Wasserstein Proximal of GANs

Alex Tong Lin[†], Wuchen Li[†], Stanley J. Osher[†], and Guido Montúfar[† ‡]

[†]University of California, Los Angeles
[‡]Max Planck Institute

January 28, 2019

Abstract

We introduce a new method for training GANs by applying the Wasserstein-2 metric proximal on the generators. The approach is based on Wasserstein information geometry. It defines a parametrization invariant natural gradient by pulling back optimal transport structures from probability space to parameter space. We obtain easy-to-implement iterative regularizers for the parameter updates of implicit deep generative models in GANs. Our experiments demonstrate that this method improves the speed and stability of training in terms of wall-clock time and Fréchet Inception Distance (FID) learning curves.

1. Introduction

Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) are a powerful approach to learning generative models. Here, a discriminator tries to tell apart the data generated by a real source and the data generated by a generator, whereas the generator tries to fool the discriminator. This adversarial game is formulated as an optimization problem over an implicit generative model for the generator. An implicit generative model is a parametrized family of functions mapping a noise source to sample space. In trying to fool the discriminator, the generator should try to recreate the probability density of the real source.

The problem of matching a target density can be formulated as the minimization of a discrepancy measure. The Kullback–Leibler (KL) divergence is known to be difficult when the densities have a low dimensional support set, as is commonly the case in applications with structured data and high dimensional sample spaces. An alternative approach to define a discrepancy measure between densities is optimal transport, a.k.a. Wasserstein distance or Earth Mover's distance. This has been used recently to define the loss function for learning generative models (Montavon et al., 2016; Frogner et al., 2015). In particular, the Wasserstein GAN (Arjovsky et al., 2017) has attracted much interest in recent years.

Besides defining the loss function, optimal transport can also be used to introduce structures serving the *optimization* itself, in terms of the gradient operator. In full probability space, this method is known as the Wasserstein steepest descent flow (Jordan et al., 1998; Otto, 2001). In this paper we derive the Wasserstein steepest descent flow for deep generative models in GANs. We use the Wasserstein-2 metric function, which allows us to obtain a Riemannian structure and a corresponding natural (i.e., Riemannian) gradient. A well known example of a natural gradient is the Fisher-Rao natural gradient, which is induced by the KL-divergence. In learning problems, one often finds that the natural gradients offer advantages compared to the Euclidean gradient (Amari, 1998; 2016). In GANs, the densities under consideration typically have a small support set, which prevents implementations of the Fisher-Rao natural gradient. Therefore, we propose to use the gradient operator induced by the Wasserstein-2 metric on probability models (Li & Montúfar, 2018a;b).

We propose to compute the parameter updates of the generators in GANs by means of a proximal operator where the proximal penalty is a squared constrained Wasserstein-2 distance. In practice, the constrained distance can be approximated by a neural network. In implicit generative models, the constrained Wasserstein-2 metric exhibits a simple structure. We generalize the Riemannian metric and introduce two methods: the relaxed proximal operator for generators and the semi-backward Euler method. Both approaches lead to practical numerical implementations of the Wasserstein distance and the proximal operator for GANs. The method can be easily implemented as a drop-in regularizer for the generator updates. Experiments demonstrate that this method improves the stability of training and reduces the training time.

This paper is organized as follows. In Section 2 we introduce the Wasserstein natural gradient and proximal optimization methods. In Section 3 we derive practical computational methods and study their theoretical properties. In Section 4,

we demonstrate the effectiveness of the proposed methods in experiments with various types of GANs. In Section 5 we review related work.

2. Wasserstein natural proximal optimization

In this section, we briefly present the Wasserstein natural gradient and the corresponding proximal method.

2.1. Motivation and Illustration

The natural gradient method is an approach to parameter optimization in probability models, which has been promoted especially within information geometry (Amari, 2016; Ay et al., 2017). This method chooses the steepest descent direction when the size of the step is measured by means of a metric on probability space. In this way, the natural gradient is parameterization invariant (Amari, 1998) and provides more stability in training. In contrast, the ordinary gradient method follows the steepest descent direction calculated from Euclidean distance in parameter space. This can be unstable because distances in parameter space do not reflect distances in probability space, and the parameterization of the model affects the descent direction.

If $F(\theta)$ is the loss function, the steepest descent direction is the vector $d\theta$ that solves

$$\min_{d\theta} F(\theta + d\theta) \quad \text{subject to} \quad D(\rho_\theta, \rho_{\theta+d\theta}) = \epsilon \qquad (1)$$

for a small enough ϵ. Here D is a divergence function on probability space. Expanding the divergence to second order and solving leads to an update of the form

$$d\theta \propto G(\theta)^{-1} \nabla_\theta F(\theta),$$

where G is the Hessian of D. Usually the Fisher-Rao metric is considered for G, which corresponds to having D as the KL-divergence.

In this work, we use structures derived from optimal transport. Concretely, we replace D in equation (1) with the Wasserstein-p distance. This is defined as

$$W_p(\rho_\theta, \rho_{\theta^k})^p = \inf \int_{\mathbb{R}^n \times \mathbb{R}^n} \|x - y\|^p \pi(x, y) dx dy, \quad (2)$$

where the infimum is over all joint probability densities $\pi(x, y)$ with marginals ρ_θ, ρ_{θ^k}. We focus on $p = 2$. In this case, the Wasserstein-2 metric introduces a metric tensor in probability space making it an infinite dimensional Riemannian manifold. Later on, we will introduce a finite dimensional metric tensor G on the parameter space of a generative model.

The Wasserstein metric allows us to introduce a natural gradient even when the support of the distributions is low dimensional and the Fisher-Rao natural gradient is not well defined. We will use the proximal operator, which computes the parameter update by minimizing the loss function plus a penalty on the step size. This saves us the need to compute the matrix G explicitly. As we will show, the Wasserstein metric can be translated to practical proximal methods for implicit generative models.

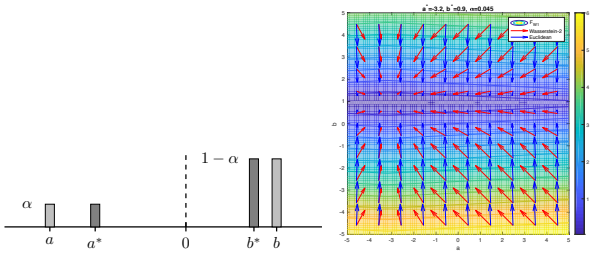

Figure 1. Illustration of the Wasserstein proximal operator. Here the Wasserstein proximal penalizes parameter steps in proportion to the mass being transported, which results in updates pointing towards the minimum of the loss function. The Euclidean proximal penalizes all parameters equally, which results in updates naively orthogonal to the level sets of the loss function.

We next present a toy example, with explicit calculations, to illustrate the effectiveness of Wasserstein proximal operator.

Consider a probability model consisting of mixtures of pairs of delta measures. Let $\Theta = \{\theta = (a, b) \colon a < 0 < b\}$, and define

$$\rho(\theta, x) = \alpha \delta_a(x) + (1 - \alpha)\delta_b(x),$$

where $\alpha \in [0, 1]$ is a given ratio and $\delta_a(x)$ is the delta measure supported at point a. See Figure 2.1. For a loss function F, writing $\theta = (a, b)$ the proximal update is

$$\theta^{k+1} = \arg\min_{\theta \in \Theta} F(\theta) + \frac{1}{2h} D(\rho_\theta, \rho_{\theta^k}),$$

We check the following common choices of the function D to measure the distance between θ and θ^k, $\theta \neq \theta^k$.

1. Wasserstein-2 distance:

$$W_2(\rho_\theta, \rho_{\theta^k})^2 = \alpha(a - a^k)^2 + (1 - \alpha)(b - b^k)^2;$$

2. Euclidean distance:

$$\|\theta - \theta^k\|^2 = (a - a^k)^2 + (b - b^k)^2;$$

3. Kullback–Leibler divergence:

$$D_{\mathrm{KL}}(\rho_\theta \| \rho_{\theta^k}) = \int_{\mathbb{R}^n} \rho(\theta, x) \log \frac{\rho(\theta, x)}{\rho(\theta^k, x)} dx = \infty;$$

4. L^2-distance:

$$L^2(\rho_\theta, \rho_{\theta^k}) = \int_{\mathbb{R}^n} |\rho(\theta, x) - \rho(\theta^k, x)|^2 dx = \infty.$$

As we see, the KL-divergence and L^2-distance take value infinity, which tells the two parameters apart, but does not quantify the difference in a useful way. The Wasserstein-2 and Euclidean distances still work in this case. The Euclidean distance considers the difference in the locations of the delta measures, but not their relative weights. On the other hand, the Wasserstein-2 takes these into account. As illustrated in Figure 1 and shown in the next proposition, the Wasserstein proximal update decreases the Wasserstein-1 loss function more strongly than the Euclidean proximal update.

Proposition 1 *Given $\theta^* = (a^*, b^*) \in \Theta$, consider the Wasserstein-1 metric as the loss function, i.e.,*

$$F_{W_1}(\theta) := W_1(\rho_\theta, \rho_{\theta^*}) = \alpha|a - a^*| + (1-\alpha)|b - b^*|.$$

If θ_W^{k+1} and θ_E^{k+1} denote the Wasserstein and Euclidean proximal parameter updates, from an initial parameter θ, then

$$F_{W_1}(\theta_E^{k+1}) \geq F_{W_1}(\theta_W^{k+1}).$$

Proof in Appendix B.

2.2. Wasserstein Natural Gradient

We next present the Wasserstein natural gradient operator for general probability models.

Definition 2 (Wasserstein natural gradient operator) *Given a loss function $F\colon \Theta \to \mathbb{R}$ and a probability model $\rho_\theta := \rho(\theta, x)$ with locally injective parametrization, the Wasserstein natural gradient operator is given by*

$$\operatorname{grad} F(\theta) = G(\theta)^{-1} \nabla_\theta F(\theta),$$

where $G(\theta) = (G(\theta)_{ij})_{1 \leq i,j \leq d} \in \mathbb{R}^{d \times d}$ is given by

$$G(\theta)_{ij} = \int_{\mathbb{R}^n} \Big(\nabla_{\theta_i} \rho(\theta, x), \mathcal{G}(\rho_\theta) \nabla_{\theta_j} \rho(\theta, x) \Big) dx,$$

and $\mathcal{G}(\rho_\theta)$ is the Wasserstein-2 metric tensor in probability space. More precisely, $\mathcal{G}(\rho) = (-\Delta_\rho)^{-1}$ is the inverse of the elliptical operator $\Delta_\rho := \nabla \cdot (\rho \nabla)$.

