entropy

MDPI

Article

Utilizing Amari-Alpha Divergence to Stabilize the Training of Generative Adversarial Networks

Likun Cai [1,2,3,*], Yanjie Chen [1,2,3], Ning Cai [1], Wei Cheng [4] and Hao Wang [1]

1 School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China; chenyj1@shanghaitech.edu.cn (Y.C.); ningcai@shanghaitech.edu.cn (N.C.); wanghao1@shanghaitech.edu.cn (H.W.)
2 Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 200050, China
3 University of Chinese Academy of Sciences, Beijing 100049, China
4 NEC Laboratories America, Inc. (NEC Labs), NEC Corporation, Princeton, NJ 08540, USA; weicheng@nec-labs.com
* Correspondence: cailk@shanghaitech.edu.cn

Received: 25 February 2020; Accepted: 31 March 2020; Published: 4 April 2020

Abstract: Generative Adversarial Nets (GANs) are one of the most popular architectures for image generation, which has achieved significant progress in generating high-resolution, diverse image samples. The normal GANs are supposed to minimize the Kullback–Leibler divergence between distributions of natural and generated images. In this paper, we propose the Alpha-divergence Generative Adversarial Net (Alpha-GAN) which adopts the alpha divergence as the minimization objective function of generators. The alpha divergence can be regarded as a generalization of the Kullback–Leibler divergence, Pearson χ^2 divergence, Hellinger divergence, etc. Our Alpha-GAN employs the power function as the form of adversarial loss for the discriminator with two-order indexes. These hyper-parameters make our model more flexible to trade off between the generated and target distributions. We further give a theoretical analysis of how to select these hyper-parameters to balance the training stability and the quality of generated images. Extensive experiments of Alpha-GAN are performed on SVHN and CelebA datasets, and evaluation results show the stability of Alpha-GAN. The generated samples are also competitive compared with the state-of-the-art approaches.

Keywords: Alpha divergence; generative adversarial network; unsupervised image generation; deep neural networks

1. Introduction

In recent years, deep learning has achieved incredible performance in theoretical research and many application scenarios, such as image classification [1,2], natural language processing [3], and speech recognition [4]. For high-dimensional data generation, deep neural networks-based generative models, particularly Generative Adversarial Networks (GANs) [5] have quickly become the most powerful model in unsupervised learning of image generation. Compared with the previous strategies, GANs have the power to generate high-resolution and vivid images. In a nutshell, GANs provide a framework to learn implicit distribution of a given target dataset $\mathcal{X}$. There are typically two networks in GANs architecture: a generator network $G(\cdot)$ that produces vivid images, and a discriminator network $D(\cdot)$ that outputs scores on input images. The generator $G(\cdot)$ adopts a given latent noise z as input which is sampled from arbitrary

distribution. The discriminator $D(\cdot)$ measures the difference between distributions of the generated fake samples and the real data. The core concept of GANs is to simultaneously train the discriminator and generator in turn by reducing the gap between two distributions under certain distance measurements. Recent work indicates that GANs have been making remarkable progress in a wide range of applications including image, video generation [6], image-to-image translation [7], and image super-resolution [8].

The original GAN model introduced by Goodfellow et al. proposed to minimize the Jensen–Shannon (JS) divergence between p_{real} and p_{fake}, where p_{real} denotes the distribution of input real data and p_{fake} stands for the distribution of generated data. The cross-entropy loss is used for the output unit of discriminator. There exist two major challenges in the training of GANs. One is the performance balance between the generator and the discriminator. In the practical training process of GANs, the discriminators tend to learn better than the generators in most cases. As a result, it is difficult for the generators to study from data since the loss of discriminators becomes close to 0. Another problem is the model collapsing, which means the generated samples will collapse within a few data modes and become lack of diversity. In the literature, tremendous effort has been made to improve not only the ability to generate high-quality images but also convergence stability. For example, Least Square GAN (LSGAN) [9] leverages the Pearson χ^2 divergence and adopts the least square loss for critic output. In [10], the authors proposed a new mechanism f-GAN to give an elegant generalization of GAN and extended the value function to arbitrary f-divergence. Compared with the original GANs, another contribution of f-GAN is that only single-step back-propagation is needed. Thus, there is no inner loop in the algorithm. However, the model collapsing problem remains unsolved by f-GAN [11].

To overcome the existing problems of GANs, one effective mechanism is proposed by Arjovsky et al.: the Wasserstein-GAN (i.e., WGAN) [12]. There are two main improvements in WGAN: a new objective based on the Wasserstein distance (or Earth Mover distance) and the weight clipping method. The Wasserstein distance has been proved to have better convergence performance than Kullback–Leibler divergence and Jensen–Shannon divergence in [12]. WGAN applies the approximated *Wasserstein distance* to estimate the distance between real and fake samples on a discriminator. The Kantorovich–Rubinstein duality is used to formulate the optimization. The original WGAN also requires the discriminator to be 1-Lipschitz continuous function, which can be achieved by clamping the model weights within a compact space ($\mathcal{W} = [-0.01, 0.01]^l$), called weight clipping method. With these two methods to improve the model stability, WGAN can train the model till optimality to make the model less prone to collapse. However, this approach may lead to undesired behavior in practice [13]. To alleviate this effect, Gulrajani et al. proposed WGAN with gradient penalty (WGAN-GP) [13]. WGAN-GP introduces a soft penalty for the violation of 1-Lipschitz constraint, which guarantees high-quality image generation at the cost of the increasing computation complexity. Recently, many researchers have paid more attention to optimizing network architecture to improve training stability. For example, SN-GANs [14] create a novel weight normalization technique which is called Spectral Normalization to stabilize the training process. Although these approaches effectively improve the training stability, they provide less flexibility to strike a balance between the training stability and the desired quality of generated images.

In this paper, we propose Alpha Generative Adversarial Networks (Alpha-GANs) to train the generative model, which leverages the alpha divergence in information geometry [15]. We note that there is another so-called Alpha-GAN in [16]. However, there is no direct connection between these two models. Alpha-GAN in [16] is an application of GANs in natural image matting, while our proposed Alpha-GAN provides better objective function for the training scheme of GANs. Previous work has addressed the advances of alpha divergence and generalized it to many domains [17,18]. The alpha divergence can be seen as a generalization of multiple divergence functions, including Kullback–Leibler divergence [19], reverse KL divergence, Pearson χ^2 divergence, and Hellinger distance. Each one corresponds to a unique value of alpha in the alpha-divergence family. We assume that a real-world data distribution is denoted by

p_{data}, the goal of generator network G is to recover p_{real} through its generated distribution p_{fake} such that p_{fake} is as close to p_{real} as possible. However, it is always a tricky problem to keep the balance between G and D in existing approaches. The key contribution of our method is to propose a new value function that generalizes the alpha divergence to a tractable optimization problem in the generative model. Our new formulation involves two-order hyper-parameters for $D(\boldsymbol{x}_{\text{real}})$ and $D(\boldsymbol{x}_{\text{fake}})$ respectively which control the trade-off between p_{real} and p_{fake} in the training process. Moreover, we provide theoretical analysis to suggest effective guidance to select these hyper-parameters to strike a balance between the training stability and the desired quality of generated images.

