Supplementary Materials for "ShowFace: Coordinated Face Inpainting with Memory Disentangled Refinement Networks"

Zhuojie Wu[1]
zhuojiewu@bupt.edu.cn
Xingqun Qi[1]
xingqunqi@gmail.com
Zijian Wang[1]
wangzijianbupt@bupt.edu.cn
Wanting Zhou[1]
wanting.zhou@bupt.edu.cn
Kun Yuan[2]
yuankun03@kuaishou.com
Muyi Sun[3,] ✉
muyi.sun@cripac.ia.ac.cn
Zhenan Sun[3]
znsun@nlpr.ia.ac.cn

[1] School of Artificial Intelligence, Beijing University of Posts and Telecommunications (BUPT), Beijing, China

[2] Kuaishou Technology, Shenzhen, China

[3] Center for Research on Intelligent Perception and Computing (CRIPAC), National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China

Abstract

In the supplementary materials, we first introduce the detailed network architectures of the **coarse network, guided refinement network, and the reconstruction network of the InCo^2 loss**. Then, we give the detailed of the objective functions and training algorithm. Meanwhile, we introduce the ablation study of memory design. Finally, we illustrate more qualitative comparisons and visual results.

1 Detailed Network Architectures

The detailed architecture of the **coarse network** is shown in **Table 1**. The **ec** and **dc** represent the encoder and decoder respectively. In the coarse network, vanilla convolution is replaced by partial convolution[4]. BN denotes Batch Normalization, Act indicates the type of non-linear layer, ReLU denotes ReLU non-linear activation, LReLU indicates the Leaky ReLU activation with the slope of 0.2. The last convolutional layer employs a Tanh non-linear activation function. **Table 2** and **Table 3** illustrate the architectures of the **guided refinement network** and **reconstruction network of the InCo^2 loss**, respectively. The Gated Resblock is shown in **Figure 1**. In the guided refinement network, we employ SPADE [5] to fuse the

✉ Muyi Sun is the corresponding author.

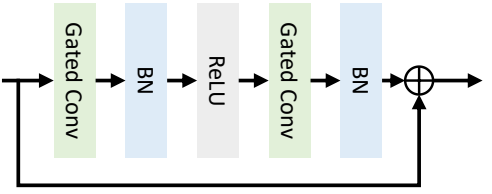

Figure 1: Illustration of the Gated Resblock in the guided refinement network. In each Gated Resblock, there are two Gated Convlution, two BN and one ReLU layers.

features after Mask-Region Enhanced Module (MREM). The SPADE Resblock is illustrated in **Figure 2**.

For the memory $\mathbb{M}$, we set the number of slots in each semantic-aware memory block as 128, and the dimension of each memory slot as 256. The memory consists of 14 semantic-aware memory blocks. In practice, the decay rate is set to 0.999 in Eq.3 (**in the main paper**).

Table 1: The architecture of the coarse network.

Layer	Settings	BN	Act	Input	Output
ec_1	7×7, 64	N	ReLU	$\boldsymbol{I}_{in}, \boldsymbol{M}$	$\boldsymbol{F}_{ec\_1}$
ec_2	5×5, 128	Y	ReLU	$\boldsymbol{F}_{ec\_1}$	$\boldsymbol{F}_{ec\_2}$
ec_3	5×5, 256	Y	ReLU	$\boldsymbol{F}_{ec\_2}$	$\boldsymbol{F}_{ec\_3}$
ec_4	3×3, 512	Y	ReLU	$\boldsymbol{F}_{ec\_3}$	$\boldsymbol{F}_{ec\_4}$
ec_5	3×3, 512	Y	ReLU	$\boldsymbol{F}_{ec\_4}$	$\boldsymbol{F}_{ec\_5}$
ec_6	3×3, 512	Y	ReLU	$\boldsymbol{F}_{ec\_5}$	$\boldsymbol{F}_{ec\_6}$
ec_7	3×3, 512	Y	ReLU	$\boldsymbol{F}_{ec\_6}$	$\boldsymbol{F}_{ec\_7}$
dc_1	3×3, 512	Y	LReLU	$\boldsymbol{F}_{ec\_7}, \boldsymbol{F}_{ec\_6}$	$\boldsymbol{F}_{dc\_1}$
dc_2	3×3, 512	Y	LReLU	$\boldsymbol{F}_{dc\_1}, \boldsymbol{F}_{ec\_5}$	$\boldsymbol{F}_{dc\_2}$
dc_3	3×3, 512	Y	LReLU	$\boldsymbol{F}_{dc\_2}, \boldsymbol{F}_{ec\_4}$	$\boldsymbol{F}_{dc\_3}$
dc_4	3×3, 256	Y	LReLU	$\boldsymbol{F}_{dc\_3}, \boldsymbol{F}_{ec\_3}$	$\boldsymbol{F}_{dc\_4}$
dc_5	3×3, 128	Y	LReLU	$\boldsymbol{F}_{dc\_4}, \boldsymbol{F}_{ec\_2}$	$\boldsymbol{F}_{dc\_5}$
dc_6	3×3, 64	Y	LReLU	$\boldsymbol{F}_{dc\_5}, \boldsymbol{F}_{ec\_1}$	$\boldsymbol{F}_{dc\_6}$
dc_7	3×3, 3	N	Tanh	$\boldsymbol{F}_{dc\_6}, \boldsymbol{I}_{in}$	$\boldsymbol{I}_{out}$

