ICLR Reproducibility Challenge: Improving Generalization and Stability of Generative Adversarial Networks

Junze Li, Siyuan Li, Wanhao Zhou
School of Computer and Communication Sciences, EPFL, Switzerland

***Abstract*—Generative Adversarial Networks (GANs) are powerful methods widely used in generative models. However, research has pointed out that training GAN suffers from *poor generalization* and *instability* problems. Methods involved in tackling such problems typically penalize the gradient during the training process in various ways. In this report, we study a new method called *zero-centered gradient penalty (0-GP)*. Through theoretical analysis and extensive experiments, we show the prospect of the proposed method could help improve the model's generalization ability and stabilize the training process.**

I. Introduction

GANs [1] perform well on various tasks concerning learning complex real-world distributions and has been proved powerful as generative models. However, training GANs is inevitably faced with two major problems.

Poor generalization capability. Measuring the generalization capability is tricky. Arora et al. [2] defines a gap between the optimal divergence of two distributions and the empirical one. Arora and Zhang [3] attempts to measure the generalization capability by number of modes learned by GAN. Zhang et al. [4] shows that the generalization capability of GANs is a trade-off between generalization and discrimination capabilities. Low capacity discriminator lacks discriminative power but elevates generalization capability, whereas high capacity discriminator solicits overfitting. To this end, improving the generalization capability of GANs is hard when balancing the generalization and discrimination.

Training instability. The training process of GAN is usually unstable, subject to changes of hyperparameters and lack of convergence guarantee [5]. Research has shown that the instability characteristic, e.g. gradient exploding, has much to do with the loss function, where the original GAN leverages the Jensen-Shannon divergence to quantify the divergence of real and model probability distributions [6]. Arjovsky et al. [7] substitutes the original GAN loss function with an approximation of Wasserstein-1 distance, which is a weaker assumption than Jensen-Shannon divergence, and the resulting Wasserstein GAN (WGAN) demonstrates good tractability. Heusel et al. [8] proposes the Two Time-Scale Update Rule (TTUR) for training GANs with stochastic gradient descent, which forces GANs to converge to the local equilibria.

In the paper we study [1], the author first addresses problem regarding the generalization capacity of GANs and the gradient exploding characteristic from an alternative perspective. The proposed gradient penalty method, namely *zero-centered gradient penalty (0-GP)*, is therefore carefully designed to tackle the gradient exploding problem and help propel the model to generalize better.

In this report[2], following the author's idea, we manage to give extra (or alternative) explanations for problems and proofs discussed in the paper and conduct extensive experiments in an attempt to reproduce the results listed in the original paper. The intuition behind the proposed solution is as follows. Optimizing the discriminator using the original loss of GAN (as well as that of WGAN) innately contradicts the attributes of the theoretically optimal discriminator. The problem also results from the limited size of data in contrast to the high-dimension space. Also, the gradient exploding problem aggravates as the learned model distribution gets closer to the real data distribution.

Throughout the experiments, we compare the proposed 0-GP penalty method with the following baselines.

1) The original GAN without penalties (No-GP).
2) WGAN with one-centered gradient penalty (1-GP)[9].
3) GAN with zero-centered gradient penalty on samples (0-GP-sample)[5].

p_r	the target distribution
p_g	the model distribution
p_z	the noise distribution
$supp(p)$	support of the distribution p
$G(\cdot)$	function of the generator
$D(\cdot)$	function of the discriminator
$\boldsymbol{x} \sim p_r$	a real sample
$\boldsymbol{z} \sim p_z$	a noise vector from distribution p_z
$\boldsymbol{y} = G(\boldsymbol{z})$	a generated sample
$\mathcal{D}_r = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$	set of n real samples
$\mathcal{D}_g^{(t)} = \{\boldsymbol{y}_1^{(t)}, \cdots, \boldsymbol{y}_m^{(t)}\}$	m generated samples at t-th step
$\mathcal{D}^{(t)} = \mathcal{D}_r \cup \mathcal{D}_g^{(t)}$	training dataset at t-th step

Table I: List of notations

[1] https://openreview.net/pdf?id=ByxPYjC5KQ

[2] For GitHub issue, please refer to: https://github.com/reproducibility-challenge/iclr_2019/issues/84

We use notations described in Table I.

II. Preliminaries

A. GAN

The original loss function of GAN is defined as:

$$\begin{aligned}\min_G \max_D \mathcal{L} = \ & \mathbb{E}_{\boldsymbol{x}\sim p_r}\big[\log(D(\boldsymbol{x})\big] \\ & + \mathbb{E}_{\boldsymbol{z}\sim p_z}\big[\log(1-D(G(\boldsymbol{z})))\big]\end{aligned} \tag{1}$$

The discriminator D is designed to maximize the Eq. 1 whereas the generator G minimizes it, turning the optimization process into a two-player minimax game.