The main contributions of our work can be summarized as follows:

(1) We derive a new objective function for GANs inspired by the alpha divergence. Our new formulation preserves the order parameters of alpha divergence which can be further manipulated in the training progress. We note that f-GAN gives another generalization form of alpha divergence. Compared with the derivation in f-GAN, our Alpha-GAN has a more tractable formulation for optimization.
(2) The proposed adversarial loss function of Alpha-GAN has a similar formulation to Wasserstein-GAN. It introduces two hyper-parameters to strike a balance between G and D, and it can converge stably without any 1-Lipschitz constraints. Thus, Alpha-GAN can be regarded as an upgrade of WGAN, and the experimental results show the advanced performance of our model.
(3) Through our new value function, we dig out a trade-off between training stability and quality of generated images in GANs. The two properties can be directly controlled by adjusting the hyper-parameters in the objective.

The rest of this paper is organized as follows. Section 2 briefly reviews the background of alpha divergence and some state-of-the-art architectures of GANs. More details about our proposed Alpha-GAN are formally stated in Section 3. In Section 4, experimental results are shown. Finally, we conclude our work in Section 5.

2. Background and Related Work

In this section, we introduce necessary background information regarding the alpha-divergence family and entropy and explain its relationship with novel generative models. Then, we go over some state-of-the-art GANs in the literature.

2.1. Entropy and Alpha Divergence

Before introducing the alpha divergence, we first review the concept of information entropy. The information entropy is proposed by Shannon, which is an important definition in information theory. Give a random variable X and its probability density $p(x)$, the entropy can be defined as:

$$H(X) = \int_x p(x)I(x) = -\int_x p(x)\log p(x)dx \tag{1}$$

It can be seen that if $p(x)$ gets closer to uniform distribution, the corresponding entropy will be greater. Although information entropy has few direct applications in machine learning, the cross-entropy is widely used in machine learning which is a derived from basic entropy:

$$H_{\text{CE}}(p,q) = \mathbb{E}_p[-\log q] = -\int_x p(x)\log q(x)dx \tag{2}$$

Cross-entropy is used to evaluate the difference between two distributions.

Kullback–Leibler (KL) divergence is another method to measure the disparity of distributions, which is also called relative entropy. KL divergence is generalized as the value function of original GANs. Given two probability densities p and q of random variable X, the KL divergence can be defined as:

$$D_{\mathrm{KL}}(p\|q) = -\int_x p(x)\log\frac{q(x)}{p(x)}dx = \int_x p(x)\log\frac{p(x)}{q(x)}dx \tag{3}$$

We can find that there is a relationship between entropy, cross-entropy and KL divergence:

$$H_{\mathrm{CE}}(p,q) = H(p) + D_{\mathrm{KL}}(p\|q) \tag{4}$$

Divergence function is a critical part of the overall framework of the GANs, since it is used to measure the difference between two data distributions p_{real} and p_{fake}. The regular GANs use Kullback–Leibler divergence as the critic measurement, which proves not to be the optimal choice in previous studies. In this work, we use the alpha divergence, and derive a new objective for GANs in Equation (13). We first give a brief review of alpha divergence upon which our Alpha-GAN model is based. Here, we mainly introduce two kinds of alpha divergence: Amari-alpha divergence [15] and Rényi-alpha divergence [20]. Considering two probability densities p and q of random variable $\boldsymbol{\theta}$, these two forms of alpha divergence can be defined on $\{\alpha : \alpha \in \mathbb{R} \setminus \{0,1\}\}$ as follows:

- Amari-alpha divergence:

$$D_A\left[p\|q\right] = \frac{1}{\alpha(\alpha-1)}\left(\int p(\boldsymbol{\theta})^{\alpha}q(\boldsymbol{\theta})^{1-\alpha}d\theta - 1\right). \tag{5}$$

- Rényi-alpha divergence:

$$D_R\left[p\|q\right] = \frac{1}{\alpha-1}\log\left(\int p(\boldsymbol{\theta})^{\alpha}q(\boldsymbol{\theta})^{1-\alpha}d\theta\right). \tag{6}$$

These divergences are related to the Chernoff α-coefficient $c_\alpha(p:q) = \int p(\boldsymbol{\theta})^{\alpha}q(\boldsymbol{\theta})^{1-\alpha}d\theta$ [21]. Please note that when $\alpha \to 0$, the Kullback–Leibler divergence can be recovered from both Amari and Rényi divergence while $\alpha \to 1$ leads to the reverse Kullback–Leibler divergence [22,23]. We present some other special cases of the Amari-alpha-divergence family in Table 1. It can be regarded as the final criterion of our Alpha-GAN with some simple manipulations. We also include some useful properties of Amari-alpha divergence [23] in the following:

Theorem 1 (Convexity). *Given two distributions p and q, the alpha divergence $D_A\left[p\|q\right]$ is a convex function with respect to both p and q. So for any $0 \geq \lambda \geq 1$, we have*

$$D_A\left[\lambda p_1 + (1-\lambda)p_2\|\lambda q_1 + (1-\lambda)q_2\right] \leq \lambda D_A\left[p_1\|q_1\right] + (1-\lambda)D_A\left[p_2\|q_2\right].$$

Theorem 2 (Strict Positivity). *The alpha divergence is a strictly positive function $D_A\left[p\|q\right] \geq 0$, and it has a unique minimum $D_A\left[p\|q\right] = 0$ if and only if $p = q$.*

Theorem 3 (Duality). $D_A^{(\alpha)}\left[p\|q\right] = D_A^{(1-\alpha)}\left[q\|p\right]$.

Table 1. Special cases in Amari-α divergence family.

α	Form	Divergence
$\alpha \to -1$	$\frac{1}{2}\int \frac{(q(\boldsymbol{\theta})-p(\boldsymbol{\theta}))^2}{p(\boldsymbol{\theta})} d\theta$	Reverse χ^2 divergence
$\alpha \to 0$	$\int q(\boldsymbol{\theta}) \log\left(\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta})}\right) d\theta$	Kullback–Leibler divergence
$\alpha \to 1$	$\int p(\boldsymbol{\theta}) \log\left(\frac{p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})}\right) d\theta$	Reverse KL divergence
$\alpha \to \frac{1}{2}$	$2\int \left(\sqrt{p(\boldsymbol{\theta})} - \sqrt{q(\boldsymbol{\theta})}\right)^2 d\boldsymbol{\theta}$	Hellinger divergence
$\alpha \to 2$	$\frac{1}{2}\int \frac{(p(\boldsymbol{\theta})-q(\boldsymbol{\theta}))^2}{q(\boldsymbol{\theta})} d\boldsymbol{\theta}$	Pearson χ^2 divergence

We will show how to adopt Amari-alpha divergence in our proposed GAN objective in Section 3. An effective guidance to select appropriate hyper-parameters for the rich alpha-divergence family is also provided.

