Supplementary

Hongyu Liu Bin Jiang * Yi Xiao Chao Yang
College of Computer Science and Electronic Engineering
Hunan University
{kumapower, jiangbin, yangchaoedu, yixiao_csee}@hnu.edu.cn

A. Definition of Masked Region in Feature Maps

As the CSA layer works based on both the masked region M and unmasked region $\overline{M}$ in feature maps, thus we need to give a definition of masked region in feature maps. In our implementation, we introduce a masked image in which each pixel value of known regions is 0 and that for unknown regions is 1. When considering centering masks, since the CSA layer locates at the resolution of 32×32 and the centering mask covers half of the input image I_{in}, we set the size of region M in feature maps as 16×16. While for irregular masks, following the idea of SH [35], we first define a network that has the same architecture with the encoder of rough network but with the network width of 1, the network has only convolution layers and all the elements of the filters are 1/16. Then taking the masked image as input, we obtain the feature with 32×32 resolution which is the 3rd down-sample output of the network. Finally, for the value at each position of the feature, we set those values larger than 5/16 to 1, which means this position belongs to masked region M in feature maps.

B. Network Architectures

As a supplement to the content of Section 3, we will report more details of our network architectures in the following. First, Table 1 and Table 10 depict the specific design of architecture of our rough network and refinement network respectively. On one hand, the architecture of rough network is the same as pix to pix [17]. On the other hand, the refinement network uses 3×3 convolutions to double the channel and uses 4×4 convolutions to reduce the spatial size to half. Then, the architecture of patch and feature patch discriminators are shown in Table 2 and Table 3 respectively, where the VGG 4-3 denotes all the layers before Relu 4_3 of VGG-16 network.

The architecture of rough network
[Layer 1] Conv. (4, 4, 64), stride=2;
[Layer 2] LReLU; Conv. (4, 4, 128), stride=2; IN;
[Layer 3] LReLU; Conv. (4, 4, 256), stride=2; IN;
[Layer 4] LReLU; Conv. (4, 4, 512), stride=2; IN;
[Layer 5] LReLU; Conv. (4, 4, 512), stride=2; IN;
[Layer 6] LReLU; Conv. (4, 4, 512), stride=2; IN;
[Layer 7] LReLU; Conv. (4, 4, 512), stride=2; IN;
[Layer 8] LReLU; Conv. (4, 4, 512), stride=2;
[Layer 9] ReLU; DeConv. (4, 4, 512), stride=2; IN; Concatenate(Layer 9, Layer 7);
[Layer 10] ReLU; DeConv. (4, 4, 512), stride=2; IN; Concatenate(Layer 10, Layer 6);
[Layer 11] ReLU; DeConv. (4, 4, 512), stride=2; IN; Concatenate(Layer 11, Layer 5);
[Layer 12] ReLU; DeConv. (4, 4, 512), stride=2; IN; Concatenate(Layer 12, Layer 4);
[Layer 13] ReLU; DeConv. (4, 4, 256), stride=2; IN; Concatenate(Layer 13, Layer 3);
[Layer 14] ReLU; DeConv. (4, 4, 128), stride=2; IN; Concatenate(Layer 16, Layer 2);
[Layer 15] ReLU; DeConv. (4, 4, 64), stride=2; IN; Concatenate(Layer 17, Layer 1);
[Layer 16] ReLU; DeConv. (4, 4, 3), stride=2; Tanh;

Table 1. The architecture of the Rough network. IN represents InstanceNorm and LReLU donates leaky ReLU with the slope of 0.2.

The architecture of patch discriminator
[layer 1] Conv. (4, 4, 64), stride=2; LReLU;
[layer 2] Conv. (4, 4, 128), stride=2; IN; LReLU;
[layer 3] Conv. (4, 4, 256), stride=2; IN; LReLU;
[layer 4] Conv. (4, 4, 512), stride=1; IN; LReLU;
[layer 5] Conv. (4, 4, 1), stride=1;

Table 2. The architecture of the patch discriminative network. IN represents InstanceNorm and LReLU donates leaky ReLU with the slope of 0.2.

*Corresponding author

The architecture of feature patch discriminator
[layer 1] VGG 4_3 layer
[layer 2] Conv. (4, 4, 512), stride=2; LReLU;
[layer 3] Conv. (4, 4, 512), stride=1; IN; LReLU;
[layer 4] Conv. (4, 4, 512), stride=1;

Table 3. The architecture of the feature patch discriminative network. IN represents InstanceNorm and LReLU donates leaky ReLU with the slope of 0.2.

C. Quantitative Comparison of Ablation Study

Effect of CSA layer When examining the effect of CSA layer, we select validation images from *butte* categories of Places2 dataset and replace the CSA layer with a conventional 3×3 layer and the contextual attention layer [40] respectively. Table 4 lists the evaluation results. From the results in Table 4, we can see that the CSA layer outperforms all the other layers.

