Wasserstein GAN with Quadratic Transport Cost

Huidong Liu, Xianfeng Gu, Dimitris Samaras
Stony Brook University
Stony Brook, NY 11794, USA
{huidliu, gu, samaras}@cs.stonybrook.edu

Abstract

Wasserstein GANs are increasingly used in Computer Vision applications as they are easier to train. Previous WGAN variants mainly use the l_1 transport cost to compute the Wasserstein distance between the real and synthetic data distributions. The l_1 transport cost restricts the discriminator to be 1-Lipschitz. However, WGANs with l_1 transport cost were recently shown to not always converge. In this paper, we propose WGAN-QC, a WGAN with quadratic transport cost. Based on the quadratic transport cost, we propose an Optimal Transport Regularizer (OTR) to stabilize the training process of WGAN-QC. We prove that the objective of the discriminator during each generator update computes the exact quadratic Wasserstein distance between real and synthetic data distributions. We also prove that WGAN-QC converges to a local equilibrium point with finite discriminator updates per generator update. We show experimentally on a Dirac distribution that WGAN-QC converges, when many of the l_1 cost WGANs fail to [22]. Qualitative and quantitative results on the CelebA, CelebA-HQ, LSUN and the ImageNet dog datasets show that WGAN-QC is better than state-of-art GAN methods. WGAN-QC has much faster runtime than other WGAN variants.

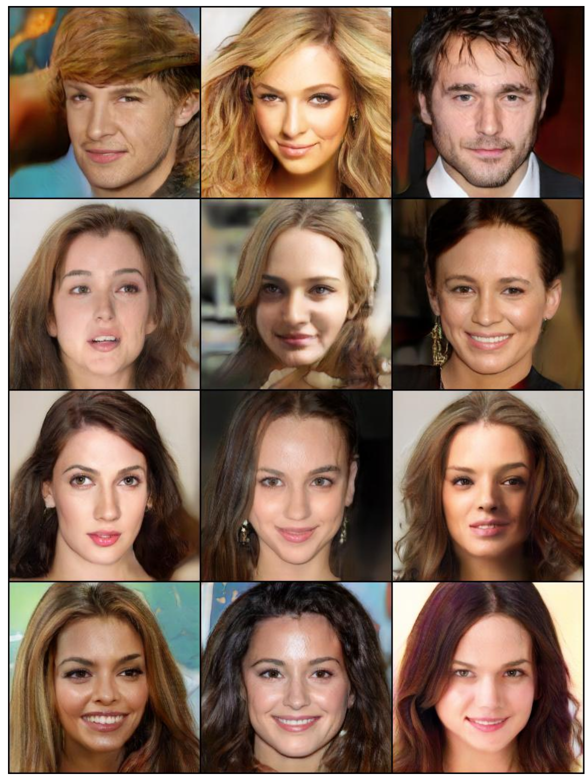


Figure 1. Faces of size 256×256 randomly generated by WGAN-QC on CelebA-HQ. (Best seen in color)

1. Introduction

Generative Adversarial Networks (GANs) [11] successfully model data distributions, and have been used in many vision applications such as image synthesis [20, 34, 28], image inpainting [39, 40], semantic segmentation [15, 26], etc.

While widely used, GANs are known to be hard to train. GANs need to solve a min-max saddle point optimization problem [1]. Due to the competition between the discriminator and the generator, it is difficult to train a GAN to consistently produce meaningful images. Hence, a number of authors have attempted to stabilize GAN training [3, 29, 12, 24, 14]. The Boundary Equilibrium GAN (BEGAN) [3] adopts Proportional Control Theory to balance the training between generator and discriminator. [29] proposes strategies to regularize the gradient of the discriminator, leading to more stable GAN training. The Wasserstein GAN family [2, 12, 19, 24, 13, 10, 9] employs the Wasserstein distance to measure the distance between the distributions of real and synthetic data. The Wasserstein distance guarantees that even if there is no support between the real and generated data distributions, the discriminator still provides gradients to the generator, unlike the Jensen-Shannon (JS) divergence used in the original GAN objective [2].

It is still unclear whether GANs converge. Recent work [25, 23, 22] has shown that analyzing the Jacobian of the gradient field of the GAN parameters near the equilibrium point provides insights into the convergence properties of GAN training. Specifically, for gradient descent algorithms, if all the eigenvalues of the Jacobian have positive real part near the equilibrium point, then the GAN training algorithm will converge given a small enough learning rate. Previous Wasserstein GAN variants mainly use the l_1 transport cost, because the discriminator can be restricted to be 1-Lipschitz [2, 12, 24] so that it can be used to approximate the Wasserstein distance between real and synthetic data distributions. The Sliced WGAN (SWGAN) [8] used the quadratic transport cost and computed the sliced Wasserstein distance [5] between real and synthetic data distributions. The 0-centered gradient penalty methods [22, 35] are proposed to stabilize GAN training, and are shown to be local convergent. However, a large-scale study [7] concluded that the regularization term in [22] actually leads to significant drop of Inception Scores (IS) [31] using the suggested regularization parameter. The regularization term trades off training stability with generated image quality.

In this paper, we improve the stability of GAN training, and, at the same time, optimize the discriminator such that the optimal discriminator can be used to compute the exact quadratic Wasserstein distance [33, 8], by using the quadratic transport cost in WGANs. WGANs with l_1 transport cost can be solved by a Two-Step method [19]. However, generalizing from l_1 to quadratic transport cost is non-trivial, because the quadratic transport cost does not satisfy the triangle inequality condition [19]. Note that the quadratic transport cost is also used in SWGAN [8], but 1) convergence is not guaranteed in [8], 2) the sliced Wasserstein Distance is a different metric from the Wasserstein distance [5], 3) SWGAN computes the generator from the primal form of Optimal Transport (OT) [36], whereas our proposed method computes the discriminator from the dual form of OT.