2.2. Generative Adversarial Networks

In recent years, generative adversarial networks (GANs) have been one of the most attractive architectures in machine-learning systems. Since it was proposed by Goodfellow et al. in 2014 [5], tremendous variants of GAN have been produced by researchers. Most of them preserve the initial framework of vanilla GAN, which consists of two neural networks: a generator G and a discriminator D. G and D will adversarially learn from each other during the training phase. Figure 1 illustrates the schematic diagram of vanilla GANs, where G generates a fake image from a random latent code $\boldsymbol{z} \sim p_z$ and D learns to distinguish between real and fake samples. The key idea of GANs is usually defined as a game-play problem with a min–max objective. Researchers aim to obtain the optimal generator, which can generate high-resolution vivid images similar to natural images by fine-tuning the hyper-parameters properly. Next, we briefly show some popular objectives used to train a generative model.

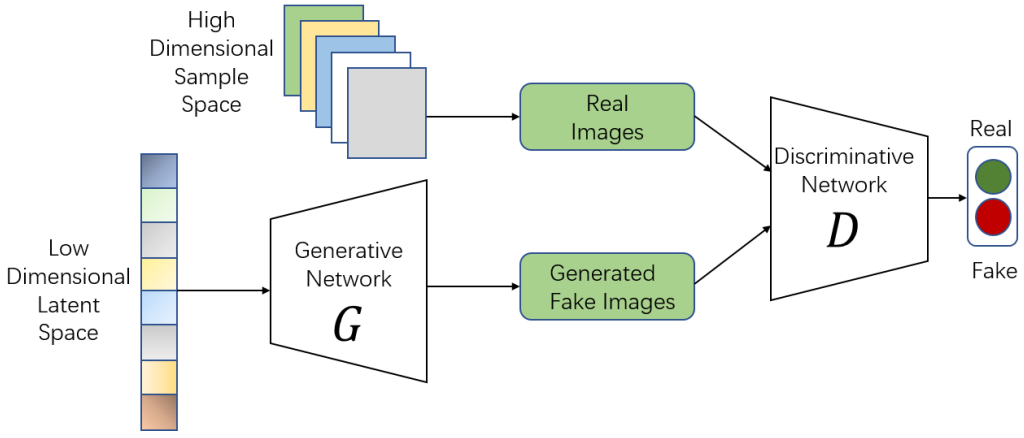


Figure 1. The architecture of vanilla GANs.

2.2.1. Vanilla GAN

The original GAN proposed in [5] can be defined as a contest between two networks G and D. The min–max objective is formally defined as follows:

$$\min_G \max_D \mathbb{E}_{\boldsymbol{x}\sim p_{\text{real}}} [\log(D(\boldsymbol{x}))] + \mathbb{E}_{\boldsymbol{x}\sim p_{\text{G}}} [\log(1 - D(\boldsymbol{x}))], \tag{7}$$

where $\boldsymbol{x}$ stands for the input image, p_{real} and p_{fake} represent the distribution of real-world and the generated data respectively. This objective follows the formulation of binary cross-entropy loss. The outputs of discriminator $D(\cdot)$ are confined within $[0, 1]$ through a sigmoid activation unit. The final critic value function of vanilla GANs can be formulated as the Jensen–Shannon divergence between p_{real} and p_{fake}:

$$\begin{aligned} \mathcal{C}(G) &= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{real}}}\left[\log \frac{p_{\text{real}}(\boldsymbol{x})}{p_{\text{real}}(\boldsymbol{x}) + p_{\text{fake}}(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x}\sim p_{\text{fake}}}\left[\log \frac{p_{\text{fake}}(\boldsymbol{x})}{p_{\text{real}}(\boldsymbol{x}) + p_{\text{fake}}(\boldsymbol{x})}\right] \\ &= 2 \cdot JS(p_{\text{real}} \| p_{\text{fake}}) - \log 4, \end{aligned} \tag{8}$$

The above min–max optimization problem is a popular mechanism in deep generative models. However, this model suffers from the unbalanced training problem of two neural networks.

2.2.2. LSGAN

One of the GANs' variants is LSGAN [9]. Compared with vanilla GANs, LSGANs substitute the binary cross-entropy loss with a least square loss, which has better properties for optimization and is less likely to saturate. LSGAN adopts the Pearson χ^2 divergence as the decision criterion. It is computed as follows:

$$\begin{aligned} \min_D \mathcal{V}_{\text{LSGAN}}(D) &= \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{real}}}\left[(D(\boldsymbol{x}) - 1)^2\right] + \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_z}\left[(D(G(\boldsymbol{z})))^2\right], \\ \min_G \mathcal{V}_{\text{LSGAN}}(G) &= \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_z}\left[(D(G(\boldsymbol{z})) - 1)^2\right], \end{aligned} \tag{9}$$

where $\boldsymbol{z} \sim p_z$ is the input latent noise of generator. LSGANs adopt the Pearson χ^2 divergence as the decision criterion:

$$\begin{aligned} \mathcal{C}(G) &= \frac{1}{2}\int_{\mathcal{X}} \frac{(2p_{\text{fake}}(\boldsymbol{x}) - (p_{\text{real}}(\boldsymbol{x}) + p_{\text{fake}}(\boldsymbol{x}))^2}{p_{\text{real}}(\boldsymbol{x}) + p_{\text{fake}}(\boldsymbol{x})} dx \\ &= \frac{1}{2}\chi^2_{\text{Pearson}}(p_{\text{real}} + p_{\text{fake}} \| 2p_{\text{fake}}), \end{aligned} \tag{10}$$

Nowozin et al. proposed a new mechanism f-GAN to give an elegant generalization of GAN and extended the value function to arbitrary f-divergence including χ^2 divergence in [10]. However, there still exists model collapsing for LSGANs and f-GAN.

2.2.3. Wasserstein-GAN

To further enhance the stability of GANs, Arjovsky et al. applied the Earth Mover (also called Wasserstein-1) distance, which is used to measure optimal transport cost between two distributions [12]. The Wasserstein distance is defined as:

$$W(\mathbb{P}_{\text{r}}, \mathbb{P}_{\text{g}}) = \inf_{\gamma \in \Pi(\mathbb{P}_{\text{r}}, \mathbb{P}_{\text{g}})} \mathbb{E}_{(x,y)\sim\gamma}\left[\|x - y\|\right]. \tag{11}$$

where $\Pi(\mathbb{P}_{\text{r}}, \mathbb{P}_{\text{g}})$ represents all joint distributions of $\mathbb{P}_{\text{r}}$ and $\mathbb{P}_{\text{g}}$. Wasserstein GANs also employ the Kantorovich–Rubinstein duality of Wasserstein-1 distance to construct the value function:

$$\min_G \max_D \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_{\text{r}}}\left[D(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_{\text{G}}}\left[D(\boldsymbol{x})\right]. \tag{12}$$

where D should be a 1-Lipschitz function and the weight parameters are clipped within the numerical interval $[-c, c]$. Gulrajani et al. proposed WGAN with gradient penalty (WGAN-GP) [13]. WGAN-GP introduces a soft penalty for the violation of 1-Lipschitz constraint, which guarantees high-quality image generation at the cost of the increasing computation complexity. SN-GANs [14] create a novel weight normalization technique which is called Spectral Normalization to stabilize the training process. Although these approaches effectively improve the training stability, they provide less flexibility to strike a balance between the training stability and the desired quality of generated images.

