CAGAN: Consistent Adversarial Training Enhanced GANs

Yao Ni, Dandan Song, Xi Zhang, Hao Wu and **Lejian Liao**
Lab of High Volume language Information Processing & Cloud Computing
Beijing Lab of Intelligent Information Technology
School of Computer Science & Technology, Beijing Institute of Technology
{niyao, sdd, xi_zhang, hao_wu, liaolj}@bit.edu.cn

Abstract

Generative adversarial networks (GANs) have shown impressive results, however, the generator and the discriminator are optimized in finite parameter space which means their performance still need to be improved. In this paper, we propose a novel approach of adversarial training between one generator and an exponential number of critics which are sampled from the original discriminative neural network via dropout. As discrepancy between outputs of different sub-networks of a same sample can measure the consistency of these critics, we encourage the critics to be consistent to real samples and inconsistent to generated samples during training, while the generator is trained to generate consistent samples for different critics. Experimental results demonstrate that our method can obtain state-of-the-art Inception scores of 9.17 and 10.02 on supervised CIFAR-10 and unsupervised STL-10 image generation tasks, respectively, as well as achieve competitive semi-supervised classification results on several benchmarks. Importantly, we demonstrate that our method can maintain stability in training and alleviate mode collapse.

1 Introduction

Generative adversarial networks [Goodfellow *et al.*, 2014] are one of the most prominent approaches of generative models. They provide an attractive approach to train generative models that directly map hidden codes to real-life data distribution, and are widely used in a number of tasks such as high quality image generation [Berthelot *et al.*, 2017; Grinblat *et al.*, 2017; Takeru Miyato, 2018] and semi-supervised classification [Springenberg, 2015; Dai *et al.*, 2017; LI *et al.*, 2017a].

Despite the success of GANs, training GANs to converge to an equilibrium point is still a challenge because, theoretically, the generator and the discriminator are required to be updated directly in function space. However, the generator and the discriminator are presented with deep nets and are optimized in finite parameter space [Goodfellow, 2017]. What's more, the notorious gradient vanishing problem of sigmoid function stops the generator from learning anything when the discriminator is too strong [Arjovsky *et al.*, 2017], causing unstable training process and even mode collapse.

Existing GAN variants of controlling discriminator's performance fall into two branches: revising the objective of discriminator [Arjovsky *et al.*, 2017; Nowozin *et al.*, 2016; Mao *et al.*, 2016], or applying multiple discriminators [Arora *et al.*, 2017; Nguyen *et al.*, 2017; Durugkar *et al.*, 2016]. Many modified objective variants are proven more effective than the original one in theory, however, there is a lack of evidence in practical application [Lucic *et al.*, 2017].

Some recent studies have shown that multiple discriminators (aka critics) can be trained to alleviate the training instability and mode collapse problems. Traditional methods such as MIX+GAN [Arora *et al.*, 2017], GMAN [Durugkar *et al.*, 2016] and D2GAN [Nguyen *et al.*, 2017] are assembled models which use several deep neural networks as critics, while [Liu and Tuzel, 2016] is to enforce a weight-sharing constraint between a pair of GANs. But the critics in these methods are limited to a small number, otherwise parameter explosion will make them impractical for computation.

The motivation of our method is to create nearly infinite numbers of critics with tolerable expense. Additionally, we require these critics to coordinately reflect different aspects of criterions to avoid redundancy. That is, they should be consistently supportive when judging real samples, but can perform variously to distinguish different defects of generated samples, such as "blur", "incomplete", or "distortion".

Under such inspiration, in this paper, we construct an exponential number of critics by the *Dropout* technique. Such a manner has three advantages. First, original adversarial objective can be optimized in nearly infinite parameter space thanks to an exponential number of candidate sub-networks selected via dropout; Second, these critics share parameters to avoid parameter explosion; The last, it can keep the benefit of avoiding feature co-adaptation which dropout has shown [Srivastava *et al.*, 2014].

Furthermore, we take advantage of adversarial consistency to reduce critics redundancy. The critics are trained to be consistent to the scores of real images, and be inconsistent to the scores of generated images. As different critics can be seen as different feature extractors and each critic detects a subset of features, requiring the critics to be consistent to real images is equal to asking critics evaluate real image as real with high certainty even when some features are not detected; and

requiring the critics to be inconsistent to generated images is equal to encouraging critics explore different criterions to evaluate images. For the generator, accordingly, its goal is to generate images along the manifold that the critics are consistent to. We refer to our approach as Consistent Adversarial Training Enhanced GANs (CAGAN). Specifically, the main contributions of this paper are: (1) We construct an exponential number of critics with the dropout technique. (2) We propose consistent adversarial training constraints and apply them to enhance GAN models. (3) Experimental results illustrate that CAGAN achieves state-of-the-art Inception scores of 9.17 and 10.02 on CIFAR-10 and STL-10 respectively, as well as obtains competitive results on semi-supervised benchmarks. (4) We demonstrate that CAGAN can maintain a stable training process of WGAN-GP and alleviate the mode collapse problem of Improved GAN.

