GeONet: A neural operator for learning the Wasserstein geodesic

Anonymous authors
Paper under double-blind review

Abstract

Optimal transport (OT) offers a versatile framework to compare complex data distributions in a geometrically meaningful way. Traditional methods for computing the Wasserstein distance and geodesic between probability measures require mesh-dependent domain discretization and suffer from the curse-of-dimensionality. We present *GeONet*, a mesh-invariant deep neural operator network that learns the non-linear mapping from the input pair of initial and terminal distributions to the Wasserstein geodesic connecting the two endpoint distributions. In the offline training stage, GeONet learns the saddle point optimality conditions for the dynamic formulation of the OT problem in the primal and dual spaces that are characterized by a coupled PDE system. The subsequent inference stage is instantaneous and can be deployed for real-time predictions in the online learning setting. We demonstrate that GeONet achieves comparable testing accuracy to the standard OT solvers on a simulation example and the CIFAR-10 dataset with considerably reduced inference-stage computational cost by orders of magnitude.

1 Introduction

Recent years have seen tremendous progress in statistical and computational optimal transport (OT) as a lens to explore machine learning problems. One prominent example is to use the Wasserstein distance to compare data distributions in a geometrically meaningful way, which has found various applications, such as in generative models (Arjovsky et al., 2017), domain adaptation (Courty et al., 2017) and computational geometry (Solomon et al., 2015). Computing the optimal coupling and the optimal transport map (if it exists) can be expressed in a fluid dynamics formulation with the minimum kinetic energy (Benamou and Brenier, 2000). Such dynamical formulation defines geodesics in the Wasserstein space of probability measures, thus providing richer information for interpolating between data distributions that can be used to design efficient sampling methods from high-dimensional distributions (Finlay et al., 2020). Moreover, the Wasserstein geodesic is also closely related to the optimal control theory (Chen et al., 2021), which has applications in robotics and control systems (Krishnan and Martínez, 2018; Inoue et al., 2021).

Traditional methods for numerically computing the Wasserstein distance and geodesic require domain discretization that is often mesh-dependent (i.e., on regular grids or triangulated domains). Classical solvers such as Hungarian method (Kuhn, 1955), the auction algorithm (Bertsekas and Castanon, 1989), and transportation simplex (Luenberger and Ye, 2015), suffer from the curse-of-dimensionality and scale poorly for even moderately mesh-sized problems (Klatt et al., 2020; Genevay et al., 2016; Benamou and Brenier, 2000). Entropic regularized OT (Cuturi, 2013) and the Sinkhorn algorithm (Sinkhorn, 1964) have been shown to efficiently approximate the OT solutions at low computational cost, handling high-dimensional distributions (Benamou et al., 2015). However, high accuracy is computationally obstructed with a small regularization parameter (Altschuler et al., 2017; Dvurechensky et al., 2018). Recently, a machine learning method to compute the Wasserstein geodesic for a *given* input pair of probability measures has been considered in (Liu et al., 2021).

A major challenge of using the OT-based techniques is that one needs to recompute the Wasserstein distance and geodesic for new input pair of probability measures. Thus, issues of scalability on large-scale datasets and suitability in the online learning setting are serious concerns for modern machine learning, computer graphics, and natural language processing tasks (Genevay et al., 2016;

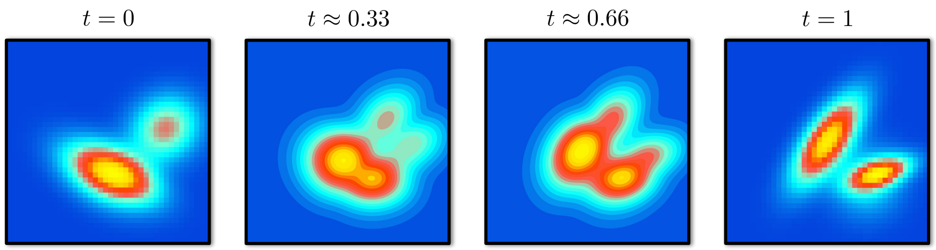

Figure 1: A predicted geodesic from our mesh-invariant GeONet at different spatial resolutions, namely, sampling points (i.e., collocations) are randomly taken in training and low-resolution inputs can be adapted into high-resolution geodesics (i.e., super-resolution).

Solomon et al., 2015; Kusner et al., 2015). This motivates us to tackle the problem of learning the Wasserstein geodesic from an *operator learning* perspective.

There is a recent line of work on learning neural operators for solving general differential equations or discovering equations from data, including DeepONet (Lu et al., 2021), Fourier Neural Operators (Li et al., 2020b), and physics-informed neural networks/operators PINNs (Raissi et al., 2019) and PINOs (Li et al., 2021). Those methods are mesh-independent, data-driven, and designed to accommodate specific physical laws governed by certain partial differential equations (PDEs).

Our contributions. In this paper, we propose a deep neural operator learning framework *GeONet* for the Wasserstein geodesic. Our method is based on learning the optimality conditions in the dynamic formulation of the OT problem, which is characterized by a coupled PDE system in the primal and dual spaces. Our main idea is to recast the learning problem of the Wasserstein geodesic from training data into an operator learning problem for the solution of the PDEs corresponding to the primal and dual OT dynamics. Our method does not depend on any pre-specified mesh and it can learn the highly non-linear Wasserstein geodesic operator from a wide collection of training distributions that are not necessarily sampled at the same locations in the domain. Therefore, GeONet is also suitable for zero-shot super-resolution applications on images, i.e., it is trained on lower resolution and predicts at higher resolution without seeing any higher resolution data (Shocher et al., 2018). See Figure 1 for an example of the predicted higher-resolution Wasserstein geodesic connecting two lower-resolution Gaussian mixture distributions by GeONet.

Surprisingly, the training of our GeONet does not require the true geodesic data for connecting the two endpoint distributions. Instead, it only requires the training data as boundary pairs of initial and terminal distributions. The reason that GeONet needs much less input data is because its training process is implicitly informed by the OT dynamics such that the continuity equation in the primal space and Hamilton-Jacobi equation in the dual space must be simultaneously satisfied to ensure zero duality gap. Since the geodesic data are typically difficult to obtain without resorting to some traditional numerical solvers, the *amortized inference* nature of GeONet, where inference on related training pairs can be reused (Gershman and Goodman, 2014), has substantial computational advantage over standard computational OT methods and machine learning methods for computing the geodesic designed for single input pair of distributions (Peyré and Cuturi, 2019; Liu et al., 2021).