The original GAN loss function models the divergence of two distributions using Jensen-Shannon divergence, which could be divergent in low dimension probability space. WGAN instead uses the approximation of Wasserstein-1 distance, a weaker condition compared with Jensen-Shannon divergence to stabilize the training of GAN and helpfully alleviates mode collapse [7]. The loss function of WGAN is defined as:

$$\begin{aligned}\min_G \max_D \mathcal{L}_W = \ & \mathbb{E}_{\boldsymbol{x}\sim p_r}\big[D(\boldsymbol{x})\big] \\ & - \mathbb{E}_{\boldsymbol{z}\sim p_z}\big[D(G(\boldsymbol{z}))\big]\end{aligned} \tag{2}$$

where D satisfies the Lipschitz condition with constant equals to 1.

B. Optimality of the discriminator

Theoretical Optimal. Given the model distribution p_g and the true target distribution p_r, as described in [1], the theoretically optimal discriminator D^* trained on the set $supp(p_r) \cup supp(p_g)$ for a fixed generator G has the following property:

$$D^*(\boldsymbol{v}) = \frac{p_r(\boldsymbol{v})}{p_r(\boldsymbol{v}) + p_g(\boldsymbol{v})}, \forall \boldsymbol{v} \in supp(p_r) \cup supp(p_g) \tag{3}$$

As the training progresses, the learned generator approximates p_g to p_r. When the training reaches the global equilibrium, $p_g = p_r$ and the optimal discriminator yields $D^*(\boldsymbol{v}) = \frac{1}{2}, \forall \boldsymbol{v} \in supp(p_r) \cup supp(p_g)$.

Empirical Optimal. The underlying distribution of real data p_r is unknown and thus is approximated by training data $\mathcal{D}^{(t)}$ which is limited in size. The optimal discriminator trained on $\{\mathcal{D}^{(t)}\}_t$ is defined as the empirically optimal discriminator $\hat{D}^*$.

ϵ-Optimal. When trained on the empirical dataset $\{\mathcal{D}^{(t)}\}_t$, a weaker optimality condition is defined as follows.

Given ϵ, the discriminator D is ϵ-optimal iff.

$$\begin{aligned} D(\boldsymbol{x}) &\geq \frac{1}{2} + \frac{\epsilon}{2}, \forall \boldsymbol{x} \in \mathcal{D}_r \\ D(\boldsymbol{y}) &\leq \frac{1}{2} - \frac{\epsilon}{2}, \forall \boldsymbol{y} \in \mathcal{D}_g \end{aligned} \tag{4}$$

III. Generalization Capability and Stability of GANs

The section is organized in two parts. First, based on the author's theory and our understanding, we show that GAN without proper regularization is inherently with the following problems:

1) Training the discriminator D leads to the empirically optimal $\hat{D}^*$ rather than the theoretically optimal D^*.
2) As p_g approaches p_r, i.e., $\boldsymbol{y}^{(t)}$ gets closer to $\boldsymbol{x}$, maintaining the optimality of the discriminator leads to gradient exploding in the original GAN loss without penalties.

Second, we introduce the author's gradient penalty method and give theoretical comparation with other penalty methods mentioned in Section I.

A. Pitfalls of GANs

Here we limit the discussion to the WGAN loss, since it admits a simpler arithmatic form. Its loss function is defined in Eq. 2.

1) Loss of theoretical optimality: Training on the empirical dataset $\{\mathcal{D}^{(t)}\}_t$, D is pushed to $\hat{D}^*$ rather than D^*.

Proof: Considering the attribute of the loss function, the discriminator D is said to be empirically optimal such that:

$$\begin{aligned} D(\boldsymbol{x}) &= 1, \forall \boldsymbol{x} \in \mathcal{D}_r \\ D(\boldsymbol{y}) &= 0, \forall \boldsymbol{x} \in \mathcal{D}_g^{(t)} \end{aligned} \tag{5}$$

By definition in Section II-B, this is attained in the empirical optimal point, ergo $\hat{D}^*$. We will see that the optimal state $\hat{D}^*$ does not reflect the true divergence between p_g and p_r, which is supposed to be minimized when training GANs. Given that $\boldsymbol{x}_i \sim p_r$, $\boldsymbol{y}_j = G(\boldsymbol{z}_j) \sim p_g$, without loss of generality, we could denote $\boldsymbol{X}$ and $\boldsymbol{Y}$ as a continuous random variable[3] taking values in $\{\boldsymbol{x}_i\}_i$ and $\{\boldsymbol{y}_j\}_j$ respectively. Thus, we have:

$$\begin{aligned} \mathbb{P}(\boldsymbol{x} \in \mathcal{D}_g^{(t)}) &= \sum_{i=1}^{|\mathcal{D}_g^{(t)}|} \mathbb{P}(\boldsymbol{X} = \boldsymbol{y}_i) = 0, \forall \boldsymbol{x} \in \mathcal{D}_r \\ \mathbb{P}(\boldsymbol{y} \in \mathcal{D}_r) &= \sum_{j=1}^{|\mathcal{D}_r|} \mathbb{P}(\boldsymbol{Y} = \boldsymbol{x}_j) = 0, \forall \boldsymbol{y} \in \mathcal{D}_g^{(t)} \end{aligned} \tag{6}$$

