Adversarial Feedback Loop
—Supplementary—

Firas Shama Roey Mechrez Alon Shoshan Lihi Zelnik-Manor
Technion - Israel Institute of Technology
{shfiras@campus,roey@campus,shoshan@campus,lihi@ee}.technion.ac.il

1. Implementation Details

Simple 2D case The generator architecture is a sequence of four layers of fc (fully-connected)-$ReLU$, where the input is 2D points from a normal distribution. All hidden layers dimension is 512. The discriminator architecture is identical to the generator except in the last layer where the output is only a single scalar (real or fake). The feedback module is $fc - ReLU - fc$, with hidden layer dimension of 512. It is fed from the activation map of the first discriminator $ReLU$, and corrects the input activations of last fc layer in the generator. Since the generator has no batch-normalization, we refrain from using it. As a result we use $\alpha = 1$ at test-time.

The model is trained with WGAN-GP [2] adversarial loss, with $\lambda = 0.1$ and five critic (discriminator) training iterations per one for the generator.

Image generation on CIFAR-10 As mentioned previously, we adopt the same training scheme, objectives and parameters of each used method. In all the methods, we attached a single feedback module of $conv$–BN–$ReLU$–$conv - BN$, where $conv$ is a 3×3 convolution layer with the same number of feature maps as the input feature map coming from the discriminator and padding of one to conserve the spatial dimensions of the features.

Face generation latent space interpolation: Here we choose an arbitrary pair of input vectors and perform a linear interpolation between them (in input space), we feed every interpolation step to the network and observe how our AFL improves the final generation results, see Figure 1.

Super-resolution We used the official model of ESRAG [4] with a generator of 23 Residual-in-Residual Dense Block (RRDB) and kept the same training scheme & parameters. In order to improve the baseline results, we used the pre-trained generator and discriminator provided by the authors, to which we added four dual-input feedback modules. Each feedback module is $conv - ReLU - conv - ReLU - conv$, where $conv$ is a 3×3 convolutional layer. Note that we do not add a $ReLU$ activation on after the last layer, in order to allow propagation of negative correcting values. Table 1 describes how each feedback module is connected. Specifically, its inputs & target feature maps, and which up-scale is performed on the input channel of the module. Note that we adopted the same up-scale module used in the generator, which is a nearest-neighbor (NN) upscale followed by a $conv$ layer.

