
GAN-QP: A Novel GAN Framework without Gradient Vanishing and Lipschitz Constraint

Jianlin Su

School of Mathematics
Sun Yat-sen University
Guangdong, China
bojone@spaces.ac.cn

Abstract

We know SGAN may have a risk of gradient vanishing. A significant improvement is WGAN, with the help of 1-Lipschitz constraint on discriminator to prevent from gradient vanishing. Is there any GAN having no gradient vanishing and no 1-Lipschitz constraint on discriminator? We do find one, called GAN-QP.

To construct a new framework of Generative Adversarial Network (GAN) usually includes three steps: 1. choose a probability divergence; 2. convert it into a dual form; 3. play a min-max game. In this article, we demonstrate that the first step is not necessary. We can analyse the property of divergence and even construct new divergence in dual space directly. As a reward, we obtain a simpler alternative of WGAN: GAN-QP. We demonstrate that GAN-QP have a better performance than WGAN in theory and practice.

1 From Divergence to GAN

1.1 Divergence

Most Generative Adversarial Networks (GANs, Goodfellow et al. (2014)) are based on a certain form of probability divergence. A divergence is a function \mathcal{D} of two variables p, q satisfies the following definition:

Definition 1 If \mathcal{D} is function of two variables p, q satisfies the following properties:

1. $\mathcal{D}[p, q] \geq 0$;
2. $p = q \Leftrightarrow \mathcal{D}[p, q] = 0$.

We say \mathcal{D} is a divergence between p and q .

Compared with the axiomatic definition of distance, a divergence do not need symmetry and triangle inequality necessarily. Divergence only keeps the fundamental property for measuring the difference between p and q .

1.2 Dual Form

If p, q represent two probability distributions, $\mathcal{D}[p, q]$ becomes a functional and we call it probability divergence. For example, we have Jensen-Shannon divergence (JS divergence):

$$\mathcal{JS}[p(x), q(x)] = \frac{1}{2} \int p(x) \log \frac{p(x)}{\frac{1}{2}[p(x) + q(x)]} dx + \frac{1}{2} \int q(x) \log \frac{q(x)}{\frac{1}{2}[p(x) + q(x)]} dx \quad (1)$$

In most cases, we can find a dual form for a probability divergence (Nowozin et al., 2016). For example, the dual form of JS divergence is

$$\begin{aligned}\mathcal{JS}[p(x), q(x)] &= \max_T \frac{1}{2} \int p(x) \log \sigma(T(x)) dx + \frac{1}{2} \int q(x) \log(1 - \sigma(T(x))) dx + \log 2 \\ &= \max_T \frac{1}{2} \mathbb{E}_{x \sim p(x)} [\log \sigma(T(x))] + \frac{1}{2} \mathbb{E}_{x \sim q(x)} [\log(1 - \sigma(T(x)))] + \log 2\end{aligned}\quad (2)$$

Here $\sigma(x) = 1/(1 + e^{-x})$ is sigmoid function. The dual form can convert the integral of original divergence into a sampling form, which allows us to estimate it by Monte Carlo method. That is the essential to GAN. The dual form always has a max operation, which means a divergence is a supremum of a family of functions.

1.3 Min-Max Game

With the help of dual form of probability divergence, we can train a generator to generate the distribution we are interested via playing a min-max game. For example, using (2) we have

$$G, T = \arg \min_G \arg \max_T \mathbb{E}_{x \sim p(x)} [\log \sigma(T(x))] + \mathbb{E}_{x = G(z), z \sim q(z)} [\log(1 - \sigma(T(x)))] \quad (3)$$

For a fixed T , the goal of G is

$$\begin{aligned}G &= \arg \min_G \mathbb{E}_{x \sim p(x)} [\log \sigma(T(x))] + \mathbb{E}_{x = G(z), z \sim q(z)} [\log(1 - \sigma(T(x)))] \\ &= \arg \min_G \mathbb{E}_{x = G(z), z \sim q(z)} [\log(1 - \sigma(T(x)))]\end{aligned}\quad (4)$$

However, the loss $\log(1 - \sigma(T(x)))$ is not always good for optimization, so we usually use a equivalent loss, such as $-\log \sigma(T(x))$ and $-T(x)$. Namely, we may adjust the loss of generator for a better optimization, rather than playing the original min-max game.

2 Divergence in Dual Space

2.1 Steps to GAN

From the above discussion, we can see that to construct a GAN usually includes three steps:

1. choose a probability divergence;
2. convert it into a dual form;
3. play a min-max game.

But we know that only the last two steps are useful for practice. The first step is only a theoretical concept and is not very important for a GAN. Therefore, a natural thought is: why not analyse the property of divergence and even construct new divergence in dual space directly? Our following content will demonstrate this thought is a very simple approach to build and understand GANs.