Table 2: The architecture of the guided refinement network.

Layer	Settings	Stride	Norm	Act
Gated Conv	3×3, 64	2	BN	ReLU
Gated Conv	3×3, 128	2	BN	ReLU
Gated Conv	3×3, 256	2	BN	ReLU
Gated Resblock	3×3, 256 3×3, 256	-	BN	ReLU
Gated Resblock	3×3, 256 3×3, 256	-	BN	ReLU
Gated Resblock	3×3, 256 3×3, 256	-	BN	ReLU
Gated Resblock	3×3, 256 3×3, 256	-	BN	ReLU
Upsample	2	-	-	-
SPADE Resblock	3×3, 256 3×3, 256	-	-	LReLU
Upsample	2	-	-	-
SPADE Resblock	3×3, 128 3×3, 128	-	-	LReLU
Upsample	2	-	-	-
SPADE Resblock	3×3, 64 3×3, 64	-	-	LReLU
Out_Conv	3×3, 3	1	-	Tanh

Table 3: The architecture of the reconstruction network for InCo^2 loss.

Encoder	Settings	BN	Act	**Decoder**	Settings	BN	Act
ResBlock	3×3, 64 3×3, 64	Y	LReLU	Upsample	2	-	-
MaxPool	2	-	-	ResBlock	3×3, 256 3×3, 256	Y	LReLU
ResBlock	3×3, 128 3×3, 128	Y	LReLU	Upsample	2	-	-
MaxPool	2	-	-	ResBlock	3×3, 128 3×3, 128	Y	LReLU
ResBlock	3×3, 256 3×3, 256	Y	LReLU	Upsample	2	-	-
MaxPool	2	-	-	ResBlock	3×3, 64 3×3, 64	Y	LReLU
ResBlock	3×3, 512 3×3, 512	Y	LReLU	Conv	3×3, 3	N	Tanh

