 and Figure 9. It is shown that the recovered images by our approach contain rich facial textures. The facial textures are also highly consistent with the ground-truth images. Therefore, the ability of our approach to generate high-fidelity facial images is justified.

F. Network Architecture

We show the architecture details of the generator and discriminators in our model in Table 2 and Table 3 respectively. For high resolution images (1024×1024), the architecture of our model is adjusted slightly, as shown in Table 4 and Table 5.

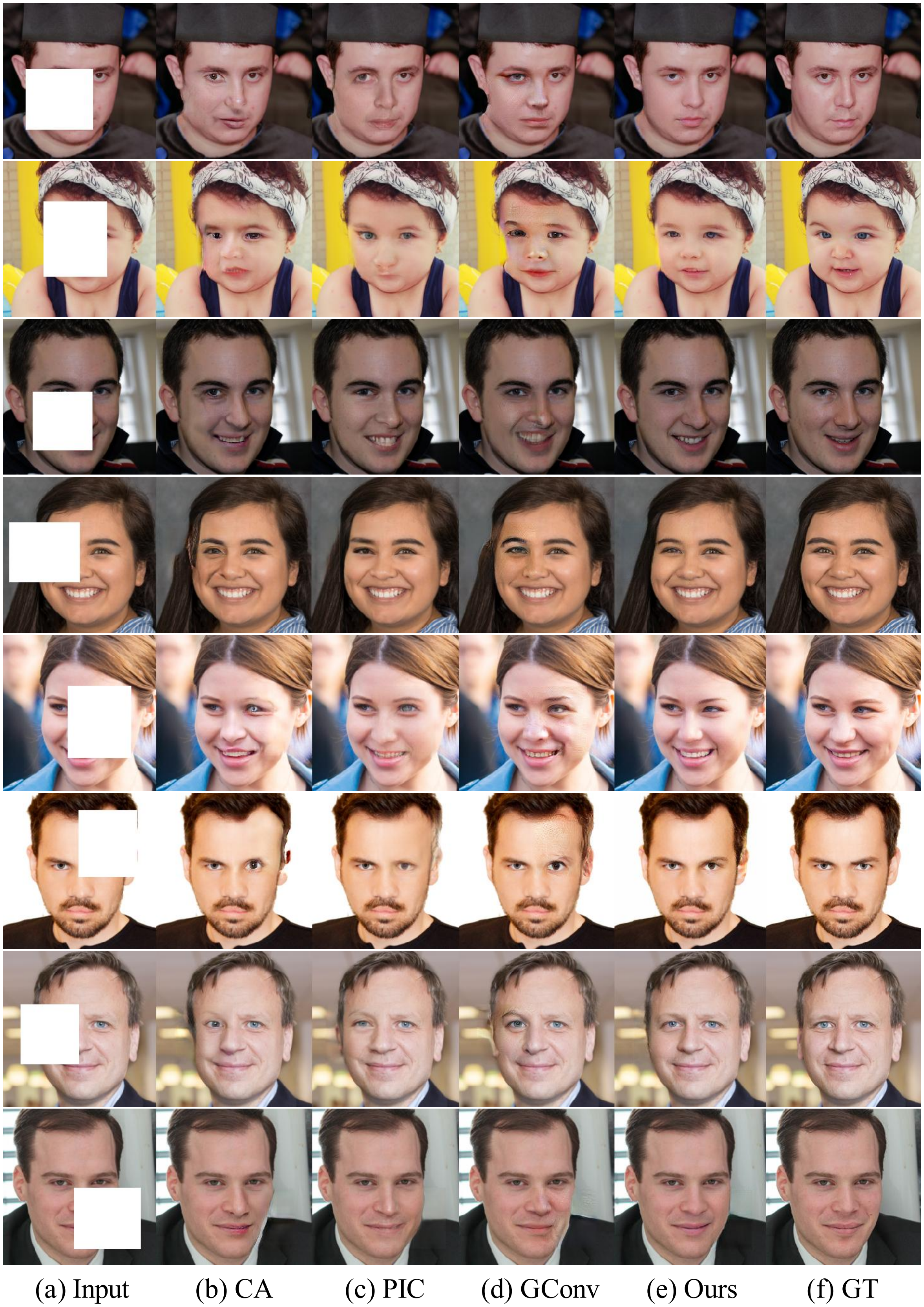
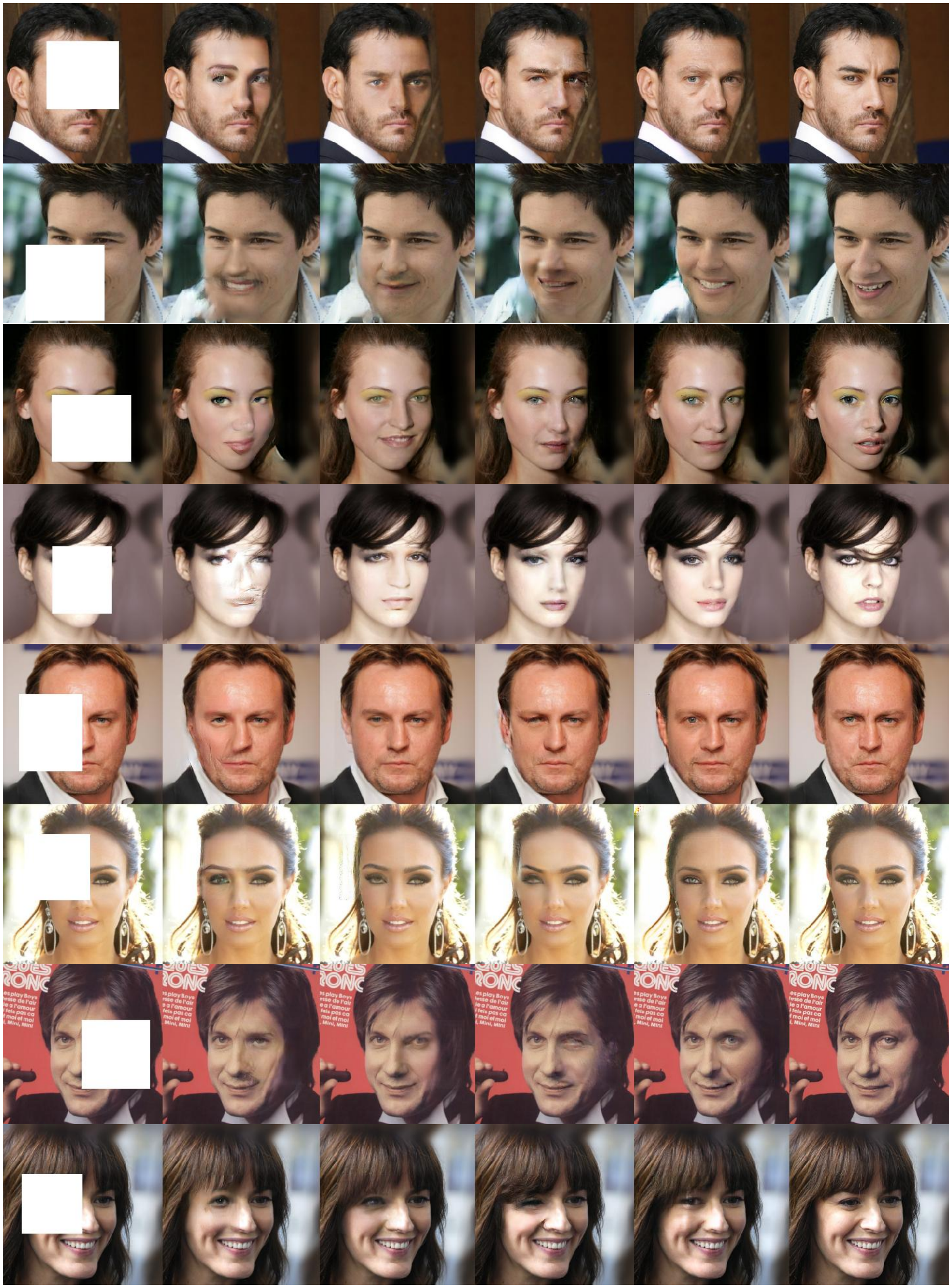


Figure 1. Comparisons on Flickr-Faces-HQ [2] by different methods with random rectangular masks. Three state-of-the-art methods are compared: CA [4], PIC [8] and GConv [5]. Best viewed with zoom-in and pay attention to the details on facial components.



