Neural Optimal Transport with Lagrangian Costs

Aram-Alexandre Pooladian [1,2] **Carles Domingo-Enrich** [1,2] **Ricky T. Q. Chen** [2] **Brandon Amos** [2]

Abstract

Computational efforts in optimal transport traditionally revolve around the squared-Euclidean cost. In this work, we choose to investigate the optimal transport problem between probability measures when the underlying metric space is non-Euclidean, or when the cost function is understood to satisfy a *least action principle*, also known as a *Lagrangian* cost. These two generalizations are useful when connecting observations from a physical system, where the transport dynamics are influenced by the geometry of the system, such as obstacles, and allows practitioners to incorporate *a priori* knowledge of the underlying system. Examples include barriers for transport, or enforcing a certain geometry, *e.g.*, paths must be circular. Our contributions are of computational interest, where we demonstrate the ability to efficiently compute geodesics and amortize spline-based paths. We demonstrate the effectiveness of this formulation on existing synthetic examples in the literature, where we solve the optimal transport problems in the absence of regularization.

1. Introduction

Optimal transport under the squared-Euclidean cost $c(x, y) = \frac{1}{2}\|x - y\|^2$, from both a mathematical and computational regard, can only be described as elegant. Its connection to convex functions via Brenier's theorem (Brenier, 1991) has allowed for both numerical analysts (Jacobs and Léger, 2020) and machine learning researchers (Bunne et al., 2022b; Amos, 2023; Korotin et al., 2019) to push the boundaries of computational optimal transport in recent years. This connection has also been influential in domains such as economics and statistics (Carlier et al., 2016; Chernozhukov et al., 2017), high-energy particle physics (Manole et al., 2022), computational biology (Schiebinger et al., 2019; Bunne et al., 2021; 2022a), computer vision (Feydy et al., 2017), among others.

[1]New York University [2]Meta AI. Correspondence to: Brandon Amos <bda@meta.com>, Aram-Alexandre Pooladian <aram-alexandre.pooladian@nyu.edu>.

Workshop on New Frontiers in Learning, Control, and Dynamical Systems at the International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA, 2023.

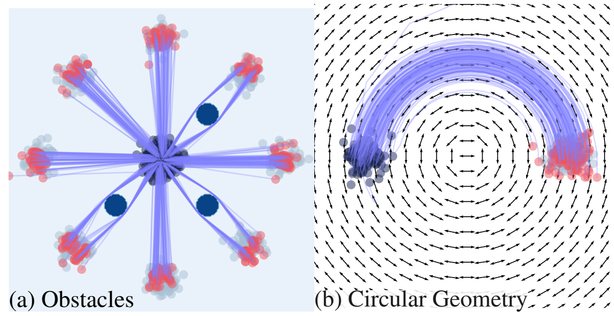

Figure 1. Optimal transport paths with Lagrangian costs on the obstacles setting from Liu et al. (2022) and circular geometry from Scarvelis and Solomon (2023).

We argue that, apart from this elegance, there is little reason practitioners should *default* to this cost in their applications. The purpose of this paper is to provide a computational framework that allows practitioners to enforce transport with respect to a more general notion of cost. To this end, our goal is to numerically solve the optimal transport problem when the underlying cost of displacement is governed by a *least action principle*: For two points $x, y \in \mathbb{R}^d$, the *displacement cost* $c(x, y)$ is

$$c(x, y) := \inf_{\gamma \in \mathcal{C}(x,y)} \left\{ \int_0^1 \mathcal{L}(\gamma_t, \dot{\gamma}_t, t)\, \mathrm{d}t \right\} \tag{1}$$

where $\mathcal{C}(x, y)$ is the set of smooth, time dependent curves that connect x and y, *i.e.* so $\gamma_0 = x$ and $\gamma_1 = y$, and

$$\mathcal{L} : \mathbb{R}^d \times \mathbb{R}^d \times [0, 1] \to \mathbb{R} \cup \{+\infty\} \tag{2}$$

is the *Lagrangian function* which ultimately governs the cost of transport. The Lagrangian takes as arguments the position of the curve $\gamma_t \in \mathbb{R}^d$ at time t and the velocity at that point $\dot{\gamma}_t := \mathrm{d}\gamma_t / \mathrm{d}t$. This definition is inspired by Lagrangian mechanics: equations of motion that are based on energies of a system, rather than forces. As outlined in Villani (2009, Chapter 7), and briefly discussed in Section 2.2, this notion of cost can be lifted to the space of probability measures, instead of just being between two fixed points over a space.

If $\mathcal{L}(x, v, t) \coloneqq \frac{1}{2}\|v\|^2$, then $c(x, y)$ recovers the squared-Euclidean distance of transport (*cf.* Chen et al. (2016, Eq. 2) and Example 1). However, the Lagrangians that we consider are more eccentric and incorporate 1) *potential energy terms* with a function $U : \mathbb{R}^d \to \mathbb{R}$, *i.e.*

$$\mathcal{L}(x, v, t) \coloneqq \tfrac{1}{2}\|v\|^2 - U(x), \tag{3}$$

and 2) *position-dependent costs*:

$$\mathcal{L}(x, v, t) \coloneqq \tfrac{1}{2}\|v\|^2_{A(x)} = \tfrac{1}{2}v^\top A(x)v, \tag{4}$$

e.g. for representing geodesic distances. Examples 2 and 3, respectively, give more details on 1) and 2). In Fig. 1(a), we use smooth potential functions U to act as obstacles between the source Gaussian and the 8-Gaussian mixture. In Fig. 1(b), we learn the least-cost paths between two Gaussians when the cost of displacement is lowest along circular trajectories. We stress that, despite this being a non-standard notion of cost, there still exists an optimal transport map (Eq. (9)) expressed explicitly as the minimizer to an optimization problem which we are able to learn.

We consider two categories of optimal transport problems: i) transport between a *pair* of probability measures (μ, ν), or ii) transport between consecutive pairs of measures $\{(\rho_i, \rho_{i+1}\}_{i=0}^{K-1}$. In i), we assume the practitioner is interested in modeling physical systems, and has access to their Lagrangian of interest, either through the potential function U or position-dependent metric A, and wants to know the optimal displacement and cost between μ and ν. For ii), the practitioner has access to samples from K probability measures, which they believe to be traversing optimally under some underlying Riemannian metric which is not known. We *learn* this Riemannian metric to uncover the geometry of the space. We stress that both modifications allow for the practitioner to employ in-domain knowledge to the cost function, as opposed to the squared-Euclidean cost, which remains information agnostic.

Contributions. We aim to numerically solve optimal transport problems where the cost function follows Eq. (1) and the measures are in an underlying continuous space. Our approach involves parameterizing the solution to the Lagrangian optimal transport problem using neural networks. The non-standard cost leads to two computational challenges for obtaining 1) the displacement cost Eq. (1) and minimizing path, 2) the c-transform of the Lagrangian cost. We overcome both of these by using amortized optimization, *e.g.* as in Amos (2022), to obtain approximate solutions. We then parameterize the metric of a Riemannian manifold using a neural network and learn it by minimizing the optimal transport cost and show that this improves the metric recovery experiments in Scarvelis and Solomon (2023).

