Learning Generative Models with Sinkhorn Divergences

Aude Genevay
CEREMADE,
Université Paris-Dauphine

Gabriel Peyré
CNRS and DMA,
École Normale Supérieure

Marco Cuturi
CREST ENSAE
Université Paris-Saclay

Abstract

The ability to compare two degenerate probability distributions, that is two distributions supported on low-dimensional manifolds in much higher-dimensional spaces, is a crucial factor in the estimation of generative models.It is therefore no surprise that optimal transport (OT) metrics and their ability to handle measures with non-overlapping supports have emerged as a promising tool. Yet, training generative machines using OT raises formidable computational and statistical challenges, because of *(i)* the computational burden of evaluating OT losses, *(ii)* their instability and lack of smoothness, *(iii)* the difficulty to estimate them, as well as their gradients, in high dimension. This paper presents the first tractable method to train large scale generative models using an OT-based loss called Sinkhorn loss which tackles these three issues by relying on two key ideas: *(a)* entropic smoothing, which turns the original OT loss into a differentiable and more robust quantity that can be computed using Sinkhorn fixed point iterations; *(b)* algorithmic (automatic) differentiation of these iterations with seamless GPU execution. Additionally, Entropic smoothing generates a family of losses interpolating between Wasserstein (OT) and Energy distance/Maximum Mean Discrepancy (MMD) losses, thus allowing to find a sweet spot leveraging the geometry of OT on the one hand, and the favorable high-dimensional sample complexity of MMD, which comes with unbiased gradient estimates.

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84.

1 Introduction

Several important statistical problems boil down to fitting densities, *i.e.* estimating the parameters of a chosen model that *fits* observed data in some meaningful way. While the standard approach is maximum likelihood estimation, this approach is often flawed in machine learning tasks where the sought after distribution is obtained in a generative fashion, *i.e.* described using a sampling mechanism (often a non-linear function mapping a low dimensional latent random vector to a high dimensional space). Indeed, in these settings, the density is singular in the sense that it only has positive probability on a low-dimensional manifold of the observation space and is zero elsewhere. To remedy these issues, and in line with several recent proposals [2, 26, 4, 1], we propose to shift away from information divergence based methods (among which the MLE) and consider instead the geometry of optimal transport [36, 31] to define such a fitting criterion.

Previous works. For purely generative models, several likelihood-free workarounds exist. Major approaches include variational autoencoders (VAE) [21], generative adversarial networks (GAN) [15] and several more variations including combinations of both [23]. The adversarial GAN approach is implicitly geometric in the sense that it computes the best achievable classification accuracy (taking for granted the training and generated datapoints have opposite labels) for a given class of classifiers as a proxy for the distance between two distributions: If accuracy is high distributions are well separated, if accuracy is low they are difficult to tell apart and lie thus at a very close distance.

Geometry was also explicitly considered when trying to minimize a flexible metric between distributions: the maximal mean discrepancy [16]. It was shown in ensuing works that the effectiveness of the MMD in that setting [25, 12] hinges on the ability to find a relevant RKHS bandwidth parameter, which is a highly nontrivial choice. The Wasserstein or earth mover's distance, long known to be a powerful tool to compare probability distributions with non-overlapping

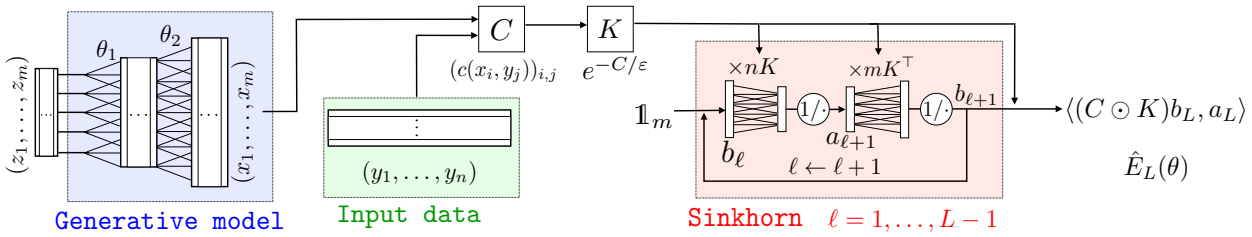


Figure 1: For a given fixed set of samples $(z_1, \ldots, z_m)$, and input data $(y_1, \ldots, y_n)$, flow diagram for the computation of Sinkhorn loss function $\theta \mapsto \hat{E}_\varepsilon^{(L)}(\theta)$. This function is the one on which automatic differentiation is applied to perform parameter learning. The display shows a simple 2-layer neural network $g_\theta : z \mapsto x$, but this applies to any generative model.

supports, has recently emerged as a serious contender to train generative models. While it was long disregarded because of its computational burden—in its original form solving OT amounts to solving an expensive network flow problem when comparing discrete measures in metric spaces—recent works have shown that this cost can be largely mitigated by settling for cheaper approximations obtained through strongly convex regularizers, in particular entropy [9, 14]. The benefits of this regularization has opened the path to many applications of the Wasserstein distance in relevant learning problems [8, 13, 18, 28]. Although the use of Wasserstein metrics for inference in generative models was considered over ten years ago in [2], that development remained exclusively theoretical until a recent wave of papers managed to implement that idea more or less faithfully using several workarounds: entropic regularization over a discrete space [26], approximate Bayesian computations [4] and a neural network parameterization of the dual potential arising from the dual OT problem when considering the 1-Wasserstein distance [1]. As opposed to this dual way to compute gradients of the fitting energy, we advocate for the use of a primal formulation, which is numerically stable, because it does not involve differentiating the (dual) solution of an OT sub-problem, as also pointed out in [5]. Additionally, introducing entropic regularization in the formulation of optimal transport allows to interpolate between a pure OT loss and a Maximum Mean Discrepency loss, thus bridging the gap between these two approaches often presented as opposed points of view. Shortly after the submission of this work, we came across the recent work by [30] which shares several ideas with our method. One distinction lies in the fact that they do not back-propagate errors across the Sinkhorn iterations, but rather use an estimate of the optimal transport matrix to compute an upper-bound on the Sinkhorn divergence, as was done for instance in [11]

