The New Involution Operator and Its Application on Generative Adversarial Networks

Xiaozhu Fang
School of Science and Engineering
The Chinese University of Hong Kong
Shenzhen, China, 518172
xiaozhufang@link.cuhk.edu.cn

Rui Jin
School of Science and Engineering
The Chinese University of Hong Kong
Shenzhen, China, 518172
ruijin@link.cuhk.edu.cn

Minghao Li
School of Science and Engineering
The Chinese University of Hong Kong
Shenzhen, China, 518172
minghaoli1@link.cuhk.edu.cn

Abstract

The new involution operator was proposed recently, but it was only implemented in the discriminative models. We first associate the involution with generative adversarial networks(GAN) to explore its effectiveness in image generation problems. We use the human face dataset CelebA. The experiments show the superior performance of involution GAN over deep convolution GAN and self-attention GAN. The generated images of involution has rather saturated and smooth colors. Besides, the hyperparameters (e.g.kernel size, group, and reduction ratio) are discussed to tuning the involution layer, where reduction size is found to play an essential role. Finally, the stability issues introduced by new involution in GAN are presented.

1 Introduction

Deep learning attracts large attention in the past decade. As a discriminative model, deep neural networks make the revolution on image classification, detection, segmentation. As for the generative model, variational autoencoders(VAE) were prevailing 8 years ago. In 2014, Ian Goodfellow first introduced an innovative model, generative adversarial networks(GAN)(1), completely redirecting the following researches about generative models.

GAN is primarily implemented on image generation. The adversarial is interpreted as the simultaneously trains between a generator to counterfeit fake images by inferring the distribution and a discriminator to distinguish the fake images from the generator and real images. GAN can be implemented with various networks. The famous operator convolution(2) and self-attention(3) both enhanced GAN's performance. A few months ago Li proposed a new operator, involution, demonstrating the high efficiency on Image Classification, and Object detection, and Segmentation(4). In this paper, we propose inserting involution into the typical GAN's structure, used for another problem Image Generation. We name new generative structure as involution GAN(iGAN).

The main contribution of this paper is (1) retrieving of new operator, involution, and explore its dependence on hyperparameters(e.g. reduction ratio, group, kernel size); (2) compare iGAN's performance with other GANs. (3) discuss iGAN's stability issues. The architecture of the paper is organized as follows. In *Methods*, we reiterate the theorem of involution. In *Experiments*, we

Paper Report of Final Project for CIE6032 (MDS6232)- Selected Topics in Deep Learning.

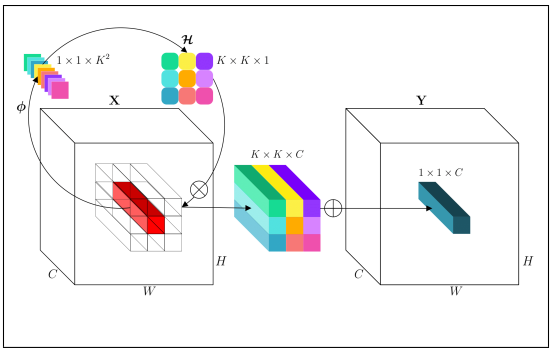


Figure 1: Schematic illustration of our proposed involution.(8)

elaborate the details of programming setting. In *Result*, we discuss iGAN's performance, stability, and its dependence on hyperparameters. The code of experiment is available at https://github.com/liminghao0914/involution_GAN.

2 Relate work

Refer to involution operator from Li's work(4), We add involution to GAN(1). To tackle GAN's unsuitability, many techniques are applied such as two time-scale update rule(TTUR)(5), Wasserstein generative adversarial networks(WGAN)(6), and WGAN with gradient penalty(WGAN-GP)(7). To evaluate the performance of iGAN, we compare the Fréchet inception distance(FID) of iGAN with deep convolution GAN(DCGAN)(2) and self-attention GAN(SAGAN)(3).

