

Accelerated WGAN update strategy with loss change rate balancing

Xu Ouyang

Ying Chen

Gady Agam

Illinois Institute of Technology

{xouyang3, ychen245}@hawk.iit.edu, agam@iit.edu

Abstract

Optimizing the discriminator in Generative Adversarial Networks (GANs) to completion in the inner training loop is computationally prohibitive, and on finite datasets would result in overfitting. To address this, a common update strategy is to alternate between k optimization steps for the discriminator D and one optimization step for the generator G . This strategy is repeated in various GAN algorithms where k is selected empirically. In this paper, we show that this update strategy is not optimal in terms of accuracy and convergence speed, and propose a new update strategy for networks with Wasserstein GAN (WGAN) group related loss functions (e.g. WGAN, WGAN-GP, Deblur GAN, and Super resolution GAN). The proposed update strategy is based on a loss change ratio comparison of G and D . We demonstrate that the proposed strategy improves both convergence speed and accuracy.

1. Introduction

GANs [8] provide an effective deep neural network framework that can capture data distribution. GANs are modeled as a min-max two-player game between a discriminator network $D_\psi(x)$ and a generator network $G_\theta(z)$. The optimization problem solved by GAN [21] is given by:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{data}} [f(D(x))] + \mathbb{E}_{z \sim p_{latent}} [f(-D(G(z)))] \quad (1)$$

where $G : Z \rightarrow X$ maps from the latent space Z to the input space X ; $D : X \rightarrow \mathbb{R}$ maps from the input space to a classification of the example as fake or real; and $f : \mathbb{R} \rightarrow \mathbb{R}$ is a concave function. In the remainder of this paper, we use the Wasserstein GAN [3] obtained when using $f(x) = x$.

GANs have been shown to perform well in various image generation applications such as: deblurring images [17], increasing the resolution of images [18], generating captions from images [5], and generating images from captions [23]. Training GANs may be difficult due to stability and convergence issues. To understand this consider the fact that

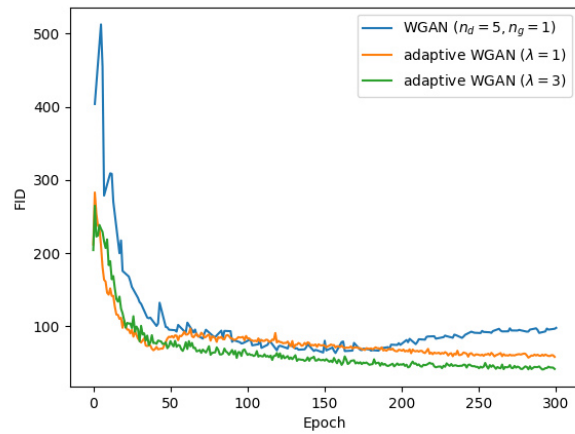


Figure 1. Comparison of the Fréchet Inception Distance (FID) for WGAN and the proposed adaptive WGAN with different coefficient λ on the CIFAR10 dataset. A lower FID means better performance. The parameters n_d and n_g show the fixed number of update steps in WGAN for the discriminator and generator respectively.

GANs minimize a probabilistic divergence between real and fake (generated by the generator) data distributions [22]. Arjovsky *et al.* [2] showed that this divergence may be discontinuous with respect to the parameters of the generator, and may have infinite values if the real data distribution and the fake data distribution do not match.

In order to solve a divergence continuous problem, WGAN [3] uses the Wasserstein-divergence by removing the sigmoid function in the last layer of the discriminator and so restricting the discriminator to Lipschitz continuous functions instead of the Jensen-Shannon divergence in the original GAN [8]. WGAN will always converge when the discriminator is trained until convergence. However, in practice, WGAN is trained with a fixed number (five) of discriminator update steps for each generator update step.

Even though WGAN is more stable than the original GAN, Mescheder *et al.* [19] proved that WGAN trained with simultaneous or alternating gradient descent steps with a fixed number of discriminator updates per generator up-

date and a fixed learning rate $\alpha > 0$ does not converge to the Nash equilibrium for a Dirac-GAN, where the Dirac-GAN [19] is a simple prototypical GAN.

The WGAN update strategy proposed in [3] suggests an empirical ratio of five update steps for the discriminator to one update step for the generator. As should be evident this empirical ratio should not hold in all cases. Further, this ratio need not be fixed throughout training. Maintaining an arbitrary fixed ratio is inefficient as it will inevitably lead to unnecessary update steps. In this paper, we address the selection of the number of training steps for the discriminator and generator and show how to adaptively change them during training so as to make the training converge faster to a more accurate solution. We focus in this work on WGAN related loss as it is more stable than the original GAN.

