Adversarial Feedback Loop

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

Abstract

Thanks to their remarkable generative capabilities, GANs have gained great popularity, and are used abundantly in state-of-the-art methods and applications. In a GAN based model, a discriminator is trained to learn the real data distribution. To date, it has been used only for training purposes, where it's utilized to train the generator to provide real-looking outputs. In this paper we propose a novel method that makes an explicit use of the discriminator in test-time, in a feedback manner in order to improve the generator results. To the best of our knowledge it is the first time a discriminator is involved in test-time. We claim that the discriminator holds significant information on the real data distribution, that could be useful for test-time as well, a potential that has not been explored before.

The approach we propose does not alter the conventional training stage. At test-time, however, it transfers the output from the generator into the discriminator, and uses feedback modules (convolutional blocks) to translate the features of the discriminator layers into corrections to the features of the generator layers, which are used eventually to get a better generator result. Our method can contribute to both conditional and unconditional GANs. As demonstrated by our experiments, it can improve the results of state-of-the-art networks for super-resolution, and image generation.

1. Introduction

Adversarial training [6] has become one of the most popular tools for solving generation and manipulation problems. For example in image generation [6, 26], super-resolution [19], image-to-image transformation [11, 34], text-to-image [27] and others. Common to all these works is the discriminator–generator information flow – via a loss function. That is, the generator output images are fed into the discriminator which produces a 'real-fake' score for each image in terms of a pre-defined loss function. This score is back-propagated to the generator through gradients.

Recent research in the GAN field discusses the design of the loss function and regularization terms. For example, the basic cross-entropy loss [6], Wasserstein distance [1], spectral normalization [24] or relativistic discriminator [13]. Where latest state-of-the-art methods [14, 15] encourage to use deeper GAN networks that are gradually trained. This discussion has contributed significantly to the advancement of GANs and using a discriminator has become highly effective. To date, after training, the discriminator is forsaken and the deep understanding of the data distribution is lost. This seems wasteful to us, hence, we seek a way to enjoy the discriminator also during test-time. In addition, encapsulating the discriminator information into a single score looses the spatial understanding of which regions are more 'real' and which are considered 'fake'. In the current scheme only limited spatial information flows with the back-propagation because the gradients are averaged over each batch.

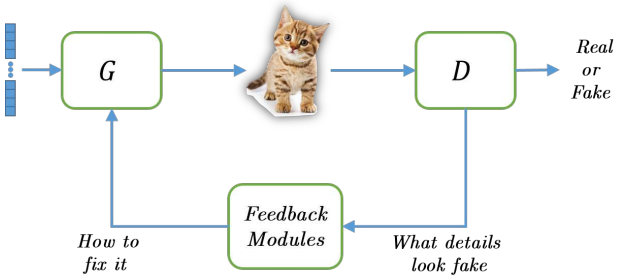


Figure 1: **The adversarial feedback loop**: Classic GAN is composed of two components: the generator (G) and the discriminator (D). In this setting, the information flow is done purely by back-propagation, during training. We propose adding a third component – the *feedback module* that transmits the discriminatory spatial information to the generator in a feedback manner at inference time.

In this paper we propose a different approach that explicitly exploits the discriminator's activations, in test-time, in order to improve the generator output. We propagate the discriminator information back to the generator utilizing an iterative feedback loop, as illustrated in Figure 1. The overall framework is as follows: We start with classic training of the generator and discriminator. Then, at test-time, the

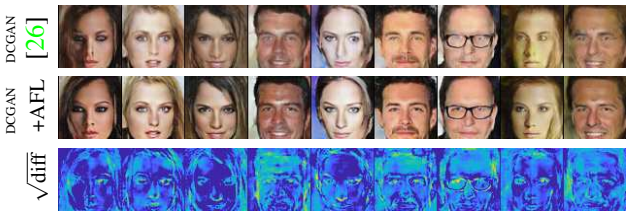


Figure 2: **Contribution of AFL:** Face generation results of (top) DCGAN [26] and (middle) integrating DCGAN within the proposed AFL framework. Faces generated by DCGAN+AFL are sharper and show fewer artifacts. (bottom) The differences between the images generated without and with AFL highlight that AFL corrects the spatial regions that suffer from artifacts, for example, the cheek of the left-most face, and the eyes of the second from left.

generator produces an output image which fed into the discriminator in order to compute its feedback. The discriminator activations are fed into a third module which we name – *feedback module*. The goal of this module is to convert the discriminator activations into 'corrections' which can then be added to the original generator activations. We repeat this process iteratively until convergences (1-3 iterations).