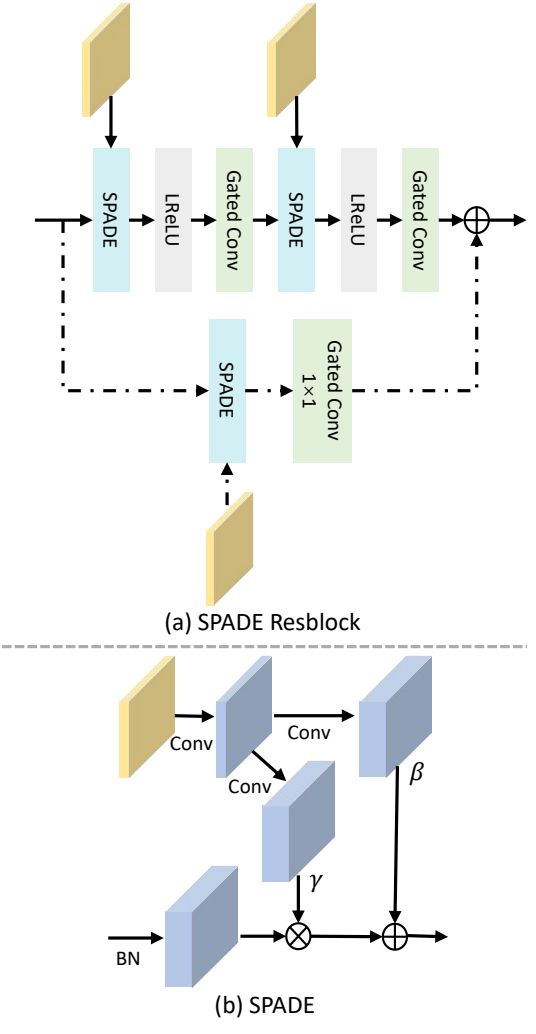

Figure 2: Illustration of the SPADE Resblock in the guided refinement network. The dash line denotes that the branch with the cascaded SPADE and Gated 1×1 convolution are used only when the channel number of the input does not equal to the output.

2 Objective Functions

In this section, we present the specific objective functions of our method, which could not be illustrated in the main paper due to the layout restrictions.

In the following, we first introduce the semantic loss [3] appropriately applied in our method. Then, we illustrate the reconstruction loss, perceptual loss, style loss, adversarial loss and total variation loss, inherited from the previous generation methods[4, 5, 7, 9].

Semantic Loss. Since the semantic map S obtained from coarse face P may bring some errors, we employ semantic loss $\mathcal{L}_{sem}$ to refine their influences, which computes the Cross Entropy of parsing maps between the completed image $\hat{I}$ and ground truth I_{gt}.

$$\mathcal{L}_{sem} = \mathbb{CE}(\mathbb{P}(I_{gt}), \mathbb{P}(\hat{I})) \tag{1}$$

where $\mathbb{P}$ denotes the inference process of face parser.

Reconstruction Loss. The reconstruction loss $\mathcal{L}_{rec}$ calculates the $L1$ distance between the completed image $\hat{I}$ and ground truth I_{gt}, which encourages the $\hat{I}$ to be similar with I_{gt} at the pixel level.

$$\mathcal{L}_{rec} = \left\| \hat{I} - I_{gt} \right\|_1 \tag{2}$$

Perceptual Consistency Loss. The perceptual loss $\mathcal{L}_{perc}$ measures the $L1$ distance between $\hat{I}$ and I_{gt} in the feature space, which penalizes the perceptual and semantic discrepancy.

$$\mathcal{L}_{perc} = \sum_i \left\| \phi_i(\hat{I}) - \phi_i(I_{gt}) \right\|_1 \tag{3}$$

where $\phi_i(\cdot)$ denotes the activation of the ith layer from the pre-trained VGG-19 network [8].

Style Consistency Loss. The style loss $\mathcal{L}_{style}$ calculates the statistical errors between the features of $\hat{I}$ and I_{gt} to constrain the style consistency.

$$\mathcal{L}_{style} = \sum_i \left\| \mathbb{G}(\phi_i(\hat{I})) - \mathbb{G}(\phi_i(I_{gt})) \right\|_1 \tag{4}$$

where $\mathbb{G}$ denotes the Gram matrix.

Adversarial Loss. We employ the discriminator D in PatchGAN [1] to match distributions between $\hat{I}$ and I_{gt}, which promotes the generator to generate realistic images.

$$\mathcal{L}_{adv} = \mathbb{E}_{I_{gt}}[log(D(I_{gt}))] + \mathbb{E}_{\hat{I}}[log(1 - D(\hat{I}))] \tag{5}$$

Total Variation Loss. We adopt the total variation loss $\mathcal{L}_{tv}$ to smooth the completed image $\hat{I}$.

$$\mathcal{L}_{tv} = \left\| \hat{I} \right\|_{tv} \tag{6}$$

In summary, the overall objective function can be formulated as:

$$\begin{aligned} \mathcal{L}_{total} =& \lambda_1 \mathcal{L}_{InCo^2} + \lambda_2 \mathcal{L}_{sem} + \lambda_3 \mathcal{L}_{rec} + \lambda_4 \mathcal{L}_{perc} \\ &+ \lambda_5 \mathcal{L}_{style} + \lambda_6 \mathcal{L}_{adv} + \lambda_7 \mathcal{L}_{tv} \end{aligned} \tag{7}$$

where $\lambda_{i,\{i=1,2,\ldots,7\}}$ are hyper-parameters to balance each item.