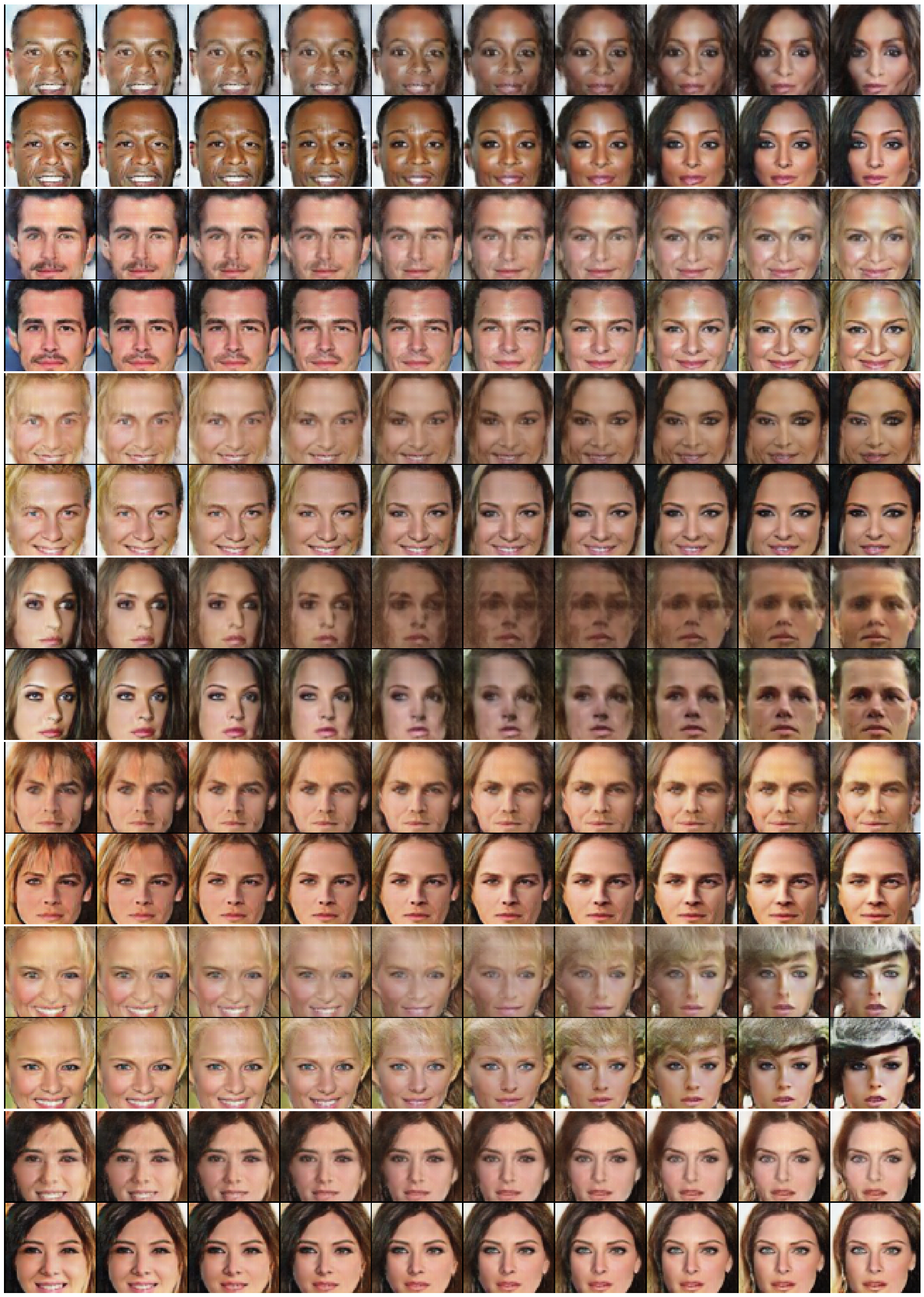


Figure 1: **Latent space interpolation:** Results of interpolation between two different input vectors. We compare DCGAN [3] baseline (odd rows) with ours, DCGAN+AFL (even rows).

feedback module index	Input A (in D)	Up-scale (for A)	Input B & Target (in G)
1	output of 9^{th} conv layer	$\times 4$	input of 13^{th} RRDB
2	output of 8^{th} conv layer	$\times 4$	input of 21^{st} RRDB
3	output of 7^{th} conv layer	$\times 2$	input of $\times 4$ upscale layer
4	output of 5^{th} conv layer	$\times 4$	output of $\times 4$ upscale layer

Figure 2: Inputs and output of each feedback module. We denote the inputs of the feedback module by A and B.

2. Additional Results

CelebA More results of our method compared with the baseline [3] exist in Figure 3. Additional results of the feedback switching pipeline, where we replace the input of the discriminator with a reference image, are shown in Figures 4, 5 and 6.

Super-resolution Additional results of the contribution of AFL in the super-resolution task are presented in Figures 8, 9, 10, 11 and 12 for images from PIRM challenge [1] data-set. In addition, an interesting result is shown in Figure 7, where stripes are corrected to the true direction and merged more naturally.