2. Background on optimal transport

2.1. Kantorovich primal-dual problems and mappings

Optimal transport can be written as several equivalent infinite-dimensional optimization problems, which we outline below under mild conditions. We refer the interested reader to Santambrogio (2015) or Villani (2009) for a more detailed discussion. Let $\mu \in \mathcal{P}(\mathcal{X})$ and $\nu \in \mathcal{P}(\mathcal{Y})$ be two probability measures defined on $\mathcal{X}$ and $\mathcal{Y}$, respectively, which are complete, separable metric spaces (for simplicity, one can consider $\mathbb{R}^d$ endowed with the Euclidean metric). Let $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a lower semicontinuous, real-valued cost function (for simplicity, one can consider any bounded convex cost function). The *primal (Kantorovich) formulation*, attributed to Kantorovich (1942), is given by

$$\mathrm{OT}_c(\mu, \nu) \coloneqq \inf_{\pi \in \Gamma(\mu,\nu)} \iint_{\mathcal{X}\times\mathcal{Y}} c(x, y)\, \mathrm{d}\pi(x, y)\,, \tag{5}$$

where $\Gamma(\mu, \nu) \subset \mathcal{P}(\mathcal{X} \times \mathcal{Y})$ is the set of transportation couplings between μ and ν i.e. $\pi \in \Gamma(\mu, \nu)$ if

$$\int_{\mathcal{Y}} \mathrm{d}\pi(x, y) = \mathrm{d}\mu(x)\,, \quad \int_{\mathcal{X}} \mathrm{d}\pi(x, y) = \mathrm{d}\nu(x)\,. \tag{6}$$

Under our specifications on the cost function, an equivalent optimization problem called the *dual (Kantorovich) formulation* (*e.g.* Villani (2009, Theorem 5.10)), is given by

$$\mathrm{OT}_c(\mu, \nu) \coloneqq \sup_{g\in L^1(\nu)} \int g^c(x)\, \mathrm{d}\mu(x) + \int g(y)\, \mathrm{d}\nu(y)\,, \tag{7}$$

where $L^1(\nu)$ is the set of integrable functions with respect to ν, and g^c is the *c-transform of* g, written

$$g^c(x) \coloneqq \inf_{y\in\mathcal{Y}} J(y; x) \text{ where } J(y; x) \coloneqq c(x, y) - g(y)\,. \tag{8}$$

When the infimum in Eq. (8) is attained, the minimizer is

$$\hat{y}(x; c, g) \coloneqq \operatorname*{argmin}_{y\in\mathcal{Y}} \{c(x, y) - g(y)\}\,. \tag{9}$$

When the supremum in Eq. (7) is attained, we write $\hat{g}$ as the maximizer, called the optimal Kantorovich potential. Consequently, we define the *optimal transport map* as the minimizer $\hat{y}(\cdot; c, \hat{g})$, which is Eq. (9) applied to the optimal Kantorovich potential: given $x \in \mathcal{X}$, $\hat{y}(x; c, \hat{g})$ corresponds to the optimal displacement from μ to ν.

2.2. Optimal transport with Lagrangian costs

We now suppose our probability measures exist on compact subsets $\mathcal{X} = \mathcal{Y} \subseteq \mathbb{R}^d$. We associate the cost of displacing x to y with an *action* that is to be minimized over a time horizon $[0, 1]$. Borrowing terminology from physics, these actions will take the form of *Lagrangian* functionals, which are functions that depend on the position of a curve γ_t, its velocity, $\dot{\gamma}_t$, and time $t \in [0,1]$;

$$(\gamma_t, \dot{\gamma}_t, t) \longmapsto \mathcal{L}(\gamma_t, \dot{\gamma}_t, t)\,, \tag{10}$$

where curves in $\mathcal{C}$ are understood to be smooth and absolutely continuous curves over $\mathbb{R}^d$, indexed by time in $[0,1]$, *cf.* Villani (2009, Chapter 7). The Lagrangian induces an *energy* E on curves defined by

$$E(\gamma; x, y) \coloneqq \left\{ \int_0^1 \mathcal{L}(\gamma_t, \dot{\gamma}_t, t)\, \mathrm{d}t \right\} \tag{11}$$

The *cost of displacement*, or *action*, is then given by

$$c(x, y) \coloneqq \inf_{\gamma \in \mathcal{C}(x,y)} E(\gamma; x, y)\,. \tag{12}$$

Though initially defined between two points on the manifold, this cost can be appropriated "lifted" to the space of probability measures, resulting in what is known as *Lagrangian Optimal Transport*. Indeed, under mild assumptions on $\mathcal{L}$, the generalized notion of transport vis-à-vis minimizers to Eq. (8) is defined. A thorough discussion is found in Villani (2009, Chapter 7), specifically Theorem 7.21 and Remark 7.25. The following conditions are sufficient for Eq. (12) to define a valid notion of transport: $\mathcal{L}$ is twice continuously differentiable and strictly convex in v, with $\nabla_v^2 \mathcal{L} \succ 0$ everywhere, and $\mathcal{L}$ does not depend (explicitly) on t. These conditions are satisfied in all our problem considerations.

Remark 1. For simplicity, we present the background for manifolds $(\mathbb{R}^d, g)$ where g is potentially a non-Euclidean metric. These same discussions hold when we instead consider a general smooth manifold $\mathcal{M}$ and its associated metric g; *cf. e.g.* Feldman and McCann (2002).

Example 1 (Euclidean kinetic energy)**.** *The squared Euclidean distance can be recovered (*cf.* Chen et al. (2016, Eq. 2)) by taking the Lagrangian as the kinetic energy:*

$$\mathcal{L}(\gamma_t, \dot{\gamma}_t, t) \coloneqq \tfrac{1}{2}\|\dot{\gamma}_t\|^2\,, \tag{13}$$

Example 2 (Euclidean kinetic and potential energy)**.** *Adding a potential function*[1] $U : \mathbb{R}^d \to \mathbb{R}$*,* i.e.

$$\mathcal{L}(\gamma_t, \dot{\gamma}_t, t) \coloneqq \frac{1}{2}\|\dot{\gamma}_t\|^2 - U(\gamma_t)\,, \tag{14}$$

The potential provides a way of specifying how "easy" it is to pass through regions of the space: this includes the obstacles an in Fig. 1(a) and Fig. 2 where the potential takes low values and prevents the paths from crossing them:

Example 3 (Squared geodesic distances on Riemannian manifolds)**.** *A final modification lies in the choice of the underlying metric g, which may be* position dependent. *As we are operating on $\mathbb{R}^d$ for simplicity, the metric g at a point $x \in \mathbb{R}^d$ is given by the scaled inner product $\langle u, v\rangle_x \coloneqq \langle u, A(x)v\rangle$ for any $u, v \in \mathbb{R}^d$, for some $A(\cdot) : \mathbb{R}^d \to \mathbb{S}^d_{++}$ positive-definite. This allows us to consider Lagrangians that induce squared-geodesic distances:*

$$\mathcal{L}(\gamma_t, \dot{\gamma}_t, t; A) = \frac{1}{2}\|\dot{\gamma}_t\|^2_{A(\gamma_t)}\,. \tag{15}$$

The circular geometry in Fig. 2(a) shows Example 3 where the metric is given by the positive-definite matrix

$$A(x) \coloneqq \begin{pmatrix} \frac{x_1^2}{\|x\|^2} & 1 - \frac{x_1 x_2}{\|x\|^2} \\ 1 - \frac{x_1 x_2}{\|x\|^2} & \frac{x_2^2}{\|x\|^2} \end{pmatrix}\,. \tag{16}$$

3. Neural OT with Lagrangian costs

We first focus on computationally solving for the Kantorovich dual in Eq. (7) between two measures $\mu \in \mathcal{P}(\mathcal{X})$ and $\nu \in \mathcal{P}(\mathcal{Y})$ when the cost function is of the form Eq. (14) or Eq. (15). All components of the Lagrangian are known, *i.e.* the Lagrangian potential U or the underlying metric A is known, and we assume access to samples from μ and ν. The Kantorovich potential $g \in L^1(\nu)$ in Eq. (7) is a function $g : \mathcal{Y} \to \mathbb{R}$. Alg. 1 summarizes our solution.

