PAPER

Improving Noised Gradient Penalty with Synchronized Activation Function for Generative Adversarial Networks

Rui YANG[†a)], **Raphael SHU**[††], *Nonmembers*, *and* **Hideki NAKAYAMA**[†], *Member*

SUMMARY Generative Adversarial Networks (GANs) are one of the most successful learning principles of generative models and were wildly applied to many generation tasks. In the beginning, the gradient penalty (GP) was applied to enforce the discriminator in GANs to satisfy Lipschitz continuity in Wasserstein GAN. Although the vanilla version of the gradient penalty was further modified for different purposes, seeking a better equilibrium and higher generation quality in adversarial learning remains challenging. Recently, DRAGAN was proposed to achieve the local linearity in a surrounding data manifold by applying the noised gradient penalty to promote the local convexity in model optimization. However, we show that their approach will impose a burden on satisfying Lipschitz continuity for the discriminator. Such conflict between Lipschitz continuity and local linearity in DRAGAN will result in poor equilibrium, and thus the generation quality is far from ideal. To this end, we propose a novel approach to benefit both local linearity and Lipschitz continuity for reaching a better equilibrium without conflict. In detail, we apply our synchronized activation function in the discriminator to receive a particular form of noised gradient penalty for achieving local linearity without losing the property of Lipschitz continuity in the discriminator. Experimental results show that our method can reach the superior quality of images and outperforms WGAN-GP, DiracGAN, and DRAGAN in terms of Inception Score and Fréchet Inception Distance on real-world datasets.

key words: *GAN, gradient penalty, local linearity*

1. Introduction

Generative Adversarial Networks (GANs) [1] gain a significant role in generative models. The generator plays a minmax game against the discriminator. Typically, such adversarial learning policy aims to achieve a dynamic equilibrium between generator and discriminator when networks obtained the convergence, to generate high-quality images imitatively. Hereon, the convergence of generator and discriminator is dynamic while their common equilibrium point is fixed as a certain point [1].

Lipschitz-1 continuity was adopted as a condition for training WassersteinGAN [2]. Wasserstein distance (W-distance) can support an ideal dynamic equilibrium in GANs' training owing to the fact that the probability measure of W-distance is strictly weaker than KL-divergence [3] and exhibited excellent performance. One famous approach of Lipschitz constraint in GANs is adding the gradient penalty (GP) in the discriminator loss, as in WGAN-GP [4]. The intuition of gradient penalty is that if the gradients derived from the discriminator are not greater than 1 almost everywhere, we can assert that the discriminator satisfies the Lipschitz-1 continuity.

Besides, different variants of gradient penalty in GANs can profoundly impact the properties of GANs' training, such as convergence [5], gradient exploding [4], local linearity [6], or generation diversity [7], etc. Therefore, different variants of GP will result in various equilibrium situations and thus affect the generation quality.

In DiracGAN [5], the authors proved that applying the zero-centering gradient penalty onto real samples can improve the convergence. The authors found that the one-centering gradient penalty in WGAN-GP makes the learning direction oscillate around the equilibrium center.

DiracGAN improved the convergence problem, while it cannot prevent gradient explosion [7]. DRAGAN [8] was one meaningful attempt to solve both problems. The authors noticed that the gradients of random samples around image inputs are very sharp in the discriminator when the model collapse occurred. As a result, they proposed the noised version of gradient penalty by penalizing real images with a small perturbation to enlarge the penalized space and encourage the discriminator to escape from the local minima. Consequentially, as mentioned in [6], such operations can encourage the local linearity and benefit the training process due to the convexity of linear functions in optimization.

This study found that the gradient penalty in DRAGAN could be a burden in chasing Lipschitz-1 continuity when strengthening its local linearity. In detail, when applying the gradient penalty onto the perturbed image, the positiveness of feature maps before activation layers in the discriminator is less likely the same as the positiveness of feature maps of the original image, while only weight parameters corresponding to positive features will be upgraded based on ReLU function. Whereas, ensuring sufficient penalty onto real images is the key point of Lipschitz continuity, as proved in DiracGAN [5]. In other words, applying noised gradient penalty will release the penalty of original images, which hampers the Lipschitz continuity and thus obstructs the convergence.

To this end, we design a masked synchronized activation function that can activate the feature maps of perturbed images and original images synchronously. To be specific, our synchronized activation function can derive an improved version of the noised gradient penalty by avoiding the conflict between penalizing perturbed noised images or origi-

Manuscript received January 29, 2022.
Manuscript revised May 1, 2022.
Manuscript publicized May 27, 2022.

[†]The authors are with the University of Tokyo, Tokyo, 113–8654 Japan.
[††]The author is with the Amazon AI, Seattle, WA, USA.
a) E-mail: yangrui@nlab.ci.i.u-tokyo.ac.jp

DOI: 10.1587/transinf.2022EDP7019

Table 1 Different types of gradient penalties.

Method	Prevent Gradient Exploding	Keep Lipschitz Continuity	Guarantee Local Linearity
WGANGP	✓	**Lipschitz-1**	✗
DiracGAN	✗	**Lipschitz-0**	✗
DRAGAN	✓	**obstructed**	✓
Ours	✓	**Lipschitz-k**	✓

nal images in DRAGAN, which refers to local linearity and Lipschitz continuity, respectively. Hence, our method can acquire both advantages in local linearity and Lipschitz continuity without the conflict in DRAGAN. We conclude features of past works and our method in Table 1. As shown in Table 1, applying the gradient penalty onto larger input space, such as the linear interpolation in WGANGP or adding a noise perturbation in DRAGAN and ours, can prevent gradient from exploding. DRAGAN acquires local linearity while loses the feature in Lipschitz continuity. Furthermore, due to the noised gradient penalty, our method can still prevent the gradient from exploding as in DRAGAN.