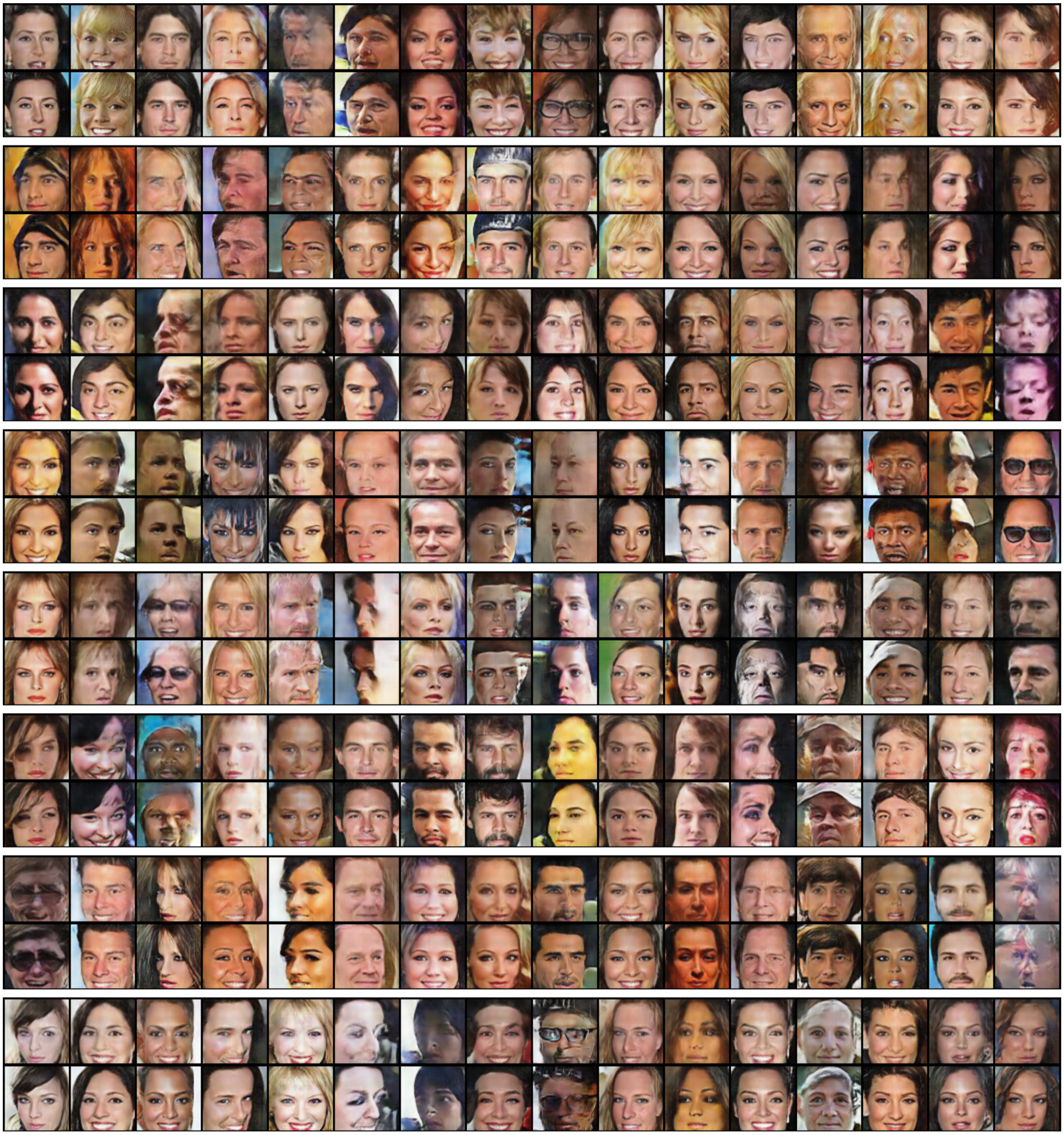


Figure 3: **Full batch results**. We compare DCGAN [3] baseline (odd rows) with ours, DCGAN+AFL (even rows).

