Interpolating between Optimal Transport and MMD using Sinkhorn Divergences

Jean Feydy
DMA, École Normale Supérieure
CMLA, ENS Paris-Saclay

Thibault Séjourné
DMA, École Normale Supérieure

François-Xavier Vialard
LIGM, UPEM

Shun-ichi Amari
Brain Science Institute, RIKEN

Alain Trouvé
CMLA, ENS Paris-Saclay

Gabriel Peyré
CNRS, DMA, École Normale Supérieure

Abstract

Comparing probability distributions is a fundamental problem in data sciences. Simple norms and divergences such as the total variation and the relative entropy only compare densities in a point-wise manner and fail to capture the geometric nature of the problem. In sharp contrast, Maximum Mean Discrepancies (MMD) and Optimal Transport distances (OT) are two classes of distances between measures that take into account the geometry of the underlying space and metrize the convergence in law.

This paper studies the Sinkhorn divergences, a family of geometric divergences that interpolates between MMD and OT. Relying on a new notion of geometric entropy, we provide theoretical guarantees for these divergences: positivity, convexity and metrization of the convergence in law. On the practical side, we detail a numerical scheme that enables the large scale application of these divergences for machine learning: on the GPU, gradients of the Sinkhorn loss can be computed for batches of a million samples.

1 Introduction

Countless methods in machine learning and imaging sciences rely on comparisons between probability distributions. With applications ranging from shape matching (Vaillant and Glaunès, 2005; Kaltenmark et al., 2017) to classification (Frogner et al., 2015) and generative model training (Goodfellow et al., 2014), a common setting is that of *measure fitting*: given a unit-mass, positive empirical distribution $\beta \in \mathcal{M}_1^+(\mathcal{X})$ on a feature space $\mathcal{X}$, a loss function $\mathrm{L} : \mathcal{M}_1^+(\mathcal{X}) \times \mathcal{M}_1^+(\mathcal{X}) \rightarrow \mathbb{R}$ and a model distribution $\alpha_\theta \in \mathcal{M}_1^+(\mathcal{X})$ parameterized by a vector θ, we strive to minimize $\theta \mapsto \mathrm{L}(\alpha_\theta, \beta)$ through gradient descent. Numerous papers focus on the construction of suitable models $\theta \mapsto \alpha_\theta$. But which loss function L should we use? If $\mathcal{X}$ is endowed with a *ground distance* $\mathrm{d} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_+$, taking it into account can make sense and help descent algorithm to overcome spurious local minima.

Geometric divergences for Machine Learning. Unfortunately, simple dissimilarities such as the Total Variation norm or the Kullback-Leibler relative entropy do not take into account the distance d on the feature space $\mathcal{X}$. As a result, they do not metrize the convergence in law (aka. the weak* topology of measures) and are unstable with respect to deformations of the distributions' supports. We recall that if $\mathcal{X}$ is compact, α_n converges weak* towards α (denoted $\alpha_n \rightharpoonup \alpha$) if for all continuous test functions $f \in \mathcal{C}(\mathcal{X})$, $\langle \alpha_n, f \rangle \rightarrow \langle \alpha, f \rangle$ where $\langle \alpha, f \rangle \overset{\text{def.}}{=} \int_{\mathcal{X}} f \mathrm{d}\alpha = \mathbb{E}[f(X)]$ for any random vector X with law α.

The two main classes of losses $\mathrm{L}(\alpha, \beta)$ which avoid these shortcomings are Optimal Transport distances and Maximum Mean Discrepancies: they are continuous with respect to the convergence in law and metrize its topology when the feature space $\mathcal{X}$ is compact. That is, $\alpha_n \rightharpoonup \alpha \Leftrightarrow \mathrm{L}(\alpha_n, \alpha) \rightarrow 0$. The main purpose of this paper is to study the theoretical properties of a new class of *geometric* divergences which interpolates between these two families and thus offers an extra degree of freedom through a parameter ε that can be cross-validated in typical learning scenarios.

1.1 Previous works

OT distances and entropic regularization. A first class of geometric distances between measures

Proceedings of the 22$^{\text{nd}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

is that of Optimal Transportation (OT) costs, which are computed as solutions of a linear program (Kantorovich, 1942) (see (1) below in the special case $\varepsilon = 0$). Enjoying many theoretical properties, these costs allow us to lift a "ground metric" on the feature space $\mathcal{X}$ towards a metric on the space $\mathcal{M}_+^1(\mathcal{X})$ of probability distributions (Santambrogio, 2015). OT distances (sometimes referred to as Earth Mover's Distances (Rubner et al., 2000)) are progressively being adopted as an effective tool in a wide range of situations, from computer graphics (Bonneel et al., 2016) to supervised learning (Frogner et al., 2015), unsupervised density fitting (Bassetti et al., 2006) and generative model learning (Montavon et al., 2016; Arjovsky et al., 2017; Salimans et al., 2018; Genevay et al., 2018; Sanjabi et al., 2018). However, in practice, solving the linear problem required to compute these OT distances is a challenging issue; many algorithms that leverage the properties of the underlying feature space $(\mathcal{X}, \mathrm{d})$ have thus been designed to accelerate the computations, see (Peyré and Cuturi, 2017) for an overview.

Out of this collection of methods, entropic regularization has recently emerged as a computationally efficient way of approximating OT costs. For $\varepsilon > 0$, we define

$$\mathrm{OT}_\varepsilon(\alpha,\beta) \overset{\text{def.}}{=} \min_{\pi_1=\alpha,\pi_2=\beta} \int_{\mathcal{X}^2} \mathrm{C}\,\mathrm{d}\pi + \varepsilon \mathrm{KL}(\pi|\alpha\otimes\beta) \quad (1)$$

$$\text{where} \quad \mathrm{KL}(\pi|\alpha\otimes\beta) \overset{\text{def.}}{=} \int_{\mathcal{X}^2} \log\left(\frac{\mathrm{d}\pi}{\mathrm{d}\alpha\mathrm{d}\beta}\right)\mathrm{d}\pi,$$

where $\mathrm{C}(x,y)$ is some symmetric positive cost function (we assume here that $\mathrm{C}(x,x) = 0$) and where the minimization is performed over coupling measures $\pi \in \mathcal{M}_1^+(\mathcal{X}^2)$ as (π_1, π_2) denotes the two marginals of π. Typically, $\mathrm{C}(x,y) = \|x-y\|^p$ on $\mathcal{X} \subset \mathbb{R}^\mathrm{D}$ and setting $\varepsilon = 0$ in (1) allows us to retrieve the Earth Mover ($p = 1$) or the quadratic Wasserstein-2 ($p = 2$) distances.

The idea of adding an entropic barrier $\mathrm{KL}(\,\cdot\,|\alpha\otimes\beta)$ to the original linear OT program can be traced back to Schrödinger's problem (Léonard, 2013) and has been used for instance in social sciences (Kosowsky and Yuille, 1994; Galichon and Salanié, 2010) and computer vision (Chui and Rangarajan, 2000). Crucially, as highlighted in (Cuturi, 2013), the smooth problem (1) can be solved efficiently on the GPU as soon as $\varepsilon > 0$: the celebrated Sinkhorn algorithm (detailed in Section 3) allows us to compute efficiently a smooth, geometric loss OT_ε between sampled measures.