Table 1: Comparison of our method GeONet with other existing neural operators and networks for learning dynamics from data. PINN can be found in (Raissi et al., 2019) and the machine learning based minimax method can be found in (Liu et al., 2021).

Method characteristic	Classical NNs	PINNs	Minimax	GeONet (Ours)
no need for retraining for new input	✓			✓
satisfies the associated PDEs		✓	✓	✓
does not require known geodesic data		✓	✓	✓
mesh independence		✓	✓	✓

Once GeONet training is complete, the inference stage for predicting the geodesic connecting new initial and terminal data distributions requires only a forward pass of the network and thus it can be performed in real time. In contrast, standard OT methods re-compute the Wasserstein distance and geodesic for each new input distribution pair. This is an appealing feature of amortized inference to use a pre-trained GeONet for fast geodesic computation or fine-tuning on a large number of future data distributions. Detailed comparison between our proposed method GeONet with other existing neural operators and networks for learning dynamics from data can be found in Table 1.

The rest of the paper is organized as follows. In Section 2, we review some background on static and dynamic formulations for the OT problem, as well as on the neural operator learning framework. In Section 3, we present the proposed GeONet method. In Section 4, we report the numeric performance on a synthetic experiment and a real image dataset.

2 BACKGROUND

2.1 OPTIMAL TRANSPORT PROBLEM: STATIC AND DYNAMIC FORMULATIONS

The optimal mass transportation problem, first considered by the French engineer Gaspard Monge, is to find an optimal map T^* for transporting a source distribution μ_0 to a target distribution μ_1 that minimizes some cost function $c : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$:

$$\min_{T:\mathbb{R}^d \to \mathbb{R}^d} \left\{ \int_{\mathbb{R}^d} c(x, T(x)) \, \mathrm{d}\mu_0(x) : T_\sharp \mu_0 = \mu_1 \right\}, \tag{1}$$

where $T_\sharp \mu$ denotes the pushforward measure defined by $(T_\sharp \mu)(B) = \mu(T^{-1}(B))$ for measurable subset $B \subset \mathbb{R}^d$. In this paper, we focus on the quadratic cost $c(x, y) = \frac{1}{2}\|x - y\|_2^2$. The Monge problem (1) induces a metric, known as the *Wasserstein distance*, on the space $\mathcal{P}_2(\mathbb{R}^d)$ of probability measures on $\mathbb{R}^d$ with finite second moments. In particular, the 2-Wasserstein distance can be expressed in the relaxed Kantorovich form:

$$W_2^2(\mu_0, \mu_1) := \min_{\gamma \in \Gamma(\mu_0, \mu_1)} \left\{ \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|_2^2 \, \mathrm{d}\gamma(x, y) \right\}, \tag{2}$$

where minimization over γ runs over all possible couplings $\Gamma(\mu_0, \mu_1)$ with marginal distributions μ_0 and μ_1. Problem (2) has the dual form (cf. Villani (2003))

$$\frac{1}{2} W_2^2(\mu_0, \mu_1) = \sup_{\varphi \in L^1(\mu_0), \, \psi \in L^1(\mu_1)} \left\{ \int_{\mathbb{R}^d} \varphi \, \mathrm{d}\mu_0 + \int_{\mathbb{R}^d} \psi \, \mathrm{d}\mu_1 : \varphi(x) + \psi(y) \leqslant \frac{\|x - y\|_2^2}{2} \right\}. \tag{3}$$

Problems (1) and (2) are both referred as the *static OT* problem, which has close connection to fluid dynamics. Specifically, the Benamou-Brenier dynamic formulation (Benamou and Brenier, 2000) expresses the Wasserstein distance as a minimal kinetic energy flow problem:

$$\begin{gathered} \frac{1}{2} W_2^2(\mu_0, \mu_1) = \min_{(\mu, \mathbf{v})} \int_0^1 \int_{\mathbb{R}^d} \frac{1}{2} ||\mathbf{v}(x, t)||_2^2 \, \mu(x, t) \, \mathrm{d}x \, \mathrm{d}t \\ \text{subject to } \; \partial_t \mu + \mathrm{div}(\mu \mathbf{v}) = 0, \;\; \mu(\cdot, 0) = \mu_0, \;\; \mu(\cdot, 1) = \mu_1, \end{gathered} \tag{4}$$

where $\mu_t := \mu(\cdot, t)$ is the probability density flow at time t satisfying the continuity equation (CE) constraint $\partial_t \mu + \mathrm{div}(\mu \mathbf{v}) = 0$ that ensures the conservation of unit mass along the flow $(\mu_t)_{t \in [0,1]}$. To solve (4), we apply the Lagrange multiplier method to find the saddle point in the primal and dual variables. In particular, for any flow μ_t initializing from μ_0 and terminating at μ_1, the Lagrangian function for (4) can be written as

$$\mathcal{L}(\mu, \mathbf{v}, u) = \int_0^1 \int_{\mathbb{R}^d} \left[\frac{1}{2} \|\mathbf{v}\|_2^2 \mu + (\partial_t \mu + \mathrm{div}(\mu \mathbf{v})) \, u \right] \, \mathrm{d}x \, \mathrm{d}t, \tag{5}$$

where $u := u(x, t)$ is the dual variable for the continuity equation. Using integration-by-parts under suitable decay condition for $\|x\|_2 \to \infty$, we find that the optimal dual variable u^* satisfies the Hamilton-Jacobi (HJ) equation for the dynamic OT problem

$$\partial_t u + \frac{1}{2} \|\nabla u\|_2^2 = 0, \tag{6}$$

and the optimal velocity vector field is given by $\mathbf{v}^*(x,t) = \nabla u^*(x,t)$. Hence, we obtained that the Karush–Kuhn–Tucker (KKT) optimality conditions for (5) are solution (μ^*, u^*) to the following system of partial differential equations (PDEs):

$$\begin{cases} \partial_t \mu + \operatorname{div}(\mu \nabla u) = 0, \ \ \partial_t u + \frac{1}{2}\|\nabla u\|_2^2 = 0, \\ \qquad \mu(\cdot, 0) = \mu_0, \ \ \mu(\cdot, 1) = \mu_1. \end{cases} \tag{7}$$

In addition, solution to the Hamilton-Jacobi equation (6) can be viewed as an interpolation $u(x,t)$ of the Kantorovich potential between the initial and terminal distributions in the sense that $u^*(x,1) = \psi^*(x)$ and $u^*(x,0) = -\varphi^*(x)$ (both up to some additive constants), where ψ^* and φ^* are the optimal Kantorovich potentials for solving the static dual OT problem (3). A detailed derivation of the primal-dual optimality conditions for the dynamical OT formulation is provided in Appendix A.