In all, we highlight our contributions as follows:

- We show the noised gradient penalty in DRAGAN obstructs the Lipschitz continuity of the discriminator;
- We propose a synchronized activation function for improving the noised gradient penalty to enable the local linearity in the discriminator without obstructing its Lipschitz continuity;
- Experiments show that our method achieves higher generation quality and faster convergence speed against WGAN, WGAN-GP, DiracGAN, and DRAGAN in image generation tasks.

2. Background

2.1 WassersteinGAN

The original intention of applying gradient penalty on the discriminator in WassersteinGAN is to force the discriminator to satisfy Lipschitz-1 continuity based on Kantorovich-Rubinstein duality [9]:

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_\theta}[f(x)]. \quad (1)$$

Here, if the function $f(\cdot)$ (i.e., the discriminator) satisfies that the Lipschitz constant is not greater than one, then W-distance can be correctly applied in the adversarial training. Although W-distance has better properties than JS-divergence or f-divergences in GANs, correctly implementing Lipschitz-1 constraint in the adversarial training is also very important. In practice, one option is to apply the one-centering gradient penalty as in WGAN-GP:

$$\mathop{\mathbb{E}}_{\hat{x}\sim\mathbb{P}_{\hat{x}}} \left[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2\right], \quad (2)$$

where $D(\cdot)$ is the discriminator, $\hat{x}$ is the linear interpolation between real image and fake image (later the same).

In some recent works, other types of gradient penalty were proposed to achieve better performance when training GANs. For instance, infinite norm was used in [10] via a viewpoint from SVM. Besides, R1 penalty also obtained great success in StyleGAN [11] and StyleGAN2 [12], [13].

2.2 DiracGAN

DiracGAN analyzed the eigenvalues of the Jacobian of the associated gradient vector field [5] and proposed the zero-centering gradient penalty which can ensure the convergence ability significantly. Here is the gradient penalty term in the discriminator loss of DiracGAN:

$$\mathbb{E}_{p_{D(x)}} \left[\|\nabla D_\psi(x)\|^2\right], \quad (3)$$

where $p_{D(x)}$ is the data distribution and ψ is the discriminator parameter. Moreover, the authors also show that DiracGAN performs well in the cases of only penalizing fake images or linearly combining gradient penalties from both real images and fake images.

Zero-centering gradient penalty solves the local convergence problem based on their proof of training dynamics. However, only penalizing such samples can hardly prevent gradient from exploding as introduced in [7].

2.3 DRAGAN

DRAGAN found that when mode collapse occurred, the discriminator often has sharp gradients around the real samples. As it is hard to find out the exact reason causing mode collapse, punishing the gradients of surrounding points close to the real samples is one feasible method. Here is the gradient penalty term in DRAGAN:

$$\mathbb{E}_{x\sim P_{\text{real}},\delta\sim U(-\sigma,\sigma)} \left[\|\nabla_{\mathbf{x}} D_\theta(x+\delta)\|^2 - k\right], \quad (4)$$

where δ is sampled from a Uniform distribution with the range of a single standard deviation σ of current input batch x. k is the Lipschitz-k constant. Fedus et. al. [6] further discussed this work and reinterpreted the internal mechanism. In practice, penalizing points in local scope help the discriminator to be close to linear around the local data manifold. The convexity of linear functions helps optimize during the training process to overcome the gradient exploding problem mentioned earlier without harming the convergence property in an ideal case.

3. Deriving Noised GP from Synchronized Activation Function

3.1 Motivation: Solving the Conflict in DRAGAN

By penalizing gradients of the discriminator from images with a small perturbation, such a noised gradient penalty in