Contributions. The main contributions of this paper are twofold : (i) a theoretical comtribution regarding a new OT-based loss for generative models, (ii) a simple numerical scheme to learn under this loss. (i) We introduce the Sinkhorn loss, based on regularized optimal transport with an entropy penalty, and we prove that when the smoothing parameter $\varepsilon = +0$ we recover pure OT loss whereas letting $\varepsilon = +\infty$ leads to MMD. The addition of entropy is important to reduce sample complexity and gradient bias, and thus allows us to take advantage of the good geometrical properties of OT without its drawbacks in high-dimensions. (ii) We propose a computationally tractable and stable approach to learn with that Sinkhorn loss, which enables inference for any differentiable generative model. It operates by adding L additional pooling layers (application of a filtering kernel K and pointwise divisive non-linearities), as illustrated on Figure (1). As routinely done in standard deep-learning architecture frameworks, the training is then achieved using stochastic gradient descent and automatic differentiation. This provides accurate and stable approximation of the loss and its gradient, at a reasonable extra computational cost, and streams nicely on GPU hardware.

Notations. For a matrix A, $A^\top$ denotes its transpose. For two vectors (or matrices) $\langle u, v \rangle \stackrel{\text{def.}}{=} \sum_i u_i v_i$ is the canonical inner product (the Frobenius dot-product for matrices). We define $\mathbb{1}_m \stackrel{\text{def.}}{=} (1/m, \ldots, 1/m) \in \mathbb{R}_+^m$ the uniform histogram, so that for $P \in \mathbb{R}^{n \times m}$, $P\mathbb{1}_m \in \mathbb{R}^n$ and $P^\top \mathbb{1}_n \in \mathbb{R}^m$ stand for the row and column averages of P. We denote $\mathcal{M}_+^1(\mathcal{X})$ the set of probability distributions (positive Radon measures of unit mass) over a metric space $\mathcal{X}$. δ_x stands for the Dirac (unit mass) distribution at point $x \in \mathcal{X}$. For some continuous map $g : \mathcal{Z} \to \mathcal{X}$, we denote $g_\sharp : \mathcal{M}_+^1(\mathcal{Z}) \to \mathcal{M}_+^1(\mathcal{X})$ the associated push-forward operator, which is a linear map between distributions. This corresponds to defining, for $\zeta \in \mathcal{M}_+^1(\mathcal{Z})$ and $B \subset \mathcal{X}$, $(g_\sharp \zeta)(B) = g^{-1}(B)$; or equivalently, that $\int_{\mathcal{X}} \varphi \mathrm{d}(g_\sharp \zeta) = \int_{\mathcal{Z}} \varphi \circ g \mathrm{d}\zeta$ for continuous functions φ on $\mathcal{X}$; or equivalently that a random sample x from $g_\sharp \zeta$ can be obtained as $x = g(z)$ where z is a random

sample from ζ.

2 Minimum Kantorovich Estimation

Density fitting. We consider a data set of N (usually very large) observations $(y_1, \dots, y_N) \in \mathcal{X}^N$ and we want to learn a generative model that produces samples that are similar to that dataset. Samples $x = g_\theta(z)$ from the generative model are defined by taking as input a sample $z \in \mathcal{Z}$ from some reference measure ζ (typically a uniform or a Gaussian measure in a low-dimensional space $\mathcal{Z}$) and mapping it through a differentiable function $g_\theta : \mathcal{Z} \to \mathcal{X}$. Formally, this corresponds to defining the generative model measure μ_θ from which x is drawn as $\mu_\theta = g_{\theta\#}\zeta$. Our goal is to find θ which minimizes a certain loss $\mathcal{L}$ between μ_θ and the empirical measure ν associated with the data

$$\theta \in \operatorname*{argmin}_\theta \mathcal{L}(\mu_\theta, \nu) \quad \text{where} \quad \nu \overset{\text{def.}}{=} \frac{1}{N}\sum_{j=1}^{N} \delta_{y_j}. \quad (1)$$

While we focus here for simplicity on the case of deterministic encoding functions g_θ between ζ and μ_θ, our method extends to more general probabilistic generative models, such as VAE [21].

Distances between measures. Maximum likelihood estimation (MLE) is obtained by setting $\mathcal{L}(\mu_\theta, \nu) = -\sum_j \log \frac{\mathrm{d}\mu_\theta}{\mathrm{d}x}(y_j)$, where $\frac{\mathrm{d}\mu}{\mathrm{d}x}$ is the density of μ_θ with respect to a fixed reference measure (a typical choice is $\mathrm{d}x$ being the Lebesgue measure in $\mathcal{X} = \mathbb{R}^d$). This MLE loss can be seen as a discretized version of the relative entropy (a.k.a. the Kullback-Leibler divergence). A major issue with this approach is that in general generative models defined this way (when $\mathcal{Z}$ has a much smaller dimensionality than $\mathcal{X}$) have singular distributions (*i.e.* supported on a low-dimensional manifold), without density with respect to a fixed measure, and therefore MLE cannot be considered.

The usual workaround is to assume that $\mathcal{X}$ is equipped with some distance $d_\mathcal{X}$, and consider weak metrics, which take into account spatial displacement of these measures, enabling the comparison of singular measures. A classical construction for such a loss function $\mathcal{L}$ is through duality (see *e.g* [34]), namely by considering a dual norm $\mathcal{L}(\mu, \nu) = \|\mu - \nu\|_B^*$ where $\|\xi\|_B^* = \sup\left\{\int_\mathcal{X} h(x)\mathrm{d}\xi(x) \;;\; h \in B\right\}$. Here B is a "unit ball" of continuous functions that should contain 0 in its interior. This ensures that $\|\cdot\|_B^*$ is well defined even for singular inputs, and it is a norm which metrizes the weak convergence of measures (so that for instance $\mathcal{L}(\delta_x, \delta_{x'}) \to 0$ as $x \to x'$), see [31, Sec.7.2.1] for more details. Classical instances of such settings include the 1-Wasserstein distance (obtained by setting $B = \{g \;;\; \|\nabla g\|_\infty \leqslant 1\}$ the set of 1-Lipschitz functions) and reproducing kernel Hilbert spaces (letting $B = \left\{g \;;\; \|k \star g\|_{L^2(\mathcal{X})} \leqslant 1\right\}$ where k is an appropriate convolution kernel). The latter define the class of Maximum Mean Discrepency losses [16] defined by

$$MMD_k(\mu, \nu) = \mathbb{E}_{\mu\otimes\mu}[k(X, X')] + \mathbb{E}_{\nu\otimes\nu}[k(Y, Y')] - 2\mathbb{E}_{\mu\otimes\nu}[k(X, Y)] \quad (2)$$