2.2 Learning neural operators

A neural operator generalizes a neural network that learns a mapping $\Gamma^\dagger : \mathcal{A} \to \mathcal{U}$ between infinite-dimensional function spaces $\mathcal{A}$ and $\mathcal{U}$ (Kovachki et al., 2021; Li et al., 2020a). Typically, $\mathcal{A}$ and $\mathcal{U}$ contain functions defined over a space-time domain $\Omega \times [0,T]$, where Ω is taken as a subset of $\mathbb{R}^d$, and the mapping of interest $\Gamma^\dagger$ is implicitly defined through certain differential operator. For example, the physics informed neural network (PINN) (Raissi et al., 2019) aims to use a neural network to find a solution to the PDE

$$\partial_t u + \mathcal{D}[u] = 0, \tag{8}$$

given the boundary data $u(\cdot,0) = u_0$ and $u(\cdot,T) = u_T$, where $\mathcal{D} := \mathcal{D}(a)$ denotes a non-linear differential operator in space that may depend on the input function $a \in \mathcal{A}$. Different from the classical neural network learning paradigm that is purely data-driven, a PINN has less input data (i.e., some randomly sampled data points from the solution $u = \Gamma^\dagger(a)$ and the boundary conditions) since the solution operator $\Gamma^\dagger$ has to obeys the induced physical laws governed by (8). Even though the PINN is mesh-independent, it only learns the solution for a *single* instance of the input function a in the PDE (8). In order to learn the dynamical behavior of the inverse problem $\Gamma^\dagger : \mathcal{A} \to \mathcal{U}$ for an entire family of $\mathcal{A}$, we consider the operator learning perspective.

The idea of using neural networks to approximate any non-linear continuous operator stems from the universal approximation theorem for operators (Chen and Chen, 1995; Lu et al., 2021). In particular, we construct a parametric map by a neural network $\Gamma : \mathcal{A} \times \Theta \to \mathcal{U}$ for a finite-dimensional parameter space Θ to approximate the true solution operator $\Gamma^\dagger$. In this paper, we adopt the *DeepONets* architecture to model Γ (Lu et al., 2021). We refer the readers to Appendix E for some basics of DeepONet and its enhanced versions. Then, the neural operator learning problem for finding the optimal $\theta^* \in \Theta$ can be done in the classical risk minimization framework via

$$\begin{aligned} \theta^* = \operatorname{argmin}_{\theta\in\Theta} \quad & \mathbb{E}_{(a,u_0,u_T)\sim\mu} \Big[\big\| (\partial_t + \mathcal{D})\Gamma(a,\theta) \big\|^2_{L^2(\Omega\times(0,T))} \\ & + \lambda_0 \big\| \Gamma(a,\theta)(\cdot,0) - u_0 \big\|^2_{L^2(\Omega)} + \lambda_T \big\| \Gamma(a,\theta)(\cdot,T) - u_T \big\|^2_{L^2(\Omega)} \Big], \end{aligned} \tag{9}$$

where the input data (a, u_0, u_T) are sampled from some joint distribution μ. In (9), we minimize the PDE residual corresponding to $\partial_t u + \mathcal{D}[u] = 0$ while constraining the network by imposing boundary conditions. The loss function has weights $\lambda_0, \lambda_T > 0$. Given a finite sample $\{(a^{(i)}, u_0^{(i)}, u_T^{(i)})\}_{i=1}^n$, and data points randomly (typically uniformly) sampled in the space-time domain $\Omega \times (0,T)$, we may minimize the empirical loss analog of (9) by replacing $\|\cdot\|_{L^2(\Omega\times(0,T))}$ with the discrete L^2 norm over domain $\Omega \times (0,T)$. Computation of the exact differential operators ∂_t and $\mathcal{D}$ can be conveniently exploited via automatic differentiation in standard deep learning packages.

Neural operators offer advantages over more traditional neural networks that typically depend on some fixed mesh discretization into a finite-dimensional space and require the retraining for each different input (a, u_0, u_T) to the dynamical system (8). In contrast, neural operators performs amortized inference to learn the dynamics for a family of inputs and thus no retraining is needed for the subsequent inference stage. In addition, neural operators are flexible enough to transmute context from the original problem into a more targeted task, producing a discretization-invariant approach that often achieves more desirable results such as in the zero-shot super-resolution applications. See more details in Section 4.3.

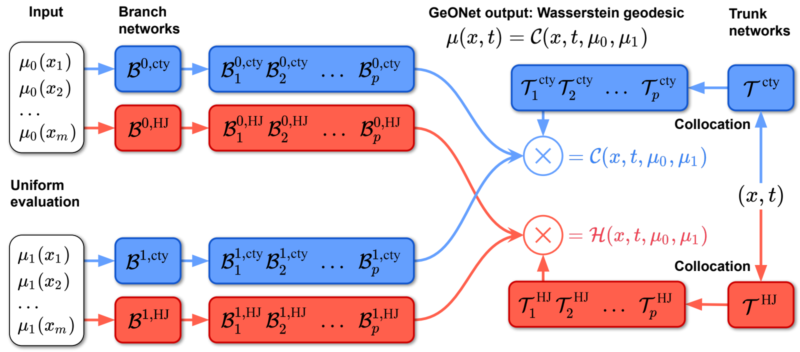

Figure 2: Architecture of GeONet, containing six neural networks to solve the continuity and Hamilton-Jacobi (HJ) equations, three for each. We minimize the total loss, and the continuity solution yields the geodesic. GeONet branches and trunks output vectors of dimension p, in which we perform multiplication among neural network elements to produce the continuity and HJ solutions.