(a) Input (b) CA (c) PIC (d) GConv (e) Ours (f) GT

Figure 2. Comparisons on CelebA-HQ [1] by different methods with random rectangular mask. Three state-of-the-art methods are compared: CA [4], PIC [8] and GConv [5]. Best viewed with zoom-in and pay attention to the details on facial components.

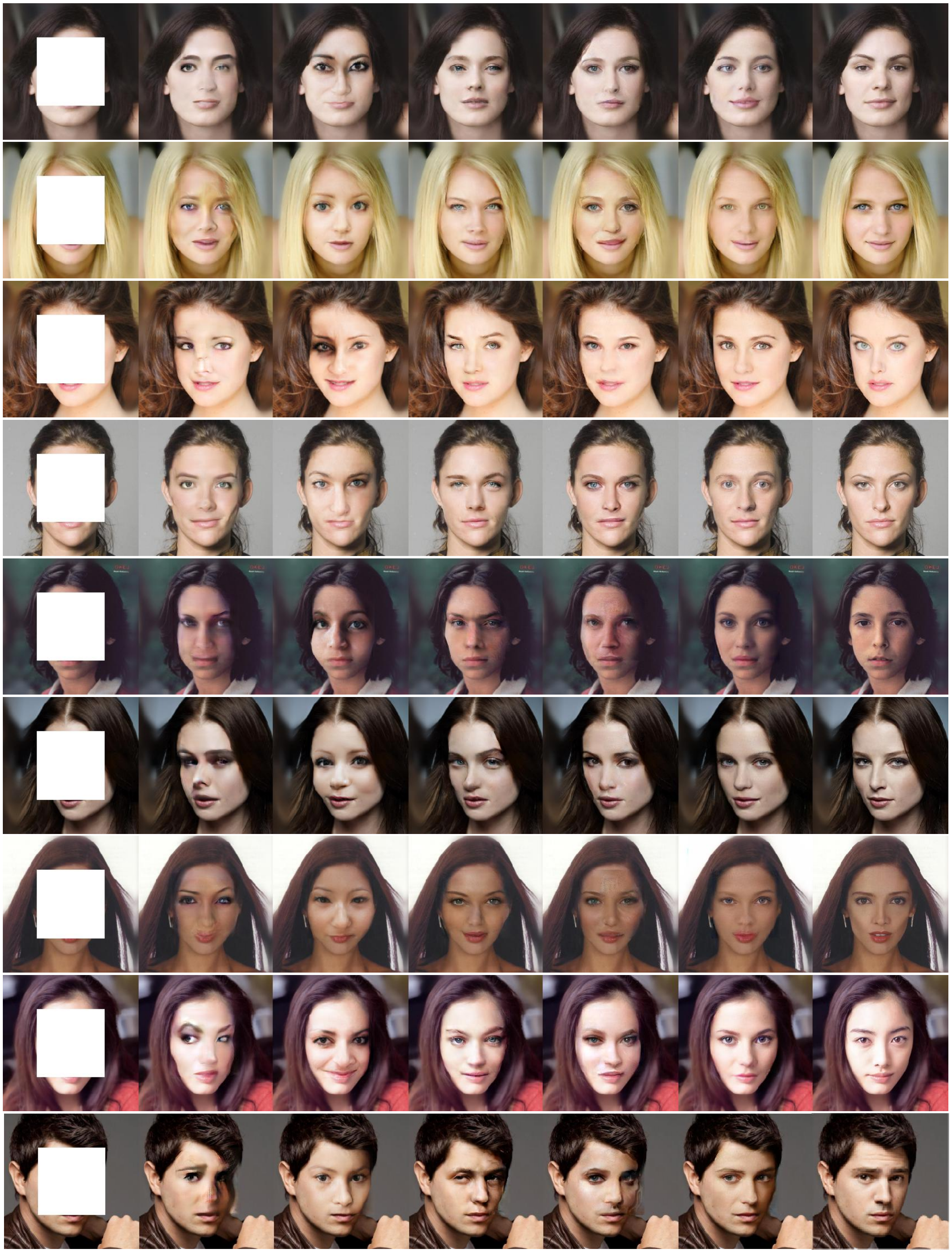


Figure 3. Comparisons on CelebA-HQ [1] by different methods with a center mask (128 $\times$ 128). Four state-of-the-art methods are compared: CA [4], PIC [8], GConv [5] and PEN-Net [6]. Best viewed with zoom-in and pay attention to the details on facial components.