Optimal transport distances. In this article, we advocate for a different approach, which is to consider generic optimal transport (OT) metrics which can be used over general spaces $\mathcal{X}$ (not just the Euclidean space $\mathbb{R}^d$ and not only the 1-Wasserstein distance). The OT metric between two probability distributions $(\mu, \nu) \in \mathcal{M}_+^1(\mathcal{X}) \times \mathcal{M}_+^1(\mathcal{X})$ supported on two metric spaces $(\mathcal{X}, \mathcal{X})$ is defined as the solution of the (possibly infinite dimensional) linear program:

$$\mathcal{W}_c(\mu, \nu) \overset{\text{def.}}{=} \min_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{X}} c(x, y)\mathrm{d}\pi(x, y), \quad (3)$$

where the set of couplings is composed of joint probability distributions over the product space $\mathcal{X} \times \mathcal{X}$ with imposed marginals (μ, ν)

$$\Pi(\mu, \nu) \overset{\text{def.}}{=} \left\{\pi \in \mathcal{M}_+^1(\mathcal{X} \times \mathcal{X}) \;;\; P_{1\sharp}\pi = \mu, P_{2\sharp}\pi = \nu\right\},$$

where $P_1(x, y) = x, P_2(x, y) = y$ are simple projector operators. Formula (3) corresponds to the celebrated Kantorovitch formulation [19] of OT (see [31] for a detailed account on the theory). Here $c(x, y)$ is the "ground cost" to move a unit of mass from x to y, and we shall make no assumptions (except for regularity) on its form. When $\mathcal{X}$ is equipped with a distance $d_\mathcal{X}$, a typical choice is to set $c(x, y) = d_\mathcal{X}(x, y)^p$ where $p > 0$ is some exponent, in which case for $p \geqslant 1$ $\mathcal{W}_c^{1/p}$ is the so-called p-Wasserstein distance between probability measures.

We introduce the regularized optimal transport problem [9, 14] defined by

$$\min_{\pi\in\Pi(\mu,\nu)} \int c(x, y)\mathrm{d}\pi(x, y) + \varepsilon \int \log\left(\frac{\pi(x, y)}{\mathrm{d}\mu(x)\mathrm{d}\nu(y)}\right)\mathrm{d}\pi(x, y) \quad (\mathcal{P}_\varepsilon)$$

And the associated regularized Wasserstein distance associated with cost c and regularization paremeter ε is defined by:

$$\mathcal{W}_{c,\varepsilon}(\mu, \nu) = \int c(x, y)\mathrm{d}\pi_\varepsilon(x, y)$$

where π_ε is the optimal coupling for the regularized OT problem $(\mathcal{P}_\varepsilon)$.

Theorem 1 (Sinkhorn Loss)**.** *The Sinkhorn loss between two measures μ, ν is defined as:*

$$\bar{\mathcal{W}}_{c,\varepsilon}(\mu, \nu) = 2\mathcal{W}_{c,\varepsilon}(\mu, \nu) - \mathcal{W}_{c,\varepsilon}(\mu, \mu) - \mathcal{W}_{c,\varepsilon}(\nu, \nu). \quad (4)$$

with the following limiting behavior in ε:

1. *as* $\varepsilon \to 0$, $\quad \bar{\mathcal{W}}_{c,\varepsilon}(\mu,\nu) \to 2\mathcal{W}_c(\mu,\nu)$
2. *as* $\varepsilon \to +\infty$, $\quad \bar{\mathcal{W}}_{c,\varepsilon}(\mu,\nu) \to MMD_{-c}(\mu,\nu)$

where MMD_{-c} is the MMD distance whose kernel is the cost from the optimal transport problem.

Remark 1. This theorem is a generalization of [27, §3.3] for continuous measures.

Proof. 1. The first part of the assumption is well known, see for instance [7].

2. Letting ε go to infinity in the regularized OT problem amounts to finding the coupling with minimum entropy in the constraint set. The problem becomes $\min_{\pi\in\Pi(\mu,\nu)} \int log(\frac{\pi(x,y)}{\mathrm{d}\mu(x)\mathrm{d}\nu(y)})d\pi(x,y)$ where $\Pi(\mu,\nu)$ is the set of couplings with marginals μ and ν. Introducing Lagrange multipliers u and v for these constraints, the dual problem becomes $\max_{u,v} \int u(x)\mathrm{d}\mu(x) + \int v(y)\mathrm{d}\nu(y) - \int \exp(u(x)+v(y))\mathrm{d}\mu(x)\mathrm{d}\nu(y)$ and the primal-dual relation is given by $d\pi(x,y) = \exp(u(x)+v(y))\mathrm{d}\mu(x)\mathrm{d}\nu(y)$. Solving the dual gives $u = v = 0$ and thus the optimal coupling is simply the product of the marginals i.e. $\pi = \mu\otimes\nu$. □

The density fitting problem can be rewritten using the Sinkhorn divergence (4):

$$\min_\theta E_\varepsilon(\theta) \quad \text{where} \quad E_\varepsilon(\theta) \stackrel{\text{def.}}{=} \bar{\mathcal{W}}_{c,\varepsilon}(\mu_\theta,\nu).$$

