



## on Information and Systems

**VOL. E106-D NO. 5**

**MAY 2023**

**The usage of this PDF file must comply with the IEICE Provisions  
on Copyright.**

**The author(s) can distribute this PDF file for research and  
educational (nonprofit) purposes only.**

**Distribution by anyone other than the author(s) is prohibited.**

A PUBLICATION OF THE INFORMATION AND SYSTEMS SOCIETY



The Institute of Electronics, Information and Communication Engineers

Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

**LETTER** Special Section on Deep Learning Technologies: Architecture, Optimization, Techniques, and Applications**Enhanced Full Attention Generative Adversarial Networks**KaiXu CHEN<sup>†</sup>, Nonmember and Satoshi YAMANE<sup>†a)</sup>, Member

**SUMMARY** In this paper, we propose improved Generative Adversarial Networks with attention module in Generator, which can enhance the effectiveness of Generator. Furthermore, recent work has shown that Generator conditioning affects GAN performance. Leveraging this insight, we explored the effect of different normalization (spectral normalization, instance normalization) on Generator and Discriminator. Moreover, an enhanced loss function called Wasserstein Divergence distance, can alleviate the problem of difficult to train module in practice.

**key words:** Generative Adversarial Networks (GANs), wasserstein divergence, attention module, generative model

## 1. Introduction

Image synthesis is an important problem in computer vision. To generate realistic images, networks with deep layers recently have been proved beyond the other shallow approaches. In generative networks, the Generative adversarial network (GAN) [1] is a prevalent generative model. The Deep convolutional generative adversarial network (DCGAN) [2] aligns 2 deep networks to generate more quality adversarial examples such as images, text or video sequences. Deep networks have abundant filter banks to promote more detailed analysis and synthesis of large examples. These convolution layers are adjusted by the propagated gradients of cost functions.

Since the convolution operator has a local receptive field, long range dependencies can only be processed after passing through several convolutional layers. Attention module, on the other hand, exhibits a better balance between the ability to model long-range dependencies and the computational and statistical efficiency. The attention module calculates response at a position as a weighted sum of the features at all positions, where the weights – or attention vectors – are calculated with only a small computational cost. Self-Attention GAN (SAGAN) [3] introduces a self-attention mechanism into convolutional GANs [2]. Armed with self-attention, the Generator can draw images in which fine details at every location are carefully coordinated with fine details in distant portions of the image.

In this paper, we propose an improved model of GAN with full attention layer in Generator, which introduce an attention module into deep convolutional GANs with the

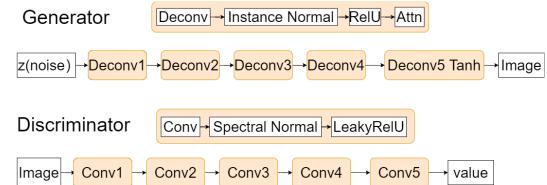


Fig. 1 Proposed network structure

improved wasserstein divergence. The attention module is complementary to convolutions and helps with modeling long range, multi-level dependencies across image regions. Then, we use the Wasserstein Divergence distance to get a stable training. We evaluate the proposed enhanced GAN with attention module across several datasets and obtained enlightening results. Our contributions can be summarized as follows:

- We propose an improved structure of GAN with full attention Generator.
- We use Wasserstein Divergence distance try to have a stable training.
- We have tried several different network structures, as shown in Fig. 1.
- We use different techniques to stabilise training.

## 2. Related Works

### 2.1 Generative Adversarial Networks

Generative models have been showing great success since the advent of GANs [1]. Since the advent of Alexnet [4], most of the focus has been put on CNNs as a discriminative model. Only until recently have generative models been more focus. The Deep Convolutional GANs (DCGAN) [2] propose and evaluate a set of constraints on the architectural topology that make them stable to train in most settings.