The main contributions of the Adversarial Feedback Loop (AFL) are two-fold. First, to the best of our knowledge, our novel adversarial feedback loop is the first use of the discriminator at test-time in the framework of GAN. Second, our scheme makes the spatial discriminative information accessible to the generator, allowing it to 'correct' artifacts and distortions thus producing higher quality images. A few motivational examples are presented in Figure 2, where it can be seen that our pipeline takes images produced by the generator and corrects those regions that suffered from artifacts.

Experiments on the CIFAR-10 [16] and CelebA [22] data-sets for the task of unsupervised image generation show that combining AFL with state-of-the-art GAN methods improves the Inception Score (IS) [28] and *Fréchet inception distance* (FID) [9] which is evident also qualitatively as better visual quality. In addition, when integrated with ESRGAN [31], a state-of-the-art method for super-resolution, AFL can further improve the results; it achieves higher Perceptual Index [2], PSNR and SSIM [32] and lower RMSE, making the results more visually appealing and more trustworthy to the ground truth.

2. Related Work

The idea of exploiting the discriminator features served as motivation to some previous works. The GAN-based methods [30, 17] proposed to use a loss based on features extracted from the discriminator layers. They compared the discriminator features of fake image to the discriminator features of real images, in a similar manner to the renowned perceptual loss [5]. In all those methods, the utility of the discriminator layers was limited to training-time, which is different from our suggestion to enjoy its benefits also in test-time.

The concept of feedback has already made its way into the training framework of several previous works that exploit the iterative estimation of the output aiming at better final results. In [25] a feedback loop was trained for hand pose estimation, in [4] feedback was used for human pose estimation, while [21] proposed to use the feedback for the problem of instance segmentation. [33] suggest a general feedback learning architecture based on recurrent networks, that can benefit from early quick predictions and from a hierarchical structure of the output in label space. An interesting solution for the task of video frames prediction was presented in [23], that introduced an unsupervised recurrent network that feeds the predictions back to the model. Feedback was also used for super-resolution by [8] that suggest a network that uses the error feedback from multiple up and down-scaling stages. A fused generator-discriminator network was proposed in [18, 20] that performs self-evaluation to its results.

To the best of our knowledge, none of these previous methods have proposed applying the concept of feedback in the framework of GAN. To place our proposed framework in context with the terminology common in works discussing feedback paradigms, one can think of the generator as a 'predictor', the discriminator as an 'error estimator' and feedback modules close the loop and convert errors from the discriminator feature space to the generator feature space.

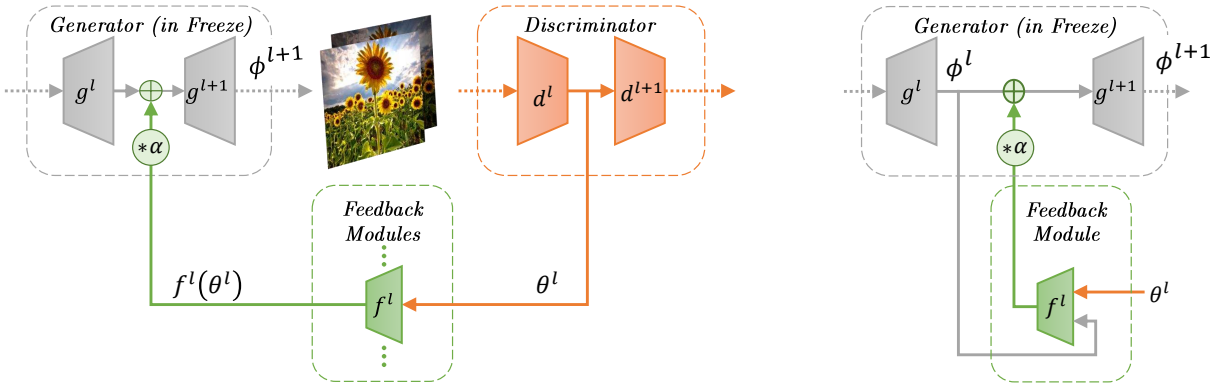
3. Method

In this section we present our AFL framework. As discussed in the introduction, all current methods use the discriminator for adversarial training only. We wish to change that and explicitly use the knowledge that the discriminator has gathered also during test-time. This way the discriminator can "leak" information to the generator by providing feedback on the generator's failures and thus assisting the generator in fixing them. We designed a solution that is generic so that it can be integrated with any GAN based network.