This implies that at any given step t, set $\mathcal{D}_g^{(t)}$ is disjoint with $\mathcal{D}_r$ with probability of 1. As suggested in Eq. 5 and 6, the attained optimality is only a reflection of the disjoint attribute between $\mathcal{D}_g^{(t)}$ and $\mathcal{D}_r$. Therefore, the optimal empirical discriminator does not actually approximate p_g to p_r close enough, concluding the proof. ■

[3]We immediately have $\forall \boldsymbol{x}, \mathbb{P}(\boldsymbol{X} = \boldsymbol{x}) = 0$, the property holds true for $\boldsymbol{Y}$ as well.

2) Gradient exploding: As p_g approaches p_r, i.e. $\boldsymbol{y}^{(t)}$ gets closer to $\boldsymbol{x}$, maintaining the optimality of the discriminator leads to gradient exploding.

Proof: Heusel et al. [8] shows that fixed number of updates for the discriminator fails to maintain the optimality of the discriminator when $\boldsymbol{y}^{(t)}$ gets closer to $\boldsymbol{x}$. Consequently, the learning rate could be set larger for the discriminator, which is exactly the basic idea of the Two Time-Scale Update Rule (TTUR) for training GANs.

In our setting, this could be interpreted as: the number of updates should increase as the training proceeds in order to maintain the optimality of the discriminator. Here, maintaining the optimality of the discriminator refers to maximizing the objective function of D before the generator updates.

We consider the setting where our goal is to maintain the ϵ-optimality, which is a weaker condition. Simplify the training set at each step as two data points, i.e. at t-th step (where t is arbitary), $\mathcal{D}^{(t)} = \{\boldsymbol{x}, \boldsymbol{y}^{(t)}\}$. Optimizing GAN generally shrinks the norm of the direction vector $\boldsymbol{u}^{(t)}$ between real samples and generated samples, where $\boldsymbol{u}^{(t)} := \boldsymbol{x} - \boldsymbol{y}^{(t)}$. The norm of the directional derivative of D in the direction $\boldsymbol{u}^{(t)}$ at $\boldsymbol{x}$ is:

$$\begin{aligned}\|(\nabla_{\boldsymbol{u}^{(t)}} D)_{\boldsymbol{x}}\| &= \lim_{\boldsymbol{y}^{(t)} \to \boldsymbol{x}} \frac{\|D(\boldsymbol{x}) - D(\boldsymbol{x} - \boldsymbol{u}^{(t)})\|}{\|\boldsymbol{x} - (\boldsymbol{x} - \boldsymbol{u}^{(t)})\|} \\ &= \lim_{\boldsymbol{y}^{(t)} \to \boldsymbol{x}} \frac{\|D(\boldsymbol{x}) - D(\boldsymbol{y}^{(t)})\|}{\|\boldsymbol{x} - \boldsymbol{y}^{(t)}\|} \\ &\geq \lim_{\boldsymbol{y}^{(t)} \to \boldsymbol{x}} \frac{|\epsilon|}{\|\boldsymbol{x} - \boldsymbol{y}^{(t)}\|} \to \infty\end{aligned} \tag{7}$$

This implies the gradients of points on the line connecting $\boldsymbol{x}$ and $\boldsymbol{y}^{(t)}$ will explode. Also, since $\mathcal{D}^{(t)}$ contains only two data points in the simplified setting, optimizing Eq. 2 w.r.t. D is to maximize:

$$D(\boldsymbol{x}) - D(\boldsymbol{y}^{(t)}) = \int_{\mathcal{C}} (\nabla_{\boldsymbol{s}} D)_{\boldsymbol{v}} \cdot d\boldsymbol{s} \tag{8}$$

where $\mathcal{C}$ is the line connecting $\boldsymbol{x}$ and $\boldsymbol{y}^{(t)}$, $\boldsymbol{v} \in \mathcal{D}^{(t)}$. Since $\|\mathcal{C}\| = \|\boldsymbol{x} - \boldsymbol{y}^{(t)}\|$ is shrinking, maximizing $D(\boldsymbol{x}) - D(\boldsymbol{y}^{(t)})$ pushes the direction derivative $(\nabla_{\boldsymbol{s}} D)_{\boldsymbol{v}}$ to explode, which is exactly described in Eq. 7. ■

B. Gradient penalty

Following the scenario in Section III-A, we derive the necessary conditions for a theoretically optimal discriminator based on the problems with GAN loss described before.

