GeONet: a neural operator for learning the Wasserstein geodesic

Andrew Gracyk[1] **Xiaohui Chen**[2]

[1]Department of Statistics, University of Illinois at Urbana-Champaign
[2]Department of Mathematics, University of Southern California

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-specific 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 simulation examples and the MNIST 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 transport map (if it exists) can be expressed in a fluid dynamics formulation with the minimum kinetic energy [Benamou and Brenier, 2000]. Such a 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, learning the continuous-time dynamical Wasserstein geodesic is a practically important task in many science and engineering domains, including developmental trajectory reconstruction in cell reprogramming [Schiebinger et al., 2019], 3D warping for shape analysis in computational geometry [Su et al., 2015], optimal control such as swarm robotics and control systems [Chen et al., 2021, Krishnan and Martínez, 2018, Inoue et al., 2021], matching supply and demand networks [Lacombe et al., 2022], computer vision such as color transfer [Bai et al., 2023], and language translation [Xu 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, machine learning methods to compute the Wasserstein geodesic for a *given* input pair of probability measures have been considered in [Liu et al., 2021, 2023, Pooladian et al., 2023, Tong et al., 2023], as well as *amortized* methods Lacombe et al. [2023], Amos et al. [2023] for generating static OT maps.

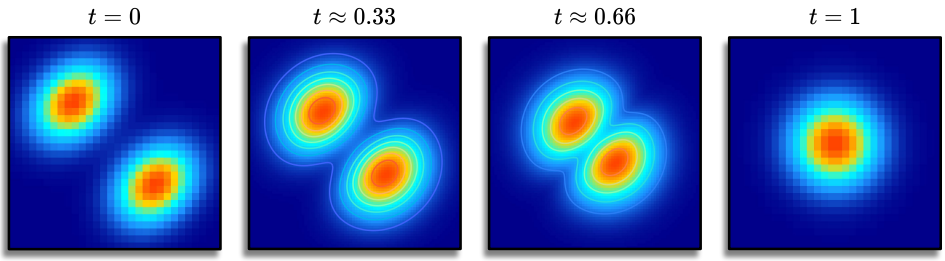

Figure 1: A geodesic at different spatial resolutions. Low-resolution inputs can be adapted into high-resolution geodesics (i.e., super-resolution) with our output mesh-invariant GeONet method.

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, 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/PINOs) [Raissi et al., 2019, 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 can learn the highly non-linear Wasserstein geodesic operator from a wide collection of training distributions. GeONet is mesh-invariant, thus it 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 a higher-resolution Wasserstein geodesic connecting two lower-resolution Gaussian mixture distributions.

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].

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. A 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.

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 $c : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$:

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

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 $c(x, y) = \|x - y\|_2^2$. The Monge problem (1) induces a metric, known as the

Table 1: We compare our method GeONet with other methodologies, including traditional neural operators, physics-based neural networks (PINNs) for learning dynamics, and traditional optimal transport solvers.

Method characteristic	Neural operator w/o physics-informed learning	PINNs	Traditional OT solvers	GeONet (Ours)
operator learning	✓			✓
satisfies the associated PDEs	✓	✓		✓
does not require known geodesic data		✓	✓	✓
output mesh independence	✓	✓		✓

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]):

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

Problems (1) and (2) are both referred to as the *static OT* problems, which have a 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:

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

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 starting 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 CE. Using integration-by-parts under suitable decay conditions for $\|x\|_2 \to \infty$, we find that the optimal dual variable u^* for the dynamic OT problem satisfies the Hamilton-Jacobi (HJ) equation

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

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 (4) are solution (μ^*,u^*) to the following system of PDEs:

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

In addition, if ψ^* and φ^* are the optimal Kantorovich potentials for solving the static dual OT problem (3), then the solution to the HJ equation (6) can be viewed as an interpolation $u(x,t)$ of the Kantorovich potentials 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). A detailed derivation of the primal-dual optimality conditions for the dynamical OT formulation is provided in Appendix B.

2.2 LEARNING NEURAL OPERATORS

Physics-informed neural networks (PINNs) [Raissi et al., 2019] aim to learn the solution of a PDE from data for a *given* input function a:

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

subject to some boundary data u_0 and u_T, where $\mathcal{D}_a$ denotes a 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 and the boundary conditions) since the solution operator $\Gamma^\dagger : \mathcal{A} \to \mathcal{U}$ is learned by obeying the induced physical laws governed by (8), and not from observations. 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 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.

A neural operator generalizes a neural network that learns the 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]. A notable example of operating learning is that $\mathcal{A}$ and $\mathcal{U}$ contain functions defined over a space-time domain $\Omega \times [0,T]$ with $\Omega \subset \mathbb{R}^d$, and the mapping of interest $\Gamma^\dagger$ is implicitly defined through a differential operator.

The idea of using neural networks to approximate any nonlinear 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 $\Gamma_\theta := \Gamma(\cdot;\theta) : \mathcal{A} \to \mathcal{U}$ for a finite-dimensional parameter $\theta \in \Theta$ to approximate the true solution operator $\Gamma^\dagger$. In this paper, we adopt the *DeepONet* architecture [Lu et al., 2021], which is suitable for their ability to learn mappings from pairings of initial input data to model $\Gamma^\dagger$. In the next subsection, we briefly discuss some basics of DeepONet architecture for modeling $\Gamma^\dagger$ and its enhanced version. Then, the neural operator learning problem is to find an optimal $\theta^* \in \Theta$ as a minimizer of the classical risk minimization problem

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

where the input data (a, u_0, u_T) are sampled from some joint distribution ν. In (9), we minimize the PDE residual loss corresponding to $\partial_t u + \mathcal{D}_a[u] = 0$ while constraining the network by imposing boundary conditions. The loss function has weights $\lambda_0, \lambda_T > 0$. Given a finite set of samples $\{(a^{(i)}, u_0^{(i)}, u_T^{(i)})\}_{i=1}^n$, and data points randomly 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 $\mathcal{D}_a$ can be conveniently exploited via automatic differentiation in standard deep learning packages.

2.3 DEEP OPERATOR NETWORKS

The DeepONet architecture [Lu et al., 2021] is based on the universal approximation theorem for operators [Chen and Chen, 1995], which says a general nonlinear continuous operator $\Gamma^\dagger$ may be approximated as follows:

$$\Gamma^\dagger(u)(x,t) \approx \sum_{k=1}^{p} \mathcal{B}_k\big(u(x_1),\dots,u(x_m);\theta\big)\cdot \mathcal{T}_k(x,t;\xi), \tag{10}$$

where $\mathcal{B}_k, \mathcal{T}_k$ are scalar elements of output of neural networks $\mathcal{B}, \mathcal{T}$, and p is the number of such elements. For instance, we may take $\mathcal{B}$ and $\mathcal{T}$ as artificial neural networks parameterized by θ, ξ respectively. Networks $\mathcal{B}, \mathcal{T}$ are referred to as the *branch* and *trunk* networks, respectively.

The unstacked DeepONet in (10) is restricted to one input function u. In our problem, since we have two initial and terminal conditions, we consider an enhanced version of DeepONet [Tan and Chen, 2022], where the operator $\Gamma^\dagger$ is approximated using two branch networks to encode for input u_0 and u_1,

$$\begin{aligned}\Gamma^\dagger(u_0,u_1)(x,t) \approx \sum_{k=1}^{p} &\mathcal{B}_k^0\big(u_0(x_1),\dots,u_0(x_m);\theta^0\big) \\ &\times\mathcal{B}_k^1\big(u_1(x_1),\dots,u_1(x_m);\theta^1\big)\times \mathcal{T}_k(x,t;\xi).\end{aligned} \tag{11}$$

In (11), the operator $\Gamma^\dagger$ is applied at the functions u_0 and u_1, and then evaluated at distinct locations $x_1,\dots,x_m$ for the branch input.

3 OUR METHOD

We present *GeONet*, a geodesic operator network for learning the 2-Wasserstein geodesic $\{\mu_t\}_{t\in[0,1]}$ connecting μ_0 to μ_1. Let $\Omega \subset \mathbb{R}^d$ be the spatial domain where the probability measures are supported. For absolutely continuous 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 $\mathrm{AC}(\mathcal{P}_2(\Omega))$. Moreover, the geodesic 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 C. In this work, our goal is to learn the non-linear operator

$$\Gamma^\dagger : \mathcal{P}_2(\Omega)\times\mathcal{P}_2(\Omega) \to \mathrm{AC}(\mathcal{P}_2(\Omega)), \tag{12}$$

$$(\mu_0,\mu_1) \mapsto \{\mu_t\}_{t\in[0,1]}, \tag{13}$$

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 (14) and three dual networks in (15) as follows:

$$\begin{aligned}\mathcal{C}(\mu_0,\mu_1)(x,t;\phi) = \sum_{k=1}^{p} &\mathcal{B}_k^{0,\mathrm{cty}}(\mu_0;\theta^{0,\mathrm{cty}}) \\ &\times\mathcal{B}_k^{1,\mathrm{cty}}(\mu_1;\theta^{1,\mathrm{cty}})\times \mathcal{T}_k^{\mathrm{cty}}(x,t;\xi^{\mathrm{cty}})\end{aligned} \tag{14}$$

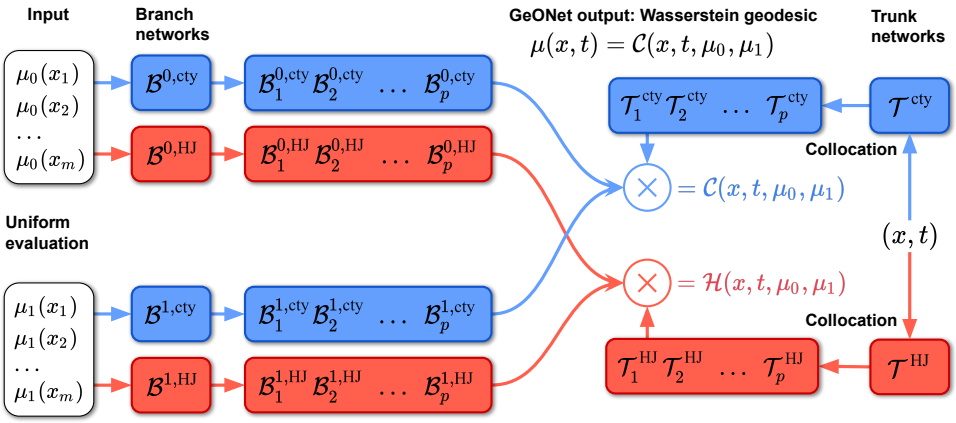

Figure 2: Architecture of GeONet. The solution to CE yields the geodesic. GeONet branches and trunks output vectors of dimension p, in which we perform multiplication among neural network elements to produce the solutions to CE and HJ.