2.2 SGAN

We start from the Standard GAN (SGAN, Goodfellow et al. (2014)) as an example to show how we can achieve the goal. From the appendix A.1, we have Lemma 1:

Lemma 1 *The following $\mathcal{D}[p(x), q(x)]$ defines a probability divergence*

$$\mathcal{D}[p(x), q(x)] = \max_T \frac{1}{2} \mathbb{E}_{x \sim p(x)} [\log \sigma(T(x))] + \frac{1}{2} \mathbb{E}_{x \sim q(x)} [\log(1 - \sigma(T(x)))] + \log 2 \quad (5)$$

It is worth to highlight that we prove $\mathcal{D}[p(x), q(x)]$ is a probability divergence¹ in dual space, not need the original definition (1). Getting rid of the original definition of divergence allows us to seek more powerful divergence in dual space.

¹Namely, satisfying the definition 1.

Now we have a divergence $\mathcal{D}[p(x), q(x)]$ defined by Lemma 1, so we can train a generator by minimizing $\mathcal{D}[p(x), q(x)]$, which results in the min-max game (3).

The difficulty arises while there is almost no intersection between $p(x)$ and $q(x)$. For example, we consider $p(x) = \delta(x - \alpha)$, $q(x) = \delta(x - \beta)$ and $\alpha \neq \beta$. Now we have

$$\mathcal{D}[p(x), q(x)] = \max_T \frac{1}{2} \log \sigma(T(\alpha)) + \frac{1}{2} \log(1 - \sigma(T(\beta))) + \log 2 \quad (6)$$

because no constraint on T , we can let $T(\alpha) \rightarrow +\infty$, $T(\beta) \rightarrow -\infty$, to obtain the maximum value of the above formula, that is

$$\mathcal{D}[p(x), q(x)] = \log 2 \quad (7)$$

So if there is almost no intersection between $p(x)$ and $q(x)$, this divergence of them is a constant $\log 2$, whose gradients are zeros. In this situation, generator can not improve via gradient descent method. And we know this situation will happen with very high probability (Arjovsky & Bottou, 2017). Therefore it's hard to train a good generative model under the framework of SGAN.

These conclusions can be popularized to any kind of f -GANs (Nowozin et al., 2016) in parallel, including LSGAN (Mao et al., 2016). And all of them suffer the same difficulty.

2.3 WGAN

We turn to a new kind of divergence by Lemma 2:

Lemma 2 *The following $\mathcal{W}[p(x), q(x)]$ defines a probability divergence*

$$\mathcal{W}[p(x), q(x)] = \max_{T, \|T\|_L \leq 1} \mathbb{E}_{x \sim p(x)}[T(x)] - \mathbb{E}_{x \sim q(x)}[T(x)] \quad (8)$$

here

$$\|T\|_L = \max_{x \neq y} \frac{|T(x) - T(y)|}{d(x, y)} \quad (9)$$

and $d(x, y)$ is any distance metric of x, y . $d(x, y) = \|x - y\|_1$, $d(x, y) = \|x - y\|_2$ is frequently-used.

The proof is in appendix A.2. Interestingly, the proof is very simple compared the corresponding proof of f -divergence.

Now we can play a new min-max game:

$$G, T = \arg \min_G \arg \max_{T, \|T\|_L \leq 1} \mathbb{E}_{x \sim p(x)}[T(x)] - \mathbb{E}_{x \sim G(z), z \sim q(z)}[T(x)] \quad (10)$$

That is what we call WGAN (Arjovsky et al., 2017).

Compared with (5), $\mathcal{W}[p(x), q(x)]$ can reasonably measure the difference of $p(x), q(x)$ while they almost have no intersection. Let us consider $p(x) = \delta(x - \alpha)$, $q(x) = \delta(x - \beta)$ and $\alpha \neq \beta$:

$$\mathcal{W}[p(x), q(x)] = \max_{T, \|T\|_L \leq 1} T(\alpha) - T(\beta) \quad (11)$$

The constraint $\|T\|_L \leq 1$ means $|T(\alpha) - T(\beta)| \leq d(\alpha, \beta)$. So we have

$$\mathcal{W}[p(x), q(x)] = d(\alpha, \beta) \quad (12)$$

The result is not a constant and its gradients are not zeros. So WGAN will not suffer gradient vanishing usually.

2.4 WGAN-GP

The essential problem of WGAN is how to constrain T in $\|T\|_L \leq 1$, which currently has several solutions: weight clipping (WC, Arjovsky et al. (2017)), gradient penalty (GP, Gulrajani et al. (2017)) and spectral normalization (SN, Miyato et al. (2018)).

Weight clipping is always unstable and has been abandoned in most cases. Spectral normalization is a better operation for not only WGANs but also many other GANs, but it constrains T in a tiny subspace of $\|T\|_L \leq 1$, wasting the modeling power of T .

It seems the best approach is gradient penalty now. Gradient penalty replaces $\|T\|_L$ with the norm of gradients $\|\nabla_x T\|$, and implement it as a penalty term:

$$\begin{aligned} T &= \arg \max_T \mathbb{E}_{x \sim p(x)}[T(x)] - \mathbb{E}_{x \sim q(x)}[T(x)] - \lambda \mathbb{E}_{x \sim pq(x)}[(\|\nabla_x T(x)\| - 1)^2] \\ G &= \arg \min_G \mathbb{E}_{x=G(z), z \sim q(z)}[-T(x)] \end{aligned} \quad (13)$$

whose $\lambda > 0$ is a hyperparameter and $pq(x)$ is the random linear interpolation of $p(x)$ and $q(x)$. It is called WGAN-GP.