### 2.2 Non-Local Models

In order to sense global structures in a large receptive field, several convolutional layers and large kernel sizes exist typically in GAN-based structures. But it will seriously harms the computational efficiency. Recently, attention mechanisms have become an integral part of models that must capture global dependencies. It computes the response at one position as a weighted sum of the features at all positions,

Manuscript received March 25, 2022.

Manuscript revised August 23, 2022.

Manuscript publicized January 12, 2023.

<sup>†</sup>The authors are with the Kanazawa University, Kanazawa-shi, 920-1192 Japan.

a) E-mail: syamane@is.t.kanazawa-u.ac.jp

DOI: 10.1587/transinf.2022DLL0007

and is able to capture long-range dependence across different parts. Self-Attention GAN [3] learns to efficiently find global, long-range dependencies within internal representations of images.

### 2.3 Wasserstein Divergence

Wasserstein GANs [5] has shown that even in very simple scenarios the JS divergence does not supply useful gradients for the Generator. On the other hand, the EM distance does not suffer from these problems of vanishing gradients. However, it is very challenging to approximate the  $k$ -Lipschitz constraint required by the Wasserstein-1 metric (W-met). The Wasserstein Divergence [6] introduce a Wasserstein divergence objective for GANs, which can faithfully approximate W-div through optimization.

## 3. Techniques to Stabilize the Training of GANs

We also investigate some techniques to stabilize the training of GANs on challenging datasets. The spectral normalization [7] are used in this paper, we try different combinations regarding spectral normalization as well as instance normalization [8]. Meanwhile, the two-timescale update rule (TTUR) [9] specifically to address slow learning in the proposed method.

### 3.1 Spectral Normalization with Instance Normalization

Spectral Normalization [7] originally proposed stabilizing the training of GANs by applying spectral normalization to the Discriminator network. Doing so constrains the Lipschitz constant of the Discriminator by restricting the spectral normal of each layer. Instance Normalization [8] show how a small change in the stylization architecture results in a significant qualitative improvement in the generated images, it can be used to train high-performance architectures for real-time image generation.

In this paper we have tried different combinations of Spectral Normalization and Instance Normalization, the details of which are showed in Table 3.

### 3.2 Imbalanced Learning Rate for Generator and Discriminator Updates

In practice, methods using regularized Discriminators typically require multiple Discriminator update steps per Generator update step during training. Independently, TTUR [9] have advocated using separate learning rates for the Generator and the Discriminator for the problem of slow learning in a regularized Discriminator, making it possible to use fewer Discriminator steps per Generator step.

## 4. Proposed Method

Most GAN-based models for image generation are built using convolution layers. Convolution processes the information in a local neighborhood, thus using convolutional layers

alone is computationally inefficient for modeling long-range dependencies in images. We adapt the non-local model of [10] to introduce attention to the Generator part, enabling the Generator to efficiently model relationships between separated spatial regions.

Non-local Networks [10] prove that more non-local blocks in general lead to better results, multiple non-local blocks can perform long-range multi-hop communication. The SAGAN [3] prove that the self-attention mechanism at the middle-to-high level feature maps (*e.g.*,  $feat_{32}$  and  $feat_{64}$ ) achieve better performance than the models with the self-attention mechanism at the low level feature maps (*e.g.*,  $feat_8$  and  $feat_{16}$ ). In our network structure, we adapt non-local model to Generator in every layer, did not adapt it in the Discriminator. Because the non-local model will increase a lot of calculation, and the Discriminator has become more stable because of the wasserstein divergence distance used, so we think it is not necessary adapt into both Generator and Discriminator.

The default architectures for Generator and Discriminator of our method are showed in Fig. 1. Regarding the Generator, the input noise is a shape of [100x1x1] tensor. We constructed a 5-layer deconvolution block. Each block includes a Deconvolution layer, a Instance Normal layer [8], a ReLu activation layer, and finally with an attention module. Note that at the end of the fifth layer we add the Tanh activation layer. Regarding the Discriminator, the input is the image information in the shape of (channels x image size x image size), here we choose the image size of 64. We construct a 5-layer convolutional block. Each block includes a convolutional layer, a Spectral Normal layer [7], and a Leaky Relu layer. The discriminator output is a score about the input image information. More detailed information in each layer of the proposed network is given in Table 1.