and

$$\mathcal{H}(\mu_0,\mu_1)(x,t;\psi) = \sum_{k=1}^{p} \mathcal{B}_k^{0,\text{HJ}}(\mu_0;\theta^{0,\text{HJ}}) \times \mathcal{B}_k^{1,\text{HJ}}(\mu_1;\theta^{1,\text{HJ}}) \times \mathcal{T}_k^{\text{HJ}}(x,t;\xi^{\text{HJ}}), \tag{15}$$

where $\mathcal{B}^{j,\text{cty}}(\mu_j(x_1),\dots,\mu_j(x_m);\theta^{j,\text{cty}}) : \mathbb{R}^m \to \mathbb{R}^p$ and $\mathcal{B}^{j,\text{HJ}}(\mu_j(x_1),\dots,\mu_j(x_m);\theta^{j,\text{HJ}}) : \mathbb{R}^m \to \mathbb{R}^p$ are *branch* neural networks taking m-discretized input of initial and terminal density values at $j = 0$ and $j = 1$ respectively, and $\mathcal{T}^{\text{cty}}(x,t;\xi^{\text{cty}}) : \mathbb{R}^d \times [0,1] \to \mathbb{R}^p$ and $\mathcal{T}^{\text{HJ}}(x,t;\xi^{\text{HJ}}) : \mathbb{R}^d \times [0,1] \to \mathbb{R}^p$ are *trunk* neural networks taking spatial and temporal inputs. 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 $\mathcal{C}_\phi(x,t,\mu_0,\mu_1) := \mathcal{C}(\mu_0,\mu_1)(x,t;\phi)$ for $\phi \in \Theta \times \Theta \times \Xi$ acts as an approximate solution to the CE, hence the true geodesic $\mu_t(x) = \Gamma^\dagger(x,t,\mu_0(x),\mu_1(x))$, 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 HJ equation. The overall architecture of GeONet is shown in Figure 2.

In our GeONet implementation, we adopt a modified multilayer perceptron (MLP) architecture, which has been shown to have great ability in improving performance for physics-informed DeepONets [Wang et al., 2021b]. We shall elaborate on this architecture in Appendix F.1 and describe our empirical findings with this modified MLP for GeONet in section 4.1.

To train the GeONet defined in (14) and (15), we minimize the empirical loss function corresponding to the system of primal-dual PDEs and boundary residuals in (7) over the parameter space $\Theta \times \Theta \times \Xi$:

$$\phi^*, \psi^* \quad = \quad \text{argmin}_{\phi,\psi \in \Theta\times\Theta\times\Xi} \quad \mathcal{L}_{\text{cty}} + \mathcal{L}_{\text{HJ}} + \mathcal{L}_{\text{BC}}, \tag{16}$$

where $\mathcal{L}_{\text{cty}}$ is the loss component in which the CE is satisfied in (17) and $\mathcal{L}_{\text{HJ}}$ is the HJ loss component in (18), while boundary conditions are incorporated in the $\mathcal{L}_{\text{BC}}$ term in (19). Automatic differentiation of our GeONet involves differentiating the coupled DeepONet architecture (cf. Figure 2) to compute the physics-informed loss terms.

Our loss function involves weight parameters $\alpha_1, \alpha_2, \beta_0, \beta_1$ to impose the physics-informed loss strength. Our coefficient tuning in the loss function is motivated and follows the general strategy outlined in [Wang et al., 2021b], where coefficients are tuned by examining errors and altered in an iterative procedure in which error is minimized. Boundary conditions are enforced to a greater extent, as precision with these affects precision in the physics loss.

We now illustrate our training procedure. The physics training is done via a *collocation* procedure, following [Raissi et al., 2019]. We randomly sample N pairs (x,t) uniformly within $\Omega \times [0,1]$, where the CE and HJ expectation terms (17) and (18) in the loss function are approximated via a discrete empirical average. For the boundary terms (19), we evaluate x among fixed locations with Ω, typically a hypercube mesh, since these are where known boundary data is given, in which the neural operator is subsequently formulated and evaluated.

$$\mathcal{L}_{\text{cty}} = \alpha_1 \, \mathbb{E}_{(\mu_0,\mu_1)\sim(\mathcal{P}_2(\Omega),\mathcal{P}_2(\Omega))} \left[|| \frac{\partial}{\partial t} \mathcal{C}_\phi(x,t) + \text{div}(\mathcal{C}_\phi(x,t) \nabla \mathcal{H}_\psi(x,t)) ||^2_{L^2(\Omega\times(0,1))} \right], \tag{17}$$

$$\mathcal{L}_{\text{HJ}} = \alpha_2 \, \mathbb{E}_{(\mu_0,\mu_1)\sim(\mathcal{P}_2(\Omega),\mathcal{P}_2(\Omega))} \left[|| \frac{\partial}{\partial t} \mathcal{H}_\psi(x,t) + \frac{1}{2} ||\nabla \mathcal{H}_\psi(x,t)||_2^2 ||^2_{L^2(\Omega\times(0,1))} \right], \tag{18}$$

$$\mathcal{L}_{\text{BC}} = \beta_0 \, \mathbb{E}_{(\mu_0)\sim(\mathcal{P}_2(\Omega))} \left[||\mathcal{C}_\phi(x,0) - \mu_0||^2_{L^2(\Omega)} \right] + \beta_1 \, \mathbb{E}_{(\mu_1)\sim(\mathcal{P}_2(\Omega))} \left[||\mathcal{C}_\phi(x,1) - \mu_1||^2_{L^2(\Omega)} \right]. \tag{19}$$

Algorithm 1 End-to-end training of GeONet

Input: data pairs $(\mu_0^{(1)}\mu_1^{(1)}), \ldots, (\mu_0^{(n)}, \mu_1^{(n)})$; batch size N; initialization of the neural network parameters $\phi, \psi \in \Theta \times \Theta \times \Xi$; weight parameters $\alpha_1, \alpha_2, \beta_0, \beta_1$; domain Ω and branch domain (mesh) $\tilde{\Omega}$.; denote $i \in \{1, \ldots, N\}$.

1: **while** $\mathcal{L}_{\text{total}}$ has not converged **do**

2: Independently draw N sample points from $(x_\Omega^i, t^i) \in U(\Omega) \times U(0,1)$, N points from $x_{\tilde{\Omega}}^i \in U(\tilde{\Omega})$, and N density pairs from $\{(\mu_0^{(\ell)}, \mu_1^{(\ell)})\}_{\ell=1}^n$, possibly repeating.

3: Compute $\mathcal{R}_{\text{cty},i} = \partial_t \mathcal{C}_{\phi,i} + \text{div}(\mathcal{C}_{\phi,i} \nabla \mathcal{H}_{\psi,i})$ at (x_Ω^i, t^i). ▷ `continuity residual`

4: Compute $\mathcal{R}_{\text{HJ},i} = \partial_t \mathcal{H}_{\psi,i} + \frac{1}{2}\|\nabla \mathcal{H}_{\psi,i}\|_2^2$ at (x_Ω^i, t^i). ▷ `HJ residual`

5: Compute $B_{0,i} = \mathcal{C}_{\phi,0,i} - \mu_0^{(i)}(x_{\tilde{\Omega}}^i)$, $B_{1,i} = \mathcal{C}_{\phi,1,i} - \mu_1^{(i)}(x_{\tilde{\Omega}}^i)$. ▷ `boundary residual`

6: Compute

$$\mathcal{L}_{\text{cty}} = \frac{\alpha_1}{N} \sum_{i=1}^N \mathcal{R}_{\text{cty},i}^2, \quad \mathcal{L}_{\text{HJ}} = \frac{\alpha_2}{N} \sum_{i=1}^N \mathcal{R}_{\text{HJ},i}^2,$$

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

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 D, 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]. On one hand, adding a small entropy term (Laplacian) ensures the unique viscosity solution for the regularized HJ equation is smooth and benefits training. On the other hand, similarly as in the static OT problem, adding Laplacian approximates the OT flow (i.e., the Wasserstein geodesic is not solved exactly).

4 NUMERIC EXPERIMENTS

In this section, we perform simulation studies and a real-data example to demonstrate GeONet. Our code is publicly available at: `https://github.com/agracyk2/GeONet`.

Error metric. We use the L^1 error $\int_\Omega |\mathcal{C} - \mu| \mathrm{d}x$ as our error metric to assess the performance, where $\mu := \mu(x,t)$ is a reference geodesic as proxy of the true geodesic without entropic regularization. The L^1 error integral is estimated by evaluating a discrete Riemann sum along a mesh and the reference is computed using the POT Python library [Solomon et al., 2015, Flamary et al., 2021]. Since $\int_\Omega |\mu| \mathrm{d}x = 1$ for all time points, the L^1 error is relative, thus a meaningful metric essentially corresponding to the percentage error between the neural operator geodesic and the reference. We also consider the L^2 and Wasserstein error metric for predicted Wasserstein geodesics (see Appendix I).

4.1 INPUT AS CONTINUOUS DENSITY: GAUSSIAN MIXTURE DISTRIBUTIONS

Since finite mixture distributions are powerful universal approximators for continuous probability density functions [Nguyen et al., 2020], we first deploy GeONet on Gaussian mixture distributions over domains of varying dimensions. We learn the Wasserstein geodesic mapping between two 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 in the mixture. Here u_i and Σ_i are the mean vectors and covariance matrices of individual Gaussian components respectively. Due to the space limit, we defer simulation setups, model training details, and error metrics to Appendices G, H and I, respectively.

We examine errors in regard to an identity geodesic (i.e., $\mu_0 = \mu_1$), a random test pairing, and an out-of-distribution (OOD) pairing. The mesh-invariant nature of the output of GeONet allows zero-shot super-resolution for adapting low-resolution data into high-resolution geodesics, which includes initial data at $t = 0, 1$. Traditional OT solvers and non-operator learning based methods have no ability to do this, as they are confined to the original mesh. Thus, we also include a random test pairing on higher resolution than training data. The result is reported in Table 2.