3 Our method

We present *GeONet*, a geodesic operator network for learning the Wasserstein geodesic $\{\mu_t\}_{t\in[0,1]}$ connecting μ_0 to μ_1 from the distance $W_2(\mu_0, \mu_1)$. Let $\Omega \subset \mathbb{R}^d$ be the spatial domain where the probability measures μ_t's are supported. For probability measures $\mu_0, \mu_1 \in \mathcal{P}_2(\Omega)$, it is well-known that the constant-speed geodesic $\{\mu_t\}_{t\in[0,1]}$ between μ_0 and μ_1 is an absolutely continuous curve in the metric space $(\mathcal{P}_2(\Omega), W_2)$, which we denote as $\text{AC}(\mathcal{P}_2(\Omega))$. μ_t solves the kinetic energy minimization problem in (4) (Santambrogio, 2015). Some basic facts on the metric geometry structure of the Wasserstein geodesic and its relation to the fluid dynamic formulation are reviewed and discussed in Appendix B. In this work, our goal is to learn a non-linear operator

$$\Gamma^\dagger : \mathcal{P}_2(\Omega) \times \mathcal{P}_2(\Omega) \to \text{AC}(\mathcal{P}_2(\Omega)), \tag{10}$$
$$(\mu_0, \mu_1) \mapsto \{\mu_t\}_{t\in[0,1]}, \tag{11}$$

based on a training dataset $\{(\mu_0^{(1)}, \mu_1^{(1)}), \dots, (\mu_0^{(n)}, \mu_1^{(n)})\}$. The core idea of GeONet is to learn the KKT optimality condition (7) for the Benamou-Brenier problem. Since (7) is derived to ensure the zero duality gap between the primal and dual dynamic OT problems, solving the Wasserstein geodesic requires us to introduce two sets of neural networks that train the coupled PDEs simultaneously. Specifically, we model the operator learning problem as an enhanced version of the unstacked DeepONet architecture (Lu et al., 2021; Tan and Chen, 2022) by jointly training three primal networks in (12) and three dual networks in (13) as follows:

$$\mathcal{C}(x, t, \mu_0, \mu_1, \phi) = \sum_{k=1}^{p} \mathcal{B}_k^{0,\text{cty}}(\mu_0, \theta^{0,\text{cty}}) \cdot \mathcal{B}_k^{1,\text{cty}}(\mu_1, \theta^{1,\text{cty}}) \cdot \mathcal{T}_k^{\text{cty}}(x, t, \xi^{\text{cty}}), \tag{12}$$
$$\mathcal{H}(x, t, \mu_0, \mu_1, \psi) = \sum_{k=1}^{p} \mathcal{B}_k^{0,\text{HJ}}(\mu_0, \theta^{0,\text{HJ}}) \cdot \mathcal{B}_k^{1,\text{HJ}}(\mu_1, \theta^{1,\text{HJ}}) \cdot \mathcal{T}_k^{\text{HJ}}(x, t, \xi^{\text{HJ}}), \tag{13}$$

where $\mathcal{B}^{j,\text{cty}}(\mu_0(x_1), \dots, \mu_0(x_m), \theta^{j,\text{cty}}) : \mathbb{R}^m \times \Theta \to \mathbb{R}^p$ and $\mathcal{B}^{j,\text{HJ}}(\mu_0(x_1), \dots, \mu_0(x_m), \theta^{j,\text{HJ}}) : \mathbb{R}^m \times \Theta \to \mathbb{R}^p$ are *branch* neural networks taking m-discretized input of initial and terminal density values at $j = 0$ and $j = 1$ respectively, and $\mathcal{T}^{\text{cty}}(x, t, \xi^{\text{cty}}) : \mathbb{R}^d \times [0, 1] \times \Xi \to \mathbb{R}^p$ and $\mathcal{T}^{\text{HJ}}(x, t, \xi^{\text{HJ}}) : \mathbb{R}^d \times [0, 1] \times \Xi \to \mathbb{R}^p$ are *trunk* neural networks taking spatial and temporal inputs (cf. Appendix E for more details on DeepONet models). Here Θ and Ξ are finite-dimensional parameter spaces, and p is the output dimension of the branch and truck networks. Denote parameter concatenations $\phi := (\theta^{0,\text{cty}}, \theta^{1,\text{cty}}, \xi^{\text{cty}})$ and $\psi := (\theta^{0,\text{HJ}}, \theta^{1,\text{HJ}}, \xi^{\text{HJ}})$. Then the primal operator network $\mathcal{C}_\phi(x, t, \mu_0, \mu_1) := \mathcal{C}(x, t, \mu_0, \mu_1, \phi)$ for $\phi \in \Theta \times \Theta \times \Xi$ acts as an approximate solution to the continuity equation, hence the true geodesic $\Gamma^\dagger(x, t, \mu_0(x), \mu_1(x)) := \mu_t(x) = \mu(x, t)$, while the dual operator network $\mathcal{H}_\psi(x, t, \mu_0, \mu_1)$ for $\psi \in \Theta \times \Theta \times \Xi$ corresponds to that of the associated Hamilton-Jacobi equation. The architecture of GeONet is shown in Figure 2.

To train the GeONet defined in (12) and (13), we minimize the empirical loss function corresponding to the system of primal-dual PDEs and boundary residuals in (7) over the parameter spaces Θ and Ξ:

$$\phi^*, \psi^* \quad = \quad \mathrm{argmin}_{\phi,\psi \in \Theta\times\Theta\times\Xi} \quad \mathcal{L}_{\text{cty}} + \mathcal{L}_{\text{HJ}} + \mathcal{L}_{\text{BC}}, \tag{14}$$

where

$$\mathcal{L}_{\text{cty},i} \quad = \quad \frac{\alpha_1}{N} \left\| \quad \frac{\partial}{\partial t}\mathcal{C}_{\phi,i} + \mathrm{div}(\mathcal{C}_{\phi,i}\nabla\mathcal{H}_{\psi,i}) \quad \right\|^2_{L^2(\Omega\times(0,1))}, \tag{15}$$

$$\mathcal{L}_{\text{HJ},i} \quad = \quad \frac{\alpha_2}{N} \left\| \quad \frac{\partial}{\partial t}\mathcal{H}_{\psi,i} + \frac{1}{2}||\nabla\mathcal{H}_{\psi,i}||_2^2 \quad \right\|^2_{L^2(\Omega\times(0,1))}, \tag{16}$$

$$\mathcal{L}_{\text{BC},i} \quad = \quad \frac{\beta_0}{N} \left\| \mathcal{C}_{\phi,0,i} - \mu_0^{(i)} \right\|^2_{L^2(\Omega)} \quad + \quad \frac{\beta_1}{N} \left\| \mathcal{C}_{\phi,1,i} - \mu_1^{(i)} \right\|^2_{L^2(\Omega)}, \tag{17}$$