Up to now, there are many other papers on the application of attention in GANs ([11]–[14]). But we first propose the approach of using the attention module only in each layer in Generator. In addition, there are many papers discussing the application of different normalization layers in GAN ([15]–[17]). We focus on the impact of Spectral Normalization and Instance Normalization in GAN, espe-

**Table 1** The default architecture of SAGAN-div for 64x64 image generation

Generator	Kernel size	Stride	Padding	Output shape
Noise	-	-	-	100x1x1
DeConv	[4x4]	1	0	512x4x4
DeConv	[4x4]	2	1	256x8x8
DeConv	[4x4]	2	1	128x16x16
DeConv	[4x4]	2	1	64x32x32
DeConv, tanh	[4x4]	2	1	channelsx64x64
Discriminator				
input	-	-	-	channelsx64x64
Conv	[4x4]	2	1	64x32x32
Conv	[4x4]	2	1	128x16x16
Conv	[4x4]	2	1	256x8x8
Conv	[4x4]	2	1	512x4x4
Conv	[4x4]	1	0	1

cially on the final generated images when attention module is placed in the Generator part.

## 5. Experiments

### 5.1 Evaluation Metrics

We choose the Frechet Inception distance (FID) [9] for quantitative evaluation. FID is a more principled and comprehensive metric, and has been shown to be more consistent with human evaluation in assessing the realism and variation of the generated samples [9]. FID calculates the Wasserstein-2 distance between the generated images and the real images in the feature space of an Inception-v3 network. Lower FID values mean closer distances between synthetic and real data distributions. In all our experiments, 10k samples are randomly generated for each model to compute the Inception score, FID. About the FID code implement, we use the source code from the FID score for PyTorch [18], it's a port of the official implementation of Fréchet Inception Distance to PyTorch.

### 5.2 Network Structures and Implementation Details

The default architecture for Generator and Discriminator of our method as presented in Table 1 and Fig. 1, and a detailed description is provided in Sect. 4. All the proposed method we trained are designed to generate 64x64 pixel images. For all models, we use the Adam optimizer [19] with  $\beta_1 = 0$  and  $\beta_2 = 0.9$  for training. By default, the learning rate for the Discriminator is set to 0.0004 and the learning rate for the Generator is set to 0.0001, where from [9]. The number of training epochs are 10,000 epochs for MNIST and FASHION-MNIST. Since the attention module was added to the Generator to determine a better way for Discriminator to discriminate and Generator to perform generative learning, we use a large epoch to ensure that the model was fully fitted on the dataset. We saved the highest rated model for final result comparison. By cross validation we determine the number of iterations for Discriminator per training step to 5 for the two dataset.

## 6. Results

We compare our SAGAN-div to the state-of-the-art DCGAN [2], WGAN-div [6], SNGAN [7]. For each method, we apply the default architectures and hyperparameters recommended by their papers. We recalculated FID score with the above said tool, FID score fo PyTorch [18], and the results of the implementation as presented in Table 2.

We can see that with the result of the MNIST dataset, our method is better than WGAN-div [6], but is worse than SNGAN [7]. We think is due to the fact that only Spectral Normal is used in Discriminator without in Generator. With the result of the FASHION-MNIST dataset, our result are close to WGAN-div [6], but at the e part result in Fig. 3, the

**Table 2** FID score comparison between SAGAN-div and the state-of-the-art methods.

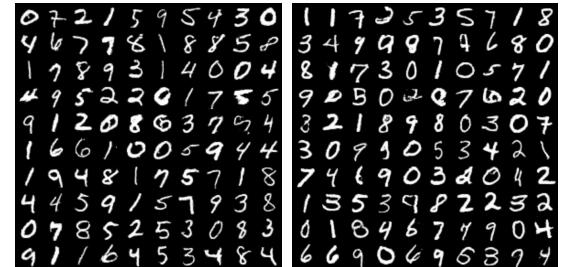
Method	MNIST	FASHION-MNIST
DCGAN [2]	45.7734	86.5129
WGAN-div [6]	83.1224	104.1503
SNGAN [7]	26.0231	46.9646
<b>SAGAN-div (best score) [d]</b>	<b>55.6425</b>	<b>100.8650</b>

