Supplementary Material

Jie Cao[1,3], Huaibo Huang[1,3], Yi Li[1,3], Ran He[1,2,3 *], and Zhenan Sun[1,2,3]

[1] Center for Research on Intelligent Perception and Computing, NLPR, CASIA
[2] Center for Excellence in Brain Science and Intelligence Technology, CAS
[3] School of Artificial Intelligence, University of Chinese Academy of Sciences
{jie.cao, huaibo.huang, yi.li}@cripac.ia.ac.cn, {rhe, znsun}@nlpr.ia.ac.cn

1 Network Architecture

In this work, we implement our method based on Pytorch. In our experiments, the batch size is 16 and the image size is $[3 \times 256 \times 256]$. Given a n-dimensional attribute vector, we apply spatial replication to change its shape to $[n \times 256 \times 256]$. The inputs of the generator and the discriminator are both the concatenations of images and attribute vectors. Hence, the input tensor shape is $[(3+n) \times 256 \times 256]$. The network architectures of the generator and the discriminator are reported in Table 1 and Table 2, respectively.

2 Additional Results

We report additional visual comparisons in this section. The single facial attribute transfer results are shown in Figures 1 and 2. The multiple facial attribute transfer results are shown in Figures 3, 4, and 5. For multiple facial attribute transfer, we also visualize the residual heatmaps between the inputs and the outputs to reveal the modified regions of images. We report more season transfer results in Figure 6 and more sketch&photo transfer results in Figure 7. Furthermore, we report the visual comparisons of our variations in the ablation study in Figure 8. Figures 9 and 10 show our results on facial attribute interpolation. We can control the intensity of the transferred attribute by interpolating the attribute vector.

These additional visual results further demonstrate the superiority of our approach. The results of other methods are not satisfying in some specific challenging cases, although they can also address most of the easy cases. For instance, STGAN always produces unrealistic facial skin when altering *age*. AttGAN tends to make modifications minor than expected when altering *gender* or *beard*. In general, our approach produces most plausible results with finer texture details.