In summary, our main contributions are:

- We propose WGAN-QC, a new Wasserstein GAN with quadratic transport cost. In WGAN-QC, we propose a modified two-step computation to optimize the discriminator during each generator update.

- We propose the novel Optimal Transport Regularizer (OTR), based on the quadratic transport cost, to stabilize the training process of WGAN-QC. We prove that the objective of the discriminator computes the exact quadratic Wasserstein distance during each generator update.

- We prove that WGAN-QC can converge to a local equilibrium point given a small enough learning rate.

- We show that, contrary to many l_1 cost WGANs, WGAN-QC converges to the real data distribution in the 1-d Dirac distribution example. Qualitative and quantitative results on the CelebA [21], CelebA-HQ [17], LSUN bedroom [38], and the ImageNet dog [30] datasets show that WGAN-QC is better than state-of-the-art GAN methods.

- We show that WGAN-QC is 3.5x and 1.8x faster than WGAN-div which is faster than WGAN-GP on the CelebA and LSUN bedroom datasets, respectively.

We show some randomly generated face images in Fig. 1. These images look very realistic.

2. Optimal Transport

Since our framework is based on the Optimal Transport (OT), we shall briefly review the definition of OT in the Monge-Kantorovich dual formulation [36, 27].

The Monge-Kantorovich dual problem is given below:

Problem 1. *Given two bounded domains X and Y and their probability measures $\nu \in \mathbb{P}(X)$, $\mu \in \mathbb{P}(Y)$, respectively, find functions ϕ and ψ to solve*

$$C(\mu,\nu) = \sup_{\phi-\psi\leq c}\left\{\int \phi(y)d\mu(y) - \int \psi(x)d\nu(x)\right\} \quad (1)$$

where $c : X \times Y \mapsto [0,+\infty]$ is the transport cost.

In practice, given empirical distributions, we write Problem 1 in the discrete case. Suppose $\hat{X} = \{x_j\}_{j\in\mathcal{J}}$ sampled from ν containing n samples and $\hat{Y} = \{y_i\}_{i\in\mathcal{I}}$ sampled from μ containing m samples, where $\mathcal{I}$ and $\mathcal{J}$ are disjoint index sets. Therefore, each element x_j has a Dirac measure of $1/n$, and y_i has a Dirac measure of $1/m$. Hence, the discrete Monge-Kantorovich dual problem is:

$$\begin{aligned} \max_{\phi,\psi} \quad & \frac{1}{m}\sum_{i\in\mathcal{I}}\phi(y_i) - \frac{1}{n}\sum_{j\in\mathcal{J}}\psi(x_j) \\ \text{s.t.} \quad & \phi(y_i) - \psi(x_j) \leq c(x_j, y_i), \\ & \forall i \in \mathcal{I},\ \forall j \in \mathcal{J} \end{aligned} \quad (2)$$

Kantarovich showed [16] that if the transport cost $c(\cdot,\cdot)$ satisfies the triangle inequality, then ϕ and ψ can be unified into just one function. In WGAN [2], WGAN-GP [12], WGAN-TS [19] etc., the l_1 transport cost is used so that ϕ and ψ are unified in one function and used as the discriminator. However, ϕ and ψ cannot be unified when the quadratic transport cost is applied. The quadratic transport cost is:

$$c(x_j, y_i) = \frac{K}{2}||x_j - y_i||_2^2 \quad (3)$$

where K is any constant positive real number. When Eq. (3) is applied, the optimal objective in Eq. (2) equals to $\frac{K}{2}\cdot W_2^2$, which is the quadratic Wasserstein distance [33, 8].

3. WGAN with Quadratic Transport Cost

It has been recently shown [22] that WGANs with l_1 transport cost do not always converge. In this section, we propose WGAN-QC, a Wasserstein GAN with quadratic transport cost. We show in the next section that WGAN-QC always converges to a local equilibrium point.

3.1. Learning the Discriminator From the Kantorovich Potential

Let D_w be the discriminator and G_θ the generator parameterized by w and θ, respectively. $\mathbb{P}_r$ is the real data distribution and $\mathbb{P}_z$ is a simple distribution (e.g. Gaussian or Uniform). When the discriminator loss in WGAN-QC is optimized, we want the discriminator to compute the exact quadratic Wasserstein distance. We also use the quadratic transport cost of Eq. (3), since it contributes to local convergence of WGAN-QC. In WGAN-QC, K in Eq. (3) is set to $1/d$, where d is the dimensionality of data x_j. We regard $\{y_i\}_{i\in\mathcal{I}}$ as real data and $\{x_j\}_{j\in\mathcal{J}}$ as synthetic data.