3. Proposed Method

We introduce our Alpha-GAN, a novel architecture of generative model based upon the minimization of alpha divergence [15]. The exact formulation of Alpha-GAN is defined in Equation (13) and we will show the relationship between Alpha-GAN and alpha divergence in Section 3.2.

3.1. Alpha-GAN Formulation

Inspired by the alpha divergence, we propose our new framework: the Alpha-GAN. In contrast to original GANs, Alpha-GAN removes the sigmoid output layer in discriminator network and substitutes the binary cross-entropy loss with our power function formulation. The proposed method further introduces two more hyper-parameters compared to WGAN. Specifically, Alpha-GAN model solves the following optimization problem:

$$\min_G \max_D \mathcal{V}_{\text{Alpha-GAN}}(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{real}}(\boldsymbol{x})} \left[|D(\boldsymbol{x})|^a \right] - \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}} \left[|D(G(\boldsymbol{z}))|^b \right]. \tag{13}$$

Please note that a, b are two-order indices for $D(\boldsymbol{x})$ and $D(G(\boldsymbol{z}))$ respectively. They are hyper-parameters introduced to balance the emphasis on $D(\boldsymbol{x})$ and $D(G(\boldsymbol{z}))$ during training process. To enhance the convergence stability, our proposed method only considers $a, b > 0$ in order to avoid the case that a term like $\frac{1}{D^a}$ appears in the loss function when $a \leq 0$ or $b \leq 0$. When the discriminator's output is smaller than 1, the loss value would be extremely large and accordingly the model would become less stable and hard to converge in training phase. Another update is that we take the absolute value of the discriminator output. Otherwise, the output would produce a trivial solution when $a < 1$ or $b < 1$. It seems like the objective function of Alpha-GAN is not immediately related to the formulation of alpha divergence in Equation (5). We will give the detailed theoretical analysis of how to derive Alpha-GAN from alpha divergence in Section 3.2. The training scheme of our Alpha-GAN is shown in Algorithm 1.

In [10], f-GAN also provides a value function related to alpha divergence. The authors generalize f-divergence to GAN objectives via a variational lower bound. The f-GAN objective with respect to alpha divergence can be defined as:

$$\begin{aligned} \mathcal{V}_{f-\text{GAN}} &= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{real}}}[g_f(V(\boldsymbol{x}))] + \mathbb{E}_{\boldsymbol{x} \sim p_{\text{fake}}}[-f^*(g_f(V(\boldsymbol{x})))], \\ f^*(t) &= \frac{1}{\alpha}(t(\alpha - 1) + 1)^{\frac{\alpha}{\alpha - 1}} - \frac{1}{\alpha} \end{aligned} \tag{14}$$

where $V(\boldsymbol{x})$ denotes the output of last layer of discriminator network and g_f is the output activation. For different values of α in alpha divergence, the activation g_f also has different formulations:

$$\begin{aligned} g_f(v) = \frac{1}{1 - \alpha} - \log(1 + \exp(-v)), \text{ for } \alpha < 1, \alpha \neq 0 \\ g_f(v) = v, \text{ for } \alpha > 1 \end{aligned} \tag{15}$$

The above objective has complex formulations and constraints, which makes it inconvenient for optimization in deep generative models. In addition, the severe model collapsing problem remains unsolved. In our proposed method, a simplified objective function is given with a similar induction process as the vanilla GANs, which has a more elegant form to balance output between stability and quality. A detailed analysis of the derivation will be shown in the next section.

Algorithm 1 Training scheme of Alpha-GAN.

Input: Batch size m, target distribution p_{real}, latent noise distribution p_z, input noise z, Adam optimizer with $\alpha, \beta_1 = 0.5, \beta_2 = 0.999$, hyper-parameters a, b, discriminator network D_ϕ and generator network G_θ, absolute function $\text{abs}(\cdot)$.

Output: Optimal generator G_θ.

1: **while** Training scheme of Alpha-GAN **do**
2: Sample $x_i^r \sim p_{\text{real}}, i = 1, \cdots, m$.
3: Sample $z_i \sim p_z, i = 1, \cdots, m$.
4: $x_i^g \leftarrow G_\theta(z_i), i = 1, \cdots, m$
5: $\phi \leftarrow \text{Adam}(-\frac{1}{m}\sum_{i=1}^{m} \nabla_\phi[\text{abs}(D_\phi(x_i^r))^a])$
6: $\phi \leftarrow \text{Adam}(\frac{1}{m}\sum_{i=1}^{m} \nabla_\phi[\text{abs}(D_\phi(x_i^g))^b])$
7: $\theta \leftarrow \text{Adam}(\frac{1}{m}\sum_{i=1}^{m} \nabla_\theta[\text{abs}(D(x_i^g))^b])$
8: **end while**
9: **return** G_θ

3.2. Theoretical Analysis

The original GAN model from Ian Goodfellow et al. proposed to minimize the Jensen–Shannon divergence:

$$C(G) = 2 \cdot JS\left(p_{\text{real}} \| p_{\text{fake}}\right) - \log 4. \tag{16}$$

The JS divergence can be written as the summation of KL divergence. Therefore, the final criterion of original GAN is the KL distance between the distributions of the ground-truth image and the generated one. However, there are many research results ([12]) showing that KL divergence is not a good objective for optimization. The alpha divergence employed in our approach can be seen as a generalization of KL divergence and we have already presented some basic properties of the alpha divergence in Section 2.1.

Next, we show how Alpha-GAN is related to the alpha divergence mentioned in Equation (5). We first give the proof of optimal discriminator D^* for arbitrary generator G.

Theorem 4. *For any fixed generator G and $a < b$, we prove the optimal discriminator D^* as:*

$$D^*(\boldsymbol{x}) = \left(\frac{b \cdot p_{fake}(\boldsymbol{x})}{a \cdot p_{real}(\boldsymbol{x})}\right)^{\frac{1}{a-b}}. \tag{17}$$

Proof. To prove the optimal D^* defined in Equation (17), we show that the objective function for discriminator D is to maximize the following Equation:

$$\begin{aligned}\mathcal{V}(D,G) &= \int_{\boldsymbol{x}} p_{\text{real}}(\boldsymbol{x})|D(\boldsymbol{x})|^a dx - \int_{\boldsymbol{z}} p_{\boldsymbol{z}}|D(G(\boldsymbol{z}))|^b dz \\ &= \int_{\boldsymbol{x}} p_{\text{real}}(\boldsymbol{x})|D(\boldsymbol{x})|^a - p_{\text{fake}}(\boldsymbol{x})|D(\boldsymbol{x})|^b dx. \end{aligned} \tag{18}$$

In Section 3.1, we already stated that we only consider $a, b > 0$ and we keep this setting in the proof. Then for $a < b$, the upper function is concave in $[0, \infty)$. We can take a derivative of it with respect to D and the optimal D^* in Equation (17) will be obtained. Since the optimal solution only lies within $[0, \infty)$, we take the absolute value of critic output. □