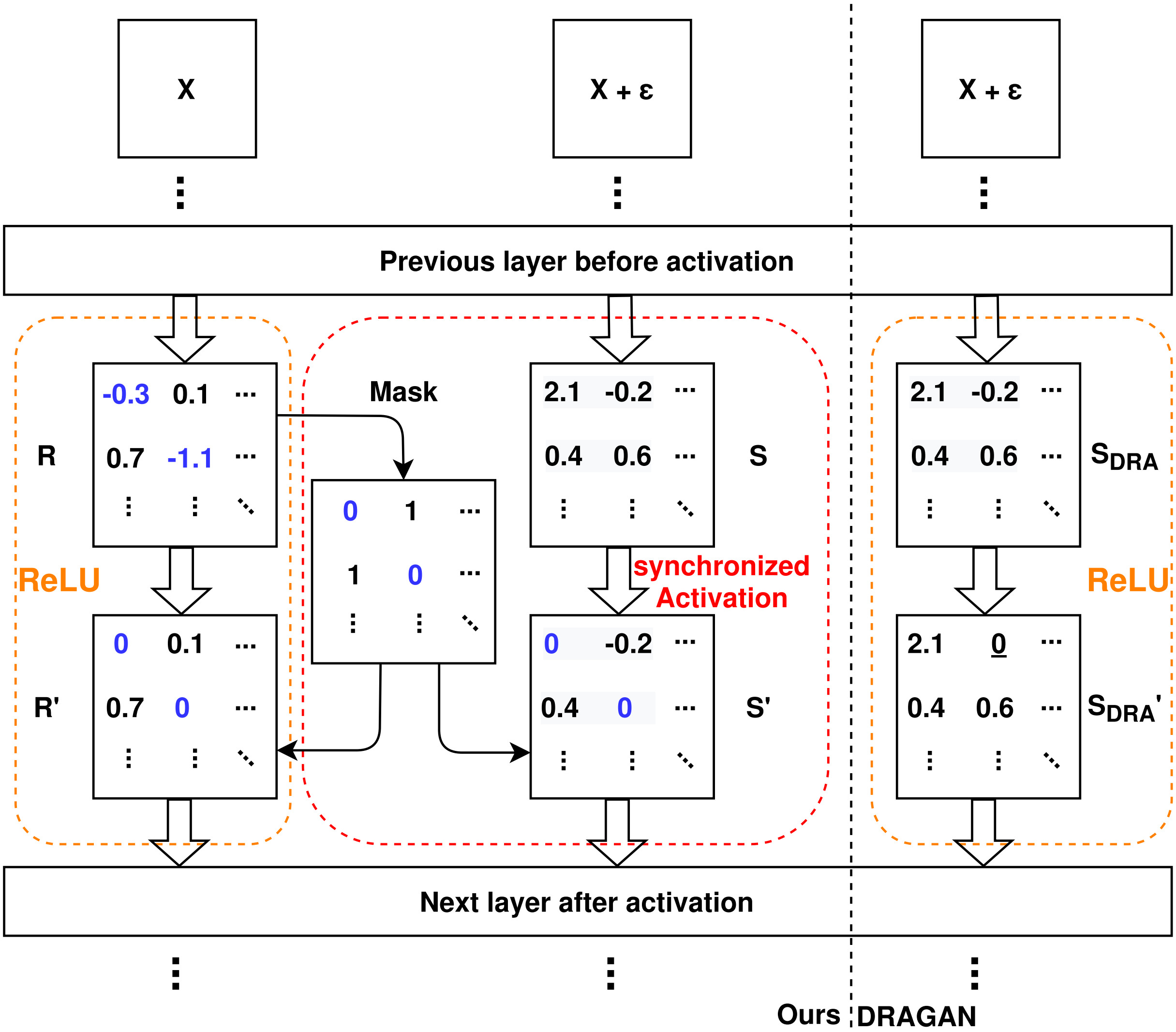


Fig. 1 One example to illustrate our synchronized activation function. ReLU refers to Rectified Linear Unit. X is the input tensor of the discriminator, and $(X + \epsilon)$ is its noised version for deriving the noised gradient penalty. R and S are feature maps before the activation function for inputs X or $(X + \epsilon)$ respectively. R' is the feature map after applying the activation function on R, and S' is the activated feature map of S. As shown in S', each element in S will be activated based on the mask generated by R, not on S. In this example, elements in S' marked as blue were rectified based on the mask despite its positiveness in S as in red square. And R' is activated via a normal ReLU function as in orange square. After all, the discriminator output of $X + \epsilon$ will derive our noised GP as in Algorithm 1.

DRAGAN can encourage the discriminator to become near-convex in the local scope. Meanwhile, the feature maps of original image inputs (for adversarial loss) in the discriminator are different from their noised version (for noised gradient penalty). Consequently, weight parameters corresponding to feature maps for noised image inputs are less likely to be activated synchronously than original image inputs based on ReLU-like activation functions, either in the back-propagation step. Whereas, ensuring a sufficient penalty for original image inputs was proved as the convergence condition of GANs as shown in [5]. Therefore, the noised gradient penalty will improve the model robustness against gradient exploding due to its better convexity yet obstructing the convergence.

3.2 Proposal: Improving the Noised GP via Synchronized Activation Function

In this section, we introduce a synchronized activation function. It is a simple modification to ReLU or any ReLU-like activation functions, such as Rectified Linear Unit (ReLU) [14], Parametric Rectified Linear Unit (P-ReLU) [15], or Leaky Rectified Linear Unit (L-ReLU) [16], etc.. The core idea is to activate feature maps from noised inputs only based on masks from original inputs.

As shown in Fig. 1, we use ReLU function as an example to interpret our synchronized activation function. Before passing feature maps through the activation layer, we first forward propagate real image tensors and surround-

Algorithm 1 WGAN with noised GP based on synchronized activation functions.

```
Require: learning rates $(\alpha_g, \alpha_d)$, batch size $B$, discriminator training iterations $m$ per generator step, training data distribution $P_{data}$, distribution $q$ of the noise perturbation, GP strength $\lambda$.
  Initialize $G$ parameter $\theta$ and $D$ parameter $\phi$ ($D_{sync}$ owns the same parameter as $D$);
  while $\theta$ has not converged do
      for $j$ = 1, ..., m do
          Sample real samples $\{x_i\}_{i=1}^{B} \sim P_{data}$ and i.i.d. noises $\{z_i\}_{i=1}^{B} \sim N(0, 1)$;
          Generate fake samples $\{y_i\}_{i=1}^{B} \leftarrow G_\theta(z_i)$;
          $\mathrm{GP}_{sync} \leftarrow (\|\nabla D_{sync}(\mathbf{x} + \delta)\| - k)^2, \delta \sim q, k \in [0, 1]$;   ▹ via Eq. (9)
          $\mathrm{Loss_D}^{(j)} \leftarrow D_\phi(G_\theta(\mathbf{z})) - D_\phi(\mathbf{x}) + \lambda \cdot \mathrm{GP_{sync}}$;   ▹ via Eq. (8)
      end for
      $\phi \leftarrow Adam(\nabla_\phi \sum_{j=1}^{m} \mathrm{Loss_D^{(j)}}, \phi, \alpha_d)$;
      Sample i.i.d. noises $\{z_i\}_{i=1}^{B} \sim N(0, 1)$;
      Generate fake samples $\{y_i\}_{i=1}^{B} \leftarrow G_\theta(z_i)$;
      $\mathrm{Loss_G} \leftarrow -D_\phi(G_\theta(\mathbf{z}))$;   ▹ via Eq. (7)
      $\theta \leftarrow Adam(\nabla_\theta \mathrm{Loss_G}, \theta, \alpha_g)$;
  end while
```