Univariate Gaussians. We choose spatial domain $x \in \Omega = [0, 10]$ discretized into a 100-point mesh. We gen-

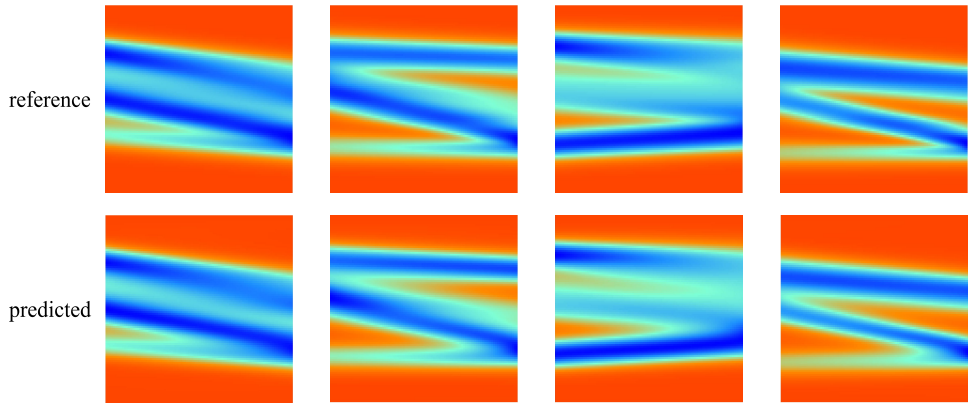

Figure 3: Four geodesics predicted by GeONet with reference geodesics computed by POT on test univariate Gaussian mixture distribution pairs with $k_0 = k_1 = 6$. The reference serves as a close approximation to the true geodesic. The vertical axis is space and the horizontal axis is time.

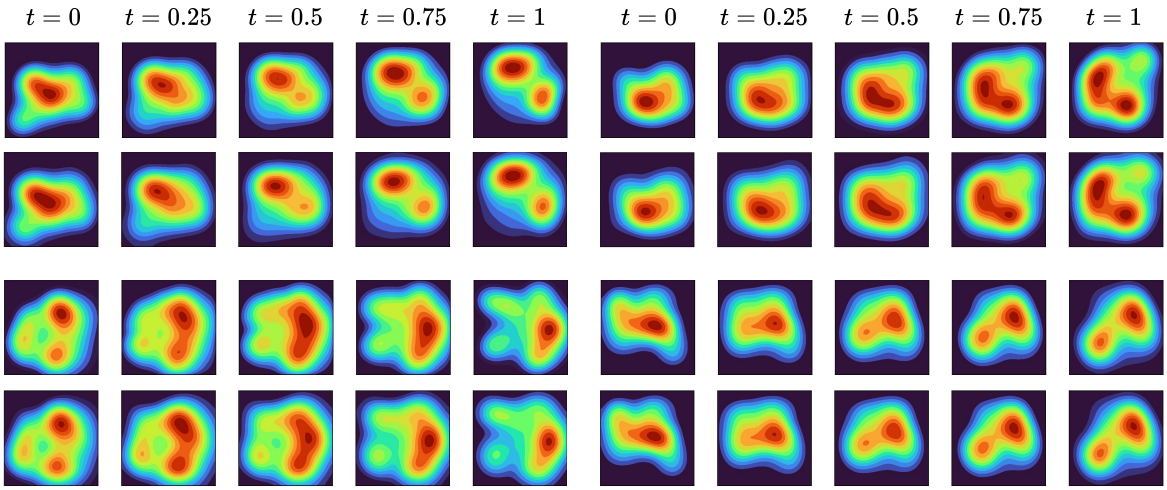

Figure 4: Geodesics predicted by GeONet on bivariate Gaussians over a square domain. The top of each pair is the reference solution computed by POT, and the bottom is GeONet.

erate $20,000$ training pairs (μ_0, μ_1) of Gaussians, taking $k_j = 6$ for the number of Gaussians in each mixture. We take means $\mu_i \in [2, 8]$ and variances $\Sigma_i \in [0.5, 0.6]$ uniformly. Empirically, we found a large batch size more suitable for training than a low one, so we take a batch size of $2,000$, meaning these many uniform collocation points are taken for both the PDE residuals and boundary points for each training iteration. We choose physical loss coefficient $\alpha_1 = 0.5, \alpha_2 = 0.25$, with boundary coefficients $\beta_0 = \beta_1 = 1$. We found these coefficients a good balance to enforce the physical constraint without sacrificing boundary restrictions after iterating these coefficients among $[0.05, 20]$ and examining the error. Additional training details are given in Appendix G.

Bivariate Gaussians. In our experiment, domain $\Omega = [0, 5] \times [0, 5] \subseteq \mathbb{R}^2$ was chosen, which was discretized into a 24×24 grid for GeONet input, meaning the branch networks took vector input of 576 in length for each in a non-convolutional architecture, but a convolutional architecture is also suitable in higher-dimensional cases as we see in Figure 9. We generate $5,000$ training pairs (μ_0, μ_1). Recall that GeONet is mesh-invariant, so the 24×24 grids can be adapted to any higher resolution, which is used in Figure 4. We use a combination of low and high variance Gaussians in the mixture, 6 of which had variance in $[0.35, 0.4]$ and 6 in $[0.75, 0.9]$, giving a total of 12 Gaussians in each mixture in each pair. Covariances were in $[-0.1, 0.1]$. Additional training details are given in Appendix G.

Training. To compute the DeepONet derivatives, we take the inner product in the enhanced DeepOnet as in equations (14), (15), and subsequently use automatic differentiation after the inner products are taken. Alternatively, we experimented by computing a Hessian for the second-order derivatives, but this is costly in terms of memory, meaning

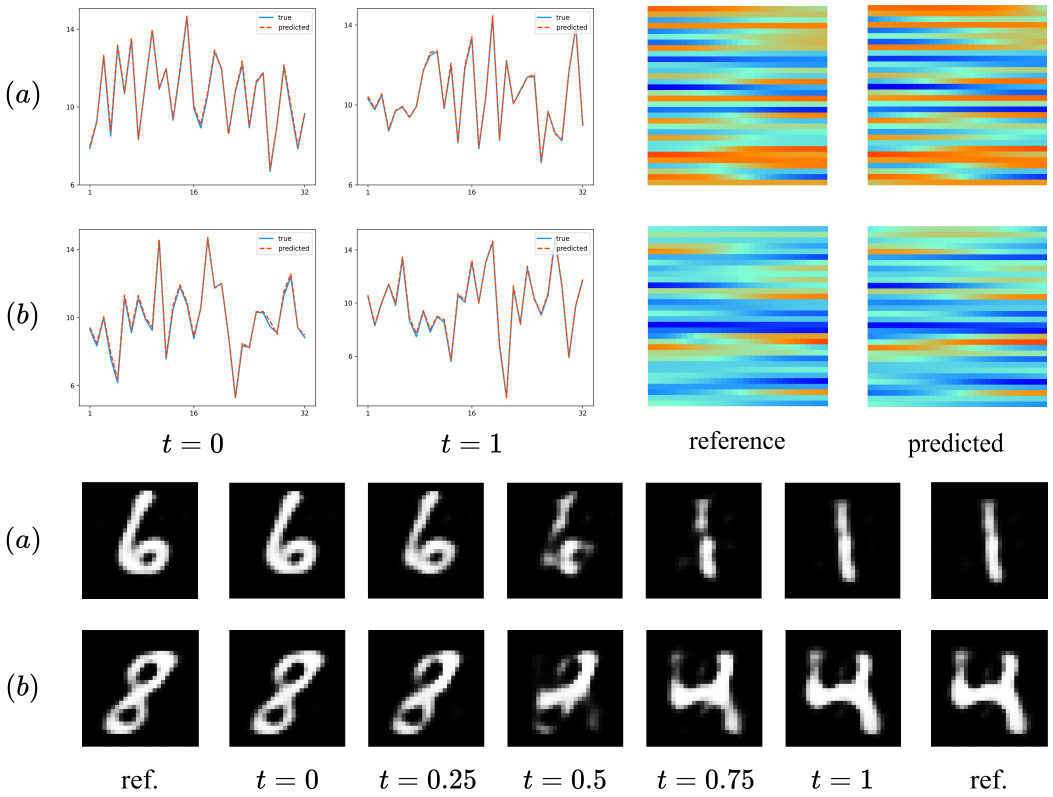

Figure 5: Beginning from the top left and going clockwise, we display the initial conditions in the encoded space, the geodesics in the encoded space, and the decoded geodesics as 28×28 images.

a large batch size cannot be used without a monumental memory cost, and so this method of differentiation is not viable.

We found that given sufficient data the GeONet with larger output dimensions slightly outperforms it with lower dimensions output. In the univariate Gaussian experiment, we take $p = 800$, which outperformed $p = 200$ by reducing training loss from approximately 2.5×10^{-4} to 1.5×10^{-4} and reducing test error by about 1%. In the bivariate experiment, changing $p = 400$ to $p = 800$ reduced training loss from approximately 2.1×10^{-5} to 1.8×10^{-5}.

Architecture generally made some difference to training loss, but not significant, making a width of around 100-200 suitable for branches and trunks. For example, increasing branch width in the univariate experiment from 100 to 150 lowered training loss by approximately 4×10^{-5}. Increasing branch width to 200 and trunk width to 150 from 150 and 100 respectively had minimal effect, lowering training loss by about 1×10^{-5}. We found the modified MLP architecture preferable, lowering final training loss from approximately 3×10^{-4} with standard architecture for univariate Gaussians.

4.2 INPUT AS POINT CLOUDS: GAUSSIAN MIXTURE DISTRIBUTIONS

GeONet can be applied to continuous densities made discrete. In scenarios with access to point clouds of data, we may use GeONet with discrete data made into empirical distributions. We test GeONet on an example of a Gaussian setup. We fix an initial and terminal distribution and sample discrete particles in $\Omega \subseteq \mathbb{R}^2$, as encompassed in [Liu et al., 2023]. The sampled particles are represented by empirical densities, in which we compare upon the transition of densities in the non-particle setting using POT as a baseline [Flamary et al., 2021]. The result is reported in Table 3 and an estimated geodesic example is shown in Figure 7. We observe that conditional flow matching (CFM) [Tong et al., 2023] and rectified flow (RF) [Liu et al., 2023] have 3-4 times comparably larger estimation errors than GeONet, except for the initial time $t = 0$, because this initial data is given and learned directly for RF and CFM. GeONet is the only framework among the comparison which captures the geodesic behavior to a considerable degree; however, we remark GeONet tends to smooth, or regularize, the solutions. Second, RF and CFM have the same fixed resolution as the input probability distribution pairing, while GeONet can estimate the density flows on higher resolution than the

Table 2: L^1 error of GeONet on 50 test data of univariate and bivariate Gaussian mixtures. We compute errors on cases of the identity geodesic, a random pairing in which $\mu_0 \neq \mu_1$, high-resolution random pairings refined to 200 and 75×75 resolutions in the 1D and 2D cases respectively, and out-of-distribution examples. We report the means and standard deviations as a percentage, making all values multiplied by 10^{-2} by those of the table.

