Improved Performance of GANs via Integrating Gradient Penalty with Spectral Normalization

Hongwei Tan[1,2], Linyong Zhou[2], Guodong Wang[1], and Zili Zhang[1,3](✉)

[1] School of Computer and Information Science, Southwest University, Chongqing 400715, China
zhangzl@swu.edu.cn
[2] School of Mathematics and Statistics, GuiZhou University of Finance and Economics, Guiyang 550025, China
[3] School of Information Technology, Deakin University, Locked Bag 20000, Geelong, VIC 3220, Australia

Abstract. Despite the growing prominence of generative adversarial networks (GANs), improving the performance of GANs is still a challenging problem. To this end, a combination method for training GANs is proposed by coupling spectral normalization with a zero-centered gradient penalty technique (the penalty is done on the inner function of Sigmoid function of discriminator). Particularly, the proposed method not only overcomes the limitations of networks convergence and training instability but also alleviates the mode collapse behavior in GANs. Experimentally, the improved method becomes more competitive compared with some of recent methods on several datasets.

Keywords: Generative Adversarial Networks · Gradient penalty · Spectral normalization · Training stability · Networks convergence

1 Introduction

Generative Adversarial Networks (GANs) [10] are powerful deep generative models which can be used to learn complex probability distributions. Especially in image research, GANs have been successfully applied to a variety of tasks, including image generation [21,30], image super-resolution [5,16], image-to-image translation [15], image in-painting [37], domain adaptation [35] and many more.

However, while very powerful, GANs are known to be notoriously hard to train. To improve the performance of GANs, the general strategies for stabilizing training are to carefully design the model, such as by crafting the network architectures [16,30,31], by modifying the objective functions [2,21], by

Sopported by the National Natural Science Foundation of China: Managing Uncertainty in Service Based Software with Intelligent Adaptive Architecture (No. 61732019).

changing the gradient update modes [13,23,25] and by implementing the penalty or regularization techniques [11,24,34]. Despite practical advances, the performance of GANs still has plenty of place for improvement, especially in stability and convergence.

In this study, we integrate spectral normalization with a zero-centered gradient penalty technique to improve the performance of GANs, which the coalition can either demonstrably improve the stability and convergence of model or effectively alleviate the mode collapse behavior in GANs. Due to the fact that the update dynamic of discriminator network comes completely from the inner function of Sigmoid function, we find that the penalty is more effectively implemented on the inner function than directly done on the discriminator. Meanwhile, the Lipschitz constant of discriminator is leveraged to prevent the expansion of the model gradient. In addition, a training trick is introduced, clipping the gradient norm of network weights, which is conducive to further boosting the training stability. On the other hand, to achieve better convergence, spectral normalization (SN) [24] is added into the discriminator, along with batch normalization (BN) [14] in the generator. This amounts to implementing the penalty on SNGAN [24]. Unlike the original SNGAN, a modified GAN model is trained with spectral normalization. By doing so, our model captures the optimal networks convergence, particularly in the discriminator, it almost converges to a constant. In the experiments, the overall trend of the gradient variation is introduced to reflect the stability of GANs training. At the same time, the results reveal that our method leads to GANs training stability, good networks convergence and improving the quality of generated samples.

In summary, our contributions are as follows:

- By integrating the zero-centered gradient norm penalty on the inner function of Sigmoid function of discriminator with spectral normalization, a method is crafted to improve the performance of GANs.
- A modified SNGAN is introduced, which can demonstrably boost performance. As a training trick, we find that appropriately clipping the gradient norm of network weights assists in improving the stability of GANs training.
- We leverage the overall trend of the gradient variation to mirror the stability of GANs training, where the gradient variations are computed by the average gradient L_2 norm with a batch size samples in each generator update.

The rest of this paper is organized as follows: Sect. 2 introduces the background and related work. Section 3 provides some theoretical underpinnings for our method and proposes a gradient penalty method on a modified SNGAN model. Section 4 examines the performance of the proposed method via a series of experiments on synthetic and benchmark datasets. Finally, the conclusions are drawn in Sect. 5.