For convenience of the reader, we provide details on this definition in Appendix B. Our main focus here will be on deriving practical computational methods that allow us to apply these structures to optimization in GANs.

Consider the gradient flow of the loss function:

$$\frac{d\theta}{dt} = -\text{grad} F(\theta) = -G(\theta)^{-1} \nabla_\theta F(\theta). \tag{3}$$

There are several discretization schemes for a gradient flow of this type. One of them is the forward Euler method, known as the steep descent method:

$$\theta^{k+1} = \theta^k - h G(\theta^k)^{-1} \nabla_\theta F(\theta^k), \tag{4}$$

where $h > 0$ is the learning rate (step size).

In practice we usually do not have a closed formula for the metric tensor $G(\theta)$. In equation 4, we need to solve for the inverse Laplacian operator, the Jacobian of the probability model, and compute the inverse of $G(\theta)$. When the parameter $\theta \in \Theta$ is high dimensional, these computations are impractical. Therefore, we will consider a different approach based on the proximal method.

2.3. Wasserstein Natural Proximal

To practically apply the Wasserstein natural gradient, we present another way to discretize the gradient flow, known as the proximal method. The proximal operator computes updates of the form

$$\theta^{k+1} = \arg\min_\theta \; F(\theta) + \frac{\text{Dist}(\theta, \theta^k)^2}{2h}, \tag{5}$$

where Dist is an iterative regularization term, given by the Riemannian metric function as follows:

$$\text{Dist}(\theta, \theta^k)^2 = \inf \Big\{ \int_0^1 \dot\theta_t^\intercal G(\theta_t) \dot\theta_t dt \colon \theta_0 = \theta, \; \theta_1 = \theta^k \Big\}.$$

Here the infimum is taken among all continuously differentiable parameter paths $\theta_t = \theta(t) \in \Theta$, $t \in [0, 1]$.

The proximal operator is defined implicitly, in terms of a minimization problem, but in some cases it can be written explicitly. Interestingly, it allows us to consider an iterative regularization term in the parameter update.

We observe that there are two time variables in the proximal update (5). One is the time discretization of gradient flow, known as the learning rate $h > 0$; the other is the time variable in the definition of Riemannian distance $\text{Dist}(\theta, \theta^k)$. The variation in the time variable of the Riemannian distance can be further simplified.

Proposition 3 (Semi-backward Euler method) *The iteration*

$$\theta^{k+1} = \arg\min_\theta F(\theta) + \frac{\tilde D(\theta, \theta^k)^2}{2h}, \tag{6}$$

with

$$\tilde D(\theta, \theta^k)^2 = \int_{\mathbb{R}^n} \Big(\rho_\theta - \rho_{\theta^k}, \mathcal{G}(\rho_{\tilde\theta})(\rho_\theta - \rho_{\theta^k}) \Big) dx,$$

and $\tilde\theta = \frac{\theta + \theta^k}{2}$, is a consistent time discretization of the Wassserstein natural gradient flow (3).

Here the distance term in (5) is replaced by $\tilde D$, which is obtained by a mid-point approximation in time. The mid-point $\tilde\theta$ can be chosen in many ways between θ and θ^k. For simplicity and symmetry of $\tilde D(\theta, \theta^k)$, we let $\tilde\theta = \frac{\theta + \theta^k}{2}$.

Formula (6) is called the semi-backward Euler method (SBE), because it can also be expressed as

$$\theta^{k+1} = \theta^k - hG(\tilde{\theta})^{-1}\nabla_\theta F(\theta^{k+1}) + o(h).$$

We point out that all methods described above (forward Euler method (4), backward Euler method (5), and semi-backward Euler method (6)), are time consistent discretizations of the Wasserstein natural gradient flow (3) with first order accuracy in time. See details about this in Appendix B.

We shall focus on the semi-backward Euler method (6) and derive practical formulas for the iterative regularization term.

3. Computational methods

In this section, we present two methods for implementing the Wasserstein natural proximal for GANs. The first method is based on solving the variational formulation of the proximal penalty over an affine space of functions. This leads to a low-order version of the Wasserstein metric tensor $\mathcal{G}(\rho_\theta)$. The second method is based on a formula for the Wasserstein metric tensor for 1-dimensional sample spaces, which we relax to the case of arbitrary dimensions.

3.1. Implicit Generative Models

Before proceeding, we briefly recall the setting of GANs. For each parameter $\theta \in \Theta \subseteq \mathbb{R}^d$, let the generator be given by $g_\theta \colon \mathbb{R}^m \to \mathbb{R}^n$; $z \mapsto x = g(\theta, z)$. This takes an input Z in latent space $\mathbb{R}^m$ with distribution $p(z)$ to an output $X = g(\theta, Z)$ in sample space $\mathbb{R}^n$ with distribution $\rho(\theta, x)$. We have then

$$\mathbb{E}_{Z\sim p} f(g(\theta, Z)) = \mathbb{E}_{X\sim\rho_\theta} f(X), \text{ for any } f \in C_c^\infty(\mathbb{R}^n).$$

3.2. Affine Space Variational Approximation

The mid point approximation $\tilde{D}$ can be written using dual coordinates (cotangent space) of probability space in the variational form

$$\tilde{D}(\theta, \theta^k)^2 = \sup_{\Phi\in C^\infty(\mathbb{R}^n)} \Big\{ \int_{\mathbb{R}^n} \Phi(x)(\rho(\theta, x) - \rho(\theta^k, x)) - \frac{1}{2}\|\nabla\Phi(x)\|^2 \rho(\tilde{\theta}, x)\, dx \Big\}.$$

In order to obtain an explicit formula, we consider a function approximator of the form

$$\Phi_\xi(x) = \sum_j \xi_j \psi_j(x) = \xi^\top \Psi(x),$$

where $\Psi(x) = (\psi_j(x))_{j=1}^K$ are given basis functions on sample space $\mathbb{R}^n$ and $\xi = (\xi_j)_{j=1}^K \in \mathbb{R}^K$ is the parameter. In other words, consider

$$\tilde{D}(\theta, \theta^k)^2 = \sup_{\xi\in\mathbb{R}^K} \Big\{ \int_{\mathbb{R}^n} \Phi_\xi(x)(\rho(\theta, x) - \rho(\theta^k, x)) - \frac{1}{2}\|\nabla\Phi_\xi(x)\|^2 \rho(\tilde{\theta}, x)\, dx \Big\}. \tag{7}$$

Theorem 4 (Affine metric function $\tilde{D}$) *Consider some $\Psi = (\psi_1, \dots, \psi_K)^\top$ and assume that $M(\theta) = (M_{ij}(\theta))_{1\le i,j\le K} \in \mathbb{R}^{K\times K}$ is a regular matrix with entries*

$$M_{ij}(\theta) = \mathbb{E}_{Z\sim p}\Big(\sum_{l=1}^n \partial_{x_l}\psi_i(g(\tilde{\theta}, Z))\partial_{x_l}\psi_j(g(\tilde{\theta}, Z))\Big),$$

$\tilde{\theta} = \frac{\theta+\theta^k}{2}$. Then,

$$\tilde{D}(\theta, \theta^k)^2 = \Big(\mathbb{E}_{Z\sim p}[\Psi(g(\theta, Z)) - \Psi(g(\theta^k, Z))]\Big)^\top M(\tilde{\theta})^{-1}\Big(\mathbb{E}_{Z\sim p}[\Psi(g(\theta, Z)) - \Psi(g(\theta^k, Z))]\Big).$$

There are many possible choices for the basis Ψ. We will focus on for degree one and degree two polynomials.

If $K = n$, $\psi_k(x) = x_k$, $k = 1, \dots, n$, then $M(\theta)$ is the identity matrix. Thus

$$\tilde{D}(\theta, \tilde{\theta})^2 = \|\mathbb{E}_{Z\sim p}(g(\theta, Z) - g(\theta^k, z))\|^2.$$

In Appendix C we provide an explicit solution of (7) in the case of degree two polynomials.

3.3. Relaxation from 1-D

We next present a second method for approximating $\tilde{D}$. In the case of implicit generative models with 1-dimensional sample space, the constrained Wasserstein-2 metric tensor has an explicit formula. This allows us to define a relaxed Wasserstein metric for implicit generative models with sample spaces of arbitrary dimension.

Theorem 5 (1-D sample space) *If $n = 1$, then*

$$\mathrm{Dist}(\theta_0, \theta_1)^2 = \inf\Big\{ \int_0^1 \mathbb{E}_{Z\sim p}\|\frac{d}{dt} g(\theta(t), Z)\|^2\, dt \colon \theta(0) = \theta_0, \theta(1) = \theta_1 \Big\},$$

where the infimum is taken over all continuously differentiable parameter paths. Therefore, we have

$$\tilde{D}(\theta, \theta^k)^2 = \mathbb{E}_{Z\sim p}\|g(\theta, Z) - g(\theta^k, Z)\|^2.$$

In sample spaces of higher dimension, we do not have the explicit formula for $\tilde{D}$. The relaxed metric consists of using the same formulas from the theorem. Later on, we show that this formulation of $\tilde{D}$ still provides a metric with parameterization invariant properties in the proximal update.

3.4. Algorithms

The Wasserstein natural proximal method for GANs optimizes the parameter θ of the generator by the proximal iteration (6). We do this in a few ways:

(1) The first and simplest method follows Section 3.3, and updates the generator by:

$$\theta^{k+1} = \arg\min_{\theta\in\Theta} F(\theta) + \frac{1}{2h}\mathbb{E}_{Z\sim p}\|g(\theta, Z) - g(\theta^k, Z)\|^2,$$

We call this the Relaxed Wasserstein Proximal (RWP) method.