3 Methods

3.1 Involution

Involution was proposed by Li as a new innovative operator(4), functioning like convolution and self-attention. The motivation of involution is following. Most figures have latent connections among features, which are stored in channels. Let K denote kernel size, B, C, W, H denote batch size, channel, weight, and height. The involution kernel $\mathcal{H}_{i,j} \in \mathbb{R}^{K \times K \times 1}$ is calculated as follows.

$$\mathcal{H}_{i,j} = \phi(X_{i,j}) = W_1 \sigma(W_0 X_{i,j}) \tag{1}$$

where $X_{i,j}$ is the feature at location height i and weight j; $W_0 \in \mathbb{R}^{C \times \frac{C}{r}}$ and $W_1 \in \mathbb{R}^{K \times K \times \frac{C}{r}}$ are weight parameters to train; σ is the activation function. The hyperparameter r in W_0 and W_1 is the reduction ratio. Having calculated the involution kernel. The output $Y_{i,j}$ can be derived.

$$Y_{i,j} = \sum_{(p,q) \in \Omega} \mathcal{H}_{i,j,p,q} X_{p,q} \tag{2}$$

where Ω is the $K \times K$ block centering at (i, j). Fig.1 illustrates the pipeline of involution for one group. To step further, the channel C can be divided into several groups and each group can do involution by itself.

3.2 Convolution & self-attention

According to Fig.1, the specialty of involution is the kernel with consistency along channels, which is channel-agnostic. On the other hand, the involution kernel is spatial-specific, varying at different locations. These characteristics are reciprocal to convolution.

- Convolution. (1)spatial-agnostic: shares parameters in space to satisfy the shift invariant system; (2)channel-specific: exists redundant parameters in channel dimension.
- Involution. (1) spatial-specific: distinguishes kernel in space, but share the hyperparameters to generate kernel(like self-attention); (2)channel-agnostic: reduce the memory in channel to increase the spatial kernel size.

Involution is very close to self-attention. The motivation of self-attention is to acknowledge kernel's spatial variety but to explore the latent connection by Q, K, V. Similarly, involution explores the channel's latent connection by W_0, W_1. Instead of applying self-attention in H,W in Fig.1, involution intuitively applies attention mechanism in C. Given the typical expression of self-attention as follows.

$$Y_{i,j} = \sum_{(p,q)\in\Omega} ((XW^Q)(XW^K)^T)_{i,j,p,q}((XW^V))_{p,q} \tag{3}$$

Compared with Eqn.1, there is $\mathcal{H}_{i,j,p,q} \approx ((XW^Q)(XW^K)^T)_{i,j,p,q}$. The similarity is apparent in algebra.

3.3 Generative adversarial network

For WGAN-GP, we do not spare more time to elaborate on the derivation and consequence of loss and network structure. We select Wasserstein distance, also called earth move distance(EMD), for its superiority in convergence(7). Gradient penalty is used to enhance stability by enforcing Lipschitz continuity. Besides, the spectral normalization is demonstrate better than batch and layer normalization for GANs(9)(10). Especially for WGAN, we should be careful about the activation function and normalization, which affects the magnitude of Wasserstein loss.

3.4 Fréchet inception distance

The evaluation of GAN is also essential. Fréchet inception distance (FID) is firstly introduced by Heusel to make a quantitative evaluation for images generated by GANs(5).

The core idea for FID is using Fréchet Distance to test the similarity between model samples and real samples. Let $p(\cdot)$ be the distribution of model samples and $p_w(\cdot)$ the distribution of the samples from real world. The Fréchet distance $d(\cdot,\cdot)$ between the Gaussian with mean and covariance (m, C) obtained from $p(\cdot)$ and the Gaussian (m_w, C_w) obtained from $p_w(\cdot)$ is called the "Fréchet Inception Distance" (FID), which is given by:

$$d^2((m, C), (m_w, C_w)) = ||m - m_w||_2^2 + \mathrm{Tr}(C + C_w - 2(CC_w)^{1/2})$$

The smaller FID we get from the generative model, represent a better performance of the model to generate the simulation images, which means more "real" those generated images are.