$\mathcal{L}_{\text{cty}} = \sum_{i=1}^n \mathcal{L}_{\text{cty},i}$, $\mathcal{L}_{HJ} = \sum_{i=1}^n \mathcal{L}_{\text{HJ},i}$ and $\mathcal{L}_{\text{BC}} = \sum_{i=1}^n \mathcal{L}_{\text{BC},i}$. Here, $\mathcal{C}_{\phi,i}(x,t) := \mathcal{C}_\phi(x,t,\mu_0^{(i)}(x),\mu_1^{(i)}(x))$ and $\mathcal{C}_{\phi,t,i}(x) := \mathcal{C}_\phi(x,t,\mu_0^{(i)}(x),\mu_1^{(i)}(x))$ denote the evaluation of neural network $\mathcal{C}_\phi$ over the i-th distribution of the n training data at space location x and time point t. The same notation applies for the Hamilton-Jacobi neural networks. $\mathcal{L}_{\text{cty}}$ is the loss component in which the continuity equation is satisfied, and $\mathcal{L}_{\text{HJ}}$ is the Hamilton-Jacobi equation loss component. The boundary conditions are incorporated in the $\mathcal{L}_{\text{BC}}$ term, and $\alpha_1, \alpha_2, \beta_1, \beta_2$ are scaling weights for the strength to impose the physics-informed loss. Automatic differentiation of our GeONet involves differentiating the coupled DeepONet architecture (cf. Figure 2) in order to compute the physics-informed loss terms, which we discuss in Appendix F.

In practice, we only have access to point clouds of the training distribution data (μ_0, μ_1) and the Wasserstein geodesic is output on a discretized space-time domain. Following (Raissi et al., 2019), we utilize a *collocation* procedure as follows. We sample pairs (x,t) randomly and uniformly within the bounded domain $\Omega \times [0,1]$, i.e., $U(\Omega) \times U([0,1])$. These pairs are resampled during each epoch in our method. Then, we evaluate the continuity and Hamilton-Jacobi residuals displayed in (15) and (16) at such sampled values, in which the loss is subsequently minimized.

In GeONet, we implement the fully connected artificial neural networks (ANNs) for the branch and trunk networks. In our collocation procedure, we take equispaced locations $x_1, \ldots, x_m$ within Ω, a bounded domain in $\mathbb{R}^d$, typically a hypercube, for branch input. We additionally take M spatial and temporal locations (x,t) for the trunk, but these are evaluated randomly and uniformly among the hypercube, different for each density pair (μ_0, μ_1). The total number of training data is $N = nM$. GeONet training pseudo-code is in Algorithm 1.

Algorithm 1 End-to-end training of GeONet

Input: data pairs $(\mu_0^{(1)}\mu_1^{(1)}), \ldots, (\mu_0^{(n)}, \mu_1^{(n)})$; discretization size N; initialization of the neural network parameters $\phi, \psi \in \Theta\times\Theta\times\Xi$; weight parameters $\alpha_1, \alpha_2, \beta_0, \beta_1$.

1: **while** $\mathcal{L}_{\text{total}}$ has not converged **do**
2: Independently draw N sample points $\{(x_\ell^i, t_\ell^i)\}_{\ell=1}^N$ from $U(\Omega)\times U(0,1)$
3: Compute $\Phi_i = \partial_t \mathcal{C}_{\phi,i} + \mathrm{div}(\mathcal{C}_{\phi,i}\nabla\mathcal{H}_{\psi,i})$. ▷ `continuity residual`
4: Compute $\Psi_i = \partial_t \mathcal{H}_{\psi,i} + \frac{1}{2}\|\mathcal{H}_{\psi,i}\|_2^2$. ▷ `Hamilton-Jacobi residual`
5: Compute $B_{0,i} = \mathcal{C}_{\phi,0,i} - \mu_0^{(i)}(x_\ell^i)$, $B_{1,i} = \mathcal{C}_{\phi,1,i} - \mu_1^{(i)}(x_\ell^i)$. ▷ `boundary residual`
6: Compute

$$\mathcal{L}_{\text{cty}} = \frac{\alpha_1}{N}\sum_{i=1}^n \|\Phi_i\|^2_{L^2(\Omega\times(0,1))}, \quad \mathcal{L}_{\text{HJ}} = \frac{\alpha_2}{N}\sum_{i=1}^n \|\Psi_i\|^2_{L^2(\Omega\times(0,1))},$$

$$\mathcal{L}_{\text{BC}} = \frac{1}{N}\sum_{i=1}^n \beta_0\|B_{0,i}\|^2_{L^2(\Omega)} + \beta_1\|B_{1,i}\|^2_{L^2(\Omega)},$$

where $\|f\|^2_{L^2(\Omega\times(0,1))} := \sum_{k=1}^M f^2(x_k,t_k)$ and $\|f\|^2_{L^2(\Omega)} := \sum_{k=1}^M f^2(x_k)$.
7: Compute $\mathcal{L}_{\text{total}}(\phi,\psi) = \mathcal{L}_{\text{cty}} + \mathcal{L}_{\text{HJ}} + \mathcal{L}_{\text{BC}}$.
8: Minimize $\mathcal{L}_{\text{total}}(\phi,\psi)$ to update ϕ and ψ. ▷ `minimize the loss function`
9: **end while**

Entropic regularization. Our GeONet is compatible with entropic regularization, which is related to the Schrödinger bridge problem and stochastic control (Chen et al., 2016). In Appendix C, we propose the *entropic-regularized GeONet* (ER-GeONet), which learns a similar system of KKT conditions for the optimization as in (7). In the zero-noise limit as the entropic regularization parameter $\varepsilon \downarrow 0$, the solution of the optimal entropic interpolating flow converges to solution of the Benamou-Brenier problem (4) in the sense of the method of vanishing viscosity (Mikami, 2004; Evans, 2010).

Gradient enhancement. In practice, we fortify the base method by adding extra residual terms of the differentiated PDEs to our loss function of GeONet. Such gradient enhancement technique has been used to strengthen PINNs (Yu et al., 2022), which improves efficiency as fewer data points are needed to be sampled from $U(\Omega) \times U(0, 1)$, and prediction accuracy as well. We introduce the gradient-enhanced version of GeONet with more details in Appendix D.