	$L_1^-(\%)$	$L_2^-(\%)$	$SSIM^+$	$PSNR^+$
With Conv	2.56	0.54	0.819	23.71
With CA	2.51	0.56	0.817	23.74
With CSA	**2.37**	**0.52**	**0.823**	**24.04**

Table 4. The effect of CSA layer. $^-$Lower is better. $^+$Higher is better

Effect of CSA layer at different positions In order to compare the effect of CSA layer at different positions, we select validation images from *canyon* categories of Places2 dataset to make quantitative comparisons. Table 5 lists the evaluation results. From the results in Table 5, we find that better tradeoff between efficiency and performance can be achieved by our model when the CSA layer is embedded into the 3th down-sample positions.

	$L_1^-(\%)$	$L_2^-(\%)$	$SSIM^+$	$PSNR^+$
4	3.06	0.75	0.797	22.14
2	2.92	**0.70**	**0.803**	**22.61**
3	**2.83**	0.71	0.802	22.48

Table 5. The effect of CSA layer at different positions. $^-$Lower is better. $^+$Higher is better

Effect of consistency loss In order to verify the validity of consistency loss L_c , we select validation images from *butte* categories of Places2 dataset to make quantitative comparisons. Table 6 lists the evaluation results. From the results in Table 6, we can see that the consistency loss can help our model performances better.

Effect of feature patch discriminator We further conduct experiments to validate the effect of feature patch discriminator. We select validation images from *canyon* categories of Places2 dataset to make quantitative comparisons. Table 7 lists the evaluation results. From the results in Table 7, it can be seen that our feature patch discriminator is better than others.

	$L_1^-(\%)$	$L_2^-(\%)$	$SSIM^+$	$PSNR^+$
No L_c	2.39	0.53	0.823	23.92
With L_c	**2.37**	**0.52**	**0.823**	**24.04**

Table 6. The effect of consistency loss. $^-$Lower is better. $^+$Higher is better

	$L_1^-(\%)$	$L_2^-(\%)$	$SSIM^+$	$PSNR^+$
a	3.07	0.77	0.793	22.12
b	2.99	0.77	0.794	22.16
c	**2.83**	**0.71**	**0.802**	**22.48**

Table 7. The effect of feature patch discriminator. a, b and c are respectively the results when we use patch discriminator,patch and SRFeat feature discriminators [29], patch and our feature patch discriminators. $^-$Lower is better. $^+$Higher is better

D. More Comparisons Results

More comparisons with CA [40], SH [35], PC [23] and GC [39] on Paris StreetView [8], Places2 [24] and CelebA [43] are also conducted. Please refer to Fig 1 and 2 for more results on Places2 and CelebA with centering mask. And for comparison on irregular masks, please refer to Fig 3 and 4 for results on Paris StreetView and CelebA datasets. Table 8 lists the evaluation results with centering mask on Place2 dataset, the scene categories selected from Places2 is *butte*. Table 9 lists the evaluation results with irregular masks on Paris StreetView dataset. It is obvious that our model outperforms state-of-the-art approaches in both structural consistency and detail richness, and the local pixel continuity is well assured since the CSA layer considers the semantic relevance between the holes features. As a side contribution, we will release the pre-trained model and codes.

	$L_1^-(\%)$	$L_2^-(\%)$	$SSIM^+$	$PSNR^+$
CA	4.08	1.02	0.704	20.69
SH	4.04	0.91	0.738	21.55
CSA	**2.37**	**0.52**	**0.823**	**24.04**

Table 8. Comparison results over Place2 (*butte*) with centering hole between CA [40], SH [35], and Ours. $^-$Lower is better. $^+$Higher is better

E. More Results on CelebA, Paris StreetView, Places2

CelebA Fig 5 and Fig 6 show more results obtained by our full model with centering and irregular masks respectively, where the model is trained on CelebA dataset. We resize image to 256×256 for both training and evaluation.

Paris StreetView We also perform experiments on our full model trained on Paris StreetView dataset with irregular

	Mask	PC	GC	CSA
$L_1^-(\%)$	10-20%	1.47	1.14	**1.05**
	20-30%	2.12	1.71	**1.41**
	30-40%	3.49	3.19	**2.69**
	40-50%	4.58	4.49	**3.70**
$L_2^-(\%)$	10-20%	0.17	0.14	**0.08**
	20-30%	0.28	0.22	**0.13**
	30-40%	0.60	0.57	**0.45**
	40-50%	0.86	0.90	**0.68**
$PSNR^+$	10-20%	28.91	29.58	**32.67**
	20-30%	26.78	27.43	**30.32**
	30-40%	23.27	23.19	**24.85**
	40-50%	21.67	21.33	**23.10**
$SSIM^+$	10-20%	0.937	0.945	**0.972**
	20-30%	0.894	0.920	**0.951**
	30-40%	0.815	0.846	**0.873**
	40-50%	0.678	0.731	**0.768**

Table 9. Comparison results over Paris StreetView with irregular mask between PC [23], GC [39], and Ours. $^-$Lower is better. $^+$Higher is better

masks, and the results are shown in Fig 7. We resize image to 256×256 for both training and evaluation.