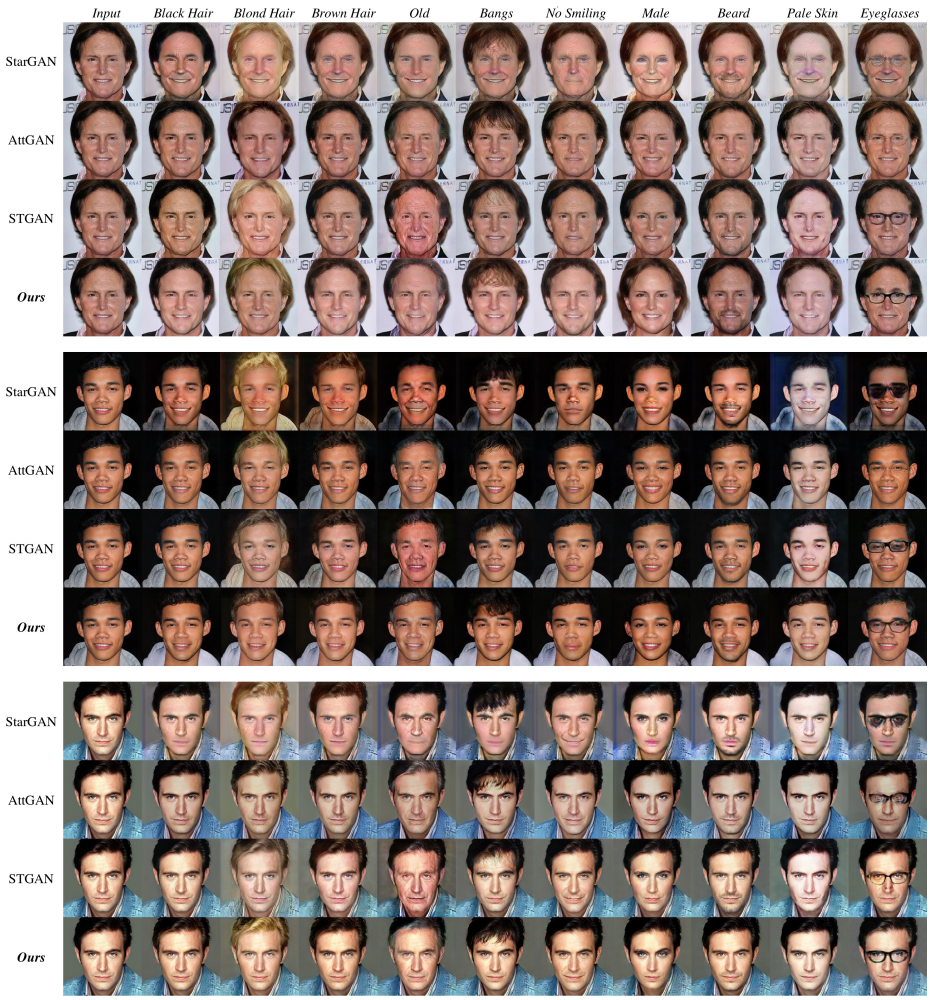

Fig. 1. Visual examples of single facial attribute editing results. From top to bottom, the rows are results of StarGAN, AttGAN, STGAN, and our INIT.

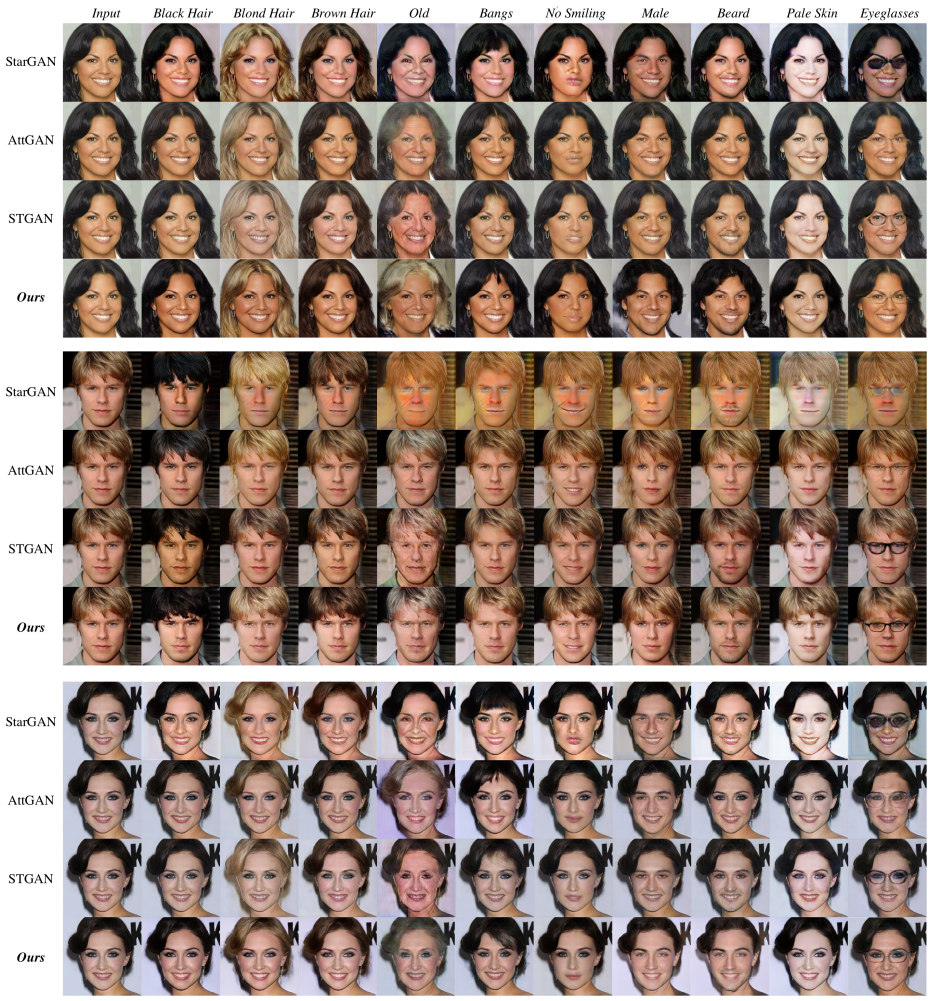

Fig. 2. Visual examples of single facial attribute editing results. From top to bottom, the rows are results of StarGAN, AttGAN, STGAN, and our INIT.