A Discussion on OT *vs.* MMD As proved in Theroem 1, the Sinkhorn loss interpolates between a pure OT loss for $\varepsilon = 0$ and MMD losses for $\varepsilon = +\infty$. As such, when $\varepsilon \to +\infty$, our loss takes advantage of the good properties of MMD losses, and in particular a favorable sample complexity of $O(1/\sqrt{n})$ (decay rate of the approximation of the true loss with a mini-batch of size n) and unbiased gradient estimates when using mini-batches. Note that sample complexity estimates have not been proved for the Sinkhorn loss, but empirical evidence (see curves in supplementary material) shows that its behavior is similar to that of MMD when epsilon is not too small. In contrast, the unregularized OT loss suffers from a sample complexity of $O(1/n^{1/d})$, see [37] for a recent account on this point. Using MMD to train generative models has been shown to be successful in [12, 25]. The improved Wasserstein GAN approach [17] (which penalizes the squared norm of the gradient of the dual potential) is similar to an MMD (in fact a dual Sobolev norm). By tuning the ε parameter, our method is able to take the best of both worlds, to blend the non-flat geometry of OT with the high-dimensional rigidity of MMD losses. Additionally, the Sinkhorn loss, as is the case for the original OT problem, can be defined with any cost c, whereas MMD losses are only meaningful when used with positive definite kernels k. The postivity of the Sinkhorn loss is yet to be proved but empirical evidence (see supplementary) strongly points in that direction. Eventually, in the specific case where $c = \|\cdot\|_p$ for $1 < p < 2$, the associated MMD loss is the energy distance [35]. It was also used to fit generative models in [3], while [24] uses MMD with a gaussian kernel. Note that contrary to what [3] claims, the energy distance cannot be presented as a cure to solve the bias of OT estimation in high-dimension, since the two distances are fundamentally different.

3 Sinkhorn AutoDiff Algorithm

Computing an approximation of ∇E is itself a difficult problem, even when $\varepsilon = 0$. In the latter case, a workaround is to use, instead of differentiating the "primal" formula (3), the optimum of the "dual" formula, resulting in $\nabla E_0(\theta) = \int_{\mathcal{Z}} \nabla[h\circ g_\theta](z)\mathrm{d}\zeta(z)$, where h is an optimal dual continuous potential for $\mu = \mu_\theta$, see [1]. This requires the use of approximate semi-discrete OT solvers (because μ_θ is a continuous measure while ν is discrete), which typically operate by approximating the continuous dual potential h, see for instance [14] which uses an RKHS expansion, or [1] which uses a deep-network expansion. While the dual formalism is appealing (in particular because it involves only integration over $\mathcal{Z}$ and not the product space $\mathcal{Z}\times\mathcal{X}$), the resulting gradient formula requires differentiating the dual potential, which tends to be difficult to compute and unstable. A very similar conclusion is reached by [5] (see in particular their Proposition 3).

We propose a different route, by making two key simplifications: (i) approximate the function $E_\varepsilon(\theta)$ by a size-(m,n) mini-batch sampling $\hat{E}_\varepsilon(\theta)$ to make it amenable to stochastic gradient descent ; (ii) approximate $\hat{E}_\varepsilon(\theta)$ by L-steps of the Sinkhorn algorithm [9] to obtain an algorithmic loss $\hat{E}^{(L)}_\varepsilon(\theta)$ which is amenable to automatic differentiation.

(i) Mini-batch sampling loss. We replace the initial functional $E_\varepsilon(\theta)$ by an expectation over mini-batches of size (m,n), with leads to consider

$$\min_\theta \mathbb{E}(\hat{E}_\varepsilon(\theta)) \quad \text{where} \quad \hat{E}_\varepsilon(\theta) \stackrel{\text{def.}}{=} \mathcal{W}_{c,\varepsilon}(\hat{\mu}_\theta,\hat{\nu}) \qquad (5)$$

$$\text{and} \quad \left\{ \begin{array}{l} \hat{\mu}_\theta \stackrel{\text{def.}}{=} \frac{1}{m}\sum_{i=1}^m \delta_{x_i}, \\ \hat{\nu} \stackrel{\text{def.}}{=} \frac{1}{n}\sum_{j\in J} \delta_{y_j}, \end{array} \right. \quad \left\{ \begin{array}{l} (z_i)_{i=1}^m \stackrel{\text{i.i.d}}{\sim} \zeta, \\ \forall\, i,\ x_i \stackrel{\text{def.}}{=} g_\theta(z_i), \end{array} \right.$$

The expectation is taken over the samples $(z_i)_{i=1}^m$ (drawn independently according to ζ) and the indexes

$J \subset \{1, \dots, N\}$ with $|J| = n$. As (m, n) increases, $\mathbb{E}(\hat{E}_\varepsilon)$ approaches E_ε, and convergence of minimizers is studied in [4].

At a given iterate of this stochastic gradient descent scheme (see pseudo-code 1), one draws a mini-batch $(z_i)_{i=1}^m \overset{\text{i.i.d}}{\sim} \zeta$ and a subset J of observations, and aims at computing the gradient of

$$\hat{E}(\theta) = \min_{P \in \mathbb{R}_+^{m \times n}} \left\{ \langle P, \hat{c} \rangle \; ; \; P \mathbb{1}_m = \mathbb{1}_n, P^\top \mathbb{1}_m = \mathbb{1}_n \right\}, \tag{6}$$

where we defined $\hat{c} \overset{\text{def.}}{=} \left[c(g_\theta(z_i), y_j) \right]_{i,j} \in \mathbb{R}^{m \times n}$ (which depends on θ because the x_i's do). Note that this is simply a rephrasing of (3) in the case where both input measures are discrete (sums of Dirac masses), so that couplings π can be treated as matrices $P \in \mathbb{R}^{m \times n}$, namely $\pi = \sum_{i,j} P_{i,j} \delta_{(z_i, y_j)} \in \mathcal{M}_+^1(\mathcal{Z} \times \mathcal{X})$.