Places2 Fig 8 shows more results obtained by our full model with centering masks, where the model is trained on Places2 dataset. The scene categories selected from Places2 dataset are canyon and butte. We also resize the images to 256×256 for both training and evaluation.

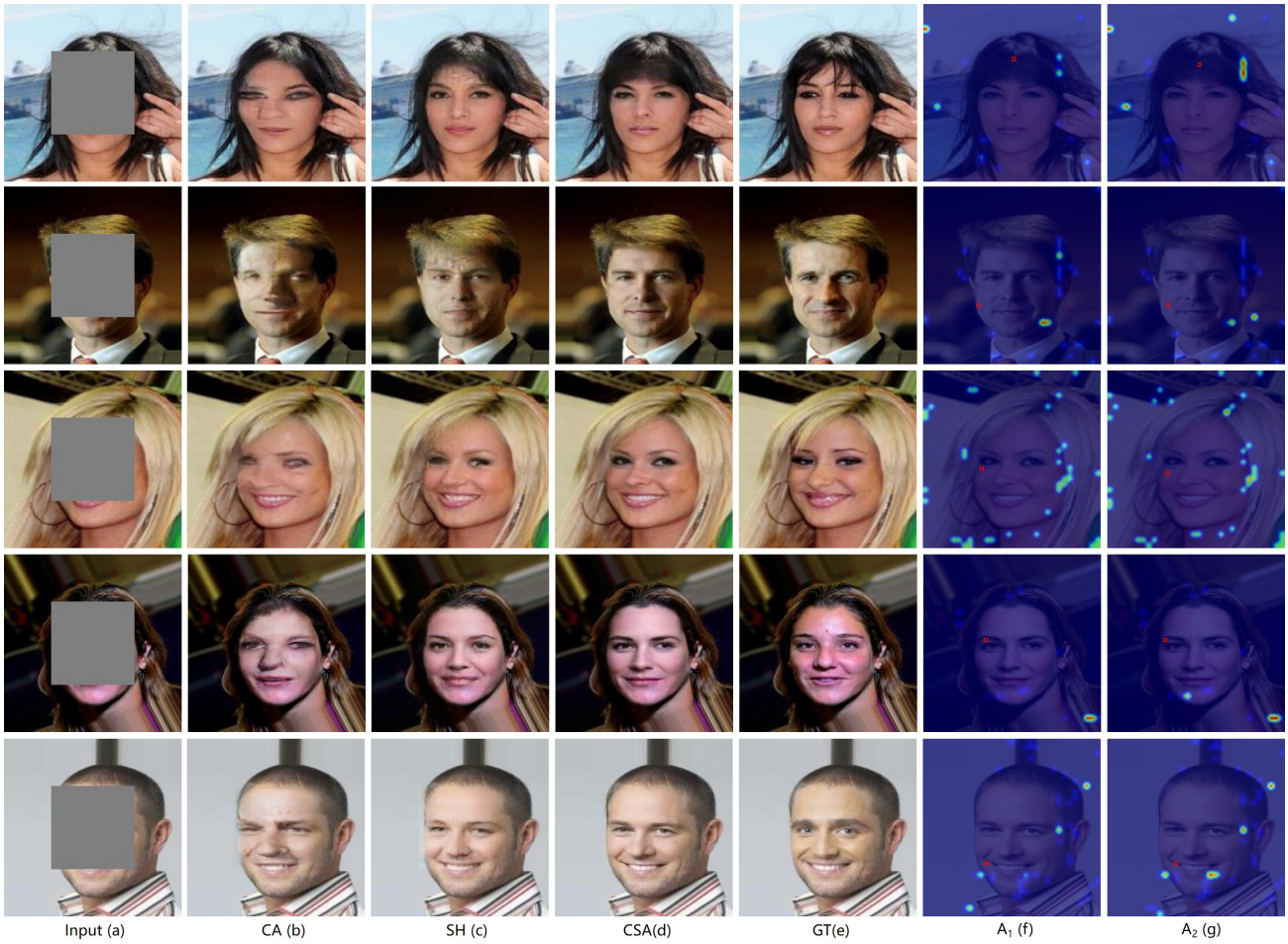

Figure 1. Qualitative comparisons on Celeba with centering masks. A_1 and A_2 are attention maps of two adjacent pixels, the 1st, 2nd, and 3rd rows are the attention maps of up and down adjacent pixels, the 4th and 5th rows are the attention maps of left and right adjacent pixels.