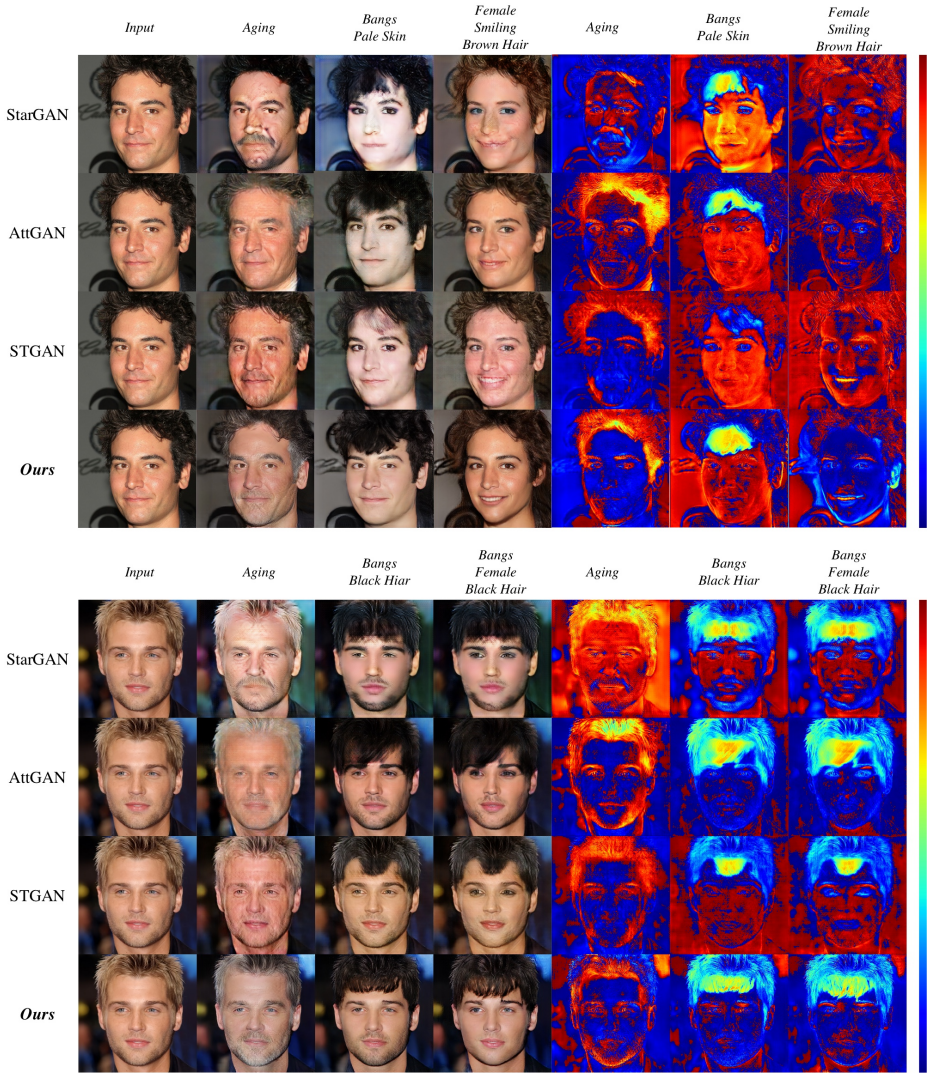

Fig. 3. Visual examples of multiple facial attribute editing results. The residual heat maps visualize the differences between the inputs and the outputs. Please zoom in for better visualization.