MMD norms. Still, to define geometry-aware distances between measures, a simpler approach is to integrate a positive definite kernel $k(x,y)$ on the feature space $\mathcal{X}$. On a Euclidean feature space $\mathcal{X} \subset \mathbb{R}^\mathrm{D}$, we typically use RBF kernels such as the Gaussian kernel $k(x,y) = \exp(-\|x-y\|^2/2\sigma^2)$ or the energy distance (conditionally positive) kernel $k(x,y) = -\|x-y\|$. The kernel loss is then defined, for $\xi = \alpha - \beta$, as

$$\mathrm{L}_k(\alpha,\beta) \overset{\text{def.}}{=} \tfrac{1}{2}\|\xi\|_k^2 \overset{\text{def.}}{=} \tfrac{1}{2}\int_{\mathcal{X}^2} k(x,y)\,\mathrm{d}\xi(x)\mathrm{d}\xi(y). \quad (2)$$

If k is universal (Micchelli et al., 2006) (i.e. if the linear space spanned by functions $k(x,\cdot)$ is dense in $\mathcal{C}(\mathcal{X})$) or characteristic (Sriperumbudur et al., 2010), we know that $\|\cdot\|_k$ metrizes the convergence in law. Such Euclidean norms, introduced for shape matching in (Glaunes et al., 2004), are often referred to as "Maximum Mean Discrepancies" (MMD) (Gretton et al., 2007). They have been used extensively for generative model (GANs) fitting in machine learning (Li et al., 2015; Dziugaite et al., 2015). MMD norms are cheaper to compute than OT and have a smaller *sample complexity* – i.e. approximation error when sampling a distribution.

1.2 Interpolating between OT and MMD using Sinkhorn divergences

Unfortunately though, the "flat" geometry that MMDs induce on the space of probability measures $\mathcal{M}_1^+(\mathcal{X})$ does not faithfully lift the ground distance on $\mathcal{X}$. For instance, on $\mathcal{X} = \mathbb{R}^\mathrm{D}$, let us denote by α_τ the translation of α by $\tau \in \mathbb{R}^\mathrm{D}$, defined through $\langle \alpha_\tau, f\rangle = \langle \alpha, f(\cdot + \tau)\rangle$ for continuous functions $f \in \mathcal{C}(\mathbb{R}^\mathrm{D})$. Wasserstein distance discrepancies defined for $\mathrm{C}(x,y) = \|x-y\|^p$ are such that $\mathrm{OT}_0(\alpha,\alpha_\tau)^{\frac{1}{p}} = \|\tau\|$.

In sharp contrast, MMD norms rely on *convolutions* with a (Green) kernel and mimic electrostatic potentials (Schmaltz et al., 2010). In practice, as evidenced in Figure 5, we thus observe *vanishing gradients* next to the extreme points of the measures' supports: source particles *shield* each other, and we do not recover clean translations.

Sinkhorn divergences. On the one hand, OT losses have appealing *geometric* properties; on the other hand, cheap MMD norms scale up to large batches with a low sample complexity. Why not *interpolate* between them to get the best of both worlds?

Following (Genevay et al., 2018) (see also (Ramdas et al., 2017; Salimans et al., 2018; Sanjabi et al., 2018)) we consider a new cost built from OT_ε that we call a *Sinkhorn divergence*:

$$\mathrm{S}_\varepsilon(\alpha,\beta) \overset{\text{def.}}{=} \mathrm{OT}_\varepsilon(\alpha,\beta) - \tfrac{1}{2}\mathrm{OT}_\varepsilon(\alpha,\alpha) - \tfrac{1}{2}\mathrm{OT}_\varepsilon(\beta,\beta). \quad (3)$$

Such a formula satisfies $\mathrm{S}_\varepsilon(\beta,\beta) = 0$ and interpolates between OT and MMD (Ramdas et al., 2017):

$$\mathrm{OT}_0(\alpha,\beta) \xleftarrow{0\leftarrow\varepsilon} \mathrm{S}_\varepsilon(\alpha,\beta) \xrightarrow{\varepsilon\to+\infty} \tfrac{1}{2}\|\alpha-\beta\|_{-\mathrm{C}}^2. \quad (4)$$

The entropic bias. Why bother with the autocorrelation terms $\mathrm{OT}_\varepsilon(\alpha,\alpha)$ and $\mathrm{OT}_\varepsilon(\beta,\beta)$? For positive values of ε, in general, $\mathrm{OT}_\varepsilon(\beta,\beta) \neq 0$ so that minimizing $\mathrm{OT}_\varepsilon(\alpha,\beta)$ with respect to α results in a biased solution: as evidenced by Figure 1, the gradient of OT_ε drives α towards a shrunk measure whose support is *smaller* than that of the target measure β. This is most evident as ε tends to infinity: $\mathrm{OT}_\varepsilon(\alpha,\beta) \rightarrow \iint \mathrm{C}(x,y)\,\mathrm{d}\alpha(x)\,\mathrm{d}\beta(y)$, a quantity that is minimized if α is a Dirac distribution located at the median (*resp.* the mean) value of β if $\mathrm{C}(x,y) = \|x-y\|$ (*resp.* $\|x-y\|^2$).

In the general case, as showcased in (Chui and Rangarajan (2000), Figure 3), the blurry transport plan π_ε solution of the OT_ε problem (1) links source points to *fuzzy* sets of target points whose diameters are proportional to ε (*resp.* $\sqrt{\varepsilon}$). As we minimize the associated transport cost, we thus see the samples of α converge towards the median (*resp.* mean) of their ε-neighbors, *inside* the convex hull of β's support.

In the literature, the formula (3) has been introduced more or less empirically to fix the *entropic bias* present in the OT_ε cost: with a structure that mimics that of a squared kernel norm (2), it was assumed or conjectured that S_ε would define a positive definite loss function, suitable for applications in ML. This paper is all about *proving* that this is indeed what happens.

1.3 Contributions

The purpose of this paper is to show that the Sinkhorn divergences are convex, smooth, positive definite loss functions that metrize the convergence in law. Our main result is the theorem below, that ensures that one can indeed use S_ε as a *reliable* loss function for ML applications – whichever value of ε we pick.

Theorem 1. *Let $\mathcal{X}$ be a compact metric space with a Lipschitz cost function $\mathrm{C}(x,y)$ that induces, for $\varepsilon > 0$, a* positive universal *kernel $k_\varepsilon(x,y) \overset{\text{def.}}{=} \exp(-\mathrm{C}(x,y)/\varepsilon)$. Then, S_ε defines a symmetric positive definite, smooth loss function that is convex in each of its input variables. It also metrizes the convergence in law: for all probability Radon measures α and $\beta \in \mathcal{M}_1^+(\mathcal{X})$,*

$$0 \;=\mathrm{S}_\varepsilon(\beta,\beta) \;\leqslant\; \mathrm{S}_\varepsilon(\alpha,\beta), \tag{5}$$

$$\alpha=\beta \quad\Longleftrightarrow\quad \mathrm{S}_\varepsilon(\alpha,\beta)=0, \tag{6}$$

$$\alpha_n \rightharpoonup \alpha \quad\Longleftrightarrow\quad \mathrm{S}_\varepsilon(\alpha_n,\alpha)\rightarrow 0. \tag{7}$$

Notably, these results also hold for measures with bounded support on a Euclidean space $\mathcal{X} = \mathbb{R}^\mathrm{D}$ endowed with ground cost functions $\mathrm{C}(x,y) = \|x-y\|$ or $\mathrm{C}(x,y) = \|x-y\|^2$ – which induce Laplacian and Gaussian kernels respectively.