Eq. 3 shows that:

$$D^*(\boldsymbol{x}) = D^*(\boldsymbol{y}) = \frac{1}{2}, \forall \boldsymbol{x} \in supp(p_r), \boldsymbol{y} \in supp(p_g) \tag{9}$$

Thus, $\forall \boldsymbol{v} \in supp(p_r) \cup supp(p_g)$, following the similar idea in Eq. 8, we have:

$$D^*(\boldsymbol{x}) - D^*(\boldsymbol{y}) = \int_{\tilde{\mathcal{C}}} (\nabla_{\boldsymbol{s}} D^*)_{\boldsymbol{v}} \cdot d\boldsymbol{s} = 0 \tag{10}$$

Unlike Eq. 8 where $\mathcal{C}$ is defined as the line connecting data points in $\mathcal{D}_r$ and $\mathcal{D}_g^{(t)}$, $\tilde{\mathcal{C}}$ is an arbitrary path from $\boldsymbol{y}$ to $\boldsymbol{x}$ since continuous probability distributions p_g and p_r are dense sets.

Consequently, to push an empirically discriminator D that we train toward D^*, we require D have the following properties:

1) $(\nabla D)_{\boldsymbol{v}} \to 0, \forall \boldsymbol{v} \in supp(p_r) \cup supp(p_g)$
2) $D(x) - D(y) = \int_{\tilde{\mathcal{C}}} (\nabla_{\boldsymbol{s}} D)_{\boldsymbol{v}} \cdot d\boldsymbol{s} \to 0, \forall \boldsymbol{x} \sim p_r, \boldsymbol{y} \sim p_g$, $\tilde{\mathcal{C}}$ is an arbitrary path from $\boldsymbol{y}$ to $\boldsymbol{x}$.

An immediate conclusion that the gradient penalty that helps D satisfy the aforementioned conditions admit the form:

$$\lambda_1 \mathbb{E}_{\boldsymbol{v}} \left[\|(\nabla D)_{\boldsymbol{v}}\|^2 \right] + \lambda_2 \mathbb{E}_{\boldsymbol{x}, \boldsymbol{y}} \left[(D(\boldsymbol{x}) - D(\boldsymbol{y}))^2 \right] \tag{11}$$

However, the second term does not guarantee that $(\nabla_{\boldsymbol{s}} D)_{\boldsymbol{v}} \to 0, \forall \boldsymbol{v} \in supp(p_r) \cup supp(p_g)$.

Zero-Centered Gradient Penalty (0-GP). To better address the problem, the gradient penalty calculates the expectation w.r.t. points on the arbitrary path $\tilde{\mathcal{C}}$ instead of the set $supp(p_r) \cup supp(p_g)$. Zero-centered gradient penalty (0-GP) is defined as:

$$\lambda \mathbb{E}_{\boldsymbol{v} \in \tilde{\mathcal{C}}} \left[\|(\nabla D)_{\boldsymbol{v}}\|^2 \right] \approx \lambda \mathbb{E}_{\widetilde{\boldsymbol{x}}} \left[\|(\nabla D)_{\widetilde{\boldsymbol{x}}}\|^2 \right] \tag{12}$$

where the arbitrary path $\tilde{\mathcal{C}}$ is approximated by a straight line from $\boldsymbol{y}$ to $\boldsymbol{x}$, i.e. $\widetilde{\boldsymbol{x}} = \alpha \boldsymbol{x} + (1 - \alpha)\boldsymbol{y}$ and $\alpha \sim \mathcal{U}(0, 1)$. Combining with the original GAN loss, the total objective function is:

$$\mathcal{L}_{\text{0-GP}} = \mathcal{L} - \lambda \mathbb{E}_{\widetilde{\boldsymbol{x}}} \left[\|(\nabla D)_{\widetilde{\boldsymbol{x}}}\|^2 \right] \tag{13}$$

One-Centered Gradient Penalty (1-GP). Eq. 2 requires that D satisfy the Lipschitz condition with constant equals to 1. One-Centered Gradient Penalty (1-GP) [9] was proposed to fulfill the requirement instead of the originally used gradient weight clipping technique [7] and proved successful in preventing gradient exploding [10]. The loss function is similar to 0-GP, taking the form:

$$\mathcal{L}_{\text{1-GP}} = \mathcal{L}_W - \lambda \mathbb{E}_{\widetilde{\boldsymbol{x}}} \left[(\|(\nabla D)_{\widetilde{\boldsymbol{x}}}\| - 1)^2 \right] \tag{14}$$

where $\widetilde{\boldsymbol{x}}$ is defined identical to that of the 0-GP.