3 Training Algorithm

The pseudo code of MDRNets is shown in **Algorithm 1**. We denote the input image as $I_{in} = I_{gt} \otimes M$, the pre-trained coarse network as G_{coarse}, the face parser as G_{parser}, the reconstruction network as G_{re}, our MDRNets as $G_{MDRNets}$, and discriminator as D. The encoder of reconstruction network is denoted as D_{ec}. For face parser G_{parser}, we employ the open-source pre-trained face segmentation model BiseNet[10]. To begin with, G_{coarse} and G_{re} are trained with the training set of CelebA-HQ [2]. Then $G_{MDRNets}$ is trained on the corresponding training set. All models are optimized using Adam optimizer with β_1=0.9 and β_2=0.99. For the G_{coarse} and G_{re}, we fixed the learning rate of 2e-4 to train 100 epochs and 30 epochs, respectively. For the $G_{MDRNets}$, we firstly fix the learning rate of 2e-4 to train 35 epochs, and then linearly decay the learning rate to zero for last 10 epochs.

Algorithm 1 The pseudo code of MDRNets.

Input:

I_{gt}: Image data;
M: Mask data;
E: The number of epoch = 45;

Output:

$\hat{I}$: Completed image;
Step1: Image pre-processing;
Resized into 256 $\times$ 256;
Masked image $I_{in} = I_{gt} \otimes M$;
Step2: Network Initialization;
Initialize weights of G_{coarse}, G_{re}, $G_{MDRNets}$, D;
$w = \{w_{G_{coarse}}, w_{G_{re}}, w_{G_{MDRNets}}, w_D\} = 0$;
Step3: Pre-train G_{coarse} and G_{re};
$P = G_{coarse}(I_{in}, M)$;
$\hat{I_{gt}} = G_{re}(I_{gt})$;
Step4: Network Training;
Load $w_{G_{coarse}}$, $w_{G_{parser}}$, and D_{ec};
for $t = 0$ to E **do**
    Compute the prediction $\hat{I} = G_{MDRNets}(I_{in}, M)$;
    Compute the $\mathcal{L}_{adv}^{D}$ loss in Eq.13;
    Update D;
    Compute the $\mathcal{L}_{total}$ loss in Eq.15;
    Update $G_{MDRNets}$;
end for

4 Ablation Study of the Memory Design

The number of slots in each memory block is a question worth considering. That is, how many slots do we need to store the latent vectors for each semantic ?

In the Table 4, we present the results for different m. We can clearly see that the $m = 128$ could obtain the best results in the disentangled memory. Meanwhile, we conduct a comparison on whether the memory needs to be disentangled. According to the results, **the disentangled memory with** $m = 128$ obtains better performance for face inpainting.

Table 4: The evaluation results of Memory Design Ablation. m denotes the slot number for each memory block. Non-Disentangled denotes using non-disentangled memory.

Metrics	Mask Ratio	Disentangled					Non-Disentangled
		m=32	m=64	m=128	m=256	m=512	same volume as m=128
$L1(\%)\downarrow$	1-20%	0.627	**0.584**	0.585	0.600	0.618	0.662
FID $\downarrow$		3.194	**2.308**	2.369	2.670	2.927	5.244
PSNR $\uparrow$		35.214	35.718	**35.772**	35.441	35.397	34.987
SSIM $\uparrow$		0.963	0.968	**0.968**	0.966	0.965	0.962
$L1(\%)\downarrow$	20-40%	1.547	1.452	**1.451**	1.494	1.532	1.639
FID $\downarrow$		8.017	6.421	**6.410**	7.741	7.626	15.452
PSNR $\uparrow$		28.344	28.593	**28.669**	28.391	28.452	28.339
SSIM $\uparrow$		0.891	0.898	**0.900**	0.893	0.893	0.893
$L1(\%)\downarrow$	40-60%	3.046	2.951	**2.937**	3.000	3.037	3.177
FID $\downarrow$		14.124	12.523	**12.086**	15.557	13.712	24.495
PSNR $\uparrow$		24.017	24.079	24.177	23.996	**24.901**	24.152
SSIM $\uparrow$		0.791	0.796	0.800	0.791	0.793	0.800

5 Additional Qualitative Comparisons

In **Figure 3** and **Figure 4**, we present qualitative comparisons at different mask ratios on the CelebA-HQ and FFHQ datasets. $1\% \sim 20\%$, $20\% \sim 40\%$ and $40\% \sim 60\%$ represent the three types of different mask ratios respectively. In **Figure 3** and **Figure 4**, each two-column represents the comparisons of one mask ratio. As can be seen from these figures, all the methods can achieve satisfactory results when the mask ratio is small with $1\% \sim 20\%$. With the increase of the mask ratio, the previous methods suffer from artifacts, blurred structures, texture detail losses, and especially the lack of face coordination. In contrast, our method achieves satisfactory results with semantic-level coordination.