This theorem legitimizes the use of the *unbiased* Sinkhorn divergences S_ε instead of OT_ε in model-fitting applications. Indeed, computing S_ε is roughly as expensive as OT_ε (the computation of the corrective factors being cheap, as detailed in Section 3) and the "debiasing" formula (3) allows us to guarantee that the unique minimizer of $\alpha \mapsto \mathrm{S}_\varepsilon(\alpha,\beta)$ is the target distribution β (see Figure 1). Section 3 details how to implement these divergences efficiently: our algorithms scale up to millions of samples thanks to freely available GPU routines. To conclude, we showcase in Section 4 the typical behavior of S_ε compared with OT_ε and standard MMD losses.

2 Proof of Theorem 1

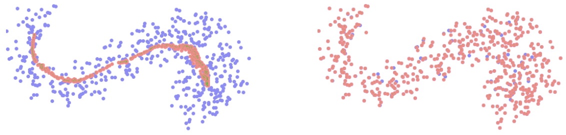
We now give the proof of Theorem 1. Our argument relies on a new Bregman divergence derived from a weak* continuous entropy that we call the *Sinkhorn entropy* (see Section 2.2). We believe this (convex) entropy function to be of independent interest. Note that all this section is written under the assumptions of Theorem 1; the proof of some intermediate results can be found in the appendix.

2.1 Properties of the OT_ε loss

First, let us recall some standard results of regularized OT theory (Peyré and Cuturi, 2017). Thanks to the Fenchel-Rockafellar theorem, we can rewrite Cuturi's loss (1) as

$$\mathrm{OT}_\varepsilon(\alpha,\beta) \overset{\text{def.}}{=} \max_{(f,g)\in\mathcal{C}(\mathcal{X})^2} \langle\alpha,f\rangle + \langle\beta,g\rangle \tag{8}$$
$$-\varepsilon\langle\alpha\otimes\beta, \exp\left(\tfrac{1}{\varepsilon}(f\oplus g-\mathrm{C})\right)-1\rangle,$$

where $f \oplus g$ is the tensor sum $(x,y)\in\mathcal{X}^2 \mapsto f(x)+g(y)$. The primal-dual relationship linking an optimal transport plan π solving (1) to an optimal dual pair



(a) $\mathrm{L} = \mathrm{OT}_\varepsilon$ (b) $\mathrm{L} = \mathrm{S}_\varepsilon$

Figure 1 – **Removing the entropic bias.** Solution α (in red) of the fitting problem $\min_\alpha \mathrm{L}(\alpha,\beta)$ for some β shown in blue. Here, $\mathrm{C}(x,y) = \|x-y\|$ on the unit square $\mathcal{X}$ in $\mathbb{R}^2$ and $\varepsilon = $ `.1`. The positions of the red dots were optimized by gradient descent, starting from a normal Gaussian sample.

(f, g) that solves (8) is

$$\pi = \exp\big(\tfrac{1}{\varepsilon}(f \oplus g - \mathrm{C})\big) \cdot (\alpha \otimes \beta). \tag{9}$$

Crucially, the first order optimality conditions for the dual variables are equivalent to the primal's marginal constraints ($\pi_1 = \alpha, \pi_2 = \beta$) on (9). They read

$$f = \mathrm{T}(\beta, g) \;\; \alpha\text{-a.e.} \quad \text{and} \quad g = \mathrm{T}(\alpha, f) \;\; \beta\text{-a.e.}, \tag{10}$$

where the "Sinkhorn mapping" $\mathrm{T} : \mathcal{M}_1^+(\mathcal{X}) \times \mathcal{C}(\mathcal{X}) \to \mathcal{C}(\mathcal{X})$ is defined through

$$\mathrm{T} : (\alpha, f) \mapsto \Big(y \in \mathcal{X} \mapsto \min_{x\sim\alpha}{}_{\varepsilon} \big[\mathrm{C}(x,y) - f(x)\big] \Big), \tag{11}$$

with a SoftMin operator of strength ε defined through

$$\min_{x\sim\alpha}{}_{\varepsilon}\, \varphi(x) \overset{\text{def.}}{=} -\varepsilon \log \int_{\mathcal{X}} \exp\big(-\tfrac{1}{\varepsilon}\varphi(x)\big)\, \mathrm{d}\alpha(x). \tag{12}$$

Dual potentials. The following proposition recalls some important properties of OT_ε and the associated dual potentials. Its proof can be found in Section B.1.

Proposition 1 (Properties of OT_ε)**.** *The optimal potentials (f, g) exist and are unique (α, β)-a.e. up to an additive constant, i.e. $\forall K \in \mathbb{R}$, $(f + K, g - K)$ is also optimal. At optimality, we get*

$$\mathrm{OT}_\varepsilon(\alpha, \beta) = \langle \alpha, f\rangle + \langle \beta, g\rangle. \tag{13}$$

We recall that a function $\mathrm{F} : \mathcal{M}_1^+(\mathcal{X}) \to \mathbb{R}$ is said to be *differentiable* if there exists $\nabla \mathrm{F}(\alpha) \in \mathcal{C}(\mathcal{X})$ such that for any displacement $\xi = \beta - \beta'$ with $(\beta, \beta') \in \mathcal{M}_1^+(\mathcal{X})^2$, we have

$$\mathrm{F}(\alpha + t\xi) = \mathrm{F}(\alpha) + t\langle\, \xi,\, \nabla\mathrm{F}(\alpha)\rangle + o(t). \tag{14}$$

The following proposition, whose proof is detailed in Section B.2, shows that the dual potentials are the gradients of OT_ε.

Proposition 2. OT_ε *is* weak* continuous *and* differentiable. *Its gradient reads*

$$\nabla \mathrm{OT}_\varepsilon(\alpha, \beta) = (f, g) \tag{15}$$

where (f, g) satisfies $f = \mathrm{T}(\beta, g)$ and $g = \mathrm{T}(\alpha, f)$ on the whole domain $\mathcal{X}$ and T is the Sinkhorn mapping (11).

Let us stress that even though the solutions of the dual problem (8) are defined (α, β)-a.e., the gradient (15) is defined on the whole domain $\mathcal{X}$. Fortunately, an optimal dual pair (f_0, g_0) defined (α, β)-a.e. satisfies the optimality condition (10) and can be *extended* in a canonical way: to compute the "gradient" pair $(f, g) \in \mathcal{C}(\mathcal{X})^2$ associated to a pair of measures (α, β), using $f = \mathrm{T}(\beta, g_0)$ and $g = \mathrm{T}(\alpha, f_0)$ is enough.

2.2 Sinkhorn and Haussdorf divergences

Having recalled some standard properties of OT_ε, let us now state a few *original* facts about the corrective, symmetric term $-\frac{1}{2}\mathrm{OT}_\varepsilon(\alpha, \alpha)$ used in (3). We still suppose that $(\mathcal{X}, \mathrm{d})$ is a compact set endowed with a symmetric, Lipschitz cost function $\mathrm{C}(x, y)$. For $\varepsilon > 0$, the associated *Gibbs kernel* is defined through

$$k_\varepsilon : (x, y) \in \mathcal{X} \times \mathcal{X} \mapsto \exp\big(-\mathrm{C}(x, y)/\varepsilon\big). \tag{16}$$

Crucially, we now assume that k_ε is a *positive universal* kernel on the space of signed Radon measures.