ing image tensors (real images with perturbations) through the same convolution and normalization layers, respectively. Before the ReLU activation layer, we define R and S as non-activated feature maps from real image input tensors and surrounding image input tensors respectively, and create rectified tensors R' and S' which are multiplied by 0 temporarily. Then, R and S will be activated separately: R is activated by a normal ReLU activation function, and our synchronized activation function will activate S as shown below:

$$R'_i = \mathbb{1}(R_i > 0)R_i, (ReLU) \tag{5}$$

$$S'_i = \mathbb{1}(R_i > 0)S_i.(Ours) \tag{6}$$

Each element S'_i in S' should be activated or not is dependent on the mask from R yet S. Here, subscript i in Eq. (5), 6 refers to the ith element in its specific feature map. Meanwhile, R will be activated by the normal ReLU and get R'. Next, we will pass both activated feature maps R' and S' through the next layer and repeat all operations until the final output layer. Outputs from $(X+\epsilon)$ will be used for applying gradient penalty, and R' for reducing the discriminator loss.

After receiving noised gradient penalty based on our synchronized activation function, we still need to compute other parts of the discriminator loss and the generator loss. Details will be introduced in Sect. 4 and Algorithm 1.

4. Loss functions

4.1 Generator Loss

WassersteinGAN is trained based on Eq. (1), and different variants defined different types of motivation for training the discriminator. This work focus on improving the GP in discriminator loss which is not related to the generator directly. Therefore, the generator loss in our experiments of $G(\cdot)$ are all the same:

$$\mathrm{Loss_G} = -\mathop{\mathbb{E}}_{z\sim p(z)}[D(G(z))]\,. \tag{7}$$

4.2 Discriminator Loss

Common discriminator variants of WGAN-GP loss functions majorly have the adversarial loss term and the gradient penalty term and can be written as follow:

$$\begin{aligned}\mathrm{Loss_D} = -&\mathop{\mathbb{E}}_{x\sim q_{\mathrm{data}}(x)}[D(x)] + \\ &\mathop{\mathbb{E}}_{z\sim p(z)}[D(G(z))] + \lambda\cdot\mathrm{GP_{sync}},\end{aligned} \tag{8}$$

where $\mathrm{Loss_D}$ is the discriminator loss, $D(G(z))$ are outputs of the discriminator of fake images generated from the generator $G(\cdot)$, and vice versa for $D(x)$. $\mathrm{GP_{sync}}$ refers to the gradient penalty based on our synchronized activation function. λ is the hyper-parameter to control the strength of the gradient penalty. In the synchronized discriminator D_{sync} based on our proposed activation function, the gradient penalty term is:

$$\mathrm{GP_{sync}} = \mathop{\mathbb{E}}_{x\sim p_{\mathrm{data}}(x),\delta\sim q}\left[(\|\nabla D_{sync}(x+\delta)\| - k)^2\right], \tag{9}$$

where x were sampled from the real image distribution. k is the GP center varying from zero to one and we tested zero-centering and one-centering GP as ablations in our experiments. The perturbation σ has two definitions in practice: 1) In original implementation by the authors in DRAGAN, δ refers to the standard deviation of the real image distribution in one batch as in Eq. (4); 2) Some common implementations of DRAGAN define the edge of the local range as random fake images and also obtain idea results. Therefore, we set both implementations as ablation studies in our experiments to compare their performances for noised GP. Nevertheless, other factors also matter the model performance and it is hard to assert which one is better. Besides, $\mathrm{D_{sync}}(\cdot)$ performs linearity in the gradient penalty term and $\mathrm{D}(\cdot)$ performs non-linearity in adversarial losses.

We also use the hinge loss to stabilize and improve the training in the discriminator [17]:

$$\begin{aligned}\mathrm{HingeLoss_D} = &\mathop{\mathbb{E}}_{x\sim q_{data}(x)}[\min(0, -1 + D(x))] + \\ &\mathop{\mathbb{E}}_{z\sim p(x)}[\min(0, -1 - D(G(z))]\,,\end{aligned} \tag{10}$$

where $min(0, x)$ returns the minimum value between the input x and the zero.

5. Experiments

As illustrated above, our synchronized activation function can derive the improved noised gradient penalty from acquiring both advantages from local linearity and Lipschitz continuity. Nevertheless, it is hard to evaluate the goodness of the equilibrium directly. Besides, the target of the image generation task is to train a model which can generate high-quality images successfully, and better equilibrium allows GANs to maintain more information and details to reach this target. Therefore, we design several experiments to evaluate and compare our method with past methods via testing the generation quality and their convergence speed to show the goodness of each method.

Firstly, we train the image generation task based on past gradient penalty and ours and record raw evaluation metrics. Then, we compare the best FID and IS during the training process for quantitative analysis. Next, we draw curve charts for evaluation metrics by comparing the convergence speed. Besides, we also show statistics of diversity analysis among all categories of CIFAR-10 to discuss the performance of methods. Finally, we also show visual samples for qualitative analysis.

5.1 Settings

5.1.1 Dataset

We tested and compared our method with past works based on CelebA (203k training images, resize to 64x64 resolution), CIFAR-10 (50k training images, 10 classes, 32x32 resolution), and Tiny-ImageNet (100k training images, 200 classes, 64x64 resolution).

5.1.2 Comparison

We set three past works as baselines to compare with our work. Firstly, WGAN-GP started the boom to use the GP to help train the GANs. Next, DiracGAN solved the oscillating problem in convergence in WGAN-GP based on zero-centering GP. Afterwards, DRAGAN noticed the gradient exploding problem in DiracGAN and further improved the local linearity based on their noised GP. Finally, we recorded two kinds of evaluation metrics to show how well can different kinds of GP lead to the equilibrium. All settings used Eq. (8) and λ was always set as ten based on past literatures.