2 Background and Related Work

2.1 Backgroud

GANs [10] form a broad class of generative models in which a min-max two-player game is played between a generative network $G(\boldsymbol{z})$ and discriminative network $D(\boldsymbol{x})$ whose purpose, respectively, is to map random noise to samples and discriminate real and generated samples. Formally, the GAN objective [10] involves finding a Nash equilibrium to the following min-max problem:

$$\min_{G} \max_{D} \mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z}[\log(1 - D(G(\boldsymbol{z})))], \tag{1}$$

where p_{data} and p_z denote real data distribution (target distribution) and latent distribution (prior distribution such as $N(\mathbf{0}, \boldsymbol{I})$ or $U[-1, 1]$), respectively. According to (1), the loss functions of GANs discriminator network and generator network are as follows:

$$L_D = -\mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log D(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z}\sim p_z}[\log(1 - D(G(\boldsymbol{z})))], \tag{2}$$

$$L_{G_1} = \mathbb{E}_{\boldsymbol{z}\sim p_z}[\log(1 - D(G(\boldsymbol{z})))] \quad (Saturating), \tag{3}$$

$$L_{G_2} = -\mathbb{E}_{\boldsymbol{z}\sim p_z}[\log D(G(\boldsymbol{z}))] \quad (Non - Saturating), \tag{4}$$

where L_{G_1} and L_{G_2} denote the saturating and non-saturating loss function, respectively. From a more pratical standpoint, L_{G_2} makes network training more stable than L_{G_1} [5,8,10,16,34].

2.2 Related Work

Several recent work have focused on addressing the instability and convergence to improve the performance of GANs, where the gradient penalty-based methods are one of the most effective methods. WGAN-GP [11] first used the gradient norm to penalize the criterion function of WGAN [2], which effectively alleviated the limitation of Lipschitz condition in WGAN and significantly improved the stability of WGAN. However, Mescheder et al. [22] proved that the zero-centered gradient penalty converges more easily than the 1-centered gradient penalty (WGAN-GP is a typical 1-centered gradient penalty). Along this line of research, WGAN-LP [29] (WGAN based on the zero-centered gradient penalty) and GAN-0GP [34] (GAN based on the zero-centered gradient penalty) were proposed, respectively. Our proposed method falls under the same category, hopefully provides some context for understanding some of these methods. Specifically, our penalty is done on the inner function of Sigmoid function of discriminator rather than directly penalized on the discriminator as above mentioned methods.

From the optimization perspective, several normalization techniques commonly applied to deep neural networks training have been applied to GANs, such as batch normalization (BN) [14,30], layer normalization(LN) [11] and spectral normalization(SN) [24]. Generally, the BN and LN are simultaneously operated on discriminator and generator, while the SN is only done on discriminator in GANs. In the study, normalization is executed on a modified model (Table 1) based on SNGAN [24], which its discriminator and generator are normalized with the SN and BN, respectively. More importantly, a combination model, the modified model in cooperation with the zero-centered gradient norm penalty (on the inner function), can make the networks more stable and better convergence.

3 Method

In this section, we will lay out the theoretical groundwork for our proposed method. In addition, a training trick (clipping the gradient norm of network weights) and a modified SNGAN will be introduced. Finally, our method will be formally proposed.

3.1 Controlling Gradient Variation

The instability of GAN training is mainly caused by its gradient update instability, while the gradient information of the generator is transmitted by the discriminator gradient [1]. Thus, controlling the gradient update of discriminator can effectively control the instability of GANs training. In image generation, let $\boldsymbol{X} = Supp(\mathbb{P}_{data}) \cup Supp(\mathbb{P}_g)$, where $\mathbb{P}_{data}$ and $\mathbb{P}_g$ denote the real distribution and generative distribution, respectively. Suppose m, n denote the height and width of input image, respectively, and M is the maximal second order derivative of the loss function L_D (2) in $\boldsymbol{X}$, and then $| \min(\nabla L_D) - \max(\nabla L_D) | \leq M \cdot L\sqrt{12mn}$, where L is the Lipschitz constant of discriminator network [27]. According to this, controlling the Lipschitz constant can effectively control the gradient variation amplitude of discriminator and make GANs training stable. Despite controlling the Lipschitz constant of network is not easy (in fact, even for the two-layer neural networks, the exact computation of the quantity is NP-hard problem [36]), the following theorem [12] provides a feasible scheme.