3.1. Framework Overview

Given a generator-discriminator architecture, G and D respectively, we denote their layers by $\{g^l\}_{l=1}^n$ and $\{d^l\}_{l=1}^n$, where l is the layer index. These layers (or some) are connected via feedback modules $\{f^l\}_{l=1}^n$[1]. The input to each feedback module is the activation map of the corresponding layer of the discriminator $\theta^l = d^l(\theta^{l-1})$. The output

[1]Each of which consists of two convolutional layers: $(CONV-BN-RELU-CONV-BN)$



(a) Generator–Discriminator–Feedback module pipeline (b) Dual-input feedback module

Figure 3: **The feedback framework:** (a) The proposed feedback module passes information from the discriminator to the generator thus "learning" how to correct the generated image in order to make it more real in terms of the discriminator score. (b) It is also possible to let the feedback module consider both the features of the discriminator and the features of the generator.

of each feedback module is added to the corresponding activation map of the generator $\phi^l = g^l(\phi^{l-1})$, thus forming a *skip-connection*, such that the generator activation maps change to:

$$\phi^{l+1} = g^{l+1}\left(\phi^l + f^l(\theta^l)\right). \tag{1}$$

See Figure 3(a) for illustration. Each feedback module is further associated with a scalar parameter α^l that multiplies its output. Setting $\alpha^l = 0$ deactivates the l'th module altogether, while $\alpha^l \neq 0$ tunes the contribution of the feedback.

The basic feedback module connects between equally-sized activation layers of the discriminator and the generator. We also suggest a slightly more complex form, where the feedback modules are given as input the activation maps of both the discriminator and the generator (concatenated, noted as $[\cdot,\cdot]$), as illustrated in Figure 3(b), such that the generator activation maps change to:

$$\phi^{l+1} = g^{l+1}\left(\phi^l + f^l([\theta^l, \phi^l])\right). \tag{2}$$

3.2. Training

The training scheme consists of two phases. The first phase is identical to the common practice in training GANs. The feedback modules are inactive and we apply standard adversarial network training, in which the generator, G, and the discriminator, D, are trained according to the selected base method. The outcome is a trained generator and a trained discriminator that can differentiate between real images and fake images produced by the generator.

The second training phase is where we activate the feedback modules and train them. This is done while freezing the generator G, but allowing the discriminator D to keep updating. This way the feedback modules learn to correct the generator results in order to improve them based on the feedback given from the discriminator. Since the output from the generator improves, we must allow the discriminator to continue and refine its weights,otherwise, the generator would generate adversarial examples to fool the fixed discriminator.

We next write in further detail the steps of the second phase of the training:

First iteration $t = 0$ Given input x (e.g., a random vector), the generator produces an initial output image $y_0 = G(x)$ that is fed into the discriminator.

The $t > 0$ **iteration** We set $\alpha^l = 1$ and use the following update equation:

$$y_{t+1} = G\left(x, \mathcal{E}(y_t)\right) \tag{3}$$

where $\mathcal{E}$ aggregates the output of all the feedback modules:

$$\mathcal{E}(y_t) = \{f^l\left(\theta^l(y_t)\right)\}_{l=1}^n \tag{4}$$

or

$$\mathcal{E}(y_t) = \{f^l\left([\theta^l(y_t), \phi^l(y_{t-1})]\right)\}_{l=1}^n \tag{5}$$

depending on which feedback module type is used. In practice, in almost all our experiment two iterations sufficed, i.e. until we get y_1.

The Objective The feedback modules are trained with the same objective as the baseline generator (e.g. cross entropy, Wasserstein distance, etc.), while replacing every instance of the term $G(x)$ with the term $G(x, \mathcal{E})$.

3.3. Testing

At test-time we freeze the entire network including the generator, the discriminator and the feedback modules. The activation levels of the feedback modules are tuned by setting $\alpha^l = \alpha$, i.e.

$$y_{t+1} = G\left(x, \alpha \cdot \mathcal{E}(y_t)\right). \tag{6}$$

Typically the impact of the corrections from the feedback modules needs to be attenuated and we have found empirically that best results are obtained when $0.1 \leq \alpha \leq 0.2$. This way only the stronger corrections really contribute.

Note, that because of the batch-norm layer in each feedback module, its output signal is forced to be with the same strength (variance) as the generator features, such that multiplying the output by a small α is sufficient in order to preserve the original features, and marginally correct them. Unless otherwise specified, in all our experiments we used the value of $\alpha = 0.2$.

In principle the test-time process could also be repeated iteratively, however, we have found that it suffices to run a single iteration to achieve satisfactory results.

Computation Complexity The computation complexity of the additional training phase depends on the amount of loop iterations used and the complexity of the feedback modules. In our experiments with single loop iteration and single feedback module, the second training phase took roughly 10% more time than the first phase.