After that, we substitute the optimal $D^*(\boldsymbol{x})$ into the initial objective function as defined in Equation (13). We can reformulate it as follows:

$$\begin{aligned}\mathcal{C}(G) &= \max_D V(D,G) \\ &= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{real}}}\left[D_G^*(\boldsymbol{x})^a\right] - \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}\left[D_G^*(G(\boldsymbol{z}))^b\right] \\ &= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{real}}}\left[D_G^*(\boldsymbol{x})^a\right] - \mathbb{E}_{\boldsymbol{x}\sim p_{\text{fake}}}\left[D_G^*(\boldsymbol{x})^b\right] \\ &= \int_{\boldsymbol{x}} p_{\text{real}}(\boldsymbol{x})\left(\frac{b\cdot p_{\text{fake}}(\boldsymbol{x})}{a\cdot p_{\text{real}}(\boldsymbol{x})}\right)^{\frac{a}{a-b}} dx - \int_{\boldsymbol{x}} p_{\text{fake}}(\boldsymbol{x})\left(\frac{b\cdot p_{\text{fake}}(\boldsymbol{x})}{a\cdot p_{\text{real}}(\boldsymbol{x})}\right)^{\frac{b}{a-b}} dx \\ &= \int_{\boldsymbol{x}} \left(\frac{b}{a}\right)^{\frac{a}{a-b}} p_{\text{real}}(\boldsymbol{x})^{\frac{b}{b-a}} p_{\text{fake}}(\boldsymbol{x})^{\frac{a}{a-b}} dx - \int_{\boldsymbol{x}} \left(\frac{b}{a}\right)^{\frac{b}{a-b}} p_{\text{real}}(\boldsymbol{x})^{\frac{b}{b-a}} p_{\text{fake}}(\boldsymbol{x})^{\frac{a}{a-b}} dx. \end{aligned} \tag{19}$$

If we denote $\alpha = \frac{b}{b-a}$, $1-\alpha = \frac{a}{a-b}$, and set $c = \left(\frac{b}{a}\right)^{\frac{a}{a-b}} - \left(\frac{b}{a}\right)^{\frac{b}{a-b}}$, we can obtain from above Equation:

$$\begin{aligned}\mathcal{C}(G) &= c\cdot \int_{\boldsymbol{x}} p_{\text{real}}(\boldsymbol{x})^{\alpha}\cdot p_{\text{fake}}(\boldsymbol{x})^{1-\alpha} dx \\ &= c\cdot(\alpha(\alpha-1)D_A\left[p_{\text{real}}\|p_{\text{fake}}\right]+1). \end{aligned} \tag{20}$$

The final training criterion of generator G shown above can be seen as a linear transformation of the alpha divergence. Hence, our Alpha-GAN aims to reduce the distance measured by alpha divergence. We can manipulate the order α of divergence function through adjusting the value of hyper-parameters a and b.

3.3. Selection of Hyper-Parameters

Our Alpha-GAN uses two hyper-parameters a and b to control the update rate of $D(\boldsymbol{x})$ and $D(G(\boldsymbol{z}))$. The derivation in Equations (19) and (20) already states that changing a, b is equivalent to adjusting the order of alpha divergence. The relationship between a and b represents the preferences of model on real images or fake images. In practice, it is flexible for users to balance the training stability and the desired quality of generated images according to their specific requirements. One key problem is how to select proper hyper-parameters to obtain the optimal model. Here we give some useful suggestions on parameters selection:

- $\frac{b}{2} \le a \le b$: To prove the optimal discriminator D^* in Theorem 4, the hyper-parameters have been set to $a < b$ to satisfy the optimal condition. In the experiments of Alpha-GAN, we find that the scope can be reduced to $\frac{b}{2} \le a \le b$. This will help us to determine ratio between two parameters in the

applications. Noting that $a = b$ could also lead to good generation results while it does not satisfy the optimal condition in Theorem 4. We interpret this phenomenon as that the Alpha-GAN has a similar formulation like WGAN when $a = b$. These can be written as:

$$\begin{aligned} \mathcal{V}_{\text{Alpha-GAN}}(D,G) &= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{real}}}\left[|D(\boldsymbol{x})|\right] - \mathbb{E}_{\boldsymbol{z}\sim p_z}\left[|D(G(\boldsymbol{z}))|\right], \\ \mathcal{V}_{\text{WGAN}}(D,G) &= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{real}}}\left[D(\boldsymbol{x})\right] - \mathbb{E}_{\boldsymbol{z}\sim p_z}\left[D(G(\boldsymbol{z}))\right] \end{aligned} \tag{21}$$

We believe that the setting of $a = b$ will have some similar convergence properties like WGAN.

- $a, b \geq 0.4$: For the training stability of Alpha-GAN model, we only consider $a, b > 0$ to avoid forms like $\frac{1}{D^a}$ as stated in Section 3.1. Otherwise, the loss will be extremely unstable when $D \ll 0$. In evaluation experiments, when we set $a, b < 0.4$, the model cannot converge successfully, and the generated images are very blurred. The small values of parameters mean that the gradients feedback will be multiplied by a small coefficient in back-propagation. It is hard for the generator and discriminator to learn useful information from image data in such settings. Thus, we recommend setting the parameters to $a, b \geq 0.4$.
- $a, b \leq 1$: This suggestion is also summarized from the experimental results, and may not always be valid. In the image generation experiments of Alpha-GAN, we find that the loss curves fluctuate largely when $a, b > 1$. However, the quality of generated images is not too bad. We believe the model will be difficult to converge well when it is faced with more complex problems, such as larger image datasets.

One way to select proper hyper-parameters is referring to the special cases of alpha divergence as shown in Table 1. For example, we observe that a good convergence result will be obtained when the parameters are set as $2a = b$, which corresponds to the Pearson χ^2 divergence in alpha-divergence family. We already denote $\alpha = \frac{b}{b-a}$ in Equation (20). Then for $2a = b$, we can get:

$$\begin{aligned} \mathcal{C}(G) &= c \cdot \int_{\boldsymbol{x}} p_{\text{real}}(\boldsymbol{x})^2 \cdot p_{\text{fake}}(\boldsymbol{x})^{-1} dx \\ &= c \cdot (D_{\chi^2}[p_{\text{real}} \| p_{\text{fake}}] + 1). \end{aligned} \tag{22}$$

And we also suggest $a, b \leq 1$ in the previous analysis. Here we further simplify the parameters as $a = \frac{1}{2}, b = 1$. Then, the adversarial loss of Alpha-GAN can be written as:

$$\mathbb{E}_{\boldsymbol{x}\sim p_{\text{real}}(\boldsymbol{x})}\left[D(\boldsymbol{x})^{\frac{1}{2}}\right] - \mathbb{E}_{\boldsymbol{z}\sim p_z}\left[D(G(\boldsymbol{z}))\right]. \tag{23}$$

Noting that we do not claim the values of $a = \frac{1}{2}, b = 1$ or Pearson χ^2 divergence are optimal for Alpha-GAN. It is one of our observations that such parameter settings can bring stable convergence performance in applications, thus we give a piece of reasonable advice to initialize a and b. In [9], the author employs Pearson χ^2 divergence to generalize the LSGAN. Compared with the adversarial loss of LSGAN in Equation (9), our model has a totally different formulation in Equation (23). Our Alpha-GAN is derived from a special case of alpha divergence, not directly from χ^2 divergence. We also evaluate the effects of different settings under diverse hyper-parameters and the effectiveness of our mechanism will be shown in Section 4.