Theorem 1. *Let $F : \mathbb{R}^n \longrightarrow \mathbb{R}$ be a C^1-function. Then $F(\boldsymbol{x})$ is Lipschitz function with the Lipschitz constant L for all $\boldsymbol{x} \in \mathbb{R}^n$ if and only if $\| \, \|\nabla F\|^2 \, \|_\infty \leq L^2$.*

Theorem 1 (proof see [12] page 73, Lemma 4.3) provides an approach to determine that $F(\boldsymbol{x})$ is Lipschitz function with Lipschitz constant L, namely, when the square of maximum gradient norm of $F(\boldsymbol{x})$ is less than or equal to the square of a constant L, and then $F(\boldsymbol{x})$ is a Lipschitz function. In addition, the theorem also gives the fact: the Lipschitz constant can control the expansion of gradient. To this end, the loss function of discriminator network (2) is transformed into the following penalty form:

$$L'_D = L_D + \lambda \mathbb{E}_{\boldsymbol{x} \sim p_{data}}[\max\{\|\nabla D(\boldsymbol{x})\|^2, L^2\}], \tag{5}$$

where λ and L are the penalty coefficient and the Lipschitz constant of the discriminator network. According to Theorem 1, this penalty form can effectively control the variation amplitude of discriminator gradient and make GANs gradient update stable.

3.2 Penalizing the Inner Function of Sigmoid

Let $D(\boldsymbol{x}) = Sigmoid(f(\boldsymbol{x}))$, where $Sigmoid(\cdot)$ is the last layer activation function of discriminator. Accordingly, the min-max optimization problem (1) is transformed into the following form:

$$\min_{G} \max_{f} \mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log Sigmoid(f(\boldsymbol{x}))] + \mathbb{E}_{\boldsymbol{z}\sim p_z}[\log(1 - Sigmoid(f(G(\boldsymbol{z}))))]. \quad (6)$$

The loss functions of discriminator and generator, L_D and L_{G_2}, are also replaced correspondingly, and their gradients can be easily proven that

$$\nabla_\phi L_D = -\mathbb{E}_{\boldsymbol{x}\sim p_{data}}[(1 - D(\boldsymbol{x}))\nabla_\phi f(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[D(\boldsymbol{x})\nabla_\phi f(\boldsymbol{x})], \quad (7)$$
$$\nabla_\theta L_G = -\mathbb{E}_{\boldsymbol{z}\sim p_z}[(1 - D(G(\boldsymbol{z})))f(G(\boldsymbol{z}))\nabla_{\boldsymbol{x}} f(G(\boldsymbol{z}))\boldsymbol{J}_\theta G(\boldsymbol{z})], \quad (8)$$

where ϕ, θ, $\boldsymbol{J}_\theta G(\boldsymbol{z})$ denote the discriminator network parameters, the generator network parameters and the Jacobi matrix of $G(\boldsymbol{z})$ with respect to θ, respectively. The gradients (7), (8) imply that the update dynamic for GANs training is completely provided by the gradient of the function $f(\boldsymbol{x})$, namely the gradient of the inner function of Sigmoid function. Due to the Lipschitz constant of Sigmoid function is always 0.25, thus controlling the Lipschitz constant of the function $f(\boldsymbol{x})$ is equivalent to controlling the Lipschitz constant of discriminator $D(\boldsymbol{x})$. In view of this, the penalty form of loss function L_D and the loss function L_{G_2} will be transformed into the following form:

$$L_D^* = L_D + \lambda\mathbb{E}_{\boldsymbol{x}^*\sim p^*}[\max\{\|\nabla f(\boldsymbol{x}^*)\|^2, L^2\}], \quad (9)$$
$$L_{G_2} = -\mathbb{E}_{\boldsymbol{x}\sim p_g}[\log(Sigmoid(f(\boldsymbol{x})))], \quad (10)$$