The strategy we propose for balancing the training of the generator and discriminator is based on the discriminator and generator loss change ratios (r_d and r_g , respectively). Instead of a fixed update strategy we decide whether to update the Generator or discriminator by comparing the weighted loss change ratios r_d and $\lambda \cdot r_g$ where the weight λ is a hyper-parameter assigning coefficient to r_g . The default value $\lambda = 1$ leads to higher convergence speed and better performance in nearly all the models we tested and is already superior to a fixed strategy. It is possible to further optimize this parameter for additional gains either using prior knowledge (*e.g.* giving preference to training the generator as in the original WGAN) or in empirical manner. Note, however, that the proposed approach accelerates convergence even without optimizing λ .

In addition to the acceleration aspect of the proposed update strategy, we studied its convergence properties. To do so we followed the methodology by Mescheder *et al.* [19]. Following this methodology, we demonstrate that the proposed strategy can reach a local minimum point for the Dirac-GAN problem, whereas the original update strategy cannot achieve it.

To further demonstrate the advantage of our strategy, we train the WGAN [3], WGAN-GP [9], Deblur-GAN [17], and SR-WGAN [18] using the proposed strategy using different image datasets. These represent a wide range of WGAN applications. Experimental results show that in general the proposed strategy converges faster while achieving in many cases better accuracy. An illustration is provided in Figure 1 where the proposed adaptive WGAN is compared with the traditional WGAN update strategy.

The main contribution of this paper is proposing an adaptive update strategy for WGAN instead of the traditional fixed update strategy in which the update rate is set empirically. This results in accelerated training and in many cases with higher performance. Following a common convergence analysis procedure, we show that the proposed strategy can reach local convergence for Dirac-GAN, unlike

the traditional fixed update strategy which cannot do so. Experimental results on several WGAN problems with several datasets show that the proposed adaptive update strategy results in faster convergence and higher performance.

2. Related work

The question of which training methods for GANs actually converge was investigated by Mescheder *et al.* [19] where they introduce the Dirac-GAN. The Dirac-GAN consists of a generator distribution $p_\theta = \delta_\theta$ and a linear discriminator $D_\psi(x) = \psi \cdot x$. In their paper they prove that a fixed point iteration $F(x)$ is locally convergent to x , when the absolute values of the eigenvalues of the Jacobian $F'(x)$ are all smaller than 1. They further prove that for the Dirac-GAN, with both simultaneous and alternative gradient descent updates, the absolute values of the eigenvalues of the Jacobian $F'(x)$ in GANs with unregularized gradient descent (which include the original GAN, WGAN and WGAN-GP) are all larger or equal to 1, thus showing that these types of GANs are not necessarily locally convergent for the Dirac-GAN. To address this convergence issue, Mescheder *et al.* [19] added gradient penalties to the GAN loss and proved that regularized GAN with these gradient penalties can reach local convergence. This solution does not apply to WGAN and WGAN-GP which remain not locally convergent problems. Note that the WGAN is generally more stable and easier to train compared with GAN and hence the need for the adaptive update scheme we propose in this paper.

Heusel *et al.* [11] attempt to address the convergence problem in a different way by altering the learning rate. In their approach they use a two time-scale update rule (TTUR) for training GANs with stochastic gradient descent using arbitrary GAN loss functions. Instead of empirically setting the same learning rate for both the generator and discriminator, TTUR uses different learning rates for them. This is done in order to address the problem of slow learning for regularized discriminators. They prove that training GANs with TTUR can converge to a stationary local Nash equilibrium under mild condition based on stochastic approximation theory. In their experiments on image generation, they show that WGAN-GP with TTUR gets better performance. Note however that empirically setting the learning rate is generally difficult and even more so when having to set two learning rates jointly. This makes applying this solution more difficult.

It is well understood that the complexity of the generator should be higher than that of the discriminator, a fact that makes GANs harder to train. Balancing the learning speed of the generator and discriminator is a fundamental problem. Unbalanced GANs [10] attempt to address this by pre-training the generator using variational autoencoder (VAE [15]), and using this pre-

trained generator to initialize the GAN weights during GAN training. An alternative solution is proposed in BEGAN [4], where the authors introduce an equilibrium hyper-parameter ($\mathbb{E}[f(-D(G(z)))]/\mathbb{E}[f(D(x))]$) to maintain the balance between the generator and discriminator. Training the two neural networks in this approach is time consuming and the equilibrium hyper-parameter is not suitable for all GAN training cases. For example it is not suitable when there is a content loss in the generator loss term L_G as L_G and the discriminator loss L_D are not on the same scale.

A similar issue to the unbalanced training of the generator and discriminator in GANs arises in imbalanced training of multiple task networks. The GradNorm [6] approach provides a solution to balancing multitask network training based on gradient magnitudes. In this approach, the authors multiply each of the single-task loss terms by weights, and automatically update those weights by computing a gradient normalization term. A relative inverse training rate ($L_{\text{current}}/L_{\text{initial}}$) for each task is used to compute this normalization term. This strategy depends on a common loss term minimization where individual task terms are weighted and so is not suitable for GANs where there is no shared layer as in multi-task networks.