(ii) Sinkhorn iterates. One major advantage of regularizing the optimal transport problem is that it becomes solvable efficiently using Sinkhorn's algorithm [32] (when dealing with discrete measures), and leads to a differentiable loss function (as first noticed in [9, 10]). Such a regularization is known to be equivalent to restricting the search space in (6) to couplings having the so-called scaling form

$$P_{i,j} = a_i K_{i,j} b_j \quad \text{where} \quad K_{i,j} \overset{\text{def.}}{=} e^{-\hat{c}_{i,j}/\varepsilon}.$$

Matrix K is the so-called Gibbs kernel. Note that K depends implicitly on θ (because matrix $\hat{c}$ does), and contains therefore all of the geometric information related to the ability of θ to sample points near the dataset. Starting with $b^{(0)} \overset{\text{def.}}{=} \mathbb{1}_m$, $\ell \leftarrow 0$, Sinkhorn iterates read

$$a_{\ell+1} \overset{\text{def.}}{=} \frac{\mathbb{1}_n}{K b_\ell} \quad \text{and} \quad b_{\ell+1} \overset{\text{def.}}{=} \frac{\mathbb{1}_m}{K^\top a_{\ell+1}} \tag{7}$$

where $\frac{\cdot}{\cdot}$ denotes component-wise division. The main computational burden of (7) are the matrix-vector multiplication, which stream extremely well on GPU architectures, and therefore nicely add to a typical deep network architecture with L additional layer of linear operations (K can be interpreted as a localized linear filtering) and entry-wise non-linear operations (here divisions).

For a given budget L of iterations, our final loss is then obtained by using $P_L \overset{\text{def.}}{=} \operatorname{diag}(a_L) K \operatorname{diag}(b_L)$ as a proxy for the optimal transport coupling, and thus

$$\hat{E}_\varepsilon^{(L)}(\theta) \overset{\text{def.}}{=} \langle \hat{c}, P_L \rangle = \sum_{i=1}^m \sum_{j=1}^n \hat{c}_{i,j} a_{L,i} b_{L,j} K_{i,j} \tag{8}$$

where it is once again important to remind that $K, \hat{c}, b_L, a_L$ depend on θ. As $\varepsilon \to 0$ and $L \to +\infty$, one can show that the P_L computed by Sinkhorn's iterates approaches a solution to (6), with linear convergence rate (deteriorating as $\varepsilon \to 0$), so that $\hat{E}_\varepsilon^{(L)}(\theta)$ is a smooth proxy for $E_\varepsilon(\theta)$ which can be differentiated in a fast and stable way, while being accurate as $\varepsilon \to 0$ and (m, n, L) increase. It is however important to realize that for large scale and high dimensional learning applications, empirical considerations [9, 22, 13] suggest that, unlike relevant applications of the same scheme in graphics [33], a relatively strong regularization—a large ε—leads not only to more stable results but also faster convergence, so that the value for L can be set quite low.

Learning the cost function Aside from the regularization parameter, a key element of the Sinkhorn loss is the choice of the ground cost c on the data space. In some cases, using a simple metric such as the L^2 norm is sufficient to compare two data points, but when dealing with high-dimensional objects, choosing c is more critical. In such cases, we propose to learn the cost c with the following parametrization

$$c_\varphi(x, y) \overset{\text{def.}}{=} \|f_\varphi(x) - f_\varphi(y)\| \quad \text{where} \quad f_\varphi : \mathcal{X} \to \mathbb{R}^p,$$

where f_φ can be modelled by a neural network, and can be seen as a feature extractor that reduces the dimensionality of $\mathcal{X}$ through a mapping onto $\mathbb{R}^p$.

Learning the cost function here is very similar to learning a parametric kernel in an MMD model, as done in [24]. The optimization problem becomes a min-max problem over (θ, φ) instead of a simple minimization problem over θ

$$\min_\theta \max_\varphi \bar{\mathcal{W}}_{c_\varphi, \varepsilon(\mu_\theta, \nu)}$$

where in practice $\bar{\mathcal{W}}_{c_\varphi, \varepsilon}$ is approximated by minibatches and Sinkhorn, as mentioned above.

Putting everything together. We can now describe efficiently our scheme and use Figure 1 again for that purpose. In that figure, the generator (blue) and real data (green) parts are combined to compute a pairwise distance matrix $\hat{c}$. This matrix, as in MMD-GAN's approach [25] is all we need. We do, however, significantly depart from a "flat" MMD approach in the red block of the figure, in which a finite number of Sinkhorn steps are used to approximate the Wasserstein distance. These Sinkhorn steps are used to evaluate (forward pass) and compute the gradient (backward pass) of that proxy as described in Algorithm 2. Samples are repeatedly taken by taking push-forwards of samples of the initial measure in $\mathcal{Z}$ to perform SGD as described in Algorithm 1.

Note that the procedure $\texttt{AutoDiff}_\theta$ corresponds to classical reverse mode automatic differentiation of L steps

of the Sinkhorn iteration, and has therefore naturally the same complexity as Sinkhorn, *i.e.* $O(Lmn)$ operations, with an extra storage cost required to run the backward iteration with no additional computational overhead.

The training procedure is the same as [1],[24] and consists in alterning n_c optimisation steps to train the cost function f_φ and an optimisation step to train the generator g_θ. Following implementation advice from these papers,we clip the weights φ to ensure a bounded gradient in the maximization and use RMSProp as an optimizer.

Algorithm 1 SGD with Auto-diff

Input: θ_0, φ_0, $(y_j)_{j=1}^n$ (the real data), m(batch size), L (number of Sinkhorn iterations), ε (regularization parameter), α (learning rate)
Output: θ, φ
$\theta \leftarrow \theta_0$, $\varphi \leftarrow \varphi_0$,
for $k = 1, 2, \ldots$ **do**
  for $t = 1, 2, \ldots, n_c$ **do**
    Sample $(y_j)_{j=1}^m$ from the observations.
    Sample $(z_i)_{i=1}^m \overset{\text{i.i.d}}{\sim} \zeta$, $(x_i)_{i=1}^m \overset{\text{def.}}{=} g_\theta(z_1^m)$
    $\texttt{grad}_\varphi \leftarrow \texttt{AutoDiff}_\varphi\Big(2\hat{W}_{\varphi,\varepsilon}^{(L)}(x_1^m, y_1^m) - \hat{W}_{\varphi,\varepsilon}^{(L)}(x_1^m, x_1^m) - \hat{W}_{\varphi,\varepsilon}^{(L)}(y_1^m, y_1^m)\Big)$
    $\varphi \leftarrow \varphi + \alpha\texttt{RMSProp}(\texttt{grad}_\varphi)$.
    $\varphi \leftarrow \texttt{clip}(\varphi, -c, c)$
  end for
  Sample $(y_j)_{j=1}^m$ from the observations.
  Sample $(z_i)_{i=1}^m \overset{\text{i.i.d}}{\sim} \zeta$, $(x_i)_{i=1}^m \overset{\text{def.}}{=} g_\theta(z_1^m)$
  $\texttt{grad}_\theta \leftarrow \texttt{AutoDiff}_\theta\Big(2\hat{W}_{\varphi,\varepsilon}^{(L)}(x_1^m, y_1^m) - \hat{W}_{\varphi,\varepsilon}^{(L)}(x_1^m, x_1^m) - \hat{W}_{\varphi,\varepsilon}^{(L)}(y_1^m, y_1^m)\Big)$
  $\theta \leftarrow \theta - \alpha\texttt{RMSProp}(\texttt{grad}_\theta)$.
end for