As for experiments in WGAN-GP [4], ours, and DRAGAN [8], these methods penalize extra points instead of real or fake samples. Thus, they have the same training time. While DiracGAN [5] is faster due to the direct penalization on real or fake samples.

5.1.3 Ablation

As for the noised gradient penalty based on our synchronized activation function, we use four different settings which are combinations between 1) define local range with a single standard deviation or treat random fake images as the edge of local range; 2) using one-centering GP to gamble on acquiring more information during the oscillation or using zero-centering GP to maintain a better convergence.

In Table 2, we named our method with settings of 'random fake images as the edge of local range' and 'one-centering GP' as 'SYNC-R.F.-1'. And vice versa for cases of 'singe standard deviation as the edge of local range' and 'zero-centering', as in '-S.S.' and '-0'.

As the loss function in DRAGAN [8] can be treated as the non-synchronized-activation version of our one-centering 'Sync-S.S.' experiment, we set them in the same group for comparison. Identically, WGAN-GP [4] is similar to the one-centering 'Sync-L.I.' case, and DiracGAN [5] is similar to the zero-centering 'Sync-R.F.' case.

We did not compare the setting of 'L.I.' with the zero-centering GP due to WGAN-GP [4] did not mention or validate such a setting in the original paper. Besides, although other possible combinations (zero-centering 'S.S.' and one-centering 'R.F.') cannot gain their advantages in past works, we also test their performances based on our synchronized activation function for comparison.

5.1.4 Network Architecture

As for the experimental comparison, we used the famous BigGAN [18] architecture as the base model and only changed the versions of gradient penalty in the discriminator loss function. Besides, we used a mean-pooling layer to instead of the max-pooling layer in BigGAN to avoid non-linearity except for ReLU function. In this case, all of the non-linearity are bring from the activation function which controlled the ablation to verify our synchronized activation function.

We have several reasons to select BigGAN model as the network architecture for all experiments. First, BigGAN model is more competitive against past models. Although past models (DCGAN [19], SAGAN [20], SNGAN [17], etc.) obtained great success, BigGAN achieved state-of-the-art performance. Second, BigGAN model only has one single max-pooling layer. Therefore, changing such a max-pooling layer to a mean-pooling layer will not bring a remarkable difference in performance. Third, BigGAN model used spectral normalization [17] to keep its training stability, and there is no other terms in the loss function. In this case, we can exclude interference factors via other types of penalties, yet some other powerful network architectures (e.g., StyleGAN [11] and StyleGAN2 [12], [13]) cannot.

5.1.5 Hyper-Parameters

We trained the generator and the discriminator once-by-once in turns. All models were trained by Adam optimizer [21] with the generator learning rate set as 2e-4 and the discriminator learning rate set as 2e-4, respectively. The Beta1 was 0.9 and Beta2 was 0.999 in Adam optimizer. The exponential moving average (EMA) were started after first 1k training iterations. The batch size was 64 for Tiny-ImageNet experiments and was 256 for CIFAR-10 and CelebA experiments. For CIFAR-10 experiments, we trained models with enough iterations and recorded the metrics curves for comparison while applying the early-stop mechanism for Tiny-ImageNet and CelebA experiments. The dimension number of the input noise is 128 for all experiments. We used Py-

Table 2 Best FID (lower is better) and their corresponding IS (higher is better) during training processes. As IS is meaningless for human face dataset, so we only list FID for CelebA experiments. Our methods are marked with †. For penalized objects, '-R.F.' stands for 'random fake samples as the scope of local data manifold'; '-S.S.' for 'a single standard deviation of real images as the scope of local data manifold'; '-L.I.' for 'treat the linearly interpolated images as the local center and use a single standard deviation of them as the local scope'. We mark the best scores bold for each similar methods and square the best scores among all the methods.

Method	Penalized Objects	GP Center	CIFAR-10		Tiny-ImageNet-200		CelebA
			IS	FID	IS	FID	FID
WGAN-GP	L.I.	1	7.836±0.095	8.179	10.315±0.192	29.925	8.711
†Sync.	L.I.	1	$\boxed{\mathbf{8.301\pm0.112}}$	7.160	$\boxed{\mathbf{12.085\pm0.218}}$	28.340	$\boxed{\mathbf{6.497}}$
†Sync.	R.F.	1	8.250±0.056	$\boxed{\mathbf{6.698}}$	12.058±0.149	**27.887**	6.557
DRAGAN	S.S.	1	7.579±0.109	9.384	10.591±0.113	29.545	7.188
†Sync.	S.S.	1	**8.244±0.094**	**6.982**	**12.071±0.231**	**29.158**	**6.914**
DiracGAN	R.F.	0	7.677±0.094	9.185	9.968±0.159	31.098	7.511
†Sync.	R.F.	0	8.238±0.093	7.398	11.214±0.229	30.821	**6.512**
†Sync	S.S.	0	**8.271±0.084**	**6.795**	**11.935±0.262**	$\boxed{\mathbf{27.839}}$	7.350

torch [22] version 1.1.0 to organize all experiments.

5.1.6 Evaluation Metrics

For evaluations, we used the famous Fréchet Inception Distance (FID) [23] to quantify the model performance (lower is better). We also compared Inception Score (IS, higher is better). We sampled the best-performed model based on FID and also recorded its IS due to FID is relatively better than IS [24], [25]. For testing the best model performance, we calculated and recorded the IS and FID score every 1k training steps. For every experiment, we sampled 50k random samples to calculate the FID and IS.