3. Method

3.1. Adaptive update strategy

In this section, we present our proposed update strategy to automatically set the update rate of the generator and discriminator instead of using a fixed rate as is commonly done. In WGAN or any GAN based on the related WGAN loss, the Nash equilibrium is reached when the generator and discriminator loss terms stop changing. That is:

$$|L_g^c - L_g^p| = 0 \quad \& \quad |L_d^c - L_d^p| = 0 \quad (2)$$

where L_g^c, L_d^c represent the generator and discriminator loss in the current iteration respectively, and L_g^p, L_d^p represent the respective loss terms in the previous iteration. Since we play a min-max game in WGAN, it is crucial to balance the generator and discriminator losses. The loss terms L_g^c, L_d^c are given by:

$$\begin{aligned} L_g^c &= E_{p(z)}[f(D_\psi(G_\theta(z)))] \\ L_d^c &= E_{p_D(x)}[f(D_\psi(x))] + E_{p(z)}[f(-D_\psi(G_\theta(z)))] \end{aligned} \quad (3)$$

Comparing L_g^c, L_d^c directly to decide on an update policy is not possible because they are on different scales and so we define relative loss terms that can be compared. The relative loss terms are defined by computing the difference between the current and previous loss values and normalizing the difference by the loss magnitude. The relative change loss terms for the generator and discriminator are defined

by:

$$\begin{aligned} r_g &= |(L_g^c - L_g^p)/L_g^p| \\ r_d &= |(L_d^c - L_d^p)/L_d^p| \end{aligned} \quad (4)$$

To prioritize the update of one component over the other as commonly done in GANs, we use an coefficient λ . Thus, if $r_d > \lambda \cdot r_g$, we update the discriminator, or otherwise update the generator. A larger loss change ratio of one component means that this component is in greater need for update. The details of our proposed adaptive WGAN are provided in Algorithm 1.

Algorithm 1 Proposed adaptive update strategy

- **parameters:** learning rate (α); clipping parameter (c); loss coefficient (λ); batch size (m).
- **variables:** generator parameters (θ); discriminator parameters (ψ); generator loss change ratio (r_g); discriminator loss change ratio (r_d). The loss change ratios are initialized to 1.

```

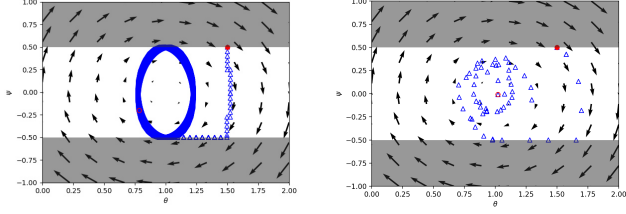
1: while  $\theta$  has not converged do
2:   Sample  $\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$  a batch from the real data
3:   Sample  $\{z^{(i)}\}_{i=1}^m \sim p(z)$  a batch of prior samples
4:   if  $r_d > \lambda \cdot r_g$  then           # update the discriminator
5:      $g_\psi \leftarrow \nabla_\psi [\frac{1}{m} \sum_{i=1}^m f_\psi(x^{(i)}) - \frac{1}{m} \sum_{i=1}^m f_\psi(g_\theta(z^{(i)}))]$ 
6:      $\psi \leftarrow \psi + \alpha \cdot \text{RMSProp}(\psi, g_\psi)$ 
7:      $\psi \leftarrow \text{clip}(\psi, -c, c)$ 
8:   else                             # update the generator
9:      $g_\theta \leftarrow -\nabla_\theta [\frac{1}{m} \sum_{i=1}^m f_\psi(g_\theta(z^{(i)}))]$ 
10:     $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$ 
11:   end if
12:   if first iteration then
13:      $L_g^p, L_d^p = L_g, L_d$ 
14:   end if
15:    $L_g^c, L_d^c = L_g, L_d$ 
16:    $r_g, r_d = |(L_g^c - L_g^p)/L_g^p|, |(L_d^c - L_d^p)/L_d^p|$ 
17:    $L_g^p, L_d^p = L_g, L_d$ 
18: end while
```

3.2. Convergence evaluation

In this section, we demonstrate that with our proposed update strategy, WGAN can reach local convergence for Dirac-GAN, whereas it cannot do so with a fixed update strategy. The GAN objective function is given by:

$$L(\theta, \psi) = E_{p_D(x)}[f(D_\psi(x))] + E_{p(z)}[f(-D_\psi(G_\theta(z)))] \quad (5)$$

The discriminator attempts to maximize this function whereas the generator attempts to minimize it. The goal



(a) fixed updates

(b) adaptive updates

Figure 2. Gradient convergence of different GANs for the Dirac-GAN problem. The shadow areas in Figures (a) and (b) mark the Lipschitz constraint on discriminator parameter $(-0.5, 0.5)$. The initial point $(1.5, 0.5)$ is marked in red and the end point is marked in hollow red. Ideally, the end point should be $(1.0, 0.0)$. Note that the granularity in the proposed adaptive approach is not fine and so the path traced by it is somewhat less uniform.