Definition 1 (Sinkhorn negentropy)**.** *Under the assumptions above, we define the Sinkhorn negentropy of a probability Radon measure $\alpha \in \mathcal{M}_1^+(\mathcal{X})$ through*

$$\mathrm{F}_\varepsilon(\alpha) \overset{\text{def.}}{=} -\tfrac{1}{2}\mathrm{OT}_\varepsilon(\alpha, \alpha). \tag{17}$$

The following proposition is the cornerstone of our approach to prove the positivity of S_ε, providing an alternative expression of F_ε. Its proof relies on a change of variables $\mu = \exp(f/\varepsilon)\,\alpha$ in (8) that is detailed in the Section B.3 of the appendix.

Proposition 3. *Let $(\mathcal{X}, \mathrm{d})$ be a compact set endowed with a symmetric, Lipschitz cost function $\mathrm{C}(x, y)$ that induces a* positive *kernel k_ε. Then, for $\varepsilon > 0$ and $\alpha \in \mathcal{M}_1^+(\mathcal{X})$, one has*

$$\tfrac{1}{\varepsilon}\mathrm{F}_\varepsilon(\alpha) + \tfrac{1}{2} = \min_{\mu \in \mathcal{M}^+(\mathcal{X})} \langle \alpha, \log \tfrac{\mathrm{d}\alpha}{\mathrm{d}\mu}\rangle + \tfrac{1}{2}\|\mu\|_{k_\varepsilon}^2. \tag{18}$$

The following proposition, whose proof can be found in the Section B.4 of the appendix, leverages the alternative expression (18) to ensure the convexity of F_ε.

Proposition 4. *Under the same hypotheses as Proposition 3,* F_ε *is a* strictly convex *functional on $\mathcal{M}_1^+(\mathcal{X})$.*

We now define an auxiliary "Hausdorff" divergence that can be interpreted as an OT_ε loss with *decoupled* dual potentials.

Definition 2 (Hausdorff divergence)**.** *Thanks to Proposition 2, the Sinkhorn negentropy* F_ε *is differentiable in the sense of* (14). *For any probability measures $\alpha, \beta \in \mathcal{M}_1^+(\mathcal{X})$ and regularization strength $\varepsilon > 0$, we can thus define*

$$\mathrm{H}_\varepsilon(\alpha, \beta) \overset{\text{def.}}{=} \tfrac{1}{2}\langle \alpha - \beta, \nabla\mathrm{F}_\varepsilon(\alpha) - \nabla\mathrm{F}_\varepsilon(\beta)\rangle \;\geqslant\; 0.$$

It is the symmetric Bregman divergence induced by the strictly convex functional F_ε (Bregman, 1967) and is therefore a positive definite quantity.

2.3 Proof of the Theorem

We are now ready to conclude. First, remark that the dual expression (8) of $\mathrm{OT}_\varepsilon(\alpha, \beta)$ as a maximization of linear forms ensures that $\mathrm{OT}_\varepsilon(\alpha,\beta)$ is *convex* with respect to α and with respect to β (but *not* jointly convex if $\varepsilon > 0$). S_ε is thus convex with respect to both inputs α and β as a sum of the functions OT_ε and F_ε – see Proposition 4.

Convexity also implies that,

$$\begin{aligned}\mathrm{OT}_\varepsilon(\alpha,\alpha)+\langle\beta-\alpha,\,\nabla_2\mathrm{OT}_\varepsilon(\alpha,\alpha)\rangle &\leqslant \mathrm{OT}_\varepsilon(\alpha,\beta),\\ \mathrm{OT}_\varepsilon(\beta,\beta)+\langle\alpha-\beta,\,\nabla_1\mathrm{OT}_\varepsilon(\beta,\beta)\rangle &\leqslant \mathrm{OT}_\varepsilon(\alpha,\beta).\end{aligned}$$

Using (15) to get $\nabla_2\mathrm{OT}_\varepsilon(\alpha,\alpha) = -\nabla\mathrm{F}_\varepsilon(\alpha)$, $\nabla_1\mathrm{OT}_\varepsilon(\beta,\beta) = -\nabla\mathrm{F}_\varepsilon(\beta)$ and summing the above inequalities, we show that $\mathrm{H}_\varepsilon \leqslant \mathrm{S}_\varepsilon$, which implies (5).

To prove (6), note that $\mathrm{S}_\varepsilon(\alpha,\beta)=0 \Rightarrow \mathrm{H}_\varepsilon(\alpha,\beta)=0$, which implies that $\alpha=\beta$ since F_ε is a *strictly* convex functional.

Finally, we show that S_ε metrizes the convergence in law (7) in the Section B.5 of the appendix.

3 Computational scheme

We have shown that Sinkhorn divergences (3) are positive definite, convex loss functions on the space of probability measures. Let us now detail their *implementation* on modern hardware.

Encoding measures. For the sake of simplicity, we focus on discrete, *sampled* measures on a Euclidean feature space $\mathcal{X} \subset \mathbb{R}^\mathrm{D}$. Our input measures α and $\beta \in \mathcal{M}_1^+(\mathcal{X})$ are represented as sums of weighted Dirac atoms

$$\alpha = \sum_{i=1}^{\mathrm{N}} \boldsymbol{\alpha}_i\,\delta_{\boldsymbol{x}_i}, \qquad \beta = \sum_{j=1}^{\mathrm{M}} \boldsymbol{\beta}_j\,\delta_{\boldsymbol{y}_j} \tag{19}$$

and encoded as two pairs $(\boldsymbol{\alpha},\boldsymbol{x})$ and $(\boldsymbol{\beta},\boldsymbol{y})$ of float arrays. Here, $\boldsymbol{\alpha} \in \mathbb{R}_+^\mathrm{N}$ and $\boldsymbol{\beta} \in \mathbb{R}_+^\mathrm{M}$ are *non-negative* vectors of shapes [N] and [M] that sum up to 1, whereas $\boldsymbol{x} \in (\mathbb{R}^\mathrm{D})^\mathrm{N}$ and $\boldsymbol{y} \in (\mathbb{R}^\mathrm{D})^\mathrm{M}$ are real-valued tensors of shapes [N, D] and [M, D].

3.1 The Sinkhorn algorithm(s)

Working with dual vectors. Proposition 1 is key to the modern theory of regularized Optimal Transport: it allows us to compute the OT_ε cost – and thus the Sinkhorn divergence S_ε, thanks to (3) – using dual variables that have the same *memory footprint* as the input measures: solving (8) in our discrete setting, we only need to store the *sampled values* of the dual potentials f and g on the measures' supports.