Further, more qualitative comparisons with large mask on the CelebA-HQ and FFHQ datasets are shown in **Figure 5** and **Figure 6**. It can be seen that our method generates more coordinated, textured, and realistic results than its counterparts.

6 Additional Visual Results

Figure 7 shows more visual results achieved by our method on the CelebA-HQ and FFHQ datasets. It can be observed that our method could generate desirable results with high image quality as the ground-truth faces. As an ill-posed problem, our MDRNets could generate plausible and coordinated contents.

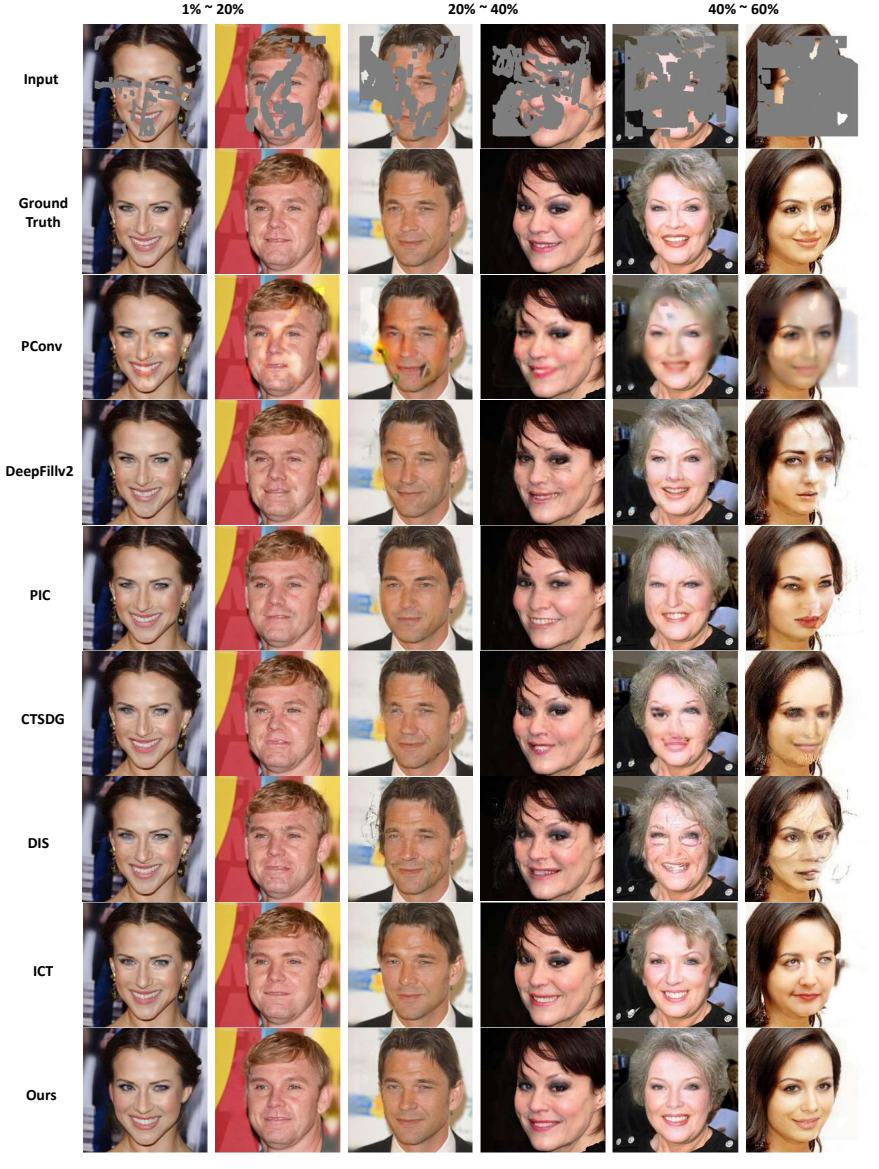

Figure 3: The qualitative comparisons with different mask ratios on the CelebA-HQ dataset. $1\% \sim 20\%$, $20\% \sim 40\%$ and $40\% \sim 60\%$ represent different mask ratios respectively, and each two-column represents a mask ratio. Zoom in for better details