**Table 3** FID score comparison with different combination of Spectral Normalization and Instance Normalization.

Method	MNIST	FASHION-MNIST
attnG+SN+IN [a]	86.7291	127.7808
attnG+SN+noIN [b]	59.8766	107.4283
attnG+noSN+IN [c]	101.1107	153.7099
<b>attnG+SNinD+INinG [d]</b>	<b>55.6425</b>	<b>100.8650</b>



(a) real images



(b) attnG+SN+IN



(c) attnG+SN+noIN



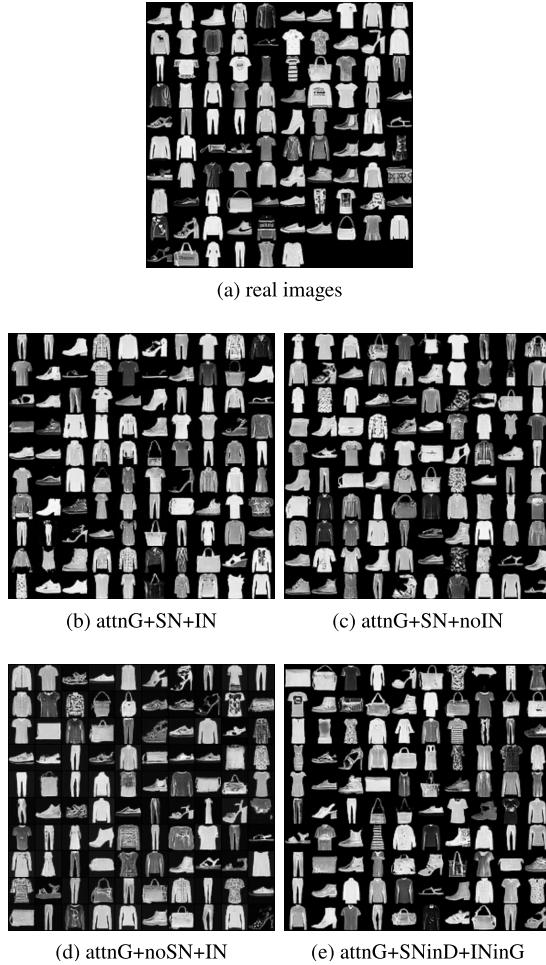
(d) attnG+noSN+IN

(e) attnG+SNinD+INinG

**Fig. 2** Comparisons of images generated by different network structure, use the MNIST dataset.

edges of the generated garments are clear, and the outlines of the more difficult parts (high heels) are also well generated.

In this experiment, for stable training, we also tried different combinations of Spectral Normalization and Instance Normalization. We would like to provide a discussion about how to use the different normalization layer with our proposed network structure, self-attention module with Gener-



**Fig. 3** Comparisons of images generated by different network structure, use the FASHION-MNIST dataset.

ator, as well as to verify their effectiveness.

We offer four different structures, which are, **SN with IN in D and G, with SN without IN in D and G, with IN without SN in D and G, SN used in D with IN used in G.** The results of the FID are given in Table 3.

We can see that, the best combination is SN in Discriminator and IN in Generator. Using SN and IN at the same time will increase the calculation parameters and the result is not very well (Table 3 [a]). Just using IN will make the final result bad (Table 3 [c]). The result of using SN in Discriminator and Generator (Table 3 [b]) is close to the result of using SN in Discriminator and IN in Generator (Table 3 [d]), but the latter combination is significantly better. So, we deduce that the best combination of using self-attention block in Generator is to use Spectral Normalization in Discriminator and Instance Normalization in Generator.