WGAN-GP works well in many cases but it is just an empirical trick. Recently some researchs reveal some irrationality of WGAN-GP (Wu et al., 2017). Another disadvantage of WGAN-GP is the slow speed, while calculating the exact gradients needs more heavy computation.

3 GAN with Quadratic Potential

3.1 A Quadratic Divergence

From the discussion of SGAN and WGAN, we can see that an ideal divergence should not have any constraints on T (like SGAN) and should give a reasonable measurement of $p(x), q(x)$ while they almost have no intersection (like WGAN).

Here we propose such a divergence:

Lemma 3 *The following $\mathcal{L}[p(x), q(x)]$ defines a probability divergence*

$$\begin{aligned} \mathcal{L}[p(x), q(x)] &= \max_T \mathbb{E}_{(x_r, x_f) \sim p(x_r)q(x_f)} \left[T(x_r, x_f) - T(x_f, x_r) - \frac{(T(x_r, x_f) - T(x_f, x_r))^2}{2\lambda d(x_r, x_f)} \right] \end{aligned} \quad (14)$$

$\lambda > 0$ is a hyperparameter and $d(x_r, x_f)$ is any distance metric of x_r, x_f . $d(x_r, x_f) = \|x_r - x_f\|_1, d(x_r, x_f) = \|x_r - x_f\|_2$ is frequently-used.

The proof is shown in appendix A.3. It is like a conventional divergence plus a quadratic potential, so we call it quadratic potential divergence (QP-div).

Now we check $\mathcal{L}[p(x), q(x)]$ with $p(x) = \delta(x - \alpha), q(x) = \delta(x - \beta)$ to demonstrate QP-div will have a reasonable performance to two extreme distribution:

$$\mathcal{L}[p(x), q(x)] = \max_T T(\alpha, \beta) - T(\beta, \alpha) - \frac{(T(\alpha, \beta) - T(\beta, \alpha))^2}{2\lambda d(\alpha, \beta)} \quad (15)$$

Let $z = T(\alpha, \beta) - T(\beta, \alpha)$, that is only a maximum value problem of quadratic function $z - \frac{z^2}{2\lambda d(\alpha, \beta)}$. We know it is $\frac{1}{2}\lambda d(\alpha, \beta)$, so

$$\mathcal{L}[p(x), q(x)] = \frac{1}{2}\lambda d(\alpha, \beta) \quad (16)$$

We can see that $\mathcal{L}[p(x), q(x)]$ has similar property like $\mathcal{W}[p(x), q(x)]$, but with no constraints on T . It is very friendly to practice.

3.2 From QP-div to GAN-QP

In theory, we can play a min-max game on QP-div to train a generative model

$$\begin{aligned} G, T &= \arg \min_G \arg \max_T \mathbb{E}_{(x_r, x_f) \sim p(x_r)q(x_f)} [\mathcal{L}(x_r, x_f)] \\ \mathcal{L}(x_r, x_f) &= T(x_r, x_f) - T(x_f, x_r) - \frac{(T(x_r, x_f) - T(x_f, x_r))^2}{2\lambda d(x_r, x_f)} \end{aligned} \quad (17)$$

However, $\mathcal{L}(x_r, x_f)$ is not a good loss for generator because there is a $d(x_r, x_f)$ in the denominator. Generator wants to minimize $\mathcal{L}(x_r, x_f)$, which will minimize $d(x_r, x_f)$ correspondingly. And we know any ready-to-use distance may not be used as a perfect metric of two samples.

We find that using $T(x_r, x_f) - T(x_f, x_r)$ as the loss of generator is enough. That results the following generative model called GAN with Quadratic Potential (GAN-QP):

$$\begin{aligned} T &= \arg \max_T \mathbb{E}_{(x_r, x_f) \sim p(x_r)q(x_f)} [\mathcal{L}(x_r, x_f)] \\ G &= \arg \min_G \mathbb{E}_{(x_r, x_f) \sim p(x_r)q(x_f)} [T(x_r, x_f) - T(x_f, x_r)] \end{aligned} \quad (18)$$

Futher discussion can be found in appendix *B*.

3.3 BiGAN-QP

Like BiGAN (Donahue et al., 2016) or ALI (Dumoulin et al., 2016), we can add an inference model $E(x)$ into GAN-QP. Only need to replace x_r with $(x_r, E(x_r))$ and replace x_f with $(G(z), z)$, we have

$$\begin{aligned} T &= \arg \max_T \mathbb{E}_{x \sim p(x), z \sim q(z)} \left[\Delta T - \frac{\Delta T^2}{2\lambda d(x, E(x); G(z), z)} \right] \\ G &= \arg \min_G \mathbb{E}_{x \sim p(x), z \sim q(z)} [\Delta T] \\ \Delta T &= T(x, E(x); G(z), z) - T(G(z), z; x, E(x)) \end{aligned} \quad (19)$$