We follow recent work on neural optimal transport, *e.g.* as in Taghvaei and Jalali (2019); Makkuva et al. (2020); Korotin et al. (2019); Fan et al. (2021a); Amos (2023), and represent the Kantorovich potential as a neural network g_θ with parameters θ. With this parameterization, we recast Eq. (7) as $\max_\theta \ell_{\text{dual}}(\theta)$ where

$$\ell_{\text{dual}}(\theta) \coloneqq \int g_\theta^c(x)\, \mathrm{d}\mu(x) + \int g_\theta(y)\, \mathrm{d}\nu(y) \tag{17}$$

and the c-transform g_θ^c incorporates the Lagrangian function. We solve Eq. (17) with stochastic gradient method using

$$\nabla_\theta \ell_{\text{dual}}(\theta) = \int \nabla_\theta g_\theta^c(x)\, \mathrm{d}\mu(x) + \int \nabla_\theta g_\theta(y)\, \mathrm{d}\nu(y), \tag{18}$$

where g_θ^c is differentiated with *Danskin's envelope theorem* (Danskin, 1966; Bertsekas, 1971), *i.e.*

$$\nabla_\theta g_\theta^c(x) = \nabla_\theta J(\hat{y}(x); x, c, g_\theta) = -\nabla_\theta g_\theta(\hat{y}(x))\,. \tag{19}$$

We follow Taghvaei and Jalali (2019), as well as other OT work based on neural networks, and approximate Eq. (17) and Eq. (18) with Monte-Carlo estimates of the integrals as they are not computable in closed-form. Computing these estimates still requires overcoming the following challenges:

Challenge 1 (Computing the c-transform)**.** *Estimating Eq. (17) and Eq. (18) require obtaining the c-transform g_θ^c and the corresponding minimizing point $\hat{y}(x; c, g)$. This requires solving the optimization problem in Eq. (8) for* ***every*** *x, which does not have a closed-form solution.*

[1] Not to be confused with *Kantorovich* potentials (Section 2.1)

Algorithm 1 Neural Lagrangian Optimal Transport

inputs: measures μ and ν, Kantorovich potential g_θ, c-transform predictor y_ϕ, and spline predictor φ_η
while unconverged **do**
  sample batches $\{x_i\}_{i=1}^N \sim \mu$ and $\{y_i\}_{i=1}^N \sim \nu$
  obtain the amortized c-transform predictor $y_\phi(x_i)$ for $i \in [N]$
  fine-tune the c-transform by numerically solving Eq. (9), warm-starting with $y_\phi(x_i)$
  update the potential with gradient estimate of $\nabla_\theta \ell_{\text{dual}}$ (Eq. (18))
  update the c-transform predictor y_ϕ using a gradient estimate of Eq. (20)
  update the spline predictor φ_η using a gradient estimate of Eq. (23)
end while
return optimal parameters θ, ϕ, η

Prior OT work for the squared-Euclidean cost settings had to overcome a similar challenge when the c-transform becomes the Fenchel or convex conjugate operation: Taghvaei and Jalali (2019); Korotin et al. (2021) use numerical solvers such as L-BFGS, Adam, and other gradient-based methods, Makkuva et al. (2020); Korotin et al. (2019; 2021) use an amortized approximation, and Amos (2023) combines the amortized approximation with a numerical solver. For c-transforms, Fan et al. (2021a) uses an amortized approximation to overcome Challenge 1.

We follow these works and overcome Challenge 1 by amortizing the solution to Eq. (8). This involves parameterizing an approximate c-transform map $\hat{y}_\phi \approx \hat{y}$ that we learn with a regression-based loss

$$\min_\phi \int \|\hat{y}(x) - y_\phi(x)\| \, \mathrm{d}\mu(x) \,. \tag{20}$$

The conjugation model $\hat{y}_\phi$ is only an approximation and may be inaccurate as the potential g_θ changes during training. An inaccurate approximation to the c-transform results in a poor approximation to the objective in Eq. (17); to improve it, we follow Amos (2023) and fine-tune the c-transform prediction with a few steps of L-BFGS to solve Eq. (9) and warm-starting it with the amortized prediction.

Challenge 2 (Computing the cost c)**.** *Evaluating the Lagrangian cost c that arises in the c-transform in Eqs. (8) and (20) involves solving the optimization problem in Eq. (12) over paths. While closed-form solutions exist for simple manifolds,* e.g. *straight paths on Euclidean space or great arcs on spherical manifolds, the more general settings we consider do not admit closed-form solutions and need to be numerically solved.*

Computationally representing paths and solving for Riemannian geodesics and Lagrangian paths in Eq. (12) outside of the context of optimal transport is an active research area. We follow Beik-Mohammadi et al. (2021); Detlefsen et al. (2021) and parameterize the space of paths between x and y with a cubic spline $\gamma_\varphi(x, y)$ (where the parameters are φ). This spline parameterization transforms the optimization problem in Eq. (12) to an optimization problem over the continuous-valued parameters of the spline as

$$\varphi^\star(x, y) \coloneqq \operatorname*{argmin}_{\varphi \in \Phi(x,y)} E(\varphi; x, y) \tag{21}$$

where

$$E(\varphi; x, y) \coloneqq \left\{ \int_0^1 \mathcal{L}((\gamma_\varphi)_t, (\dot{\gamma}_\varphi)_t, t) \, \mathrm{d}t \right\} \tag{22}$$

and $\Phi(x, y)$ is the space of cubic splines between x and y; see Appendix A for more details.

Solving Eq. (21) for every c-transform within every evaluation for the OT cost is computationally intractable, so we propose to amortize the path computation using objective-based amortization as in Amos (2022). Given points x and y, we parameterize the spline amortization model with $\varphi_\eta(x, y) \approx \varphi^\star(x, y)$ and train it to compute the paths necessary for the OT cost, *i.e.*,

$$\min_\eta \int E(\varphi_\eta; x, \hat{y}(x)) \, \mathrm{d}\mu(x) \,. \tag{23}$$

4. Metric learning with Lagrangian OT

We now follow Scarvelis and Solomon (2023) and consider optimal transport problems where the ground-truth displacement is given by geodesics induced by non-Euclidean geometries, like Eq. (16). However, we crucially *do not* assume knowledge of the underlying positive-definite matrix-valued function A that induces the Riemannian geometry. Our goal is to instead *learn* A on the basis of sequential pairs of probability measures $\{(\rho_i, \rho_{i+1})\}_{i=1}^{K-1}$, which is motivated by recent applications in single-cell genomic profiling; see *e.g.* Schiebinger et al. (2019) and Bunne et al. (2022b). Let A_ϑ be the neural network parameterization of a positive-definite matrix, with the network weights given by ϑ. The matrix-valued function A_ϑ then induces the cost c_ϑ, given by

$$c_\vartheta(x, y) \coloneqq \inf_{\gamma \in \mathcal{C}(x,y)} \left\{ \int_0^1 \tfrac{1}{2} \|\dot{\gamma}_t\|^2_{A_\vartheta(\gamma_t)} \, \mathrm{d}t \right\} . \tag{24}$$

Algorithm 2 Metric learning with Lagrangian OT

inputs: measures $\{(\rho_i, \rho_{i+1})\}_{i=1}^{K-1}$, metric A_ϑ, potentials g_{θ_i}, c-transform predictors y_{ϕ_i}, spline predictors φ_{η_i},
while unconverged **do**
  update ϑ using $\nabla_\vartheta \ell_{\text{dual}}$ (with the terms in Eq. (27))
  update the OT approximation θ_i, ϕ_i, η_i with an iteration of Alg. 1
end while
return optimal parameters ϑ, θ_i, ϕ_i, η_i

We can then learn a metric that results in a geometry with a minimal OT cost, *i.e.* $\min_\vartheta \ell_{\text{metric}}(\vartheta)$ where

$$\ell_{\text{metric}}(\vartheta) \coloneqq \frac{1}{K}\sum_{i=1}^{K-1} \text{OT}_{c_\vartheta}(\rho_i, \rho_{i+1}). \tag{25}$$

To ensure A_ϑ does not collapse, we embed rotational invariances into R_ϑ; further technical details for R_ϑ are addressed in Section 4.1. We use the neural networks from Section 3 to approximate the OT maps, resulting in

$$\ell_{\text{metric}}(\vartheta) \approx \max_{\{\theta_i\}_{i=1}^{K-1}} \frac{1}{K}\sum_{i=1}^{K-1} \ell_{\text{dual}}(\theta_i; \rho_i, \rho_{i+1}, \vartheta), \tag{26}$$

Equation (26) is a delicate min-max optimization scheme that we solve with alternating descent-ascent. For a fixed metric A_ϑ, the inner maximization problem is the same as Section 3, but with $K-1$ networks. The only difference is the outer minimization step, which we can compute efficiently via sequential applications of the envelope theorem. Noting that only the first term in Eq. (25) depends on A_ϑ, the gradient of Eq. (26) can be computed with the terms

$$\begin{aligned}\nabla_\vartheta \ell_{\text{dual}}(\theta; \rho_i, \rho_{i+1}, \vartheta) &= \nabla_\vartheta \int g^{c_\vartheta}\, \mathrm{d}\rho_i \\ &= \int \nabla_\vartheta g^{c_\vartheta}\, \mathrm{d}\rho_i \\ &= \int \nabla_\vartheta c_\vartheta(x, \hat{y}(x))\, \mathrm{d}\rho(x) \\ &= \int \nabla_\vartheta E_\vartheta(\varphi_{\eta_i}, x, \hat{y}(x)))\, \mathrm{d}\rho_i\,.\end{aligned} \tag{27}$$

The full update of A_ϑ then takes the average gradient of these $K-1$ gradient computations. Thus, the inner maximization step requires $K-1$ applications of Alg. 1, and the outer minimization step freezes the inner parameters, leaving only an average update for A_ϑ. We stress that a primary difference in this setting limited finite-sample access to the $K-1$ measures from which we are to learn the ground-truth metric, and output paths and optimal transport maps. Alg. 2 overviews the general algorithm.