(2) The second method is based on the discussion from Section 3.2, approximating Φ by linear functions. We update the generator by:

$$\theta^{k+1} = \arg\min_{\theta\in\Theta} F(\theta) + \frac{1}{2h}\|\mathbb{E}_{Z\sim p}\Big(g(\theta, Z) - g(\theta^k, Z)\Big)\|^2,$$

We call this the Order-1 SBE (O1-SBE) method. In an analogous way, we can approximate Φ by quadratic functions (details in Appendix C), to obtain the Order-2 SBE (O2Diag-SBE) method:

$$\begin{aligned}\theta^{k+1} &= \arg\min_{\theta\in\Theta} F(\theta)\\ &+ \frac{1}{h}\bigg(\frac{1}{2}\|\mathbb{E}_{Z\sim p}[g(\theta, Z) - g(\theta^k, Z)] - \mathbb{E}_{Z\sim p}[Qg(\theta^k, Z)]\|^2\\ &\quad + \frac{1}{2}\mathbb{E}_{Z\sim p}[\langle g(\theta, Z), Qg(\theta, Z)\rangle]\\ &\quad - \frac{1}{2}\mathbb{E}_{Z\sim p}[\big\langle g(\theta^k, Z), Qg(\theta^k, Z)\big\rangle]\\ &\quad - \frac{1}{2}\mathbb{E}_{Z\sim p}[\|Qg(\theta^k, Z)\|^2]\bigg),\end{aligned}$$

where $Q = \mathrm{diag}(q_i)_{i=1}^n$ is the diagonal matrix

$$\begin{aligned}q_i =& \frac{1}{2}\frac{\mathbb{E}_{Z\sim p}[(g(\theta, Z)_i - g(\theta^k, Z)_i)^2]}{\mathrm{Var}(g(\theta^k, Z)_i)}\\ &+ \frac{\mathrm{Cov}_{Z\sim p}(g(\theta, Z)_i, g(\theta^k, Z)_i)}{\mathrm{Var}_{Z\sim p}(g(\theta^k, Z)_i)} - 1.\end{aligned}$$

where $g(\theta, Z)_i$ is the ith component.

These methods can be regarded as iterative regularizers. RWP penalizes the expected squared norm of the differences between samples (second moment differences). O1-SBE penalizes the squared norm of the expected differences between samples. O2Diag-SBE penalizes a combination of squared norm of the expected differences plus variances. They all encode the statistical information of generators. All these approaches **regularize the generator** by the expectation and variance of the samples.

The method is implemented as shown in Algorithm 1. We give a detailed practical guide in Appendix D. Shortly, we discuss convergence and consistency of these methods.

Algorithm 1 Wasserstein Natural Proximal

Require: F_ω, a parameterized function to minimize (e.g., Wasserstein-1 with a parameterized discriminator); g_θ, the generator.
Require: Optimizer$_{F_\omega}$; Optimizer$_{g_\theta}$.
Require: h proximal step-size; B batch size; max iterations; generator iterations.
for $k = 0$ **to** max iterations **do**
  Sample real data $\{x_i\}_{i=1}^B$ and latent data $\{z_i\}_{i=1}^B$
  $\omega^k \leftarrow \text{Optimizer}_{F_\omega}\Big(\frac{1}{B}\sum_{i=1}^B F_\omega(g_\theta(z_i))\Big)$
  for $\ell = 0$ **to** generator iterations **do**
    Sample latent data $\{z_i\}_{i=1}^B$
    $\tilde{D}$ = RWP, or O1-SBE, or O2Diag-SBE (Sec. 3.4)
    $\theta^k \leftarrow \text{Optimizer}_{g_\theta}\Big(\frac{1}{B}\sum_{i=1}^B F_\omega(g_\theta(z_i)) + \frac{1}{2h}\tilde{D}(\theta, \theta^k)^2\Big)$.
  end for
end for

3.5. Theoretical guarantees

We show that the Wasserstein natural proximal algorithms introduced in the previous sections are consistent.

Theorem 6 *Algorithm 1 provides a consistent numerical time discretization of the gradient flow*

$$\frac{d}{dt}\theta = -\tilde{G}(\theta)^\dagger \nabla_\theta F(\theta).$$

Here $\tilde{G}^\dagger$ is the pseudo inverse of the Hessian of $\tilde{D}$ and is a positive semi-definite matrix. In particular, the loss function is a Lyapunov function of gradient flow, meaning that it is non-increasing along the gradient flow. If θ^* is a critical point of F and $\lambda_{\min}\Big(\tilde{G}(\theta^*)^\dagger \operatorname{Hess} F(\theta^*)\Big) > 0$, then $\theta(t)$ locally converges to θ^*.

This theorem implies that the Wasserstein natural proximal methods developed in the previous sections, have all the expected properties of natural (Riemannian) gradient flows, including parametrization invariance. We note that with the approximation, G might not always be strictly positive definite, possibly introducing more critical points to the flow. This is a general phenomenon in gradient optimization with approximation and can be addressed by a variety of simple methods, such as the Levenberg-Marquard modification (Chong & Zak, 2013).

The Wasserstein metric in probability models introduces different convergence rates and convergence regions. We demonstrate the advantages of the method in the following experimetns.

4. Experiments

Here we present numerical experiments using the Relaxed Wasserstein Proximal (RWP) and the Semi-Backward Euler (SBE) methods in order to perform Wasserstein gradient-descent on various GANs. We find that the Relaxed Wasserstein Proximal provides both better speed (measured by wallclock) and stability in training GANs.

4.1. Experimental Setup

The Relaxed Wasserstein Proximal (RWP), O1-SBE, and O2Diag-SBE algorithms are intended to be an easy-to-implement, drop-in replacement to improve speed and convergence of GAN training. They apply regularizations on the generator updates during training. This is different from most GAN training, which focuses on regularizing the discriminator, e.g. with a gradient penalty (Gulrajani et al., 2017b; Petzka et al., 2017; Kodali et al., 2018; Adler & Lunz, 2018; Miyato et al., 2018). There has been limited exploration in regularizing the generator (Chen et al., 2016). Specifically, we modify the update rule for the generator by:

- Update for ℓ number of iterations before updating the discriminator:

$$\theta \leftarrow \text{Optimizer}_\theta\left(\text{Original loss} + \frac{1}{2h}\tilde{D}(\theta, \theta^k)\right)$$

where $\tilde{D}(\theta, \theta^k)$ can be chosen to be one of the distances found in the previous section 3.4. So two hyperparameters are introduced: the proximal step-size h, and the number of iterations ℓ. In some GANs, one may update the discriminator a number of times and then update the generator a number of times, and then repeat; we will call one loop of this update an *outer-iteration*. A more detailed description of the algorithm is given in Appendix D.

We test the Relaxed Wassersteing Proximal, O1-SBE, and O2Diag-SBE regularization on three GAN types: Vanilla GANs (Goodfellow et al., 2014) (Jenson-Shannon), WGAN-GP (Gulrajani et al., 2017a), and DRAGAN (Kodali et al., 2018). We use the CIFAR-10 dataset (Krizhevsky, 2009), and the aligned and cropped CelebA dataset (Liu et al., 2015). And we utilize the DCGAN (Radford et al., 2015) architecture for the discriminator and generator. To measure the quality of generated samples, we employ the Fréchet Inception Distance (FID) (Heusel et al., 2017) both to measure performance and to measure convergence of GAN training (lower FID is better); we used 10,000 generated images to measure the FID. For CIFAR-10, we measure the FID every 1000 outer-iterations.

It is tricky to compare the result of using the regularizers, as it performs multiple generator iterations. We thus align the comparison according to wallclock time (this procedure was also used by Heusel et al., 2017).

Our particular hyperparameter choices for training are given in Appendix E.1. Note that since we are testing these effectiveness of these regularizations as a drop-in tool, the hyperparameters (i.e. not h nor ℓ) are chosen to work well before applying our regularization.

Samples from the models are provided in Appendix F. We also performed latent space walks (Radford et al., 2015) to show RWP regularization does not cause the GAN to memorize. For details see Appendix G).

4.2. Results on the CIFAR-10 dataset

We summarize our results for the tested regularizations on CIFAR10. We see in Figure 2 that RWP, O1-SBE, and O2Diag-SBE improve the speed of convergences regarding wallclock time. In DRAGAN, our experiments show that the regularizations provide better stability, in the sense of less oscillations in FID values. We also see that our method achieves lower FID values. This is especially true in the case of O1-SBE and O2Diag-SBE, and in particular under WGAN-GP where they achieve the same final FID, but about six times faster. Overall, we see that the fastest method to train CIFAR10 out of the presented methods is Vanilla GANs with O1-SBE or RWP. For O2Diag-SBE, it is possible that different h and ℓ values can improve wallclock time. However we did not do much hyperparameters tuning, and were still able to obtain excellent performance.

In the appendix G, we also examine if our models exhibit memorization by performing a latent-space walk and we find that it does not. Thus, even when we are updating the generator multiple times per outer-iteration (see the previous section), it still does not overfit. In appendix E.1, we provide all hyperparameter settings for our experiments.

4.3. Results on the CelebA dataset

Our results on the CelebA dataset are presented in Figure 3. In this case, we only examine the effect of RWP, O1-SBE, and O2Diag-SBE on the Vanilla and WGAN-GP GANs, as they are the two most popular ones.

For Vanilla GANs, we see that RWP, O1-SBE, as well as O2Diag-SBE improve the speed of GAN training according to wallclock time, and they also achieve a lower FID.

For WGAN-GP, our regularizations are comparable to the standard WGAN-GP. In Figure 3, we see that adding the regularizations actually improves the stability of WGAN-GP under higher learning rates (0.002 vs 0.0001, 20 times larger) and higher momentum (Adam $\beta_1 = 0.5$ v.s. 0). Overall, it seems the fastest method to train CelebA according to the methods we have presented is Vanilla GAN with O2Diag-SBE.