Zero-Centered Gradient Penalty on Samples (0-GP-sample). Mescheder et al. [5] proposes [4] a gradient penalty method that is zero-centered and works on samples to force the gradient of samples in $\mathcal{D}_r \cup \mathcal{D}_g$ to be $\mathbf{0}$. It is different from 0-GP, which picks samples on a path between $\boldsymbol{y}$ and $\boldsymbol{x}$. The loss function is:

$$\mathcal{L}_{\text{0-GP-sample}} = \mathcal{L} - \lambda \mathbb{E}_{\boldsymbol{v}} \left[\|(\nabla D)_{\boldsymbol{v}}\|^2 \right] \tag{15}$$

where $\boldsymbol{v} \in \mathcal{D}^{(t)}$.

[4] This is a simplified version of the gradient penalty proposed by Roth et al. [11], which has been proved to have comparable performances.

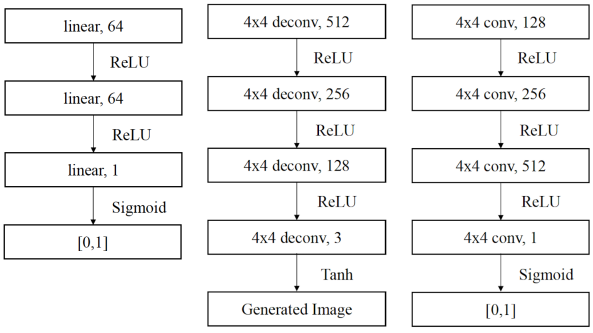
IV. Experiments

In this section, we present our reproduction results as well as analyses of the experiments described in the original paper. We test different gradient penalty schemes of GANs on synthetic dataset and real-world dataset: MNIST and CIFAR-10. For experiments done on ImageNet in the paper, we use CIFAR-10 instead due to the constraint of computation power. Tested gradient penalty schemes include:

1) The original GAN without penalties (No-GP).
2) WGAN with one-centered gradient penalty (1-GP).
3) GAN with zero-centered gradient penalty on samples (0-GP-sample).
4) GAN with zero-centered gradient penalty (0-GP).

The architectures of discriminators and generators remain invariant in each experiment.

A. Experimental Setup



(a) (b) (c)

Figure 1: Architecture of discriminator and generator

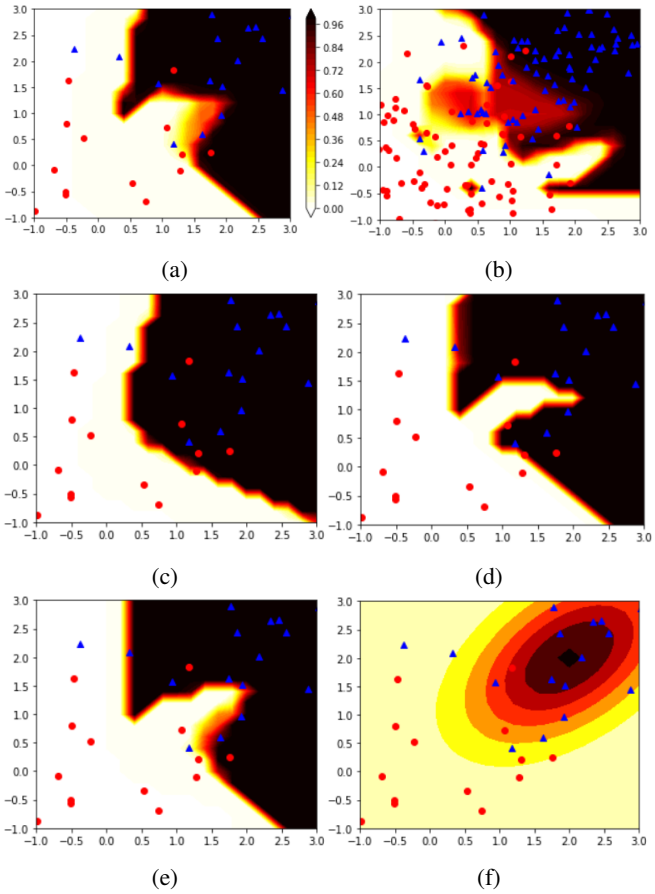
Fig. 1a illustrates the architecture of the discriminator trained on synthetic dataset; Fig. 1b and Fig. 1c represent the architectures of the generator and discriminator used on real-world dataset.