4.1. Parameterizing the metric A_ϑ with rotations

We parameterize the metric A_ϑ to predict a rotation $R_\vartheta(x)$ of a fixed matrix B, *i.e.*

$$A_\vartheta(x) \coloneqq R_\vartheta(x) B R_\vartheta(x)^\top, \tag{28}$$

where

$$B \coloneqq \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix} \tag{29}$$

and

$$R_\vartheta(x) \coloneqq \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \tag{30}$$

The eigenvalues of $A_\vartheta(x)$ are always the eigenvalues of B, *i.e.*, 1 and 0.1, and parameterizing the rotation forces B to be rotated so the data movement is along the smallest eigenvector as in Fig. 3.

5. Experiments

We performed several experiments in settings from prior published work. Our synthetic examples for Lagrangians with potentials are from Liu et al. (2022); Koshizuka and Sato (2022). The metric learning problems are from Scarvelis and Solomon (2023).

5.1. Lagrangians with barrier potentials

We first consider Lagrangians of the form

$$L(x, v, t) \coloneqq \tfrac{1}{2}\|v\|^2 - U(x)\,, \tag{31}$$

with $U : \mathbb{R}^d \to \mathbb{R}$. In many of our examples, U will take the form of a barrier that limits transport, which in principle should yield $U(x) = -\infty$ if x is in the domain of U. To ensure L remains smooth, we consider smooth (but sharp) barrier functions U, as in prior work. Precise definitions of the functions are deferred to Appendix B. In Figure 2, we learn the optimal transport map between two measures with a box or slit constraint; in Figure 1(a), we learn the optimal transport map between a Gaussian and a Gaussian mixture, with barriers (the box and slit), a hill, and a well.

5.2. Riemannian metric learning

We consider three ground-truth Riemannian metrics $A(\cdot)$, which are given by the arrows in Fig. 4. To be precise, grey arrows in Fig. 3 show the direction of the smallest eigenvector at that point *i.e.* the easiest direction to move in. Note that Figure 3(a) is the circle metric from Eq. (16), and the other two metrics are non-smooth metrics that cause splitting Fig. 3(b) or reflections Fig. 3(c). In each task, we

Table 1. Alignment scores ℓ_{align} for metric recovery in Fig. 4. (higher is better)

	Circle	Mass Splitting	X Paths
Scarvelis and Solomon (2023)	0.995	0.839	0.916
Our approach	$\mathbf{0.997 \pm 0.002}$	$\mathbf{0.986 \pm 0.001}$	$\mathbf{0.957 \pm 0.001}$

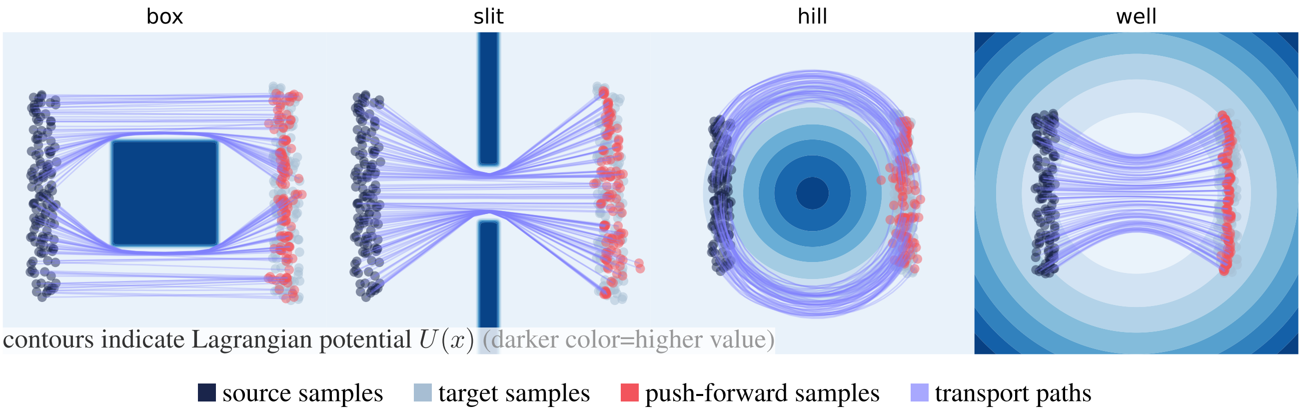

Figure 2. OT maps and paths for settings considered in Koshizuka and Sato (2022).

are given samples from $K - 1$ pairs of probability measures which were generated according to these Riemannian metrics. Our task is to learn the metric on the basis of the samples alone, and ideally recover the transport path exactly. The precise formulas for $A(\cdot)$ and descriptions of the learning tasks are in Appendix C.

We quantify our ability to recover the ground-truth metric through the alignment score from Scarvelis and Solomon (2023):

$$\ell_{\text{align}}(A, \hat{A}) := \frac{1}{d|\mathcal{D}|} \sum_{x \in \mathcal{D}} \sum_{i=1}^{d} |u_i(x)^\top \hat{u}_i(x)| \in [0, 1], \quad (32)$$

where $\mathcal{D}$ is a finite discretization of the space, and $u_i(x)$ (resp. $\hat{u}_i(x)$) is the (unit) eigenvector with eigenvalue λ_i (resp. $\hat{\lambda}_i$) for the matrix $A(x)$ (resp. $\hat{A}(x)$). Our results are reported in Table 1, where we perform the same experiment over three randomized trials, and report the same metric values from Scarvelis and Solomon (2023). Notably, we see a roughly 17% improvement in the "Mass Splitting" example, with near-perfect recovery in all cases.

Finally, in Fig. 4, we plot our learned geodesics that are learned from the data. Unlike (Scarvelis and Solomon, 2023), our formulation allows us to output these geodesics, and does not require a separate training scheme. Indeed, in their approach, the authors first learn the metric, then use a separate procedure to learn a continuous normalizing flow to act as the geodesics. In contrast, since we learn geodesic paths, we are able to effortlessly output the paths in Fig. 4.

6. Related work

6.1. Lagrangian Schrödinger bridges

Koshizuka and Sato (2022) and Liu et al. (2022) are the closest for this subproblem that we consider in Section 3. The former studies the *Stochastic Optimal Transport* (SOT) problem, which amounts to optimal transport on path space, with the path dictated by a stochastic differential equation (SDE). The authors consider Lagrangian costs, and use neural SDEs to model the trajectories. In Liu et al. (2022), they investigate the *Schrödinger Bridge Problem* (SBP), which can be distilled to optimal transport with entropic regularization (Léonard, 2012), also using neural SDEs, and have a particular focus on Mean-Field Games, which is not a focus of this work. Note that the (SBP) is a special case of (SOT); see Koshizuka and Sato (2022, Figure 2).