Algorithm 2 Sinkhorn loss $\hat{W}_{\varphi,\varepsilon}^{(L)}(x_1^m, y_1^m)$

Input: $\theta, (x_i)_{i=1}^m, (y_j)_{j=1}^m, \varepsilon$
Output: w
$\forall\,(i,j)$, $\hat{c}_{i,j} \overset{\text{def.}}{=} \|f_\varphi(x_i) - f_\varphi(y_j)\|$
$K_{i,j} = e^{-\frac{\hat{c}_{i,j}}{\varepsilon}}$
$b \leftarrow \mathbb{1}_n$,
for $\ell = 1, 2, \ldots, L$ **do**
  $a \leftarrow \frac{\mathbb{1}_n}{Kb}$, $b \leftarrow \frac{\mathbb{1}_n}{K^T a}$
end for
return $w = \langle (K \odot \hat{c})b,\, a\rangle$ (see (8))

4 Applications

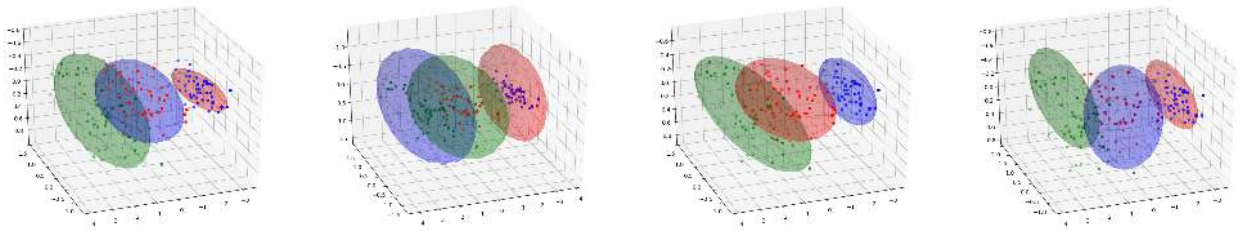
We consider two popular problems in machine learning to illustrate the versatility of our method. The first one relies on fitting labeled data with uniform distribution supported on ellipses (note that this could be any parametric shape but ellipses here were a good fit). The second problem consists in tuning a neural network to generate images, first with a fixed cost (on MNIST dataset) and then with a parametric cost (on CIFAR10 dataset). In both cases, we used simple initializations (see details below) and the algorithm yielded similar results when rerun, meaning that the results displayed are representative of the performance of the algorithm and that the procedure is quite stable.

4.1 Data Fitting with Ellipses.

As mentioned earlier, a strength of the Wasserstein distance is its ability to fit a singular probability distribution to an empirical measure (data). That singular probability may be supported on a subset of the space on a lower dimensional manifold, or simply have a degenerate density that becomes null for some subsets of the original space. To illustrate this principle, we consider in what follows a simple 3D example that can easily be visualized.

We use the Iris dataset (3 classes, 50 observations each in 4 dimensions) projected in 3D using PCA. This defines the empirical measure ν in $\mathbb{R}^3$. If we were to find a probability distribution μ_θ bound to be itself an empirical measure of K atoms (in that case parameter θ would contain exactly the locations of those K points in addition to their weight), then minimizing the 2-Wasserstein distance of μ_θ to ν would be *strictly equivalent* to the K-means problem [6]. In that sense, quantization can be regarded as the most elementary example of Wasserstein loss minimization of degenerate families of probability distributions.

The model we consider is instead composed of K ellipses with uniform density: Each ellipse is parametrized by a 3×3 matrix A_k (the square root of its covariance matrix) and a center $\alpha_k \in \mathbb{R}^3$, so that $\theta = (A_k, \alpha_k)_k$. Therefore, our results can't be directly compared to that of clustering algorithms, in the sense that we do automatically recover, within such ellipses, entire areas of interest (and not voronoi cells). We assume in this illustration that each ellipse has equal mass $1/K$. To recover these ellipses through a push forward, we use a uniform ground density ζ over K centred unit balls, translated and dilated for each ellipse using the push-forward defined by $g_\theta(z) = A_k z + \alpha_k$ if z is in the k-th ball.



(a) MMD (b) $\varepsilon = 1$ (c) $\varepsilon = 0.1$ (d) $\varepsilon = 0.01$

Figure 2: Ellipses after convergence of the stochastic gradient descent with $L = 20$, $m = 200$

MMD			$\varepsilon = 1$			$\varepsilon = 0.1$			$\varepsilon = 0.001$		
36	0	0	50	0	0	44	0	0	33	0	0
0	39	13	0	50	38	0	38	5	0	37	3
0	11	42	0	36	47	0	8	40	0	12	25

Table 1: Evaluation of the fit after convergence of the algorithm : entry (i, j) corresponds to the number of points from class j that are inside ellipse i

Numerical Illustration. The fit is obtained using the cost $c(x, y) = \|x - y\|^2$, the ellipse matrices $(A_k)_k$ are all initialized with the identity matrix (which corresponds to the unit ball) and centers $(\alpha_k)_k$ are initialized with the K-means algorithm. We fixed a maximal buget of Sinkhorn iterations $L = 20$ to be competitive with MMD time-wise, with a minibatch size $m = 200$ for both algorithms. Figure 2 displays the results of our method for different values of ε and for MMD with a gaussian kernel (with manually tuned bandwith). The influence of the regularization parameter ε is crucial: too much regularization (large ε,(b)) leads to a loose fit of the data but not regularizing enough leads to very slow convergence of the Sinkhorn algorithm and also yeilds poor performance (d) or requires more cpu time if we increase the total iteration budget. Since the Iris data is labeled, we can asses the fit of the model by checking the class repartition in each ellipse, as summarized in table 1. Each entry (i, j) corresponds to the number of points from class j that are inside ellipse i (recall there are 50 points per class). The performance difference between MMD and Sinkhorn here is not obvious, once the bandwidth parameter of the kernel is carefully tuned, but we found out that this parameter was more sensitive than ε, as the range of values that yield acceptable results are smaller.