is to find a Nash-equilibrium, where both components cannot improve their utility. The optimization is normally done using an alternating gradient descent where when training the generator the parameters are updated by:

$$\begin{aligned}\theta_{t+1} &= \theta_t + \alpha \cdot v(\theta_t) \\ \psi_{t+1} &= \psi_t\end{aligned}\quad (6)$$

where α is the learning rate and v is the gradient vector field; and when training the discriminator the parameters are updated by:

$$\begin{aligned}\psi_{t+1} &= \psi_t + \alpha \cdot v(\psi_t) \\ \theta_{t+1} &= \theta_t\end{aligned}\quad (7)$$

The Dirac-GAN [19] consists of a generator distribution $p_\theta = \delta_\theta$ and a linear discriminator $D_\psi(x) = \psi \cdot x$. The true data distribution p_D is given by a Dirac-distribution concentrated at 1. Thus, there is one parameter θ in the generator and one parameter ψ in the discriminator. For WGAN, we define $f(t) = t$ and add a Lipschitz constraint $(-0.5, 0.5)$ on the discriminator as in the original WGAN. Thus, the GAN objective function in Equation 5 is given by:

$$L(\theta, \psi) = \psi \cdot 1 - \psi \cdot \theta \quad (8)$$

The unique equilibrium point of the objective function in Equation 5 is $\theta = 1, \psi = 0$. Since $v(\theta, \psi) = 0$ if and only if $(\theta, \psi) = (1, 0)$ as shown by:

$$v(\theta, \psi) = \begin{pmatrix} \psi \\ 1 - \theta \end{pmatrix} \quad (9)$$

Thus, when training the generator, the parameters update in Equation 6 are given by:

$$\begin{pmatrix} \theta_{t+1} \\ \psi_{t+1} \end{pmatrix} = \begin{pmatrix} 1 & \alpha \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_t \\ \psi_t \end{pmatrix} \quad (10)$$

When training the discriminator, the parameters update in Equation 7 are given by:

$$\begin{pmatrix} \theta_{t+1} \\ \psi_{t+1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -\alpha + \frac{\alpha}{\theta_t} & 1 \end{pmatrix} \begin{pmatrix} \theta_t \\ \psi_t \end{pmatrix} \quad (11)$$

We employ our proposed update strategy using an alternating gradient descent based on Equations 10 and 11. We decide on the component to update by comparing the loss change ratios (r_d and r_g) as described in Algorithm 1. The coefficient λ is set to 1. For comparison we also apply the original WGAN update strategy of alternating gradient descent with fixed update steps (5 discriminator updates for each generator update). The results are shown in Figure 2. As can be observed in sub-figure (a) fixed WGAN updates ($n_d = 5, n_g = 1$) do not converge whereas in sub-figure (b) adaptive updates following the proposed approach do converge to the Nash equilibrium point $(1, 0)$.

3.3. Network architectures

To demonstrate our proposed adaptive training strategy as described in Algorithm 1 we evaluated several network architectures with and without adaptive training. Specifically, we evaluated standard WGAN [3] and WGAN-GP [9] networks for image synthesis, Deblur GAN [17] for image deblurring using a Conditional Adversarial Network [20], and Super Resolution WGAN [13] for increasing image resolution using perceptual loss, content loss, and WGAN loss. Except for modifying the update strategy to become adaptive, we retained the original optimizer and loss functions in each of the evaluated networks, as can be found in the papers referenced above. We show the loss function for each model as in Table 1.

In our update strategy we set the coefficient λ to different values. We observe that higher values of λ (up to 10) perform better when the generation task is complex (e.g. in Deblur GAN and Super Resolution WGAN). Smaller values of λ ($1 \sim 5$) work for the WGAN and WGAN-GP image generation networks. Increasing the value of λ results in training more the generator which is necessary due to the

Model	Generator loss	Discriminator loss
WGAN	$D(G(z))$	$D(x) - D(G(z))$
WGAN-GP	$D(G(z))$	$D(x) - D(G(z)) + 10 * Gradient_Penalty$
Deblur GAN	$D(G(z)) + 100 * content_loss$	$D(x) - D(G(z)) + 10 * Gradient_Penalty$
SR WGAN	$D(G(z)) + content_loss + perceptual_loss$	$D(x) - D(G(z))$

Table 1. List of WGAN loss functions group evaluated in this paper.