	GeONet L^1 error for Gaussian mixtures				
Experiment	$t = 0$	$t = 0.25$	$t = 0.5$	$t = 0.75$	$t = 1$
1D identity	2.67 ± 0.750	2.85 ± 0.912	3.04 ± 1.02	2.86 ± 0.898	2.63 ± 0.696
1D random	4.92 ± 2.00	5.43 ± 3.02	5.76 ± 3.56	5.26 ± 3.25	4.65 ± 1.50
1D high-res.	4.76 ± 1.53	5.49 ± 3.00	6.01 ± 3.53	5.59 ± 2.99	4.77 ± 1.49
1D OOD	14.1 ± 4.34	18.8 ± 5.96	22.2 ± 7.32	19.2 ± 6.14	13.8 ± 4.68
2D identity	6.50 ± 1.15	7.68 ± 0.915	7.69 ± 0.924	7.70 ± 0.889	6.42 ± 1.11
2D random	6.59 ± 1.01	7.10 ± 0.869	7.13 ± 0.892	7.04 ± 0.780	6.33 ± 0.835
2D high-res.	6.66 ± 0.766	7.71 ± 1.26	7.88 ± 1.21	7.59 ± 0.979	6.29 ± 0.723
2D OOD	10.2 ± 1.18	9.82 ± 1.12	9.98 ± 1.23	9.67 ± 1.03	9.92 ± 0.944

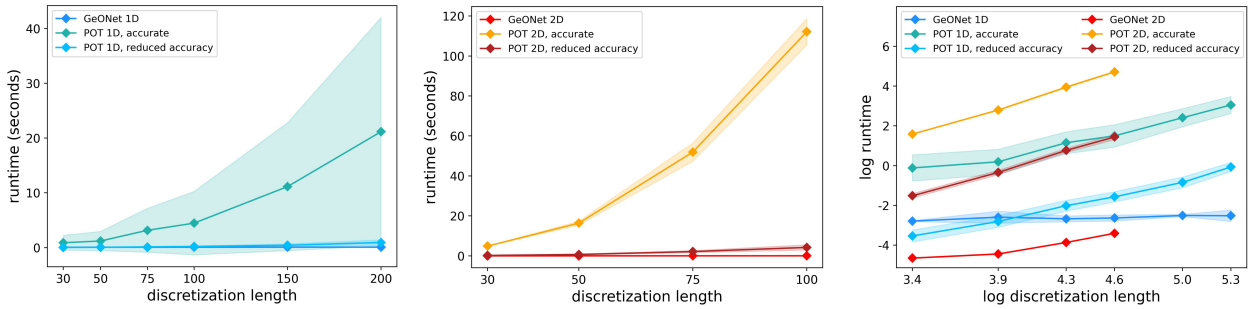

Figure 6: We compare GeONet to the classical POT library on 1D and 2D Gaussians in terms of mean and standard deviations of runtime on an unmodified scale as well as one that is log-log using discretization length in one dimension as the x-axis, taken over 30 pairs. We use 20-time steps for 1D and 5 for 2D. Finer meshes are omitted for 2D for computational reasonableness.

input pairing (cf. the third row in Figure 7).

4.3 A REAL DATA APPLICATION

Our next experiment was on the MNIST dataset of 28×28 images of single-digit numbers. It is difficult for GeONet to capture the geodesics between digits: MNIST resembles jump-discontinuous data, and is relatively piecewise constant otherwise, which is troublesome for physics-informed learning. To remedy our problems with MNIST, we use a pre-trained autoencoder to encode the MNIST digits into a low-dimensional representation $v \in \mathbb{R}^{32}$ with an encoder Φ and a decoder $\Phi^{-1} : v \to \mathbb{R}^{28} \times \mathbb{R}^{28}$ mapping the encoded representation into newly-formed digits resembling that which was fed into the encoder. The encoded data is made nonnegative via shifting upwards by a constant (we choose 10), and normalized over the domain to satisfy the density condition. This prepares the encoded data for GeONet input. We employ GeONet upon the encoded representations, learning the geodesic between highly irregular encoded data. The data can be decoded by unnormalizing and shifting downwards by the arbitrary constant. For normalization constants at $t \neq 0, 1$, we use interpolation between the constants at $t = 0, 1$.

Table 6 reports the L^1 errors for geodesic estimated in the encoded space and recovered images in the ambient space. As expected, the ambient-space error is much larger than the encoded-space error, meaning that the geodesics in the encoded space and ambient image space do not coincide. Figure 5 shows the learned geodesics in the encoded space and decoded images on the geodesics.

4.4 RUNTIME COMPARISON

Our method is highlighted by the fact that it is almost instantaneous: it is highly suitable when many geodesics are needed quickly, or over fine meshes. Traditional optimal transport solvers are greatly encumbered when evaluated over a fine grid, but the mesh-invariant output nature of GeONet bypasses this. In Figure 6, we illustrate GeONet versus POT, a traditional OT library. GeONet greatly out-

Table 3: L^1 error between GeONet, the conditional flow matching (CFM) library's optimal transport solver Tong et al. [2023], and rectified flow (RF) Liu et al. [2023], using POT again as a baseline for comparison. All values are multiplied by 10^{-2} to those of the table.

	L^1 comparison error on 2D Gaussian mixture point clouds				
Experiment	$t = 0$	$t = 0.25$	$t = 0.5$	$t = 0.75$	$t = 1$
GeONet	22.9 ± 1.08	28.8 ± 1.01	30.0 ± 1.10	29.6 ± 0.877	22.6 ± 1.02
CFM	0.0 ± 0.0	94.1 ± 3.68	98.9 ± 2.41	91.8 ± 4.15	75.9 ± 3.77
RF	0.0 ± 0.0	103 ± 2.48	112 ± 3.61	112 ± 5.03	91.3 ± 3.79

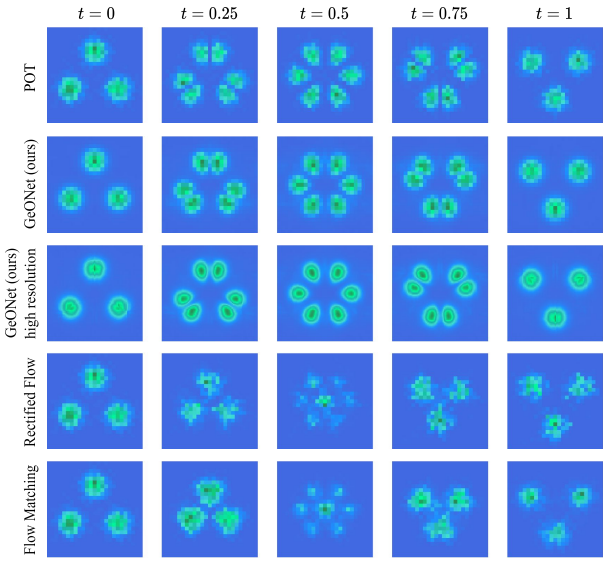

Figure 7: We compare to GeONet to the alternative methodology in a discrete setting, using POT as ground truth. GeONet is the only method among the comparison which captures the geodesic behavior among the translocation of points.

performs POT for fine grids in terms of runtime, especially if POT is used to compute an accurate solution. Even when POT is used with equivalent accuracy, GeONet still outperforms, most illustrated in the log-log plot. The log-log plot also demonstrates that our method speeds computation up to orders of magnitude. We restrict the accuracy of POT by employing a stopping threshold of 0.5 for 1D and 10.0 for 2D. We found these choices were comparable to GeONet, remarking a threshold of 10.0 in the 2D case is sufficiently large so that even larger thresholds have limited effect on error.

4.5 OUT-OF-DISTRIBUTION GENERALIZATION

We discuss GeONet on out-of-distribution data in the test setting upon Gaussian mixture data. Our error results are provided in Table 2. For univariate Gaussians, we choose means in $[1, 9]$, which was expanded from the domain $[2, 8]$ in training. This increased relative error by about 10%. Variances were in $[0.3, 0.4]$. A 100-point mesh is used for evaluations with POT regularization parameter $\epsilon = 6 \times 10^{-4}$. For 2D Gaussians, we test on 16 mixture components (training has 12). Means were in $[0.6, 4.4] \times [0.6, 4.4]$, which was expanded from $[0.8, 4.2] \times [0.8, 4.2]$ in training. There were 8 components in the mixture with variance in $[0.25, 0.3]$ and the other 8 in $[0.65, 0.8]$, which have lower variances than those in training. Covariances are within $[-0.15, 0.15]$ for off-diagonal components in each covariance matrix. Evaluations were over a 24×24 mesh, the same used as neural operator input.

4.6 LIMITATIONS

There are several limitations we would like to discuss. First, GeONet's branch network input exponentially increases in spatial dimension, necessitating extensive input data even in moderately high-dimensional scenarios. One strategy to mitigate this is through leveraging low-dimensional data representations as in the MNIST experiment. GeONet is near instantaneous for any dimension, but its dimension-based restrictions to perform are mostly hindered by the ability to handle neural network input in the branches. Second, GeONet mandates predetermined evaluation points for branch input, a requisite grounded in the pairing of initial conditions. It is of interest to extend GeONet to include training input data pairs on different resolutions. Third, given the regularity of the OT problem [Hütter and Rigollet, 2021, Caffarelli, 1996], developing a generalization error bound for assessing the predictive risk of GeONet is an important future work. Finally, the dynamical OT problem is closely connected to the mean-field planning with an extra interaction term [Fu et al., 2023]. Extending the current operator learning perspective to such problems would be interesting.

Acknowledgements

Andrew Gracyk was supported by the NSF under grant No. 1922758. Xiaohui Chen was partially supported by NSF CAREER grant DMS-2347760, NSF grant DMS-2413404, and a gift from the Simons Foundation.

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.

Brandon Amos, Samuel Cohen, Giulia Luise, and Ievgen Redko. Meta optimal transport. In *ICML*, 2023.

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.

Yikun Bai, Bernhard Schmitzer, Matthew Thorpe, and Soheil Kolouri. Sliced optimal partial transport. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13681–13690, 2023. doi: 10.1109/CVPR52729.2023.01315.

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.

Luis A. Caffarelli. Boundary regularity of maps with convex potentials–ii. *Annals of Mathematics*, 144(3):453–496, 1996. ISSN 0003486X. URL http://www.jstor.org/stable/2118564.

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.

Guosheng Fu, Siting Liu, Stanley Osher, and Wuchen Li. High order computation of optimal transport, mean field planning, and potential mean field games. *Journal of Computational Physics*, 491:112346, 2023. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2023.112346. URL https://www.sciencedirect.com/science/article/pii/S0021999123004412.

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.

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

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.