2 Related Work

The framework of GANs [Goodfellow *et al.*, 2014] was proposed to estimate generative models via an adversarial process. And it has attracted huge attention since DCGAN [Radford *et al.*, 2015] showed its impressive results on image generation. However, the well-known delicate and unstable training process of GANs makes it a persisting challenge to control the performance of the discriminator [Takeru Miyato, 2018]. Many variants are proposed to solve the problem or make improvement, and they can be divided into two main categories: modifying the objective of the discriminator, or employing additional discriminators to feed back useful gradient information to the generator.

There are several recent works attempting to revise the loss function of the discriminator. f-GAN [Nowozin *et al.*, 2016] is proposed to generalize GAN to f-divergence based on the observation that the Jensen-Shannon divergence is a special case of f-divergence. WGAN [Arjovsky *et al.*, 2017] is proposed to use Earth-Mover distance as its objective function. BEGAN [Berthelot *et al.*, 2017] uses an auto-encoder as the discriminator and optimizes a lower bound of the Wasserstein distance between auto-encoder loss distributions on real and fake data. LSGAN [Mao *et al.*, 2016] proposes a least-squares loss function for the discriminator and shows that minimizing the objective is equal to minimizing Pearson χ^2 divergence. In spite of their theoretical proof of solving the instability of GAN, a large-scale study provided by [Lucic *et al.*, 2017] found that there is no evidence that any of these methods outperform the original GAN.

An alternative direction is to train multiple discriminators. GMAN [Durugkar *et al.*, 2016] trains many discriminators to boost the learning of generator. [Warde-Farley and Bengio, 2017] propose DFM to assist the generator to generate images that match the statistics of real samples with a Denoising AutoEncoder. MIX+GAN [Arora *et al.*, 2017] proves that there exists an equilibrium in infinite mixture deep nets, and shows that training a mixture of generator and discriminators can stabilize training as well as improve the performance in some cases. AdaGAN [Tolstikhin *et al.*, 2017] proposes an iterative procedure to add a new component into a mixture model by running a GAN on a re-weighted sample. D2GAN [Nguyen *et al.*, 2017] employs two discriminators, one of which rewards high scores for real samples and the other one favorites generated samples.

Apart from above mentioned methods, there have been other efforts in improving GANs. [Springenberg, 2015] proposes CatGAN to replace the binary discriminator in original GAN with a multi-class classifier. ALI [Dumoulin *et al.*, 2016] adds an inference model to GANs and jointly learns a generation net. Triple GAN [LI *et al.*, 2017a] adds a classifier to help GAN framework characterize conditional distribution. Bayesian GAN [Saatchi and Wilson, 2017] proposes a framework to marginalize the weights of the generator and discriminator nets. Bad GAN [Dai *et al.*, 2017] shows that bad GAN is the requirement for good semi-supervised classification performance. SN-GANs [Takeru Miyato, 2018] propose to use spectral normalization to stabilize the training of discriminator. MGAN [Hoang *et al.*, 2018] proposes an adversarial learning process between multiple generators and a discriminator, as well as a classifier specifying which generator a sample comes from. MMD GAN [Li *et al.*, 2017b] proposes adversarial kernel learning to improve the model expressiveness and computational efficiency.

Dropout [Srivastava *et al.*, 2014] is an effective method to avoid network from over-fitting and can prevent units from co-adapting. [Pathak *et al.*, 2016] use dropout as a manner of avoiding over-fitting and obtaining impressive result in semantic in-painting. In order to check the 1-Lipschitz continuity in real data manifold, CT-GAN [Wei *et al.*, 2018] inputs a real sample to the critic net twice via dropout in hidden layers, and compares the difference between outputs. Recently AM-GAN [Zhou *et al.*, 2018] appends dropout to the discriminator for only supervised image generation and studies how class labels and associated losses influence GAN's training.

3 Preliminaries

The original GAN framework consists of a generator G and a discriminator D. To give the generator G the ability of mapping random noise $\mathbf{z} \sim p(\mathbf{z})$ to the real data distribution $\mathbf{x} \sim \mathbb{P}_r$, the discriminator is trained to tell apart fake samples $\tilde{\mathbf{x}} = G(\mathbf{z})$ from the real input data, and the generator is optimized to generate plausible samples that fool the discriminator. The minimax game between D and G is:

$$\min_G \max_D \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathbb{P}_r}[\log(D(\mathbf{x}))] + \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}\sim\mathbb{P}_g}[\log(1 - D(\tilde{\mathbf{x}}))] \quad (1)$$

where $\mathbb{P}_r$ and $\mathbb{P}_g$ are real data distribution and generated data distribution respectively.