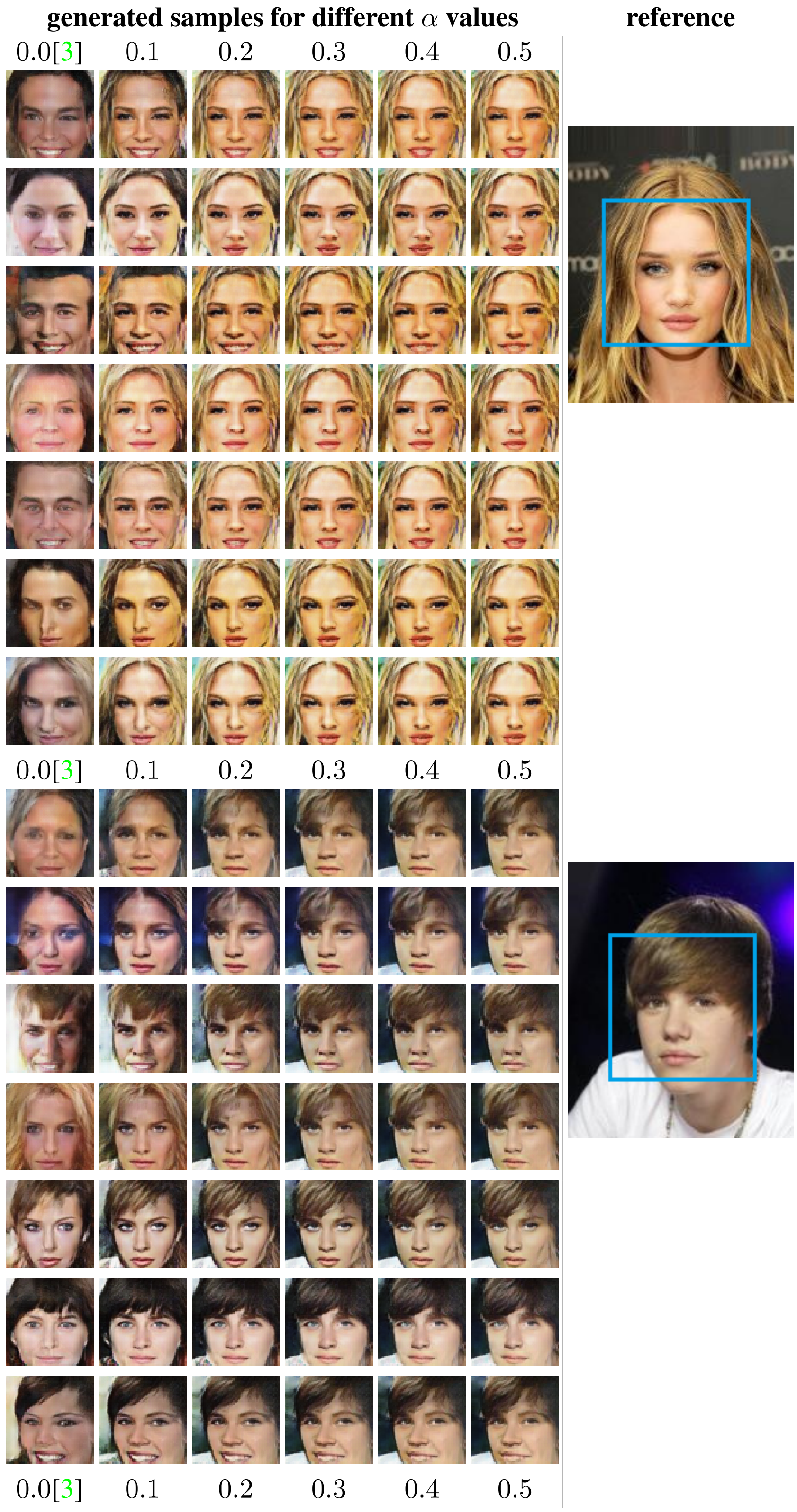


Figure 4: **Generation with reference:** More Results of using the feedback-switching-pipeline