4. Experiments

In this section, we conduct extensive experiments to evaluate the proposed method. We compare Alpha-GAN with some baseline models to show the competitive results of our approach. The algorithms

are all implemented with PyTorch [24] in this section. The source code can be found in https://github.com/cailk/AlphaGAN.

4.1. Datasets

There are three datasets involved in our paper, including the handwritten digital dataset MNIST [25], and two real-world image datasets SVHN [26] and CelebA [27].

- **MNIST**: MNIST is a widely used database of handwritten digits, which contains a training set of 60,000 images and a test set of 10,000 images. There are 10 labels from '0' to '9' for dataset and all digits are normalized to 28×28. We use MNIST to evaluate the trade-off effect between two hyper-parameters in the value function.
- **SVHN**: SVHN is a real-world color image dataset obtained from house numbers in Google Street View images. Its training set contains 73,257 32×32 digital images. SVHN dataset is similar to MNIST, but comes from a harder problem since all digits are sampled in natural scene.
- **CelebA**: Last dataset used in this paper is CelebA, which is a large-scale face attribute dataset with more than 200,000 images. Samples are all 64×64 color celebrity images. CelebA is an important dataset in the scenario of image generation since it only contains information of face attribute and is easy to learn for GANs.

4.2. Model Architectures and Implementation Details

The architecture of our generator and discriminator is designed based on the InfoGAN [28]. The generator network is fed by a latent variable $z \sim \mathcal{N}_{128}(0, I)$. It contains a fully connection layer that upscales the input tensor to size $512 \times 2 \times 2$, four transposed convolution layers (kernel size = 4×4, stride = 2, padding = 1) and the *tanh* activation layer. The discriminator network consists of 4 convolution layers that extract features from 32×32 inputs. ReLU activation function is used after each layer in generator network and Leaky-ReLU for discriminator network. Batch normalization is employed in each layer of both networks.

For our Alpha-GAN in Equation (13), we remove the last activation layer of discriminator like WGAN [12], and we apply an *abs* function to the critic output. We employ Adam optimizer [29] with learning rate of 0.0002 and decay rates of $\beta_1 = 0.5, \beta_2 = 0.999$ to train the generator network. In addition, the discriminator network is also trained using an Adam optimizer with learning rate of 0.0002. The total number of epochs is 50 for MNIST, SVHN, and 30 for CelebA. All experiments are conducted in a machine with one NVIDIA GTX 1080 GPU.

4.3. Evaluation Metrics

Measuring the quality of generated images is usually a more tricky and challenging problem than simply generating vivid images. It is almost impossible to directly establish an objective on the space of natural and generated images. To measure the quality of generated image samples, we employ the Fréchet Inception Distance (FID) proposed in [30], which is a commonly used metric for GANs. The FID is supposed to be more advanced than Inception Score (IS) [31] which is another metric to evaluate deep generative models. Suppose two multivariate Gaussians $X_{\text{real}} \sim \mathcal{N}(\mu_1, \Sigma_1)$ and $X_{\text{fake}} \sim \mathcal{N}(\mu_2, \Sigma_2)$ are the 2048-dimensional activation outputs of the Inception-v3 [32] pool_3 layer for real and generated samples respectively. The FID can be defined as follows:

$$\text{FID} = \|\mu_1 - \mu_2\|^2 + \text{Tr}(\Sigma_1 + \Sigma_2 - 2(\Sigma_1\Sigma_2)^{\frac{1}{2}}). \tag{24}$$

FID compares the statistics of fake images to real ones, instead of only evaluating the generated samples. Thus, FID will give a more reliable standard to measure the effect of GANs. For FID score, low is better, meaning real and generated samples are more similar, measured by the distance between their activation distributions.

4.4. The Influence of Hyper-Parameters

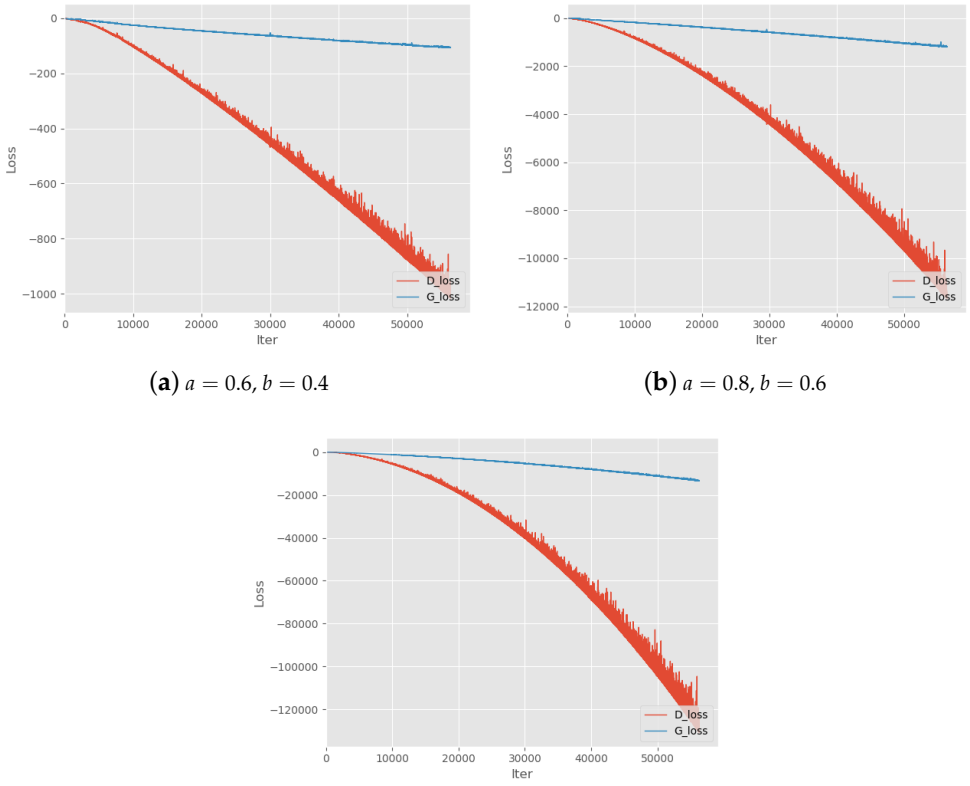
In our Alpha-GAN, we introduce two hyper-parameters a, b to the objective function, and we interpret them as how favorite the model want to learn from real and fake data distribution. To verify the influence of changing values of a and b in Equation (13), we conducted extensive experiments on MNIST dataset to demonstrate the trade-off between p_{real} and p_{fake}.