4 NUMERICAL EXPERIMENTS

4.1 GAUSSIAN MIXTURE DISTRIBUTIONS

Since finite mixture distributions are powerful universal approximator for continuous probability density functions (Nguyen et al., 2020), we first deploy GeONet on Gaussian mixture distributions over two-dimensional domains. We learn the Wasserstein geodesic mapping between initial and terminal distributions of the form $\mu_j(x) = \sum_{i=1}^{k_j} \pi_i \mathcal{N}(x|u_i, \Sigma_i)$ subject to $\sum_{i=1}^{k_j} \pi_i = 1$, where $j \in \{0, 1\}$ corresponds to initial and terminal distributions μ_0, μ_1, and k_j denotes the number of components taken in the mixture. Here u_i and Σ_i are the mean vectors and covariance matrices of individual Gaussian components, respectively.

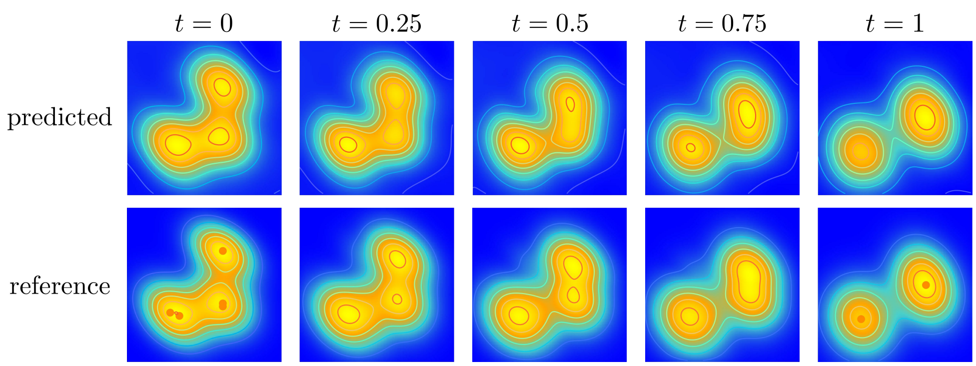

Figure 3: A geodesic predicted by GeONet and a reference geodesic computed by POT on a test Gaussian mixture distribution pair (μ_0, μ_1) with $k_0 = 5$ (with stacked means) and $k_1 = 2$. The reference serves as a close approximation to the true geodesic due to entropic regularization. Gaussian means are shown in the references at $t = 0, 1$.

Table 2: MSE of GeONet on 50 test data of Gaussian mixtures over a 50×50 mesh. All values are multiplied by 10^{-3}. We report the means and standard deviations of the MSE. Training is done with fixed $k_0 = 5, k_1 = 2$ in all cases with π_i randomly chosen. Training data at the boundaries has resolution 30×30.

	GeONet error for Gaussian mixtures				
Number of Gaussians	$t = 0$	$t = 0.25$	$t = 0.5$	$t = 0.75$	$t = 1$
Identity $k_0 = k_1 = 2$	0.44 ± 0.44	1.40 ± 0.73	2.10 ± 1.20	1.40 ± 0.70	0.42 ± 0.43
$k_0 = 3, k_1 = 2$	0.39 ± 0.33	2.60 ± 2.80	4.40 ± 5.10	2.90 ± 2.90	0.57 ± 0.61
$k_0 = 4, k_1 = 2$	0.43 ± 0.37	2.90 ± 2.60	5.10 ± 4.70	3.00 ± 2.70	0.42 ± 0.40
$k_0 = 5, k_1 = 2$	0.33 ± 0.22	2.20 ± 1.80	4.10 ± 3.50	2.60 ± 2.20	0.44 ± 0.51
OOD $k_0 = 2, k_1 = 5$	1.20 ± 0.51	9.30 ± 3.60	16.0 ± 7.70	11.0 ± 5.00	11.0 ± 4.80

Experiment details. In our experiment, domain $\Omega = [0,5] \times [0,5] \subseteq \mathbb{R}^2$ was chosen, which was discretized into a 30×30 grid for GeONet input, meaning the branch networks took vector input of 900 in dimension each. We generate $1,500$ training pairs (μ_0, μ_1) with 900 collocations for each pair, yielding a total of 1.35×10^6 training data points. Recall that GeONet is mesh-invariant, so the 30×30 grids can be adapted to any higher resolution. In Table 4, mean vectors were $u_i \in [1.3, 3.7]^2$, variance $\sigma_{0,i}^2, \sigma_{1,i}^2 \in [0.4, 1.0]$, and covariance $\sigma_{01,i} \in [-0.4, 0.4]$. Additional training details are given in Appendix H.

We use the mean squared error (MSE) $\int_\Omega |\mathcal{C} - \mu|^2 dx$ as our error metric to assess the performance, where μ is a reference geodesic as proxy of the true geodesic without entropic regularization. This integral is made discrete by evaluating a Riemann sum along a mesh. The reference is computed using the Convolutional Wasserstein Barycenter framework within the POT Python library (Solomon et al., 2015; Flamary et al., 2021). POT takes a regularization parameter, which "smooths" the solutions. In our comparison, we choose $\nu = 0.003$.

Performance. Our baseline results were collected by probing GeONet on the identity geodesic by choosing $\mu_0 = \mu_1$ with $k_0 = k_1 = 2$ in Table 4. The baseline identity geodesic provides a benchmark for comparing and interpreting the MSEs across different setups. From Table 4, we can draw the following observations. First, prediction accuracy is the worst at $t = 0.5$ (i.e., the barycenter of μ_0 and μ_1), while the loss boundary conditions (17) allow greater precision for $t = 0, 1$, which suggests that lack of data-enforced conditions along the inner region of the time continuum would cause greater error. Visualization of a geodesic predicted by GeONet and a reference geodesic computed by POT is shown in Figure 3. Second, since the testing pairs $k_0 = 2, 3, 4$ with fixed $k_1 = 2$ can be viewed as degenerate cases of the training setting $k_0 = 5, k_1 = 2$, the *in-distribution* MSEs are comparable across these setups. Third, MSEs for predicting the trivial identity geodesic in the intermediate $t = 0.25, 0.5, 0.75$ are uniformly smaller than other in-distribution setups since the former is an easier task. Finally, we performed a similar experiment with training and testing on the same Gaussian mixture distribution family with $k_0 = k_1 = 5$ and $\pi_i = 0.2$ for all i. Similar conclusions can be drawn, and the MSEs are presented in Appendix G.