5.2 Quantitative Analysis

As shown in Table 2, we compared two kinds of evaluation metrics for three real-world datasets. We trained all experiments with the same architecture except for the type of GP in the discriminator loss. From Table 2: 1)First of all, our method based on the synchronized activation function can generate superior quality compared with other types of GP; 2)In the case of ablation studies, among four different ablations, the setting of '-R.F.-0' performed inferior performance than other ablations in terms of FID and IS. In a complex dataset, the '-R.F.-0' setting cannot defeat past works w.r.t. the FID score while the IS is superior; 3) In order to further analyze whether this setting sacrificed the generation quality to balance other properties, we will compare the convergence speed and the generation diversity latter; 4) Besides, the setting of '-R.F.-1' and the setting of '-S.S.-0' performed equally excellent in terms of FID. We suppose that '-R.F.' can exert a risky training strategy in one-centering GP while '-S.S.' can cooperate with a steady training strategy in zero-centering GP as introduced in Sect. 4.2.

Table 3 Statistics of LPIPS (higher is more diverse) among all categories of CIFAR-10 experiments.

Method	Mean	Std	Min	Max
WGAN-GP	0.2150	0.0757	0.1142	0.3701
DRAGAN	**0.2344**	0.0712	0.1089	0.3760
DiracGAN	0.2316	0.0861	0.1111	0.4273
Sync-R.F.-1	0.2136	0.0692	0.0993	0.3513
Sync-S.S.-1	0.2106	0.0729	0.1173	0.3514
Sync-R.F.-0	**0.2346**	0.0746	0.1106	0.3719
Sync-S.S.-0	0.2151	0.0865	0.0950	0.4227

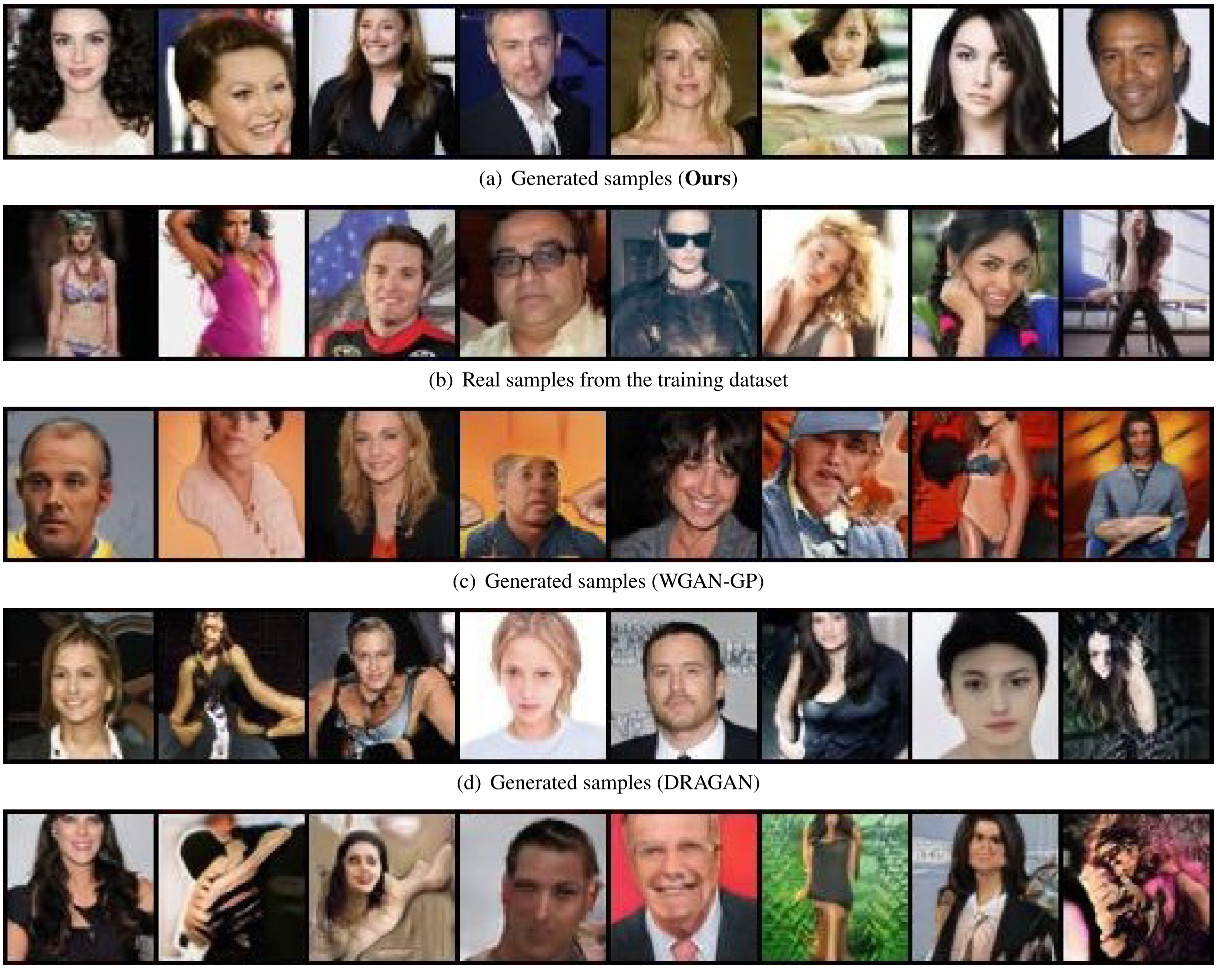
5.3 Diversity Analysis

In Table 3, we show LPIPS statistics for CIFAR-10 experiments to compare the generation diversity. Generally speaking, our '-R.F.-0' setting can achieve similar diversity as DRAGAN. In contrast, all methods obtained negligible differences, which means that our methods can still keep a standard performance in the generation diversity.