B. Value surfaces of discriminators in bivariate Gaussian distribution

In this experiment, illustrated in Fig. 2 and 3, we randomly pick samples from a bivariate Gaussian distribution as the real data points (marked in blue); also, we sample from another Gaussian distribution as the noise input (marked in red). We use Multi-Layer Perceptron (MLP) as the discriminator and apply different gradient penalty schemes.

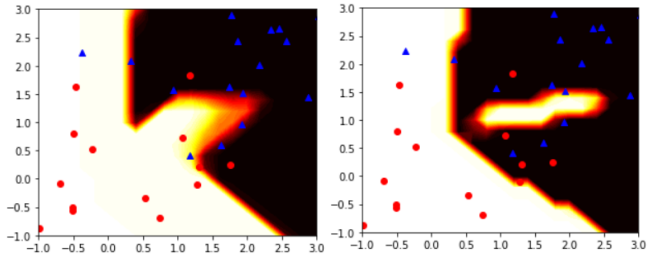
We aim to test the generalization capability of different training methods of GANs. If the model distribution converges to the target distribution, i.e. the theoretical optimality, this indicates good generalization capability of the network.

The discriminators trained with different gradient penalties are as follows (corresponding to the labels in Fig. 2):



(a) (b) (c) (d) (e) (f)

Figure 2: Value surfaces of discriminators with 10000 iterations. Red points are generated data points; blue points are real data points



(a) (b)

Figure 3: Value surfaces of discriminators with 2000 iterations

(a) The discriminator trained without gradient penalty (No-GP).
(b) The discriminator trained without gradient penalty (No-GP) but on a larger dataset.
(c) The discriminator trained with one-centered gradient penalty (1-GP).
(d) The discriminator trained with zero-centered gradient

penalty on real or generated samples (0-GP-sample).

(e) The discriminator trained with zero-centered gradient penalty (0-GP).

(f) The theoretically optimal discriminator, representing the ground truth.

Fig. 3a illustrates the effect of 0-GP-sample in the first 2000 iterations; Fig. 3b illustrates that of 0-GP.

From both figures, we observe that:

1) No-GP has an even worse generalization properties when trained on a larger dataset. This implies that the theoretical optimality cannot be achieved by using larger training samples.
2) Both 0-GP-sample and 1-GP do not help to improve the generalization capacity of GANs.
3) The value surface in Fig. 2e is smoother than that in Fig. 2d, indicating the discriminator generalizes better.
4) Compared with the ground truth, the value surface in Fig. 3b is smoother than in that in Fig. 3b.

To sum up, the discriminator trained with 0-GP yields the smoothest value surface, indicating its generalization capability is better than any of other schemes. Also, 0-GP generates a value surface most similar to that of the theoretically optimal one. This is important for GANs when confronted with unknown data.

C. Results on MNIST

We use Adam optimizer [12] with default $\beta_1 = 0.5$ and $\beta_2 = 0.9$ as suggested in the original paper to train all GANs. Results on MNIST dataset are displayed in Fig. 4. Training methods and parameters are as follows (corresponding to the labels in Fig. 4:

(a) No-GP, iter=10 epochs.
(b) 0-GP-sample, iter=10 epochs.
(c) 1-GP, iter=10 epochs.
(d) 0-GP, iter=10 epochs.
(e) 0-GP, iter=50 epochs.
(f) 0-GP, iter=10 epochs, $\beta_1 = 0.9$, $\beta_2 = 0.9$.
(g) 0-GP, iter=5 epochs, trained with TTUR.

After iteration of 10 epochs, GANs trained with No-GP and 1-GP exhibit mode collapse. 0-GP-sample and 0-GP are stable and can still learn different modes in MNIST. From Fig. 4d, we observe that 0-GP has a better performance than 0-GP-sample on generalization.

When we change the learning hyperparameters β_1 and β_2 in Adam, results of 0-GP are shown in Fig. 4f, which indicate that the 0-GP is robust against changes of hyperparameters.

Also, we train the 0-GP with TTUR, setting the learning rate of discriminator 3 times higher than that of the generator. Results shown in Fig. 4g show comparable quality with those trained with 10 epochs. The training of GANs with 0-GP could thus be accelrated.