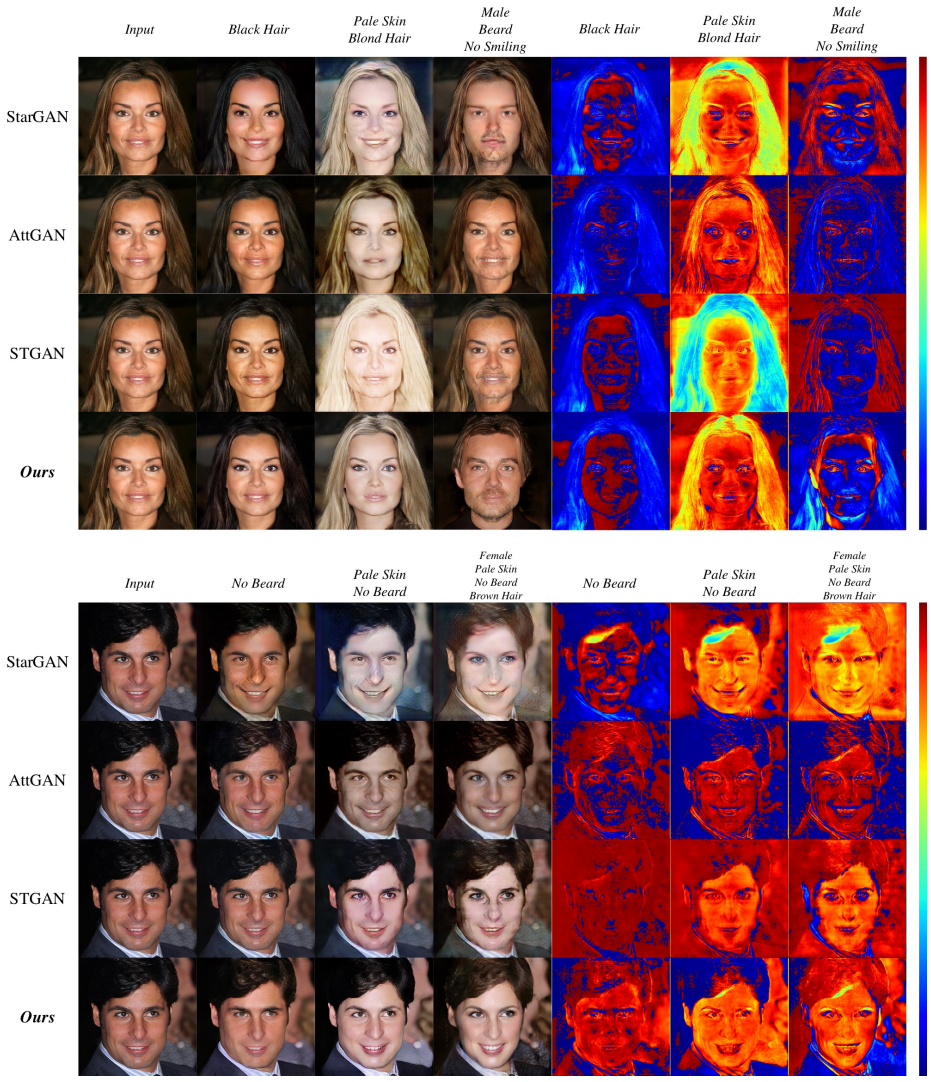

Fig. 4. Visual examples of multiple facial attribute editing results. The residual heat maps visualize the differences between the inputs and the outputs. Please zoom in for better visualization.