3.1 WGAN and WGAN-GP

The Wasserstein GAN [Arjovsky *et al.*, 2017] was proposed to address the problem of instability in GAN training. In order to avoid the discontinuities and vanishing gradients of the original GAN, WGAN was proposed to use Earth-Mover distance as the evaluation of discrepancy between real data distribution and model distribution. The objectives of the critic and the generator are derived as:

$$\mathcal{L}_D^{WGAN} = \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}\sim\mathbb{P}_g}[D(\tilde{\mathbf{x}})] - \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathbb{P}_r}[D(\mathbf{x})] \quad (2)$$

$$\mathcal{L}_G^{WGAN} = -\mathop{\mathbb{E}}_{\mathbf{z}\sim p(\mathbf{z})}[D(G(\mathbf{z}))] \tag{3}$$

and the critic D should be restricted to the space of 1-Lipschitz functions $\|D\|_L \leq 1$ which is imposed by weight clipping to lay the weights of the critic network within a compact space $[-c, c]$, say, $[-0.01, 0.01]$.

[Gulrajani *et al.*, 2017] gives an alternative way of imposing the 1-Lipschitz continuity by appending a gradient penalty term to the objective of the critic:

$$\mathcal{L}_D^{GP} = \mathcal{L}_D^{WGAN} + \lambda_{GP}\mathop{\mathbb{E}}_{\hat{\mathbf{x}}\sim\mathbb{P}_{\hat{\mathbf{x}}}}\left[(\|\nabla_{\hat{\mathbf{x}}}D(\hat{\mathbf{x}})\|_2 - 1)^2\right] \tag{4}$$

where $\hat{\mathbf{x}}$ is uniformly sampled from straight lines between generated and real sample points. λ_{GP} is often set to 10, and the loss function of the generator is kept the same as the original WGAN, i.e. $\mathcal{L}_G^{GP} = \mathcal{L}_G^{WGAN}$.

3.2 Improved GAN

The Improved GAN [Salimans *et al.*, 2016] generalizes the objective of original discriminator from 2-class classification problem to $K+1$ classes case where real samples are associated with class labels $y \in \{1, ..., K\}$ and generated samples correspond to the $(K+1)^{th}$ label. Such a generalization is suitable for semi-supervised learning. The objective of the discriminator is:

$$\begin{aligned}\mathcal{L}_D^{Imp} = &-\mathop{\mathbb{E}}_{\mathbf{x},y\sim\mathbb{P}_{\mathbf{x},y}}[\log D(y|\mathbf{x})] - \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}\sim\mathbb{P}_g}[\log D(K+1|\tilde{\mathbf{x}})] \\ &- \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathbb{P}_r}[\log(1 - D(K+1|\mathbf{x}))]\end{aligned} \tag{5}$$

and the objective of the generator changes to generate data that match feature statistic of real data:

$$\mathcal{L}_G^{Imp} = \left\| \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathbb{P}_r} f(\mathbf{x}) - \mathop{\mathbb{E}}_{\mathbf{z}\sim p(\mathbf{z})} f(G(\mathbf{z})) \right\|^2 \tag{6}$$

where $f(.)$ denotes the activation on an intermediate layer of the discriminator.

4 Methods

In this section, we describe our methodology, the basic idea, design and implementation of our Consistent Adversarial Training Enhanced GAN (CAGAN).

4.1 Multiple Critics Construction via Dropout

Recent studies have shown that multiple critics can be trained to alleviate the training instability and mode collapse problem of GAN. The motivation of our method is to create nearly infinite numbers of critics with tolerable expense.

We propose to produce multiple critics by randomly selecting sub-networks from a deep neural networks via dropout. It is realized by adding dropout layers to discriminator. Suppose the keep probabilities of dropouts are set to be 0.5, a discriminator has M hidden layers and each hidden layer has averagely N nodes, then the number of different critics that can be generated by dropouts will be $2^{M\times N}$. For example, in ResNet, M is 3 and N is $8 \times 8 \times 128$, thus it will generate approximately 2^{24576} sub-networks via dropout, which is nearly infinite in practice.

