Adversarial Feedback Loop
—Supplementary—

Firas Shama[1] Roey Mechrez[1] Alon Shoshan[1] Lihi Zelnik-Manor[1,2]
[1] Technion - Israel Institute of Technology [2] Alibaba Group
{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 [3] 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 ESRGAN [6] 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.

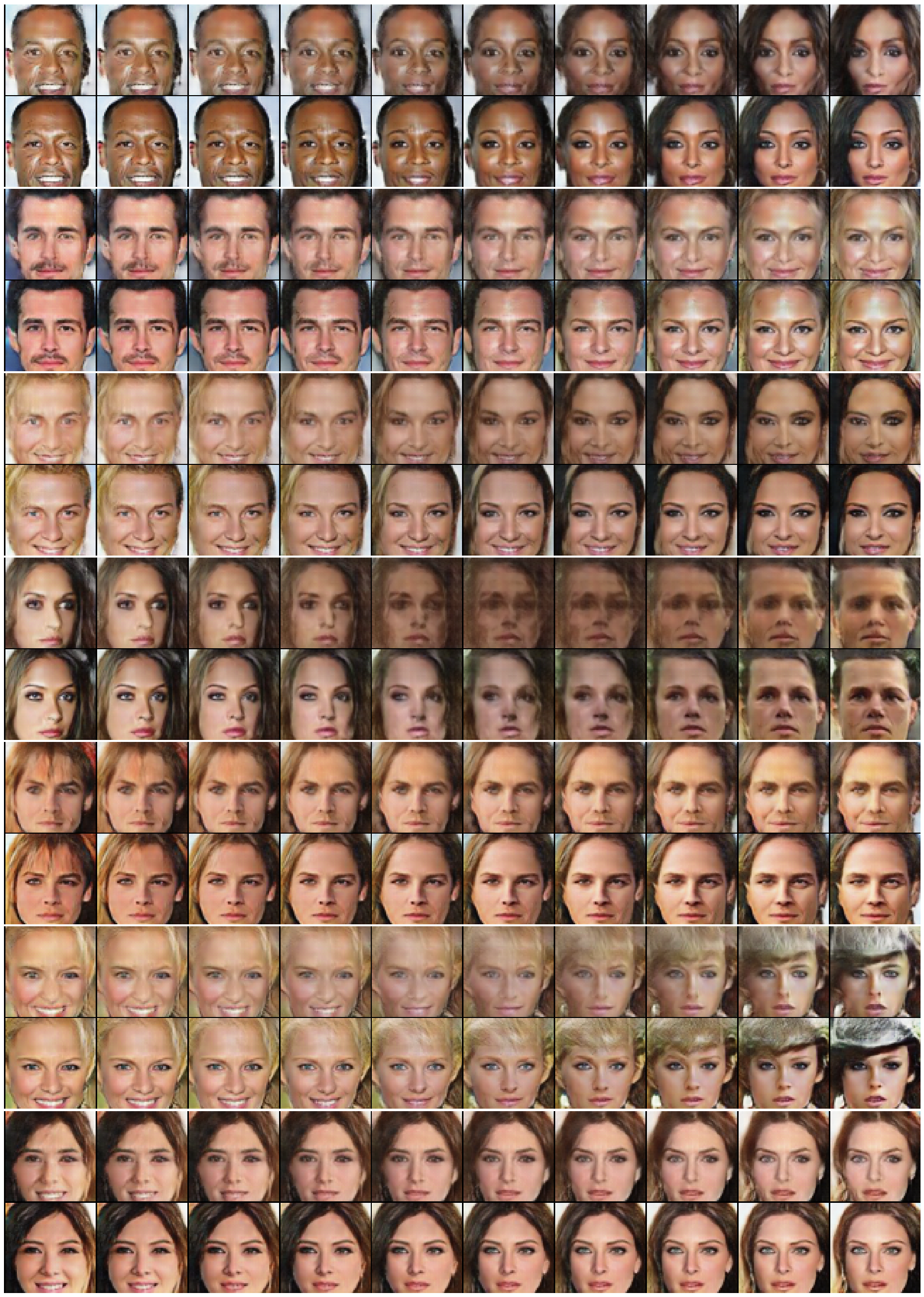


Figure 1: **Latent space interpolation:** Results of interpolation between two different input vectors. We compare DCGAN [5] baseline (odd rows) with ours, DCGAN+AFL (even rows).

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

Table 1: Inputs and output of each feedback module. We denote the inputs of the feedback module by A and B.