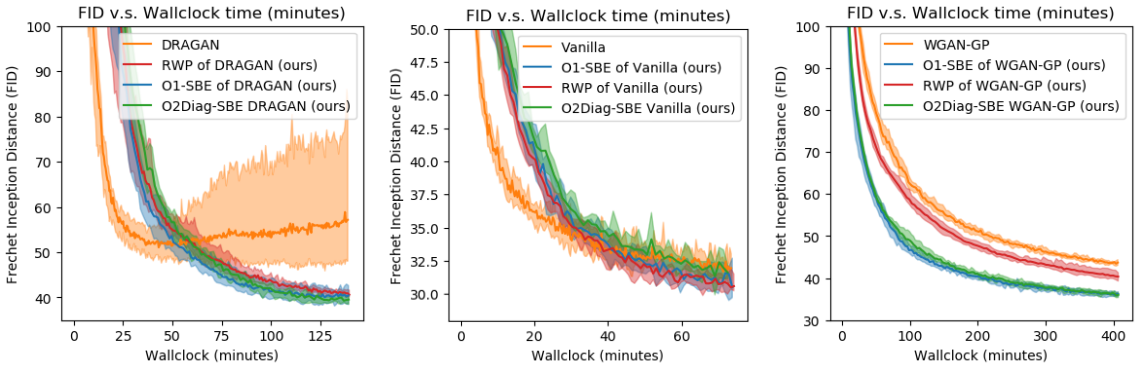

Figure 2. The effect of using RWP, O1-SBE, and O2Diag-SBE regularization on the CIFAR-10 dataset. The experiments are averaged over 5 runs. The bold lines are the average, and the enveloping lines are the minimum and maximum. From the three graphs, we see that using the easy-to-implement RWP, O1-SBE, O2Diag-SBE regularizations all improve speed as measured by wallclock time, and it also can achieve a lower FID.

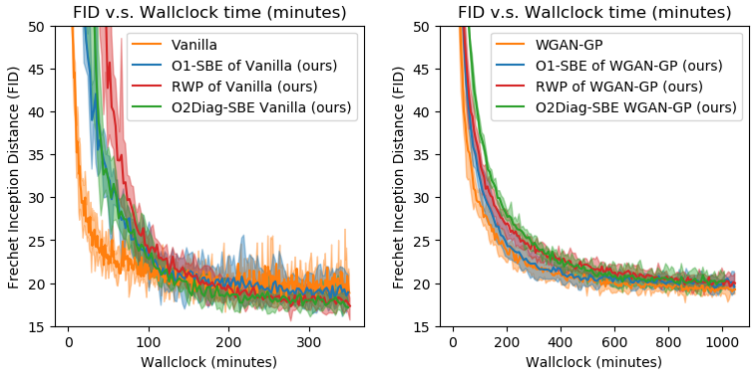

Figure 3. The effect of RWP, O1-SBE, and O2Diag-SBE regularization on Vanilla GANs and WGAN-GP, on the CelebA dataset. The experiment was averaged over 5 runs. The bold lines are the average, and the enveloping lines are the minimum and maximum. We see that RWP, O1-SBE, and O2Diag-SBE regularizations improve the speed (via wallclock time), and achieve lower FID values. For RWP, we note multiple generator iterations might cause initial learning to fail, but once it starts then it remains successful. It is practically easy to detect, so we show successful runs. Note for the standard Vanilla GANs (i.e. no regularization), we removed a troublesome run whose FID values spiked up so the average is over 4 runs instead. For WGAN-GP, we see that RWP, O1-SBE, and O2Diag-SBE are comparable to it.

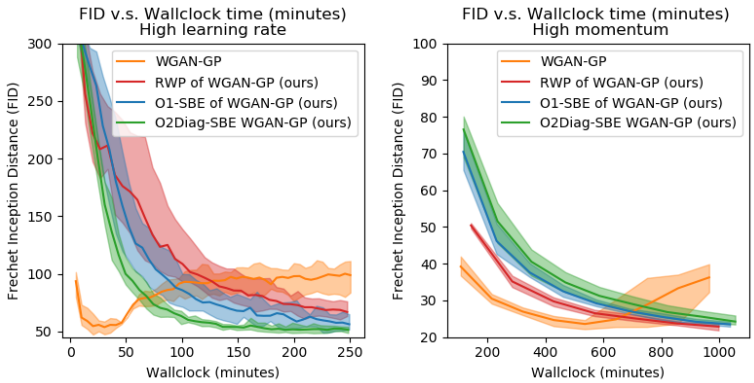

Figure 4. Here we see that RWP, O1-SBE, and O2Diag-SBE all improve the training by providing a lower FID when the learning rate or momentum is high. For a high learning rate, WGAN-GP quickly reaches an FID of around 50, but then it diverges and goes up to 100. The regularizations help in stabilizing the FID to be about 50. For a high momentum, we see that WGAN-GP quickly reaches an FID of around 20, but again quickly jumps up, whereas our regularizations help keep the FID at approximately 20, which we consider optimal and converged.

5. Related Works

In the literature, many different aspects of optimal transport have been considered in machine learning and GANs.

1. *Wasserstein Loss function.* Many studies apply the Wasserstein distance as the loss function (Frogner et al., 2015; Montavon et al., 2016). On the one hand, the Wasserstein distance is a statistical distance depending on the metric of the sample space. So it introduces a statistical estimator, named the minimal Wasserstein estimator (Bassetti et al., 2006), depending on the geometry of the data. On the other hand, the Wasserstein distance is useful for comparing probability distributions supported on lower dimensional sets. In the framework of GAN, the loss function is chosen as the Wasserstein-1 distance function (Arjovsky et al., 2017). In its computations, the discriminator, also called the Kantorovich dual variable, needs to satisfy the 1-Lipschitz condition. Many studies work on the regularization of the discriminator to fulfill this condition (Gulrajani et al., 2017b; Petzka et al., 2017). In contrast to current works, we apply the Wasserstein-2 distance to construct gradient operators in optimizations of GANs. It will result at an iterative regularization on generators.

2. *Wasserstein Gradient flows.* The Wasserstein-2 metric provides a metric tensor structure (Lott, 2007; Otto, 2001; Li, 2018), under which the probability space forms an infinite-dimensional Riemannian manifold, named the density manifold (Lafferty, 1988). The gradient flow in the density manifold links with many transport-related partial differential equations (Villani, 2009; Nelson, 1985). A famous example is that the Fokker-Planck equation, the probability transition equation of Langevin dynamics, is the gradient flow of the KL-divergence function. In this perspective, two angles have been developed in the learning communities. Firstly, many groups try to leverage the gradient flow structure in probability space supported on the parameter space. They study the stochastic gradient descent by the transition equation in the probability over parameters (Mei et al., 2018). Secondly, many nonparametric models have been studied, such as the Stein gradient descent method (Liu, 2017). It is a generalization of Wasserstein gradient flow. Also, (Frogner & Poggio, 2018) consider an approximate inference method for computing Wasserstein gradient flow. Here an approximation towards Kantorovich dual variables is introduced. Comparing to these works, we consider Wasserstein structure constrained on parameter space. In this direction, (Carlen & Gangbo, 2003) studied the constrained Wasserstein gradient with fixed mean and variance. Here the density subset is still infinite dimensional. Many approaches also focus on Gaussian families or elliptical distributions (Takatsu, 2011). The Wasserstein gradient flow in Gaussian family has been studied by (Malagò et al., 2018). Compared to previous works, our approach applies the Wasserstein gradient to work on implicit generative models.

3. *Wasserstein Proximal operator.* The gradient flow is often computed or approximated by the proximal operator. In full probability space with Wasserstein-2 distance, this proximal iteration is often named the Jordan-Kinderlehrer-Otto (JKO) scheme (Jordan et al., 1998). It is also the backward Euler method in a Riemannian manifold concerning Wasserstein-2 metric tensor. Many numerical methods have been build in this direction (Caluya & Halder, 2018). Compared to current works, we propose the proximal operator within probability models. Instead of applying the Backward Euler method (JKO) on parameter space, we consider the semi-backward method. See similar approaches in (Vantzos et al., 2017). We further approximate the Wasserstein proximal in affine function space. It results at an analytical iterative regularization term depending on the statistical information of generators. In future, many another sampling efficient computational method could also be considered.

6. Discussion

In this work, we develop methods to implement the Wasserstein natural gradient method for learning implicit generative models. To practically apply the method, we consider a proximal parameter update. We obtain explicit formulas expressed regarding statistics of the generated samples, which can be implemented at little to no additional cost over current methods. One salient aspect of our approach is that it regularizes the generator, whereas much of the present work focuses on regularizing the discriminator. Experimentally, we found that the proposed method does not harm in simple data sets, but that it can provide substantial benefits in more complex data sets, allowing us to obtain a better minimizer in the sense of FID, with faster convergence speeds in wall-clock time. Moreover, our method can offer benefits concerning stability to the choice of hyperparameters such as step size and momentum. It can save the needs for extensive hyperparameter tuning that is typically required to achieve state of the art results with current methods.

References

Adler, J. and Lunz, S. Banach Wasserstein GAN. *ArXiv e-prints*, June 2018.

Amari, S. Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10(2):251–276, 1998.

Amari, S. *Information Geometry and Its Applications*. Number volume 194 in Applied mathematical sciences. Springer, Japan, 2016.

Ambrosio, L., Gigli, N., and Savaré Giuseppe. *Gradient*

Flows: In Metric Spaces and in the Space of Probability Measures. Birkhäuser Basel, Basel, 2005.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein GAN. *arXiv:1701.07875 [cs, stat]*, 2017.

Ay, N., Jost, J., Lê, H. V., and Schwachhöfer, L. J. *Information Geometry*. Ergebnisse der Mathematik und ihrer Grenzgebiete A @series of modern surveys in mathematics$l3. Folge, volume 64. Springer, Cham, 2017.

Bassetti, F., Bodini, A., and Regazzini, E. On minimum kantorovich distance estimators. *Statistics & Probability Letters*, 76(12):1298 – 1302, 2006. ISSN 0167-7152. doi: https://doi.org/10.1016/j.spl.2006.02.001. URL http://www.sciencedirect.com/science/article/pii/S0167715206000381.

Caluya, K. F. and Halder, A. Proximal recursion for solving the fokker-planck equation, 2018.

Carlen, E. A. and Gangbo, W. Constrained Steepest Descent in the 2-Wasserstein Metric. *Annals of Mathematics*, 157 (3):807–846, 2003.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 2172–2180. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6399-infogan-interpretable-representation-learning-by-information-maximizing-generative-adversarial-nets.pdf.

Chong, E. and Zak, S. *An Introduction to Optimization*. Wiley Series in Discrete Mathe. Wiley, 2013. ISBN 9781118279014. URL https://books.google.de/books?id=8J_ev5ihKEoC.