The above way of using the dropout technique to generate multiple critics has three benefits: (1) The adversarial game can be optimized in an exponential and nearly infinite space. (2) A huge number of critics are constructed while parameter explosion is avoided because they share parameters. (3) It can prevent the generator from over-fitting the finite discrete data. Without dropout, networks are trained on a finite discrete dataset $\tilde{\mathbb{P}}_r$ sampled from actual continuous infinite real distribution $\mathbb{P}_r$. As a result, the global optimum $\mathbb{P}_g = \tilde{\mathbb{P}}_r$ of this game fails to capture the structure of $\mathbb{P}_r$ [Durugkar *et al.*, 2016]. But our method can allow the game escape the degenerate situation where $\mathbb{P}_g = \tilde{\mathbb{P}}_r$ when converging.

4.2 Consistent Adversarial Training Objectives

Under the condition of multiple critics, we propose a consistent adversarial training method to train the generator and multiple critics. Our basic idea is to require these critics to coordinately reflect different aspect of criterions to avoid redundancy. Concretely, in each iteration, we randomly select two temporal subnets from the discriminator neural network via dropout, and require the two critics to be consistent to real samples and be inconsistent to generated samples when training D. Then G is optimized to generate samples that critics are consistent to. Formally, the consistent adversarial objectives of the critics D and the generator G are:

$$\mathcal{L}_D^{CA} = \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathbb{P}_r}[C(\mathbf{x})] - \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}\sim\mathbb{P}_g}[C(\tilde{\mathbf{x}})] \tag{7}$$

$$\mathcal{L}_G^{CA} = \mathop{\mathbb{E}}_{\mathbf{z}\sim p(\mathbf{z})}[C(G(\mathbf{z}))] \tag{8}$$

where $C(.)$ is the function of evaluating the consistency of two critics on a sample. On the image generation task:

$$C^{gen}(\mathbf{x}) = \|D_1(\mathbf{x}) - D_2(\mathbf{x})\|^2 + \frac{\lambda_f}{d_f}\|f_1(\mathbf{x}) - f_2(\mathbf{x})\|^2 \tag{9}$$

$D_1(.)$ and $f_1(.)$ correspond to the output and the penultimate activation of the first randomly selected critic, while $D_2(.)$ and $f_2(.)$ are for the second critic. d_f is the dimension of the penultimate layer and λ_f is a hyper-parameter. For semi-supervised learning, the consistency can be written as:

$$\begin{aligned}C^{semi}(\mathbf{x}) \;=\; &\frac{1}{d_c}\|\mathrm{Softmax}(D_1(\mathbf{x})) - \mathrm{Softmax}(D_2(\mathbf{x}))\|^2 \\ &+ \frac{\lambda_f}{d_f}\|f_1(\mathbf{x}) - f_2(\mathbf{x})\|^2\end{aligned} \tag{10}$$

where d_c is the dimension of the critic's output layer.

4.3 Equilibrium of Consistency Loss

As the consistency loss needs to be combined with the original generative adversarial loss to train generator and critics, in practice, the consistency loss of critics for generated samples will increase explosively when the update times of critics are more than that of generator during each iteration (such as WGAN and WGAN-GP). To balance the consistency loss between real images and generated images, we borrow the idea of Proportional Control Theory (which is also used in [Berthelot *et al.*, 2017]) to maintain such an equilibria condition:

$$\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathbb{P}_r}[C(\mathbf{x})] = \mathop{\mathbb{E}}_{\mathbf{z}\sim p(\mathbf{z})}[C(G(\mathbf{z}))] \tag{11}$$

The objectives of our consistent adversarial training are derived as:

$$\begin{cases} \mathcal{L}_D^{CA} = \mathbb{E}_{\mathbf{x}}[C(\mathbf{x})] - k_t \mathbb{E}_{\mathbf{z}}[C(G(\mathbf{z}))] & \text{for } D \\ \mathcal{L}_G^{CA} = \mathbb{E}_{\mathbf{z}}[C(G(\mathbf{z}))] & \text{for } G \\ k_{t+1} = k_t + \lambda_k(\mathcal{L}_R^{CA} - \mathcal{L}_G^{CA}) & \text{for } k \end{cases} \tag{12}$$

where $\mathcal{L}_R^{CA} = \mathbb{E}_{\mathbf{x}}[C(\mathbf{x})]$. We initialize $k_0 = 0$, and update k with learning rate $\lambda_k = 0.001$ after each generator iteration.