Julien Lacombe, Julie Digne, Nicolas Courty, and Nicolas Bonneel. Learning to generate wasserstein barycenters. *J. Math. Imaging Vis.*, 65(2):354–370, oct 2022. ISSN 0924-9907. doi: 10.1007/s10851-022-01121-y. URL https://doi.org/10.1007/s10851-022-01121-y.

Julien Lacombe, Julie Digne, Nicolas Courty, and Nicolas Bonneel. Learning to generate wasserstein barycenters. *Journal of Mathematical Imaging and Vision*, 65(2):354–370, 2023. doi: 10.1007/s10851-022-01121-y. URL https://doi.org/10.1007/s10851-022-01121-y.

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.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

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.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. Multisample flow matching: Straightening flows with minibatch couplings. In *ICML*, 2023.

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.

Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, Lia Lee, Jenny Chen, Justin Brumbaugh, Philippe Rigollet, Konrad Hochedlinger, Rudolf Jaenisch, Aviv Regev, and Eric S. Lander. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943, 2019. ISSN 0092-8674. doi: 10.1016/j.cell.2019.01.006.

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.

Zhengyu Su, Yalin Wang, Rui Shi, Wei Zeng, Jian Sun, Feng Luo, and Xianfeng Gu. Optimal mass transport for shape matching and comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(11):2246–2259, 2015. doi: 10.1109/TPAMI.2015.2408346.

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

Alexander Tong, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Kilian Fatras, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport, 2023.

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. On the eigenvector bias of fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384:113938, October 2021a. ISSN 0045-7825. doi: 10.1016/j.cma.2021.113938. URL http://dx.doi.org/10.1016/j.cma.2021.113938.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed deeponets. *Science Advances*, 7(40):eabi8605, 2021b. doi: 10.1126/sciadv.abi8605. URL https://www.science.org/doi/abs/10.1126/sciadv.abi8605.

Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, and Lei Li. Vocabulary learning via optimal transport for neural machine translation. In Chengqing Zong, Fei

Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7361–7373, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.571. URL https://aclanthology.org/2021.acl-long.571.

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.

GeONet: a neural operator for learning the Wasserstein geodesic (Supplementary Material)

Andrew Gracyk[1] **Xiaohui Chen**[2]

[1]Department of Statistics, University of Illinois at Urbana-Champaign
[2]Department of Mathematics, University of Southern California

A TRAINING ALGORITHM

B DERIVATION OF PRIMAL-DUAL OPTIMALITY CONDITIONS FOR DYNAMICAL OT PROBLEM

The primal-dual analysis is a standard technique in the optimization literature such as in analyzing certain semidefinite programs [Chen and Yang, 2021]. Recall the Benamou-Brenier fluid dynamics formulation of the static optimal transport problem

$$\min_{(\mu,\mathbf{v})} \int_0^1 \int_{\mathbb{R}^d} \frac{1}{2}||\mathbf{v}(x,t)||_2^2 \, \mu(x,t) \, \mathrm{d}x \, \mathrm{d}t \tag{20}$$

$$\text{subject to } \ \partial_t \mu + \text{div}(\mu \mathbf{v}) = 0, \tag{21}$$

$$\mu(\cdot,0) = \mu_0, \ \ \mu(\cdot,1) = \mu_1. \tag{22}$$

Here, equation (21) is referred to as the *CE* (CE), preserving the unit mass of the density flow $\mu_t = \mu(\cdot,t)$. We write the Lagrangian function for any flow $(\mu_t)_{t\in[0,1]}$ initializing from μ_0 and terminating at μ_1 as

$$L(\mu,\mathbf{v},u) = \int_0^1 \int_{\mathbb{R}^d} \left[\frac{1}{2}\|\mathbf{v}\|_2^2 \mu + (\partial_t \mu + \text{div}(\mu \mathbf{v}))\, u \right] \mathrm{d}x \, \mathrm{d}t, \tag{23}$$

where $u := u(x,t)$ is the dual variable for (CE). To find the optimal solution μ^* for the minimum kinetic energy (20), we study the saddle point optimization problem

$$\min_{(\mu,\mathbf{v})\in(\text{CE})} \max_u L(\mu,\mathbf{v},u), \tag{24}$$

where the minimization over $(\mu,\mathbf{v})$ runs over all flows satisfying (CE) such that $\mu(\cdot,0) = \mu_0$ and $\mu(\cdot,1) = \mu_1$. Note that if $\mu \notin$ (CE), then by scaling with arbitrarily large constant, we see that

$$\max_u \int_0^1 \int_{\mathbb{R}^d} (\partial_t \mu + \text{div}(\mu \mathbf{v}))\, u \, \mathrm{d}x \, \mathrm{d}t = +\infty. \tag{25}$$

Thus,

$$\min_{(\mu,\mathbf{v})\in(\text{CE})} \int_0^1 \int_{\mathbb{R}^d} \frac{1}{2}||\mathbf{v}||_2^2 \mu \, dx \, dt = \min_{(\mu,\mathbf{v})} \max_u L(\mu,\mathbf{v},u) \tag{26}$$

$$\geqslant \max_u \min_{(\mu,\mathbf{v})} L(\mu,\mathbf{v},u), \tag{27}$$

where the minimization over $(\mu, \mathbf{v})$ is unconstrained. Using integration-by-parts and suitable decay for vanishing boundary as $\|x\|_2 \to \infty$, we have

$$L(\mu, \mathbf{v}, u) = \int_0^1 \int_{\mathbb{R}^d} \left[\frac{1}{2}\|\mathbf{v}\|_2^2 \mu - \mu \partial_t u - \langle \mathbf{v}, \nabla u \rangle \mu \right] \, \mathrm{d}x \, \mathrm{d}t + \int_{\mathbb{R}^d} \left[\mu(\cdot, 1)u(\cdot, 1) - \mu(\cdot, 0)u(\cdot, 0) \right] \, \mathrm{d}x.$$

Now, we fix μ and u, and minimize $L(\mu, \mathbf{v}, u)$ over $\mathbf{v}$. The optimal velocity vector is $\mathbf{v}^* = \nabla u$, and we have

$$\max_u \min_\mu L(\mu, \mathbf{v}^*, u) = \int_0^1 \int_{\mathbb{R}^d} \left[-\left(\frac{1}{2}\|\nabla u\|_2^2 + \partial_t u \right) \mu \right] \, \mathrm{d}x \, \mathrm{d}t + \int_{\mathbb{R}^d} \left[u(\cdot, 1)\mu_1 - u(\cdot, 0)\mu_0 \right] \, \mathrm{d}x, \tag{28}$$

for any flow μ_t satisfying the boundary conditions $\mu(\cdot, 0) = \mu_0$ and $\mu(\cdot, 1) = \mu_1$. If $\frac{1}{2}\|\nabla u\|_2^2 + \partial_t u \neq 0$, then by the same scaling argument above, we have

$$\min_\mu \int_0^1 \int_{\mathbb{R}^d} \left[-\left(\frac{1}{2}\|\nabla u\|_2^2 + \partial_t u \right) \mu \right] \, \mathrm{d}x \, \mathrm{d}t = -\infty \tag{29}$$

because μ is unconstrained (except for the boundary conditions). Then we deduce that

$$\min_{(\mu, \mathbf{v}) \in (\mathrm{CE})} \int_0^1 \int_{\mathbb{R}^d} \frac{1}{2}||\mathbf{v}||_2^2 \mu \geqslant \max_{u \in (\mathrm{HJ})} \left\{ \int_{\mathbb{R}^d} u(\cdot, 1)\mu_1 - \int_{\mathbb{R}^d} u(\cdot, 0)\mu_0 \right\}, \tag{30}$$

where $u \in$ (HJ) means that u solves the ***HJ equation*** (HJ)

$$\partial_t u + \frac{1}{2}\|\nabla u\|_2^2 = 0. \tag{31}$$

From (30), we see that the duality gap is non-negative, and it is equal to zero if and only if (μ^*, u^*) solves the following system of PDEs

$$\begin{cases} \partial_t \mu + \mathrm{div}(\mu \nabla u) = 0, \;\; \partial_t u + \frac{1}{2}\|\nabla u\|_2^2 = 0, \\ \mu(\cdot, 0) = \mu_0, \;\; \mu(\cdot, 1) = \mu_1. \end{cases} \tag{32}$$

PDEs in (32) are referred to as the Karush–Kuhn–Tucker (KKT) conditions for the Wasserstein geodesic problem.

C METRIC GEOMETRY STRUCTURE OF THE WASSERSTEIN SPACE AND GEODESIC

In this section, we review some basic facts on the metric geometry properties of the Wasserstein space and geodesic. We first discuss the general metric space (X, d), and then specialize to the Wasserstein (metric) space $(\mathcal{P}_p(\mathbb{R}^d), W_p)$ for $p \geqslant 1$. Furthermore, we connect to the fluid dynamic formulation of optimal transport. Most of the materials are based on the reference books [Burage et al., 2001, Ambrosio et al., 2008, Santambrogio, 2015].

C.1 GENERAL METRIC SPACE

Definition C.1 (Absolutely continuous curve). Let (X, d) be a metric space. A curve $\omega : [0, 1] \to X$ is *absolutely continuous* if there is a function $g \in L^1([0, 1])$ such that for all $t_0 < t_1$, we have

$$d(\omega(t_0), \omega(t_1)) \leqslant \int_{t_0}^{t_1} g(\tau) \, \mathrm{d}\tau. \tag{33}$$

Such curves are denoted by $\mathrm{AC}(X)$.

Definition C.2 (Metric derivative). If $\omega : [0, 1] \to X$ is a curve in a metric space (X, d), the *metric derivative* of ω at time t is defined as

$$|\omega'|(t) := \lim_{h \to 0} \frac{d(\omega(t + h), \omega(t))}{|h|}, \tag{34}$$

if the limit exists.

The following theorem generalizes the classical Rademacher theorem from a Euclidean space into any metric space in terms of the metric derivative.

Theorem C.3 (Rademacher). If $\omega : [0,1] \to X$ is Lipschitz continuous, then the metric derivative $|\omega'|(t)$ exists for almost every $t \in [0,1]$. In addition, for any $0 \leqslant t < s \leqslant 1$, we have

$$d(\omega(t), \omega(s)) \leqslant \int_t^s |\omega'|(\tau)\,\mathrm{d}\tau. \tag{35}$$

Theorem C.3 tells us that absolutely continuous curve ω has a metric derivative well-defined almost everywhere, and the "length" of the curve ω is bounded by the integral of the metric derivative. Thus, a natural definition of the length of a curve in a general metric space is to take the best approximation over all possible meshes.