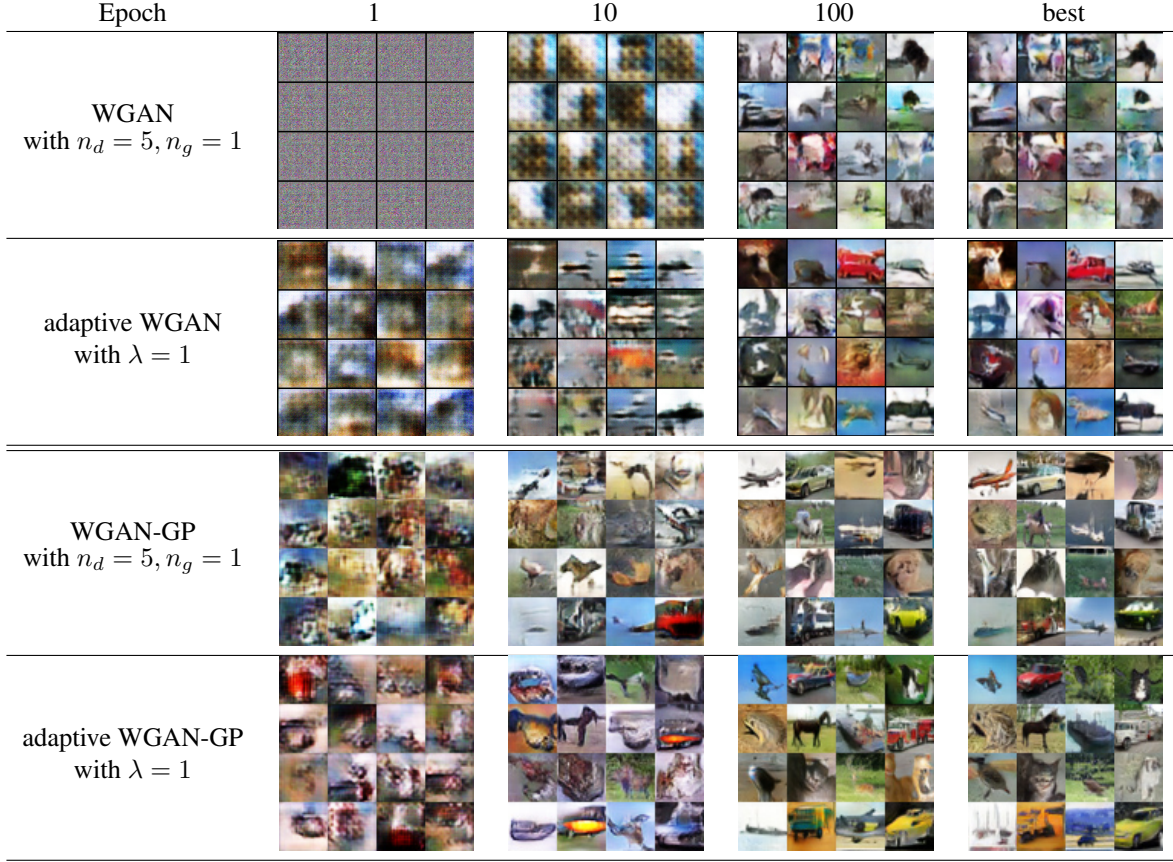


Figure 3. Examples of generated images using WGAN and the proposed adaptive WGAN trained on the CIFAR-10 dataset. The parameters n_g, n_d are the fixed number of update steps for the generator and discriminator (as suggested in the original papers). In order to ignore the coefficient λ , we set it to 1. Results are shown for several epochs.

increased complexity of the generator. Note that training the generator more is in contrast to the suggestion in the original WGAN [3] paper where it is suggested to perform 5 training steps for the discriminator for each step of the generator. Experimental evaluation results are provided in the next section.

4. Experimental evaluation

In this section, we train different GANs and compare with our updating strategy: WGAN, WGAN-GP, TTUR, Gradient Penalty, Deblur GAN and Super Resolution WGAN, and evaluate them both in quantitative and qualitative ways.

4.1. Experimental setup

We train both WGAN and adaptive WGAN for 500 epochs and using RmsProp optimizer (learning rate=0.00005 for both G and D). We train WGAN-GP and adaptive WGAN-GP for 20k iterations and using Adam [14] optimizer (learning rate=0.0001 for both G and D, $\beta_1 =$

0.5, $\beta_2 = 0.9$). And both in original WGAN and WGAN-GP training, we follow the setting in [3] [9] that updates five times for D per updating one time for G. In adaptive WGAN and adaptive WGAN-GP training, we tried different λ : 1, 3, 5, 10.

Meanwhile, we compare with the WGAN-GP TTUR (learning rate of G: $lr_g = 0.0001$, learning rate of D: $lr_d = 0.0003$ in TTUR; in adaptive WGAN-GP, $lr_g=0.0003$ and $lr_d=0.0003$ in order to keep in step), Unbalanced GAN, Gradient Penalty (we follow the hyper-parameters setting in original Gradient Penalty) and with TTUR ($lr_g=0.0001$ and $lr_d=0.0003$) and with our strategy ($lr_g=0.0003$ and $lr_d=0.0003$ in order to keep in step).

We train both Deblur GAN and adaptive Deblur GAN for 1500 epochs and using Adam optimizer (learning rate=0.0001 for both G and D, $\beta_1 = 0.5, \beta_2 = 0.999$). We train Deblur GAN 5 time on D and 1 time on G in each batch which followed the [17]. In adaptive Deblur GAN training, we tried λ : 1, 10.