Frogner, C. and Poggio, T. Approximate inference with Wasserstein gradient flows. *ArXiv e-prints*, June 2018.

Frogner, C., Zhang, C., Mobahi, H., Araya-Polo, M., and Poggio, T. Learning with a Wasserstein Loss. *arXiv:1506.05439 [cs, stat]*, 2015.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5767–5777. Curran Associates, Inc., 2017a. URL http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5767–5777. Curran Associates, Inc., 2017b. URL http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 6626–6637. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7240-gans-trained-by-a-two-time-scale-update-rule-converge-to-a-local-nash-equilibrium.pdf.

Jordan, R., Kinderlehrer, D., and Otto, F. The Variational Formulation of the Fokker–Planck Equation. *SIAM Journal on Mathematical Analysis*, 29(1):1–17, 1998.

Kodali, N., Hays, J., Abernethy, J., and Kira, Z. On convergence and stability of GANs, 2018. URL https://openreview.net/forum?id=ryepFJbA-.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Lafferty, J. D. The density manifold and configuration space quantization. *Transactions of the American Mathematical Society*, 305(2):699–741, 1988.

Li, W. Geometry of probability simplex via optimal transport. *arXiv:1803.06360 [math]*, 2018.

Li, W. and Montúfar, G. Natural gradient via optimal transport. *arXiv:1803.07033 [cs, math]*, 2018a.

Li, W. and Montúfar, G. Ricci curvature for parametric statistics via optimal transport. *arXiv:1807.07095 [cs, math, stat]*, 2018b.

Liu, Q. Stein Variational Gradient Descent as Gradient Flow. *arXiv:1704.07520 [stat]*, 2017.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Lott, J. Some Geometric Calculations on Wasserstein Space. *Communications in Mathematical Physics*, 277(2):423–437, 2007.

Malagò, L., Montrucchio, L., and Pistone, G. Wasserstein Riemannian Geometry of Positive Definite Matrices. *arXiv:1801.09269 [math, stat]*, 2018.

Mei, S., Montanari, A., and Nguyen, P.-M. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33): E7665–E7671, 2018.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1QRgziT-.

Montavon, G., Müller, K.-R., and Cuturi, M. Wasserstein Training of Restricted Boltzmann Machines. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 3718–3726. Curran Associates, Inc., 2016.

Nelson, E. *Quantum Fluctuations*. Princeton series in physics. Princeton University Press, Princeton, N.J, 1985.

Otto, F. The geometry of dissipative evolution equations the porous medium equation. *Communications in Partial Differential Equations*, 26(1-2):101–174, 2001.

Petzka, H., Fischer, A., and Lukovnicov, D. On the regularization of Wasserstein GANs. *arXiv:1709.08894 [cs, stat]*, 2017.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015. URL http://arxiv.org/abs/1511.06434.

Takatsu, A. Wasserstein geometry of Gaussian measures. *Osaka Journal of Mathematics*, 48(4):1005–1026, 2011.

Vantzos, O., Azencot, O., Wardeztky, M., Rumpf, M., and Ben-Chen, M. Functional thin films on surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 23(3):1179–1192, March 2017. ISSN 1077-2626. doi: 10.1109/TVCG.2016.2605083.

Villani, C. *Optimal Transport: Old and New*. Number 338 in Grundlehren der mathematischen Wissenschaften. Springer, Berlin, 2009.

A. Review of Wasserstein Information geometry

In this section, we briefly review the geometry of L^2-Wasserstein metric tensor in the probability set and probability models.

Consider the probability density set with finite second moment $\mathbb{P}_2(\mathbb{R}^n)$. Consider a metric function $W_2\colon \mathcal{P}_2(\mathbb{R}^n)\times\mathcal{P}_2(\mathbb{R}^n)\to\mathbb{R}_+$,

$$W_2(\rho_0,\rho_1)^2=\inf_{\Phi_t}\Big\{\int_0^1\int_{\mathbb{R}^n}\|\nabla\Phi(t,x)\|^2\rho(t,x)dxdt\colon \\ \partial_t\rho(t,x)+\nabla\cdot(\rho(t,x)\nabla\Phi(t,x))=0, \\ \rho(0,x)=\rho_0(x),\ \rho(1,x)=\rho_1(x)\Big\}, \tag{8}$$

where the infimum is taken among all feasible Borel potential functions $\Phi\colon[0,1]\times\mathbb{R}^n\to\mathbb{R}$ and continuous density path $\rho\colon[0,1]\times\mathbb{R}^n\to\mathbb{R}_+$ satisfying the continuity equation.

The variational formulation in (8) introduces a Riemannian structure in density space. We plain this as follows. Consider the set of smooth and strictly positive probability densities

$$\mathcal{P}_+=\Big\{\rho\in C^\infty(\mathbb{R}^n)\colon\rho(x)>0,\ \int_{\mathbb{R}^n}\rho(x)dx=1\Big\} \\ \subset\mathcal{P}_2(\mathbb{R}^n).$$

Denote $\mathcal{F}:=C^\infty(\mathbb{R}^n)$ the set of smooth real valued functions. The tangent space of $\mathcal{P}_+$ is given by

$$T_\rho\mathcal{P}_+=\Big\{\sigma\in\mathcal{F}\colon\int_{\mathbb{R}^n}\sigma(x)dx=0\Big\}.$$

Given $\Phi\in\mathcal{F}$ and $\rho\in\mathcal{P}_+$, define

$$V_\Phi(x):=-\nabla\cdot(\rho(x)\nabla\Phi(x)).$$

Thus $V_\Phi\in T_\rho\mathcal{P}_+$. The elliptic operator $\nabla\cdot(\rho\nabla)$ identifies the function Φ modulo additive constants with the tangent vector V_Φ of the space of densities.

Given $\rho\in\mathcal{P}_+$, $\sigma_i\in T_\rho\mathcal{P}_+$, $i=1,2$, define

$$g_\rho^W(\sigma_1,\sigma_2)=\int_{\mathbb{R}^n}(\nabla\Phi_1(x),\nabla\Phi_2(x))\rho(x)dx,$$

where $\Phi_i(x)\in\mathcal{F}/\mathbb{R}$, such that $-\nabla\cdot(\rho\nabla\Phi_i)=\sigma_i$. Denote $\Phi_i=-\Delta_\rho)^{-1}\sigma_i$, then

$$g_\rho^W(\sigma_1,\sigma_2)=\int_{\mathbb{R}^n}\Big(\sigma_1(x),(-\Delta_\rho)^{-1}\sigma_2(x)\Big)dx.$$

The inner product g^W endows $\mathcal{P}_+$ with a Riemannian metric tensor. In other words, the variational problem equation 8 is a geometric action energy in $(\mathcal{P}_+, g^W)$.

Given a loss function $F\colon \mathcal{P}_+ \to \mathbb{R}$, the Wasserstein gradient operator in $(\mathcal{P}_+, g^W)$ is given as follows.

$$\begin{aligned}\text{grad}_W F(\rho) =& \Big((-\Delta_\rho)^{-1}\Big)^{-1} \frac{\delta}{\delta\rho(x)} F(\rho)\\ =& -\nabla\cdot(\rho\nabla\frac{\delta}{\delta\rho(x)}F(\rho)).\end{aligned}$$

Thus the gradient flow satisfies

$$\frac{\partial\rho}{\partial t} = -\text{grad}_W F(\rho) = \nabla\cdot(\rho\nabla\frac{\delta}{\delta\rho(x)}F(\rho)).$$

More analytical results on the Wasserstein-2 gradient flow are provided in (Ambrosio et al., 2005).

We next consider Wasserstein-2 metric and gradient operator constrained on statistical models. A statistical model is defined by a triplet $(\Theta, \mathbb{R}^n, \rho)$. For simple presentation of paper, we assume $\Theta \subset \mathbb{R}^d$ and $\rho\colon \Theta \to \mathcal{P}(\mathbb{R}^n)$ is a parameterization function. In this case, $\rho(\Theta) \subset \mathcal{P}(\mathbb{R}^n)$. We assume that the parameterization map ρ is locally injective and under suitable regularities. We define a Riemannian metric g on $\rho(\Theta)$ by pulling back the Wasserstein-2 metric tensor g^W.

Definition 7 (Wasserstein statistical manifold) *Given $\theta \in \Theta$ and $\dot\theta_i \in T_\theta\Theta$, $i = 1, 2$, we define*

$$g_\theta(\dot\theta_1, \dot\theta_2) = \int_{\mathbb{R}^n} \Big((\dot\theta_1, \nabla_\theta\rho), (-\Delta_\rho)^{-1}(\nabla_\theta\rho, \dot\theta_2)\Big)dx.$$

In other notations,

$$g_\theta(\dot\theta_1, \dot\theta_2) = \int_{\mathbb{R}^n} \nabla\Phi_1(x)\nabla\Phi_2(x)\rho(\theta, x)dx,$$

where

$$-\nabla\cdot(\rho(\theta, x)\nabla\Phi_i(x)) = (\nabla_\theta\rho(\theta, x), \dot\theta_i).$$

Here $\nabla_\theta\rho = (\frac{\partial}{\partial\theta_i}\rho(\theta, x))_{i=1}^d \in \mathbb{R}^d$ and $(\cdot,\cdot)$ is an Euclidean inner product in $\mathbb{R}^d$.