Definition C.4 (Curve length). For a curve $\omega : [0,1] \to X$, we define its *length* as

$$\text{Length}(\omega) := \sup\left\{\sum_{k=0}^{n-1} d(\omega(t_k), \omega(t_{k+1})) : n \geqslant 1, 0 = t_0 < t_1 < \ldots < t_n = 1\right\}. \tag{36}$$

Note that if $\omega \in \text{AC}(X)$, then

$$d(\omega(t_k), \omega(t_{k+1})) \leqslant \int_{t_k}^{t_{k+1}} g(\tau)\,\mathrm{d}\tau \tag{37}$$

so that

$$\text{Length}(\omega) \leqslant \int_0^1 g(\tau)\,\mathrm{d}\tau < \infty, \tag{38}$$

i.e., the curve ω is of bounded variation.

Lemma C.5. If $\omega \in \text{AC}(X)$, then

$$\text{Length}(\omega) = \int_0^1 |\omega'|(\tau)\,\mathrm{d}\tau. \tag{39}$$

Definition C.6 (Length space and geodesic space). Let $\omega : [0,1] \to X$ be a curve in (X,d).

1. The space (X,d) is a *length space* if
$$d(x,y) = \inf\left\{\text{Length}(\omega) : \omega(0) = x, \omega(1) = y, \omega \in \text{AC}(X)\right\}. \tag{40}$$
2. The space (X,d) is a *geodesic space* if
$$d(x,y) = \min\left\{\text{Length}(\omega) : \omega(0) = x, \omega(1) = y, \omega \in \text{AC}(X)\right\}. \tag{41}$$

Definition C.7 (Geodesic). Let (X,d) be a length space.

1. A curve $\omega : [0,1] \to X$ is said to be a *constant-speed geodesic* between $\omega(0)$ and $\omega(1)$ if
$$d(\omega(t), \omega(s)) = |t-s| \cdot d(\omega(0), \omega(1)), \tag{42}$$
for any $t, s \in [0,1]$.
2. If (X,d) is further a geodesic space, a curve $\omega : [0,1] \to X$ is said to be a *geodesic* between $x_0 \in X$ and $x_1 \in X$ if it minimizes the length among all possible curves such that $\omega(0) = x_0$ and $\omega(1) = x_1$.

Note that in a geodesic space (X,d), a constant-speed geodesic is indeed a geodesic. In addition, we have the following equivalent characterization of the geodesic in a geodesic space.

Lemma C.8. Let (X,d) be a geodesic space, $p > 1$, and $\omega : [0,1] \to X$ a curve connecting x_0 and x_1. Then the following statements are equivalent.

1. ω is a constant-speed geodesic.
2. $\omega \in \text{AC}(X)$ such that for almost every $t \in [0,1]$, we have
$$|\omega'|(t) = d(\omega(0), \omega(1)). \tag{43}$$
3. ω solves
$$\min\left\{\int_0^1 |\tilde{\omega}'|^p\,\mathrm{d}t : \tilde{\omega}(0) = x_0, \tilde{\omega}(1) = x_1\right\}. \tag{44}$$

C.2 WASSERSTEIN SPACE

Since the Wasserstein space $(\mathcal{P}_p(\mathbb{R}^d), W_p)$ for $p \geqslant 1$ is a metric space, the following definition specializes Definition C.2 to the Wasserstein metric derivative.

Definition C.9 (Wasserstein metric derivative). Let $\{\mu_t\}_{t\in[0,1]}$ be an absolutely continuous curve in the Wasserstein (metric) space $(\mathcal{P}_p(\mathbb{R}^d), W_p)$. Then the *metric derivative* at time t of the curve $t \mapsto \mu_t$ with respect to W_p is defined as

$$|\mu'|_p(t) := \lim_{h\to 0} \frac{W_p(\mu_{t+h}, \mu_t)}{|h|}. \tag{45}$$

For $p = 2$, we write $|\mu'|(t) := |\mu'|_2(t)$.

In the rest of this section, we consider probability measures μ_t that are absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^d$ and we use μ_t to denote the probability measure, as well as its density, when the context is clear.

Theorem C.10. Let $p > 1$ and assume $\Omega \in \mathbb{R}^d$ is compact.

Part 1. If $\{\mu_t\}_{t\in[0,1]}$ is an absolutely continuous curve in $W_p(\Omega)$, then for almost every $t \in [0, 1]$, there is a velocity vector field $\mathbf{v}_t \in L^p(\mu_t)$ such that

1. μ_t is a weak solution of the CE $\partial_t\mu_t + \operatorname{div}(\mu_t\mathbf{v}_t) = 0$ in the sense of distributions (cf. the definition in (51) below);
2. for almost every $t \in [0, 1]$, we have
$$\|\mathbf{v}_t\|_{L^p(\mu_t)} \leqslant |\mu'|_p(t), \tag{46}$$
where $\|\mathbf{v}_t\|^p_{L^p(\mu_t)} = \int_\Omega \|\mathbf{v}_t\|_2^p \,\mathrm{d}\mu_t$.

Part 2. Conversely, if $\{\mu_t\}_{t\in[0,1]}$ are probability measures in $\mathcal{P}_p(\Omega)$, and for each $t \in [0, 1]$ we suppose $\mathbf{v}_t \in L^p(\mu_t)$ and $\int_0^1 \|\mathbf{v}_t\|_{L^p(\mu)} \,\mathrm{d}t < \infty$ such that $(\mu_t, \mathbf{v}_t)$ solves the CE, then we have

1. $\{\mu_t\}_{t\in[0,1]}$ is an absolutely continuous curve in $(\mathcal{P}_p(\mathbb{R}^d), W_p)$;
2. for almost every $t \in [0, 1]$,
$$|\mu'|_p(t) \leqslant \|\mathbf{v}_t\|_{L^p(\mu_t)}. \tag{47}$$

As an immediate corollary, we have the following dynamical representation of the Wasserstein metric derivative.

Corollary C.11. If $\{\mu_t\}_{t\in[0,1]}$ is an absolutely continuous curve in $(\mathcal{P}_p(\mathbb{R}^d), W_p)$, then the velocity vector field $\mathbf{v}_t$ given in Part 1 of Theorem C.10 must satisfy
$$\|\mathbf{v}_t\|_{L^p(\mu_t)} = |\mu'|_p(t). \tag{48}$$

Corollary C.11 suggests that $\mathbf{v}_t$ can be viewed as the *tangent vector field* of the curve $\{\mu_t\}_{t\in[0,1]}$ at time point t. Moreover, Corollary C.11 suggests the following (Euclidean) gradient flow for tracking particles in $\mathbb{R}^d$: let $y(t) := y_x(t)$ be the trajectory starting from $x \in \mathbb{R}^d$ (i.e., $y(0) = x$) that evolves according the ordinary differential equation (ODE)

$$\frac{\mathrm{d}}{\mathrm{d}t} y(t) = \mathbf{v}_t(y(t)). \tag{49}$$

The dynamical system (49) defines a flow $Y_t : \Omega \to \Omega$ of vector field $\mathbf{v}_t$ on Ω via

$$Y_t(x) = y(t). \tag{50}$$

Then, it is straightforward to check that the pushforward measure flow $\mu_t := (Y_t)_\sharp\mu_0$ and the chosen velocity vector field $\mathbf{v}_t$ in the ODE (49) is a weak solution of the CE $\partial_t\mu_t + \operatorname{div}(\mu_t\mathbf{v}_t) = 0$ in the sense that

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_\Omega \phi \,\mathrm{d}t = \int_\Omega \langle \nabla\phi, \mathbf{v}_t\rangle \,\mathrm{d}\mu_t, \tag{51}$$

for any $\mathcal{C}^1$ function $\phi : \Omega \to \mathbb{R}$ with compact support.

Theorem C.12 (Constant-speed Wasserstein geodesic). Let $\Omega \in \mathbb{R}^d$ be a convex subset and $\mu, \nu \in \mathcal{P}_p(\Omega)$ for some $p > 1$. Let γ be an optimal transport plan under the cost function $\|x - y\|_p^p$. Define

$$\pi_t : \Omega \times \Omega \to \Omega,$$
$$\pi_t(x, y) = (1 - t)x + ty,$$

as the linear interpolation between x and y in Ω. Then, the curve $\mu_t = (\pi_t)_\sharp \gamma$ is a constant-speed geodesic in $(\mathcal{P}_p(\mathbb{R}^d), W_p)$ connecting $\mu_0 = \mu$ and $\mu_1 = \nu$.

If μ has a density with respect to the Lebesgue measure on $\mathbb{R}^d$, then there is an optimal transport map T from μ to ν [Brenier, 1991]. According to Theorem C.12, we obtain *McCann's interpolation* [McCann, 1997] in the Wasserstein space as

$$\mu_t = [(1 - t)\mathrm{id} + tT]_\sharp \mu, \tag{52}$$

which is a constant-speed geodesic in $(\mathcal{P}_p(\mathbb{R}^d), W_p)$. id is the identity function in $\mathbb{R}^d$.

To sum up, we collect the following facts about the geodesic structure and dynamical formulation of the OT problem. Let $p > 1$, and $\Omega \subset \mathbb{R}^d$ be a convex subset (either compact or have no mass escaping at infinity).

1. The metric space $(\mathcal{P}_p(\Omega), W_p)$ is a geodesic space.
2. For $\mu, \nu \in \mathcal{P}_p(\Omega)$, a constant-speed geodesic $\{\mu_t\}_{t\in[0,1]}$ in $(\mathcal{P}_p(\Omega), W_p)$ between μ and ν (i.e., $\mu_0 = \mu$ and $\mu_1 = \nu$) must satisfy $\mu_t \in \mathrm{AC}(\mathcal{P}_p(\Omega))$ and
$$|\mu'|(t) = W_p(\mu(0), \mu(1)) = W_p(\mu, \nu) \tag{53}$$
for almost every $t \in [0, 1]$.
3. The above μ_t solves
$$\min \left\{ \int_0^1 |\tilde{\mu}'|^p(t)\, \mathrm{d}t : \tilde{\mu}(0) = \mu, \tilde{\mu}(1) = \nu, \tilde{\mu} \in \mathrm{AC}(\mathcal{P}_p(\Omega)) \right\}. \tag{54}$$
4. The above μ_t solves the Benamou-Brenier problem
$$W_p^p(\mu, \nu) = \min \left\{ \int_0^1 \|\mathbf{v}_t\|_{L^p(\tilde{\mu}_t)}^p \, \mathrm{d}t : \tilde{\mu}(0) = \mu, \tilde{\mu}(1) = \nu, \partial_t \tilde{\mu}_t + \mathrm{div}(\tilde{\mu}_t \mathbf{v}_t) = 0 \right\}, \tag{55}$$
and $\mu_t = \mu(\cdot, t)$ defines a constant-speed geodesic in $(\mathcal{P}_p(\Omega), W_p)$.