When the quadratic transport cost is used, ϕ and ψ in (3) cannot be unified into one function. We need to select either ϕ or ψ as the discriminator in WGAN-QC such that the optimal discriminator can be used to compute the quadratic Wasserstein distance. In fact, we care more about the discriminator's value and gradients on the synthetic samples, because the generator is updated according to their gradients. So, we select ψ to be the discriminator as it is defined on synthetic samples. In fact, Eq. (2) can be solved by linear programming. In this equation, if we substitute $\phi(y_i)$ by H_i and $\psi(x_j)$ by H_j, we denote H_i^* and H_j^* to be the optimal solutions for H_i and H_j, respectively. So, we can regress each $D_w(x_j)$ in the discriminator, to H_j^*. We regress $\frac{1}{m}\sum_{i\in\mathcal{I}} D_w(y_i)$ to $\frac{1}{m}\sum_{i\in\mathcal{I}} H_i^*$ such that the optimal discriminator computes the quadratic Wasserstein distance. Thus, the discriminator provides an ascent direction for generator updates. We regress the discriminator as:

$$\begin{aligned}\min_w \quad & \frac{1}{2}\left(\frac{1}{m}\sum_{i\in\mathcal{I}} D_w(y_i) - \frac{1}{m}\sum_{i\in\mathcal{I}} H_i^*\right)^2 \\ & + \frac{1}{2}\left(\frac{1}{n}\sum_{j\in\mathcal{J}} \left(D_w(x_j) - H_j^*\right)^2\right)\end{aligned} \tag{4}$$

The generator loss is:

$$\min_\theta \quad \mathcal{L}(\theta) = -\frac{1}{n}\sum_{j\in\mathcal{J}} D_w(G_\theta(z_j)) \tag{5}$$

3.2. Optimal Transport Regularization

There could be infinite solutions to Eq. (4). We need to regularize the discriminator. Hence, we introduce the Optimal Transport Regularizer (OTR) to stabilize the training process of WGAN-QC. The empirical optimal transport mapping is computed after the linear programming step:

$$\sigma(j) = \arg\min_{i\in\mathcal{I}} \frac{K}{2}||x_j - y_i||_2^2 + H_j^* - H_i^* \tag{6}$$

Essentially, Eq. (6) tries to find $H_{\sigma(j)}^* - H_j^* = c(x_j, y_{\sigma(j)})$, and Lemma 3.1 in [19] guarantees that for each x_j we can always find a $y_{\sigma(j)}$ such that $H_{\sigma(j)}^* - H_j^* = c(x_j, y_{\sigma(j)})$ [1]. Therefore, x_j minimizes

$$H_{\sigma(j)}^* = \inf_{j\in\mathcal{J}} \{H_j^* + c(x_j, y_{\sigma(j)})\} \tag{7}$$

We use the $D_w(x_j)$ to regress H_j^* and x_j minimizes Eq. (7), thus x_j is a local minimum and the first order derivative of Eq. (7) should be 0 in the continuous case, i.e.

$$\nabla_x D_w(x_j) + K(x_j - y_{\sigma(j)}) = 0 \tag{8}$$

Therefore, we propose the following Optimal Transport Regularizer (OTR) for WGAN-QC:

$$\frac{1}{2\cdot n}\sum_{j\in\mathcal{J}} \left(||\nabla_x D_w(x_j)|| - K||y_{\sigma(j)} - x_j||\right)^2 \tag{9}$$

where $||\cdot||$ is the l_2 norm. Eq. (8) holds only when Eq. (3), the quadratic transport cost, is applied in OT. Thus, OTR is specific to WGAN-QC. Eq. (8) has another explanation. According to Brenier's theorem [6, 18], for an optimal discriminator, if x_j is transformed to $y_{\sigma(j)}$, then Eq. (8) holds[2].

3.3. The Discriminator Loss of WGAN-QC

The complete discriminator loss of WGAN-QC (Algorithm 1) in each generator update step is:

$$\begin{aligned}&\min_w \ \mathcal{L}(w) \\ &= \frac{1}{2}\left(\frac{1}{m}\sum_{i\in\mathcal{I}} D_w(y_i) - \frac{1}{m}\sum_{i\in\mathcal{I}} H_i^*\right)^2 \\ &+ \frac{1}{2}\left(\frac{1}{n}\sum_{j\in\mathcal{J}} \left(D_w(x_j) - H_j^*\right)^2\right) \\ &+ \frac{\gamma}{\sqrt{K}n}\sum_{j\in\mathcal{J}} \left(||\nabla_x D_w(x_j)|| - K||y_{\sigma(j)} - x_j||\right)^2\end{aligned} \tag{10}$$

The coefficient γ in OTR balances the regression and regularization terms. The regularization term is obtained by multiplying (9) by $2\gamma/\sqrt{K}$. We found it is easier to set different values of γ for different image sizes.

Next we show in Theorem 1 that the optimal discriminator D_w^* during each generator update can be used to compute the exact quadratic Wasserstein distance between the real and synthetic data distributions for any $\gamma > 0$.

Theorem 1. *If the discriminator in Eq.(10) has sufficient capacity such that the optimal objective of Eq.(10) is 0, then for any $\gamma > 0$, and any optimal solution D_w^* to Eq.(10),*

$$\frac{1}{m}\sum_{i\in\mathcal{I}} D_w^*(y_i) - \frac{1}{n}\sum_{j\in\mathcal{J}} D_w^*(x_j) \tag{11}$$

is the quadratic Wasserstein distance between $\hat{X}$ and $\hat{Y}$. [3]

[1] See the complete proof in the supplementary material
[2] Please see supplementary material for complete proof
[3] Please refer to supplementary material for the proof.