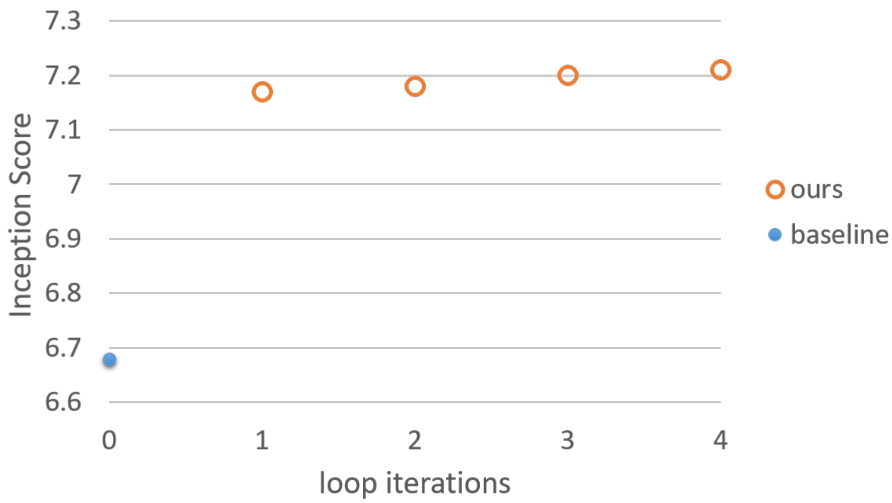


Figure 2: **Iteration impact:** Inception score of our method with WGAN-GP [7] network on CIFAR10 dataset. Feedback loop continues to improve the quality with more iterations, however, most of the benefit of the feedback iteration are achieved from single feedback iteration.

2. Additional Results

Iterations impact In all of our experiments we trained the feedback modules for single iteration, and reported the evaluation results of single feedback loop iteration as well. To show the result of network with higher feedback iterations, we chose the WGAN-GP [7] based network with our feedback modules that are used in CIFAR10 experiment. Figure 2 shows the Inception score achieved when we use higher iterations in test time. Results show that feedback loop continues to improve the quality with more iterations, however, most of the benefit of the feedback iteration are achieved from single feedback iteration.

LSUN In the task of unsupervised image generation, we performed additional experiment on the LSUN [8] bedrooms 64×64 dataset. In this experiment we chose the same baseline network [5] that was used for CelebA dataset, with the same set of feedback modules. As evaluation method we used the FID measure. Results show that when fused with our feedback modules, the network FID score was improved from 27.6 to **13**.

CelebA More results of our method compared with the baseline [5] 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. Note that we do not present results with higher α values, as the images do not continue to change notably.

We show the comparison between our method versus a simple image interpolation between the network output and the reference image in Figure 7, the results show that simple interpolation between the reference image and the network output achieves inferior results compared to our proposed pipeline, e.g. the faces are blended and blurry. In fact, one can think of the proposed method as an interpolation in latent space rather than image space.

Super-resolution Additional results of the contribution of AFL in the super-resolution task are presented in Figures 9, 10, 11, 12 and 13 for images from PIRM challenge [2] data-set. In addition, an interesting result is shown in Figure 8, where stripes are corrected to the true direction and merged more naturally.

We performed evaluation of our method on more datasets: Set5 [1],Set14 [9] and BSD100 [4], see Table 2 that summarizes the evaluation results in terms of PI [2] and RMSE. Note that after adding the feedback modules, an evident improvement in the perceptual quality is seen through lower PI.

Dataset	**PI[2]**		**RMSE**	
	baseline [6]	**+AFL**	**baseline [6]**	**+AFL**
Set5[1]	3.792	**3.497**	**8.11**	8.3
Set14[9]	2.916	**2.852**	15.04	**14.43**
BSD100[4]	2.483	**2.357**	16.38	**15.95**

Table 2: Super resolution performance on more datasets, after adding the feedback modules, an evident improvement in the perceptual quality is seen through lower PI.