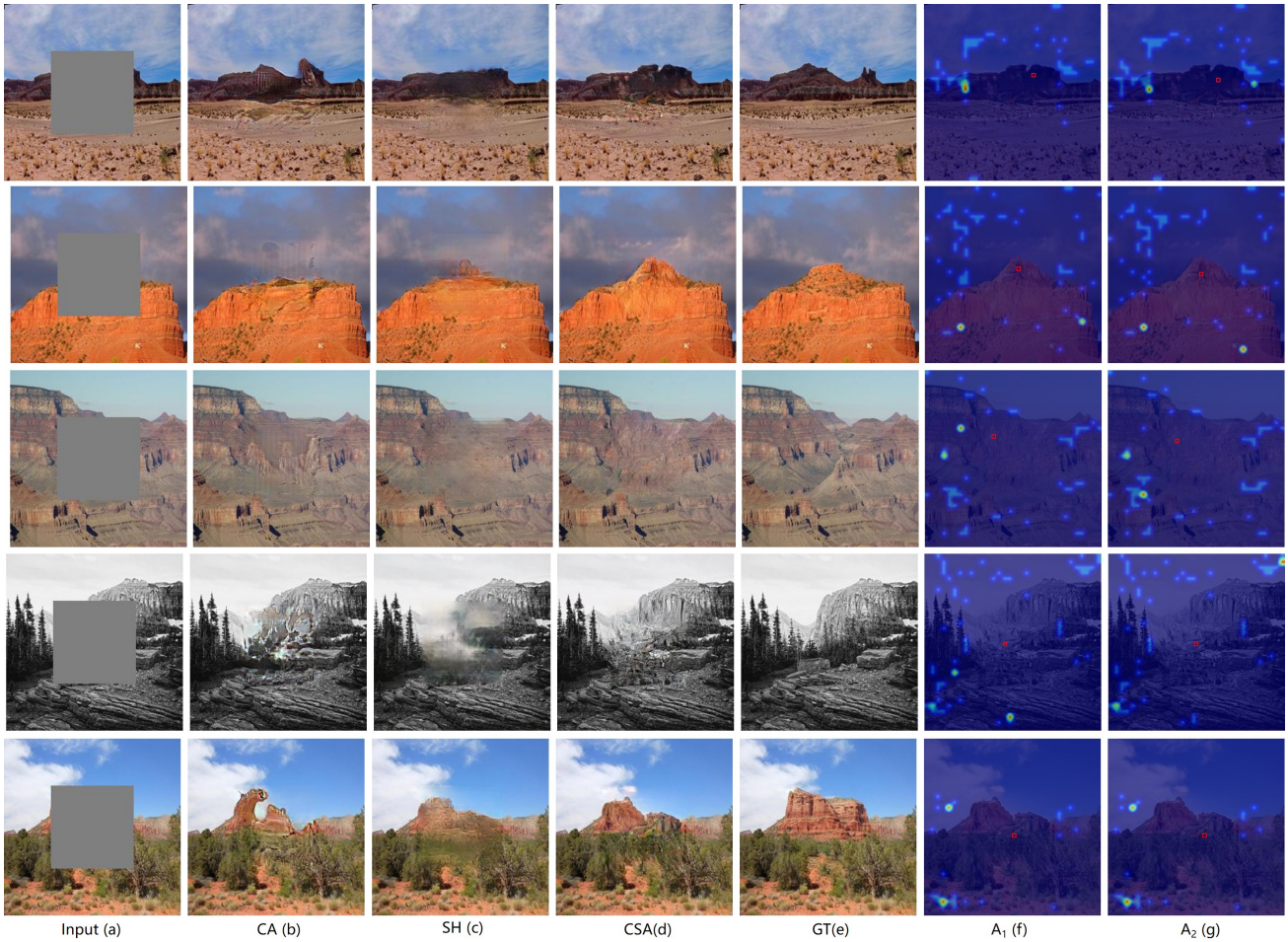

Figure 2. Qualitative comparisons on Places2 with centering masks. A_1 and A_2 are attention maps of two adjacent pixels, the 1st, 2nd, and 3rd rows are the attention maps of up and down adjacent pixels, the 4th and 5th rows are the attention maps of left and right adjacent pixels.