Algorithm 1 WGAN-QC

Input: Real data Y, batch size m, k_D and γ. Adam parameters, α, β_1, β_2
Output: G_θ, D_w
while θ has not converged **do**
  Sample $\{y_i\}_{i\in\mathcal{I}} \sim \mathbb{P}_r$ from real data.
  Sample $\{z_j\}_{j\in\mathcal{J}} \sim \mathbb{P}_z$ random noise.
  Let $x_j = G_\theta(z_j), \forall j \in \mathcal{J}$.
  Solve the Linear Programming problem in Eq. (2), and obtain H^*.
  $H_t^* \leftarrow H_t^* - (\sum_{k\in\mathcal{I}\cup\mathcal{J}} H_k^*)/(m+n), \forall t \in \mathcal{I}\cup\mathcal{J}$.
  for $t = 0$ **to** k_D **do**
    $g_w \leftarrow$ the gradient of (10).
    $w \leftarrow$ Adam(g_w, w, α, β_1, β_2)
  end for
  $g_\theta \leftarrow \nabla_\theta - \frac{1}{n}\sum_{j\in\mathcal{J}} D_w(G_\theta(z_j))$
  $\theta \leftarrow$ Adam(g_θ, θ, α, β_1, β_2)
end while

4. Convergence Analysis

WGAN with l_1 transport cost cannot always converge [22]. In this section, we analyze the convergence properties of WGAN-QC under finite discriminator iterations per generator iteration. First, we write the loss functions of the discriminator and generator under continuous distributions. Let $H_r(y)$ and $H_s(x)$ be the outputs of the linear programming part. $T : X \mapsto Y$ denotes that x is transported to y using Eq. (6). The loss of the discriminator is then:

$$\begin{aligned} &\min_w \ \mathcal{L}_D(w,\theta) \\ &= \frac{1}{2}\left(\mathbb{E}_{\mathbb{P}_r(y)}[D_w(y)] - \mathbb{E}_{\mathbb{P}_r(y)}[H_r(y)]\right)^2 \\ &+ \frac{1}{2}\mathbb{E}_{\mathbb{P}_s(x)}[(D_w(x) - H_s(x))^2] \\ &+ \frac{\lambda}{2}\mathbb{E}_{\mathbb{P}_s(x)}[\left(||\nabla_x D_w(x)|| - K||y_{T(x)} - x||\right)^2] \end{aligned} \tag{12}$$

where $\lambda = 2\gamma/\sqrt{K}$, $\mathbb{P}_s(x)$ denotes the probability of synthetic data, $\mathbb{P}_r(y)$ denotes the probability of real data, and $\mathbb{E}.[\cdot]$ denote expectation. The loss of the generator is

$$\min_\theta \quad \mathcal{L}_G(w,\theta) = -\mathbb{E}_{\mathbb{P}_z(z)}[D_w(G_\theta(z))] \tag{13}$$

In order to analyze the local convergence of WGAN-QC, we analyze the Jacobian of the gradient field of WGAN-QC. For simultaneous gradient descent the gradient field is

$$g(w,\theta) = \begin{pmatrix} \nabla_w \mathcal{L}_D(w,\theta) \\ \nabla_\theta \mathcal{L}_G(w,\theta) \end{pmatrix} \tag{14}$$

The gradient update operator is expressed as:

$$U(w,\theta) = \begin{pmatrix} w - \alpha\nabla_w \mathcal{L}_D(w,\theta) \\ \theta - \alpha\nabla_\theta \mathcal{L}_G(w,\theta) \end{pmatrix} \tag{15}$$

where α is the learning rate. The Jacobian of the gradient field is expressed as:

$$g'(w,\theta) = \begin{pmatrix} \nabla_w^2 \mathcal{L}_D(w,\theta) & \nabla_{w,\theta}^2 \mathcal{L}_G(w,\theta) \\ \nabla_{\theta,w}^2 \mathcal{L}_D(w,\theta) & \nabla_\theta^2 \mathcal{L}_G(w,\theta) \end{pmatrix} \tag{16}$$

We define $\mathcal{M}_G$ and $\mathcal{M}_D$ as the solution spaces for G and D respectively:

$$\begin{aligned} &\mathcal{M}_G := \{\theta | \mathbb{P}_s(G_\theta(z)) = \mathbb{P}_r(y)\}, \\ &\mathcal{M}_D := \{w | \mathcal{L}_D(w,\theta^*) = 0, \ \theta^* \in \mathcal{M}_G\} \end{aligned} \tag{17}$$

(w^*, θ^*) is an equilibrium point if $w^* \in \mathcal{M}_D$ and $\theta^* \in \mathcal{M}_G$. We define:

$$r(w) = \mathbb{E}_{\mathbb{P}_r(y)}[|D_w(y)|^2 + ||\nabla_y D_w(y)||_2^2] \tag{18}$$

From Eq. (12), we have $D_{w^*}(y) = 0$ and $\nabla_y D_{w^*}(y) = 0$, and thus $r(w^*) = 0$.

In order to analyze the convergence of our algorithm, we need two assumptions.

Assumption 1. *We assume that the generator G has sufficient expressive power that $\mathbb{P}_s(G_{\theta^*}(z)) = \mathbb{P}_r(y)$.*

Assumption 2. *If (w, θ) is not the equilibrium point, then $\partial_w^2 r(w^*) \neq 0$.*

Assumption 1 is the feasibility assumption. Assumption 2 means that near the equilibrium point, the discriminator geometry is described by the second order derivative of r [4].

The second order derivative of OTR in WGAN-QC is:[5].

Lemma 1. *The second order derivative of the regularization term*

$$\frac{\lambda}{2}\mathbb{E}_{\mathbb{P}_s(x)}[\left(||\nabla_x D_w(x)|| - K||y_{T(x)} - x||\right)^2] \tag{19}$$

with respect to (w, θ) at the equilibrium point is given by:

$$M_R = \lambda \cdot \mathbb{E}_{\mathbb{P}_s(x)}[\nabla_{w,x} D_{w^*}(x) \nabla_{w,x} D_{w^*}(x)^T] \tag{20}$$

Next, we give the Jacobian of the gradient field $g(w,\theta)$.