That is what we called BiGAN-QP. In theory, (19) is enough for training the whole model. But actually it is hard to converge. We need to add some *guiding term* to guide the training process. We can use reconstruction mse loss as the guiding term:

$$\begin{aligned} T &= \arg \max_T \mathbb{E}_{x \sim p(x), z \sim q(z)} \left[\Delta T - \frac{\Delta T^2}{2\lambda d(x, E(x); G(z), z)} \right] \\ G &= \arg \min_G \mathbb{E}_{x \sim p(x), z \sim q(z)} \left[\Delta T + \beta_1 \|z - E(G(z))\|^2 + \beta_2 \|x - G(E(x))\|^2 \right] \\ \Delta T &= T(x, E(x); G(z), z) - T(G(z), z; x, E(x)) \end{aligned} \quad (20)$$

In practice, we find the following altering can improve quality of reconstruction:

$$\begin{aligned} T &= \arg \max_T \mathbb{E}_{x \sim p(x), z \sim q(z)} \left[\Delta T - \frac{\Delta T^2}{2\lambda d(x, E(x); G(z), z)} \right] \\ G &= \arg \min_G \mathbb{E}_{x \sim p(x), z \sim q(z)} \left[\Delta T + \beta_1 \|z - E(G_{ng}(z))\|^2 + \beta_2 \|x - G(E_{ng}(x))\|^2 \right] \end{aligned} \quad (21)$$

where G_{ng} and E_{ng} means we stop its gradient at its output.

4 Experiments

4.1 Experimental Details

Our experiments are mainly conducted on CelebA HQ dataset (Karras et al., 2017). Our basic setup follows DCGANs (Radford et al., 2015), and is implemented in Keras (Chollet et al., 2015), and available in my repository². We use the Adam optimizer (Kingma & Ba, 2014), with a constant learning rate of 2×10^{-4} and $\beta_1 = 0.5, \beta_2 = 0.999$ in both T and G . We train GAN-QP with two D steps per G step.

In (18), discriminator T is a model with both real sample x_r and fake sample x_f as inputs. But in our experiments, we find just using one sample as input has generated good performance. In other words, $T(x_r, x_f) \equiv T(x_r)$ is enough. We try the architecture like $T(x_r, x_f) = D([E(x_r), E(x_f)])$ but there is no obvious improvement. So the final loss we use is

$$\begin{aligned} T &= \arg \max_T \mathbb{E}_{(x_r, x_f) \sim p(x_r)q(x_f)} \left[T(x_r) - T(x_f) - \frac{(T(x_r) - T(x_f))^2}{2\lambda d(x_r, x_f)} \right] \\ G &= \arg \min_G \mathbb{E}_{(x_r, x_f) \sim p(x_r)q(x_f)} [T(x_r) - T(x_f)] \end{aligned} \quad (22)$$

²<https://github.com/bojone/gan-qp>

Hyperparameter λ is

$$\lambda = \begin{cases} \frac{10}{w \times h \times ch}, & \text{while } d(x_r, x_f) = \|x_r - x_f\|_1 \\ \frac{10}{\sqrt{w \times h \times ch}}, & \text{while } d(x_r, x_f) = \|x_r - x_f\|_2 \end{cases} \quad (23)$$

w, h, ch is the width, height and the number of channels of the input images. We test both L1 norm $d(x_r, x_f) = \|x_r - x_f\|_1$ and L2 norm $d(x_r, x_f) = \|x_r - x_f\|_2$ in our experiments, but they have no significant statistical difference.

The quantitative index we use to evaluate a GAN is Frechet Inception Distance (FID, Heusel et al. (2017)). We also re-implement it in Keras.

4.2 Basic Comparison

Firstly, we compared GAN-QP with WGAN-GP, WGAN-SN (WGAN with spectral normalization), SGAN-SN (SGAN with spectral normalization), LSGAN-SN (LSGAN with spectral normalization) on 128x128 resolution. λ in WGAN-GP (13) we use is 10. Then Batch Normalization is removed from discriminator of WGAN-GP and other hyperparameters are as the same as GAN-QP. Each experiment is repeated twice for obtaining more reliable conclusion. The comparison is shown in Figure 1 and Table 1.

We can see that there is no obvious difference between GAN-QP-L1 and GAN-QP-L2, which means GAN-QP is robust to distance metric. The best two results come from GAN-QP and WGAN-SN. The worst is WGAN-GP. Generally, the FID curve of WGAN-SN and SGAN-SN is more smoother and GAN-QP is more shaking. But GAN-QP keeps the best performance as same as SGAN-SN.