5.4 Qualitative Analysis

In Fig. 2, we show randomly generated samples and compare with real samples. Generally, our method is less likely to generate weird samples, and the backgrounds in our samples are more clear.

From Fig. 3, we can find that our methods demonstrated a higher convergence speed to reach the peak performance during the training process. Many reasons can lead to faster convergence in GANs training. We summarize two possible reasons based on past literature. First, strictly keeping Lipschitz continuity in the discriminator can lead to the stable training process [4], and therefore lead to faster



(a) Generated samples (**Ours**)

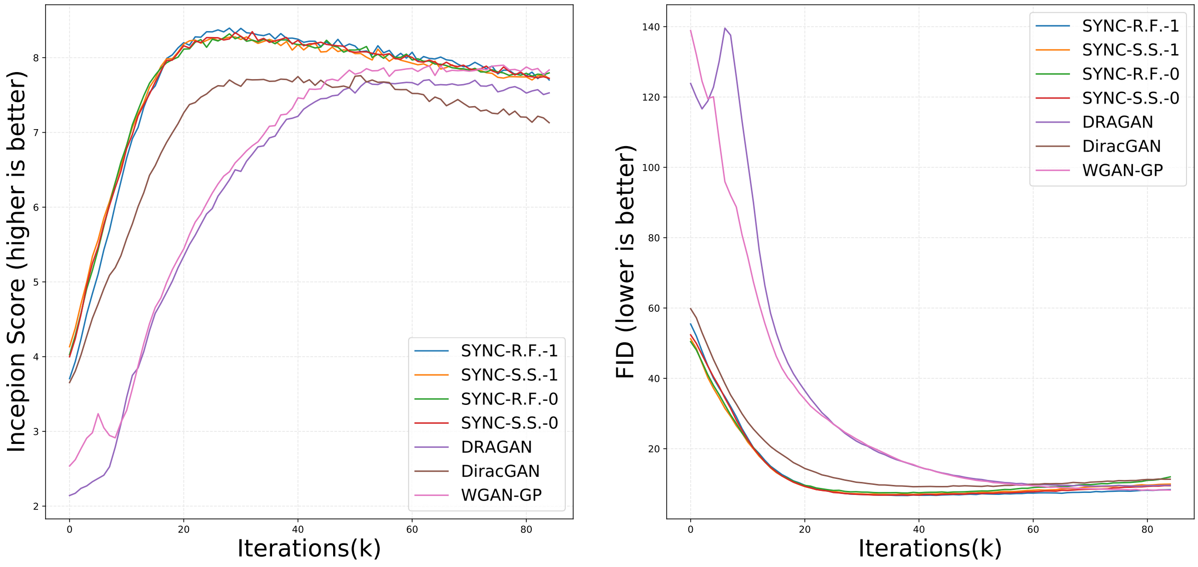
(b) Real samples from the training dataset

(c) Generated samples (WGAN-GP)

(d) Generated samples (DRAGAN)

(e) Generated samples (DiracGAN)

Fig. 2 CelebA generation samples and real samples.



(a) Inception Score Curve Map

(b) FID Curve Map

Fig. 3 IS and FID curve maps by training iterations. Our methods demonstrated better generation qualities and faster convergence speed. ‘k’ refers to one thousand training iterations here. One reminder is that in Table 2, we sampled from the trained iteration, which achieved the best FID and then recorded the corresponding IS in that iteration. Some models can achieve a little bit better IS before or after the best FID iteration.

convergence. DRAGAN and WGAN-GP penalized limited samples (real, fake, or linear combinations), and DRAGAN penalized random samples near the real or fake samples. In contrast, ours penalized the whole local region, larger than past works. Second, local linearity can benefit the training process due to the convexity of linear functions in optimization [6]. Our method specially designed the synchronized activation function to keep better linearity in the local range.

6. Discussion

6.1 Penalized Points and GP Center

Some past works discussed the penalized points between two cases: the linear interpolation between real and fake samples or only penalize onto real samples with noise. In our work, the linear range (random fake samples as the edge of local data manifold) or the spherical range (single standard deviation of real samples as the edge of local data manifold) demonstrated their properties to a certain extent. At the same time, the major contributor is the noised GP based on our synchronized activation function.

Discussion between zero or one-centering GP also makes sense. Dislike evaluation metrics in other computer vision tasks (classification, segmentation, etc), computing FID and IS are timely expensive. If the project time is not limited, calculate FID scores for every iteration to select the best result. Therefore, sampling evaluation metrics scores after certain iterations are more feasible under a limited training time for the whole generation task, which means the best model might be neglected. The zero-centering case can lead the optimizing direction to one point that reduces the oscillation in adversarial learning and increases the lower bound of model performance. The one-centering case will lead to a circling case in optimization which brings chances to meet better equilibrium. In other words, less oscillation increases the sampling lower bound of the generator performance, while higher fluctuation may bring some extraordinary generators occasionally. Thus, which center is better will depend on the training difficulty of dataset size and the scale of network architecture. In our experiments, the setting of using random fake images with one-centering GP or using a single standard deviation with zero-centering GP is recommended for higher generation quality. As for higher generation divergence, the setting of using random fake images with zero-centering GP is appreciated.

6.2 Recent Works

GP-based methods are well developed in some recent works. Basically, recent GPs tend to solve other problems yet local linearity. SVM-GANs [10], [26] proposed to penalize the infinite norm in GP to achieve a maximum-margin classifier in the discriminator. WGAN-div and one of its special case [7], [27] defined GP as a high order exponential loss function rather than mean-square loss in past works. Generalization and stability can be improved significantly based on their GP. Gradient normalization [28] changed GP to the normalization and achieved larger capability in the discriminator. Our method follows the traditional Frobenius norm and mean-square-error as past works [4] and changes penalized objects merely. Thus, our method is compatible with mentioned recent works because these methods have no relation to the activation function.