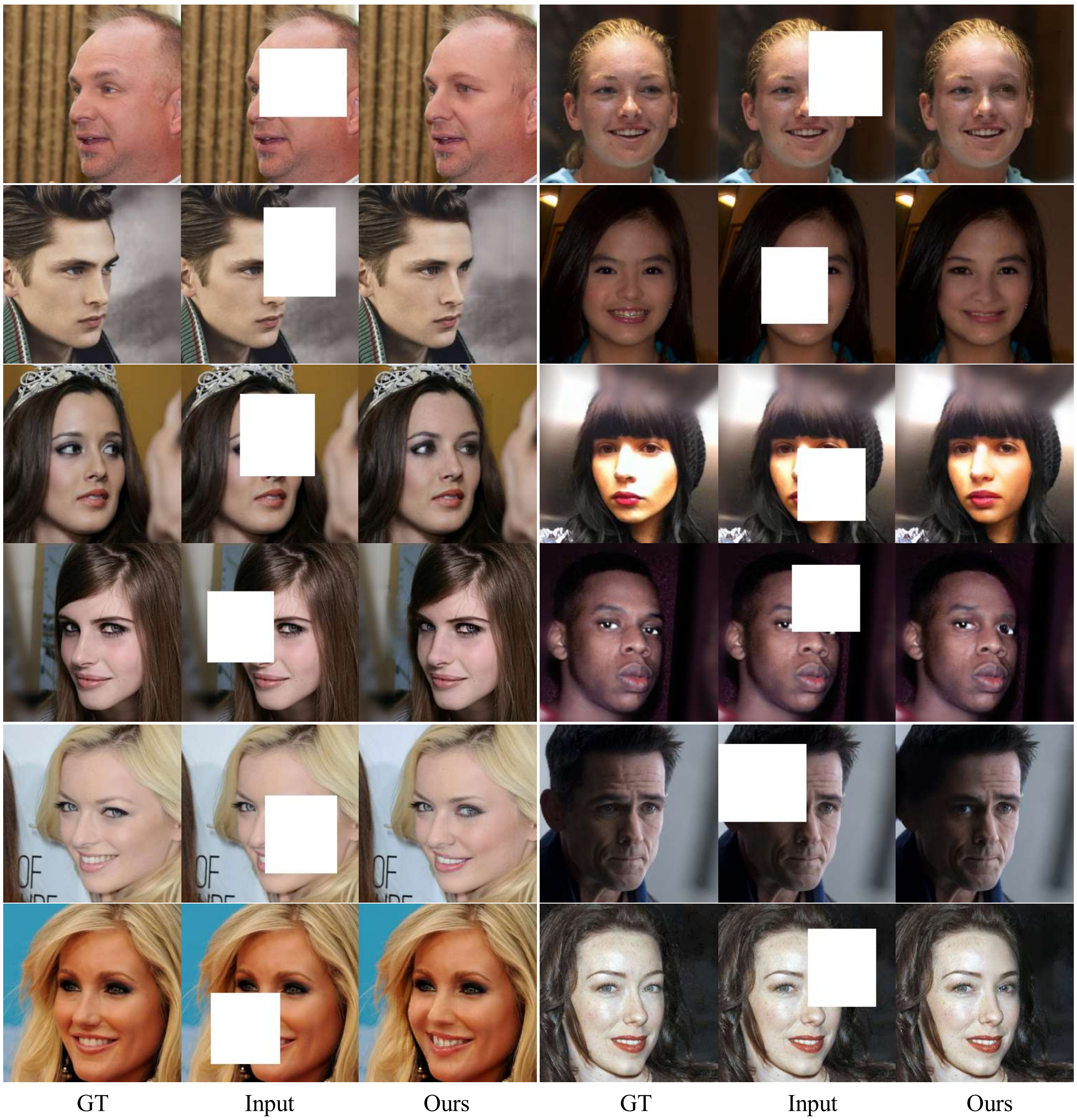


Figure 4. The results of processing faces in profile or complex illuminations. All these images are included in the CelebA-HQ test set. Best viewed with zoom-in.

Figure 5. Results on CelebA-HQ with irregular masks.

Figure 6. Results on high-resolution images of CelebA-HQ (1024×1024).

Figure 7. Results on high-resolution images of CelebA-HQ (1024×1024).

Figure 8. Results on high-resolution images of CelebA-HQ (1024×1024).

Figure 9. Results on high-resolution images of CelebA-HQ (1024×1024).

Layer 1	Conv(7, 7, 64), stride=2; ReLU
Layer 2	Conv(5, 5, 128), stride=2; BN; ReLU
Layer 3	Conv(3, 3, 256), stride=2; BN; ReLU
Layer 4	Conv(3, 3, 512), stride=2; BN; ReLU
Layer 5	Conv(3, 3, 512), stride=2; BN; ReLU
Layer 6	Conv(3, 3, 512), stride=2; BN; ReLU
Layer 7	Conv(3, 3, 512), stride=2; BN; ReLU
Layer 8	Conv(3, 3, 512), stride=2; BN; ReLU
Layer 9	Upsample(factor = 2); Concat(w/ Layer 7); Conv(3, 3, 512), stride=1; BN; LReLU(slope = 0.2);
Layer 10	Upsample(factor = 2); Concat(w/ Layer 6); Conv(3, 3, 512), stride=1; BN; LReLU(slope = 0.2);
Layer 11	Upsample(factor = 2); Concat(w/ Layer 5); Conv(3, 3, 512), stride=1; BN; LReLU(slope = 0.2);
Layer 12	Upsample(factor = 2); Concat(w/ Layer 4); Conv(3, 3, 512), stride=1; BN; LReLU(slope = 0.2); Dual Spatial Attention Module(DSA);