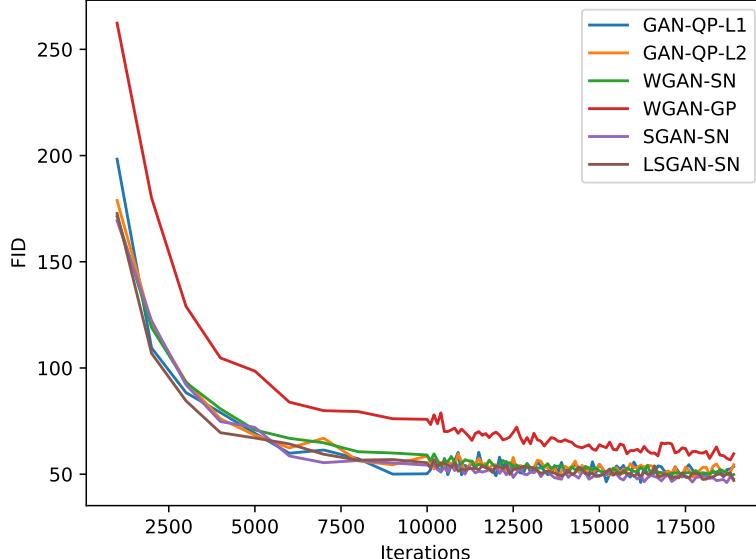


Figure 1: FID comparison of GAN-QP with L1/L2 distance and WGAN-GP,WGAN-SN,SGAN-SN,LSGAN-SN. WGAN-GP is generally worse than others. The best two is GAN-QP and SGAN-SN.

Table 1: Final performance of GANs on 128x128 resolution.

	GAN-QP-L1 / L2	WGAN-GP	WGAN-SN	SGAN-SN	LSGAN-SN
Best FID	45.0 / 44.7	55.5	47.8	44.5	45.8
Speed	1x / 1x	1.5x	1x	1x	1x

4.3 Higher Resolution

On 128x128 resolution, SGAN-SN and GAN-QP has the same best performance. If we turn to 256x256 resolution, we can see that GAN-QP achieves a better FID than SGAN-SN (Table 2). It even works well on 512x512 resolution (Figure 2).

Table 2: Final performance of GAN-QP and SGAN-SN on 256x256 resolution.

	GAN-QP	SGAN-SN
Best FID	22.7	27.9



Figure 2: Random samples from GAN-QP on 512x512 resolution. The final FID is 26.64. And it costs 2 days to finish training on one gtx 1080ti.

4.4 Reconstruction of BiGAN-QP

We also use a simpler version of (21) to evaluate the performance of BiGAN-QP:

$$\begin{aligned}
 T &= \arg \max_T \mathbb{E}_{x \sim p(x), z \sim q(z)} \left[\Delta T - \frac{\Delta T^2}{2\lambda d(x, E(x); G(z), z)} \right] \\
 G &= \arg \min_G \mathbb{E}_{x \sim p(x), z \sim q(z)} \left[\Delta T + \beta_1 \|z - E(G_{ng}(z))\|^2 + \beta_2 \|x - G(E_{ng}(x))\|^2 \right] \\
 \Delta T &= T(x, E(x)) - T(G(z), z)
 \end{aligned} \tag{24}$$

while $\beta_1 = 4 / \dim_z$ and $\beta_2 = 6 / \dim_x$. The result is shown in Figure 3.

5 Conclusion

In this paper, we demonstrate that we can explore probability divergence directly, which is more convenient and flexible. Starting from this idea, we find out a novel divergence called QP-div, which has excellent characteristics, does not require the 1-Lipschitz constraint and does not require the extract gradient penalty. As a concrete example, we construct a new framework of GAN equipped QP-div: GAN-QP. And the experiments demonstrate the stability and superiority of GAN-QP.



Figure 3: Reconstruction(right) of real images(left) on 256x256 resolution.

References

- Arjovsky, M., & Bottou, L. (2017). Towards principled methods for training generative adversarial networks.
- Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein gan.
- Chollet, F., et al. (2015). *Keras*. <https://keras.io>.
- Donahue, J., Krähenbühl, P., & Darrell, T. (2016). Adversarial feature learning.
- Dumoulin, V., Belghazi, I., Poole, B., Lamb, A., Arjovsky, M., Mastropietro, O., & Courville, A. (2016). Adversarially learned inference.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative adversarial networks. *Advances in Neural Information Processing Systems*, 3, 2672-2680.
- Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). Improved training of wasserstein gans.
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium.
- Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *Computer Science*.
- Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., & Smolley, S. P. (2016). Least squares generative adversarial networks.
- Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). Spectral normalization for generative adversarial networks.
- Nowozin, S., Cseke, B., & Tomioka, R. (2016). f-gan: Training generative neural samplers using variational divergence minimization.
- Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *Computer Science*.
- Wu, J., Huang, Z., Thoma, J., Acharya, D., & Van Gool, L. (2017). Wasserstein divergence for gans.

Supplemental Materials

A Detailed Derivation

A.1 (5) is a divergence

Firstly, it is trivial to see that $\mathcal{D}[p(x), q(x)]$ is nonnegative since we can always let $T(x) \equiv 0$:

$$\begin{aligned}\mathcal{D}[p(x), q(x)] &= \max_T \frac{1}{2} \mathbb{E}_{x \sim p(x)} [\log \sigma(T(x))] + \frac{1}{2} \mathbb{E}_{x \sim q(x)} [\log(1 - \sigma(T(x)))] + \log 2 \\ &\geq \frac{1}{2} \mathbb{E}_{x \sim p(x)} [\log \sigma(0)] + \frac{1}{2} \mathbb{E}_{x \sim q(x)} [\log(1 - \sigma(0))] + \log 2 \\ &= 0\end{aligned}\tag{25}$$