4. Experiments

In this section we present experiments conducted on several data-sets, tasks and networks that demonstrate the contributions of our method. In all cases we add our AFL to existing methods while adopting their training details: the architecture, the loss function, the normalization term and the hyper-parameters. Our modifications are reserved to the second training phase, in which we train the feedback module and to the testing phase, where we use the feedback module during generation. More experiments on the supplementary.

4.1. Empirical analysis on a simple 2D case

Before diving into applications, we first perform empirical evaluation of AFL in a simple, yet illustrative, scenario. The goal here is to show that AFL is effectively able to utilize the discriminator information in order to improve the generated results.

The scenario we chose is generation of 2D coordinates that lie on a 'Swiss roll'. The generator gets a random input points $x \in \mathbb{R}^2$ and is trained to generate points $y \in \mathbb{R}^2$ that fit into the 'real' data distribution, represented by the discriminator. The discriminator is trained to classify each sample as 'real' or 'fake' according to a given ground-truth swiss-roll data distribution.

As architecture, we chose both the generator and the discriminator to consist of a sequence of four fully-connected layers. For the feedback we used a single module, that corrects the input of the last layer of the generator. The objective we used was the WGAN-GP [7] adversarial loss. Please refer to the supplementary for implementation details. As baseline model the generator and the discriminator were trained for $8K$ iterations. Then we froze the generator, added the feedback module and trained it with the same discriminator for another $8K$ iterations.

Our results are presented in Figure 4. It can be observed that the baseline generator succeeds to generate samples close to the real data distribution, however, using the proposed AFL pipeline improves the generation accuracy and results in a distribution that is much closer to the real one. The AFL module identifies that inaccuracies in the generated points, corrects them, and leads to better results.

Method	Inception score		FID	
	Baseline	+AFL	Baseline	+AFL
DCGAN [26]	$6.64 \pm .14$	**7.02** $\pm .06$	31.3	**30.3**
WGAN-GP[7]	$6.68 \pm .06$	**7.17** $\pm .05$	33.1	**28.4**
SN-GAN [24]	$7.42 \pm .08$	**7.73** $\pm .05$	23.1	**20.7**

Table 1: **Image generation with AFL:** Inception scores and FID of recent methods in unsupervised image generation on CIFAR-10. Combining our proposed AFL with any one of three baselines improves the quality significantly.

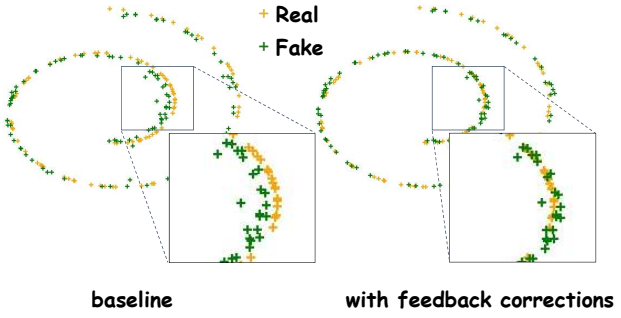
4.2. Image generation on CIFAR-10

As a second test-case we chose the task of unsupervised image generation on CIFAR-10 dataset. The goal is to generate 32×32 images given as input vectors with normally distributed elements, $x \in \mathbb{R}^{128}$. The generated images should match the diversity and quality of the training dataset used by the discriminator. We choose the commonly used Inception Score (IS) [28] (higher is better) and *Fréchet inception distance* (FID) [9] (lower is better) as a quality measure of the generated images, as it has been shown to be strongly correlated [10, 3] with generated data quality. As accepted, we compute the mean IS and FID over $50K$ samples.

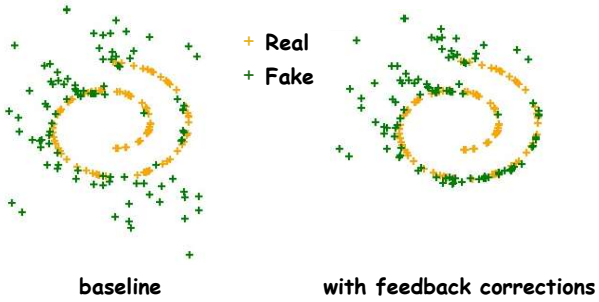
Many methods have been proposed before for this task. In order to demonstrate the generality of our approach to the choice of architecture and loss, we performed experiments with three different baseline methods. The first is WGAN-GP [7][2] and the second is SN-GAN [24][3].Both of

[2]Used pyTorch implementation github/caogang/wgan-gp

[3]Used pyTorch implementation github/christiancosgrove/sn-gan



(a) Test input variance = training input variance



(b) Test input variance = $5\times$ training input variance

Figure 4: **Swiss-roll generation:** Results of generating points that lie on a swiss-roll. (a) When the variance of the random input during test-time matches the variance of the inputs used for training the baseline generator does a decent job. Adding AFL corrects the small inaccuracies and yields a distribution almost identical to the real one. (b) When the variance of the random input is increased, the baseline generator fails. Conversely, using AFL has still succeeds to some extent to reproduce the swiss-roll distribution.

these methods are GAN based, built on standard CNN architecture, and try to comply to Lipschitz [1] constraints by limiting the weights/gradients of the discriminator. To show that AFL is effective even when no smart or complex normalization/constraints are applied, we choose a standard DCGAN [26] network as a third baseline.