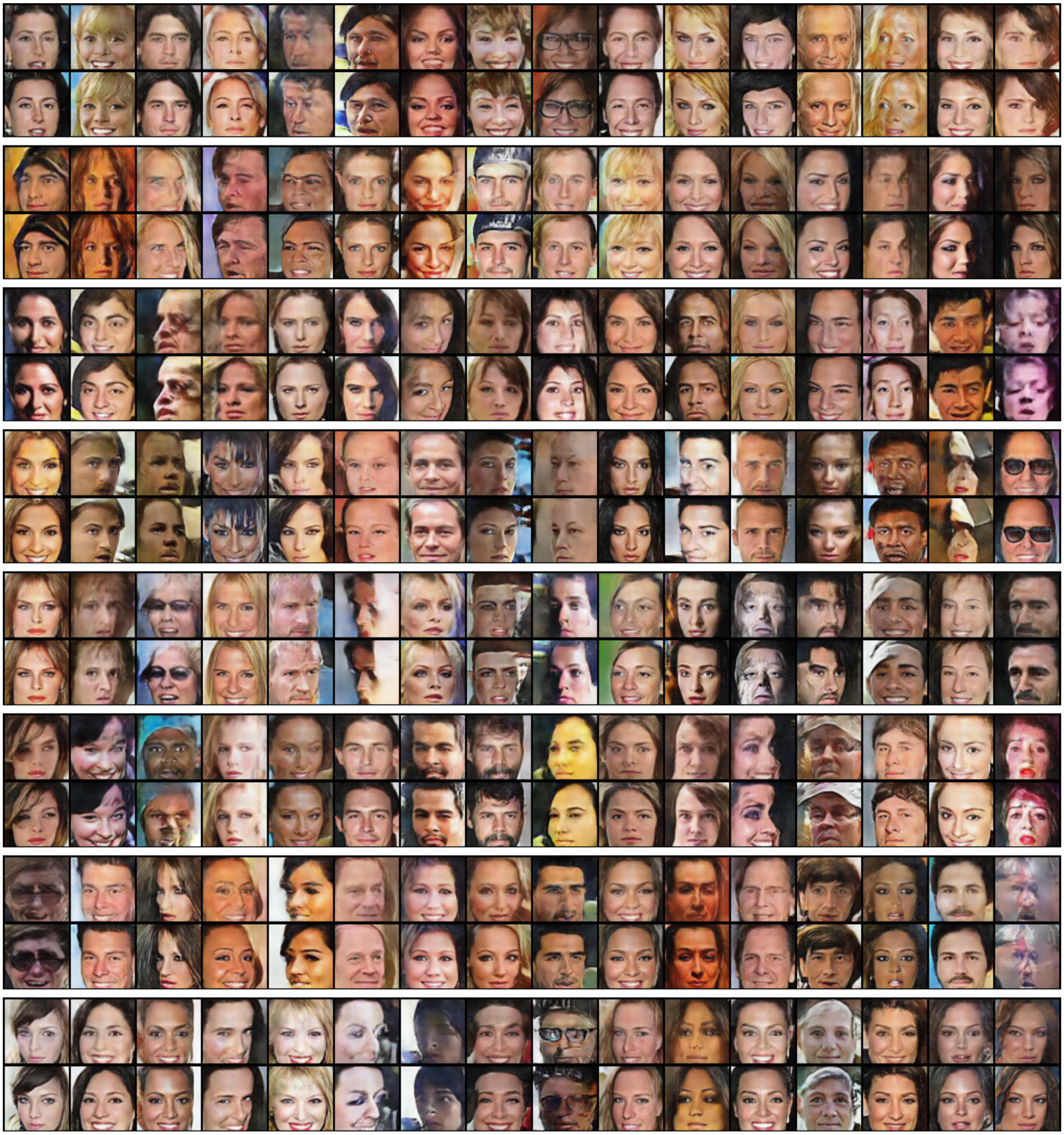


Figure 3: **Full batch results**. We compare DCGAN [5] baseline (odd rows) with ours, DCGAN+AFL (even rows).