Next, we show $\mathcal{D}[p(x), p(x)] = 0$, which is also simple:

$$\begin{aligned}\mathcal{D}[p(x), p(x)] &= \max_T \frac{1}{2} \mathbb{E}_{x \sim p(x)} [\log \sigma(T(x)) + \log(1 - \sigma(T(x)))] + \log 2 \\ &= \max_T \frac{1}{2} \mathbb{E}_{x \sim p(x)} [\log (\sigma(T(x))(1 - \sigma(T(x))))] + \log 2\end{aligned}\tag{26}$$

It is not difficult to prove that the maximum value of $\sigma(z)(1 - \sigma(z))$ is $\frac{1}{4}$ at $z = 0$, so we have

$$\mathcal{D}[p(x), p(x)] = \frac{1}{2} \mathbb{E}_{x \sim p(x)} [\log \frac{1}{4}] + \log 2 = 0\tag{27}$$

Finally, we show $\mathcal{D}[p(x), q(x)] > 0$ if $p(x) \neq q(x)$.³ Let

$$\sigma(T(x)) = \frac{p(x)}{p(x) + q(x)} = \lambda(x)\tag{28}$$

we have

$$\begin{aligned}\mathcal{D}[p(x), q(x)] &= \frac{1}{2} \int \left[p(x) \log \frac{p(x)}{p(x) + q(x)} + q(x) \log \frac{q(x)}{p(x) + q(x)} \right] dx + \log 2 \\ &= \int \left(\frac{p(x) + q(x)}{2} \right) \left[\lambda(x) \log \lambda(x) + (1 - \lambda(x)) \log (1 - \lambda(x)) \right] dx + \log 2\end{aligned}\tag{29}$$

Because of $p(x) \neq q(x)$, $\lambda(x) \not\equiv 1/2$, and we know $\lambda \log \lambda + (1 - \lambda) \log(1 - \lambda) < -\log 2$ if $\lambda \neq \frac{1}{2}$. Therefore,

$$\mathcal{D}[p(x), q(x)] > \int \left(\frac{p(x) + q(x)}{2} \right) (-\log 2) dx + \log 2 = 0\tag{30}$$

A.2 (8) is a divergence

Firstly, it is trivial to see that $\mathcal{W}[p(x), q(x)]$ is nonnegative since we can always let $T(x) \equiv 0$:

$$\mathcal{W}[p(x), q(x)] \geq \mathbb{E}_{x \sim p(x)}[0] - \mathbb{E}_{x \sim q(x)}[0] = 0\tag{31}$$

Next, $\mathcal{W}[p(x), p(x)] = 0$ is also trivial. So we only need to show $\mathcal{W}[p(x), q(x)] > 0$ if $p(x) \neq q(x)$. It is actually not difficult because we only need to let $T_0(x) = \text{sign}(p(x) - q(x))$:

$$\mathbb{E}_{x \sim p(x)}[T_0(x)] - \mathbb{E}_{x \sim q(x)}[T_0(x)] = \int (p(x) - q(x)) \cdot \text{sign}(p(x) - q(x)) dx > 0\tag{32}$$

That means $\mathcal{W}[p(x), p(x)] > 0$.

³Strictly, $p(x) \neq q(x)$ is not enough. The sufficient condition is $\int_{\{x | p(x) \neq q(x)\}} dx > 0$.

A.3 (14) is a divergence

Firstly, it is trivial to see that $\mathcal{W}[p(x), q(x)]$ is nonnegative since we can always let $T(x_r, x_f) \equiv 0$:

$$\mathcal{L}[p(x), q(x)] \geq \mathbb{E}_{(x_r, x_f) \sim p(x_r)q(x_f)} \left[0 - \frac{0^2}{2\lambda d(x, y)} \right] = 0 \quad (33)$$

Next, we have

$$\mathcal{L}[p(x), p(x)] = \max_T \mathbb{E}_{(x_r, x_f) \sim p(x_r)p(x_f)} \left[-\frac{(T(x_r, x_f) - T(x_f, x_r))^2}{2\lambda d(x, y)} \right] \quad (34)$$

Obviously, the maximum value is zero. So $\mathcal{L}[p(x), p(x)] = 0$.

Finally, if $p(x) \neq q(x)$, we let

$$T_0(x_r, x_f) = \text{sign}(p(x_r)q(x_f) - p(x_f)q(x_r)) \quad (35)$$

now we have

$$\begin{aligned} \gamma_1 &= \mathbb{E}_{(x_r, x_f) \sim p(x_r)q(x_f)} [T_0(x_r, x_f) - T_0(x_f, x_r)] \\ &= \iint p(x_r)q(x_f) [T_0(x_r, x_f) - T_0(x_f, x_r)] dx_r dx_f \\ &= \iint [p(x_r)q(x_f) - p(x_f)q(x_r)] T_0(x_r, x_f) dx_r dx_f \\ &= \iint [p(x_r)q(x_f) - p(x_f)q(x_r)] \cdot \text{sign}(p(x_r)q(x_f) - p(x_f)q(x_r)) dx_r dx_f > 0 \\ \gamma_2 &= \mathbb{E}_{(x_r, x_f) \sim p(x_r)q(x_f)} \left[\frac{(T_0(x_r, x_f) - T_0(x_f, x_r))^2}{2\lambda d(x, y)} \right] \geq 0 \end{aligned} \quad (36)$$