We train both Super Resolution WGAN and adaptive Super Resolution WGAN for 1000 epochs and using RmsProp



Figure 4. Comparison of Super resolution WGAN (SR-WGAN) using a fixed and the proposed adaptive update strategy on the x4 DIV2K dataset (input image size is 510×282 , target image size is 2040×1182). (a) fixed update; (b) adaptive update; (c) target image. The proposed adaptive strategy (b) gets similar results to the fixed strategy (a) with faster convergence speed.

optimizer (learning rate=0.001 for both G and D). We train Super Resolution WGAN one time on D and one time on G in each batch which followed the [13]. In adaptive Super Resolution WGAN training, we tried λ : 1, 10.

4.2. Datasets

To train the WGAN we use a 100 dimensional random noise vectors as input. For targets we use 3 datasets: the CIFAR-10 [16] which includes 50,000 training examples and 10,000 validation examples; the LSUN [28] conference room dataset which has 229,069 training examples and 300 validation examples; and the labeled faces in the wild (LFW [12]) dataset which has 13,233 examples split into 10587 training examples, 1323 validation examples, and 1323 testing examples. For all of three datasets, we set up the image size to 64×64 to match the original paper [3] setting.

To train the WGAN-GP and compare with the TTUR, Gradient Penalty, we all use the 128 dimensional random noise vectors as the input, and the CIFAR-10 dataset as the targets where the images are resized to 32×32 to match the original paper [9] setting.

To train the Deblur GAN we use the Caltech-UCSD Birds-200-2011 [26] dataset which has 200 classes of bird images with size 256×256 . We synthesize blurred images from the original images using a sequence of six 3×3 Gaussian kernel convolutions. We then use the synthetic blurred images as the inputs and the corresponding original images as targets.

To train the Super Resolution WGAN (SR-WGAN) we use the DIV2K [1] dataset containing a diverse set of RGB images. In this set there are 700 training images, 100 validation images, and 100 test images. The images in this dataset are of various sizes. To synthesize the source data we down-scale each image by a factor of two, 4 times thus resulting in images having 1/16 size in each spatial dimension. The original images are then used as the corresponding targets.

4.3. Qualitative evaluation

Figure 3 shows generated images with WGAN and WGAN-GP trained on the CIFAR-10 dataset using a fixed training strategy and the proposed adaptive training strategy.



Figure 5. Comparison of Deblur GAN updated with a fixed and the proposed adaptive update strategy evaluated on the CUB-200-2011 dataset (image size is 256×256). (a) blurred image; (b) fixed update; (c) adaptive update. The proposed adaptive strategy (c) gets similar results to the fixed strategy (b) with faster convergence speed.

As can be observed, the proposed adaptive WGAN training strategy progresses faster than a fixed training strategy. When comparing the best epoch (the epoch with the best FID) results, we observe that the adaptive WGAN results are more realistic compared with the original WGAN results. Both WGAN-GP and the adaptive WGAN-GP can get some meaningful results, but the adaptive strategy progresses faster.

Figure 4 shows example results using the Super resolution WGAN with a fixed and the proposed adaptive strategy on the DIV2K dataset. Figure 5 shows example results using the Deblur GAN with a fixed and the proposed adaptive strategy on the synthetic CUB-200-2011 bird dataset. In both examples, the proposed adaptive strategy reaches the results of the fixed update strategy faster. Thus, we demonstrate that we can accelerate training while maintaining stability and convergence. A quantitative evaluation is provided in the next section.

4.4. Quantitative evaluation

To evaluate WGAN and WGAN-GP for image synthesis we use the Inception Score (IS) [25] and the Fr chet Inception Distance (FID) [7]. We train the networks using both the fixed strategy and the proposed adaptive strategy (with different coefficient λ values) on the CIFAR-10 dataset and record the first epoch when a target FID value is obtained. We record in addition the total number of G and D updates. The results are shown in Table 2. As can be observed the

Model	Parameters	FID (reached epoch)				Best epoch		Total epochs	
		200	100	50	40	IS	FID	u_g	u_d
WGAN	$n_d = 5, n_g = 1$	17	46	*	*	3.45	63.62	65166	325834
adaptive WGAN	$\lambda = 1$	6	19	456	*	3.99	48.10	253793	137206
adaptive WGAN	$\lambda = 3$	10	20	158	312	4.61	35.81	304873	86126
adaptive WGAN	$\lambda = 5$	10	28	172	339	4.54	37.32	323075	67924
adaptive WGAN	$\lambda = 10$	17	64	431	*	4.46	45.95	358568	32431

Table 2. Comparison of fixed WGAN update strategy with the proposed adaptive WGAN update strategy with different coefficients λ trained on the CIFAR10 dataset (batchsize=64, imagesize=64x64, 500 epochs). Columns 3-6 show the first epoch that reached the target FID value. Columns 7-8 show the result of the best epoch up to 500. The last to columns show the actual number of updates of G and D at the best epoch. As can be observed, while the fixed training strategy gives preference to training D, the proposed adaptive strategy ends up giving preference to training G and by doing so converges faster and to a better result. The parameters n_g, n_d is the set number of fixed update steps. A '*' indicates the score could not be reached by the training strategy. Note that the proposed adaptive strategy with a default parameter $\lambda = 1$ performs better than a fixed update strategy and so the adaptive strategy performs better than a fixed strategy without using this parameter.