4.4 Our WGAN-GP Enhancement

Based on the above design of our CAGAN, we implement it to enhance WGAN-GP and apply it to the image generation task. Using Equation 9 to measure consistency, we combine Equation 12 with the objectives of WGAN-GP as the objectives of our CAGAN for image generation task:

$$\begin{cases} \mathcal{L}_D^{gen} = \mathcal{L}_D^{GP} + \lambda_{CA}\mathcal{L}_D^{CA} \\ \mathcal{L}_G^{gen} = \mathcal{L}_G^{GP} + \lambda_{CA}\mathcal{L}_G^{CA} \end{cases} \tag{13}$$

4.5 Our Improved GAN Enhancement

We further implement our CAGAN to enhance the Improved GAN and apply it to the semi-supervised classification task. Employing Equation 10 evaluating consistency, we combine Equation 12 with the objectives of Improved GAN as the objectives of our CAGAN for semi-supervised classification:

$$\begin{cases} \mathcal{L}_D^{semi} = \mathcal{L}_D^{Imp} + \lambda_{CA}\mathcal{L}_D^{CA} \\ \mathcal{L}_G^{semi} = \mathcal{L}_G^{Imp} + \lambda_{CA}\mathcal{L}_G^{CA} \end{cases} \tag{14}$$

5 Experiments

In this section, we evaluate the performance of CAGAN on image generation and semi-supervised classification tasks.

5.1 Image generation

Datasets and Evaluation Protocols

To investigate the effectiveness of our CAGAN on image generation task, we conduct experiments on two benchmark datasets: CIFAR-10 [Krizhevsky, 2009] and STL-10 [Coates *et al.*, 2011]. CIFAR-10 contains 50,000 labeled training images of size 32×32 from 10 classes. We use it under unsupervised and supervised settings. STL-10 is subsampled from ImageNet which is more diverse than CIFAR-10, and it contains 100,000 unlabeled images of size 96×96. To compare with other methods, we rescale the STL-10 images down to 48×48 as the same as other methods. We use *Inception* score for quantitative evaluation. Following [Salimans *et al.*, 2016], we compute the average Inception score over 10 independent groups of 5,000 randomly generated samples for CIFAR-10 and STL-10.

Network Structure and Hyper-parameters

We use ResNet designed by [Gulrajani *et al.*, 2017] for fair comparison except that we append dropouts with unit keep probability of (1.0, 0.8, 0.5, 0.5) to the four residual blocks, respectively. We keep the hyper-parameters all the same on CIFAR-10 and STL-10 for all the experiments. In particular, we follow original WGAN-GP set $\lambda_{GP} = 10$, mini-batch size of 64 when training D, and mini-batch size of 128 when training G. We use Adam optimizer with a learning rate of 0.0002, $\beta_1 = 0$, $\beta_2 = 0.9$ to train G and D, and the learning rate is decreased linearly to 0. For consistent adversarial hyper-parameters, we set $\lambda_f = 0.1, \lambda_{CA} = 2$, and training totally 700 epochs. For the image generation task, we modify k_t to $0.1 + k_t$ in Equation 12, where 0.1 is a pseudo factor in order to guarantee the critics has an effect on evaluating consistency of the generated samples.

Comparison to State-of-the-art

We report the Inception scores obtained by our CAGAN and comparative state-of-the-art methods (which are introduced in the Related Work section) in Table 1. The $2\times$ filters indicate that the number of feature maps in each convolutional layer of both the generator net and the critic net are doubled from 128 to 256. Photo-realistic random generated images are shown in Figure 1.

Overall, our proposed CAGAN consistently outperforms other methods on these tasks. It achieves new record inception scores of 9.17 and 10.02 on supervised CIFAR-10 and unsupervised STL-10, and the shape of generated images on supervised CIFAR-10 are demonstrated in Figure 1(b). To the best of our knowledge, our CAGAN is the first to exceed the inception score of 9 with a remarkable margin on CIFAR-10, and also the first to reach inception score of 10 on STL-10.

It is worth noting that on unsupervised CIFAR-10, the inception scores of WGAN-GP and Splitting GAN [Grinblat *et al.*, 2017] both drop when feature maps are doubled. In contrast, our model gains better performance. We think more parameters in their networks needs to be co-adapted, which restricts the expressive ability of their networks. But the critics selected via dropout can avoid this, and doubled feature maps allow critics to increase their capacity to explore more useful information. Thus they can characterize data distribution and capture class structure better.

Ablation Studies

In order to further understand the effect of each component in our model, we conduct comprehensive ablation studies on the unsupervised CIFAR-10 image generation task. All ablated models below share the same hyper-parameters during training.

(a) WGAN-GP: Original gradient penalty WGAN. (b) WGAN-GP (w/ dropout): The same mode as WGAN-GP except that dropouts are added to the discriminator network. (c) WGAN-CT: Only consistency loss of real samples being added to the critic loss, that is, critics are only constrained to be consistent to real samples. (d) WGAN-CT-G: Consistency loss of real samples being added to the critic loss, meanwhile, consistency loss of fake samples being added to the generator loss. (e) CAGAN: The full model that trains critics to be consistent to real samples and inconsistent to generated samples, and optimizes the generator to generate images that critics are consistent to. (f) CAGAN (w/ pseudo factor): The final model that we use for the image generation task.