If $\gamma_1 > \gamma_2$, then

$$\begin{aligned} &\mathbb{E}_{(x_r, x_f) \sim p(x_r)q(x_f)} \left[T(x_r, x_f) - T(x_f, x_r) - \frac{(T(x_r, x_f) - T(x_f, x_r))^2}{2\lambda d(x, y)} \right] \\ &= \gamma_1 - \gamma_2 > 0 \end{aligned} \quad (37)$$

else if $\gamma_1 \leq \gamma_2$, we can define

$$T(x_r, x_f) = \frac{\gamma_1}{2\gamma_2} \cdot T_0(x_r, x_f) \quad (38)$$

then

$$\begin{aligned} &\mathbb{E}_{(x_r, x_f) \sim p(x_r)q(x_f)} \left[T(x_r, x_f) - T(x_f, x_r) - \frac{(T(x_r, x_f) - T(x_f, x_r))^2}{2\gamma d(x, y)} \right] \\ &= \left(\frac{\gamma_1}{2\gamma_2} \right) \gamma_1 - \left(\frac{\gamma_1}{2\gamma_2} \right)^2 \gamma_2 = \frac{\gamma_1^2}{4\gamma_2} > 0 \end{aligned} \quad (39)$$

Therefore $\mathcal{L}[p(x), q(x)] > 0$.

B Analyse of GAN-QP

B.1 Optimum Solution of (14)

Lemma 4 *the optimum solution of (14) satisfies*

$$\frac{p(x_r)q(x_f) - p(x_f)q(x_r)}{p(x_r)q(x_f) + p(x_f)q(x_r)} = \frac{T(x_r, x_f) - T(x_f, x_r)}{\lambda d(x_r, x_f)} \quad (40)$$

The proof is the basic application of variational method:

$$\begin{aligned}
& \delta \iint p(x_r)q(x_f) \left[T(x_r, x_f) - T(x_f, x_r) - \frac{(T(x_r, x_f) - T(x_f, x_r))^2}{2\lambda d(x_r, x_f)} \right] dx_r dx_f \\
&= \iint p(x_r)q(x_f) \left[\delta T(x_r, x_f) - \delta T(x_f, x_r) \right. \\
&\quad \left. - \frac{(T(x_r, x_f) - T(x_f, x_r))(\delta T(x_r, x_f) - \delta T(x_f, x_r))}{\lambda d(x_r, x_f)} \right] dx_r dx_f \\
&= \iint \left[p(x_r)q(x_f) - p(x_f)q(x_r) \right. \\
&\quad \left. - \left(p(x_r)q(x_f) + p(x_f)q(x_r) \right) \frac{T(x_r, x_f) - T(x_f, x_r)}{\lambda d(x_r, x_f)} \right] \delta T(x_r, x_f) dx_r dx_f
\end{aligned} \tag{41}$$

The formula in square brackets must be identically equal to zero. Therefore

$$\frac{p(x_r)q(x_f) - p(x_f)q(x_r)}{p(x_r)q(x_f) + p(x_f)q(x_r)} = \frac{T(x_r, x_f) - T(x_f, x_r)}{\lambda d(x_r, x_f)} \tag{42}$$

B.2 Adaptive Lipschitz Constraint

From (40), it is not difficult to prove that the optimum $T(x_r, x_f)$ satisfies

$$-1 \leq \frac{T(x_r, x_f) - T(x_f, x_r)}{\lambda d(x_r, x_f)} \leq 1 \tag{43}$$

In other words, the optimum $T(x_r, x_f)$ satisfies Lipschitz constraint automatically. Therefore we can say (14) is a divergence with adaptive Lipschitz constraint.

B.3 The Divergence of Generator

We use $T(x_r, x_f) - T(x_f, x_r)$ rather than the whole $\mathcal{L}(x_r, x_f)$ as the loss of generator in (18). And we have solved the optimum solution of (14) in Lemma 4. Then we can see the ultimate goal of generator to minimize is

$$\lambda \iint p(x_r)q(x_f) \frac{p(x_r)q(x_f) - p(x_f)q(x_r)}{p(x_r)q(x_f) + p(x_f)q(x_r)} d(x_r, x_f) dx_r dx_f \tag{44}$$

Now we have Lemma 5:

Lemma 5

$$\tilde{\mathcal{L}}[p(x), q(x)] = \iint p(x_r)q(x_f) \frac{p(x_r)q(x_f) - p(x_f)q(x_r)}{p(x_r)q(x_f) + p(x_f)q(x_r)} d(x_r, x_f) dx_r dx_f \tag{45}$$

is also a probability divergence of $p(x), q(x)$.