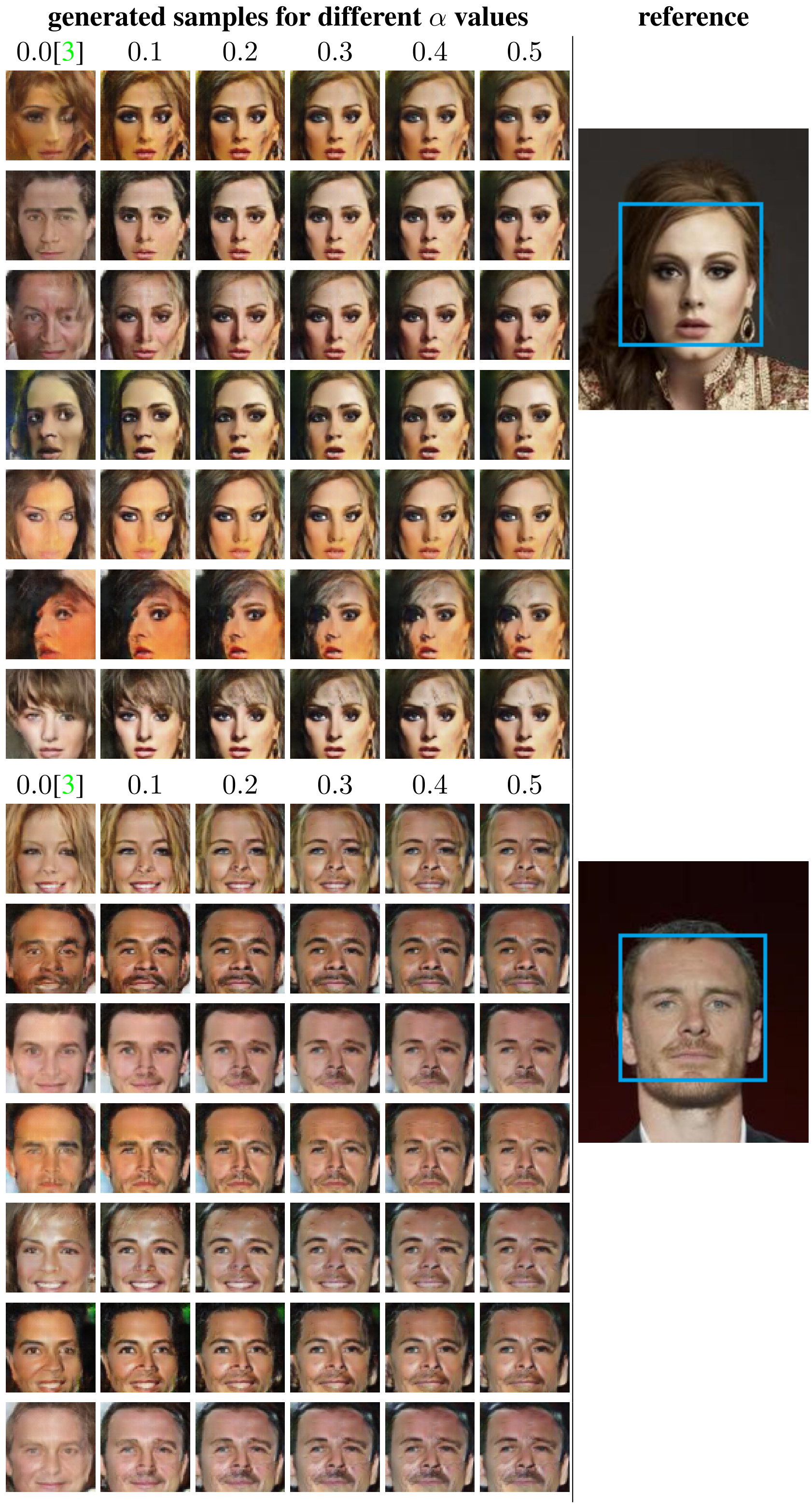


Figure 5: **Generation with reference:** More Results of using the feedback-switching-pipeline