Lemma 2. *The Jacobian of the gradient field g(w, θ) at the equilibrium point (w^*, θ^*) is given by:*

$$g'(w^*,\theta^*) = \begin{pmatrix} M_{DD} + M_R & M_{GD} \\ 0 & 0 \end{pmatrix} \tag{21}$$

where M_R is defined in Lemma 1,

$$\begin{aligned} &M_{DD} = \\ &+ \mathbb{E}_{\mathbb{P}_r(y)}[\nabla_w D_{w^*}(y)] \cdot \mathbb{E}_{\mathbb{P}_r(y)}[\nabla_w D_{w^*}(y)^\intercal] \\ &+ \mathbb{E}_{\mathbb{P}_r(y)}[\nabla_w D_{w^*}(y) \nabla_w D_{w^*}(y)^\intercal], \end{aligned} \tag{22}$$

$$M_{GD} = -\mathbb{E}_{\mathbb{P}_s(x)}[\nabla_{w,x}^2 D_{w^*}(x) \nabla_\theta G_{\theta^*}(z)^\intercal] \tag{23}$$

and $M_{DD} + M_R$ is positive definite.

[4]The second assumption is the same as Assumption III (i) in [22]

[5]Please refer to supplementary for proofs of Lemmas 1-3

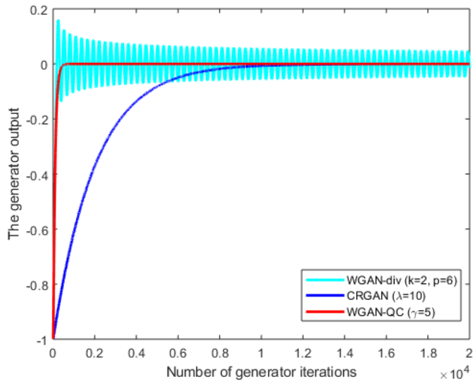


Figure 2. The generator output curve. The true distribution is a Dirac distribution at 0. WGAN-QC and CRGAN generate the true distribution whereas WGAN-div oscillates in this experiment.

Since $M_{DD} + M_R$ is positive definite, we know that it has the same eigenvalues as $g'(w^*, \theta^*)$. As the Jacobian of the gradient field of the discriminator is positive definite, the discriminator can converge to w^* [4]. Furthermore,

Lemma 3. *For simultaneous gradient updates of (w, θ) in WGAN-QC using Eq. (15), if $w = w^*$, then $\theta = \theta^*$.*

Lemma 3 shows that for simultaneous gradient descent, if the discriminator converges, then the generator converges. However, in WGAN-QC, we employ an alternating gradient descent algorithm. Therefore, we show in Theorem 2 that WGAN-QC converges to a local equilibrium point.

Theorem 2. *Suppose Assumptions 1 and 2 are satisfied, then for small enough learning rate α, there exists λ such that WGAN-QC converges to a local equilibrium point.* [6]

5. Experiments

We first study the convergence of WGAN-QC, Critic Regularization GAN (CRGAN) [22] and WGAN-div [37] on a Dirac distribution. Then, we also compare WGAN-QC with state-of-the-art GANs, PGGAN [17], WGAN-GP [12], SWGAN [8], OT-GAN [32], and BigGAN [7] on the CelebA, CelebA-HQ, LSUN bedroom and the ImageNet dog datasets. We use the default published parameters for each method. Architecture details and other experimental settings are in supplementary material.

Hyperparameter Study WGAN-QC has a hyperparameter γ. We investigate the FID scores on the CelebA dataset w.r.t. γ in Table 1. When $\gamma = 0.1$ and $\gamma = 1$, WGAN-QC achieves the best performance. Therefore, we suggest tuning γ in $[0.01, 1.0]$ for WGAN-QC.

[6] Please refer to supplementary material for proof of Theorem 2.

γ	10^{-4}	10^{-3}	10^{-2}	10^{-1}	10^{0}	10^{1}
FID	23.4	20.2	14.4	**12.9**	**11.5**	15.7

Table 1. WGAN-QC achieves the lowest FIDs at $\gamma = 0.1$ and 1. So, we suggest tuning γ in $[0.01, 1]$ on other datasets.

Method	CelebA	CelebA-HQ	LSUN
DCGAN	52.0	-	61.1
PGGAN	16.3	14.1	17.8
SWGAN	23.2	-	52.9
WGAN-GP	18.4	-	26.8
WGAN-div	15.2	13.5	15.9
WGAN-QC	**12.9**	**7.7**	**13.9**

Table 2. FID scores of different methods.

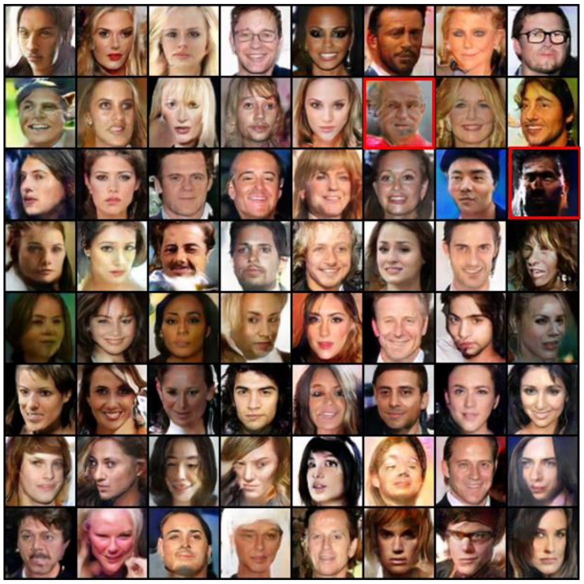
Results on a Dirac distribution We test WGAN-QC on a Dirac distribution which is concentrated at 0, with noise $z = -1$ with probability of 1. The generator is $G_\theta(z) = \theta \cdot z$. The discriminator is $D_w(x) = w \cdot x$. (w, θ) is initialized as $(0.01, 1.0)$. [22] showed that for this simple problem, the original GAN, WGAN and WGAN-GP do not converge, but CRGAN converges.