where $\boldsymbol{x}^* = t\boldsymbol{x} + (1-t)\boldsymbol{y}, t \sim U(0,1), \boldsymbol{x} \in Supp(\mathbb{P}_{data})$, $\boldsymbol{y} \in Supp(\mathbb{P}_g)$, p^* is the mixed distribution of real distribution and generated distribution. The loss functions L_D^* (discriminator) and L_{G_2} (generator) are adopted throughout this paper, where L_D^* is the loss function of the zero-centered gradient norm penalty with regard to the inner function of Sigmoid function. It is worth emphasizing that the procedure of our algorithm is similar to the algorithm 1 of [10], except for the loss functions of discriminator and generator.

3.3 Exploring the Optimal Model

In this part, an optimal model will be explored, including model structure, clipping the gradient norm of network weights, optimization method. To do this, we expound some comparative cases via the experimental results on CIFAR10 [18], where the number of updates for the generator are 100k.

A Modified SNGAN. Essentially, SNGAN [24] imposes global regularization on the network, which the discriminator was normalized with the SN, along with the BN in generator. In contrast, SN has an advantage over BN and LN in GANs, such as more efficient gradient flow, more stable optimization [19,24]. However, we experimentally find that a modified SNGAN is more effective than original SNGAN in image generation. The differences between two models can be seen from Table 1. The two network structures have not been changed (see Appendix B), we just modified some hyper parameter settings and optimization method based on SNGAN. The modification is simple yet effective. The experimental *FID* [13] value of SNGAN on CIFAR10 is 28.66 (the original SNGAN is 20.70

with hinge loss [24], the BCE loss is used here), while the modified model is 24.53 (the smaller FID value is better). Also, the proposed penalty technique (9), (10) was implemented on the two models, the former FID is 20.12 and the latter is 12.48. The preliminary results show that the modified SNGAN is more effective than the original SNGAN.

Table 1. The hyper parameter setting and optimization method for two models.

Methods	Init	LR	n-dis	Bias	Optimizer
SNGAN	D-Norm G-Norm	0.1	5	T	Adam
Modified SNGAN	D-Orth G-Xavier	0.2	1	T-F-T	OAdam

"Init" is initialization method, which the discriminator and generator in SNGAN are both initialized by normal random number ($N(0,1)$), while the discriminator and generator in the modified SNGAN is initialized by orthogonal [32] and Xavier normal [9], respectively; "LR" is learning rate that uses to the activation function LeakyReLU of discriminator; "n-dis" denotes the numbers of update of the discriminator per one update of the generator; "Bias = T" denotes all biases are true, whereas "Bias = T-F-T" denotes all of the biases of layer are false except for the first layer and last layer.

Clipping the Gradient Norm of Network Weights. In order to further improve the performance of GANs, a training trick, clipping the gradient norms of network weights, is introduced into model training. The norms are computed over all gradients together, as if they were concatenated into a single vector. This process is computationally light and easy to incorporate into existing models, which the upper limit of the clipped norms is a controllable hyper parameter (max-norm) and the lower limit is zero. Note that the operation is performed in discriminator and generator, respectively. We test the different parameters max-norm on the proposed model with the modified SNGAN. Clearly, the parameter configuration E in Table 2 is the optimal combination. As shown in Table 2, it is conducive to appropriately clip the gradient norms of network weights for boosting the quality of the generated samples (FID). Note that the hyper parameter max-norm of discriminator and generator are α and β, respectively. In this study, the parameter combination E is used in all experiments.

Table 2. Comparison of the quality of the generated samples (FID, the smaller is better) based different hyper parameter settings on CIFAR10.

Settings	A	B	C	D	**E**
α	0.001	0.0001	0.1	1	**0.01**
β	0.02	0.0002	0.2	1	**1**
FID	14.93	16.34	15.33	13.25	**10.74**

The experimental results on CIFAR10 confirmed that the performance of GANs is significantly improved by our proposed zero-gradient norm penalty model on the modified SNGAN. In next section, we will further verify the stability and convergence of the proposed model on several datasets. It is worth emphasizing that we have tried different optimization algorithms (such as Lookahead [39]+Adam [17], Lookahead+OAdam [7]) and regularization methods on generator (such as orthogonality regularization [3], the modified orthogonality regularization [5] and group sparse regularization [33]), but none of them go beyond our method.