Their Inception scores are illustrated in Table 2. We also plot the Wasserstein estimate curves of the entire learning process of these models in Figure 2. Comparing the inception scores and Wasserstein estimates, we summarize our results as follows:

Model	CIFAR-10		STL-10
	Unsupervised	Supervised	
(Real data)	11.24 ± 0.12		26.08 ± 0.26
DCGANs	6.16 ± 0.07	6.58	7.84 ± 0.07
DFM	7.72 ± 0.13	–	8.51 ± 0.13
WGAN-GP	7.86 ± 0.07	8.42 ± 0.10	9.05 ± 0.12
Splitting GAN	7.90 ± 0.09	8.73 ± 0.08	9.50 ± 0.13
CT-GAN	8.12 ± 0.12	8.81 ± 0.13	–
SN-GANs	8.24 ± 0.08	8.59 ± 0.12	9.04 ± 0.12
MGAN	8.33 ± 0.10	–	9.22 ± 0.11
our CAGAN	$\mathbf{8.35 \pm 0.09}$	$\mathbf{8.89 \pm 0.11}$	$\mathbf{9.51 \pm 0.14}$
WGAN-GP ($2\times$ filters)	7.81 ± 0.10	8.67 ± 0.14	–
Splitting GAN ($2\times$ filters)	7.80 ± 0.08	8.87 ± 0.09	–
AM-GAN ($\approx 2\times$ filters)	–	8.91 ± 0.11	–
our CAGAN ($2\times$ filters)	$\mathbf{8.42 \pm 0.07}$	$\mathbf{9.17 \pm 0.13}$	$\mathbf{10.02 \pm 0.13}$

Table 1: Inception scores of image generation on unsupervised/supervised CIFAR-10 and unsupervised STL-10.

(a) CIFAR-10 32×32 unsupervised (b) CIFAR-10 32×32 supervised (c) STL-10 48×48 unsupervised

Figure 1: Generated images by our CAGAN model.

Model	Inception score
(a) WGAN-GP	7.86 ± 0.07
(b) WGAN-GP (w/ dropout)	7.71 ± 0.07
(c) WGAN-CT	8.12 ± 0.12
(d) WGAN-CT-G	8.23 ± 0.09
(e) CAGAN	8.29 ± 0.10
(f) CAGAN (w/ pseudo factor)	$\mathbf{8.35 \pm 0.07}$

Table 2: Ablation studies of CAGAN on unsupervised CIFAR-10.

(1) For WGAN-GP, its training process is unstable and the Wasserstein estimate rises up during a long continuous epoch period. We think it is because the only critic in WGAN-GP can evaluate samples well with all features, but is not good at transferring valuable improvement suggestions to the generator. The critic is too strong for the generator to keep up with it. (2) When only adding dropout to WGAN-GP, its exponential critics decrease the Wasserstein estimate but increase its instability. The probable reason is that different critics extract different features to evaluate samples, but they give divergent information to the generator and disturb its learning process. (3) When consistent adversarial constraints are added, the other four models accordantly perform much more stably.

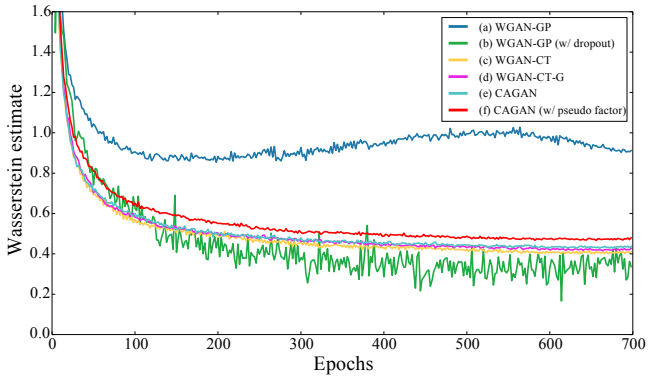


Figure 2: Wasserstein estimate of each model on unsupervised CIFAR-10 image generation task.

But to our surprise, from Table 2 and Figure 2, these models consistently agree with that higher Wasserstein estimate reward higher Inception score. After analysis, we conclude that stable but higher Wasserstein estimate promise generator more exploration space, thus multiple critics can provide various and meaningful guidance to the generator to generate better images.