Results are in Fig. 2. The x-axis is the number of generator iterations and the y-axis is the output of the generator. Since the real data distribution is a Dirac distribution concentrated at 0, the generator output should converge to 0. WGAN-QC and CRGAN generate the true distribution, i.e., the output of the generator is 0. WGAN-div is oscillating around 0, mainly because the regularization term is very small near 0 according to the suggested parameter $p = 6$.

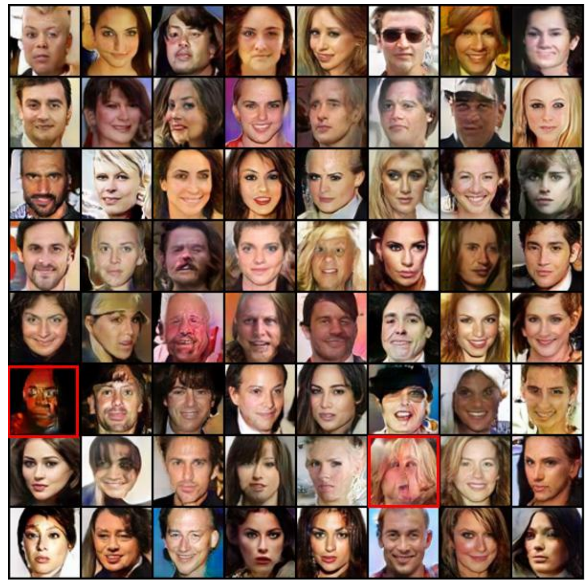
Results on the CelebA dataset Fig. 3 shows randomly generated images by each method. Many faces generated by WGAN-GP and WGAN-div have artifacts and some are incompletely generated. See faces marked with red boxes in Fig. 3 (a) and (b). CRGAN generates much better faces than WGAN-GP and WGAN-div. However, it tends to generate very similar faces (See faces marked with red and yellow boxes in Fig. 3 (c)). This suggests that CRGAN has a mode collapse problem. Fig. 3 (d) shows faces generated by WGAN-QC. All the faces generated by WGAN-QC are complete, smooth and distinct from each other. Almost all appear realistic.

FID scores of different methods on this dataset are in Table 2. WGAN-QC has the best performance of 12.9, which is 15.2% less than the second best method WGAN-div.

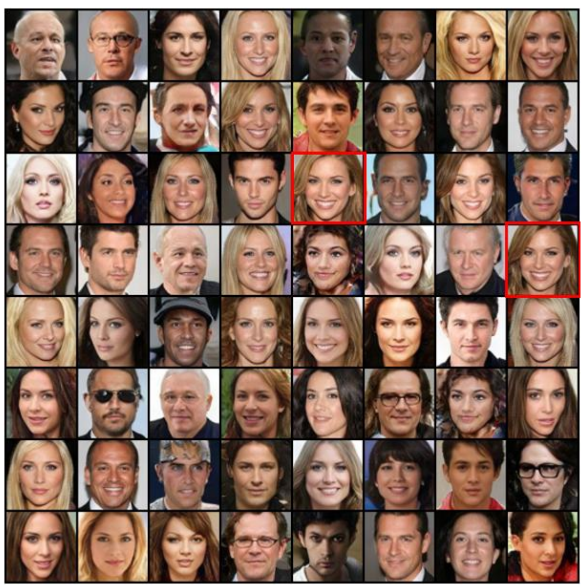
Results on the CelebA-HQ dataset We resize the face images in CelebA-HQ to 256×256 and train WGAN-QC on them. We can see that most of the randomly generated images by WGAN-QC in Figs. 1 and 4 look realistic. Even without progressive training, WGAN-QC can still generate



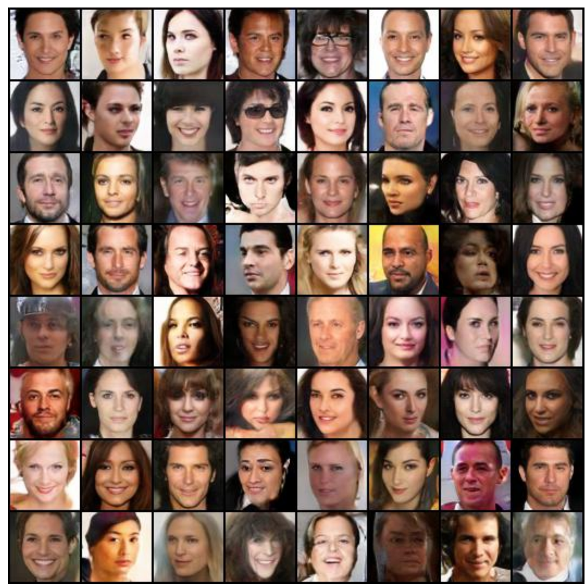
(a) WGAN-GP



(b) WGAN-div



(c) CRGAN



(d) WGAN-QC

Figure 3. Randomly generated faces by a) WGAN-GP, b) WGAN-div c) CRGAN and d) WGAN-QC. Obvious failure cases are marked with red boxes in (a) and (b). Red and yellow boxes in (c) suggest the mode collapse problem of CRGAN. All images generated by WGAN-QC are complete, natural and distinct from each other. (Best seen in color)

highly realistic 256×256 face images. We measure the performance of WGAN-QC following the strategy of [37] to compute the FID for WGAN-QC. We compare PGGAN, WGAN-div and WGAN-QC in Table 2. WGAN-QC's FID of 7.7 considerably reduces the FID of WGAN-div by 43%.