Out-of-distribution generalization. To examije the generalization performance of GeONet under distribution shifts, we consider a setup with $k_0 = 2, k_1 = 5$ which is out-of-distribution (OOD) at test time. The last row in Table 4 shows that the MSEs are significantly higher than the in-distribution settings. Figure 4 shows an example for comparing the barycenter at $t = 0.5$ predicted from GeONet and that computed from POT, where the latter serves as the reference to ground truth.

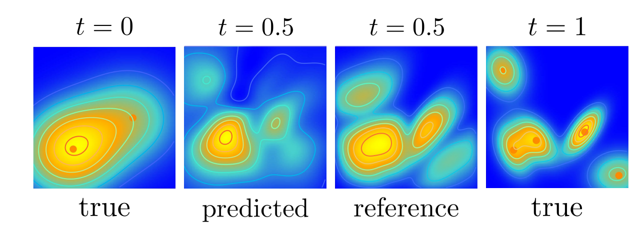

Figure 4: OOD generalization at test time with $k_0 = 2, k_1 = 5$.

4.2 A REAL DATA APPLICATION

Our next experiment was upon the CIFAR-10 dataset 32×32 images. As in Section 4.1, we chose domain $\Omega = [0,5] \times [0,5]$, in which the image RGB color channels are normalized. GeONet took image input of $32 \times 32 = 1,024$ vectors for each branch network. More training details can be found in Appendix H. In our experiment, we stack the channel-wise continuity and Hamilton-Jacobi networks to learn the geodesics for all three channels simultaneously and combine three geodesics for each normalized RGB color channel to produce the output images.

Performance. Averaged MSEs of GeONet over the three RGB channels on 50 testing pairs from CIFAR-10 data are shown in Table 3. We observe that the testing errors for identity and random pairs are similar. As in the Gaussian mixture experiment, MSEs are smaller when we move closer to the geodesic boundaries at $t = 0, 1$.

Regularization. Classical algorithms such as POT take regularization parameter ν, which results in a smoothing and approximate of the solutions. This characteristic is pronounced over a coarse mesh

Table 3: MSE of GeONet on 50 test pairings of CIFAR-10 data. All values are multiplied by 10^{-3}. Errors are averaged across all three channels after normalization as densities.

	GeONet error on CIFAR-10 data				
Testing setting	$\boldsymbol{t=0}$	$\boldsymbol{t=0.25}$	$\boldsymbol{t=0.5}$	$\boldsymbol{t=0.75}$	$\boldsymbol{t=1}$
Identity pairing	2.0 ± 1.7	3.3 ± 3.5	5.8 ± 5.8	3.9 ± 4.1	3.6 ± 4.1
Random pairing	2.1 ± 1.1	4.4 ± 3.8	7.4 ± 6.8	4.7 ± 4.5	4.0 ± 6.0

with fluctuating values, such as values in CIFAR images, which induces larger geodesic inaccuracy. In contrast, GeONet takes no regularization, which is beneficial towards better accuracy.

4.3 Zero-shot super-resolution application

GeONet is mesh-invariant and suitable for zero-shot super-resolution examples of adapting low-resolution data into high-resolution geodesics, which includes initial data at $t = 0, 1$. The neural operator can be trained only on such low-resolution information, and the super-resolution application results from the uniform sampling done within the training and physics-informed loss. Traditional OT solvers have no ability to do this, as they are confined to the original mesh. Figure 5 (left) shows examples for two lower-resolution CIFAR-10 images adapted by GeONet to connect on a geodesic of higher-resolution, where each row represents the original and higher-resolution initial and terminal images.

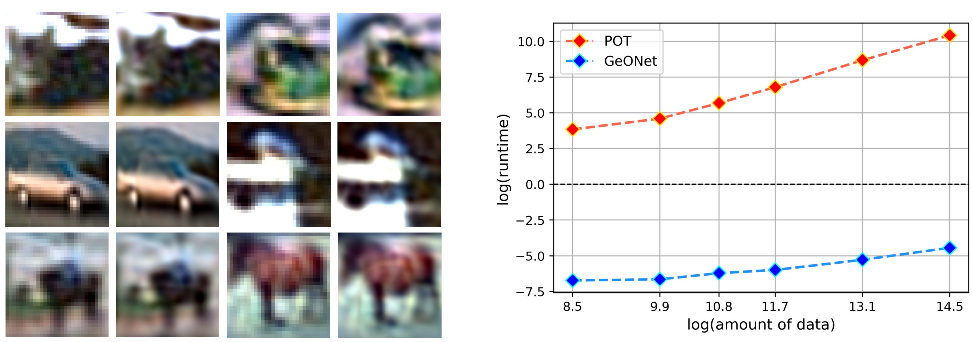

Figure 5: Left: display of zero-shot super-resolution CIFAR-10 examples. Lower- to higher-resolution geodesics on each row are produced, which includes the input images at $t = 0, 1$. Right: comparison of log-runtime in seconds on Gaussian mixtures. We average over 50 pairs with varying mesh sizes.

4.4 Runtime comparison

Inference of GeONet is instantaneous, a feature advantageous for many pairs and high resolution images. POT is greatly encumbered by a fine mesh. Figure 5 (right) demonstrates an inference-stage runtime comparison of GeONet to the POT algorithm on Gaussian mixtures considered in Section 4.1. The POT algorithm was minimally modified, only so that the pertinent distributions could be taken as input. The linear pattern of Figure 5 (right) on log-scale suggests the computational improvement of GeONet over POT is of the orders of magnitude. We remark that CIFAR-10 data in our experiment setup has the problem size corresponding to $x = 10.8$. Training time for both Gaussian mixtures and CIFAR-10 experiments was approximately 15 hours on a single NVIDIA Tesla P100 GPU. Empirically, we found training time to be faster if coefficients α_1, α_2 were scaled similarly to β_0, β_1 (i.e., more balanced physics and data losses), but some accuracy is lost. In particular, training time for the Gaussian experiment was approximately 8 hours for $\alpha_1 = \alpha_2 = \beta_0 = \beta_1$ = 1. More training details can be found in Appendix H.

REFERENCES

Jason Altschuler, Jonathan Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1961–1971, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient Flows in Metric Spaces and in the Space of Probability Measures*. Lectures in Mathematics ETH Zürich. Birkhäuser Baseluser Basel, second edition, 2008.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/arjovsky17a.html`.

Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84:375–393, 2000.

Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015. doi: 10.1137/141000439. URL `https://doi.org/10.1137/141000439`.