D ENTROPIC REGULARIZATION

Our GeONet is compatible with entropic regularization, which is closely related to the Schrödinger bridge problem and stochastic control [Chen et al., 2016]. Specifically, the entropic-regularized GeONet (ER-GeONet) solves the following fluid dynamic problem:

$$\begin{aligned} &\min_{(\mu, \mathbf{v})} \int_0^1 \int_{\mathbb{R}^d} \frac{1}{2} ||\mathbf{v}(x, t)||_2^2 \, \mu(x, t) \, \mathrm{d}x \, \mathrm{d}t \\ &\text{subject to } \ \partial_t \mu + \mathrm{div}(\mu \mathbf{v}) + \varepsilon \Delta \mu = 0, \ \ \mu(\cdot, 0) = \mu_0, \ \ \mu(\cdot, 1) = \mu_1. \end{aligned} \tag{56}$$

Applying the same variational analysis as in the unregularized case $\varepsilon = 0$ (cf. Appendix B), we obtain the KKT conditions for the optimization (56) as the solution to the following system of PDEs:

$$\partial_t \mu + \mathrm{div}(\mu \nabla u) = -\varepsilon \Delta \mu, \tag{57}$$
$$\partial_t u + \frac{1}{2} \|\nabla u\|_2^2 = \varepsilon \Delta u, \tag{58}$$

with the boundary conditions $\mu(\cdot, 0) = \mu_0, \mu(\cdot, 1) = \mu_1$ for $\varepsilon > 0$. Note that (58) is a parabolic PDE, which has a unique smooth solution u^ε. The term $\varepsilon \Delta u$ effectively regularizes the (dual) HJ equation in (7). In the zero-noise limit as $\varepsilon \downarrow 0$, the solution of the optimal entropic interpolating flow (56) converges to solution of the Benamou-Brenier problem (4) in the sense of the method of vanishing viscosity [Mikami, 2004, Evans, 2010].

E GRADIENT ENHANCEMENT

In practice, we may 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 the efficiency as fewer data points are needed to be sampled from $U(\Omega) \times U(0,1)$, and prediction accuracy as well.

The motivation behind gradient enhancement stems from minimizing the residual of a differentiated PDE. We turn our attention to PDEs of the form

$$\begin{cases} \mathcal{F}\Big(x, t, \partial_{x_1} u, \ldots, \partial_{x_d} u, \partial_{x_1 x_1} u, \ldots, \partial_{x_d x_d} u, \ldots, \partial_t u, \lambda\Big) = 0 \quad \text{on} \quad \Omega \times [0,1], \\ u(\cdot, 0) = u_0, \quad u(\cdot, 1) = u_1 \qquad\qquad \text{on} \quad \Omega, \end{cases} \tag{59}$$

for domain $\Omega \subseteq \mathbb{R}^d$, parameter vector λ, and boundary conditions u_0, u_1. One may differentiate the PDE function $\mathcal{F}$ with respect to any spatial component to achieve

$$\frac{\partial}{\partial x_\ell} \mathcal{F}\Big(x, t, \partial_{x_1} u, \ldots, \partial_{x_d} u, \partial_{x_1 x_1} u, \ldots, \partial_{x_d x_d} u, \ldots, \partial_t u, \lambda\Big) = 0. \tag{60}$$

The differentiated PDE is additionally equal to 0, similar to what we see in a PINN setup. If we substitute a neural network into the differentiated PDE of (60), what remains is a new residual, just as we saw with the neural network substituted into the original PDE. Minimizing this new residual in the related loss function characterizes the gradient enhancement method.

We utilize the same loss function in (16), but we add the additional terms

$$\mathcal{L}_{\text{GE,cty}} = \sum_{\ell=1}^{d} \gamma_\ell \, \mathbb{E}[|| \ \frac{\partial}{\partial x_\ell}(\frac{\partial}{\partial t}\mathcal{C}_\phi + \text{div}(\mathcal{C}_\phi \nabla \mathcal{H}_\psi)) \ ||^2_{L^2(\Omega \times (0,1))}], \tag{61}$$

$$\mathcal{L}_{\text{GE, HJ}} = \sum_{\ell=1}^{d} \omega_\ell \, \mathbb{E}[|| \ \frac{\partial}{\partial x_\ell}(\frac{\partial}{\partial t}\mathcal{H}_\psi + \frac{1}{2}||\nabla \mathcal{H}_\psi||_2^2) \ ||^2_{L^2(\Omega \times (0,1))}], \tag{62}$$

where the variables and neural networks that also appeared in (16) are the same. Here γ_ℓ and ω_ℓ are positive weights. The summation is taken in order to account for the gradient enhancement of each spatial component of $x \in \Omega$.

F SPECIALIZED ARCHITECTURES

F.1 MODIFIED MULTI-LAYER PERCEPTRON

Here we outline the forward pass of the modified multi-layer perceptron used throughout the experiments as presented in Wang et al. [2021b] Let σ denote an activation function (at least twice differentiable to allow automatic differentiation of the networks to satisfy the PDEs), X as neural network design input, W^i the weights of the neural network at index i, and b^i the bias at layer i. Here, X can refer to either branch or trunk inputs, as this architecture is used for both.

The forward pass is given by

$$U = \sigma(W^1 X + b^1), \quad V = \sigma(W^2 X + b^2) \tag{63}$$

$$H^1 = \sigma(W^{h,1} X + b^{h,1}) \tag{64}$$

$$Z^k = \sigma(W^{z,k} H^k + b^{z,k}) \tag{65}$$

$$H^k = (1 - Z^{k-1}) \odot U + Z^{k-1} \odot U \tag{66}$$

$$\mathcal{N}_\theta = W^\ell H^\ell + b^\ell, \tag{67}$$

where $k \in \{1, \ldots, \ell\}$, $\odot$ is an element-wise product, and $\mathcal{N}_\theta$ is the neural network final output, either a branch or a trunk.

F.2 FOURIER FEATURE ARCHITECTURE

We outline the Fourier feature architecture of Wang et al. [2021b]. We embed trunk input $y = (x, t)$ in a higher-dimensional space by taking transformations of the form

$$U = (\cos(2\pi B_x y), \sin(2\pi B_x y))^T \tag{68}$$

and passing them into trunk input. Alternatively, we consider the more elaborated architecture of Wang et al. [2021a], which requires passing in x, t into distinct Fourier embeddings of the form of U, and using separate layers for each. An element-wise product is taken before the last layer. We used this for our experiments of 4.2, but generally found the Fourier feature architecture of passing in $y = (x, t)$ to formulate U as effective as well.

G HYPERPARAMETER SETTINGS AND TRAINING DETAILS

We discuss training characteristics of GeONet based on the primary experiments. An unmodified Adam optimizer was chosen for all branch, trunk neural networks with a learning rate starting from $5\text{e}{-4}$. All layers share the same width. We use $\tanh$ activation for all neural networks. Coefficients $\alpha_1, \alpha_2, \beta_0, \beta_1$ were computed after examining errors. Coefficients were selected in the range $[0.05, 20]$. Neural network depths refer to ℓ in each modified MLP. Training is done on a NVIDIA T4 GPU.

Table 4: Architecture and training details in our Gaussian mixture experiments of Section 4 and Appendix H.

Hyperparameter	1D Gaussians	2D Gaussians
No. of initial conditions (μ_0, μ_1)	20,000	5,000
m (branch input dimension)	100	576
Branch width	150	200
Branch depth	7	7
Trunk width	100	150
Trunk depth	7	7
p (dimension of outputs)	800	800
Batch size	2,000	2,000
Final training time	$\sim$ 2 hrs	$\sim$ 2 hrs
Final training loss	$\sim 1.5\text{e}{-4}$	$\sim 1.8\text{e}{-5}$
$\alpha_1, \alpha_2, \beta_0, \beta_1$	$0.5, 0.25, 1, 1$	$0.5, 0.25, 1, 1$

Table 5: Architecture and training details in our empirical Gaussians and encoded MNIST experiments of Section 4 and Appendix H.

Hyperparameter	Empirical Gaussians	Encoded MNIST
No. of initial conditions (μ_0, μ_1)	1,000	30,000
m (branch input dimension)	625	32
Branch width	100	150
Branch depth	7	7
Trunk width	100	100
Trunk depth	5	7
p (dimension of outputs)	200	200
Batch size	1,000	1,000
Final training time	$\sim$ 2 hrs	$\sim$ 4 hrs
Final training loss	$\sim 7.0\text{e}{-4}$	$\sim 2.0\text{e}{-2}$
$\alpha_1, \alpha_2, \beta_0, \beta_1$	$0.5, 0.25, 1, 1$	$1, 1, 1, 1$

H TRAINING AND PERFORMANCE

H.1 UNIVARIATE AND BIVARIATE GAUSSIAN MIXTURE EXPERIMENTS

Performance. Our baseline results were collected by deploying GeONet on the identity geodesic in Table 2. The baseline identity geodesic provides a benchmark for comparing and interpreting the errors across different setups. The univariate cases were evaluated upon a 100 point mesh, and the bivariate upon a 40×40 mesh, except in the zero-shot super-resolution case, in which the grid is refined and previously specified. From Table 2, we can draw the following observations. The loss boundary conditions (19) allow greater precision for $t = 0, 1$, which suggests that a lack of data-enforced conditions along the inner region of the time continuum would cause greater error. Errors for predicting the univariate Gaussian 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. In the bivariate experiment, we found that error quickly rises as variance decreases, which is equivalent to a task of learning more complicated geodesics. We did not find lower variance drastically affects performance in the univariate experiment, suggesting GeONet and potentially physics-informed DeepONets in general are less effective as the dimension increases. We did not find the number of Gaussians in the mixtures drastically affected results, but naturally more complicated geodesics induce greater error, which is to be expected. We found bivariate errors are similar to the random case as in the identity case, suggesting there is some notion of base neural operator error, which may not exist with simpler data.

H.2 GAUSSIAN EMPIRICAL DENSITIES

Training. 3000 point cloud particles were sampled from mixtures composed of 3 Gaussians for source μ_0 and target μ_1. 2D histograms were constructed to turn particle data into empirical densities, with bins ranging from -7 to 7. Domain $\Omega = [0, 5] \times [0, 5]$ was discretized into a 25×25 point domain and assigned for the histograms' locations used as GeONet spatial input. A batch size of 1,000 was chosen. We take $p = 200$, $\alpha_1 = 0.5, \alpha_2 = 0.25, \beta_0 = \beta_1 = 1$, which can be altered to impose strength of the boundary and physics terms accordingly. We employ the Fourier feature network architecture of Wang et al. [2021a] for trunk networks. We take matrix B_v with elements sampled in $\mathcal{N}(0, \sigma_v^2)$, subsequently taking $(\cos(2\pi B_v v), \sin(2\pi B_v v))^T$ as input for a fully-connected network, where v is either space or time input. Our architecture for this experiment is fully outlined in F.2. Empirically, we found low variance necessary, and we chose $\sigma = 0.5$ for both $v = x, t$ for both continuity and trunk branches.