Figure 4: The qualitative comparisons with different mask ratios on the FFHQ dataset. $1\% \sim 20\%$, $20\% \sim 40\%$ and $40\% \sim 60\%$ represent different mask ratios respectively, and each two-column represents a mask ratio. Zoom in for better details

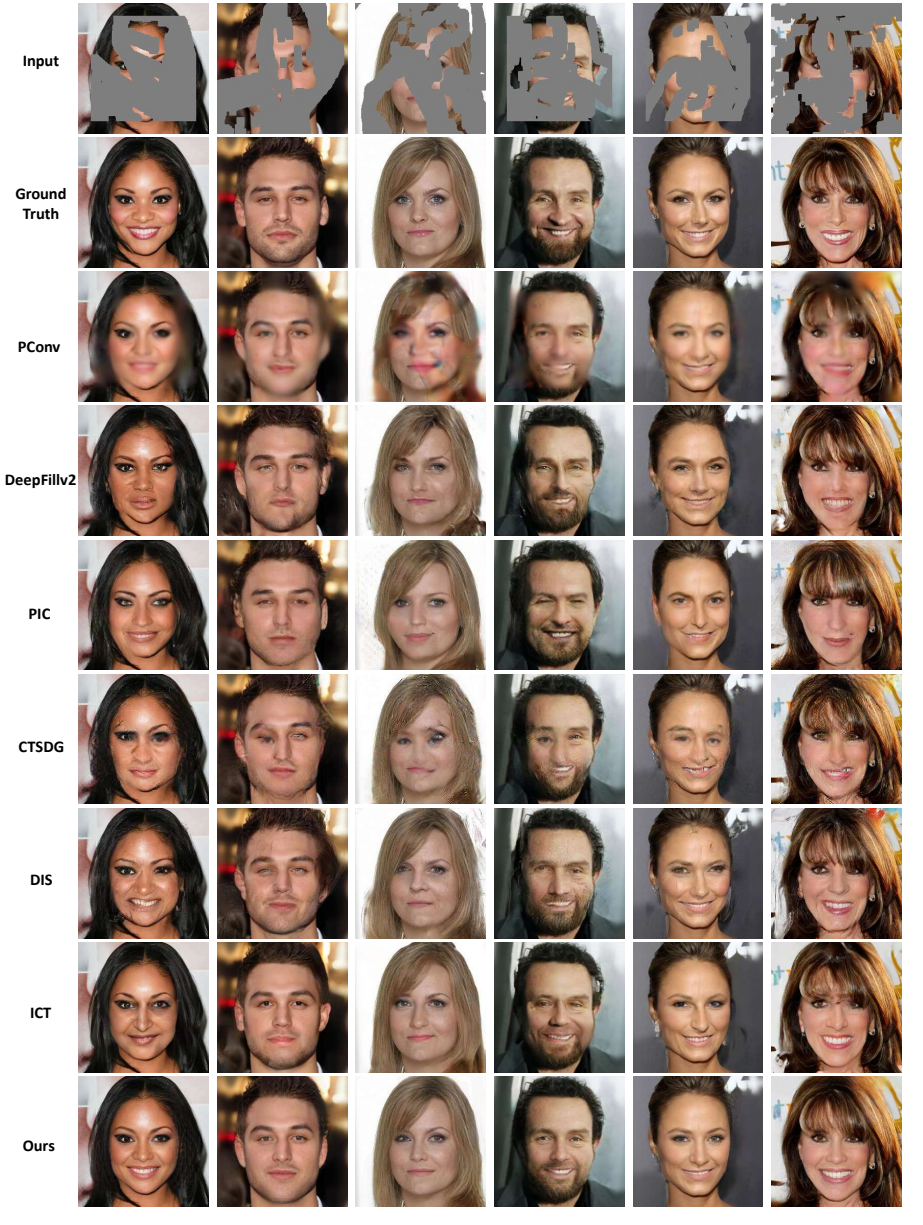

Figure 5: The qualitative comparisons on the CelebA-HQ dataset. Zoom in for better details