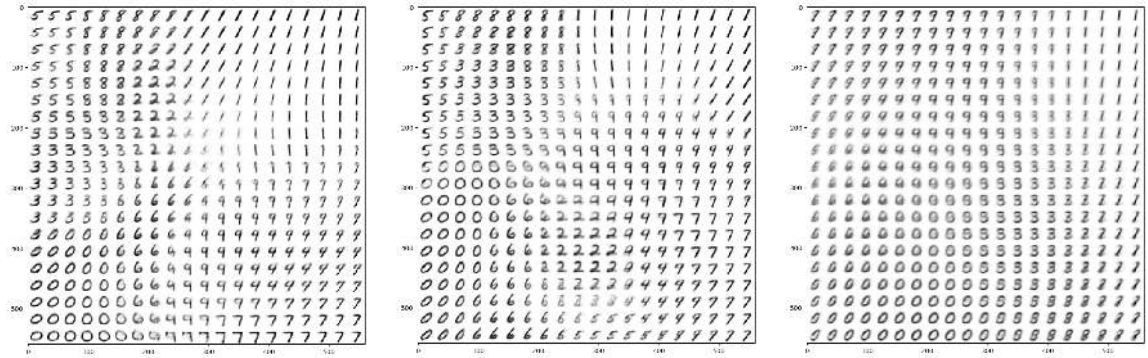
4.2 Tuning a Generative Neural Network

Image generating models such as GAN [15] or VAE [21] have become popular in recent years. The goal is to train a neural network g_θ which generates images $g_\theta(z)$ that resemble a certain data set $(y_j)_j$, given a random input z in a latent space $\mathcal{Z}$. Both methods require a second network for the training of the generative network (an adversial network in the case of GANs, an encoding network in the case of VAEs). Depending on the complexity of the data, our method can rely on the generative network alone by directly comparing its output with the data in Wasserstein distance.

With a fixed cost c This section fits a generative model where the pushforward g_θ is a multilayer perceptron. We begin with experiments on the MNIST dataset, which is a standard benchmark for this type of networks. Since the dataset is relatively simple, learning the cost is superfluous here and we use the ground cost $c(x, y) = \|x - y\|^2$, which is sufficient for these low resolution images and also the baseline in [21]. We use as g_θ a multilayer perceptron with 2 fully connected layers and the latent space is the unit square $\mathcal{Z} = [0, 1]^2$ over which we put a uniform distribution. Learning is performed in mini-batches over the MNIST dataset, with the Adam optimizer [20].

Figure 3 displays the manifold of images $g_\theta(z)$ generated by the optimized network (i.e. for equi-spaced $z \in [0, 1]^2$) after the learning procedure for different values of the hyperparameters (ε, m, L). This shows that the regularization parameter ε can be chosen quite large, which in turn leads to a fast convergence of Sinkhorn iterations. Indeed, using $\varepsilon = 1$ with only $L = 10$ Sinkhorn iterations (image (a)) yields a result similar to using $\varepsilon = 0.1$ with $L = 100$ iterations (image (b)). Regarding the size m of the mini-batches, a too small m value (e.g. $m = 10$) leads to poor results, and we observe that $m = 200$ is sufficient to learn accurately the manifold.

Learning the cost With higher-resolution datasets, such as classical benchmarks CIFAR10 or CelebA, using the ℓ^2 metric between images yields very poor results. It tends to generate images which are basically a blur of similar images. The alternative, already outlined in Algorithm 1 relies on learning another network which encodes meaningful feature vectors for the images, be-



(a) $\varepsilon = 1, m = 200, L = 10$ (b) $\varepsilon = 10^{-1}, m = 200, L = 100$ (c) $\varepsilon = 10^{-1}, m = 10, L = 300$

Figure 3: Influence of the hyperparameters on the manifold of generated digits.

(a) MMD (b) $\varepsilon = 100$ (c) $\varepsilon = 1$

Figure 4: Samples from the generator trained on CIFAR 10 for MMD and Sinkhorn loss

MMD	$\varepsilon = 100$	$\varepsilon = 10$	$\varepsilon = 1$
4.56 ± 0.07	4.81 ± 0.05	4.79 ± 0.13	4.43 ± 0.07

Table 2: Inception Scores

tween which can take the euclidean distance.

We compare our loss with different values for the regularization parameter ε to the results obtained with an MMD loss with a gaussian kernel. The experimental setting is the same as in [24] and we used the same parameters to carry out a fair comparison. We use small batches of 100 images.

Table 2 summarizes the inception scores on CIFAR10 for MMD and Sinkhorn loss with varying ε. Generative models are very hard to evaluate and there is no consensus on which metric should be used to assess their quality. We choose the inception score introduced in [29] as it is well spread, and also the reference in [12] agains which we compare our losses. The scores are evalutated on 20000 random images. Figure 4 displays a few of the associated samples (generated with the same seed). Although there is no striking difference in visual quality, the model with a Sinkhorn loss and a large regularization is the one with the best score. The decaying scores of models which have a loss closer to the true OT loss can be explained by two main factors : (i) the number of iterations required for the convergence of Sinkhorn with such ε might exceed the total iteration budget that we give the algorithm to compute the loss (to ensure reasonable training time of the model), (ii) it reflects the fact that sample complexity worsens when we get closer to OT metrics, and increasing the batch size might be beneficial in that case.