We can thus work with *dual vectors* $\boldsymbol{f} \in \mathbb{R}^\mathrm{N}$ and $\boldsymbol{g} \in \mathbb{R}^\mathrm{M}$, defined through $\boldsymbol{f}_i = f(\boldsymbol{x}_i)$ and $\boldsymbol{g}_j = g(\boldsymbol{y}_j)$, which encode an *implicit* transport plan π from α to β (9). Crucially, the optimality condition (10) now reads:

$\forall\, i \in [1,\mathrm{N}],\ \forall\, j \in [1,\mathrm{M}]$,

$$\boldsymbol{f}_i = -\varepsilon\,\mathrm{LSE}_{k=1}^{\mathrm{M}}\big(\log(\boldsymbol{\beta}_k) + \tfrac{1}{\varepsilon}\boldsymbol{g}_k - \tfrac{1}{\varepsilon}\mathrm{C}(\boldsymbol{x}_i,\boldsymbol{y}_k)\big) \tag{20}$$

$$\boldsymbol{g}_j = -\varepsilon\,\mathrm{LSE}_{k=1}^{\mathrm{N}}\big(\log(\boldsymbol{\alpha}_k) + \tfrac{1}{\varepsilon}\boldsymbol{f}_k - \tfrac{1}{\varepsilon}\mathrm{C}(\boldsymbol{x}_k,\boldsymbol{y}_j)\big) \tag{21}$$

$$\text{where}\qquad \mathrm{LSE}_{k=1}^{\mathrm{N}}(V_k) \;=\; \log\sum_{k=1}^{\mathrm{N}}\exp(V_k) \tag{22}$$

denotes a (stabilized) log-sum-exp reduction.

If $(\boldsymbol{f},\boldsymbol{g})$ is an optimal pair of dual vectors that satisfies Equations (20-21), we deduce from (13) that

$$\mathrm{OT}_\varepsilon(\boldsymbol{\alpha}_i,\boldsymbol{x}_i,\boldsymbol{\beta}_j,\boldsymbol{y}_j) = \sum_{i=1}^{\mathrm{N}}\boldsymbol{\alpha}_i\boldsymbol{f}_i + \sum_{j=1}^{\mathrm{M}}\boldsymbol{\beta}_j\boldsymbol{g}_j. \tag{23}$$

But how can we solve this *coupled* system of equations given $\boldsymbol{\alpha}$, $\boldsymbol{x}$, $\boldsymbol{\beta}$ and $\boldsymbol{y}$ as input data?

The Sinkhorn algorithm. One simple answer: by enforcing (20) and (21) alternatively, updating the vectors $\boldsymbol{f}$ and $\boldsymbol{g}$ until convergence (Kosowsky and Yuille, 1994; Cuturi, 2013). Starting from null potentials $\boldsymbol{f}_i = 0 = \boldsymbol{g}_j$, this numerical scheme is nothing but a block-coordinate ascent on the dual problem (8). One step after another, we are enforcing null derivatives on the dual cost with respect to the $\boldsymbol{f}_i$'s and the $\boldsymbol{g}_j$'s.

Convergence. The "Sinkhorn loop" converges quickly towards its unique optimal value: it enjoys a linear convergence rate (Peyré and Cuturi, 2017) that can be improved with an ε-scaling heuristic (Kosowsky and Yuille, 1994; Schmitzer, 2016). When computed through the *dual* expression (23), OT_ε and its gradients (26-27) are *robust* to small perturbations of the values of $\boldsymbol{f}$ and $\boldsymbol{g}$: monitoring convergence through the L^1 norm of the updates on $\boldsymbol{f}$ and breaking the loop as we reach a set tolerance level is thus a sensible stopping criterion.

Symmetric $\mathbf{OT}_\varepsilon$ problems. All in all, the baseline Sinkhorn loop provides an efficient way of solving the discrete problem $\mathrm{OT}_\varepsilon(\alpha,\beta)$ for generic input measures. But in the specific case of the (symmetric) corrective terms $\mathrm{OT}_\varepsilon(\alpha,\alpha)$ and $\mathrm{OT}_\varepsilon(\beta,\beta)$ introduced in (3), we can do better.

The key here is to remark that if $\alpha=\beta$, the dual problem (8) becomes a concave maximization problem that is *symmetric* with respect to its two variables f and g. Hence, there exists a (unique) optimal dual pair $(f, g=f)$ on the diagonal which is characterized in the discrete setting by the symmetric optimality condition:

$\forall i \in [1, \mathrm{N}]$,

$$\boldsymbol{f}_i = -\varepsilon\, \mathrm{LSE}_{k=1}^{\mathrm{N}} \big[\log(\boldsymbol{\alpha}_k) + \tfrac{1}{\varepsilon}\boldsymbol{f}_k - \tfrac{1}{\varepsilon}\mathrm{C}(\boldsymbol{x}_i, \boldsymbol{x}_k) \big]. \quad (24)$$

Fortunately, given $\boldsymbol{\alpha}$ and $\boldsymbol{x}$, the optimal vector $\boldsymbol{f}$ that solves this equation can be computed by iterating a *well-conditioned* fixed-point update which typically converges to satisfying precision in three iterations:

$$\boldsymbol{f}_i \leftarrow \tfrac{1}{2}\big(\boldsymbol{f}_i - \varepsilon\, \mathrm{LSE}_{k=1}^{\mathrm{N}} \big[\log(\boldsymbol{\alpha}_k) + \tfrac{1}{\varepsilon}\boldsymbol{f}_k - \tfrac{1}{\varepsilon}\mathrm{C}(\boldsymbol{x}_i, \boldsymbol{x}_k) \big] \big). \quad (25)$$

3.2 Computing the Sinkhorn divergence and its gradients

Given two pairs $(\boldsymbol{\alpha}, \boldsymbol{x})$ and $(\boldsymbol{\beta}, \boldsymbol{y})$ of float arrays that encode the probability measures α and β (19), we can now implement the Sinkhorn divergence $\mathrm{S}_\varepsilon(\alpha, \beta)$:

The *cross-correlation* dual vectors $\boldsymbol{f} \in \mathbb{R}^{\mathrm{N}}$ and $\boldsymbol{g} \in \mathbb{R}^{\mathrm{M}}$ associated to the discrete problem $\mathrm{OT}_\varepsilon(\alpha, \beta)$ can be computed using the Sinkhorn iterations (20-21).

The *autocorrelation* dual vectors $\boldsymbol{p} \in \mathbb{R}^{\mathrm{N}}$ and $\boldsymbol{q} \in \mathbb{R}^{\mathrm{M}}$, respectively associated to the symmetric problems $\mathrm{OT}_\varepsilon(\alpha, \alpha)$ and $\mathrm{OT}_\varepsilon(\beta, \beta)$, can be computed using the *symmetric* Sinkhorn update (25).

The Sinkhorn *loss* can be computed using (3) and (23):

$$\mathrm{S}_\varepsilon(\boldsymbol{\alpha}_i, \boldsymbol{x}_i, \boldsymbol{\beta}_j, \boldsymbol{y}_j) = \sum_{i=1}^{\mathrm{N}} \boldsymbol{\alpha}_i(\boldsymbol{f}_i - \boldsymbol{p}_i) + \sum_{j=1}^{\mathrm{M}} \boldsymbol{\beta}_j(\boldsymbol{g}_j - \boldsymbol{q}_j).$$

What about the gradients? In the past few years, authors have proposed to rely on the *automatic differentiation* engines provided by modern libraries, which let us differentiate the result of twenty or so Sinkhorn iterations as a mere composition of elementary operations (Genevay et al., 2018). But beware: this loop has a lot more *structure* than a generic feed forward network. Taking advantage of it is key to a x2-x3 gain in performances, as we now describe.