In particular, we denote

$$g_\theta(\dot\theta_1, \dot\theta_2) = \dot\theta_1^{\mathsf{T}} G(\theta)\dot\theta_2,$$

where $G(\theta) = (G(\theta)_{ij})_{1\le i,j\le d} \in \mathbb{R}^{d\times d}$ is the associated metric tensor defined in Theorem 2. Thus the distance function can be written into the geometry action functional

$$\begin{aligned}&\text{Dist}(\theta, \theta^k)^2\\ =& \inf\Big\{\int_0^1 \dot\theta(t)^{\mathsf{T}} G(\theta(t))\dot\theta(t)dt\colon \theta(0) = \theta,\ \theta(1) = \theta^k\Big\}\\ =& \inf\Big\{\int_0^1\int_{\mathbb{R}^n} (\partial_t\rho(\theta(t), x), \mathcal{G}(\rho_\theta)\partial_t\rho(\theta(t), x))dxdt\colon\\ &\qquad\qquad \theta(0) = \theta,\ \theta(1) = \theta^k\Big\}\\ =& \inf\Big\{\int_0^1\int_{\mathbb{R}^n} \|\nabla\Phi(t, x)\|^2\rho(\theta(t), x)dxdt\colon\\ &\qquad \partial_t\rho(\theta(t), x) + \nabla\cdot(\rho(\theta(t), x)\nabla\Phi(t, x)) = 0,\\ &\qquad\qquad \theta(0) = \theta,\ \theta(1) = \theta^k\Big\}.\end{aligned} \tag{9}$$

B. Proofs

Proof of Proposition 1. This example allows us to compute the proximal operator explicitly. On the one hand, we compute the Wasserstein proximal operator explicitly:

$$\begin{aligned}\theta_W^{k+1} &= (a_W^{k+1}, b_W^{k+1})\\ &= \arg\min_\theta F_{W_1}(\theta) + \frac{1}{2h}W(\rho_\theta, \rho_{\theta^k})^2\\ &= \arg\min_{(a,b)} \alpha|a - a^*| + (1-\alpha)|b - b^*|\\ &\qquad + \frac{1}{2h}(\alpha|a - a^k|^2 + (1-\alpha)|b - b^k|).\end{aligned}$$

I.e.,

$$\begin{aligned}a_{k+1}^W &= \arg\min_a |a - a^*| + \frac{1}{2h}|a - a^k|^2,\\ b_{k+1}^W &= \arg\min_b |b - b^*| + \frac{1}{2h}|b - b^k|^2.\end{aligned}$$

Here

$$a_W^{k+1} = \text{shrink}_{a^*}(a^k, h) = \begin{cases} a^k - h & \text{if } a^k > a^* + h;\\ a^k + h & \text{if } a^k < a^* - h;\\ a^* & \text{otherwise.}\end{cases}$$

Similarly, $b_W^{k+1} = \text{shrink}_{b*}(b^k, h)$.

On the other hand, we calculate the Euclidean proximal operator explicitly:

$$\begin{aligned}\theta_E^{k+1} &= (a_E^{k+1}, b_E^{k+1})\\ &= \arg\min_\theta F_{W_1}(\theta) + \frac{1}{2h}d_E(\theta, \theta^k)^2\\ &= \arg\min_{(a,b)} \alpha|a - a^*| + (1-\alpha)|b - b^*|\\ &\qquad + \frac{1}{2h}(|a - a^k|^2 + |b - b^k|^2).\end{aligned}$$

I.e.,

$$a_E^{k+1} = \arg\min_a \alpha|a-a^*| + \frac{1}{2h}|a-a^k|^2,$$
$$b_E^{k+1} = \arg\min_b (1-\alpha)|b-b^*| + \frac{1}{2h}|b-b^k|^2.$$

Here

$$a_E^{k+1} = \text{shrink}_{a^*}(a^k, \alpha h) = \begin{cases} a^k - \alpha h & \text{if } a^k > a^* + \alpha h; \\ a^k + \alpha h & \text{if } a^k < a^* - \alpha h; \\ a^* & \text{otherwise.} \end{cases}$$

Similarly, $b_E^{k+1} = \text{shrink}_{b*}(b^k, (1-\alpha)h)$.

Here we only need to check that for all possible cases, $F_{W_1}(\theta_E^{k+1}) > F_{W_1}(\theta_W^{k+1})$. If $a^k > a^* + h$ and $b^k > b^* + h$, then

$$\begin{aligned} F_{W_1}(\theta_W^{k+1}) &= \alpha[(a^k - a^* - h) + \frac{h}{2}] \\ &\quad + (1-\alpha)[(b^k - b^* - h) + \frac{h}{2}] \\ &= \alpha(a^k - a^*) + (1-\alpha)(b^k - b^*) - \frac{h}{2}, \end{aligned}$$

and

$$\begin{aligned} F_{W_1}(\theta_E^{k+1}) =& \alpha[(a^k - a^* - \alpha h)] + \frac{(\alpha h)^2}{2h} \\ &+ (1-\alpha)[(b^k - b^* - \alpha h)] + \frac{(1-\alpha)^2 h^2}{2h} \\ =& \alpha(a^k - a^*) + (1-\alpha)(b^k - b^*) \\ &- \frac{h}{2}[\alpha^2 + (1-\alpha)^2]. \end{aligned}$$

Since $\alpha \in [0,1]$, then $\alpha^2 + (1-\alpha)^2 \le [\alpha + (1-\alpha)]^2 = 1$, then $F_{W_1}(\theta_W^{k+1}) \le F_{W_1}(\theta_E^{k+1})$. In other cases, the proof follows similarly. We finish the proof.

Derivation of Wasserstein natural gradient. Here we briefly explain the Definition 2.

The gradient operator on a Riemannian manifold (Θ, g_θ) is defined as follows. For any $\sigma \in T_\theta\Theta$, then the Riemannian gradient $\nabla_\theta^W F(\theta) \in T_\theta\Theta$ satisfies

$$g_\theta(\sigma, \text{grad}F(\theta)) = (\nabla_\theta F(\theta), \sigma).$$

In other words,

$$\dot\theta^\mathsf{T} G(\theta)\text{grad}F(\theta) = \nabla_\theta F(\theta)^\mathsf{T}\sigma.$$

Since $\theta \subset \mathbb{R}^d$ and $G(\theta)$ is positive definite, then

$$\text{grad}F(\theta) = G(\theta)^{-1}\nabla_\theta F(\theta).$$

Proof of Proposition 3. We next present the derivation of the proposed semi-backward method.

Claim: Denote $\|\theta - \theta^k\| = h$, then

$$(\theta^k - \theta)^\mathsf{T} G(\tilde\theta)(\theta^k - \theta) = \text{Dist}(\theta, \theta^k)^2 + o(h^2), \quad (10)$$

and

$$\begin{aligned} &\frac{1}{2}(\theta^k - \theta)^\mathsf{T} G(\tilde\theta)(\theta^k - \theta) + O(h^2) \\ =& \sup_\Phi \int_{\mathbb{R}^n} \Phi(x)(\rho(\theta, x) - \rho(\theta^k, x)) \\ &- \frac{1}{2}\|\nabla\Phi(x)\|^2 \rho(\tilde\theta, x)dx. \end{aligned} \quad (11)$$

Proof of Claim. We next prove the claim. Denote the geodesic path $\theta^*(t)$, $t \in [0,1]$, with $\theta^*(0) = \theta$, $\theta^*(1) = \theta^k$, s.t.

$$\text{Dist}(\theta, \theta^k)^2 = \int_0^1 (\frac{d}{dt}\theta^*(t))^\mathsf{T} G(\theta^*(t))\frac{d}{dt}\theta^*(t)dt.$$

We reparameterize the time of $\theta^*(t)$ into the time interval $[0, h]$. Denote $\tau = ht$ and $\theta(\tau) = \theta^*(ht)$. Thus $\theta(\tau) = \theta^k + \frac{\theta - \theta^k}{h}\tau + O(\tau^2)$ and $\frac{d}{d\tau}\theta(\tau) = \frac{\theta - \theta^k}{h} + O(\tau)$,

$$\begin{aligned} &\text{Dist}(\theta, \theta^k)^2 \\ =& h\int_0^h \frac{d}{d\tau}\theta(\tau)^\mathsf{T} G(\theta(\tau))\frac{d}{d\tau}\theta(\tau)d\tau \\ =& h\int_0^h (\frac{\theta - \theta^k}{h} + O(h))^\mathsf{T} G(\tilde\theta + O(h))(\frac{\theta - \theta^k}{h} + O(h))d\tau \\ =& (\theta - \theta^k)^\mathsf{T} G(\tilde\theta)(\theta - \theta^k) + o(h^2), \end{aligned}$$

which proves equation (10).

We next prove equation (11). On the L.H.S. of equation (11),

$$\nabla_\theta \rho(\tilde\theta, x)(\theta - \theta^k) = \rho(\theta, x) - \rho(\theta^k, x) + o(h).$$

From the definition of $G(\theta)$,

$$\begin{aligned} &\frac{1}{2}(\theta - \theta^k)^\mathsf{T} G(\tilde\theta)(\theta - \theta^k) \\ &= \frac{1}{2}\int_{\mathbb{R}^n} \|\nabla\Phi(x)\|^2 \rho(\tilde\theta, x)\, dx, \end{aligned}$$

where

$$\begin{aligned} -\nabla\cdot(\rho(\tilde\theta, x)\nabla\Phi(x)) &= \nabla_\theta\rho(\tilde\theta, x)(\theta - \theta^k) \\ &= \rho(\tilde\theta, x) + o(h). \end{aligned}$$

On the R.H.S. of equation (11), the maximizer Φ^* satisfies

$$\rho(\theta, x) - \rho(\theta^k, x) + \nabla\cdot(\rho(\tilde\theta, x)\nabla\Phi^*(x)) = 0. \quad (12)$$

Applying equation (12) into the R.H.S. of (11), we have

$$\begin{aligned}
&\int_{\mathbb{R}^n} \Phi^*(x)(\rho(\theta,x)-\rho(\tilde{\theta},x)) - \frac{1}{2}\|\nabla\Phi^*(x)\|^2\rho(\tilde{\theta},x)\,dx\\
=&\int_{\mathbb{R}^n} \Phi^*(x)[-\nabla\cdot(\rho(\tilde{\theta},x)\nabla\Phi^*(x)]\\
&\quad -\frac{1}{2}\|\nabla\Phi^*(x)\|^2\rho(\tilde{\theta},x)\,dx\\
=&\int_{\mathbb{R}^n} \|\nabla\Phi^*(x)\|^2\rho(\tilde{\theta},x) - \frac{1}{2}\|\nabla\Phi^*(x)\|^2\rho(\tilde{\theta},x)\,dx\\
=&\frac{1}{2}\int_{\mathbb{R}^n} \|\nabla\Phi^*(x)\|^2\rho(\tilde{\theta},x)\,dx.
\end{aligned}$$