6.3 Limitation of Our Method

Our method realized a novel synchronized activation function to achieve local linearity. The first limitation is that there must be no other non-linearity in the discriminator except for our activation function. Thus, our method is suitable if changing other non-linearity in some models will not degenerate the model performance (such as changing the max-pooling layer to the mean-pooling layer in BigGAN). Besides, our method could be applied to any GP-based methods if they are compatible with our work.

7. Conclusion

This work provided insights into the conflict problem between local linearity and Lipschitz continuity in DRAGAN. Then, we proposed an improved version of noised GP based on our synchronized activation function to solve the conflict problem in the discriminator and improve the generation quality. Table 2 showed that our proposal opens a novel approach for improving the generation quality as well as the convergence speed based on different real-world datasets. Besides, our method can also be applied in newly proposed GANs to improve their discriminator performance and generation quality.

Acknowledgments

This work was supported by Institute of AI and Beyond of the University of Tokyo, and JSPS KAKENHI Grant Number JP19H04166.

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, pp.2672–2680, 2014.

[2] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” Int. Conf. Mach. Learn., pp.214–223, 2017.

[3] S. Liu, O. Bousquet, and K. Chaudhuri, “Approximation and convergence properties of generative adversarial learning,” Advances in Neural Information Processing Systems, vol.30, pp.5545–5553, 2017.

[4] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A.C. Courville, “Improved training of wasserstein GANs,” Advances in neural information processing systems, pp.5767–5777, 2017.

[5] L. Mescheder, A. Geiger, and S. Nowozin, “Which training methods for GANs do actually converge?,” Int. Conf. Mach. Learn., pp.3481–3490, 2018.

[6] W. Fedus, M. Rosca, B. Lakshminarayanan, A.M. Dai, S. Mohamed, and I. Goodfellow, “Many paths to equilibrium: GANs do not need

to decrease a divergence at every step," Int. Conf. Learning Representations, 2018.

[7] H. Thanh-Tung, T. Tran, and S. Venkatesh, "Improving generalization and stability of generative adversarial networks," ICLR 2019: Proc. 7th Int. Conf. Learning Representations, ICLR, 2019.

[8] N. Kodali, J. Abernethy, J. Hays, and Z. Kira, "On convergence and stability of GANs," arXiv preprint arXiv:1705.07215, 2017.

[9] C. Villani, Optimal transport: old and new, Springer Science & Business Media, 2008.

[10] A. Jolicoeur-Martineau and I. Mitliagkas, "Gradient penalty from a maximum margin perspective," arXiv preprint arXiv:1910.06922, 2020.

[11] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp.4401–4410, 2019.

[12] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp.8110–8119, 2020.

[13] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," arXiv preprint arXiv:2006.06676, 2020.

[14] V. Nair and G.E. Hinton, "Rectified linear units improve restricted boltzmann machines," Proc. 27th Int. Conf. Mach. Learn. (ICML-10), pp.807–814, 2010.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," Proc. IEEE Int. Conf. Comput. Vis., pp.1026–1034, 2015.

[16] A.L. Maas, A.Y. Hannun, and A.Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in ICML Workshop on Deep Learning for Audio, Speech and Language Processing, Citeseer, 2013.

[17] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," Int. Conf. Learning Representations, 2018.

[18] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," Int. Conf. Learning Representations, 2019.

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

[20] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," Int. Conf. Mach. Learn., pp.7354–7363, PMLR, 2019.

[21] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[22] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[23] 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, pp.6626–6637, 2017.

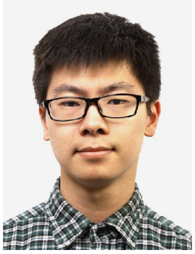
[24] A. Borji, "Pros and cons of GAN evaluation measures," Computer Vision and Image Understanding, vol.179, pp.41–65, Feb. 2019.

[25] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," Advances in neural information processing systems, pp.2234–2242, 2016.

[26] A. Jolicoeur-Martineau, "The relativistic discriminator: a key element missing from standard GAN," Int. Conf. Learning Representations, 2019.

[27] J. Wu, Z. Huang, J. Thoma, D. Acharya, and L. Van Gool, "Wasserstein divergence for GANs," Proc. European Conf. Comput. Vis. (ECCV), Sept. 2018.

[28] Y.L. Wu, H.H. Shuai, Z.R. Tam, and H.Y. Chiu, "Gradient normalization for generative adversarial networks," Proc. IEEE/CVF Int. Conf. Comput. Vis., pp.6373–6382, 2021.



Rui Yang obtained M.S. degree from The University of Tokyo in 2018. His research interests include generative adversarial networks, gradient penalty, image generation. He is currently a doctoral student in Nakayama Laboratory, the Graduate School of Information Science and Technology, The University of Tokyo.

Raphael Shu obtained Ph.D. degree from The University of Tokyo in 2020. He received the Dean's award for the scientific contribution during PhD. He is currently working as an applied scientist in Amazon AI. His research interests include generative modelling for sequential data, large-scale distributed deep learning.

Hideki Nakayama received the M.S. and Ph.D. degrees in information science from the University of Tokyo in 2008 and 2011, respectively. He was a Research Fellow of the Japan Society for the Promotion of Science (DC1) from 2008 to 2011. He is currently a full-time Associate Professor at the Graduate School of Information Science and Technology, The University of Tokyo. His research interests include generic object and image recognition, multimedia analysis, NLP and deep learning.