4 Experimental Results

In this section, the efficacy of our approach will be tested by investigating network convergence, training stability and FID value. Note that, in contrast with IS score [31], the FID value can more comprehensively measure the quality of the generated samples in GANs [4,13].

In order to verify the performance of our algorithm in the abovementioned three aspects, we conducted a set of extensive experiments of unsupervised image generation on CIFAR10 [18], SVHN (unlabeled subset) [26], STL10 [6], CelebA [20] and LSUN (bedroom) [38] datasets. Note that comparison of the networks convergence and training stability are arranged in 4.2, and the FID value in 4.3. In addition, the experiments on two synthetic datasets (8 Gaussians and 25 Gaussians) were performed to investigate the mode collapse behavior in GANs (in 4.1). We also compared our method with the representative ones. Unless otherwise noted, all of the results were obtained by PyTorch [28], where the numbers of update for GANs generator are 100k for all experiments. All codes can be found in https://github.com/thwgithub/GAN-Integration-GP-SN/.

4.1 Mixture of Gaussians

The mode collapse behavior in GANs can seriously affect the performance of GANs. To illustrate the effect of our method for alleviating the mode collapse phenomenon in GANs, a simple model was trained on two 2D mixture of Gaussians datasets (8 Gaussians arranged in a circle and 25 Gaussians in a square). Some technical details are relegated to Appendix A, including network architectures, hyper parameters and description of datasets. The experimental results are shown in Fig. 1. Compared with the original GAN, the least mode collapse behavior is demonstrated by our method, especially in 8 Gaussians dataset, 8 modes are almost learned in Fig. 1(b).

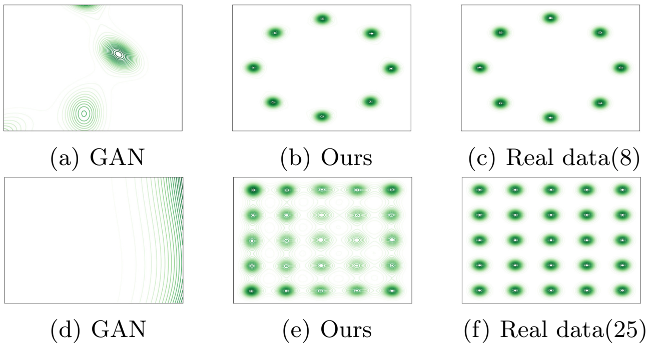

Fig. 1. Testing the mode collapse behavior on two Gaussians datasets ((a) GAN [10] on 8 Guassians; (b) our method on 8 Guassians; (c) real dataset; (d) GAN on 25 Guassians; (e) our method on 25 Guassians; (f) real dataset).

4.2 Results on Benchmark Datasets

In this subsection, we will report the network convergence and training stabiliy of the proposed method on five benchmark datasets(CIFAR10, SVHN, STL10, CelebA and LSUN(bedroom)), which all of the input images were both cropped to $3 \times 32 \times 32$. Due to space limitations, we only exhibit the results on CIFAR10 here and the other results can be found in the supplementary materials (it can be found in https://github.com/thwgithub/Sup-Materials-for-KSEM2020). We also compare against those of other congeneric gradient norm penalty algorithmss, including: WGAN-GP [11], WGAN-LP [29] and GAN-0GP [34].

For the hyper parameters setting, except for using the hyper parameters of the modified SNGAN in Table 1, we set the penalty coefficient λ to 10, as suggested in [11] and set the Lipschitz constant L at 0. The parameter max-norm of clipping weight gradient norm is set to 0.01 in discriminator and 1 in generator (the settings of Table 2). The learning rates of two networks are both 0.0002 with batchsize 64 and latent variable $\boldsymbol{z} \sim N(\mathbf{0}, \boldsymbol{I}_{128})$. As for the architecture of the generator and discriminator, we use convolutional neural networks (CNN) that more details is described in Appendix B.