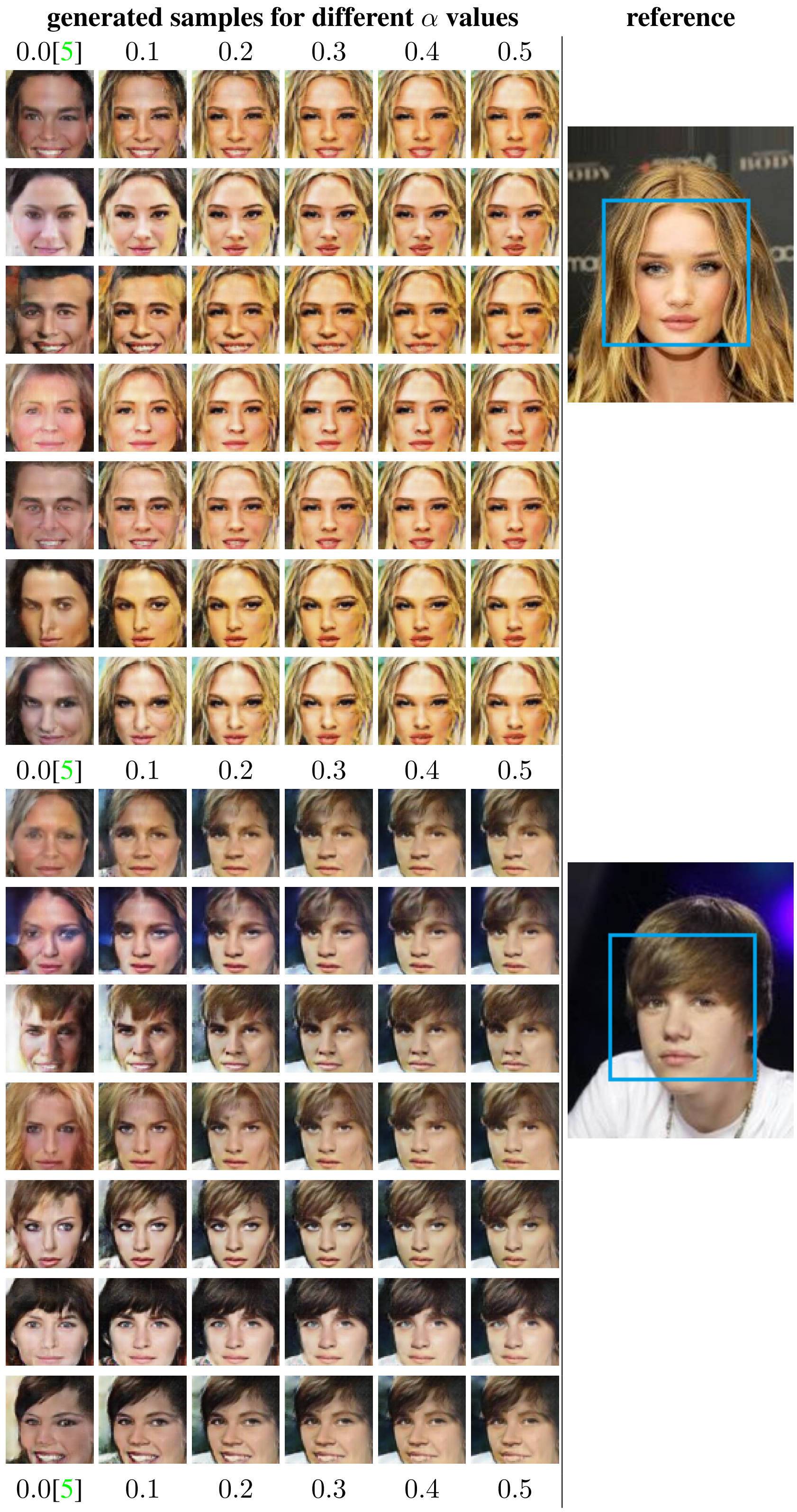


Figure 4: **Generation with reference:** More Results of using the feedback-switching-pipeline