6.2. Metric learning with OT

Although our setup is taken from Scarvelis and Solomon (2023), there are several differences between our work and theirs. They deploy a specialized duality theory based on Section 2.1, where the Kantorovich potentials must be 1-Lipschitz with respect to the weighted Euclidean metric — this is enforced using an additional regularizer in the inner maximization problem in Eq. (25). Finally to ensure the metric does not collapse, they add another regularizer on Eq. (25), for the outer minimization problem. Moreover, they use Eq. (25) *only* to fit a metric $\hat{A}_\vartheta$, and later use another optimization problem (based on continuous normalizing flows) to fit their geodesics. In contrast, our approach is self-contained, unregularized, and generalizable to other

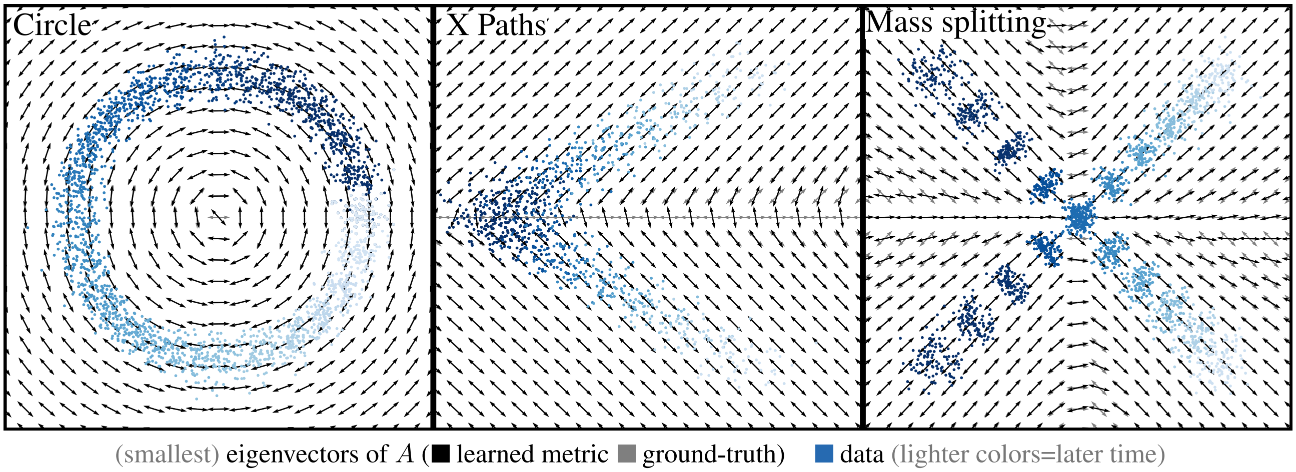

Figure 3. We successfully recover the metrics on the settings from Scarvelis and Solomon (2023).

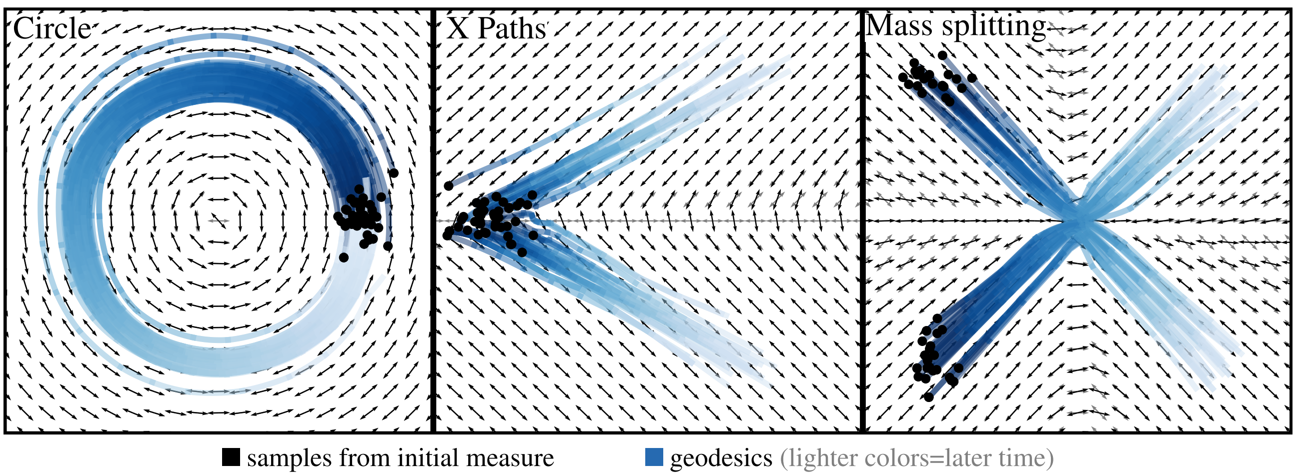

Figure 4. Our transport geodesics are able to reconstruct continuous versions of the original data that can predict the movement of individual particles given only samples from the first measure.

notions of cost. Furthermore, we are able to obtain approximations of the transport map with y_ϕ and transport paths with φ_η.

6.3. Estimation and applications of optimal transport maps under the squared-Euclidean cost

Apart from machine learning communities, the squared-Euclidean cost has also garnered much interest in traditional domains. Over the last few years starting with Hütter and Rigollet (2021), a statistical analysis of the estimation of optimal transport maps for the squared-Euclidean cost, followed swiftly by Deb et al. (2021); Manole et al. (2021); Pooladian and Niles-Weed (2021), to name a few. Applications of optimal transport in machine learning focus primarily around grounded in "generative modeling", where the notion of a learned transport map allows us to generate new samples from a target measure from which we only have access to samples (*e.g.* generating a new image). Examples of such works include Huang et al. (2021); Finlay et al. (2020a;b); Lipman et al. (2022); Pooladian et al. (2023); Onken et al. (2021); Rout et al. (2021); Bousquet et al. (2017); Balaji et al. (2020); Seguy et al. (2018); Tong et al. (2023). The work of Schiebinger et al. (2019) had a cascading effect in the machine learning community, popularizing the ability to predict single-cell genome expressions through optimal transport using limited data. Examples of such works include Bunne et al. (2021; 2022b;a); Lübeck et al. (2022) and Tong et al. (2020).

6.4. Estimation of optimal transport maps for other notions of cost

One can generalize the notion of a Brenier-map in closed form by considering a specific family of cost functions. We call a convex cost function *translation invariant* if $c(x, y) :=$

$h(x-y)$ with $h:\mathbb{R}^d\to\mathbb{R}$ convex. To the best of our knowledge, the pursuit of estimating such maps has been seldom, apart from Fan et al. (2021b) and the very recent works Cuturi et al. (2023); Uscidda and Cuturi (2023); Klein et al. (2023). Other applications include i.e. defining notions of optimal transport between datasets or different spaces Nekrashevich et al. (2023); Alvarez-Melis and Fusi (2020); Alvarez-Melis and Jaakkola (2018)

7. Conclusion

In this work, we proposed an efficient framework for computing geodesics under generalized least-action principles, or Lagrangians, leading to large-scale computation of Lagrangian Optimal Transport trajectories. Combining amortization and the use of splines, we demonstrate the capacity of our method on a suite of problems, ranging from learning non-Euclidean geometries from data, to computing optimal transport maps under (known) non-Euclidean geometries and costs. There are many remaining fundamental research avenues that arise as a result of this work. Examples include a statistical analysis of these new costs (*e.g.*, Hundrieser et al. (2023); Hütter and Rigollet (2021)), extensions to the unbalanced optimal transport setting through the Wasserstein-Fisher-Rao metric (Gallouët and Monsaingeon, 2017), and extensions to multi-marginal optimal transport (Pass, 2015).

References

David Alvarez-Melis and Nicolo Fusi. Geometric dataset distances via optimal transport. *Advances in Neural Information Processing Systems*, 33:21428–21439, 2020. Cited on page 8.

David Alvarez-Melis and Tommi S Jaakkola. Gromov-wasserstein alignment of word embedding spaces. *arXiv preprint arXiv:1809.00013*, 2018. Cited on page 8.

Brandon Amos. Tutorial on amortized optimization for learning to optimize over continuous domains. *arXiv preprint arXiv:2202.00665*, 2022. Cited on pages 2 and 4.