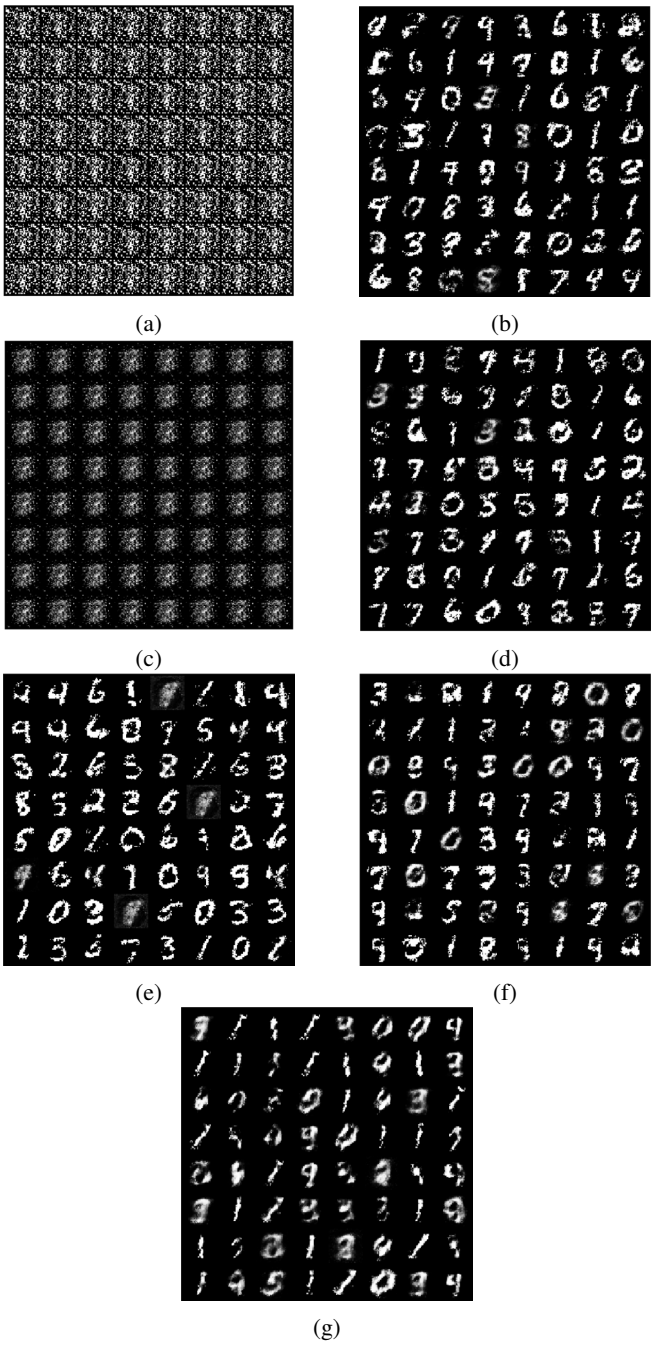
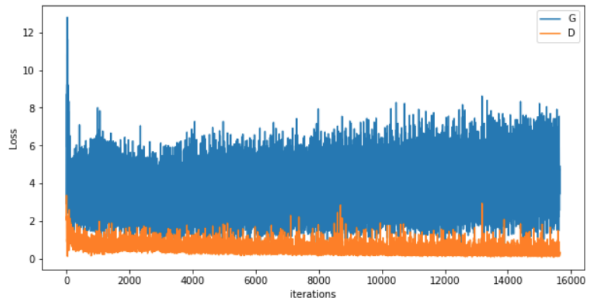


Figure 4: Results on MNIST

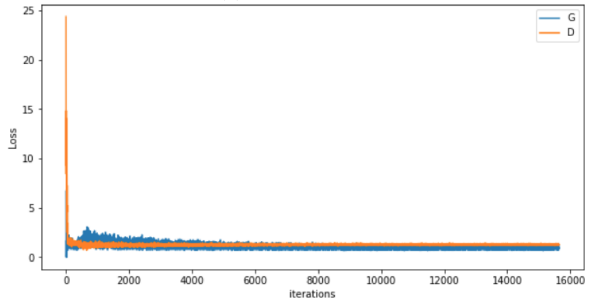
D. Results on CIFAR-10

For image generation task, we use a smaller dataset CIFAR-10 due to the constraint of computation power, and thus we are unable to report the inception score. We train GAN-No-GP, GAN-0-GP-sample and GAN-0-GP on CIFAR-10. We report the learning curves for three GANs, illustrated in Fig. 5. Generated figures are shown in Fig. 6 to 8. We could clearly observe that the original loss fluctu-

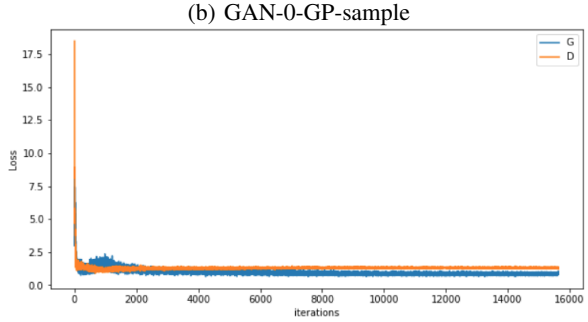
ates severely. After applying gradient penalties, both 0-GP and 0-GP-sample force the learning curve to be smoother. Similar to results on MNIST, the difference between two penalty schemes is trivial. GAN-0-GP is slightly better.