Crucially, we must remember that the Sinkhorn loop is a *fixed point* iterative solver: at convergence, its solution satisfies an equation given by the implicit function theorem. Thanks to (15), using the very definition of gradients in the space of probability measures (14) and the intermediate variables in the computation of $\mathrm{S}_\varepsilon(\alpha, \beta)$, we get that

$$\begin{aligned} \partial_{\boldsymbol{\alpha}_i} \mathbf{S}_\varepsilon(\boldsymbol{\alpha}_i, \boldsymbol{x}_i, \boldsymbol{\beta}_j, \boldsymbol{y}_j) &= \boldsymbol{f}_i - \boldsymbol{p}_i & (26) \\ \text{and } \tfrac{1}{\boldsymbol{\alpha}_i} \partial_{\boldsymbol{x}_i} \mathbf{S}_\varepsilon(\boldsymbol{\alpha}_i, \boldsymbol{x}_i, \boldsymbol{\beta}_j, \boldsymbol{y}_j) &= \nabla\varphi(\boldsymbol{x}_i), & (27) \end{aligned}$$

where $\varphi : \mathcal{X} \to \mathbb{R}$ is equal to $\boldsymbol{f}_i - \boldsymbol{p}_i$ on the $\boldsymbol{x}_i$'s and is defined through

$$\begin{aligned} \varphi(x) \;=\; &-\varepsilon \log \sum_{j=1}^{\mathrm{M}} \exp \big[\log(\boldsymbol{\beta}_j) + \tfrac{1}{\varepsilon}\boldsymbol{g}_j - \tfrac{1}{\varepsilon}\mathrm{C}(x, \boldsymbol{y}_j) \big] \\ &+ \varepsilon \log \sum_{i=1}^{\mathrm{N}} \exp \big[\log(\boldsymbol{\alpha}_i) + \tfrac{1}{\varepsilon}\boldsymbol{p}_i - \tfrac{1}{\varepsilon}\mathrm{C}(x, \boldsymbol{x}_i) \big]. \end{aligned}$$

Graph surgery with PyTorch. Assuming convergence in the Sinkhorn loops, it is thus possible to compute the gradients of S_ε *without having to backprop* through the twenty or so iterations of the Sinkhorn algorithm: we only have to differentiate the expression above with respect to x. But does it mean that we should differentiate C or the log-sum-exp operation by hand? Fortunately, no!

Modern libraries such as PyTorch (Paszke et al., 2017) are flexible enough to let us "hack" the naive `autograd` algorithm, and act as though the optimal dual vectors $\boldsymbol{f}_i, \boldsymbol{p}_i, \boldsymbol{g}_j$ and $\boldsymbol{q}_j$ did not depend on the input variables of S_ε. As documented in our reference code,

`github.com/jeanfeydy/global-divergences`,

an appropriate use of the `.detach()` method in PyTorch is enough to get the best of both worlds: an *automatic* differentiation engine that computes our gradients using the formula *at convergence* instead of the baseline backpropagation algorithm. All in all, as evidenced in Figure 3, this trick allows us to divide by a factor 2-3 the time needed to compute a Sinkhorn divergence and its gradient with respect to the $\boldsymbol{x}_i$'s.

3.3 Scaling up to large datasets

The Sinkhorn iterations rely on a single non-trivial operation: the log-sum-exp reduction (22). In the ML literature, this *SoftMax* operator is often understood as a row- or column-wise reduction that acts on $[\mathrm{N}, \mathrm{M}]$ matrices. But as we strive to implement the update rules (20-21) and (25) on the GPU, we can go further.

Batch computation. First, if the number of samples N and M in both measures is small enough, we can optimize the GPU usage by computing Sinkhorn divergences *by batches* of size B. In practice, this can be achieved by encoding the cost function C as a 3D tensor of size $[\mathrm{B}, \mathrm{N}, \mathrm{M}]$ made up of stacked matrices $(\mathrm{C}(\boldsymbol{x}_i, \boldsymbol{y}_j))_{i,j}$, while $\boldsymbol{f}$ and $\boldsymbol{g}$ become $[\mathrm{B}, \mathrm{N}]$ and $[\mathrm{B}, \mathrm{M}]$ tensors, respectively. Thanks to the broadcasting syntax supported by modern libraries, we can then compute, in parallel, loss values $\mathrm{S}_\varepsilon(\alpha_k, \beta_k)$ for k in $[1, \mathrm{B}]$.

Avoiding memory overflows. Unfortunately though, tensor-centric methods such as the one presented above cannot scale to measures sampled with large numbers N and M of Dirac atoms: as these numbers exceed 10,000, huge $[\mathrm{N}, \mathrm{M}]$ matrices stop fitting into GPU memories. To alleviate this problem, we leverage the KeOps library (Charlier et al., 2018) that provides *online* map-reduce routines on the GPU with full PyTorch integration. Performing online log-sum-exp reductions with a *running maximum*, the KeOps primitives allow us to compute Sinkhorn

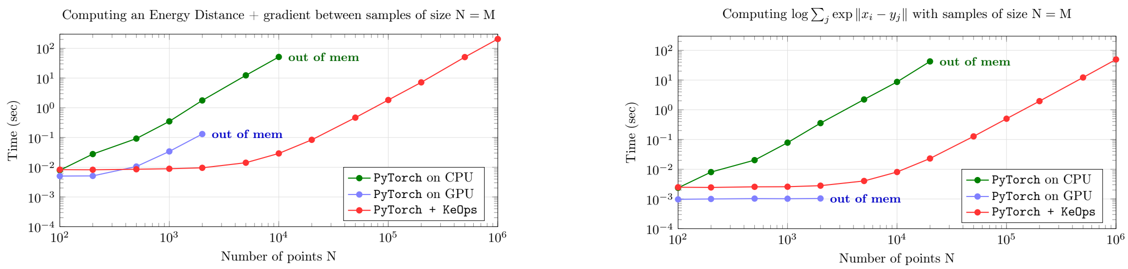


Figure 2 – **The KeOps library allows us to break the memory bottleneck.** Using CUDA routines that sum kernel values without storing them in memory, we can outperform baseline, tensorized, implementations of the energy distance. Experiments performed on $\mathcal{X} = \mathbb{R}^3$ with a cheap laptop's GPU (GTX 960M).

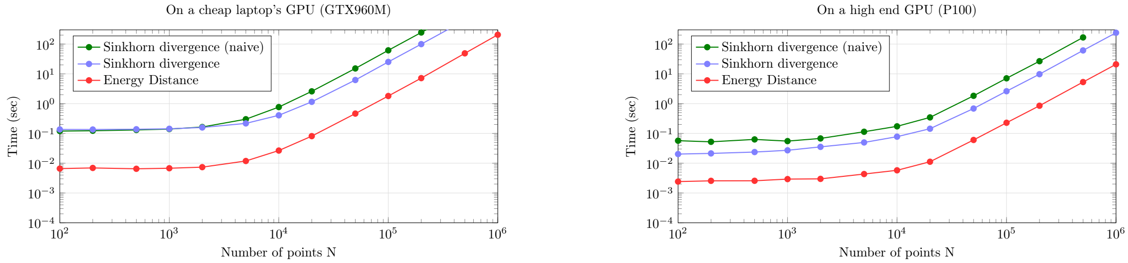


Figure 3 – **Sinkhorn divergences scale up to finely sampled distributions.** As a rule of thumb, Sinkhorn divergences take 20-50 times as long to compute as a baseline MMD – even though the explicit gradient formula (26-27) lets us win a factor 2-3 compared with a *naive* autograd implementation. For the sake of this benchmark, we ran the Sinkhorn and symmetric Sinkhorn loops with fixed numbers of iterations: 20 and 3 respectively, which is more than enough for measures on the unit (hyper)cube if $\varepsilon \geqslant .05$ – here, we work in $\mathbb{R}^3$.

divergences with a *linear* memory footprint. As evidenced by the benchmarks of Figures 2-3, computing the gradient of a Sinkhorn loss with 100,000 samples per measure is then a matter of seconds.