The AFL included a single feedback module that connected between the intermediate layer features with spatial size of 8×8. For the SN-GAN network, we used the dual-input type of feedback module as it gave better results, while for the other two networks we used the basic type of single input.

Each baseline model was trained according to its recommended scheme. Once the baseline network is trained and perform close to the reported IS, we freeze the generator and train the feedback module and the discriminator.

Table 1 compares the performance of the three baseline networks, to those with AFL. It is evident that in all three cases AFL improves the mean of both IS and FID.

Setup	IS
[7] (original setup)	$6.68 \pm .06$
[7] + longer training	$6.78 \pm .10$
[7] + bigger generator	$6.87 \pm .06$
[7] + longer training + bigger generator	$6.92 \pm .07$
[7] + VGG feedback	$6.98 \pm .07$
[7] + AFL	$\mathbf{7.17} \pm .05$

Table 2: Performance of several training setups for WGAN-GP [7]. Our method achieves best IS results

Sanity Check To validate that the feedback module indeed learns to harness the feedback from the discriminator we performed two sanity-check experiments in test-time.

In the first one we used the AFL pipeline, with WGAN-GP as baseline, but rather than using the feedback from the discriminator, we fed the AFL with normally distributed input. The results was a very low IS of 3.44 on CIFAR-10. This result shows that the feedback module develops a strong dependency on the discriminator feedback provided in test-time.

As a second sanity check we performed a similar experiment, this time using as feedback the activation maps of the discriminator, albeit, computed on a different (wrong) image from the batch. Interestingly, this abuse of the solution results in IS of 5.76, which is better than the score of the first sanity check, but still far behind the score when using the correct feedback. This stands to show that the feedback module learns to create corrections based on the discriminator feedback on the specific image.

Ablation Study To further investigate the results improvement of the presented networks in the AFL setup, we choose WGAN-GP [7] network, train it with alternative setups and compute their mean IS. Where we choose: (a) baseline network with default setup. (b) baseline with twice training iterations. (c) baseline with doubled generator parameters. (d) last two setups combined. (e) baseline with pretrained VGG19 `relu3_4` features feeding feedback modules. (f) baseline + AFL. Table 2 shows that compared to all other setups, our method achieves best IS results.

4.3. Face generation on CelebA

As a third task we chose unsupervised face image generation on CelebA dataset [16], because it allows both quantitative evaluation as well as qualitative evaluation through visualizing the generated faces. The output here consists of face image of size 64×64, thus a human observer can easily detect distortions and artifacts (unlike CIFAR-10 where visual inspection is not considered effective).

Figure 5: **Face generation on CelebA:** We compare DCGAN [26] baseline (odd rows) with ours, DCGAN+AFL (even rows). Faces generated with AFL show significantly fewer artifacts, making clear the advantage of using AFL.

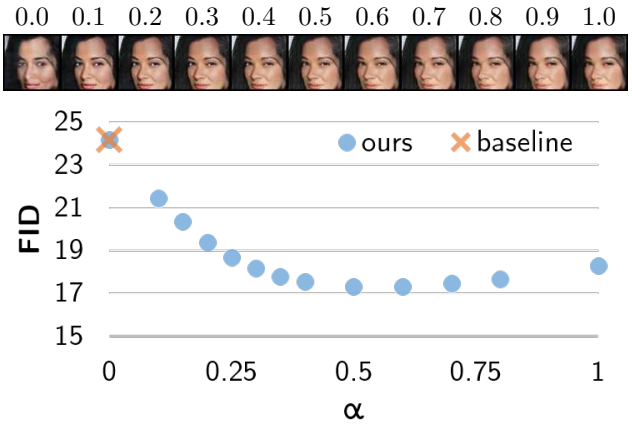


Figure 6: **Quantitative evaluation on CelebA:** The graph shows the mean FID over 50K samples for multiple α value. It can be seen that AFL always improves the score.

Figure 7: **Analysis of module contribution:** Face generation results while disabling all modules except the one specified. Shallow modules have the most impact on the result, while deep modules affect only colors and fine details.