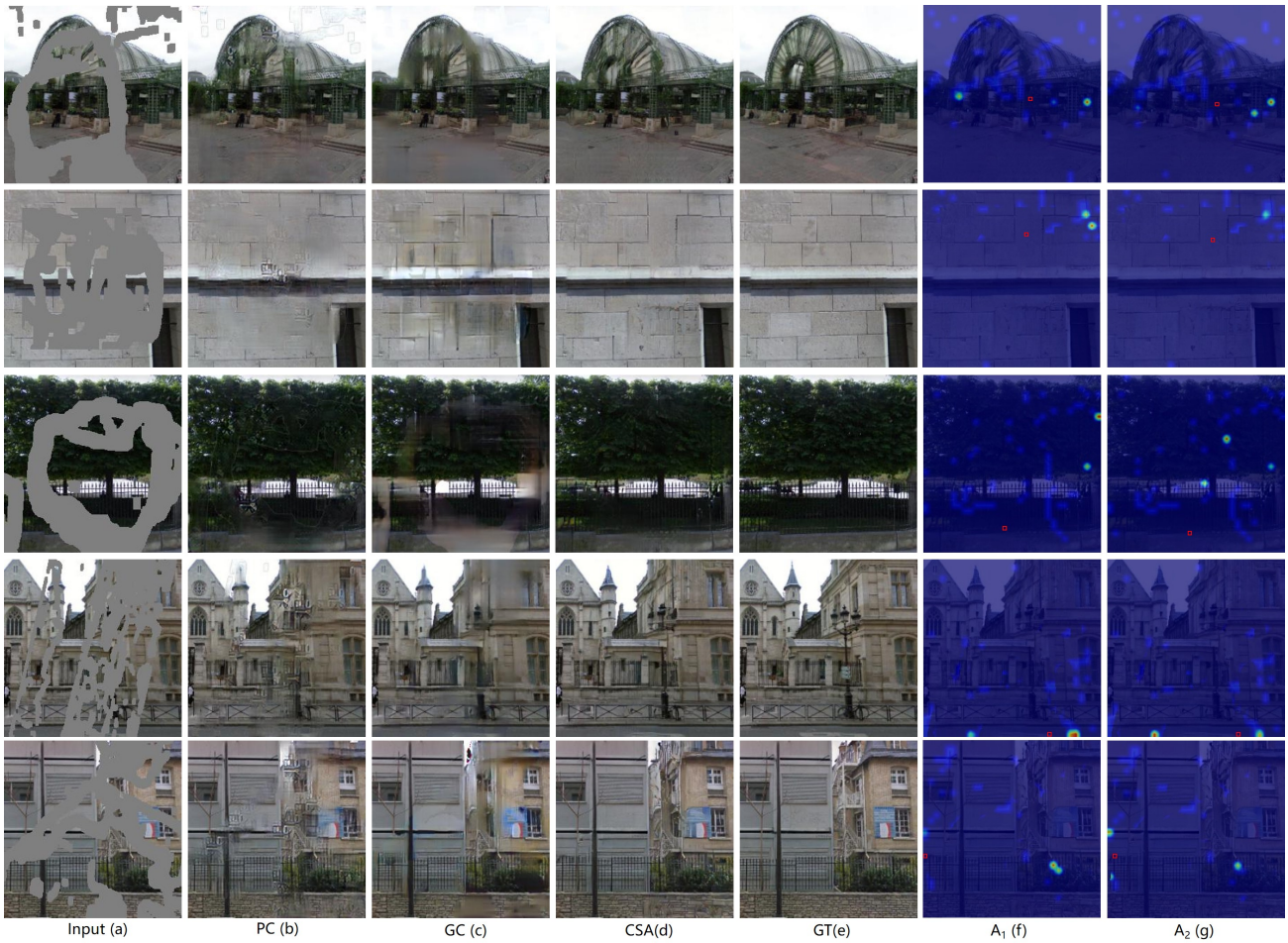

Figure 3. Qualitative comparisons on Paris StreetView with irregular masks. A_1 and A_2 are attention maps of two adjacent pixels, the 1st, 2nd, and 3rd rows are the attention maps of up and down adjacent pixels, the 4th and 5th rows are the attention maps of left and right adjacent pixels.

Figure 4. Qualitative comparisons on CelebA with irregular masks. A_1 and A_2 are attention maps of two adjacent pixels, the 1st, 2nd, and 3rd rows are the attention maps of up and down adjacent pixels, the 4th and 5th rows are the attention maps of left and right adjacent pixels