4 Numerical illustration

In the previous sections, we have provided theoretical guarantees on top of a comprehensive implementation guide for the family of *Sinkhorn divergences* S_ε. Let us now describe the *geometry* induced by these new loss functions on the space of probability measures.

Gradient flows. To compare MMD losses L_k with Cuturi's original cost OT_ε and the de-biased Sinkhorn divergence S_ε, a simple yet relevant experiment is to let a *model* distribution $\alpha(t)$ flow with time t along the "Wasserstein-2" gradient flow of a loss functional $\alpha \mapsto \mathrm{L}(\alpha, \beta)$ that drives it towards a target distribution β (Santambrogio, 2015). This corresponds to the "non-parametric" version of the data fitting problem evoked in Section 1, where the parameter θ is nothing but the vector of positions $\boldsymbol{x}$ that encodes the support of a measure $\alpha = \frac{1}{\mathrm{N}} \sum_{i=1}^{N} \delta_{\boldsymbol{x}_i}$. Understood as a "model free" idealization of fitting problems in machine learning, this experiment allows us to grasp the typical behavior of the loss function as we discover the deformations of the support that it favors.

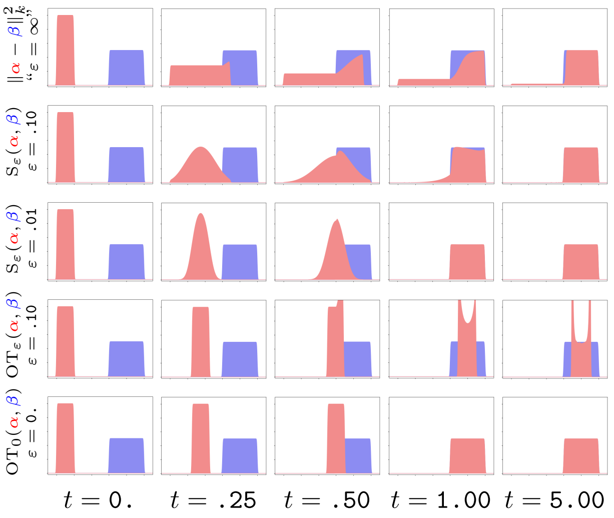


Figure 4 – Gradient flows for 1-D measures sampled with $\mathrm{N} = \mathrm{M} = 5000$ points – we display $\alpha(t)$ (in red) and β (in blue) through kernel density estimations on the segment $[0, 1]$. The legend on the left indicates the function that is minimized with respect to α. Here $k(x, y) = -\|x - y\|$, $\mathrm{C}(x, y) = \|x - y\|$ and $\varepsilon = .10$ on the second and fourth lines, $\varepsilon = .01$ on the third. In 1D, the optimal transport problem can be solved using a sort algorithm: for the sake of comparison, we can thus display the "true" dynamics of the Earth Mover's Distance in the fifth line.

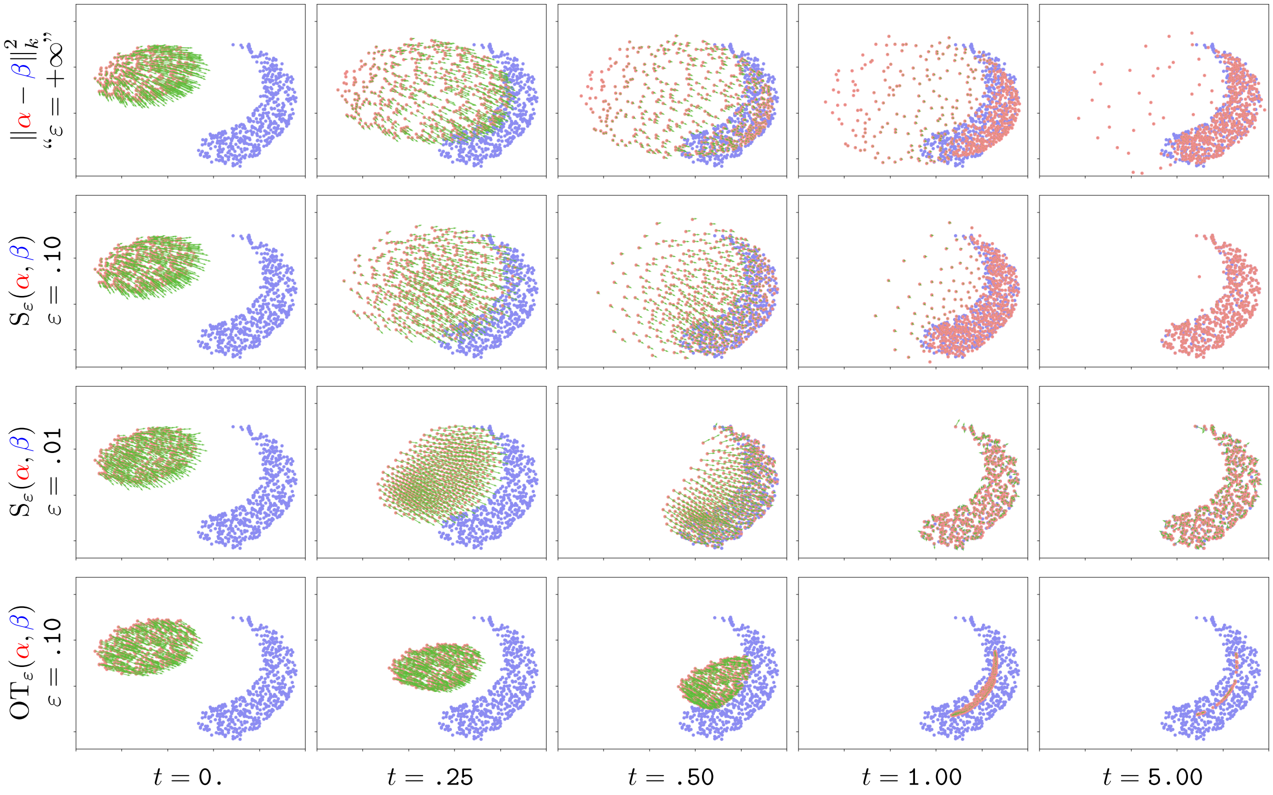


Figure 5 – **Gradient flows for 2-D measures.** The setting is the same as in Figure 4, but the measures $\alpha(t)$ (in red) and β (in blue) are now directly displayed as point clouds of $\mathrm{N} = \mathrm{M} = 500$ points. The evolution of the support of $\alpha(t)$ is thus made apparent, and we display as a green vector field the descent direction $-\nabla_{\boldsymbol{x}_i} \mathrm{L}(\alpha, \beta)$.