Brandon Amos. On amortizing convex conjugates for optimal transport. *International Conference on Learning Representations*, 2023. Cited on pages 1, 3, and 4.

Yogesh Balaji, Rama Chellappa, and Soheil Feizi. Robust optimal transport with applications in generative modeling and domain adaptation. *Advances in Neural Information Processing Systems*, 33:12934–12944, 2020. Cited on page 7.

Richard H Bartels, John C Beatty, and Brian A Barsky. *An introduction to splines for use in computer graphics and geometric modeling*. Morgan Kaufmann, 1995. Cited on page 12.

Hadi Beik-Mohammadi, Søren Hauberg, Georgios Arvanitidis, Gerhard Neumann, and Leonel Rozo. Learning riemannian manifolds for geodesic motion skills. *arXiv preprint arXiv:2106.04315*, 2021. Cited on pages 4 and 12.

Dimitri P Bertsekas. *Control of uncertain systems with a set-membership description of the uncertainty.* PhD thesis, Massachusetts Institute of Technology, 1971. Cited on page 3.

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. Cited on page 7.

Yann Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on pure and applied mathematics*, 44(4):375–417, 1991. Cited on page 1.

Charlotte Bunne, Stefan G Stark, Gabriele Gut, Jacobo Sarabia del Castillo, Kjong-Van Lehmann, Lucas Pelkmans, Andreas Krause, and Gunnar Rätsch. Learning single-cell perturbation responses using neural optimal transport. *bioRxiv*, 2021. Cited on pages 1 and 7.

Charlotte Bunne, Andreas Krause, and Marco Cuturi. Supervised training of conditional Monge maps. *arXiv preprint arXiv:2206.14262*, 2022a. Cited on pages 1 and 7.

Charlotte Bunne, Laetitia Papaxanthos, Andreas Krause, and Marco Cuturi. Proximal optimal transport modeling of population dynamics. In *International Conference on Artificial Intelligence and Statistics*, pages 6511–6528. PMLR, 2022b. Cited on pages 1, 4, and 7.

Richard L Burden, J Douglas Faires, and Annette M Burden. *Numerical analysis*. Cengage learning, 2015. Cited on page 12.

Guillaume Carlier, Victor Chernozhukov, and Alfred Galichon. Vector quantile regression: an optimal transport approach. *The Annals of Statistics*, 44(3):1165–1192, 2016. Cited on page 1.

Ricky T. Q. Chen. torchdiffeq, 2018. URL `https://github.com/rtqichen/torchdiffeq`. Cited on page 13.

Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. On the relation between optimal transport and schrödinger bridges: A stochastic control viewpoint. *Journal of Optimization Theory and Applications*, 169:671–691, 2016. Cited on pages 2 and 3.

Victor Chernozhukov, Alfred Galichon, Marc Hallin, and Marc Henry. Monge–Kantorovich depth, quantiles, ranks and signs. *The Annals of Statistics*, 45(1):223–256, 2017. Cited on page 1.

Marco Cuturi, Michal Klein, and Pierre Ablin. Monge, bregman and occam: Interpretable optimal transport in high-dimensions with feature-sparse maps. *arXiv preprint arXiv:2302.04065*, 2023. Cited on page 8.

John M Danskin. The theory of max-min, with applications. *SIAM Journal on Applied Mathematics*, 14(4):641–664, 1966. Cited on page 3.

Nabarun Deb, Promit Ghosal, and Bodhisattva Sen. Rates of estimation of optimal transport maps using plug-in estimators via barycentric projections. *Advances in Neural Information Processing Systems*, 34:29736–29753, 2021. Cited on page 7.

Nicki S. Detlefsen, Alison Pouplin, Cilie W. Feldager, Cong Geng, Dimitris Kalatzis, Helene Hauschultz, Miguel González-Duque, Frederik Warburg, Marco Miani, and Søren Hauberg. Stochman. *GitHub. Note: https://github.com/MachineLearningLifeScience/stochman/*, 2021. Cited on pages 4 and 12.

Jiaojiao Fan, Shu Liu, Shaojun Ma, Yongxin Chen, and Haomin Zhou. Scalable computation of monge maps with general costs. *arXiv preprint arXiv:2106.03812*, page 4, 2021a. Cited on pages 3 and 4.

Jiaojiao Fan, Shu Liu, Shaojun Ma, Haomin Zhou, and Yongxin Chen. Neural monge map estimation and its applications. *arXiv preprint arXiv:2106.03812*, 2021b. Cited on page 8.

Mikhail Feldman and Robert McCann. Monge's transport problem on a riemannian manifold. *Transactions of the American Mathematical Society*, 354(4):1667–1697, 2002. Cited on page 3.

Jean Feydy, Benjamin Charlier, François-Xavier Vialard, and Gabriel Peyré. Optimal transport for diffeomorphic registration. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 291–299. Springer, 2017. Cited on page 1.

Chris Finlay, Augusto Gerolin, Adam M Oberman, and Aram-Alexandre Pooladian. Learning normalizing flows from Entropy-Kantorovich potentials. *arXiv preprint arXiv:2006.06033*, 2020a. Cited on page 7.

Chris Finlay, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam Oberman. How to train your neural ode: the world of jacobian and kinetic regularization. In *International Conference on Machine Learning*, pages 3154–3164. PMLR, 2020b. Cited on page 7.

Thomas O Gallouët and Leonard Monsaingeon. A jko splitting scheme for kantorovich–fisher–rao gradient flows. *SIAM Journal on Mathematical Analysis*, 49(2):1100–1130, 2017. Cited on page 8.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009. Cited on page 12.

Chin-Wei Huang, Ricky T. Q. Chen, Christos Tsirigotis, and Aaron Courville. Convex potential flows: Universal probability distributions with optimal transport and convex optimization. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=te7PVH1sPxJ`. Cited on page 7.

Shayan Hundrieser, Gilles Mordant, Christoph Alexander Weitkamp, and Axel Munk. Empirical optimal transport under estimated costs: Distributional limits and statistical applications. *arXiv preprint arXiv:2301.01287*, 2023. Cited on page 8.

Jan-Christian Hütter and Philippe Rigollet. Minimax estimation of smooth optimal transport maps. *The Annals of Statistics*, 49(2):1166–1194, 2021. Cited on pages 7 and 8.

Matt Jacobs and Flavien Léger. A fast approach to optimal transport: The back-and-forth method. *Numerische Mathematik*, 146(3):513–544, 2020. Cited on page 1.

L. Kantorovitch. On the translocation of masses. *C. R. (Doklady) Acad. Sci. URSS (N.S.)*, 37:199–201, 1942. Cited on page 2.

Michal Klein, Aram-Alexandre Pooladian, Pierre Ablin, Eugène Ndiaye, Jonathan Niles-Weed, and Marco Cuturi. Learning costs for structured monge displacements. *arXiv preprint arXiv:2306.11895*, 2023. Cited on page 8.

Alexander Korotin, Vage Egiazarian, Arip Asadulaev, Alexander Safin, and Evgeny Burnaev. Wasserstein-2 generative networks. *arXiv preprint arXiv:1909.13082*, 2019. Cited on pages 1, 3, and 4.

Alexander Korotin, Lingxiao Li, Aude Genevay, Justin M Solomon, Alexander Filippov, and Evgeny Burnaev. Do neural optimal transport solvers work? a continuous wasserstein-2 benchmark. *Advances in Neural Information Processing Systems*, 34:14593–14605, 2021. Cited on page 4.

Takeshi Koshizuka and Issei Sato. Neural lagrangian schrödinger bridge. *arXiv preprint arXiv:2204.04853*, 2022. Cited on pages 5, 6, and 13.

Christian Léonard. From the Schrödinger problem to the Monge–Kantorovich problem. *Journal of Functional Analysis*, 262(4):1879–1920, 2012. Cited on page 6.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. Cited on page 7.

Guan-Horng Liu, Tianrong Chen, Oswin So, and Evangelos A Theodorou. Deep generalized schrödinger bridge. *arXiv preprint arXiv:2209.09893*, 2022. Cited on pages 1, 5, 6, and 13.