Methods	MNIST (# errors)	SVHN (% errors)	CIFAR-10 (% errors)
CatGAN	191 ± 10	–	19.58 ± 0.46
Improved GAN	93 ± 6.5	8.11 ± 1.3	18.63 ± 2.32
ALI	–	7.42 ± 0.65	17.99 ± 1.62
Triple-GAN	91 ± 58	5.77 ± 0.17	16.99 ± 0.36
CT-GAN	89 ± 13	–	–
Bayesian GAN	89 ± 3.4	14.1 ± 2.3	22.8 ± 2.4
our CAGAN	$\mathbf{81.9 \pm 4.5}$	$\mathbf{4.83 \pm 0.09}$	$\mathbf{12.61 \pm 0.12}$
Bad GAN (FM+VI)	86.5 ± 10.6	5.29	14.41 ± 0.30
Bad GAN (FM+LD)	$\mathbf{79.5 \pm 9.8}$	–	–
Bad GAN (FM+PT+Ent)	–	$\mathbf{4.25 \pm 0.03}$	–

Table 3: Comparison with state-of-the-art methods on 3 benchmarks. Only GAN based methods without data augmentation are included.

5.2 Semi-Supervised Image Classification

Datasets

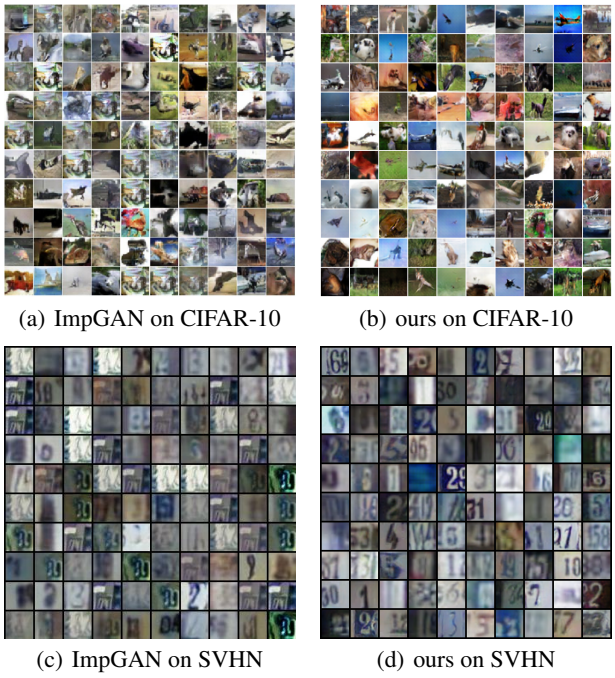
To demonstrate that our method can enhance the Improved GAN on the semi-supervised classification task, we gather three widely used benchmark datasets: MNIST, SVHN, and CIFAR-10. The same to other methods, we randomly select 100, 4,000, and 1,000 labeled images from MNIST, CIFAR-10 and SVHN datasets as supervision, and the entire training set is used for unsupervised training. We evaluate the results with classification error on the testing set of each dataset.

Network Structure and Hyper-parameters

For MNIST, We keep the network structure the same as Improved GAN. The generator and classifier are initialized with $\mathcal{N}(0, 0.1)$. We use batch size of 100, $\lambda_f = 0, \lambda_{CA} = 0.2$, and Adam optimizer with learning rate of 0.003, $\beta_1 = 0.5, \beta_2 = 0.95$ to train generator and classifier 300 epochs. For CIFAR-10, we use the same network structure as Improved GAN, both generator and classifier use 128 feature maps and are initialized with $\mathcal{N}(0, 0.01)$. We use batch size of 100, learning rate of 0.0003, $\beta_1 = 0.5$, $\beta_2 = 0.99$, and $\lambda_f = 0.1, \lambda_{CA} = 1$ to train networks 1,000 epochs. For SVHN, the experiment settings are the same with CIFAR-10 except that the batch size is set to 64 and the total number of epochs is 300.

Results

The results are reported in Table 3. For MNIST, our result is averaged over 10 random seeds. For SVHN and CIFAR-10, the mean and standard deviation are obtained from 5 times of repetitive running. Our results consistently exceed the baseline model Improved GAN with a significant margin. We find a recent Bad GAN method [Dai *et al.*, 2017] also achieves impressive results on the semi-supervised classification task, but their best results are obtained by combining different models to specifically fit different datasets. Mode collapse is a notorious problem of GAN, which can be clearly observed when GANs are applied to semi-supervised learning. The Improved GAN encounters mode collapse. Inspiringly, we find that our method can alleviate the mode collapse of Improved GAN and generate diverse samples, as shown in Figure 3.



(a) ImpGAN on CIFAR-10 (b) ours on CIFAR-10

(c) ImpGAN on SVHN (d) ours on SVHN

Figure 3: Image generation comparison between Improved GAN and our model.

6 Conclusion

This paper proposes an approach of consistent adversarial training between a generator and an exponential number of critics generated via dropout. The critics are trained to be consistent to real samples and inconsistent to generated samples, so that the critics can explore more effective features to evaluate samples, and the generator is then trained to generate samples that critics are consistent to. Experimental results demonstrate that we obtain new state-of-the-art records on supervised and unsupervised image generation tasks as well as achieve competitive results on semi-supervised benchmarks. We also show that our method can maintain the training stability of WGAN-GP and alleviate mode collapse problem of Improved GAN. With a strong generalization ability, we will apply our model to improve more GANs in our future work.