Layer 13	Upsample(factor = 2); Concat(w/ Layer 3); Conv(3, 3, 256), stride=1; BN; LReLU(slope = 0.2); Dual Spatial Attention Module(DSA);
Layer 14	Upsample(factor = 2); Concat(w/ Layer 2); Conv(3, 3, 128), stride=1; BN; LReLU(slope = 0.2); Dual Spatial Attention Module(DSA);
Layer 15	Upsample(factor = 2); Concat(w/ Layer 1); Conv(3, 3, 64), stride=1; BN; LReLU(slope = 0.2);
Layer 16	Upsample(factor = 2); Concat(w/ Input); Conv(3, 3, 3), stride=1; Sigmoid

Table 2. The architecture of the generator. BN denotes batch normalization and LReLU denotes leaky ReLU. We adopt a very similar U-Net structure as used in [3] for the generator. The difference lies in two aspects: (1) we adopt conventional convolution rather than partial convolution; (2) we equip U-net with the Dual Spatial Attention (DSA) module.

Layer 1	Conv(4, 4, C), stride=2; LReLU(slope = 0.2);
Layer 2	Conv(4, 4, $2 \times C$), stride=2; LReLU(slope = 0.2);
Layer 3	Conv(4, 4, $4 \times C$), stride=2; LReLU(slope = 0.2);
Layer 4	Conv(4, 4, $8 \times C$), stride=1; LReLU(slope = 0.2);
Layer 5	Conv(4, 4, 1), stride=1

Table 3. The architecture of discriminators. C denotes the number of channels of the convolutional layers. For the local subdivision discriminator and the four organ discriminators imposed on facial components, C equals to 32. For the global and local discriminators, C equals to 64 and 48, respectively.