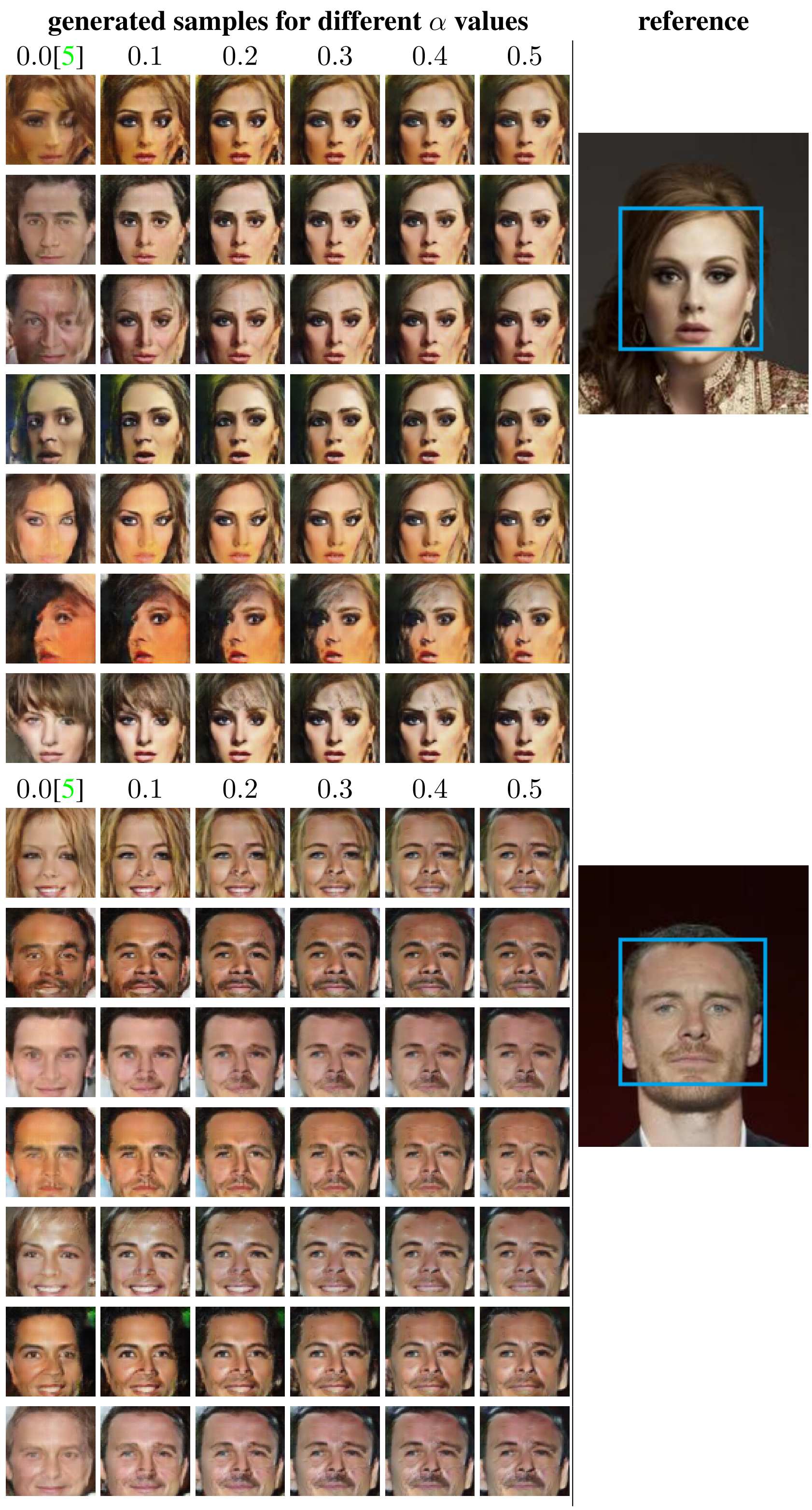


Figure 5: **Generation with reference:** More Results of using the feedback-switching-pipeline

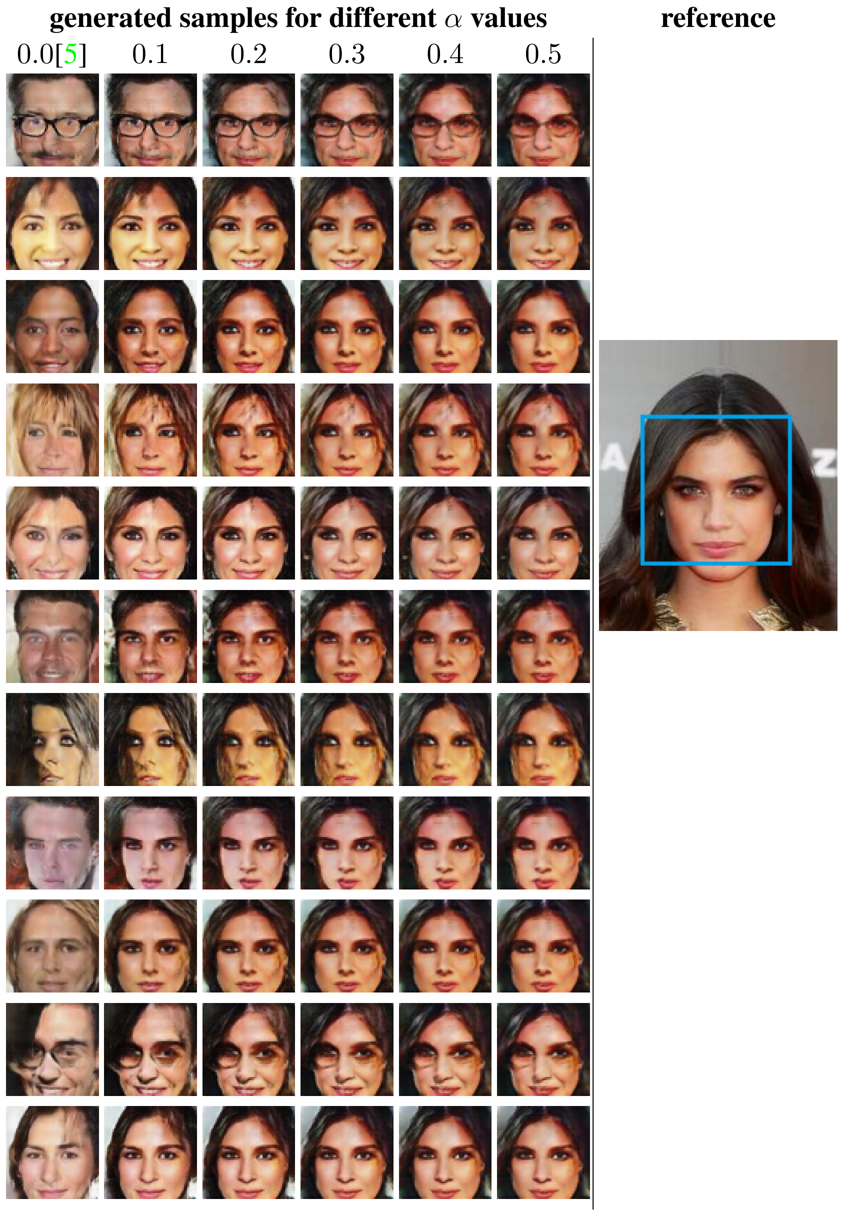


Figure 6: **Generation with reference:** More Results of using the feedback-switching-pipeline