Frederike Lübeck, Charlotte Bunne, Gabriele Gut, Jacobo Sarabia del Castillo, Lucas Pelkmans, and David Alvarez-Melis. Neural unbalanced optimal transport via cycle-consistent semi-couplings. *arXiv preprint arXiv:2209.15621*, 2022. Cited on page 7.

Ashok Makkuva, Amirhossein Taghvaei, Sewoong Oh, and Jason Lee. Optimal transport mapping via input convex neural networks. In *International Conference on Machine Learning*, pages 6672–6681. PMLR, 2020. Cited on pages 3 and 4.

Tudor Manole, Sivaraman Balakrishnan, Jonathan Niles-Weed, and Larry Wasserman. Plugin estimation of smooth optimal transport maps. *arXiv preprint arXiv:2107.12364*, 2021. Cited on page 7.

Tudor Manole, Patrick Bryant, John Alison, Mikael Kuusela, and Larry Wasserman. Background modeling for double higgs boson production: Density ratios and optimal transport. *arXiv preprint arXiv:2208.02807*, 2022. Cited on page 1.

Sky McKinley and Megan Levine. Cubic spline interpolation. *College of the Redwoods*, 45(1):1049–1060, 1998. Cited on page 12.

Maksim Nekrashevich, Alexander Korotin, and Evgeny Burnaev. Neural gromov-wasserstein optimal transport. *arXiv preprint arXiv:2303.05978*, 2023. Cited on page 8.

Derek Onken, Samy Wu Fung, Xingjian Li, and Lars Ruthotto. Ot-flow: Fast and accurate continuous normalizing flows via optimal transport. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9223–9232, 2021. Cited on page 7.

Brendan Pass. Multi-marginal optimal transport: theory and applications. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 49(6):1771–1790, 2015. Cited on page 8.

Aram-Alexandre Pooladian and Jonathan Niles-Weed. Entropic estimation of optimal transport maps. *arXiv preprint arXiv:2109.12004*, 2021. Cited on page 7.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky Chen. Multisample flow matching: Straightening flows with minibatch couplings. *arXiv preprint arXiv:2304.14772*, 2023. Cited on page 7.

Litu Rout, Alexander Korotin, and Evgeny Burnaev. Generative modeling with optimal transport maps. *arXiv preprint arXiv:2110.02999*, 2021. Cited on page 7.

Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015. Cited on page 2.

Christopher Scarvelis and Justin Solomon. Riemannian metric learning via optimal transport. *International Conference on Learning Representations*, 2023. Cited on pages 1, 2, 4, 5, 6, 7, and 13.

Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, et al. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176 (4):928–943, 2019. Cited on pages 1, 4, and 7.

Vivien Seguy, Bharath Bhushan Damodaran, Rémi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. Large-scale optimal transport and mapping estimation. In *International Conference on Learning Representations*, 2018. Cited on page 7.

Amirhossein Taghvaei and Amin Jalali. 2-wasserstein approximation via restricted convex potentials with application to improved training for gans. *arXiv preprint arXiv:1902.07197*, 2019. Cited on pages 3 and 4.

Alexander Tong, Jessie Huang, Guy Wolf, David Van Dijk, and Smita Krishnaswamy. Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics. In *International conference on machine learning*, pages 9526–9536. PMLR, 2020. Cited on page 7.

Alexander Tong, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Kilian Fatras, Guy Wolf, and Yoshua Bengio. Conditional flow matching: Simulation-free dynamic optimal transport. *arXiv preprint arXiv:2302.00482*, 2023. Cited on page 7.

Théo Uscidda and Marco Cuturi. The monge gap: A regularizer to learn all transport maps. *arXiv preprint arXiv:2302.04953*, 2023. Cited on page 8.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009. Cited on pages 1, 2, and 3.

Eric W Weisstein. Cubic spline, from mathworld—a wolfram web resource, 2008. Cited on page 12.

George Wolberg. Cubic spline interpolation: A review, 1988.
Cited on page 12.

A. Details on the use of splines for geodesics and paths

This section provides more background information and details behind the amortized splines in Section 3 that address Challenge 2.

A.1. Cubic splines

Cubic splines, *e.g.* as reviewed in McKinley and Levine (1998); Wolberg (1988); Bartels et al. (1995); Weisstein (2008); Hastie et al. (2009); Burden et al. (2015), are a widely-used method for fitting a parametric function to data. We start with a review of general splines in one dimension (Appendix A.1.1), then extend those to multiple dimensions (Appendix A.1.2), then use those for representing the Lagrangian paths and geodesics (Appendix A.2), then amortize those (Appendix A.3).

A.1.1. . . . IN ONE DIMENSION

In one dimension, a cubic spline is defined by

$$\gamma(x) := \begin{cases} \gamma_1(x) & \text{if } x_1 \leq x < x_2 \\ \gamma_2(x) & \text{if } x_2 \leq x < x_3 \\ \gamma_{n-1}(x) & \text{if } x_{n-1} \leq x < x_n \end{cases} \tag{33}$$

where $x \in \mathbb{R}$, x_i for $i \in \{1, \ldots, n\}$ are the *knot points* and

$$\gamma_i(x) := a_i + b_i x + c_i x^2 + d_i x^3 \tag{34}$$

are the cubic component functions with coefficients $\bar{\varphi}_i := [a_i, b_i, c_i, d_i]$. We write the vector of all coefficients as $\bar{\varphi} := [\bar{\varphi}_1, \ldots, \bar{\varphi}_{n-1}]$.

Challenge 3 (Parameterizing splines)**.** *The coefficients $\bar{\varphi}$ are unknown and fit to data. While they could be taken directly as the parameters for γ, it would not result in a continuous function around the knot points.*

The standard approach to resolve these discontinuities is to constrain the component functions to be continuous and have matching values and derivatives

$$\begin{aligned} \gamma_i(x_{i+1}) &= \gamma_{i+1}(x_{x+1}) \text{ for } i \in \{1, \ldots, n-1\} \\ \gamma_i'(x_{i+1}) &= \gamma_{i+1}'(x_{x+1}) \text{ for } i \in \{1, \ldots, n-1\} \\ \gamma_i''(x_{i+1}) &= \gamma_{i+1}''(x_{x+1}) \text{ for } i \in \{1, \ldots, n-1\}. \end{aligned} \tag{35}$$

These constraints, along with other conditions can be used to provide a set of basis vectors $B := [b_i]_{i=1}^m$ where $b_i \in |\bar{\varphi}|$ of spline parameterizations $\bar{\varphi}$ that satisfy Eq. (35), *e.g.* as in Hastie et al. (2009, Section 5.2.1). In other words, any linear combination of the basis vectors b_i will result in a valid parameterization. We can thus reparameterize the spline with $\varphi \in \mathbb{R}^m$ to be based on linear combinations of the basis, providing

$$\bar{\varphi} := B\varphi = \sum b_i \varphi_i \tag{36}$$

The advantage of this reparameterization is that φ is a parameterization of splines in the unconstrained reals and can therefore be treated as a standard learnable parameter for our geodesic computations.

A.1.2. . . . IN MULTIPLE DIMENSIONS

The standard extension of splines to functions of multiple dimensions, *e.g.* for graphics (Bartels et al., 1995), is to parameterize a one-dimensional spline Eq. (33) on each coordinate. We will notate these as $\gamma_\varphi : \mathbb{R} \to \mathbb{R}^d$, $\gamma_\varphi(x) := [\gamma_{\varphi_1}(x), \ldots, \gamma_{\varphi_d}(x)]$ where φ_i is the parameterization of the basis coefficients for each one-dimensional spline.

A.2. Cubic splines for geodesics and Lagrangian paths

We follow Beik-Mohammadi et al. (2021); Detlefsen et al. (2021) and represent geodesics and Lagrangian paths between two points x, y by a multi-dimensional spline $\gamma_\varphi(t)$ parameterized by φ where $t \in [0, 1]$ is the time. The basis for the splines enforce the smoothness properties in Eq. (35) as well as the boundary conditions $\gamma_\varphi(0) = x$ and $\gamma_\varphi(1) = y$.