Networks Convergence. As for the network convergence, the results of the experiment on CIFAR10 are illustrated in Fig. 2. Clearly, our method is superior to the other three methods in either discriminator (Fig. 2(a)) or generator (Fig. 2(b)). In fact, we do not care about the convergent value of the loss function (GANs) or the critic function (WGANs), only focus on the amplitude of their oscillations. For the visualization, Fig. 2 is vertically shifted. As shown in Fig. 2, both WGAN-GP and WGAN-LP get stuck in a bad convergence, GAN-0GP is greater improvement for convergence than the former two. While the convergence of our approach (blue curve) significantly outperforms the convergence of the others. It is noted that the update rate of our method is 1:1, that is, the discriminator updates one time per one time update of the generator, whereas the update rate of the others is 1:5.

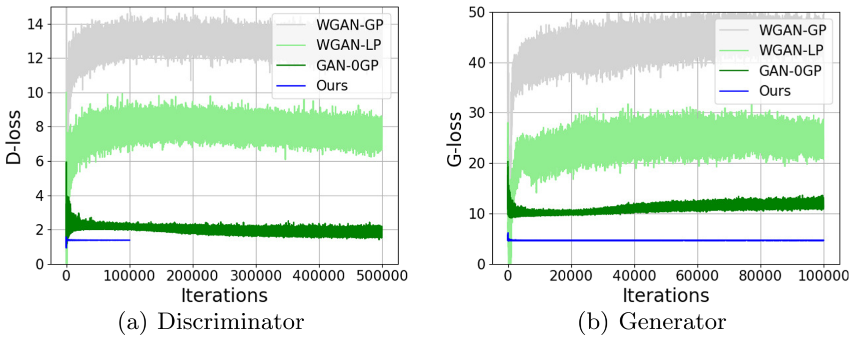

Fig. 2. Comparison of the convergence among four GANs methods ((a) the convergence of discriminator; (b) the convergence of generator).

Training Stability. In Fig. 3, the stability of GANs training in discriminator (Fig. 3(a)) and generator network (Fig. 3(b)) are exhibited. The overall trend of the gradient variation was used to mirror the stability of training, where the gradient variations were computed by the average gradient L_2 norm with a batch size samples in each generator update. To the best of our knowledge, the method is applied to measure the stability of GANs training for the first time. As observed, WGAN-GP and WGAN-LP both have a large gradient oscillation, this means that the two methods suffer from training instability. Moveover, their gradient variations are similar, it is probably because both algorithms belong to WGANs algorithms. For the GAN-0GP, the gradient behavior is relatively stable at the beginning of the training. However, with the increasing number of training, the method performs poorly. In contrast, our method (blue curve) is even more stable than the other methods with respect to the overall trend gradient variation. Also, we observe a phenomenon from Fig. 3 that the gradient variation trend of the discriminator is similar to the generator. This reveals that the gradient updates of generator and discriminator affect each other. Consequently, only implementing the penalty on the discriminator enable very stable GANs training. These results suggest that the stability of GANs training can be significantly improved by our method.

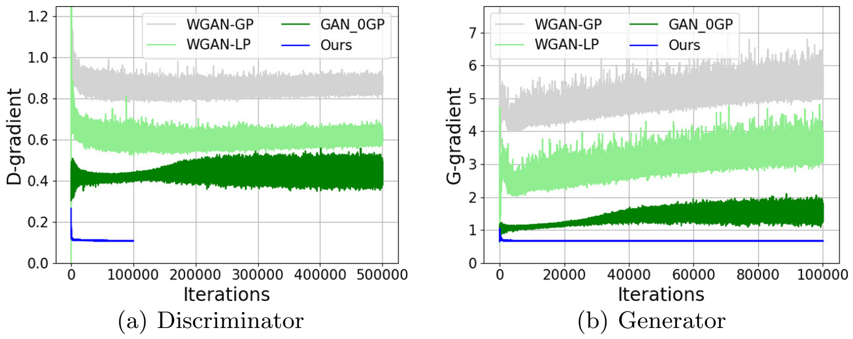

Fig. 3. Comparison of the overall trend of gradient variation among four GANs methods ((a) the gradient variation of discriminator; (b) the gradient variation of generator).