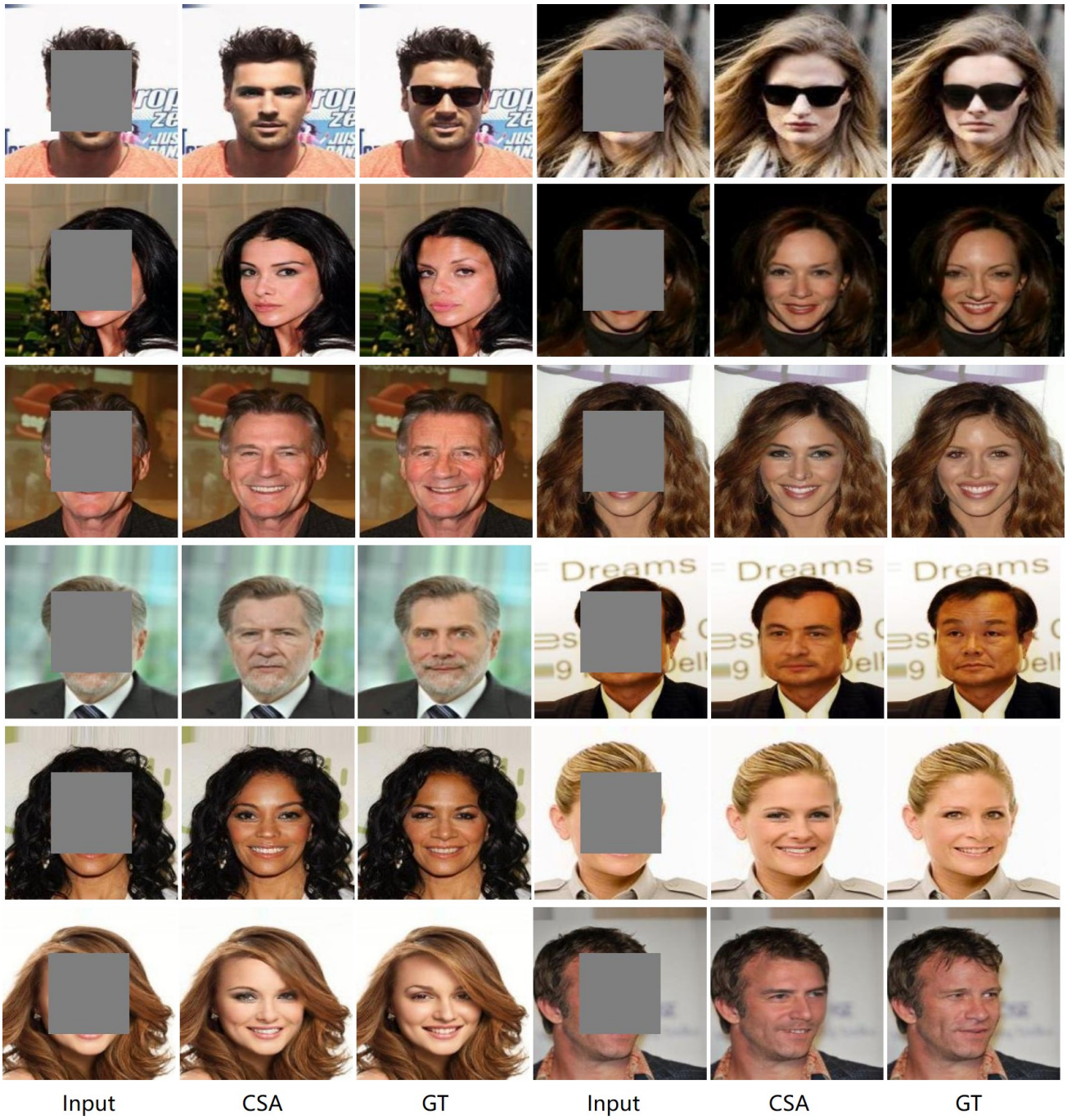

Figure 5. More results on CelebA with centering masks.