First, we test various parameter settings of Alpha-GAN to evaluate the basic convergence performance on different values of parameters a and b. The results can be found in Table 2. a and b are the two parameters in adversarial loss of Alpha-GAN. The symbol '$\surd$' denotes the models with corresponding parameter setting can converge normally and generate high-quality digital images. '–' means the quality of generated by corresponding models is slightly poor. In addition, '$\times$' means the model cannot converge and generated samples are blurred. We also find that when $a < 0.4$ or $b < 0.4$, the model will not converge. Thus, we recommend keeping the parameters greater than or equal to 0.4. In the theoretical analysis of Alpha-GAN, we suppose $a < b$ to ensure the concavity of objective function. We can see that almost all settings on $a > b$ will lead to poor model performance except for $(0.6, 0.4), (0.8, 0.6), (1.0, 0.8)$. Figure 2 shows the training loss curves of these settings, which means that models do not converge during the training progress. This also suggests the quality of generated results may get worse when handling the larger datasets. It is worth noting that all generative models with satisfactory effects have a parameter pair within ratio $\frac{b}{2} \leq a \leq b$ as we suggest in Section 3.3. Another advice we give in the analysis of parameter selection is $a, b \leq 1$, and there exist some models without such constraint which can still converge. However, the loss curves of these models look not good as shown in Figure 3.

Table 2. Convergence ability on different value selections of parameters.

Parameters	$b = 0.2$	$b = 0.4$	$b = 0.6$	$b = 0.8$	$b = 1.0$	$b = 1.5$	$b = 2.0$
$a = 0.2$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$
$a = 0.4$	$\times$	$\surd$	$\surd$	$\surd$	–	–	$\times$
$a = 0.6$	$\times$	–	$\surd$	$\surd$	$\surd$	$\times$	$\times$
$a = 0.8$	$\times$	$\times$	–	$\surd$	$\surd$	$\surd$	$\times$
$a = 1.0$	$\times$	$\times$	$\times$	–	$\surd$	$\surd$	$\times$
$a = 1.5$	$\times$	$\times$	$\times$	$\times$	$\times$	$\surd$	$\surd$
$a = 2.0$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\surd$

To show the effect of hyper-parameters on Alpha-GAN more clearly, some of the generated results are illustrated in Figure 4. The index a is gradually set as $0.3, 0.4, 0.5$ and 0.6 while b is fixed at 1. One intuitive phenomenon is that the quality of generated samples is becoming better when we increase the value of a. The image samples are very fuzzy and difficult to distinguish when $a = 0.3$, and $a = 0.4$ performs better. When $a = 0.5, 0.6$, the model can generate high-quality handwritten digits. As we interpreted before, a, b represent the restraint level of $D(\boldsymbol{x}_{\text{real}})$ and $D(\boldsymbol{x}_{\text{fake}})$ respectively in Alpha-GAN. Decreasing the value of a means reducing the gradient feedback of critic output on real data. In that case, the discriminator will learn less information from the ground-truth data, the generated results will lack diversity and become unreal.



(**a**) $a = 0.6, b = 0.4$

(**b**) $a = 0.8, b = 0.6$

(**c**) $a = 1.0, b = 0.8$

Figure 2. Training Loss curves with parameters settings $a > b$. a, b denote the two hyper-parameters in the objective function.

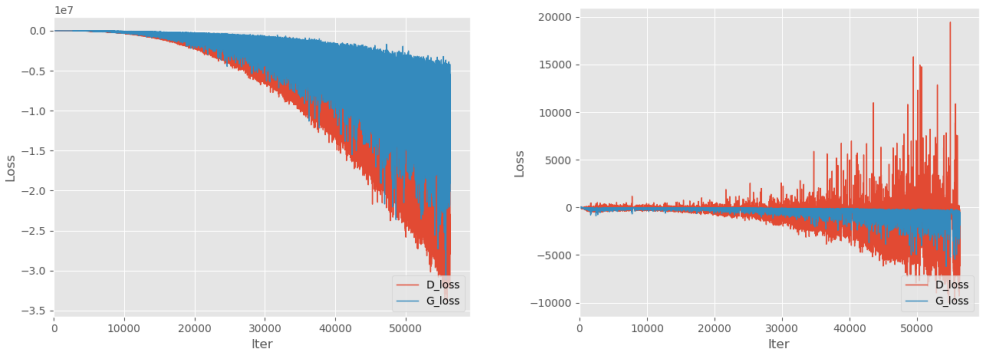
One subsequent question arises naturally that "Can the value of a be arbitrarily large to generate decent and recognizable samples?". In our experiments, we explore another property of Alpha-GAN, which indicates that the loss curve becomes less stable when a or b increases. Especially when one of the parameters is bigger than 1, the output loss becomes extremely large and model is less possible to converge. For example, the final output loss of discriminator is beyond 1e10 in Figure 3c. Therefore, it is essential to strike a balance between the training stability and the desired quality of generated images.

4.5. Generation Results

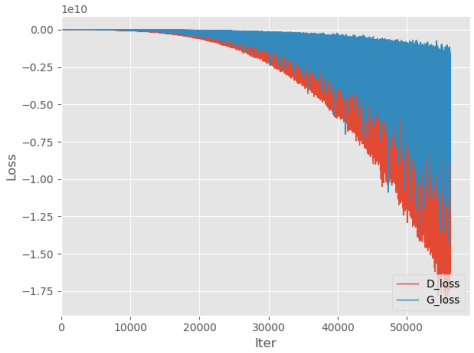
We further show the generated results on the real-world datasets SVHN and CelebA, and compare our Alpha-GAN model with some baseline approaches, including WGAN, and WGAN-GP. Models all run in the same network architecture with fine-tuning.

4.5.1. Comparison with Baseline Models

In this section, we conduct extensive experiments to compare our Alpha-GAN with some baseline generative models. The Fréchet Inception Distance is calculated for each generator trained on the CelebA dataset. As aforementioned, lower FID score means GANs can generate samples closer to real data. We randomly sample 10,000 images with each GAN model and calculate the corresponding FID scores on ground-truth dataset with over 200,000 images.

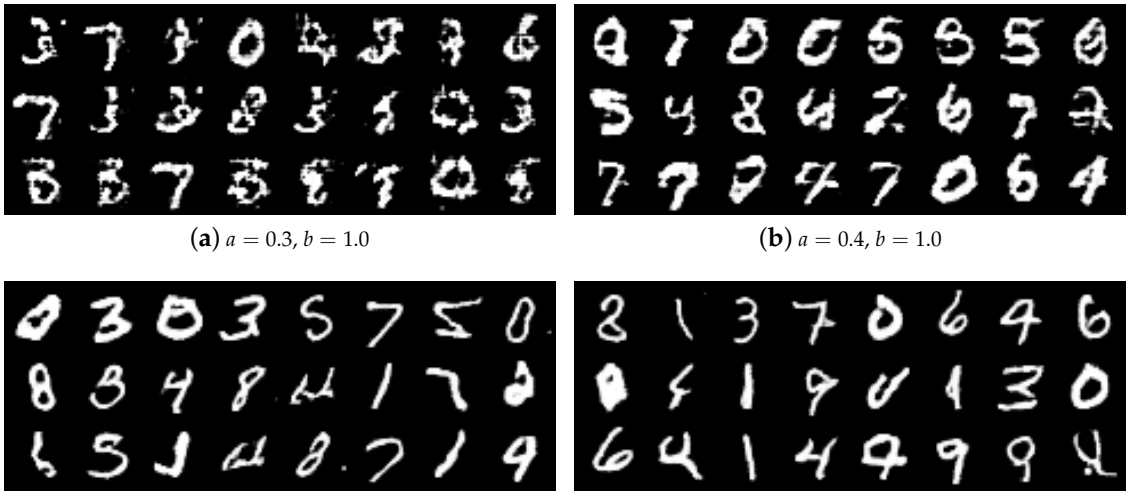


(a) $a = 1.5, b = 1.5$ **(b)** $a = 1.5, b = 2.0$



(c) $a = 2.0, b = 2.0$

Figure 3. Training Loss curves with parameters settings $a, b > 1$. a, b denote the two hyper-parameters in the objective function.