Table 3. Comparison of the generated samples quality (FID, the smaller is better).

Methods	CIFAR-10	SVHN	STL-10	CelebA	LSUN (bedroom)
LSGAN (2017)	22.20	3.84	20.17	5.10	**5.23**
SNGAN (2018)	20.70	4.53	18.11	5.56	12.05
WGAN-GP (2017)	21.89	4.09	18.19	5.01	14.61
WGAN-LP (2018)	21.01	3.62	17.40	5.12	15.21
GAN-0GP (2019)	18.91	6.10	14.49	4.53	7.14
Ours	**10.74**	**3.28**	**11.04**	**4.13**	6.59
Real datasets	0.46	0.24	0.84	0.34	0.55

4.3 Comparison of the Generated Samples Quality

The quality of the generated samples is one of the important indicators to reflect the performance of GANs model. The FID value is used to measure the quality, which the smaller FID value is better. In order to reduce the calculation error, the evaluation of FID is done on 50000 real samples and 50000 fake samples. To be more comprehensive, we compare our approach with five GANs models on five datasets and the results are summarized in Table 3. Clearly, our results (FID) are better than almost all other methods, only LSGAN (5.23) performs slightly better than our approach (5.59) on LSUN (bedroom). Especially, to our knowledge, the FIDs of our method on CIFAR10 and STL10 (10.74 and 11.04) are the state of the art in unsupervised image generation. This indicates that the quality of the generated samples in GANs can be significantly improved by our method. Note that the FIDs of real data are shown at the bottom of Table 3.

5 Conclusions

In this study, we integrated the zero-centered gradient penalty on the inner function of Sigmoid function of discriminator with spectral normalization (the modified SNGAN) to improve the performance of GANs, which is in contrast to the popular algorithms that the integrated method has better convergence, stability and the quality of the generated samples. Furthermore, our method can effectively alleviate the mode collapse behavior in GANs. In the experiments, we have illustrated evidence of improved training with several GANs algorithms on a variety of datasets and the resulting improvements in model performance. Our findings also show that WGAN-GP, WGAN-LP and GAN-0GP do not lead to networks convergence and training stability. In the future work, we would like to further dig into our ideas in more depth and come up with better performance methods.

A Training Details on Synthetic Datasets

The 8 Gaussians dataset is sampled from a mixture of 8 Gaussians of standard deviation 0.02, this means are equally spaced around a circle of radius 2. 25 Gaussians dataset, like the 8 Gaussians, is sample from a mixture of 25 Gaussians, which is arranged in a square. Two datasets consist of 100 k samples. The discriminator contains three SNLinear layers (bias: True, False and True) with 128 hidden units and LReLU (0.2) activation, and the generator contains three Linear layers (bias: False, False and True) with 256 hidden units, BN and ReLU activation.

As for the hyper parameters setting, both networks are optimized using OAdam with a learning rate of 0.0002 and $\beta_1 = 0.5$, $\beta_2 = 0.9$ (training the original GAN use Adam). The latent variable $\boldsymbol{z} \sim N(\mathbf{0}, \boldsymbol{I}_{128})$ and the penalty coefficient $\lambda = 10$ with Lipschitz constant $L = 0$. The batchsize is set to 100.

B Networks Architecture on Benchmark Datasets

See Tables 4 and 5.

Table 4. Discriminator ($3 \times 32 \times 32$).