(a) GAN-No-GP



(b) GAN-0-GP-sample



(c) GAN-0-GP

Figure 5: Loss functions of generator and discriminator

V. Conclusion

In this report, we study the paper which theorizes the innate problems w.r.t. the original GAN loss and put forward a possible solution called *zero-centered gradient penalty*, aiming at improving the generalization capability and stability. Based on the theory in the paper and our understanding, we formulate the problems related to the poor performance of GANs arising from two aspects:

1) Training the discriminator D leads to the empirically optimal $\hat{D}^*$ rather than the theoretically optimal D^*.
2) As p_g approaches p_r, maintaining the optimality of the discriminator leads to gradient exploding.

Furthermore, we reproduce most of the original paper's experiments in comparison with other popular gradient

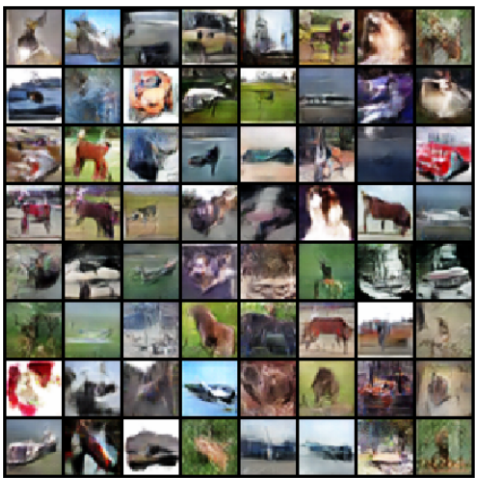


Figure 6: GAN-No-GP trained on CIFAR-10

Figure 7: GAN-0-GP-sample trained on CIFAR-10

Figure 8: GAN-0-GP trained on CIFAR-10

penalty methods: 0-GP-sample and 1-GP. Based on our experimental evidence, we observe that:

1) GANs without proper regularization perform badly and are hard to train. Among all three gradient penalty methods, 1-GP performs worst and 0-GP performs best.
2) The difference between 0-GP and 0-GP-sample displayed in our experiment is trivial, which *differs from some of the experimental results described in the original paper*. We briefly explain this phenomenon: the difference between 0-GP and 0-GP-sample takes effect when the generated distribution is close enough to the real distribution. However, when they are close to each other, the points on the line that connecting $\boldsymbol{x}$ and $\boldsymbol{y}^{(t)}$ are almost identical to the set $\mathcal{D}^{(t)}$.
3) The training of GANs with 0-GP could be accelerated with Two Time-Scale Update Rule (TTUR), i.e. setting the learning rate for the discriminator higher. Also, the training process is much more robust against changes of hyperparameters.

To conclude, the proposed 0-GP method helps stabilize the training process and force the model to generalize better. The interpretation of the generalization capability is novel and the proposed method is inspiring. The overall writing of the paper is sound, but we think there are a few things the author might consider to improve.

1) The size of training set used in image generating experiments is not clearly stated.
2) In Section 6.1, the analysis of 1-GP states that *1-GP helps improve generalization*. This contradicts with the results in Table 1.
3) The learning rate given in Appendix E.1 might be problematic for some penalty methods, e.g. 1-GP. According to our experiment, we suggest that this should be set smaller.
4) The network architecture given in Appendix E.1 might also be problematic. Through experiments on MNIST, we think there should be an extra Tanh layer added to the output layer of the generator.

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[2] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang, "Generalization and equilibrium in generative adversarial nets (gans)," *arXiv preprint arXiv:1703.00573*, 2017.

[3] S. Arora and Y. Zhang, "Do gans actually learn the distribution? an empirical study," 2017.

[4] P. Zhang, Q. Liu, D. Zhou, T. Xu, and X. He, "On the discrimination-generalization tradeoff in gans," *arXiv preprint arXiv:1711.02771*, 2017.

[5] L. Mescheder, A. Geiger, and S. Nowozin, "Which training methods for gans do actually converge?" in *International Conference on Machine Learning*, 2018, pp. 3478–3487.

[6] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," *arXiv preprint arXiv:1701.04862*, 2017.

[7] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[8] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 6626–6637.

[9] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, 2017, pp. 5767–5777.

[10] W. Fedus, M. Rosca, B. Lakshminarayanan, A. M. Dai, S. Mohamed, and I. Goodfellow, "Many paths to equilibrium: Gans do not need to decrease adivergence at every step," *arXiv preprint arXiv:1710.08446*, 2017.

[11] K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann, "Stabilizing training of generative adversarial networks through regularization," in *Advances in Neural Information Processing Systems*, 2017, pp. 2018–2028.

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