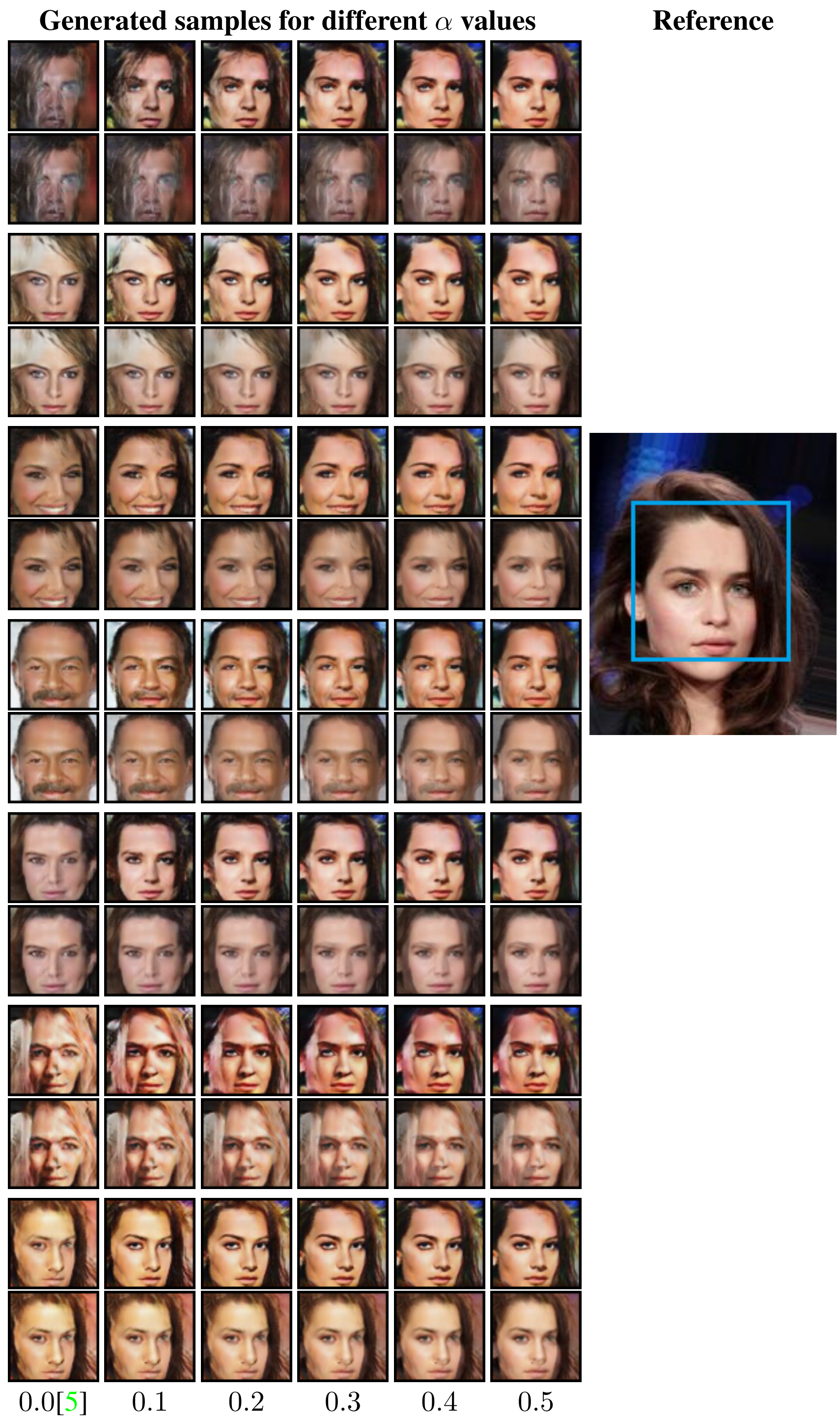
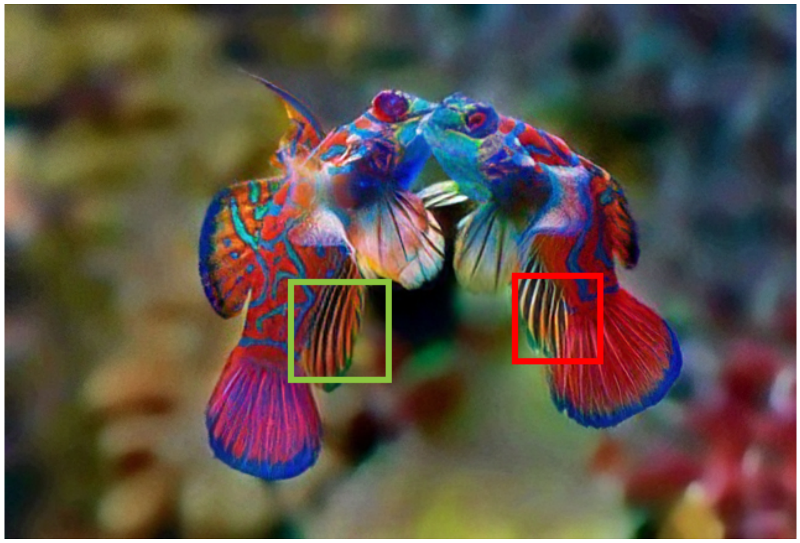
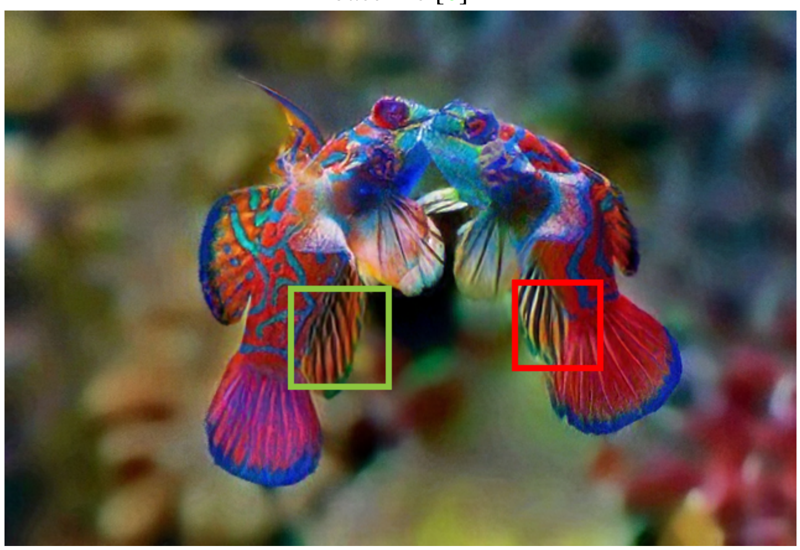