SNconv 64 3×3 S = 1 P = 1 LReLU
SNconv 64 4×4 S = 2 P = 1 LReLU
SNconv 128 3×3 S = 1 P = 1 LReLU
SNconv 128 4×4 S = 2 P = 1 LReLU
SNconv 256 3×3 S = 1 P = 1 LReLU
SNconv 256 4×4 S = 2 P = 1 LReLU
SNconv 512 4×4 S = 1 P = 0 SN
Sigmoid()

Table 5. Generator ($3 \times 32 \times 32$).

dense $512 \times 4 \times 4$
deconv 512 4×4 S = 2 P = 1 BN ReLU
deconv 256 4×4 S = 2 P = 1 BN ReLU
deconv 128 4×4 S = 2 P = 1 BN ReLU
deconv 3 3×3 S = 1 P = 1 BN
Tanh()

References

1. Arjovsky, M., Bottou, L.: Towards principled methods for training generative adversarial networks. arxiv e-prints, art. arXiv preprint arXiv:1701.04862 (2017)
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017)
3. Bansal, N., Chen, X., Wang, Z.: Can we gain more from orthogonality regularizations in training deep CNNS? In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp. 4266–4276. Curran Associates Inc. (2018)
4. Barratt, S., Sharma, R.: A note on the inception score. arXiv preprint arXiv:1801.01973 (2018)
5. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096 (2018)
6. Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pp. 215–223 (2011)
7. Daskalakis, C., Ilyas, A., Syrgkanis, V., Zeng, H.: Training gans with optimism. arXiv preprint arXiv:1711.00141 (2017)
8. Fedus, W., Rosca, M., Lakshminarayanan, B., Dai, A.M., Mohamed, S., Goodfellow, I.: Many paths to equilibrium: Gans do not need to decrease a divergence at every step. arXiv preprint arXiv:1710.08446 (2017)
9. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256 (2010)
10. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
11. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: Advances in Neural Information Processing Systems, pp. 5767–5777 (2017)
12. van Handel, R.: Probability in high dimension. Technical report (2014)
13. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems, pp. 6626–6637 (2017)
14. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
15. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134 (2017)
16. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017)
17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
18. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
19. Kurach, K., Lucic, M., Zhai, X., Michalski, M., Gelly, S.: A large-scale study on regularization and normalization in gans. arXiv preprint arXiv:1807.04720 (2018)
20. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3730–3738 (2015)

21. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2794–2802 (2017)
22. Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for gans do actually converge? arXiv preprint arXiv:1801.04406 (2018)
23. Metz, L., Poole, B., Pfau, D., Sohl-Dickstein, J.: Unrolled generative adversarial networks. arXiv preprint arXiv:1611.02163 (2016)
24. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018)
25. Nagarajan, V., Kolter, J.Z.: Gradient descent gan optimization is locally stable. In: Advances in Neural Information Processing Systems, pp. 5585–5595 (2017)
26. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011)
27. Oberman, A.M., Calder, J.: Lipschitz regularized deep neural networks converge and generalize. arXiv preprint arXiv:1808.09540 (2018)
28. Paszke, A., et al.: Automatic differentiation in pytorch (2017)
29. Petzka, H., Fischer, A., Lukovnicov, D.: On the regularization of wasserstein gans. arXiv preprint arXiv:1709.08894 (2017)
30. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
31. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in Neural Information Processing Systems, pp. 2234–2242 (2016)
32. Saxe, A.M., McClelland, J.L., Ganguli, S.: Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. arXiv preprint arXiv:1312.6120 (2013)
33. Scardapane, S., Comminiello, D., Hussain, A., Uncini, A.: Group sparse regularization for deep neural networks. Neurocomputing **241**, 81–89 (2017)
34. Thanh-Tung, H., Tran, T., Venkatesh, S.: Improving generalization and stability of generative adversarial networks. arXiv preprint arXiv:1902.03984 (2019)
35. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7167–7176 (2017)
36. Virmaux, A., Scaman, K.: Lipschitz regularity of deep neural networks: analysis and efficient estimation. In: Advances in Neural Information Processing Systems, pp. 3835–3844 (2018)
37. Yeh, R., Chen, C., Lim, T.Y., Hasegawa-Johnson, M., Do, M.N.: Semantic image inpainting with perceptual and contextual losses. arXiv preprint arXiv:1607.07539 2(3) (2016)
38. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015)
39. Zhang, M., Lucas, J., Ba, J., Hinton, G.E.: Lookahead optimizer: k steps forward, 1 step back. In: Advances in Neural Information Processing Systems, pp. 9593–9604 (2019)