As baseline network we adopted DCGAN [26]. For feedback we used the dual-input option with four feedback modules, each with two convolutional layers. First phase training followed the recommended setup until convergence and obtaining results comparable to those originally reported in [26]. The second phase trained the feedback modules with the discriminator, for the same number of epochs as in the first phase.

To quantitatively evaluate the contribution of AFL over the baseline DCGAN network, we use the *Fréchet inception distance* (FID) [9] to measure the distance between the distribution of the real data and the generated data. The real distribution is estimated based on 50K samples. We performed an evaluation on several values of α, as presented in Figure 6, all of them improve the baseline result ($\alpha = 0$) which has FID of 24.17. The best quantitative result was achieved with $\alpha = 0.5$ yielding to FID of 17.32. Arguably, in qualitative inspection, often the best looking results were with $\alpha = 0.2$, hence, we present in Figure 5 some of these images. In almost all samples we see a clear quality improvement brought in by AFL (more on the supplementary).

To provide further insight on why AFL is useful we performed another experiment, in which we study the contribution of each module of AFL. Specifically, we disabled ($\alpha^l = 0$) all the feedback modules except for the one whose impact we want to study, and repeated that for all the modules. Results are presented in 7. As could be speculated, the shallowest module has the strongest correction impact. Since it has the biggest "receptive field", it learned to correct high- mid- level errors, such as the head shape. The last module learned to correct low-level features such as colors

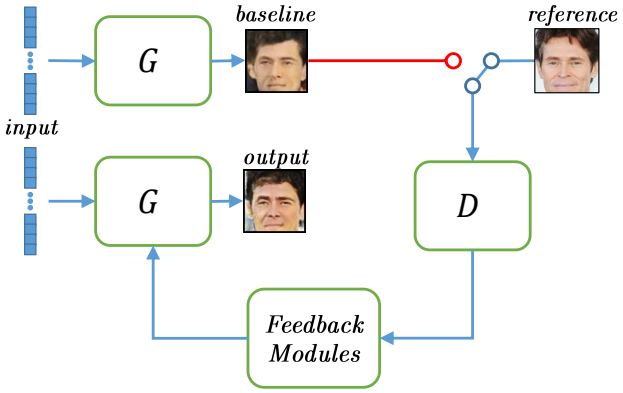


Figure 8: **Feedback switching pipeline:** Instead of using AFL in its standard pipeline, we replace the input to the discriminator with a reference image, this allows us to control the similarity of the generated image to the reference image.

and skin-tone. Finally, the intermediate modules learned to correct eye gaze, smile, nose shape etc.

Feedback Switching Another experiment we performed was in order to further study the utility of the information encompassed in the discriminator. To do that we investigated what happens when a feedback of an image is replaced by a feedback from another image, see Figure 8 for illustration. This means that the feedback modules that correct the generation of image A receive their input from a discriminator that is fed with another image B, namely:

$$y_1 = G\left(x_A, \alpha \cdot \mathcal{E}(B)\right) \tag{7}$$

Interestingly, this experiment's results show that with such a setup, the feedback modules "correct" the output image in the direction of the reference image. Rather than correcting artifacts, the AFL modifies the content of the image to match the information it gets from the discriminator. This shows that the features extracted by the discriminator are powerful. As can be seen in Figure 9, when we set the hyper-parameter α to a higher value, we get an image with higher similarity to the reference image, and vice versa. Furthermore, the faces suffer from fewer artifacts than those generated by the baseline. Potentially, with such a framework we can control the randomly generated images of GANs. More in the supplementary.

4.4. Super Resolution: AFL in conditional GAN

Our experiments so far showed the benefits of AFL in concert with a variety of GANs for unconditional image generation. In all of these cases the objective is simple, consisting of only an adversarial loss, and the structure of