In order to verify the smoothness of the face manifold learned by WGAN-QC, we interpolate between two faces randomly generated by WGAN-QC. Fig. 5 shows that the face transitions appear to be smooth. This suggests WGAN-QC captures the face manifold well.

Results on the LSUN dataset WGAN-QC has the smallest FID score of 13.9 in Table 2, 12% less than that of WGAN-div. Fig. 6 shows images generated by these methods. Some images generated by WGAN-GP and CRGAN are hard to recognize as bedrooms. Many images generated by WGAN-div are distorted. Almost all images produced by WGAN-QC are smooth and look like bedrooms.

Figure 4. Faces of size 256 $\times$ 256 randomly generated by WGAN-QC on CelebA-HQ. (Best seen in color)

Figure 5. Face interpolation by WGAN-QC. Transitions between faces appear good. (Best seen in color)

SWGAN	OT-GAN	BigGAN	WGAN-QC
5.39	8.97	10.39	**10.48**

Table 3. Inception Scores on the ImageNet dog dataset.

Results on the ImageNet dog subset The Inception Scores (IS) achieved by state-of-the-art GAN methods are shown in Table 3. WGAN-QC is much better than SWGAN and OT-GAN on this dataset. WGAN-QC gives slightly higher IS than BigGAN, even though current version of WGAN-QC is *unsupervised* learning while BigGAN is *supervised* learning.

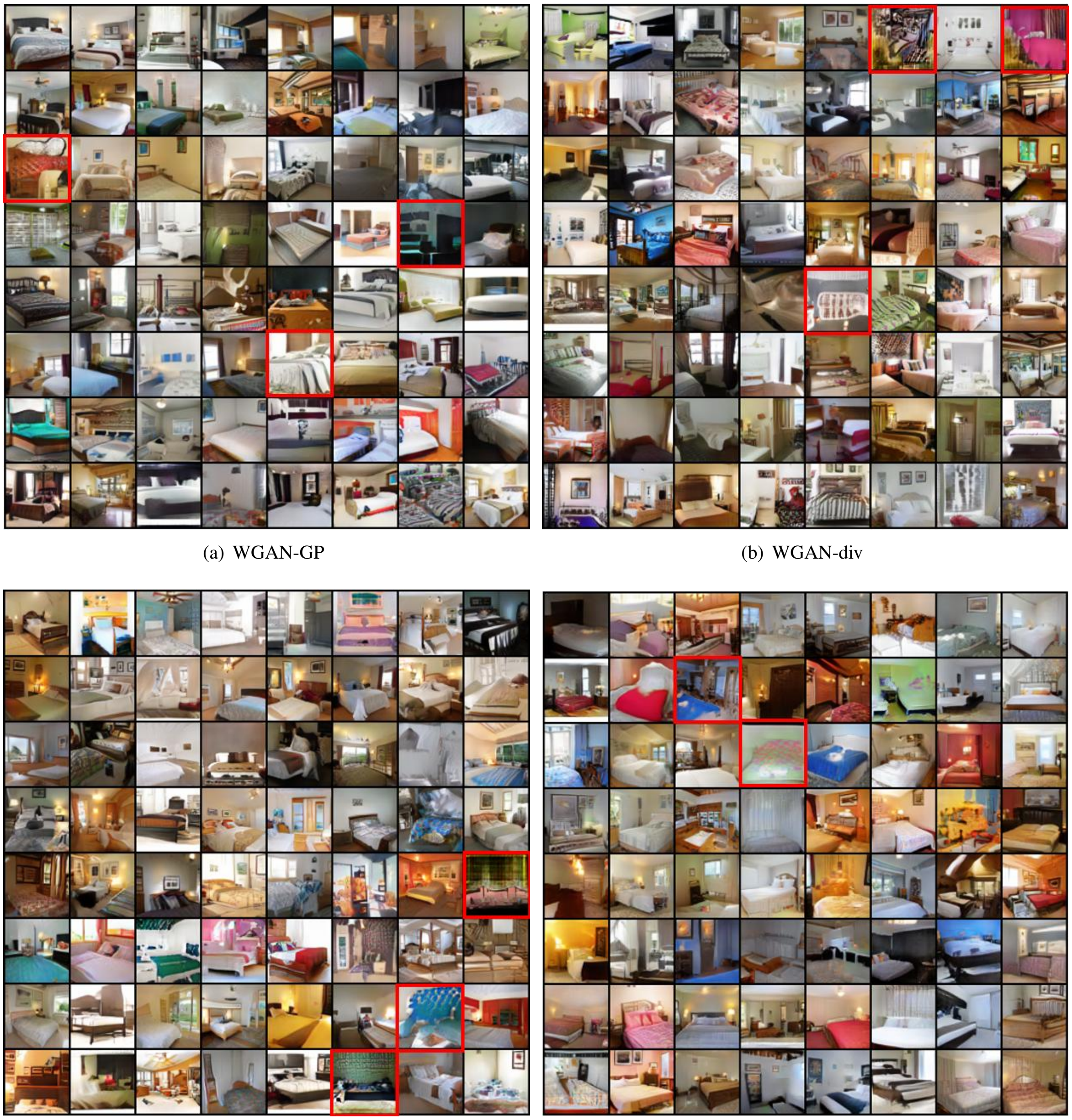
Run Time Comparison We run all comparisons on the same NVIDIA TITAN Xp under the same batch size of 64 on the CelebA and LSUN datasets. In Table 4 we show runtimes for WGAN-GP, WGAN-div and WGAN-QC. On both datasets WGAN-QC is the fastest one per iteration. Also, WGAN-QC requires the least overall training time on both datasets. WGAN-QC is 3.5x and 1.8x faster than WGAN-div on the CelebA and LSUN bedroom datasets, respectively.