Dimitri P. Bertsekas and David A. Castanon. The auction algorithm for the transportation problem. *Annals of Operations Research*, 20(1):67–96, December 1989. ISSN 0254-5330. doi: 10.1007/BF02216923.

Yann Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on Pure and Applied Mathematics*, 44(4):375–417, 1991. doi: https://doi.org/10.1002/cpa.3160440402. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160440402`.

Dmitri Burage, Yuri Burago, and Sergei Ivanov. *A Course in Metric Geometry*. Graduate studies in mathematics. American mathematical society, Providence, Rhode Island, 2001.

Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 6(4):911–917, 1995. doi: 10.1109/72.392253.

Xiaohui Chen and Yun Yang. Cutoff for exact recovery of gaussian mixture models. *IEEE Transactions on Information Theory*, 67(6):4223–4238, 2021. doi: 10.1109/TIT.2021.3063155.

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(2):671–691, 2016. doi: 10.1007/s10957-015-0803-z. URL `https://doi.org/10.1007/s10957-015-0803-z`.

Yongxin Chen, Tryphon T. Georgiou, and Michele Pavon. Optimal transport in systems and control. *Annual Review of Control, Robotics, and Autonomous Systems*, 4(1):89–113, 2021. doi: 10.1146/annurev-control-070220-100858. URL `https://doi.org/10.1146/annurev-control-070220-100858`.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2017. doi: 10.1109/TPAMI.2016.2615921.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL `https://proceedings.neurips.cc/paper/2013/file/af21d0c97db2e27e13572cbf59eb343d-Paper.pdf`.

Pavel Dvurechensky, Alexander Gasnikov, and Alexey Kroshnin. Computational optimal transport: Complexity by accelerated gradient descent is better than by sinkhorn's algorithm. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1367–1376. PMLR, 10–15 Jul 2018.

Lawrence C. Evans. *Partial Differential Equations*. American Mathematical Society, Providence, R.I., 2010. ISBN 9780821849743 0821849743.

Chris Finlay, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam M Oberman. "how to train your neural ode: the world of jacobian and kinetic regularization". In *ICML*, 2020.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic optimization for large-scale optimal transport. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3440–3448, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Samuel J. Gershman and Noah D. Goodman. Amortized inference in probabilistic reasoning. *Cognitive Science*, 36, 2014.

Daisuke Inoue, Yuji Ito, and Hiroaki Yoshida. Optimal transport-based coverage control for swarm robot systems: Generalization of the voronoi tessellation-based method. In *2021 American Control Conference (ACC)*, pages 3032–3037, 2021. doi: 10.23919/ACC50511.2021.9483194.

Marcel Klatt, Carla Tameling, and Axel Munk. Empirical regularized optimal transport: Statistical theory and applications. *SIAM Journal on Mathematics of Data Science*, 2(2):419–443, 2020. doi: 10.1137/19M1278788.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces, 2021. URL `https://arxiv.org/abs/2108.08481`.

Vishaal Krishnan and Sonia Martínez. Distributed optimal transport for the deployment of swarms. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4583–4588, 2018. doi: 10.1109/CDC.2018.8619816.

H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. doi: https://doi.org/10.1002/nav.3800020109. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109`.

Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 957–966. JMLR.org, 2015.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations, 2020a. URL `https://arxiv.org/abs/2003.03485`.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations, 2020b. URL `https://arxiv.org/abs/2010.08895`.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations, 2021. URL `https://arxiv.org/abs/2111.03794`.

Shu Liu, Shaojun Ma, Yongxin Chen, Hongyuan Zha, and Haomin Zhou. Learning high dimensional wasserstein geodesics, 2021. URL https://arxiv.org/abs/2102.02992.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nat Mach Intell*, 3:218–229, 2021.

David G. Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*. Springer Publishing Company, Incorporated, 2015. ISBN 3319188410.

Robert J. McCann. A convexity principle for interacting gases. *Advances in Mathematics*, 128(1): 153–179, 1997. URL https://www.sciencedirect.com/science/article/pii/S0001870897916340.

Toshio Mikami. Monge's problem with a quadratic cost by the zero-noise limit of h-path processes. *Probability Theory and Related Fields*, 129(2):245–260, 2004. doi: 10.1007/s00440-004-0340-4. URL https://doi.org/10.1007/s00440-004-0340-4.

T. Tin Nguyen, Hien D. Nguyen, Faicel Chamroukhi, and Geoffrey J. McLachlan. Approximation by finite mixtures of continuous density functions that vanish at infinity. *Cogent Mathematics & Statistics*, 7(1):1750861, 2020. doi: 10.1080/25742558.2020.1750861. URL https://doi.org/10.1080/25742558.2020.1750861.

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

M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.10.045. URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

Filippo Santambrogio. *Optimal Transport for Applied Mathematicians. Calculus of Variations, PDEs and Modeling*. Birkhäuser Basel, 1 edition, 2015. URL https://www.math.u-psud.fr/~filippo/OTAM-cvgmt.pdf.

Assaf Shocher, Nadav Cohen Cohen, and Michal Irani. "zero-shot" super-resolution using deep internal learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Richard Sinkhorn. A Relationship Between Arbitrary Positive Matrices and Doubly Stochastic Matrices. *The Annals of Mathematical Statistics*, 35(2):876 – 879, 1964. doi: 10.1214/aoms/1177703591. URL https://doi.org/10.1214/aoms/1177703591.

Justin Solomon, Fernando de Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Trans. Graph.*, 34(4), jul 2015. ISSN 0730-0301. doi: 10.1145/2766963. URL https://doi.org/10.1145/2766963.

Lesley Tan and Liang Chen. Enhanced deeponet for modeling partial differential operators considering multiple input functions, 2022. URL https://arxiv.org/abs/2202.08942.

Cédric Villani. *Topics in optimal transportation*. Graduate studies in mathematics. American mathematical society, Providence, Rhode Island, 2003. ISBN 0-8218-3312-X.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed deeponets. *CoRR*, abs/2103.10974, 2021. URL https://arxiv.org/abs/2103.10974.

Jeremy Yu, Lu Lu, Xuhui Meng, and George Em Karniadakis. Gradient-enhanced physics-informed neural networks for forward and inverse pde problems. *Computer Methods in Applied Mechanics and Engineering*, 393:114823, 2022. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2022.114823. URL https://www.sciencedirect.com/science/article/pii/S0045782522001438.