4 Experiments

4.1 Dataset and Pre-processing

In this paper, we have used the human face dataset, CelebA cropped data, in order to check the effect of involution. Since the original size of CelebA is 256, we resized images to the same height and width to $3 \times 64 \times 64$. The total number of images for training is $202k$. In addition, We train MNIST and show generative images in **Appendix**.

4.2 GAN Structure

To distinguish the performance of iGAN, DCGAN, and SAGAN, we ensure the other parts of networks consistent as much as possible. The simplest DCGAN is the basic structure, and SAGAN adds an additional self-attention layer in both generator and discriminator. Likewise, iGAN adds an additional involution at the same location. Fig,2 illustrates the architecture of iGAN.

- **DCGAN**: DCGAN generator consists of five layers of transpose convolution with kernel size 4×4, which takes in a noise vector of dimension 128 and upsamples it to an image of size $64 \times 64 \times 3$. The activation map size is doubled at each layer, while the number of channels is halved (except for the last layer). Each layer except for the last one is followed by a ReLU nonlinearity, while the last layer's activation function is a tanh nonlinearity. Both spectral and batch normalization are applied before ReLU but left the last layer. In terms of the symmetric, DCGAN discriminator has five convolution layers, with 4×4 kernel size and two strides, one padding. The activation function is leakyReLU with slop 0.1, following the spectral norm.

Table 1: Training parameters for GAN models.

attribute	value	attribute	value
batch size	64	total steps	50000
λ(penalty)	10	latent dimension	128
optimizer	Adam	learning rate decay	0.95
β_1(Adam)	0	β_2(Adam)	0.9
generator Learning rate	0.0001	discriminator Learning rate	0.0004

- **SAGAN**: Before the last convolution layer of DCGAN's generator, we insert the self-attention, where the channel size is 64. Set the dimension of attention matrix W^Q, W^K equal to $C \times \frac{C}{8}$. The activation function of the attention is softmax. For symmetry, another self attention is added after the discriminator's first layer.

- **iGAN**: To mimic SAGAN, we also add involution at the last second layer. We select ReLU function as $\sigma(\cdot)$ in Eqn.1, following batch normalization. Other hyperparameters are discussed in *Result*.

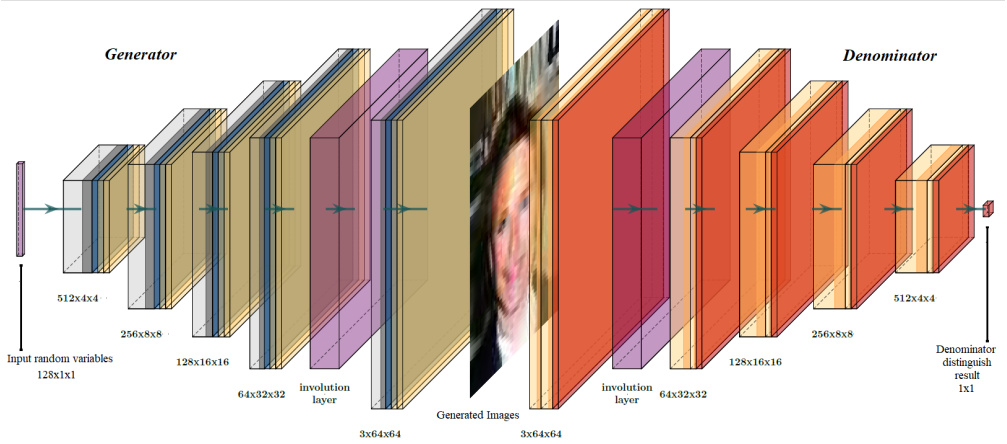


Figure 2: The network structure with iGAN as the generator and discriminator.