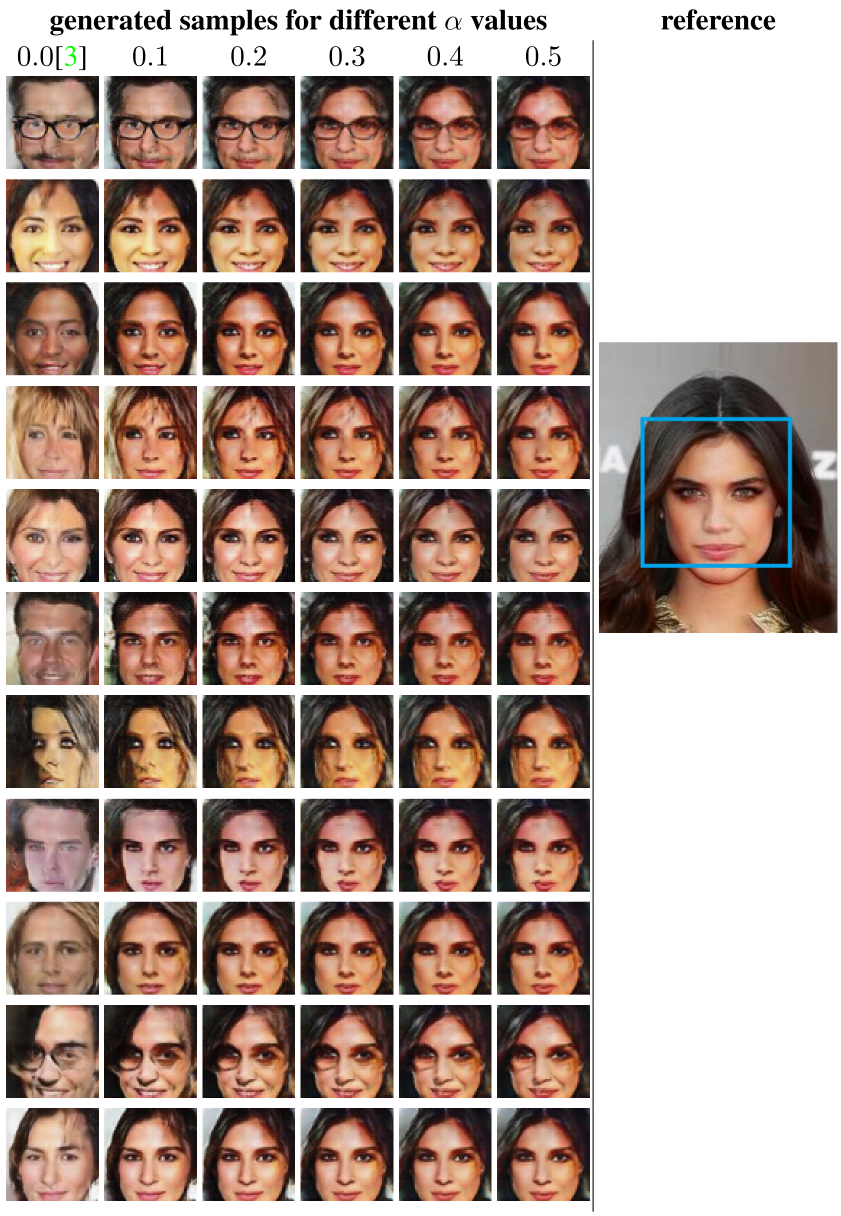
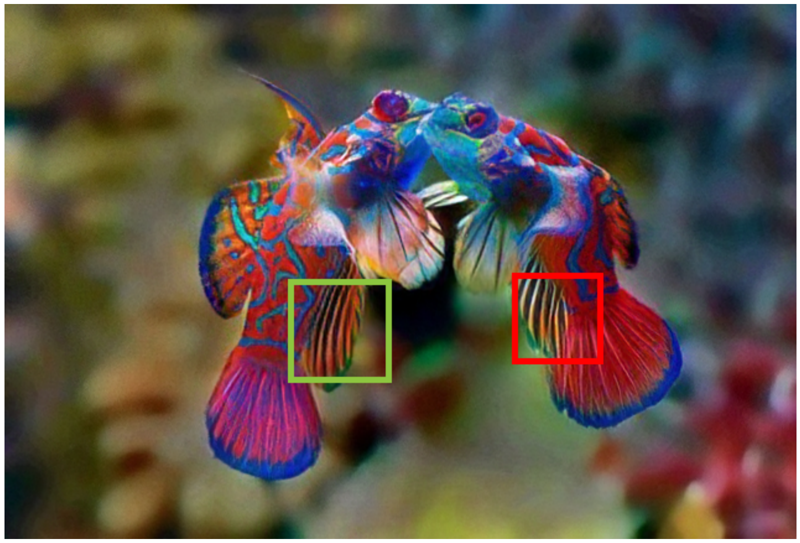
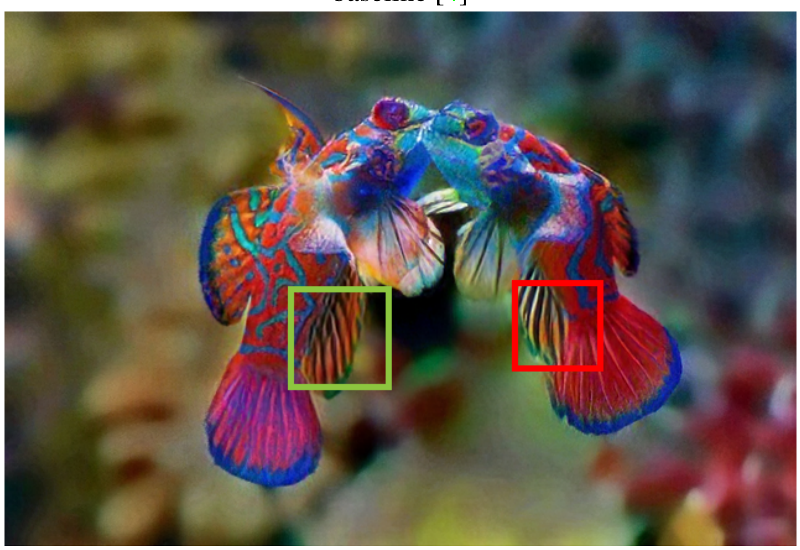