Comparing the L.H.S. and R.H.S. of (11), we prove the claim. From the claim,

$$\begin{aligned}
\theta^{k+1} =& \arg\min_{\theta\in\Theta} F(\theta) + \frac{1}{h}\frac{\mathrm{Dist}(\theta,\theta^k)^2}{2}\\
=& \arg\min_{\theta\in\Theta} F(\theta) + \frac{1}{2h}\Big\{(\theta^k-\theta)^{\mathsf{T}}G(\tilde{\theta})(\theta^k-\theta)+o(h^2)\Big\}\\
=& \arg\min_{\theta\in\Theta} F(\theta) + \frac{1}{h}\Big\{\sup_{\Phi}\int_{\mathbb{R}^n}\Phi(x)(\rho(\theta,x)-\rho(\theta^k,x))\\
&\qquad -\frac{1}{2}\|\nabla\Phi(x)\|^2\rho(\tilde{\theta},x)dx+o(h^2)\Big\}.
\end{aligned}$$

In above, we notice the fact that

$$\begin{aligned}
&(\theta^k-\theta)^{\mathsf{T}}G(\tilde{\theta})(\theta^k-\theta)+o(h^2)\\
=&\int_{\mathbb{R}^n}(\rho_{\theta^k}-\rho_\theta),\mathcal{G}(\rho_{\tilde{\theta}})(\rho_{\theta^k}-\rho_\theta)dx.
\end{aligned}$$

Thus we derive a consistent numerical method in time, known as the Semi-backward method:

$$\theta^{k+1} = \theta^k - hG(\tilde{\theta})^{-1}\nabla_\theta F(\theta^{k+1}) + o(h).$$

Proof of Theorem 4. For the constrained Wasserstein metric, we have the gradient w.r.t. the input space

$$\nabla\Phi = (\sum_j \xi_j\partial_i\psi_j(x))_{i=1}^n.$$

The norm is then

$$\begin{aligned}
\|\nabla\Phi\|^2 =& \sum_i(\sum_j\xi_j\partial_i\psi_j(x))^2 = \sum_i\sum_j\xi_j\partial_i\psi_j\sum_k\xi_k\partial_i\psi_k\\
=&\sum_j\sum_k\xi_j\xi_k(\sum_i\partial_i\psi_j(x)\partial_i\psi_k(x))\\
=&\xi^\top C(x)\xi
\end{aligned}$$

where $C_{ij}(x)=\sum_k\partial_k\psi_i\partial_k\psi_j$.

Now we consider the distance

$$\begin{aligned}
&\tilde{D}(\theta,\theta^k)^2\\
=&\sup_{\Phi\in\mathcal{F}_\xi}\int_{\mathbb{R}^n}\Phi(\rho_\theta-\rho_{\theta^k})dx-\frac{1}{2}\int_{\mathbb{R}^n}(\nabla\Phi)^2\rho_{\tilde{\theta}}dx\\
=&\sup_\xi \xi^\top(\mathbb{E}_\theta\psi-\mathbb{E}_{\theta^k}\psi)-\frac{1}{2}\xi^\top\mathbb{E}_{\tilde{\theta}}C\xi.
\end{aligned}$$

Here $\mathbb{E}_{\tilde{\theta}}C$ is a semi-positive definite matrix. Since for any $\xi\in\mathbb{R}^K$, we have

$$\xi^{\mathsf{T}}\mathbb{E}_{\tilde{\theta}}C\xi = \int_{\mathbb{R}^n}\sum_i(\sum_j\xi_j\partial_i\psi_j(x))^2\rho_\theta dx\geq 0.$$

Under the assumption that $\mathbb{E}_\theta C$ is invertible, then the optimization is a strictly concave problem. At the maximizer, we have

$$\xi^* = (\mathbb{E}_{\tilde{\theta}}C)^{-1}(\mathbb{E}_\theta\psi-\mathbb{E}_{\theta^k}\psi).$$

Thus

$$\tilde{D}(\theta,\theta^k)^2 = (\mathbb{E}_\theta\psi-\mathbb{E}_{\theta^k}\psi)^T(\mathbb{E}_{\tilde{\theta}}C)^{-1}(\mathbb{E}_\theta\psi-\mathbb{E}_{\theta^k}\psi),$$

which finishes the proof.

Derivation of SBE order 1. Here $\psi_j(x)=x_j$. Thus if $i=j$, then

$$\begin{aligned}
M_{ij}(\theta) =&\mathbb{E}_{z\sim p}\sum_{l=1}^n\partial_{x_l}\psi_i(x)\partial_{x_l}\psi_j(x)\\
=&1.
\end{aligned}$$

Otherwise, $M_{ij}(\theta)=0$, if $i\neq j$. Thus

$$M_{ij}(\tilde{\theta}) = \begin{cases}1 & \text{if } i=j;\\ 0 & \text{otherwise.}\end{cases}$$

Then we derive the result.

Proof of Theorem 5. The implicit model is given by the following push-forward relation. Denote $g_\theta\#p(z)=\rho(\theta,x)$, i.e.,

$$\int_{\mathbb{R}^m} f(g(\theta,z))p(z)dz = \int_{\mathbb{R}^n} f(x)\rho(\theta,x)dx,$$

for any $f\in C_c^\infty(\mathbb{R}^n)$.

We next rewrite the Wasserstein metric (9) in term of generators g_θ.

On the one hand, consider $f\in C_c^\infty(\mathbb{R}^n)$, then

$$\begin{aligned}
&\frac{d}{dt}\mathbb{E}_{Z\sim p(z)}f(g(\theta(t),Z))\\
=&\frac{d}{dt}\int_{\mathbb{R}^m} f(g(\theta(t),z))p(z)dz\\
=&\frac{d}{dt}\int_{\mathbb{R}^n} f(x)\rho(\theta(t),x)dx\\
=&\int_{\mathbb{R}^n} f(x)\frac{\partial}{\partial t}\rho(\theta(t),x)dx\\
=&\int_{\mathbb{R}^n} f(x)(-\nabla\cdot(\rho(\theta(t),x)\nabla\Phi(t,x)))dx\\
=&\int_{\mathbb{R}^n}\nabla f(x)\nabla\Phi(t,x)\rho(\theta(t),x)dx\\
=&\int\nabla f(g(\theta,z))\nabla\Phi(t,g(\theta,z))p(z)dz.
\end{aligned}\tag{13}$$

where the last equality holds from the push forward relation.

On the other hand, consider

$$\begin{aligned}
&\frac{d}{dt}\mathbb{E}_{Z\sim p(z)} f(g(\theta(t), Z)) \\
&= \lim_{\Delta t\to 0} \mathbb{E}_{Z\sim p(z)} \frac{f(g(\theta(t+\Delta t), Z) - f(g(\theta(t), Z))}{\Delta t} \\
&= \lim_{\Delta t\to 0} \int_{\mathbb{R}^m} \frac{f(g(\theta(t+\Delta t), z)) - f(g(\theta(t), z))}{\Delta t} p(z)dz \\
&= \int_{\mathbb{R}^m} \nabla f(g(\theta(t), z)) \frac{d}{dt} g(\theta(t), z) p(z)dz.
\end{aligned} \tag{14}$$

where ∇, $\nabla\cdot$ are gradient and divergence operators w.r.t. $x \in \mathbb{R}^n$. The second to last equality holds from the push forward relation, and the last equality holds using the integration by parts w.r.t. x. Since (13) equals (14) for any $f \in C_c^\infty(\mathbb{R}^n)$, then we have

$$\begin{aligned}
&\int \nabla f(g(\theta, z)) \nabla \Phi(t, g(\theta, z)) p(z)dz \\
=&\int \nabla f(g(\theta(t), z)) \frac{d}{dt} g(\theta(t), z) p(z)dz.
\end{aligned}$$

Thus

$$\int \nabla f(g(\theta, z)) \Big(\nabla \Phi(t, g(\theta, z)) - \frac{d}{dt} g(\theta(t), z) \Big) p(z)dz = 0.$$

If $n = 1$, then ∇f can be any function in $\mathbb{R}^1$. For each t, choosing $\nabla f(g(\theta, z)) = \nabla \Phi(t, g(\theta, z)) - \frac{d}{dt} g(\theta(t), z)$, then

$$\int |\nabla \Phi(t, g(\theta, z)) - \frac{d}{dt} g(\theta(t), z)|^2 p(z)dz = 0.$$

Thus

$$\frac{d}{dt} g(\theta(t), z) = \nabla \Phi(t, g(\theta(t), z)).$$

Thus by the definition of the push forward operation, we have

$$\begin{aligned}
&\mathbb{E}_{Z\sim p(z)} \| \frac{d}{dt} g(\theta(t), Z)\|^2 \\
=&\int_{\mathbb{R}^n} \|\nabla \Phi(t, g(\theta(t), z))\|^2 p(z)dz \\
=&\int_{\mathbb{R}^n} \|\nabla \Phi(t, x)\|^2 \rho(\theta(t), x)dx,
\end{aligned}$$

which finishes the proof.

Proof of Theorem 6. Here we only present the second order expansion of $\tilde{D}$. By taylor expansion, we simply check that

$$\tilde{D}(\theta, \theta + h) = h^\mathsf{T} G(\theta) h + o(h^2), \tag{15}$$

where

$$\begin{aligned}
\tilde{G}(\theta)_{ij} =& \Big\langle \mathbb{E}_{Z\sim p} \Psi(g(\theta, Z)) \nabla_{\theta_i} g(\theta, z), \\
& M(\theta) \mathbb{E}_{Z\sim p} \Psi(g(\theta, Z)) \nabla_{\theta_j} g(\theta, Z) \Big\rangle,
\end{aligned}$$

which is clear semi-positive definite. Similar as the proof in proposition 3, we know that the algorithm has the update

$$\theta^{k+1} = \theta^k - h G(\tilde{\theta})^\dagger \nabla_\theta F(\theta^{k+1}) + o(h).$$

This is the first order time discretization of gradient flow. We next check that

$$\frac{d}{dt} F(\theta(t)) = -\nabla_\theta F(\theta)^\mathsf{T} \tilde{G}(\theta)^\dagger \nabla_\theta F(\theta) \le 0.$$

We observe that $F(\theta)$ decreases along the gradient flow. Thus we finish the proof.

C. Order 2 Semi-backward Euler method

Here we consider the Order 2 semi-backeward Euler method. We approximate the potential Φ by a quadratic function, where the second order term is restricted to a diagonal matrix.