4.3 Training parameters

Table 1 summarizes the parameters we used for different GAN losses.It typically takes hours to train one model. Since the size of the dataset is large, We did not take large epochs due to the limited project time. Roughly the training stops at $50k * 64/200k \approx 16$ epochs, while we still obtained decent results. All experiments were carried on a machine equipped with a graphic card NVIDIA GeForce RTX 2080 SUPER, and CPU Intel(R) Core(TM) i7-9700K @3.60GHz. All experimental programs were implemented in PyTorch 1.7.1, with CUDA 11.3 enabled.

4.4 Evaluation

The performance of GANs is evaluated in the following three aspects. 1) Inspection: We inspect and compare the quality of generated images to subjectively judge the performance of GANs. 2) EMD loss: we observe and compare the tendency of loss to check convergence and stability. 3) FID: The relative distribution is based on the original dataset. 4) Memory&computation: they should be considered for the training in practice.

Figure 3: Fake images generated by iGAN(left),SAGAN(mid) and DCGAN(right). More images are in *Appendix*.

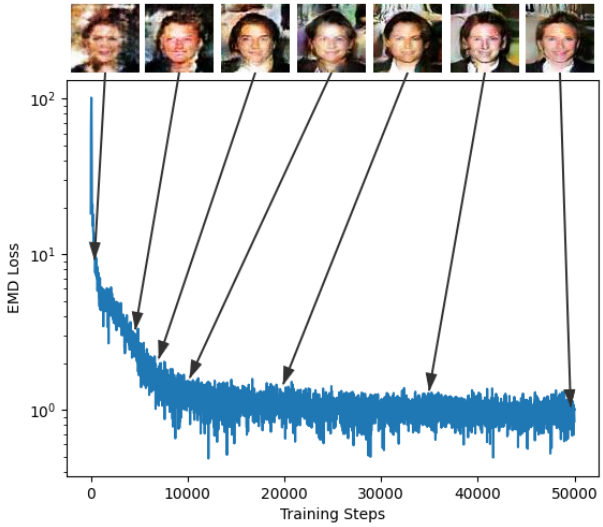


Figure 4: iGAN: the tendency of EMD loss over training steps and corresponding generated fakes images. The loss keeps slowly descending due to the decay of learning rate.

5 Result

5.1 Experiment 1: comparison of GANs

iGAN successfully generated the images, and the EMD loss curve shows in Fig.4. It also provides the representative output over the training. Obviously, with training going on, the generator's fake images are gradually approaching the authentic images. At the end of the training, the image in Fig.4 is hard to be distinguished from real ones by human eyes. According to iGAN's lowest FID score over the training in Fig.5(left), we conclude the superiority of iGAN over DCGAN and SAGAN. The generator's fake images show in Fig.3, and iGAN's images not only have the self-attention details like SAGAN, but their color also seems more saturated and smoother than SAGAN's images. It is because the RGB channel of images always has the same patterns, which reflects to the agnostic-channel of involution kernel.

5.2 Experiment 2: Hyperparameters' tunning

When Li first proposed involution, he found three important aspects affecting the performance of the involution layer, that is, involution kernel size, group of involution, and reduction ratio. We evaluate the performance from three sides, memory, computation, and accuracy. 1)Accuracy is equivalent to

Table 2: Hyperparameters of iGAN and its influence. Channel for the involution layer is C=64.