Actually Lemma 5 is a conclusion of Cauchy–Schwarz inequality. Firstly we let

$$\mu(x_r, x_f) = \frac{d(x_r, x_f)}{p(x_r)q(x_f) + p(x_f)q(x_r)} > 0 \tag{46}$$

Then by Cauchy–Schwarz inequality we have

$$\begin{aligned}
& \left(\iint \mu(x_r, x_f) p^2(x_r) q^2(x_f) dx_r dx_f \right)^2 \\
&= \left(\iint \left(\sqrt{\mu(x_r, x_f)} p(x_r) q(x_f) \right)^2 dx_r dx_f \right) \left(\iint \left(\sqrt{\mu(x_f, x_r)} p(x_f) q(x_r) \right)^2 dx_f dx_r \right) \\
&\geq \left(\iint \mu(x_r, x_f) p(x_r) q(x_f) p(x_f) q(x_r) dx_r dx_f \right)^2
\end{aligned} \tag{47}$$

So

$$\tilde{\mathcal{L}}[p(x), q(x)] = \iint \mu(x_r, x_f) p(x_r) q(x_f) \left(p(x_r) q(x_f) - p(x_f) q(x_r) \right) \mu(x_r, x_f) dx_r dx_f \geq 0 \quad (48)$$

Two sides are equal if and only if $\sqrt{\mu(x_r, x_f)} p(x_r) q(x_f) \equiv \sqrt{\mu(x_f, x_r)} p(x_f) q(x_r)$, which means $p(x) \equiv q(x)$. Therefore (45) is a probability divergence, which means to lower $T(x_r, x_f) - T(x_f, x_r)$ is actually to lower the difference between $p(x)$ and $q(x)$. The divergence is weighted by $d(x_r, x_f)$, forcing generator to focus on the sample pairs of larger distance, which is in line with our intuition.

B.4 Performance while No Intersection

We have shown the QP-div in Lemma 3 also works well while there is no intersection between $p(x)$ and $q(x)$. But now we use $T(x_r, x_f) - T(x_f, x_r)$ as the loss of generator, corresponding to the new divergence $\tilde{\mathcal{L}}[p(x), q(x)]$. Therefore we have to check the performance of $\tilde{\mathcal{L}}[p(x), q(x)]$ with $p(x) = \delta(x - \alpha)$, $q(x) = \delta(x - \beta)$. That is very easy:

$$\begin{aligned} & \tilde{\mathcal{L}}[\delta(x - \alpha), \delta(x - \beta)] \\ &= \iint \delta(x_r - \alpha) \delta(x_f - \beta) \frac{\delta(x_r - \alpha) \delta(x_f - \beta) - \delta(x_f - \alpha) \delta(x_r - \beta)}{\delta(x_r - \alpha) \delta(x_f - \beta) + \delta(x_f - \alpha) \delta(x_r - \beta)} d(x_r, x_f) dx_r dx_f \quad (49) \\ &= \frac{\delta(0)\delta(0) - \delta(\beta - \alpha)\delta(\alpha - \beta)}{\delta(0)\delta(0) + \delta(\beta - \alpha)\delta(\alpha - \beta)} d(\alpha, \beta) \end{aligned}$$

We know $\delta(0) \rightarrow \infty$ and $\delta(\alpha - \beta) = 0$ for $\alpha \neq \beta$, so the result is $d(\alpha, \beta)$, a reasonable measurement actually.

B.5 Robustness of λ

(44) showed that λ is just a scalar for $\tilde{\mathcal{L}}[p(x), q(x)]$. That means GAN-QP is insensitive to the hyperparameter λ , which is different from WGAN-GP. We only need to choose a suitable λ to make the loss more readable (not very large and not very small).

C Future Work

C.1 A Conjecture

Inspired by the form of QP-div (14), it may be extended as

Conjecture 1 *If*

$$\arg \max_T \mathbb{E}_{(x_r, x_f) \sim p(x_r) q(x_f)} [f(T(x_r, x_f))] \quad (50)$$

is a probability divergence of $p(x)$ and $q(x)$, so does

$$\arg \max_T \mathbb{E}_{(x_r, x_f) \sim p(x_r) q(x_f)} \left[f(T(x_r, x_f)) - \frac{f^2(T(x_r, x_f))}{2\lambda d(x_r, x_f)} \right] \quad (51)$$

for some λ .

Conjecture 1 means we can use

$$\frac{f^2(T(x_r, x_f))}{2\lambda d(x_r, x_f)} \quad (52)$$

as a penalty term for any other GAN's discriminator to enhance the original GAN.

C.2 Example

For example, we can enhance SGAN (3), (5) with a quadratic potential term:

$$\begin{aligned}
 T &= \arg \max_T \mathbb{E}_{(x_r, x_f) \sim p(x_r)q(x_f)} \left[f(x_r, x_f) - \frac{f^2(T(x_r, x_f))}{2\lambda d(x_r, x_f)} \right] \\
 G &= \arg \min_G \mathbb{E}_{x_r \sim p(x_r), x_f = G(z), z \sim q(z)} [f(x_r, x_f)] \\
 f(x_r, x_f) &= \frac{1}{2} \log \sigma(T(x_r)) + \frac{1}{2} \log(1 - \sigma(T(x_f))) + \log 2
 \end{aligned} \tag{53}$$

It has been validated preliminarily by our experiments. But it is still yield to be proven strictly.