Model	Parameters	LSUN Dataset					LFW Dataset				
		FID (reached epoch)			Best epoch		FID (reached epoch)			Best epoch	
		200	100	50	IS	FID	200	100	50	IS	FID
WGAN	$n_d = 5, n_g = 1$	8	42	*	3.58	135.41	75	211	*	2.60	55.82
adaptive WGAN	$\lambda = 1$	1	18	456	3.72	133.89	33	73	211	2.58	38.13
adaptive WGAN	$\lambda = 10$	5	18	82	3.61	114.26	34	77	360	2.64	37.22

Table 3. Comparison of fixed WGAN update strategy with the proposed adaptive WGAN update strategy with different coefficients λ trained on the LSUN conference room and LFW datasets (using the same experiment setting as in Table 2). The proposed training strategy converges faster and to a better result. Note that the proposed adaptive strategy with a default parameter $\lambda = 1$ performs better than a fixed update strategy and so the adaptive strategy performs better than a fixed strategy without using this parameter.

Model	Parameters	FID (reached epoch)			Best epoch	
		100	50	30	IS	FID
WGAN-GP	$n_d = 5, n_g = 1$	3	39	*	8.09	34.83
WGAN-GP TTUR	$n_d = 1, n_g = 1$	2	32	*	8.41	33.26
adaptive WGAN-GP	$\lambda = 1$	2	22	285	8.18	30.36
Unbalanced GAN	$n_d = 1, n_g = 1$	/	/	/	3.0	/
Gradient Penalty	$n_d = 1, n_g = 1$	10	44	269	5.35	27.82
Gradient Penalty TTUR	$n_d = 1, n_g = 1$	8	26	130	5.63	24.14
adaptive Gradient Penalty	$\lambda = 1$	3	15	75	5.79	25.04

Table 4. Comparison of various WGAN training schemes targeting balancing G and D training (see Section 2). Methods with “adaptive” in their name employ the proposed adaptive training scheme. The network was trained using the CIFAR-10 dataset (batchsize=64, imagesize=32x32, 1000 epochs). The symbol '/' indicates that information is not available from the original paper. As can be observed, the proposed adaptive training scheme converges faster. Note that the proposed adaptive strategy with a default parameter $\lambda = 1$ performs better than a fixed update strategy and so the adaptive strategy performs better than a fixed strategy without using this parameter.

proposed adaptive training scheme converges faster and to a better result when λ is between 1 and 5 (with best result at $\lambda = 3$). Note that a higher IS score is better whereas a lower FID score is better.

When considering the total number of G and D updates, we see that with $\lambda = 3$ the proposed adaptive training strategy trains G three to four times more than D, whereas the suggested ratio in the fixed training scheme [8] [24] [3] is to train D five times more than G. The update frequency of G and D can be effected by multiple factors such as the com-

plexity of the models, the optimizer and its parameters (e.g. learning rate), and the loss function. It is therefore difficult to empirically estimate the G and D training ratio. Moreover, even if the training ratio of G and D is somehow determined (e.g. hyper-parameter search) it may change during iterations as the algorithm gets close to convergence. The proposed adaptive training strategy alleviates the need to carefully set this parameter and provides a systematic way to continuously estimate it. While the proposed adaptive scheme still involves selecting an coefficient λ , training re-

Model	Parameters	PSNR (reached epoch)			Best epoch		Total epochs	
		24	25	25.5	PSNR	SSIM	u_g	u_d
Deblur GAN	$n_d = 5, n_g = 1$	30	126	1452	25.6323	0.7350	784833	3924167
adaptive Deblur GAN	$\lambda = 1$	8	31	38	25.6461	0.7427	3195559	1513441
adaptive Deblur GAN	$\lambda = 10$	5	16	97	25.7806	0.7500	4608989	100011

Table 5. Comparison of Deblur GAN trained without and with the proposed adaptive training scheme. The network was trained using the CUB-200-2011 bird dataset (image size=256x256). The coefficient λ in the proposed adaptive update strategy is attempted with different values. Columns 3-5 show the first epoch that reached the target PSNR value. The last two columns show the total number of training epochs for D and G. As can be observed the proposed adaptive training scheme trains the generator more than the discriminator whereas when employing the suggested fixed strategy trains the discriminator more than the generator. We observe that the proposed adaptive training strategy (with all λ values) converges faster and get a better result.