Case	kernel size	group	reduction ratio	Accuracy	Memory	Computation
#	K	G	R	min(FID)	#parameters	time per step
1	3× 3	4	2	61.51	800	0.154
2	7× 7	4	8	76.00	520	0.333
3	7× 7	4	2	112.85	2080	0.342
4	3× 3	4	8	76.01	200	0.152
5	3× 3	1	2	62.34	2336	0.152
6	1× 1	4	2	68.73	644	0.120
7	5× 5	4	2	63.37	1312	0.230

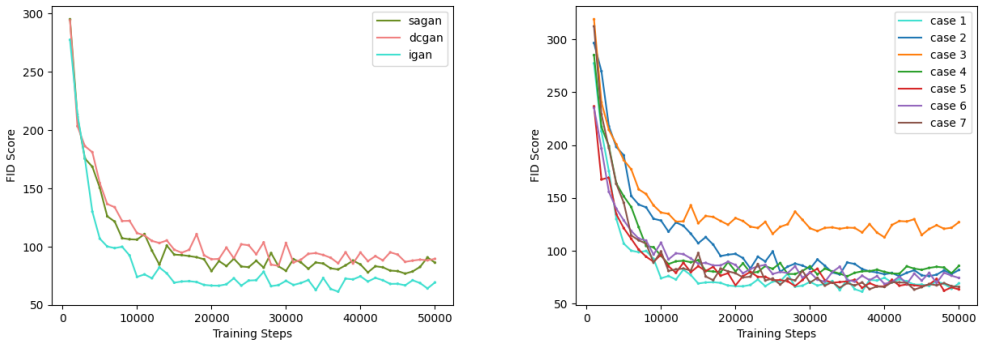


Figure 5: FID Score of different GANs(left) and of iGAN with various hyperparameters(right).

min(FID), the minimum value of FID curve(Fig.5-right). 2)Memory is estimated by the number of weight parameters of the involution layer, which is expressed by

$$\#parameters \propto (\frac{C}{G} + K * K) \times \frac{C}{R \times G} \times G \tag{4}$$

3)Computation is represented the total time cost of each steps while other layers are fixed. Table.2 presents totally seven cases we have simulated with different pair of values and the corresponding result. In short, case 1 is the best one with lower FID, memory and computation. Other conclusions can be generalized from Table.2.

- The best K is around 4×4, which is close to the common convlution kernel size. Larger K costs more computation and memory, but it does not arise FID as expected.
- The group G hardly affects FID but the larger one save more memory
- The reduction ratio R affects FID, but the trade off exists between the memory and FID.

5.3 Stability of iGAN

The stability and convergence is the key problem of GAN. Since we introduce the new involution layer, it must company with the stability issues. During our experiment, the collapse of models happens sometimes. The loss of unstable and here are the potential reasons as well as our strategies show in *Appendix*.

- In experiments, two-successive-involution layer(e.g. conv-inv-conv-inv-con block) is divergent- the gradient penalty keeps increasing over time. WGAN should be enforced by Lipschitz continuity, but the clipping method we add in involution does not help.
- The normalization is controversial for involution because it needs stability for one side, and it affects the EMD loss for another side. We tested spectral, batch normalization for the involution kernel, which all lead to the model collapse or instability.

The stability issue of new involution deserves further investigation in future research.

References

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[2] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[3] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *International conference on machine learning*. PMLR, 2019, pp. 7354–7363.

[4] D. Li, J. Hu, C. Wang, X. Li, Q. She, L. Zhu, T. Zhang, and Q. Chen, "Involution: Inverting the inherence of convolution for visual recognition," *arXiv preprint arXiv:2103.06255*, 2021.

[5] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *arXiv preprint arXiv:1706.08500*, 2017.

[6] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.

[7] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," *arXiv preprint arXiv:1704.00028*, 2017.

[8] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[9] S. Xiang and H. Li, "On the effects of batch and weight normalization in generative adversarial networks," *arXiv preprint arXiv:1704.03971*, 2017.

[10] K. Kurach, M. Lucic, X. Zhai, M. Michalski, and S. Gelly, "The gan landscape: Losses, architectures, regularization, and normalization," 2018.