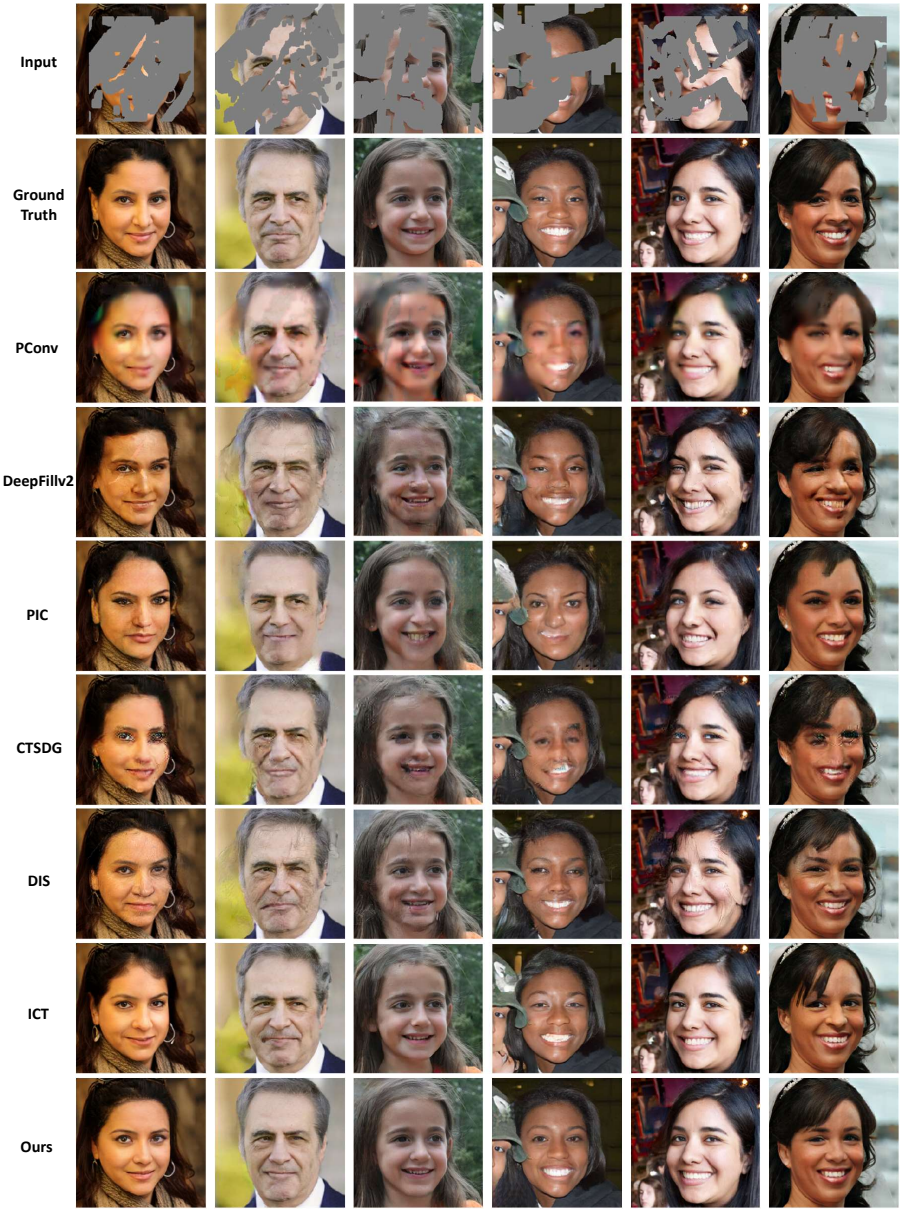

Figure 6: The qualitative comparisons on the FFHQ dataset. Zoom in for better details