(a) $a = 0.3, b = 1.0$ **(b)** $a = 0.4, b = 1.0$

(c) $a = 0.5, b = 1.0$ **(d)** $a = 0.6, b = 1.0$

Figure 4. Generated samples on MNIST dataset with different parameters settings. a, b denote the two hyper-parameters in the objective function.

Figure 5 shows the generated results on CelebA dataset of our model and some competitors. Figure 5a illustrates samples generated by WGAN without weight clipping method and the results are not quite good. According to the theoretical analysis of WGAN, the weight clipping ensures the 1-Lipschitz continuous property of discriminator and convergence stability. This explains the poor quality of shown images. The samples of original WGAN is shown in Figure 5b, we observe that the results become better but still not good enough. In Figure 5c, the results of WGAN-GP have higher quality and are clear to be recognized. Similarly, our Alpha-GAN could generate competitive samples without any gradient penalty applied. Table 3 shows the Fréchet Inception Distance of our Alpha-GAN and some prominent GAN models on CelebA and SVHN dataset. Our proposed Alpha-GAN clearly outperforms WGAN and WGAN-GP.

(**a**) WGAN (without weight clipping)

(**b**) WGAN

(**c**) WGAN-GP

(**d**) Alpha-GAN

Figure 5. Generated samples of different models on CelebA dataset.

Table 3. Frechet Inception Distance of various models on CelebA and SVHN.

Model	CelebA	SVHN
WGAN [12]	181.11	37.02
WGAN-GP [13]	193.68	50.7
Alpha-GAN (ours)	176.37	23.26

4.5.2. Generated Results

We also evaluate our model on SVHN and CelebA with several different hyperparameter settings. The generated samples are shown in Figure 6. As can be seen, Figure 6a–c illustrate some sample images with $a = 0.4$, $a = 0.5$ and $a = 0.8$ respectively. As the value of a increases, the digits figures become clear and recognizable. Figure 6d–f show some generated results on CelebA with $a = 0.4$, $a = 0.5$ and $a = 0.6$ respectively. When the dataset becomes more complex and the network architectures go deeper, the increasing value of a can bring more instability on the results as we stated before.

(a) SVHN ($a = 0.4, b = 1.0$) **(b)** SVHN ($a = 0.5, b = 1.0$) **(c)** SVHN ($a = 0.8, b = 1.0$)

(d) CelebA ($a = 0.4, b = 1.0$) **(e)** CelebA ($a = 0.5, b = 1.0$) **(f)** CelebA ($a = 0.6, b = 1.0$)

Figure 6. Generated samples of AlphaGAN.

5. Conclusions

In this paper, we propose a novel value function for GAN framework using the alpha divergence which can be regarded as a generalization of the Kullback–Leibler divergence. To improve Wasserstein-GAN, our objective introduces two more hyper-parameters to keep a balance during the

training procedure. Moreover, we conduct a theoretical analysis for selecting appropriate hyper-parameters in order to control the information of p_{data} and p_g to maintain the training stability. Furthermore, we also find some trade-off between the training convergence and generation quality. Experimental results demonstrate that attempts to generate extremely high-quality images may bring instability to GANs. A novel mechanism for explicitly controlling the two properties is explored and outperforms previous works. For future works, we hope to extend Alpha-GAN to large-scale datasets such as CIFAR10 and ImageNet.

Author Contributions: L.C. proposed the main idea. L.C. and Y.C. performed the experiments to validate the results and wrote the manuscript. N.C. and W.C. reviewed and edited the manuscript. H.W. gave advice. L.C. and Y.C. contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1097–1105.
2. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
3. Zhang, J.; Zong, C. Deep neural networks in machine translation: An overview. *IEEE Intell. Syst.* **2015**, *30*, 16–25. [CrossRef]
4. Graves, A.; Mohamed, A.r.; Hinton, G. Speech recognition with deep recurrent neural networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 6645–6649.
5. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
6. Mathieu, M.; Couprie, C.; LeCun, Y. Deep multi-scale video prediction beyond mean square error. *arXiv* **2015**, arXiv:1511.05440.
7. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–27 October 2017; pp. 2223–2232.
8. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4681–4690.
9. 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, Venice, Italy, 22–27 October 2017; pp. 2794–2802.
10. Nowozin, S.; Cseke, B.; Tomioka, R. f-gan: Training generative neural samplers using variational divergence minimization. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 271–279.
11. Goodfellow, I. NIPS 2016 tutorial: Generative adversarial networks. *arXiv* **2016**, arXiv:1701.00160.
12. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 214–223.

13. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved training of wasserstein gans. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5767–5777.
14. Miyato, T.; Kataoka, T.; Koyama, M.; Yoshida, Y. Spectral normalization for generative adversarial networks. *arXiv* **2018**, arXiv:1802.05957.
15. Amari, S.i. *Differential-Geometrical Methods In Statistics*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
16. Lutz, S.; Amplianitis, K.; Smolic, A. Alphagan: Generative adversarial networks for natural image matting. *arXiv* **2018**, arXiv:1807.10088.
17. Li, Y.; Turner, R.E. Rényi divergence variational inference. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 1073–1081.
18. Djolonga, J.; Lucic, M.; Cuturi, M.; Bachem, O.; Bousquet, O.; Gelly, S. Evaluating Generative Models Using Divergence Frontiers. *arXiv* **2019**, arXiv:1905.10768.
19. Kullback, S.; Leibler, R.A. On Information and Sufficiency. *Ann. Math. Statist.* **1951**, *22*, 79–86. [CrossRef]
20. Rényi, A. On measures of entropy and information. In Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics. The Regents of the University of California, Berkeley, CA, USA, 20 June–30 July 1961.
21. Chernoff, H. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Ann. Math. Statist.* **1952**, *23*, 493–507. [CrossRef]
22. Van Erven, T.; Harremos, P. Rényi divergence and Kullback-Leibler divergence. *IEEE Trans. Inf. Theory* **2014**, *60*, 3797–3820. [CrossRef]
23. Cichocki, A.; Amari, S.i. Families of alpha-beta-and gamma-divergences: Flexible and robust measures of similarities. *Entropy* **2010**, *12*, 1532–1568. [CrossRef]
24. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic differentiation in pytorch. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
25. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
26. Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A.Y. Reading digits in natural images with unsupervised feature learning. In Proceedings of the Advances in Neural Information Processing Systems, Granada, Spain, 12–17 December 2011.
27. Liu, Z.; Luo, P.; Wang, X.; Tang, X. Deep Learning Face Attributes in the Wild. In Proceedings of the International Conference on Computer Vision (ICCV), Santiago, Chile, 13–16 December 2015.
28. Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 2172–2180.
29. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
30. 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 Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6626–6637.

31. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved techniques for training gans. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 2234–2242.
32. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).