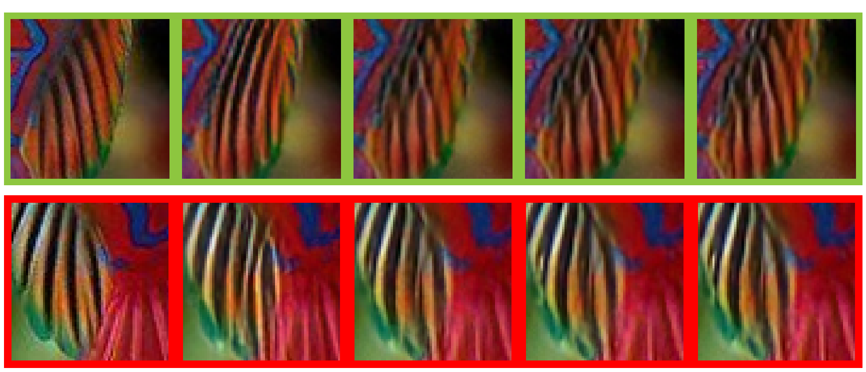
Figure 7: **Generation with reference vs Interpolation:** Results of using the feedback-switching-pipeline vs simple interpolation. First column is DCGAN [5] baseline. Odd rows: ours, even rows: image interpolation with reference. Our method produces much higher quality images compared to simple interpolation



baseline [6]



ours

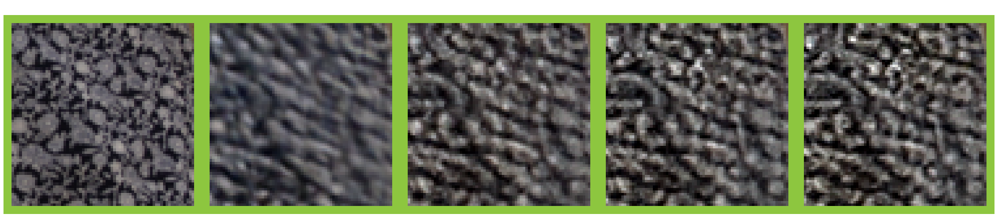


GT | baseline [6] | 1^{st} iter | 2^{nd} iter | 3^{rd} iter

Figure 8: Green patch: stripes corrected to the true direction through feedback iterations, red patch: stripes are merged together more naturally

baseline [6]

ours

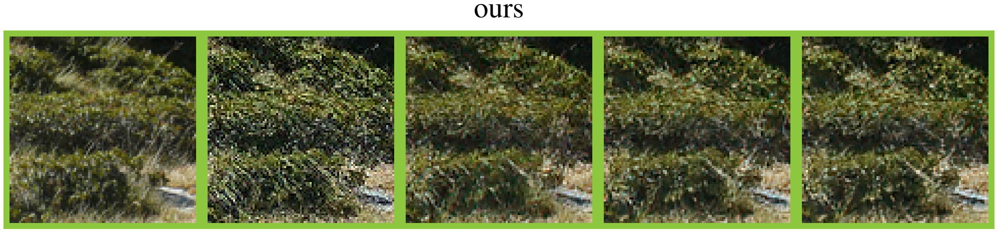


GT baseline [6] 1^{st} iter 2^{nd} iter 3^{rd} iter

Figure 9: Additional result of super-resolution task

baseline [6]