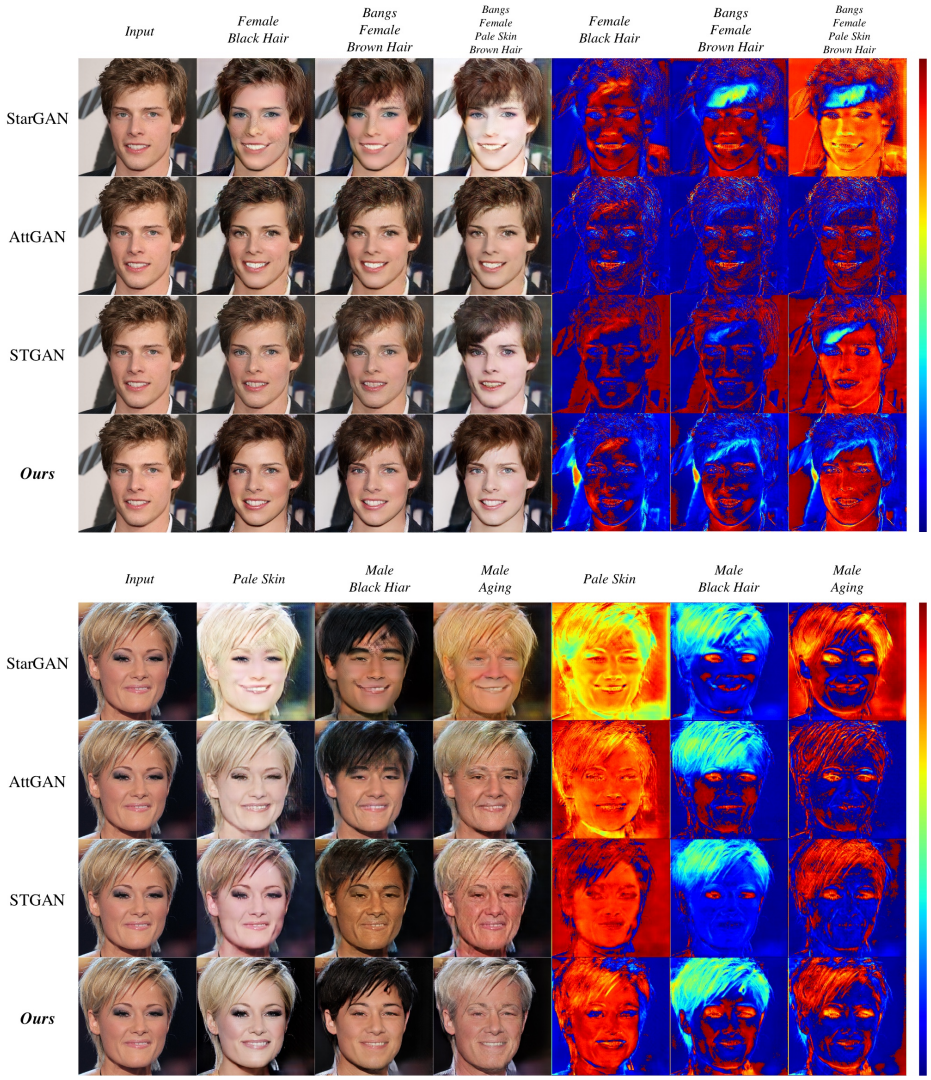

Fig. 5. Visual examples of multiple facial attribute editing results. The residual heat maps visualize the differences between the inputs and the outputs. Please zoom in for better visualization.