Model	Parameters	PSNR (reached epoch)			Best epoch		Total epochs	
		25	26	27	PSNR	SSIM	u_g	u_d
Super resolution WGAN	$n_d = 1, n_g = 1$	40	82	546	26.8646	0.7630	43837	43837
adaptive Super resolution WGAN	$\lambda = 1$	30	102	395	26.6840	0.7555	37899	49775
adaptive Super resolution WGAN	$\lambda = 10$	22	65	224	26.8230	0.7677	64592	23082

Table 6. Comparison of the super resolution WGAN (SR-WGAN) trained without and with the proposed adaptive training scheme. The network was trained using the DIV2K dataset (image size=2040x1182). The coefficient λ in the proposed adaptive update strategy is attempted with different values. Columns 3-5 show the first epoch that reached the target PSNR value. The last two columns show the total number of training epochs for D and G. We observe that the proposed adaptive training strategy (with $\lambda = 10$) converges faster and get a better result.

sults are less sensitive to the selection of this parameter and a simple default (e.g. $\lambda = 1$) may suffice.

A similar evaluation of the proposed adaptive WGAN training strategy when trained with different datasets is provided in Table 3. In this table training is done separately both with the LSUN conference room and LFW datasets. The evaluation on these datasets results in similar conclusions to the ones obtained when training with the CIFAR-10 dataset. The proposed adaptive training strategy converges faster and to a better result.

Comparison of various WGAN training schemes targeting balancing G and D training (see Section 2) is provided in Table 4. The compared methods include: WGAN-GP [9], WGAN-GP with TTUR [11], Unbalanced GAN [10], and Gradient Penalty [19]. Training is done using the CIFAR-10 dataset. Methods with “adaptive” in their name employ the proposed adaptive training scheme. As can be observed, the proposed adaptive training scheme converges faster.

The deblur GAN and super resolution WGAN networks are evaluated in Tables 5 and 6 respectively. The Deblur GAN network is trained using the Caltech-UCSD Birds-200-2011 dataset whereas the SR-WGAN network is trained on the DIV2K dataset. Training in both cases is done both with a fixed update strategy and the proposed adaptive training strategy. Evaluation is done using SSIM [27] and PSNR where in both a higher score means better results. In the evaluation we record the first epoch during training where a target PSNR value is achieved. In addition we record the total number of update steps for the generator

and the discriminator. We observe that the proposed adaptive training scheme converges faster and to a better result for both the deblur and super resolution networks. Further, here too the ratio of training steps for G and D obtained by the proposed adaptive training strategy does not match the recommended ratio of training steps thus supporting the need for the proposed adaptive training scheme.

5. Conclusion

In this paper, we propose an adaptive WGAN training strategy which automatically determines the sequence of generator and discriminator training steps. The proposed approach compares the loss change ratio of the generator and discriminator to decide on the next component (G or D) to be trained and so balances the training rate of the generator and discriminator. We show that a WGAN with this strategy could reach the local Nash Equilibrium point for the Dirac-GAN. Experimental evaluation results using different networks and datasets show that the proposed adaptive training scheme normally converges faster and to a lower minimum. Another advantage of the proposed adaptive update strategy is that it alleviates the need to empirically determine the number of update steps for the generator and discriminator. In future work, we will investigate additional update strategies suitable for various GAN structures and loss terms.

References

- [1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.
- [2] Martín Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *5th International Conference on Learning Representations, ICLR 2017*, 2017.
- [3] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 2017.
- [4] David Berthelot, Tom Schumm, and Luke Metz. BEGAN: boundary equilibrium generative adversarial networks. *CoRR*, abs/1703.10717, 2017.
- [5] Chen Chen, Shuai Mu, Wanpeng Xiao, Zexiong Ye, Liesi Wu, and Qi Ju. Improving image captioning with conditional generative adversarial nets. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, pages 8142–8150, 2019.
- [6] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 793–802. PMLR, 2018.
- [7] D. C. Dowson and B. V. Landau. The Fréchet distance between multivariate normal distributions. *Journal of Multivariate Analysis*, 12(3):450–455, September 1982.
- [8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.
- [9] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 5767–5777, 2017.
- [10] Hyungrok Ham, Tae Joon Jun, and Daeyoung Kim. Unbalanced gans: Pre-training the generator of generative adversarial network using variational autoencoder. *CoRR*, abs/2002.02112, 2020.
- [11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 6626–6637, 2017.
- [12] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.
- [13] JustinhoCHN. Srgan wasserstein. GitHub repository, 2018.
- [14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015.
- [15] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014.
- [16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012.*, pages 1106–1114, 2012.
- [17] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiri Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, pages 8183–8192, 2018.
- [18] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pages 105–114, 2017.
- [19] Lars M. Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 3478–3487, 2018.
- [20] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.
- [21] Vaishnavh Nagarajan and J. Zico Kolter. Gradient descent gan optimization is locally stable. In *Advances in Neural Information Processing Systems 30*, pages 5585–5595. 2017.
- [22] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016*, pages 271–279, 2016.
- [23] Xu Ouyang, Xi Zhang, Di Ma, and Gady Agam. Generating image sequence from description with LSTM conditional GAN. In *24th International Conference on Pattern Recognition, ICPR 2018*, pages 2456–2461, 2018.
- [24] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *4th International Conference on Learning Representations, ICLR 2016*, 2016.
- [25] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016*, pages 2226–2234, 2016.
- [26] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, California Institute of Technology, 2011.

- [27] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004.
- [28] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365, 2015.