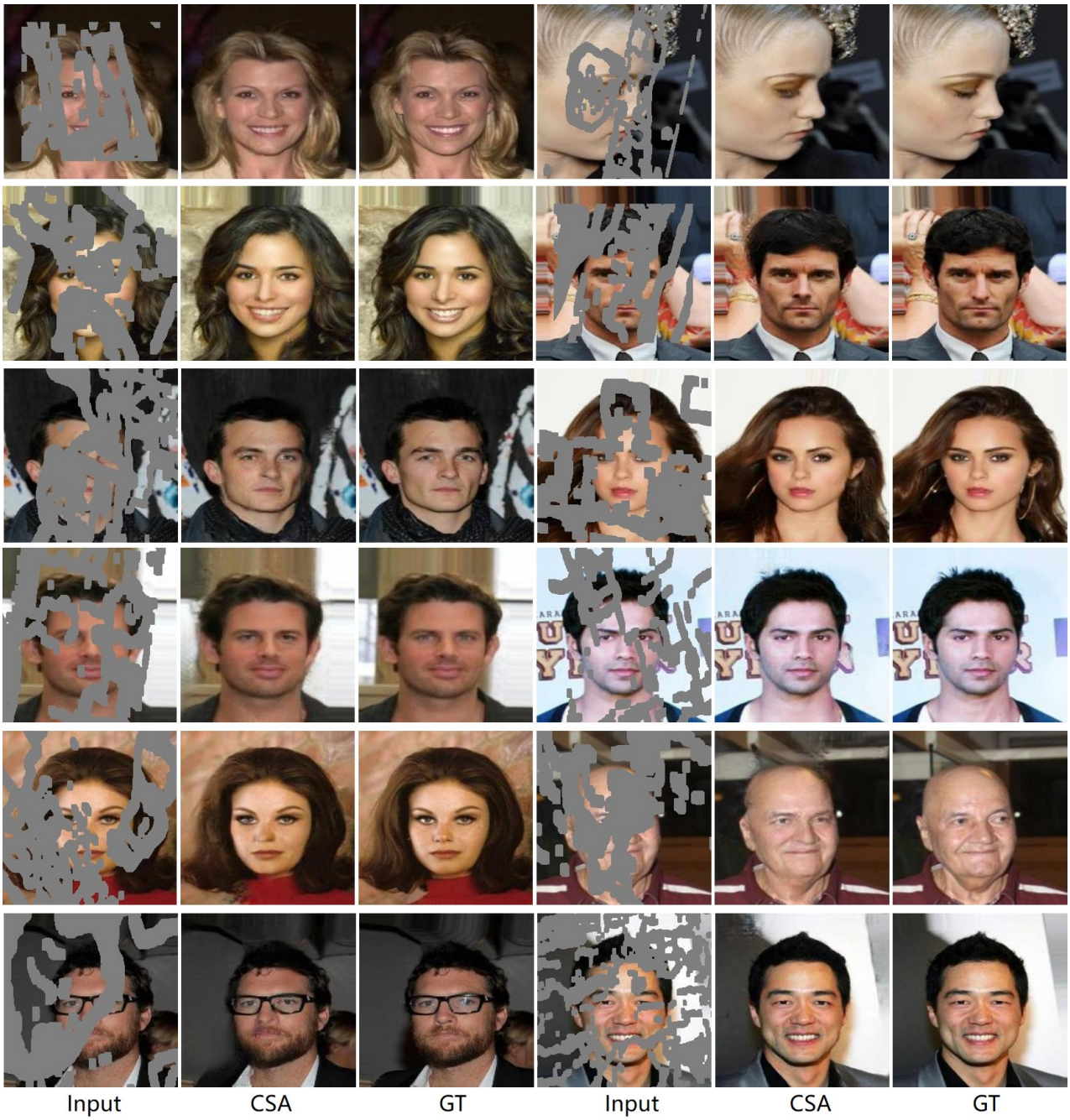

Figure 6. More results on CelebA with irregular masks.

Figure 7. More results on Paris StreetView with irregular masks.