Appendix

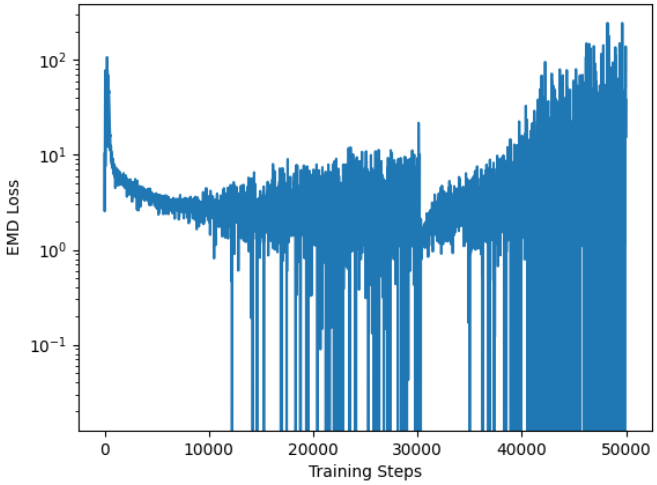


Figure 6: loss of collapsed model due to successive two involution layers

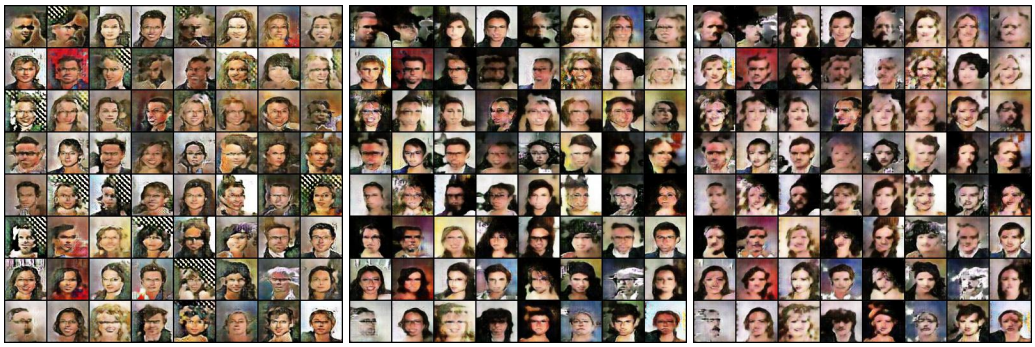


Figure 7: Genrative Figures by collapsed model due to successive two involution layers at different steps in one training.