Method	CelebA (i / o)	LSUN (i / o)
WGAN-GP	36.2ms / 5.0 days	47.8ms / 6.6 days
WGAN-div	30.6ms / 2.1 days	41.0ms / 2.4 days
WGAN-QC	**14.0ms / 0.6 days**	**18.6ms / 1.3 days**

Table 4. Running time comparison. i / o means running time per generator iteration / overall training time.

6. Conclusions and Future Work

In this paper, we proposed WGAN-QC, a WGAN with quadratic transport cost whose discriminator is regularized



(a) WGAN-GP (b) WGAN-div

(c) CRGAN (d) WGAN-QC

Figure 6. Images randomly generated by a) WGAN-GP, b) WGAN-div, c) CRGAN and d) WGAN-QC on the LSUN bedroom dataset. Obvious failure cases are marked with red boxes. (Best seen in color)

by optimal transport. We showed that the objective of the discriminator during each generator update computes the exact quadratic Wasserstein distance. We also proved that for small enough learning rates, WGAN-QC converges to a local equilibrium point. Consequently, we improved the state-of-the-art on four datasets while executing much faster than other WGAN variants.

In future work, we will extend WGAN-QC to the conditional version accepting image labels and investigate the performance of WGAN-QC on other large-scale datasets and higher-resolution images.

Acknowledgements

Chao Chen and Zhixin Shu provided valuable comments. This work was supported by a gift from Adobe, NSF grants CNS-1718014, IIS-1763981, 1762287, 1418255, 1737812 and DMS 1737876, the Partner University Fund, the SUNY2020 Infrastructure Transportation Security Center, and NSFC Grants 61772105, 61720106005 and 61432003.

References

[1] Leonard Adolphs, Hadi Daneshmand, Aurelien Lucchi, and Thomas Hofmann. Local saddle point optimization: A curvature exploitation approach. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2019.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the International Conference on Machine Learning*, 2017.

[3] David Berthelot, Thomas Schumm, and Luke Metz. BEGAN: boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[4] Dimitri P Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.

[5] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015.

[6] Yann Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on pure and applied mathematics*, 44(4):375–417, 1991.

[7] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *Proceedings of the International Conference on Learning Representations*, 2019.

[8] Ishan Deshpande, Ziyu Zhang, and Alexander G Schwing. Generative modeling using the sliced Wasserstein distance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[9] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a wasserstein loss. In *Advances in Neural Information Processing Systems*, 2015.

[10] Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2019.

[11] 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*, 2014.

[12] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, 2017.

[13] Xin Guo, Johnny Hong, Tianyi Lin, and Nan Yang. Relaxed Wasserstein with applications to GANs. *arXiv preprint arXiv:1705.07164*, 2017.

[14] R Devon Hjelm, Athul Paul Jacob, Tong Che, Adam Trischler, Kyunghyun Cho, and Yoshua Bengio. Boundary-seeking generative adversarial networks. In *Proceedings of the International Conference on Learning Representations*, 2018.

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[16] L.V. Kantorovich and G.S. Rubinstein. On a space of completely additive functions. *Vestnik Leningrad. Univ.*, 7:52–59, 1958.

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *Proceedings of the International Conference on Learning Representations*, 2018.

[18] Na Lei, Kehua Su, Li Cui, Shing-Tung Yau, and Xianfeng David Gu. A geometric view of optimal transportation and generative model. *Computer Aided Geometric Design*, 68:1–21, 2019.

[19] Huidong Liu, Xianfeng Gu, and Dimitris Samaras. A two-step computation of the exact GAN Wasserstein distance. In *Proceedings of the International Conference on Machine Learning*, 2018.

[20] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in Neural Information Processing Systems*, 2016.

[21] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision*, 2015.

[22] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *Proceedings of the International Conference on Machine Learning*, 2018.

[23] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of GANs. In *Advances in Neural Information Processing Systems*, 2017.

[24] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *Proceedings of the International Conference on Learning Representations*, 2018.

[25] Vaishnavh Nagarajan and J Zico Kolter. Gradient descent GAN optimization is locally stable. In *Advances in Neural Information Processing Systems*, 2017.

[26] Vu Nguyen, Tomas F Yago Vicente, Maozheng Zhao, Minh Hoai, and Dimitris Samaras. Shadow detection with conditional generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[27] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

[28] Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *Proceedings of the European Conference on Computer Vision*, 2018.

[29] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. In *Advances in Neural Information Processing Systems*, 2017.

[30] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[31] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, 2016.

[32] Tim Salimans, Han Zhang, Alec Radford, and Dimitris Metaxas. Improving gans using optimal transport. In *Proceedings of the International Conference on Learning Representations*, 2018.

[33] Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55:58–63, 2015.

[34] Zhixin Shu, Ersin Yumer, Sunil Hadap, Kalyan Sunkavalli, Eli Shechtman, and Dimitris Samaras. Neural face editing with intrinsic image disentangling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[35] Hoang Thanh-Tung, Truyen Tran, and Svetha Venkatesh. Improving generalization and stability of generative adversarial networks. In *Proceedings of the International Conference on Learning Representations*, 2019.

[36] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[37] Jiqing Wu, Zhiwu Huang, Janine Thoma, Dinesh Acharya, and Luc Van Gool. Wasserstein divergence for GANs. In *Proceedings of the European Conference on Computer Vision*, 2018.

[38] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. SUN Database: Large-scale scene recognition from abbey to zoo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[39] Raymond Yeh, Chen Chen, Teck Yian Lim, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with perceptual and contextual losses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[40] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the International Conference on Learning Representations*, 2018.