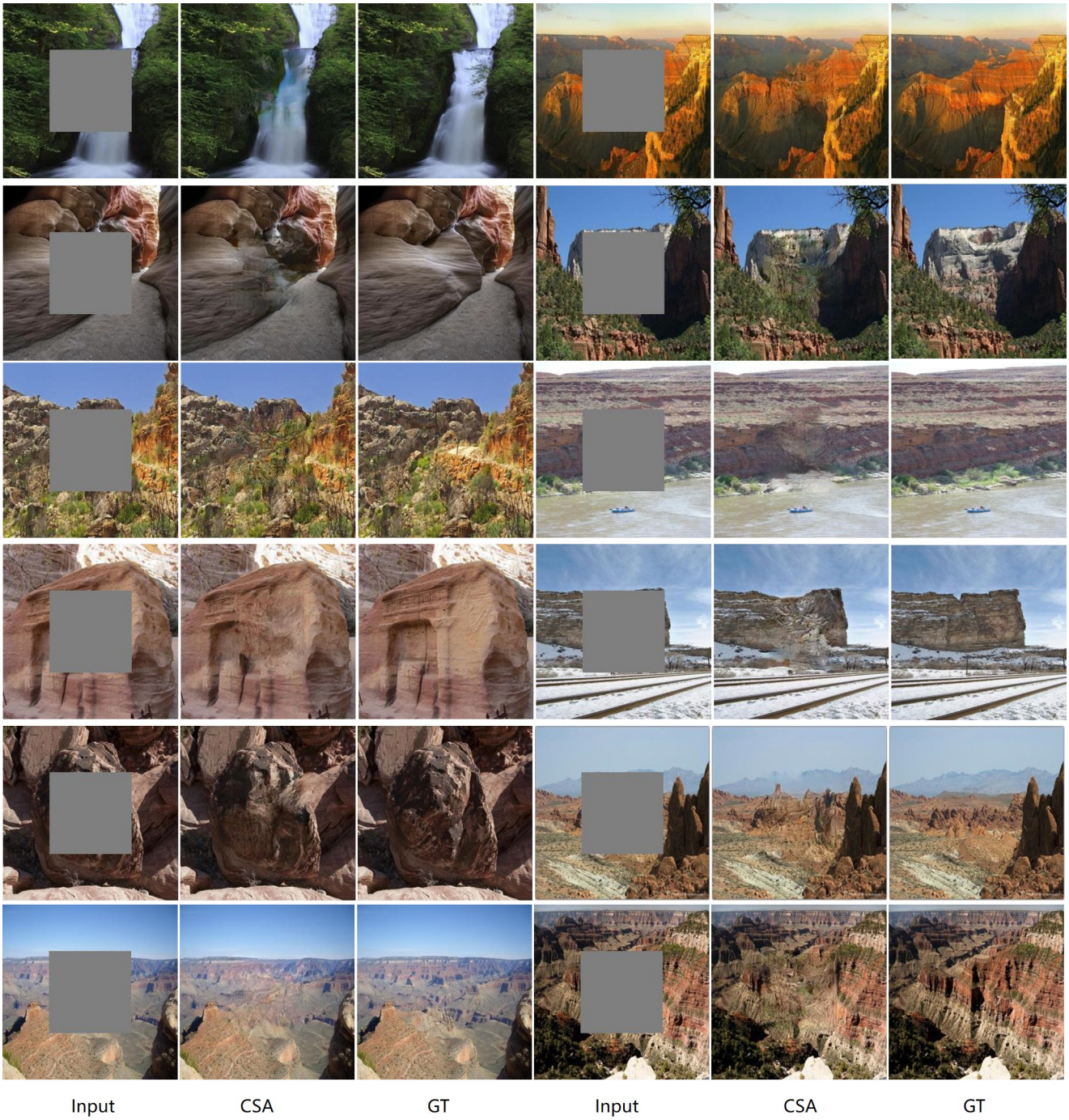

Figure 8. More results on Place2 with centering masks.

The architecture of refinement network
[Layer 1] Conv. (3, 3, 64), stride=1, padding=1;
[Layer 2] LReLU; Conv. (4, 4, 64), stride=2, dilation=2, padding=3; IN; LReLU; Conv. (3, 3, 128), stride=1, padding=1; IN;
[Layer 3] LReLU; Conv. (4, 4, 128), stride=2, dilation=2, padding=3; IN; LReLU; Conv. (3, 3, 256), stride=1, padding=1; IN;
[Layer 4] LReLU; Conv. (4, 4, 256), stride=2, dilation=2, padding=3; IN; LReLU; Conv. (3, 3, 512), stride=1, padding=1; **CSA**; IN;
[Layer 5] LReLU; Conv. (4, 4, 512), stride=2, dilation=2, padding=3; IN; LReLU; Conv. (3, 3, 512), stride=1, padding=1; IN;
[Layer 6] LReLU; Conv. (4, 4, 512), stride=2, dilation=2, padding=3; IN; LReLU; Conv. (3, 3, 512), stride=1, padding=1; IN;
[Layer 7] LReLU; Conv. (4, 4, 512), stride=2, dilation=2, padding=3; IN; LReLU; Conv. (3, 3, 512), stride=1, padding=1; IN;
[Layer 8] LReLU; Conv. (4, 4, 512), stride=2, dilation=2, padding=3; IN; LReLU; Conv. (3, 3, 512), stride=1, padding=1; IN;
[Layer 9] LReLU; Conv. (4, 4, 512), stride=2, padding=1;
[Layer 10] ReLU; DeConv. (4, 4, 512), stride=2, padding=1; IN; Concatenate(Layer 10, Layer 8);
[Layer 11] ReLU; DeConv. (3, 3, 512), stride=1, padding=1; IN; ; ReLU; DeConv. (4, 4, 512), stride=2, padding=1; IN; Concatenate(Layer 11, Layer 7);
[Layer 12] ReLU; DeConv. (3, 3, 512), stride=1, padding=1; IN; ReLU; DeConv. (4, 4, 512), stride=2, padding=1; IN; Concatenate(Layer 12, Layer 6);
[Layer 13]ReLU; DeConv. (3, 3, 512), stride=1, padding=1; IN; ReLU; DeConv. (4, 4, 512), stride=2, padding=1; IN; Concatenate(Layer 13, Layer 5);
[Layer 14] ReLU; DeConv. (3, 3, 512), stride=1, padding=1; IN; ReLU; DeConv. (4, 4, 512), stride=2, padding=1; IN; Concatenate(Layer 14, Layer 4);
[Layer 15] ReLU; DeConv. (3, 3, 256), stride=1, padding=1; IN; ReLU; DeConv. (4, 4, 256), stride=2, padding=1; IN; Concatenate(Layer 15, Layer 3);
[Layer 16] ReLU; DeConv. (3, 3, 128), stride=1, padding=1; IN; ReLU; DeConv. (4, 4, 128), stride=2, padding=1; IN; Concatenate(Layer 16, Layer 2);
[Layer 17] ReLU; DeConv. (3, 3, 64), stride=1, padding=1; IN; ReLU; DeConv. (4, 4, 64), stride=2, padding=1; IN; Concatenate(Layer 17, Layer 1);
[Layer 18] ReLU; DeConv. (3, 3, 64), stride=1, padding=1;

Table 10. The architecture of the refinement network. IN represents InstanceNorm and LReLU donates leaky ReLU with the slope of 0.2.