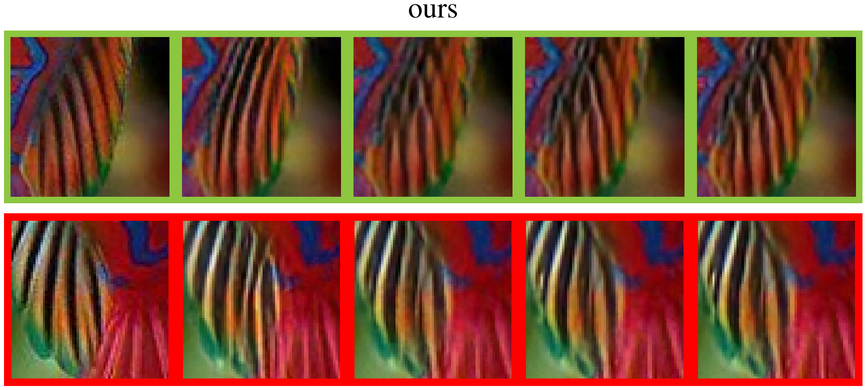
Figure 6: **Generation with reference:** More Results of using the feedback-switching-pipeline



baseline [4]



ours

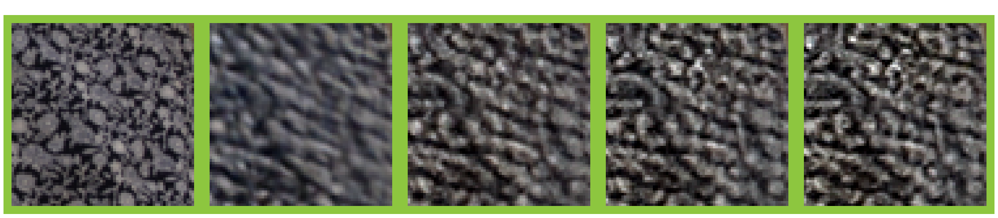


GT baseline [4] 1^{st} iter 2^{nd} iter 3^{rd} iter

Figure 7: Green patch: stripes corrected to the true direction through feedback iterations, red patch: stripes are merged together more naturally

baseline [4]

ours

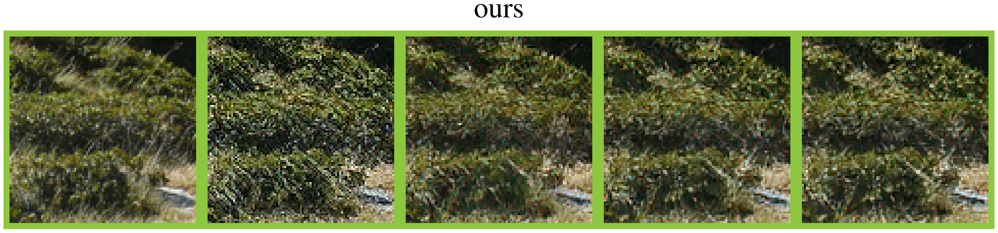


GT baseline [4] 1^{st} iter 2^{nd} iter 3^{rd} iter

Figure 8: Additional result of super-resolution task

baseline [4]