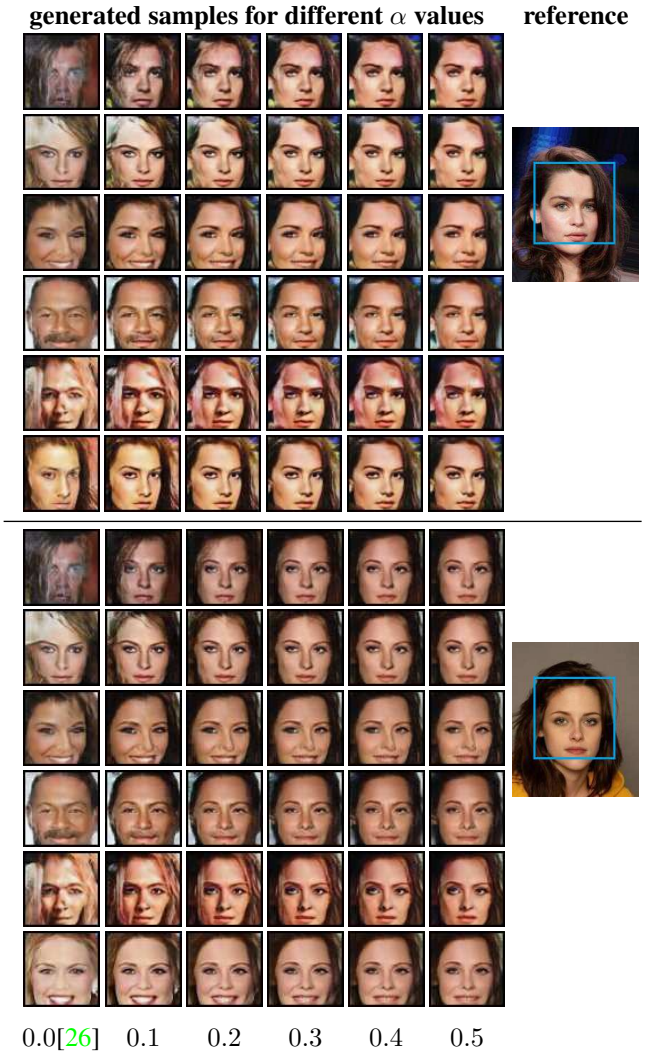


Figure 9: **Generation with reference:** Results of using the feedback-switching-pipeline of Figure 8. The feedback modules make the generated image similar to the reference one, and with fewer artifacts. This makes an interesting approach for controlled face generation. First column is DCGAN [26] baseline.

generator and discriminator are almost symmetric. This enabled straight-forward feedback loop from each discriminator layer to the equally-sized generator layer. Our next goal is to show the generality of AFL to more complex objectives and to non symmetric generator-discriminator structures. Specifically, we chose the problem of $\times 4$ super-resolution, where we hope that AFL will correct artifacts and fine details produced by the baseline network, e.g., the blurriness and checker-board patterns shown in Figure 10 (baseline).

Our solution is based on ESRGAN [31] which builds upon SRGAN [19], the first to use GANs for super-resolution. A variant of this network won the first place

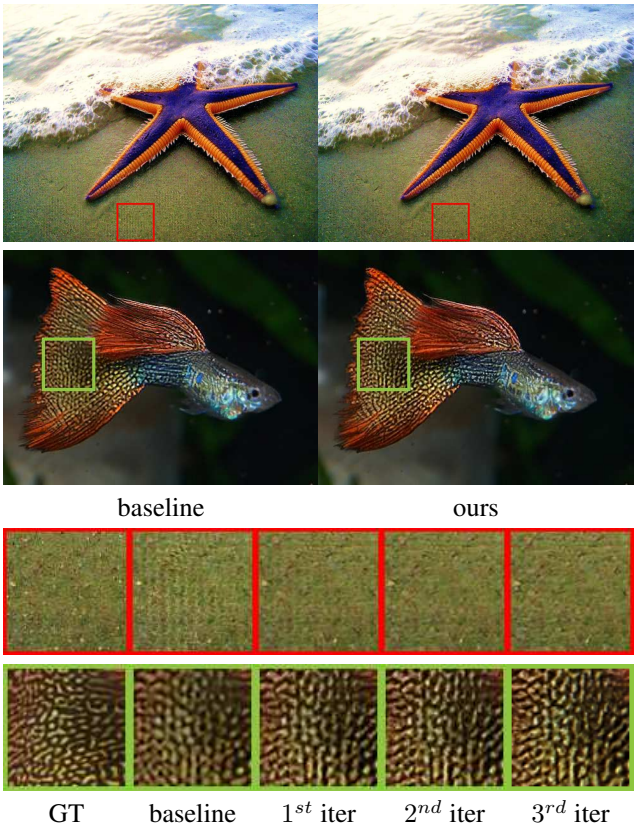


Figure 10: Our feedback loop removes artifacts and sharpen the image

in PIRM super-resolution challenge [2], but since the training code of the winner network isn't published, we used the slightly less competitive network that was officially published with the training code[4] as a model with the best visual quality. The key ideas behind ESRGAN are to use dense residual-in-residual as the main building block, optimize via relativistic GAN [13], abandon batch-normalization and use the features before activation in the perceptual loss [12]. These design decisions make the architecture of the generator more complicated than the classic architectures used in our previous experiments. Moreover, it raises a new challenge since the architectures of the generator and the discriminator differ fundamentally.