A.3. Amortized cubic splines for geodesics

Instead of computing the spline parameters φ individually for every geodesic, we propose to *amortize* them across the geodesics needed for the OT maps. This results in parameterizing an amortization model $\varphi_\eta(x, y)$ that predicts the spline parameters for a geodesic between x and y that we learn with objective-based amortization in Eq. (23).

B. Synthetic data for Lagrangians with potentials

We consider five potential functions $U(x)$ in this work. The following four potential functions are from (Koshizuka and Sato, 2022):

$$U_{\text{box}}(x) \coloneqq -M_1 \cdot \mathbf{1}_{[-0.5,0.5]^2}(x)\,, \tag{37}$$

$$U_{\text{slit}}(x) \coloneqq -M_2 \cdot \left(\mathbf{1}_{([-0.1,0.1],(-\infty,-0.25])}(x_1, x_2) + \mathbf{1}_{([-0.1,0.1],[0.25,\infty))}(x_1, x_2)\right), \tag{38}$$

$$U_{\text{hill}}(x) \coloneqq -M_3\|x\|^2\,, \tag{39}$$

$$U_{\text{well}}(x) \coloneqq -M_4 \exp(-\|x\|^2)\,, \tag{40}$$

where M_1, M_2, M_3 and M_4 are constants.

The Gaussian-mixture example is taken from (Liu et al., 2022), which amounts to the following potential function

$$U_{\text{GMM}}(x) \coloneqq -M_5 \sum_{i=1}^{3} \mathbf{1}_{B_i}(x)\,, \tag{41}$$

where $B_i \coloneqq \{x : \|x - m_i\| \leq 1.5\}$ with $m_i \in \{(6, 6), (6, -6), (-6, -6)\}$.

We approximate the hard constraints using sigmoid functions. We make the choices $M_1 = 0.01$, $M_2 = 1$, $M_3 = 0.05$, $M_4 = 0.01$, $M_5 = 0.1$ — we are unable to use the same choice of M for all potentials as a result of numerical instabilities that arise in the geodesic computation.

C. Data from (Scarvelis and Solomon, 2023)

We briefly outline the three datasets used in Section 4, all of which were taken directly from (Scarvelis and Solomon, 2023), following their open source repository `https://github.com/cscarv/riemannian-metric-learning-ot`; here we simply explain the data generating processes.

The three datasets have a similar flavor: Let γ be a time-varying curve, and suppose we have access to the matrix function $A(\cdot)$ which generates the known geometry. This allows the authors to generate a velocity field between two fixed points x and y (respectively, initial and final position of γ) using the following optimization problem

$$\min_\theta \int_0^1 \|v_{(t,\theta)}(\gamma_t)\|^2_{A(\gamma_t)}\, \mathrm{d}t + \|\gamma(1) - y\|\,, \tag{42}$$

where $v_{(t,\theta)}(\cdot)$ is a time-varying neural network (parametrized by θ) that is the solution to a neural ODE, where they also enforce the initial condition $\gamma_0 = x$. The integral in time is replaced with a sum over indices $0 = t_1 < t_2 < \ldots < t_m = 1$. For a given collection of samples from measures $\{\rho_i\}_{i=1}^{K-1}$, the authors randomly pair up the data and solve Equation (42) across batches using the (Chen, 2018) package (specifically using `odeint`). Equation (42) is solved using AdamW with a learning rate of 10^{-3} and weight-decay factor 10^{-3}, with 100 epochs of training per pair of samples. The learned solution $v_{(t,\hat{\theta})}$ is able to generate data at various time-points. With this setup in mind, we can turn to precise details for the three datasets.

Circular trajectory The circular path is enforced using the matrix

$$A(x) \coloneqq \begin{pmatrix} \frac{x_1^2}{\|x\|^2} & 1 - \frac{x_1 x_2}{\|x\|^2} \\ 1 - \frac{x_1 x_2}{\|x\|^2} & \frac{x_2^2}{\|x\|^2} \end{pmatrix}. \tag{43}$$

The goal is to generate Gaussian data that flows according to A. To this end, the authors fix four possible means (in order) $\mu \in \{(1, 0), (0, 1), (-1, 0), (0, -1)\}$, and fix $\sigma \coloneqq 0.1$, which define $\rho_i \coloneqq N(\mu_i, \sigma^2)$. 100 samples are drawn from each ρ_i, which constitutes the finite-samples that are used in the objective function Eq. (42). Once the velocity field is learned, there are 24 equispaced time-points from which they draw samples, resulting in 24 Gaussian distributions that flow according to A.

Mass-splitting trajectory In this example, $A(x) \coloneqq I - w(x)w^\top(x)$, with

$$w(x) \coloneqq \begin{cases} \left(\frac{1}{\sqrt{2}} . \frac{1}{\sqrt{2}}\right) & x_2 \geq 0\,. \\ \left(\frac{1}{\sqrt{2}} . \frac{-1}{\sqrt{2}}\right) & x_2 < 0\,. \end{cases} \tag{44}$$

In this case, there are three Gaussians, with means $\mu_i \in \{(0,0), (10,10), (10,-10)\}$ and unit variance. Again, 100 samples are drawn from each, which are randomly paired and allow the authors to numerically solve Eq. (42). Once they have a learned vector field, they generate the data at 10 equispaced time-points.

X-path trajectory In this third case example, $A(x) \coloneqq I - w(x)w^\top(x)$, with

$$w(x) \coloneqq \alpha(x)w_1(x) + \beta(x)w_2(x)\,, \tag{45}$$

with $w_1(x) \coloneqq (\frac{1}{\sqrt{2}} . \frac{1}{\sqrt{2}})$ and $w_2(x) \coloneqq (\frac{1}{\sqrt{2}} . \frac{-1}{\sqrt{2}})$, and $\alpha(x) \coloneqq 1.25\tanh(\text{ReLU}(x_1 x_2)$ and $\beta(x) \coloneqq -1.25\tanh(\text{ReLU}(-x_1 x_2)$. Here, there are two sets of two trajectories, corresponding to Gaussian data with means $\mu_i^{(1)} \in \{(-1,-1), (1,1)\}$ and $\mu_i^{(2)} \in \{(-1,1), (1,-1)\}$, all with standard deviation $\sigma \coloneqq 0.1$. As before, 100 samples are generated, and Eq. (42) is solved (twice) numerically; 10 time-points per velocity field are used to generate the total data.

D. Training specifications

D.1. Source code

We will publicly release the source code to reproduce every detail of our paper. A single training run for all of the experimental settings takes approximately 1-3 hours on our NVIDIA Tesla V100 GPU.

D.2. Hyper-parameters

Table 2. Hyper-parameters for computing the OT maps in Figs. 1 and 2 with Alg. 1.

Hyper-Parameter	Value
Number of spline knots	30
g_θ MLP layer sizes	`[64, 64, 64, 64]`
y_ϕ MLP layer sizes	`[64, 64, 64, 64]`
γ_φ MLP layer sizes	`[1024, 1024]`
MLP activations	Leaky ReLU
g_θ learning rate schedule	Cosine (starting at 10^{-4} and annealing to 10^{-2})
y_ϕ learning rate schedule	Cosine (starting at 10^{-4} and annealing to 10^{-2})
γ_φ learning rate	10^{-4} (no schedule)
Batch size	1024
c-transform solver	LBFGS (20 iterations, backtracking Armijo line search)

Table 3. Hyper-parameters for computing Figs. 3 and 4 and Table 1 with Alg. 1 and Eq. (25).

Hyper-Parameter	Value
Number of spline knots	30
g_θ MLP layer sizes	`[64, 64, 64, 64]`
y_ϕ MLP layer sizes	`[64, 64, 64, 64]`
γ_φ MLP layer sizes	`[1024, 1024]`
MLP activations	Leaky ReLU
g_θ, y_ϕ, γ_φ learning rates	10^{-4} no schedule
Batch size	1024
c-transform solver	LBFGS (20 iterations, backtracking Armijo line search)
A_ϑ learning rate	$5 \cdot 10^{-3}$
Update frequency	1 update of A_ϑ for every 10 updates of g_θ, y_ϕ, and γ_φ