Conclusion

In this paper, we presented a new computational toolbox to train large scale generative models with the Sinkhorn divergence. Thanks to the combination of entropic smoothing and automatic differentiation, it makes optimal transport applicable in arbitrary complex generative model setups. Besides, we proved that this divergence interpolates between classical OT and MMD losses, benefiting from advantages of both frameworks. Future work should focus on theoretical properties of the Sinkhorn divergence, in particular sample complexity and positivity.

Acknowledgements

GP would like to acknowledge the support of ERC Consolidator grant NORIA. MC would like to thank Zaid Harchaoui, Sebastian Nowozin and Sidakpal Singh for several discussions and preliminary ideas on using regularized OT for GANs, and acknowledge the support of a Chaire d'excellence de l'Idex Paris Saclay.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[2] Federico Bassetti, Antonella Bodini, and Eugenio Regazzini. On minimum Kantorovich distance estimators. *Statistics & probability letters*, 76(12):1298–1302, 2006.

[3] Marc G Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The Cramer distance as a solution to biased Wasserstein gradients. *arXiv preprint arXiv:1705.10743*, 2017.

[4] Espen Bernton, Pierre E Jacob, Mathieu Gerber, and Christian P Robert. Inference in generative models using the wasserstein distance. *arXiv preprint arXiv:1701.05146*, 2017.

[5] Olivier Bousquet, Sylvain Gelly, Ilya Tolstikhin, Carl-Johann Simon-Gabriel, and Bernhard Schoelkopf. From optimal transport to generative modeling: the VEGAN cookbook. *arXiv preprint arXiv:1705.07642*, 2017.

[6] Guillermo Canas and Lorenzo Rosasco. Learning probability measures with respect to optimal transport metrics. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2492–2500. Curran Associates, Inc., 2012.

[7] Guillaume Carlier, Vincent Duval, Gabriel Peyré, and Bernhard Schmitzer. Convergence of entropic schemes for optimal transport and gradient flows. *SIAM Journal on Mathematical Analysis*, 49(2):1385–1418, 2017.

[8] Nicolas Courty, Rémi Flamary, and Devis Tuia. Domain adaptation with regularized optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 274–289. Springer, 2014.

[9] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Adv. in Neural Information Processing Systems*, pages 2292–2300, 2013.

[10] M. Cuturi and A. Doucet. Fast computation of Wasserstein barycenters. In *Proceedings of the 31st International Conference on Machine Learning (ICML), JMLR W&CP*, volume 32, 2014.

[11] Marco Cuturi and Arnaud Doucet. Fast computation of wasserstein barycenters. In *International Conference on Machine Learning*, pages 685–693, 2014.

[12] GK Dziugaite, DM Roy, and Z Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *Uncertainty in Artificial Intelligence-Proceedings of the 31st Conference, UAI 2015*, pages 258–267, 2015.

[13] C. Frogner, C. Zhang, H. Mobahi, M. Araya, and T. Poggio. Learning with a Wasserstein loss. In *Adv. in Neural Information Processing Systems*, pages 2044–2052, 2015.

[14] A. Genevay, M. Cuturi, G. Peyré, and F. Bach. Stochastic optimization for large-scale optimal transport. In D. D. Lee, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Proc. NIPS'16*, pages 3432–3440. Curran Associates, Inc., 2016.

[15] 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*, pages 2672–2680, 2014.

[16] Arthur Gretton, Karsten M Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola. A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pages 513–520, 2007.

[17] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of Wasserstein GANs. *arXiv preprint arXiv:1704.00028*, 2017.

[18] Gao Huang, Chuan Guo, Matt J Kusner, Yu Sun, Fei Sha, and Kilian Q Weinberger. Supervised word mover's distance. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4862–4870. Curran Associates, Inc., 2016.

[19] L. Kantorovich. On the transfer of masses (in Russian). *Doklady Akademii Nauk*, 37(2):227–229, 1942.

[20] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[22] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Q Weinberger. From word embeddings to document distances. In *Proc. of the 32nd Intern. Conf. on Machine Learning*, pages 957–966, 2015.

[23] Anders Boesen Lindbo Larsen, Soren Kaae Sonderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1558–1566, New York, New York, USA, 20–22 Jun 2016. PMLR.

[24] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. MMD GAN: Towards deeper understanding of moment matching network. *arXiv preprint arXiv:1705.08584*, 2017.

[25] Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1718–1727, 2015.

[26] G. Montavon, K.-R. Müller, and M. Cuturi. Wasserstein training of restricted Boltzmann machines. In *Adv. in Neural Information Processing Systems*, 2016.

[27] Aaditya Ramdas, Nicolas Garcia Trillos, and Marco Cuturi. On wasserstein two-sample testing and related families of nonparametric tests. *Entropy*, 19(2), 2017.

[28] Antoine Rolet, Marco Cuturi, and Gabriel Peyré. Fast dictionary learning with a smoothed wasserstein loss. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 630–638, Cadiz, Spain, 09–11 May 2016. PMLR.

[29] 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*, pages 2234–2242, 2016.

[30] Tim Salimans, Han Zhang, Alec Radford, and Dimitris Metaxas. Improving GANs using optimal transport. In *International Conference on Learning Representations*, 2018.

[31] F. Santambrogio. *Optimal Transport for applied mathematicians*, volume 87 of *Progress in Nonlinear Differential Equations and their applications*. Springer, 2015.

[32] R. Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Ann. Math. Statist.*, 35:876–879, 1964.

[33] J. Solomon, F. de Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas. Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (Proc. SIGGRAPH 2015)*, 2015.

[34] Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, Gert RG Lanckriet, et al. On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, 6:1550–1599, 2012.

[35] Gábor J Székely and Maria L Rizzo. Testing for equal distributions in high dimension. *InterStat*, 5(16.10), 2004.

[36] Cedric Villani. *Topics in C. Transportation*. Graduate studies in Math. AMS, 2003.

[37] Jonathan Weed and Francis Bach. Sharp asymptotic and finite-sample rates of convergence of empirical measures in wasserstein distance. *arXiv preprint arXiv:1707.00087*, 2017.