Layer 1	Conv(7, 7, 64), stride=2; ReLU
Layer 2	Conv(5, 5, 128), stride=2; BN; ReLU
Layer 3	Conv(3, 3, 256), stride=2; BN; ReLU
Layer 4	Conv(3, 3, 512), stride=2; BN; ReLU
Layer 5	Conv(3, 3, 512), stride=2; BN; ReLU
Layer 6	Conv(3, 3, 512), stride=2; BN; ReLU
Layer 7	Conv(3, 3, 512), stride=2; BN; ReLU
Layer 8	Conv(3, 3, 512), stride=2; BN; ReLU
Layer 9	Conv(3, 3, 512), stride=2; BN; ReLU
Layer 10	Conv(3, 3, 512), stride=2; BN; ReLU
Layer 11	Upsample(factor = 2); Concat(w/ Layer 9); Conv(3, 3, 512), stride=1; BN; LReLU(slope = 0.2);
Layer 12	Upsample(factor = 2); Concat(w/ Layer 8); Conv(3, 3, 512), stride=1; BN; LReLU(slope = 0.2);
Layer 13	Upsample(factor = 2); Concat(w/ Layer 7); Conv(3, 3, 512), stride=1; BN; LReLU(slope = 0.2);
Layer 14	Upsample(factor = 2); Concat(w/ Layer 6); Conv(3, 3, 512), stride=1; BN; LReLU(slope = 0.2); Dual Spatial Attention Module(DSA);
Layer 15	Upsample(factor = 2); Concat(w/ Layer 5); Conv(3, 3, 512), stride=1; BN; LReLU(slope = 0.2); Dual Spatial Attention Module(DSA);
Layer 16	Upsample(factor = 2); Concat(w/ Layer 4); Conv(3, 3, 512), stride=1; BN; LReLU(slope = 0.2); Dual Spatial Attention Module(DSA);
Layer 17	Upsample(factor = 2); Concat(w/ Layer 3); Conv(3, 3, 256), stride=1; BN; LReLU(slope = 0.2);
Layer 18	Upsample(factor = 2); Concat(w/ Layer 2); Conv(3, 3, 128), stride=1; BN; LReLU(slope = 0.2);
Layer 19	Upsample(factor = 2); Concat(w/ Layer 1); Conv(3, 3, 64), stride=1; BN; LReLU(slope = 0.2);
Layer 20	Upsample(factor = 2); Concat(w/ Input); Conv(3, 3, 3), stride=1; Sigmoid

Table 4. The architecture of the generator for input images of 1024×1024. To accommodate the high resolution, we add two convolutional layers for both the encoder and decoder of the generator in Table 2.

Layer 1	Conv(4, 4, C), stride=2; LReLU(slope = 0.2);
Layer 2	Conv(4, 4, $2 \times C$), stride=2; LReLU(slope = 0.2);
Layer 3	Conv(4, 4, $4 \times C$), stride=2; LReLU(slope = 0.2);
Layer 4	Conv(4, 4, $8 \times C$), stride=2; LReLU(slope = 0.2);
Layer 5	Conv(4, 4, $8 \times C$), stride=2; LReLU(slope = 0.2);
Layer 6	Conv(4, 4, $8 \times C$), stride=1; LReLU(slope = 0.2);
Layer 7	Conv(4, 4, 1), stride=1

Table 5. The architecture of discriminators for input images of 1024×1024. To accommodate the high resolution, we add two convolutional layers for discriminators in Table 3.

References

[1] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[2] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019.

[3] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *ECCV*, pages 85–100, 2018.

[4] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *CVPR*, pages 5505–5514, 2018.

[5] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *CVPR*, 2019.

[6] Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. Learning pyramid-context encoder network for high-quality image inpainting. In *CVPR*, pages 1486–1494, 2019.

[7] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018.

[8] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *CVPR*, pages 1438–1447, 2019.