ours



GT | baseline [4] | 1^{st} iter | 2^{nd} iter | 3^{rd} iter

Figure 9: Additional result of super-resolution task

Figure 10: Additional result of super-resolution task

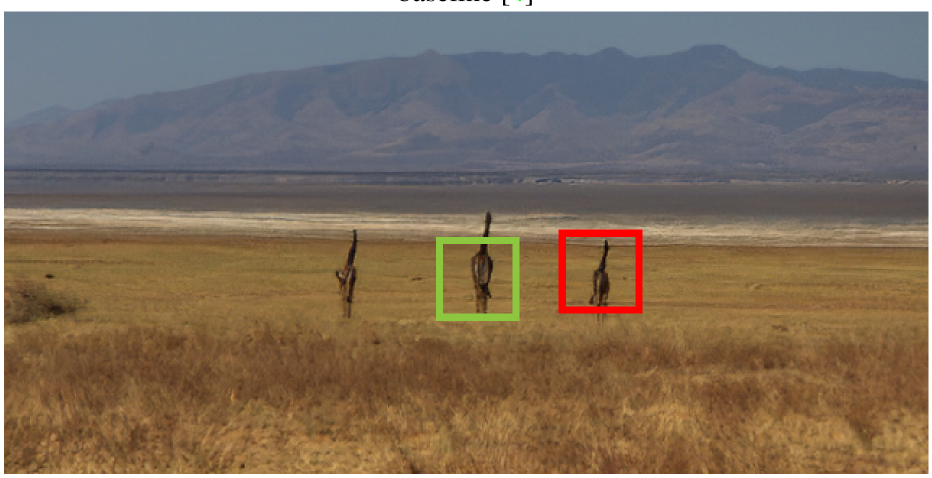
baseline [4]

ours

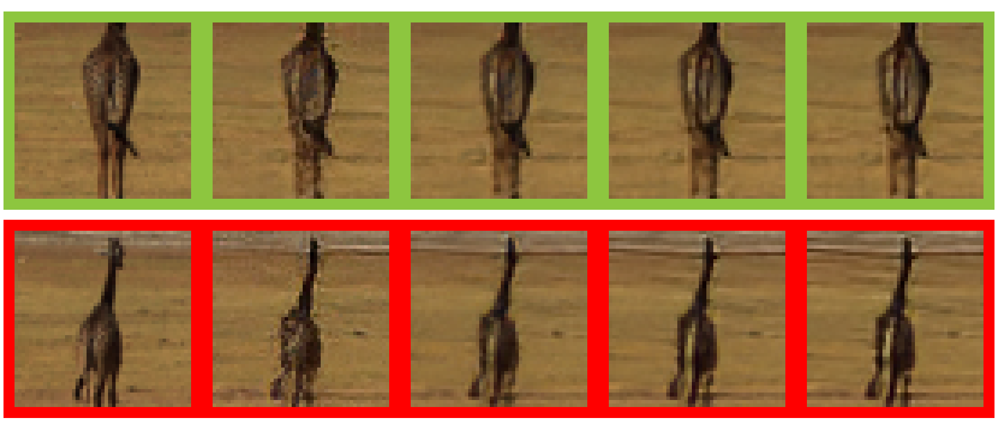
GT baseline [4] 1^{st} iter 2^{nd} iter 3^{rd} iter

Figure 11: Additional result of super-resolution task

baseline [4]



ours



GT baseline [4] 1^{st} iter 2^{nd} iter 3^{rd} iter

Figure 12: Additional result of super-resolution task

References

[1] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, and L. Zelnik-Manor. 2018 pirm challenge on perceptual image super-resolution. In *ECCVW*, 2018. 3

[2] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *NIPS*, 2017. 1

[3] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 2, 3, 4, 5, 6, 7

[4] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, C. C. Loy, Y. Qiao, and X. Tang. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCVW*, 2018. 1, 8, 9, 10, 11, 12, 13