Acknowledgments

We thank the anonymous reviewers for their helpful comments. This work was supported by National Key Research and Development Program of China (2016YFB1000902) and National Natural Science Foundation of China (61472040, 61751217). Dandan Song is the corresponding author.

References

[Arjovsky *et al.*, 2017] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, pages 214–223, 2017.

[Arora *et al.*, 2017] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). In *ICML*, pages 224–232, 2017.

[Berthelot *et al.*, 2017] David Berthelot, Tom Schumm, and Luke Metz. BEGAN: boundary equilibrium generative adversarial networks. *CoRR*, abs/1703.10717, 2017.

[Coates *et al.*, 2011] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, 2011.

[Dai *et al.*, 2017] Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Ruslan R Salakhutdinov. Good semi-supervised learning that requires a bad gan. In *NIPS*, pages 6513–6523. 2017.

[Dumoulin *et al.*, 2016] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martín Arjovsky, Olivier Mastropietro, and Aaron C. Courville. Adversarially learned inference. *CoRR*, abs/1606.00704, 2016.

[Durugkar *et al.*, 2016] Ishan P. Durugkar, Ian Gemp, and Sridhar Mahadevan. Generative multi-adversarial networks. *CoRR*, abs/1611.01673, 2016.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680. 2014.

[Goodfellow, 2017] Ian Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.00160, 2017.

[Grinblat *et al.*, 2017] Guillermo L. Grinblat, Lucas C. Uzal, and Pablo M. Granitto. Class-Splitting Generative Adversarial Networks. *ArXiv e-prints*, September 2017.

[Gulrajani *et al.*, 2017] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NIPS*, pages 5769–5779. 2017.

[Hoang *et al.*, 2018] Quan Hoang, Tu Dinh Nguyen, Trung Le, and Dinh Phung. MGAN: Training generative adversarial nets with multiple generators. In *ICLR*, 2018.

[Krizhevsky, 2009] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[LI *et al.*, 2017a] Chongxuan LI, Taufik Xu, Jun Zhu, and Bo Zhang. Triple generative adversarial nets. In *NIPS*, pages 4091–4101. 2017.

[Li *et al.*, 2017b] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabas Poczos. MMD GAN: Towards deeper understanding of moment matching network. In *NIPS*, pages 2203–2213. 2017.

[Liu and Tuzel, 2016] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *NIPS*, pages 469–477. 2016.

[Lucic *et al.*, 2017] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are GANs Created Equal? A Large-Scale Study. *ArXiv e-prints*, November 2017.

[Mao *et al.*, 2016] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, and Zhen Wang. Multi-class generative adversarial networks with the L2 loss function. *CoRR*, abs/1611.04076, 2016.

[Nguyen *et al.*, 2017] Tu Nguyen, Trung Le, Hung Vu, and Dinh Phung. Dual discriminator generative adversarial nets. In *NIPS*, pages 2667–2677. 2017.

[Nowozin *et al.*, 2016] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *NIPS*, pages 271–279. 2016.

[Pathak *et al.*, 2016] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, June 2016.

[Radford *et al.*, 2015] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

[Saatchi and Wilson, 2017] Yunus Saatchi and Andrew G Wilson. Bayesian GAN. In *NIPS*, pages 3625–3634. 2017.

[Salimans *et al.*, 2016] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In *NIPS*, pages 2234–2242. 2016.

[Springenberg, 2015] Jost Tobias Springenberg. Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks. *ArXiv e-prints*, November 2015.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 15:1929–1958, 2014.

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

[Tolstikhin *et al.*, 2017] Ilya O Tolstikhin, Sylvain Gelly, Olivier Bousquet, Carl-Johann SIMON-GABRIEL, and Bernhard Schölkopf. Adagan: Boosting generative models. In *NIPS*, pages 5430–5439. 2017.

[Warde-Farley and Bengio, 2017] David Warde-Farley and Yoshua Bengio. Improving generative adversarial networks with denoising feature matching. 2017.

[Wei *et al.*, 2018] Xiang Wei, Zixia Liu, Liqiang Wang, and Boqing Gong. Improving the improved training of wasserstein gans. *ICLR*, 2018.

[Zhou *et al.*, 2018] Zhiming Zhou, Han Cai, Shu Rong, Yuxuan Song, Kan Ren, Weinan Zhang, Jun Wang, and Yong Yu. Activation maximization generative adversarial nets. *ICLR*, 2018.