The result images with different dataset are showed in Fig. 2 and Fig. 3. Compared with real image, from the angle visible to the naked eye, it looks very much like the real images. Especially for the FASHION-MNIST dataset, regarding the more complex parts (heels, patterns of clothes, etc.), our proposed network can generate clearer outlines,

as well as synthetic images that can be distinguished by the naked eye. Meanwhile we greatly stabilize training with the use of Wasserstein divergence distance. The result of simulation experiments shows that this model can control image generation and have stable training.

## 7. Conclusion

In this paper, we proposed SAGAN-div, which incorporates a self-attention mechanism into the Generator. Meanwhile, the effect of different normalization on the generated image results in the structure of our proposed method is identified. Moreover, the Wasserstein Divergence distance offers a more stable training environment as well as higher quality samples. Our proposed method, SAGAN-div achieves the great performance on image generation on MNIST and FASHION-MNIST dataset.

## References

- [1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” Advances in neural information processing systems, vol.27, 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,” International conference on machine learning, pp.7354–7363, PMLR, 2019.
- [4] A. Krizhevsky, I. Sutskever, and G.E. Hinton, “Imagenet classification with deep convolutional neural networks,” Advances in neural information processing systems, vol.25, pp.1097–1105, 2012.
- [5] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” International conference on machine learning, pp.214–223, PMLR, 2017.
- [6] J. Wu, Z. Huang, J. Thoma, D. Acharya, and L. Van Gool, “Wasserstein divergence for gans,” Computer Vision – ECCV 2018, Lecture Notes in Computer Science, vol.11209, pp.673–688, Springer International Publishing, Cham, 2018.
- [7] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” arXiv preprint arXiv:1802.05957, 2018.
- [8] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Instance normalization: The missing ingredient for fast stylization,” arXiv preprint arXiv:1607.08022, 2016.
- [9] 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,” Advances in neural information processing systems, vol.30, 2017.
- [10] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” Proc. IEEE conference on computer vision and pattern recognition, pp.7794–7803, 2018.
- [11] Z. Mi, X. Jiang, T. Sun, and K. Xu, “Gan-generated image detection with self-attention mechanism against gan generator defect,” IEEE Journal of Selected Topics in Signal Processing, vol.14, no.5, pp.969–981, 2020.
- [12] Z. Yuan, M. Jiang, Y. Wang, B. Wei, Y. Li, P. Wang, W. Menpes-Smith, Z. Niu, and G. Yang, “Sara-gan: Self-attention and relative average discriminator based generative adversarial networks for fast compressed sensing mri reconstruction,” Frontiers in Neuroinformatics, vol.14, p.611666, 2020.
- [13] H. Lan, A.W. Toga, and F. Sepehrband, A.D.N. Initiative, et al., “Scgan: 3d self-attention conditional gan with spectral normalization for multi-modal neuroimaging synthesis,” bioRxiv, 2020.

- [14] H. Lan, A.D.N. Initiative, A.W. Toga, and F. Sepehrband, “Three-dimensional self-attention conditional gan with spectral normalization for multimodal neuroimaging synthesis,” *Magnetic Resonance in Medicine*, vol.86, no.3, pp.1718–1733, 2021.
  - [15] S. Bera and P.K. Biswas, “Noise conscious training of non local neural network powered by self attentive spectral normalized markovian patch gan for low dose ct denoising,” *IEEE Trans. Med. Imag.*, vol.40, no.12, pp.3663–3673, 2021.
  - [16] M.A.-N.I. Fahim and H.Y. Jung, “A lightweight gan network for large scale fingerprint generation,” *IEEE Access*, vol.8, pp.92918–92928, 2020.
  - [17] K. Kurach, M. Lucic, X. Zhai, M. Michalski, and S. Gelly, “The gan landscape: Losses, architectures, regularization, and normalization,” 2018.
  - [18] M. Seitzer, “pytorch-fid: FID Score for PyTorch,” *WebPage*, Aug. 2020. Version 0.2.1.
  - [19] D.P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
-