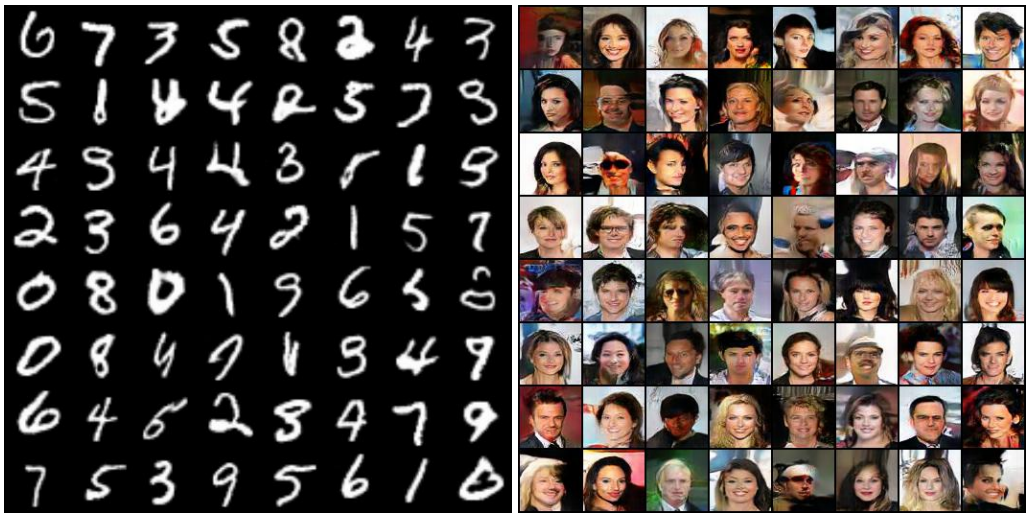


Figure 8: iGAN

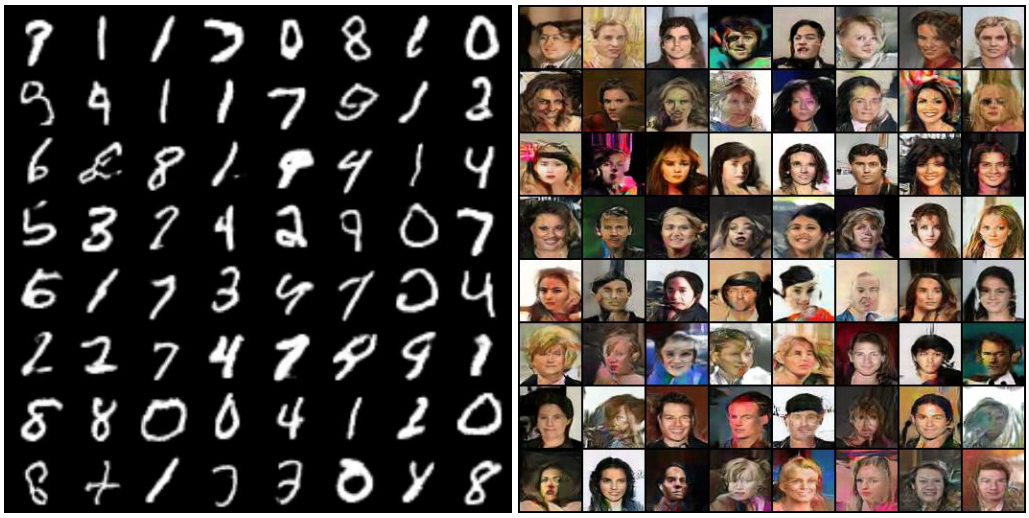
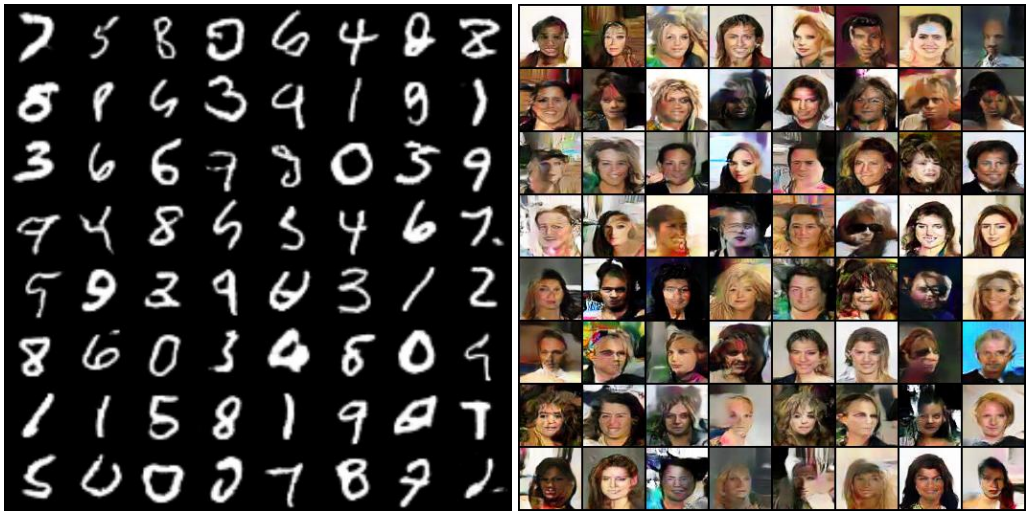


Figure 9: SAGAN



DCGAN

The matrix of generative fake images.