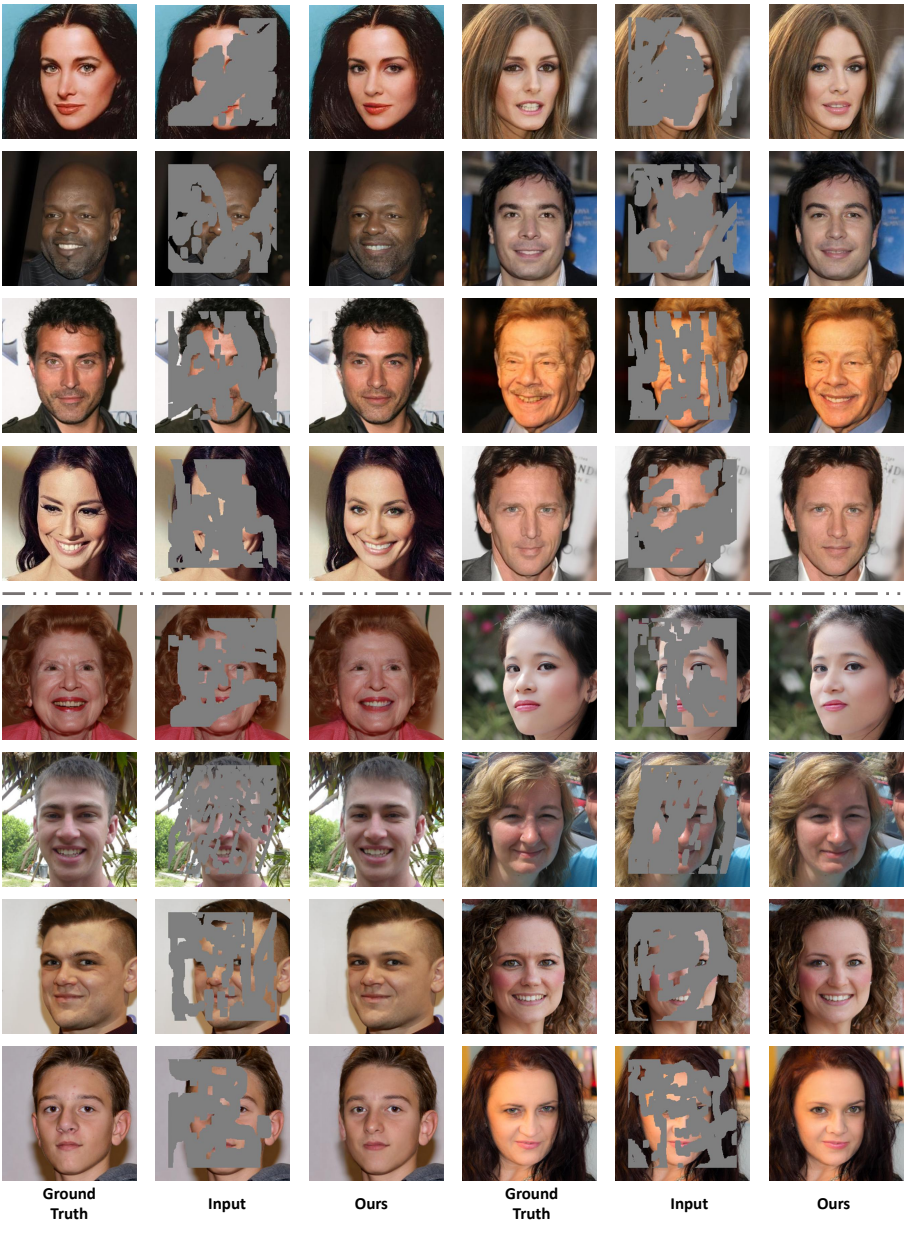

Figure 7: More visual results on the CelebA-HQ and FFHQ datasets. The first four rows are from the CelebA-HQ dataset, and the last four rows are from the FFHQ dataset. Our method could generate high-quality faces as the ground-truth faces. Zoom in for better details.

References

[1] Phillip Isola, Jun-Yan Zhu, and et al. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017.

[2] Tero Karras, Timo Aila, and et al. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[3] Yijun Li, Sifei Liu, and et al. Generative face completion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3911–3919, 2017.

[4] Guilin Liu, Fitsum A Reda, and et al. Image inpainting for irregular holes using partial convolutions. In *European Conference on Computer Vision*, pages 85–100, 2018.

[5] Hongyu Liu, Ziyu Wan, and et al. Pd-gan: Probabilistic diverse gan for image inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9371–9381, 2021.

[6] Taesung Park, Ming-Yu Liu, and et al. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

[7] Jialun Peng, Dong Liu, and et al. Generating diverse structure for image inpainting with hierarchical vq-vae. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10775–10784, 2021.

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[9] Ziyu Wan, Jingbo Zhang, and et al. High-fidelity pluralistic image completion with transformers. In *IEEE International Conference on Computer Vision*, pages 4692–4701, 2021.

[10] Changqian Yu, Jingbo Wang, and et al. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *European Conference on Computer Vision*, pages 325–341, 2018.