Proof of Order 2 Diagonal SBE Formula. Here we show that when we approximate

$$\Phi(x) = \frac{1}{2} x^T Q x + a^T x + b,$$

with a diagonal matrix $Q = \mathrm{diag}(q_1, \dots, q_N)$. We get that

$$\begin{aligned}
&\sup_{a,Q} \Phi(g(\theta, z)) - \Phi(g(\theta_{k-1}, (z)) - \frac{1}{2}\|\nabla \Phi(g(\theta_{k-1}, Z))\|^2 \\
=&\frac{1}{2}\|\mathbb{E}[g(\theta, z) - g(\theta_{k-1}, z) - Q g(\theta_{k-1}, z)]\|^2 \\
+&\frac{1}{2}\mathbb{E}[g(\theta, Z)^T Q g(\theta, Z)] - \frac{1}{2}\mathbb{E}[g(\theta_{k-1}, Z)^T Q g(\theta_{k-1}, Z)] \\
-&\frac{1}{2}\mathbb{E}[\|Q g(\theta_{k-1}, Z)\|^2]
\end{aligned}$$

which will be used in the O2Diag-SBE update. We note that $x = g(\theta, z)$ and $y = g(\theta_{k-1}, z)$. Then we have that the above becomes

$$\begin{aligned}
&\sup_{a,Q} \mathbb{E}_{x,y} \left[a^T(x - y) - \frac{1}{2} x^T Q x - \frac{1}{2} y^T Q y - \|a + Q y\|^2 \right] \\
&= \mathbb{E}_{x,y} \Bigg[a^T(x - y) - \frac{1}{2} \sum q_i x_i^2 - \frac{1}{2} q_i y_i^2 \\
&\qquad - \|a + \mathrm{diag}(q_1, \dots, q_N) y\|^2 \Bigg]
\end{aligned}$$

The above is a quadratic equation in a and $Q = \mathrm{diag}(q_1, \dots, q_N)$, so we can formulate the above into,

$$= (a, Q)\ell - \frac{1}{2}(a, Q) M (a, Q)^T$$

where $\ell = \Big(\mathbb{E}(x - y), \frac{1}{2}\mathbb{E}(x^2 - y^2) \Big)$ and

$$M = \frac{1}{B} \sum_{b=1}^{B} \begin{pmatrix} 1 \\ y_b \end{pmatrix} \begin{pmatrix} 1 \\ y_b \end{pmatrix}^T$$

which is the matrix for the quadratic term $\|a + \text{diag}(q_1, \ldots, q_N)y\|^2$. Then the maximum obtained is at,

$$(a^*, Q^*) = M^{-1}\ell$$

By explicitly computing the formula in Q^*, we finish the proof.

D. A practical description of the Wasserstein Proximal

As mentioned in Section 4.1, the Relaxed Wasserstein Proximal is meant to be an easy-to-implement, drop-in regularization. For instructional purposes, we take a specific example to showcase the algorithm: Relaxed Wasserstein Proximal on Vanilla GANs (with non-saturating gradient for the generator):

- Given:
  - A generator g_θ, and discriminator D_ω,
  - The distance function $F_\omega(g_\theta) = \mathbb{E}_{x\sim\text{real}}[\log(D_\omega(x))] - \mathbb{E}_{z\sim\mathcal{N}(0,1)}[\log(1 - D_\omega(g_\theta(z))]$,
  - Choice of optimizers, Adam_ω and Adam_θ,
  - Proximal step-sizes h, and generator iterations ℓ, and
  - Batch size B.

Then the algorithm follows:

1. Sample real data $\{x_i\}_{i=1}^B$, and latent data $\{z_i\}_{i=1}^B$.
2. Update the discriminator:

$$\omega^k \leftarrow \text{Adam}_\omega\bigg(-\frac{1}{B}\sum_{i=1}^B \log(D_\omega(x_i)) - \frac{1}{B}\sum_{i=1}^B \log(1 - D_\omega(g_\theta(z_i)))\bigg)$$

3. Sample latent data $\{z_i\}_{i=1}^B$
4. Perform Adam gradient descent ℓ number of times:

$$\theta^k \leftarrow \text{Adam}_\theta\bigg(-\frac{1}{B}\sum_{i=1}^B \log(D_\omega(g_\theta(z_i))) - \frac{1}{B}\sum_{i=1}^B \frac{1}{2h}\|g_\theta(z_i) - g_{\theta^{k-1}}(z_i)\|_2^2\bigg),$$

   for ℓ number of times.

5. Repeat the above until a chosen stopping condition (e.g. maximum number of iterations).

As one can analyze above, the only difference between the standard way of training GANs and using the Relaxed Wasserstein Proximal, are the $\|g_\theta(z_i) - g_{\theta^{k-1}}(z_i)\|_2^2$ terms and the number of generator iterations ℓ. Note that in this paper, we call a single loop of updating a discriminator a number of times and then updating the generator a number of a time, an outer-iteration.

E. Details on the Experiments

E.1. Hyperparameters for Relaxed Wasserstein Proximal experiments

The following hyperparameter settings for the RWP, Order-1 SBE, and Order-2 Diagonal SBE experiments in Section 4.1 are:

- A batch size of 64 for all experiments.
- For CIFAR-10 with WGAN-GP: The Adam optimizer with learning rate 0.0001, $\beta_1 = 0.5$, and $\beta_2 = 0.9$ for both the generator and discriminator. We used a latent space dimension of 128. For RWP, we used $h = 0.1$, and $\ell = 10$ generator iterations. For Order-1 SBE, we used $h = 0.5$, and $\ell = 5$. For Order-2 Diagonal SBE, we used $h = 0.2$ and $\ell = 5$.
- For CIFAR-10 with Vanilla and DRAGAN: The Adam optimizer with learning rate 0.0002, $\beta_1 = 0.1$, and $\beta_2 = 0.999$ for both the generator and discriminator. We used a latent space dimension of 100. For RWP, we used $h = 0.2$, and $\ell = 5$ generator iterations. For Order-1 SBE, we used $h = 0.2$ and $\ell = 5$. For Order-2 Diagonal SBE, we used $h = 0.2$ and $\ell = 5$
- For aligned and cropped CelebA with Vanilla: The Adam optimizer with learning rate 0.0002, $\beta_1 = 0.5$, and $\beta_2 = 0.999$ for both the generator and discriminator. We used a latent space dimension of 100 For RWP, we used $h = 0.2$, and $\ell = 5$ generator iterations. For Order-1 SBE, we used $h = 0.2$ and $\ell = 5$. For Order-2 Diagonal SBE, we used $h = 0.2$ and $\ell = 5$
- For aligned and cropped CelebA with WGAN-GP: The Adam optimizer with learning rate 0.0001, $\beta_1 = 0.5$, and $\beta_2 = 0.9$ for both the generator and discriminator. We used a latent space dimension of 128. For RWP, we used $h = 0.1$, and $\ell = 10$ generator iterations. For Order-1 SBE, we used $h = 0.5$ and $\ell = 5$, but we raised the number of discriminator iterations to 7 (as opposed to the usual 5. For Order-2 Diagonal SBE, we used $h = 0.2$ and $\ell = 5$
- For the high-learning rate for CelebA with WGAN-GP: The hyperparameters are the same as WGAN-GP except in the following: the learning rate is raised to

0.002, for RWP we have $h = 0.1$ and $\ell = 5$, for Order-1 SBE we have $h = 0.05$ and $\ell = 5$, for Order-2 Diagonal SBE we have $h = 0.05$ and $\ell = 3$

- For the high Adam β_1 momentum for CelebA with WGAN-GP: The hyperparameters are the same as WGAN-GP except in the following: the β_1 parameter is raised to 0.5 (as opposed to 0), for RWP we have $h = 0.1$ and $\ell = 10$, for Order-1 SBE we have $h = 0.05$ and $\ell = 5$, for Order-2 Diagonal SBE we have $h = 0.05$ and $\ell = 3$

F. Generated samples from the model

In Figure 5, we have samples generated from a Vanilla GAN with RWP regularization, trained on the CelebA dataset. The FID of these images was 17.105.

In Figure 6, we have samples generated from WGAN-GP with RWP , trained on the CIFAR-10 dataset. The FID for these images is 38.3.

Figure 5. A sample of images generated by RWP regularization on Vanilla GANs, on CelebA.

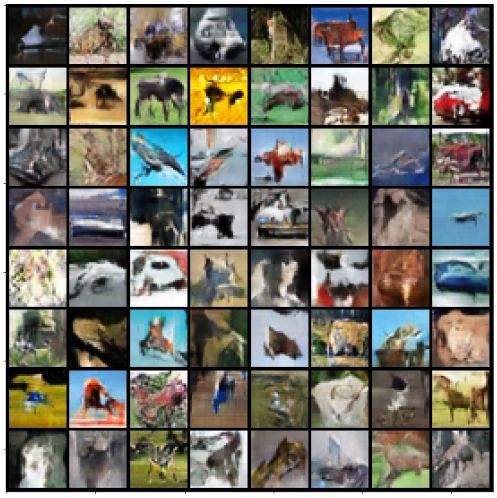

Figure 6. A sample of images generated by RWP regularization on WGAN-GP, on CIFAR-10.

G. Latent space walk

(Radford et al., 2015) suggest that walking in the latent space could detect whether a generator was memorizing. We see in Figure 7 and Figure 8 that we have smooth transitions, so this is not the case for GANs with RWP regularization. Results for Order-1 SBE, and Order-2 Diagonal SBE showed similar results.

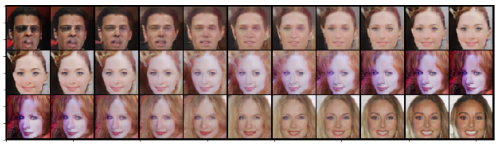

Figure 7. A latent space walk for a network with RWP regularization on Vanilla GANs, on CelebA. As we have smooth transitions, this shows the generator is not overfitting. The latent space walk is done by interpolating between 4 points in the latent space.

Figure 8. A latent space walk for a network with RWP regularization on WGAN-GP, on CIFAR-10. As we have smooth transitions, this shows the generator is not overfitting. The latent space walk is done by interpolating between 4 points in the latent space.