Performance. In this experiment, GeONet correctly captures the translocation of mass and overall geodesic behavior. The other methods are more suited for point clouds but yield high errors in learning the geodesic. GeONet tends to slightly regularize the solution by smoothing them, in which GeONet has trouble learning precision that comes with particle samples.

H.3 MNIST EXPERIMENT

Training. As described in section 4, to learn the geodesic, we ensure all values within the encoded representation are nonnegative, meaning we can shift all encoded representations by some arbitrary constant. We choose 10 for this. This constant can be deducted in later stages to ensure the valid representation is met. We normalize the data so that the density conditions are satisfied before GeONet input. A domain of $[0, 5]$ was divided into an equispaced mesh of 32 points for the encoded representation. This domain is rather arbitrary and is chosen simply for DeepONet input purposes, which can be modified as seen fit. $30,000$ encoded pairs were chosen to train GeONet and the pre-trained autoencoder, the entirety of MNIST. We used a batch size of $1,000$. Additional details are found in Appendix G.

Table 6: L^1 error of GeONet on 50 test pairings of encoded MNIST. All values are multiplied by 10^{-2}. Error was calculated upon the geodesic in both the shifted and ambient/original space.

	GeONet L^1 error on encoded MNIST data				
Test setting	$\boldsymbol{t = 0}$	$\boldsymbol{t = 0.25}$	$\boldsymbol{t = 0.5}$	$\boldsymbol{t = 0.75}$	$\boldsymbol{t = 1}$
Encoded, identity	0.923 ± 0.213	0.830 ± 0.166	0.825 ± 0.165	0.834 ± 0.173	0.931 ± 0.215
Encoded, random	1.62 ± 0.333	2.14 ± 1.22	2.78 ± 1.62	2.11 ± 1.17	1.54 ± 0.282
Ambient, identity	26.7 ± 11.2	34.0 ± 6.88	35.3 ± 8.32	36.4 ± 9.77	34.0 ± 13.2
Ambient, random	32.1 ± 16.6	58.2 ± 15.0	68.1 ± 18.8	56.4 ± 14.3	24.7 ± 10.7

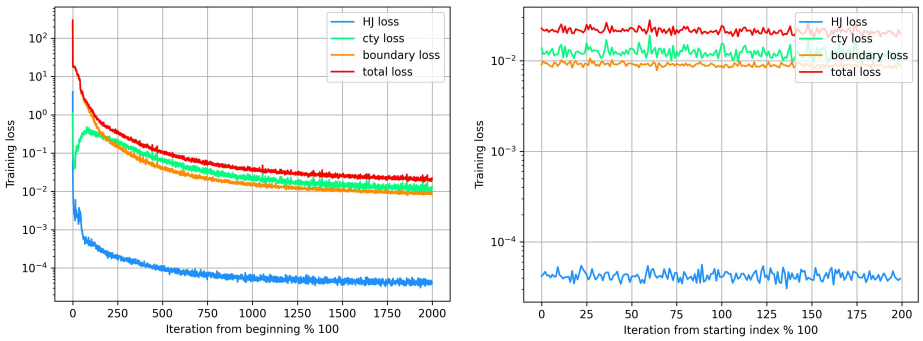

Figure 8: We examine iterations of the Adam optimizer in the total and late training on a log scale. We examine late training in order to observe oscillatory behavior between the continuity and HJ loss to see if they adversarially compete in late training. We do not observe this pattern, and the continuity loss greatly surpasses the HJ loss in value. These graphs were created using the encoded MNIST experiment.

Performance. GeONet performs well in this experiment. Scaling the physics-informed term by a constant of less than one did not prove necessary in this experiment to ensure all loss terms are met to a sufficient degree. As before, boundary terms are uniformly smaller, likely since these terms are known and included in the loss function to be minimized. The same error metric is used as in the synthetic experiments but with normalization, making the L^1 error relative. We remark OOD generalization is omitted because the distribution of the encoded data is not known. We also remark the decoded images, being the geodesic returned to its original state, do not directly translate to a geodesic performed upon an original pair of images. NaN values are omitted in the error computations, which are possible in the POT solutions due to the irregularity of the initial conditions.

Regularization. Classical geodesic algorithms require a small regularization parameter in order to be computed. This affects the synthetic experiments trivially, but we found this regularization induces greater in the MNIST experiment. This is to be considered when evaluating the errors, and true error is likely smaller between GeONet and the reference geodesics computed with POT than what is displayed. This regularization acts as a form of "smoothing" of the solutions.

I GEONET ERROR FOR ADDITIONAL ERROR METRICS

Table 7: We list mean and standard deviations of error of GeONet on 50 random $\mu_0 \neq \mu_1$ samples for alternative error metrics, being L^2 error and the Wasserstein-1 distance. We remark we use sliced Wasserstein distance for the 2D case, as this metric is computationally feasible for higher dimensional cases. We perform this for random Gaussian mixture pairings. All values are multiplied by 10^{-2} by those of the table.

	GeONet alternative metric error for random Gaussian mixtures		
Experiment	$t = 0$	$t = 0.25$	$t = 0.5$
1D, L^2	5.19 ± 1.74	6.91 ± 4.81	7.28 ± 5.39
1D, Wasserstein	0.352 ± 0.116	0.364 ± 0.178	0.403 ± 0.228
2D, L^2	6.93 ± 0.883	7.72 ± 1.23	8.11 ± 1.30
2D, Wasserstein	0.245 ± 0.0329	0.264 ± 0.0316	0.275 ± 0.0447

Experiment	$t = 0.75$	$t = 1$
1D, L^2	6.49 ± 4.36	4.81 ± 1.58
1D, Wasserstein	0.386 ± 0.166	0.347 ± 0.101
2D, L^2	7.79 ± 1.14	6.87 ± 1.05
2D, Wasserstein	0.267 ± 0.0338	0.246 ± 0.0356

J 3D GAUSSIANS FIGURE

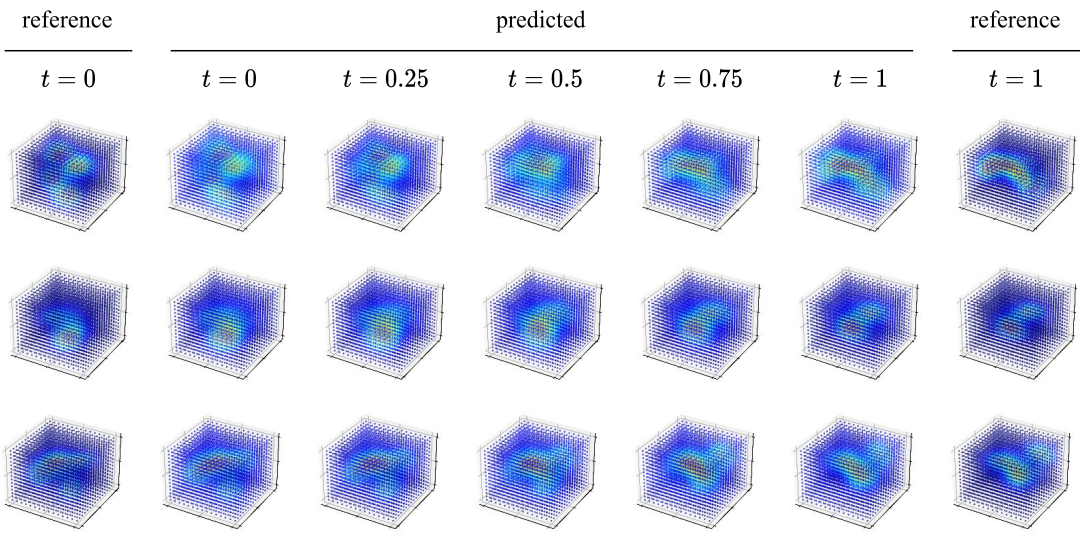

Figure 9: We illustrate GeONet on 3D Gaussians.

K SAMPLE HJ GRAPHS

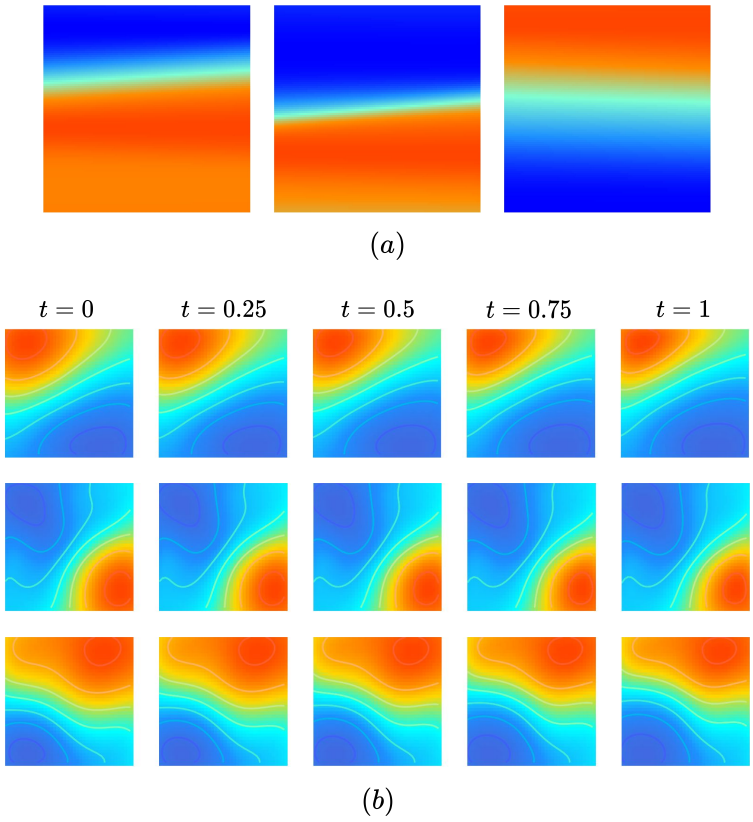

Figure 10: We present sample HJ equations for (a) three univariate Gaussian mixtures and (b) three bivariate Gaussian mixtures from the primary experiments performed in Section 4. The univariate HJ samples at certain times are the vertical cross-sections of the graphs, and the bivariate samples are given at certain times.