ours



GT | baseline [6] | 1^{st} iter | 2^{nd} iter | 3^{rd} iter

Figure 10: Additional result of super-resolution task

baseline [6]

ours

GT baseline [6] 1^{st} iter 2^{nd} iter 3^{rd} iter

Figure 11: Additional result of super-resolution task

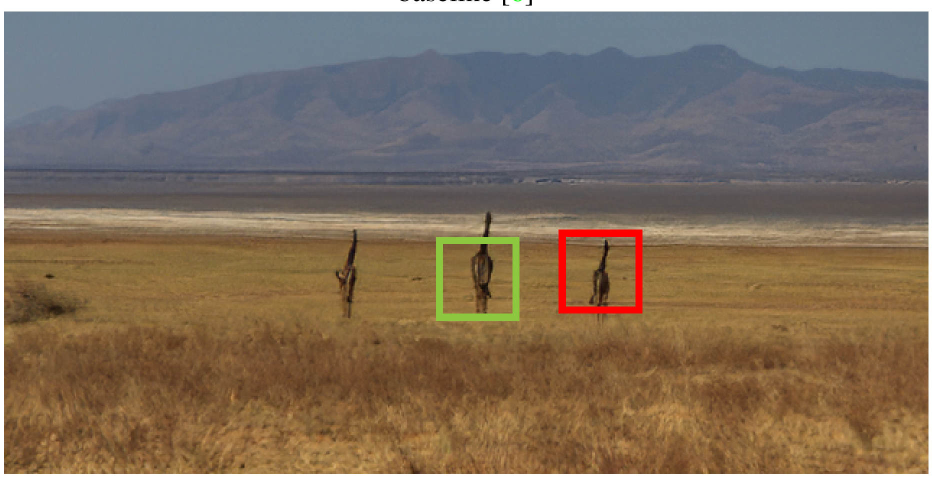
baseline [6]

ours

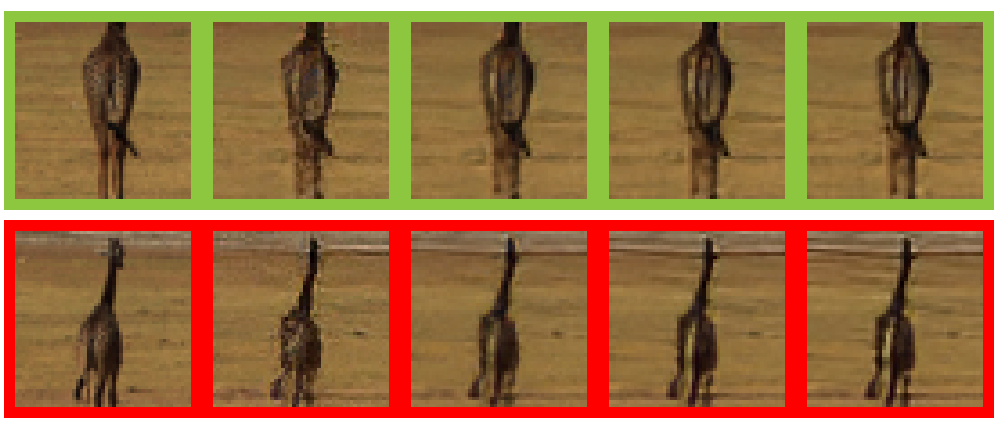
GT baseline [6] 1^{st} iter 2^{nd} iter 3^{rd} iter

Figure 12: Additional result of super-resolution task

baseline [6]



ours



GT baseline [6] 1^{st} iter 2^{nd} iter 3^{rd} iter

Figure 13: Additional result of super-resolution task

References

[1] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012. 4

[2] Yochai Blau, Roey Mechrez, Radu Timofte, Tomer Michaeli, and Lihi Zelnik-Manor. 2018 pirm challenge on perceptual image super-resolution. In *ECCVW*, 2018. 4

[3] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NIPS*, 2017. 1

[4] David Martin, Charless Fowlkes, Doron Tal, Jitendra Malik, et al. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Iccv Vancouver:, 2001. 4

[5] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 2, 3, 5, 6, 7, 8, 9

[6] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao, and Xiaoou Tang. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCVW*, 2018. 1, 4, 10, 11, 12, 13, 14, 15

[7] David Warde-Farley and Yoshua Bengio. Improving generative adversarial networks with denoising feature matching. 2016. 3

[8] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 3

[9] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010. 4