In Figures 4 and 5, $\beta = \frac{1}{\mathrm{M}} \sum_{j=1}^{\mathrm{M}} \delta_{\boldsymbol{y}_j}$ is a fixed target measure while $\alpha = \frac{1}{\mathrm{N}} \sum_{i=1}^{N} \delta_{\boldsymbol{x}_i(t)}$ is parameterized by a time-varying point cloud $\boldsymbol{x}(t) = (\boldsymbol{x}_i(t))_{i=1}^{\mathrm{N}} \in (\mathbb{R}^{\mathrm{D}})^{\mathrm{N}}$ in dimension $\mathrm{D} = 1$ or 2. Starting from a set initial condition at time $t = 0$, we simply integrate the ODE

$$\dot{\boldsymbol{x}}(t) = -\mathrm{N}\, \nabla_{\boldsymbol{x}} \big[\, \mathrm{L}\big(\tfrac{1}{\mathrm{N}} \textstyle\sum_{i=1}^{N} \delta_{\boldsymbol{x}}, \beta\big) \,\big] (\boldsymbol{x}_i(t))$$

with a Euler scheme and display the evolution of $\alpha(t)$ up to time $t = 5$.

Interpretation. In both figures, the fourth line highlights the entropic bias that is present in the OT_ε loss: $\alpha(t)$ is driven towards a minimizer that is a "shrunk" version of β. As showed in Theorem 1, the de-biased loss S_ε does not suffer from this issue: just like MMD norms, it can be used as a *reliable*, positive-definite divergence.

Going further, the dynamics induced by the Sinkhorn divergence interpolates between that of an MMD ($\varepsilon = +\infty$) and Optimal Transport ($\varepsilon = 0$), as shown in (4). Here, $\mathrm{C}(x, y) = \|x - y\|$ and we can indeed remark that the second and third lines bridge the gap between the flow of the energy distance $\mathrm{L}_{-\|\cdot\|}$ (in the first line) and that of the Earth Mover's cost OT_0 which moves particles according to an optimal transport plan.

Please note that in both experiments, the gradient of the energy distance with respect to the $\boldsymbol{x}_i$'s vanishes at the extreme points of α's support. Crucially, for small enough values of ε, S_ε recovers the translation-aware geometry of OT and we observe a *clean* convergence of $\alpha(t)$ to β as no sample lags behind.

5 Conclusion

Recently introduced in the ML literature, the Sinkhorn divergences were designed to interpolate between MMD and OT. We have now shown that they also come with many desirable properties: positivity, convexity, metrization of the convergence in law and scalability to large datasets.

To the best of our knowledge, it is the first time that a loss derived from the theory of entropic Optimal Transport is shown to stand on such a firm ground. As the foundations of this theory are progressively being settled, we now hope that researchers will be free to focus on one of the major open problems in the field: the interaction of *geometric* loss functions with concrete machine learning models.

Bibliography

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein GAN. *arXiv preprint arXiv:1701.07875.*

Bassetti, F., Bodini, A., and Regazzini, E. (2006). On minimum Kantorovich distance estimators. *Statistics & probability letters*, 76(12):1298–1302.

Bonneel, N., Peyré, G., and Cuturi, M. (2016). Wasserstein barycentric coordinates: Histogram regression using optimal transport. *ACM Transactions on Graphics*, 35(4).

Bregman, L. M. (1967). The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217.

Charlier, B., Feydy, J., and Glaunès, J. (2018). Kernel operations on the gpu, with autodiff, without memory overflows. `http://www.kernel-operations.io`. Accessed: 2018-10-04.

Chui, H. and Rangarajan, A. (2000). A new algorithm for non-rigid point matching. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 44–51. IEEE.

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

Dziugaite, G. K., Roy, D. M., and Ghahramani, Z. (2015). Training generative neural networks via maximum mean discrepancy optimization. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pages 258–267.

Franklin, J. and Lorenz, J. (1989). On the scaling of multidimensional matrices. *Linear Algebra and its applications*, 114:717–735.

Frogner, C., Zhang, C., Mobahi, H., Araya, M., and Poggio, T. A. (2015). Learning with a Wasserstein loss. In *Advances in Neural Information Processing Systems*, pages 2053–2061.

Galichon, A. and Salanié, B. (2010). Matching with trade-offs: Revealed preferences over competing characteristics. *Preprint hal-00473173.*

Genevay, A., Peyré, G., and Cuturi, M. (2018). Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pages 1608–1617.

Glaunes, J., Trouvé, A., and Younes, L. (2004). Diffeomorphic matching of distributions: A new approach for unlabelled point-sets and sub-manifolds matching. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–II. Ieee.

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

Gretton, A., Borgwardt, K. M., Rasch, M., Schölkopf, B., and Smola, A. J. (2007). A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pages 513–520.

Kaltenmark, I., Charlier, B., and Charon, N. (2017). A general framework for curve and surface comparison and registration with oriented varifolds. In *Computer Vision and Pattern Recognition (CVPR).*

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

Kosowsky, J. and Yuille, A. L. (1994). The invisible hand algorithm: Solving the assignment problem with statistical physics. *Neural networks*, 7(3):477–490.

Léonard, C. (2013). A survey of the Schrödinger problem and some of its connections with optimal transport. *arXiv preprint arXiv:1308.0215.*

Li, Y., Swersky, K., and Zemel, R. (2015). Generative moment matching networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1718–1727.

Micchelli, C. A., Xu, Y., and Zhang, H. (2006). Universal kernels. *Journal of Machine Learning Research*, 7(Dec):2651–2667.

Montavon, G., Müller, K.-R., and Cuturi, M. (2016). Wasserstein training of restricted boltzmann machines. In *Advances in Neural Information Processing Systems*, pages 3718–3726.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.

Peyré, G. and Cuturi, M. (2017). Computational optimal transport. *arXiv:1610.06519.*

Ramdas, A., Trillos, N. G., and Cuturi, M. (2017). On wasserstein two-sample testing and related families of nonparametric tests. *Entropy*, 19(2).

Rubner, Y., Tomasi, C., and Guibas, L. J. (2000). The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121.

Salimans, T., Zhang, H., Radford, A., and Metaxas, D. (2018). Improving GANs using optimal transport. *arXiv preprint arXiv:1803.05573.*

Sanjabi, M., Ba, J., Razaviyayn, M., and Lee, J. D. (2018). On the convergence and robustness of training GANs with regularized optimal transport. *arXiv preprint arXiv:1802.08249*.

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

Schmaltz, C., Gwosdek, P., Bruhn, A., and Weickert, J. (2010). Electrostatic halftoning. In *Computer Graphics Forum*, volume 29, pages 2313–2327. Wiley Online Library.

Schmitzer, B. (2016). Stabilized sparse scaling algorithms for entropy regularized transport problems. *arXiv preprint arXiv:1610.06519*.

Sriperumbudur, B. K., Gretton, A., Fukumizu, K., Schölkopf, B., and Lanckriet, G. R. (2010). Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research*, 11(Apr):1517–1561.

Vaillant, M. and Glaunès, J. (2005). Surface matching via currents. In *Biennial International Conference on Information Processing in Medical Imaging*, pages 381–392. Springer.