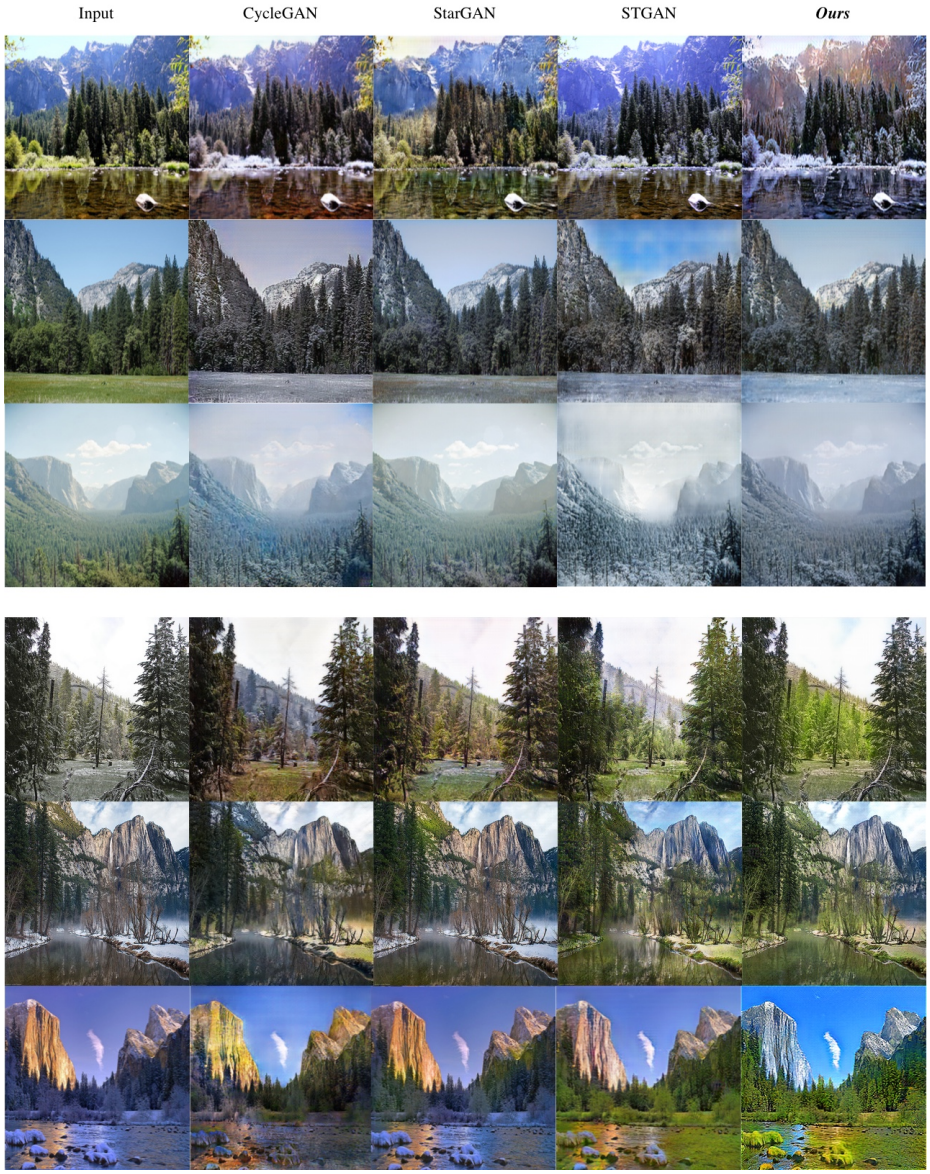

Fig. 6. Visual examples of season translation. The top three rows are *summer→winter*, and the bottom three rows are *winter→summer*.

Fig. 7. Visual examples of sketch&photo transfer. The top three rows are *sketch→shoes*, and the bottom three rows are *shoes→sketch*. The ground truths are placed on the upper left corners of the inputs.

Fig. 8. Facial attribute transfer result comparisons of our variations in ablation study. For each row, the input is placed on the low left corner of the output of variation (a). Our full model is equivalent variation (c).

Fig. 9. Facial attribute interpolation results. Our INIT can control the intensity of the transferred attribute by varying α. The interpolated attribute is *hair color*.

Fig. 10. Facial attribute interpolation results. Our INIT can control the intensity of the transferred attribute by varying α. From top to bottom, the interpolated attributes are *smiling*, *pale skin*, *male*, *beard*, and *eyeglasses*.

Layer Type	Output Shape	Parameter Number
Conv2d	[64,256,256]	40,832
InstanceNorm2d	[64,256,256]	128
ReLU	[64,256,256]	0
Conv2d	[128,128,128]	131,200
InstanceNorm2d	[128,128,128]	256
ReLU	[128,128,128]	0
Conv2d	[256,64,64]	524,544
InstanceNorm2d	[256,64,64]	512
ReLU	[256,64,64]	0
Conv2d	[256,64,64]	590,080
InstanceNorm2d	[256,64,64]	512
ReLU	[256,64,64]	0
Conv2d	[256,64,64]	590,080
InstanceNorm2d	[256,64,64]	512
Conv2d	[256,64,64]	590,080
InstanceNorm2d	[256,64,64]	512
ReLU	[256,64,64]	0
Conv2d	[256,64,64]	590,080
InstanceNorm2d	[256,64,64]	512
Conv2d	[256,64,64]	590,080
InstanceNorm2d	[256,64,64]	512
ReLU	[256,64,64]	0
Conv2d	[256,64,64]	590,080
InstanceNorm2d	[256,64,64]	512
Conv2d	[256,64,64]	590,080
InstanceNorm2d	[256,64,64]	512
ReLU	[256,64,64]	0
Conv2d	[256,64,64]	590,080
InstanceNorm2d	[256,64,64]	512
Conv2d	[256,64,64]	590,080
InstanceNorm2d	[256,64,64]	512
ReLU	[256,64,64]	0
Conv2d	[256,64,64]	590,080
InstanceNorm2d	[256,64,64]	512
Conv2d	[256,64,64]	590,080
InstanceNorm2d	[256,64,64]	512
ReLU	[256,64,64]	0
Conv2d	[256,64,64]	590,080
InstanceNorm2d	[256,64,64]	512
ConvTranspose2d	[128,128,128]	524,416
ReLU	[128,128,128]	0
ConvTranspose2d	[64,256,256]	131,136
ReLU	[64,256,256]	0
Conv2d	[3,256,256]	9,411
Sigmoid	[3,256,256]	0

Table 1. The network architecture of our generator.

Layer Type	Output Shape	Parameter Number
Conv2d	[64,128,128]	3,136
InstanceNorm2d	[64,128,128]	128
LeakyReLU	[64,128,128]	0
Conv2d	[128, 64, 64]	151,680
InstanceNorm2d	[128, 64, 64]	256
LeakyReLU	[128, 64, 64]	0
Conv2d	[256, 32, 32]	524,544
InstanceNorm2d	[256, 32, 32]	512
LeakyReLU	[256, 32, 32]	0
Conv2d	[512, 16, 16]	2,097,664
InstanceNorm2d	[512, 16, 16]	1,024
LeakyReLU	[512, 16, 16]	0
Conv2d	[1, 16, 16]	513

Table 2. The network architecture of our discriminator.