Our proposed AFL approach is generic enough to be easily adapted to such complex architectures. In this specific case we make the following adjustments: we use 4 dual-input feedback modules, spread along the generator residual blocks (see supplementary), each with three convolutional layers. Since ESRGAN avoids batch-normalization in the generator, we also remove this from the feedback modules. Since the layers of the generator and discriminator are not of the same dimension we upscale the discriminator features to match the size of the generator features.

[4]Used the official implementation github/xinntao/ESRGAN

Method	PI [2]	RMSE	PSNR	SSIM
ESRGAN	2.433	16.34	23.5	0.643
ESRGAN+AFL (ours)	**2.135**	**14.72**	**24.26**	**0.671**

Table 3: **Super-resolution results:** The table reports the mean scores over PIRM test set [2]. RMSE, PSNR, SSIM measure similarity to the ground-truth, while PI [2] measures non-reference perceptual quality. It can be clearly seen that adding AFL leads to an improvement on all scores.

We follow the training process of the original ESRGAN [31]. The network and the AFL are trained with DIV2K dataset [29], Adam optimizer and with an objective combining three loss terms: relativistic GAN [13], perceptual loss [12] and L1. At test time we set $\alpha = 1$.

As suggested in PIRM [2], we evaluate the contribution of AFL via RMSE and the Perceptual Index [PI], In addition, we also added the PSNR and SSIM [32] metrics. The scores reported in Table 3 show that adding AFL improves all measures. This supports the broad applicability of our proposed feedback approach.

Qualitative assessment of the results is provided in Figure 10. It can be seen that AFL reduces artifacts, corrects textures, and improves the realistic appearance of the generated high-resolution image. Refer to the supplementary for results on more datasets.

5. Conclusion

In this work we proposed a novel idea of applying a feedback loop at test-time as part of the GAN framework, thus utilizing the discriminator not only for training. We showed via several applications, with various architectures and setups, that such an approach improves the quality of the generated images.

Our approach has further potential to open-up new opportunities in image generation. Specifically, controlled image generation using a reference image could have broad applications. To leverage this we have built an interactive user interface, where a user can tune the impact of the feedback modules (by modifying α). We are now exploring the benefits of this user-interface, but preliminary experiments suggest that in many cases user-specific and image-specific tuning could be a good option.

Acknowledgements This research was supported by the Israel Science Foundation Grant 1089/16 and by the Ollendorf foundation.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 1, 5

[2] Yochai Blau, Roey Mechrez, Radu Timofte, Tomer Michaeli, and Lihi Zelnik-Manor. 2018 pirm challenge on perceptual image super-resolution. In *ECCVW*, 2018. 2, 8

[3] Ali Borji. Pros and cons of gan evaluation measures. *arXiv preprint arXiv:1802.03446*, 2018. 4

[4] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[5] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *NIPS*, 2016. 2

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 1

[7] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NIPS*, 2017. 4, 5

[8] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Deep backprojection networks for super-resolution. In *CVPR*, 2018. 2

[9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017. 2, 4, 6

[10] Gao Huang, Yang Yuan, Qiantong Xu, Chuan Guo, Yu Sun, Felix Wu, and Kilian Weinberger. An empirical study on evaluation metrics of generative adversarial networks. 2018. 4

[11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 1

[12] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*. 8

[13] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018. 1, 8

[14] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 1

[15] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 1

[16] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 2, 5

[17] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015. 2

[18] Justin Lazarow, Long Jin, and Zhuowen Tu. Introspective neural networks for generative modeling. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2774–2783, 2017. 2

[19] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 1, 7

[20] Kwonjoon Lee, Weijian Xu, Fan Fan, and Zhuowen Tu. Wasserstein introspective neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3702–3711, 2018. 2

[21] Ke Li, Bharath Hariharan, and Jitendra Malik. Iterative instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3659–3667, 2016. 2

[22] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. 2

[23] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016. 2

[24] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *ICLR*, 2018. 1, 4

[25] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Training a feedback loop for hand pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3316–3324, 2015. 2

[26] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 1, 2, 4, 5, 6, 7

[27] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, 2016. 1

[28] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, 2016. 2, 4

[29] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, Lei Zhang, Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, Kyoung Mu Lee, et al. Ntire 2017 challenge on single image super-resolution: Methods and results. In *CVPRW*, 2017. 8

[30] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *arXiv preprint arXiv:1711.11585*, 2017. 2

[31] 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. 2, 7, 8

[32] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 2, 8

[33] Amir R Zamir, Te-Lin Wu, Lin Sun, William B Shen, Bertram E Shi, Jitendra Malik, and Silvio Savarese. Feedback networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1808–1817. IEEE, 2017. 2

[34] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 1