




# A Machine Learning Framework for Geodesics Under Spherical Wasserstein–Fisher–Rao Metric and Its Application for Weighted Sample Generation

Yang Jing<sup>1</sup> · Jiaheng Chen<sup>1,2</sup> · Lei Li<sup>1,3,4</sup>  · Jianfeng Lu<sup>5</sup>

Received: 10 September 2022 / Revised: 9 July 2023 / Accepted: 24 October 2023 /  
Published online: 22 November 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

Wasserstein–Fisher–Rao (WFR) distance is a family of metrics to gauge the discrepancy of two Radon measures, which takes into account both transportation and weight change. Spherical WFR distance is a projected version of WFR distance for probability measures so that the space of Radon measures equipped with WFR can be viewed as metric cone over the space of probability measures with spherical WFR. Compared to the case for Wasserstein distance, the understanding of geodesics under the spherical WFR is less clear and still an ongoing research focus. In this paper, we develop a deep learning framework to compute the geodesics under the spherical WFR metric, and the learned geodesics can be adopted to generate weighted samples. Our approach is based on a Benamou–Brenier type dynamic formulation for spherical WFR. To overcome the difficulty in enforcing the boundary constraint brought by the weight change, a Kullback–Leibler divergence term based on the inverse map is introduced into the cost function. Moreover, a new regularization term using the particle velocity is introduced as a substitute for the Hamilton–Jacobi equation for the potential in dynamic formula. When used for sample generation, our framework can be beneficial for

---

✉ Lei Li  
leili2010@sjtu.edu.cn

Yang Jing  
sharkjingyang@sjtu.edu.cn

Jiaheng Chen  
jiaheng@uchicago.edu

Jianfeng Lu  
jianfeng@math.duke.edu

- <sup>1</sup> School of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai 200240, People's Republic of China
- <sup>2</sup> Committee on Computational and Applied Mathematics, University of Chicago, Chicago, IL 60637, USA
- <sup>3</sup> Institute of Natural Sciences, Qing Yuan Research Institute, MOE-LSC, Shanghai Jiao Tong University, Shanghai 200240, People's Republic of China
- <sup>4</sup> Shanghai Artificial Intelligence Laboratory, Shanghai, People's Republic of China
- <sup>5</sup> Mathematics Department, Duke University, Box 90320, Durham, NC 27708, USA

applications with given weighted samples, especially in the Bayesian inference, compared to sample generation with previous flow models.

**Keywords** Unbalanced optimal transport · Wasserstein distances · Weighted samples · Normalizing flows · Kullback–Leibler divergence · Bayesian inference

**Mathematics Subject Classification** 49Q22 · 68T07

## 1 Introduction

A proper metric of probability measures can lead to new methods and algorithms in data science, like the SVGD [34, 36] and WGAN [3, 19] methods in unsupervised learning. The family of Wasserstein distances is among the popular metrics since they can evaluate dissimilarity between two distributions, even when one or both of them are discrete data samples with disjoint supports. Wasserstein distance have quantity of state-of-art properties, which can be applied to traditional models for improvements [40, 48]. Wasserstein–Fisher–Rao (WFR) distance (a.k.a, Hellinger–Kantorovich distance) is a generalized version of Wasserstein distance [7, 8, 33], which interpolates between the quadratic Wasserstein and the Fisher–Rao metrics and generalizes optimal transport to measures with different masses. From dynamic view, WFR adds a source term in its Benamou–Brenier formula, which will lead to a weighted particle transport process compared with Wasserstein distance. The spherical WFR metric is a projected version of WFR metric for probability measures and the corresponding particle motions also have weight change besides transportation [28, 29].

Making use of the metric, one may transform a probability measure continuously into another one, by, for example, the gradient flows with certain carefully designed functional and the geodesics [1]. For example, the SVGD method can be viewed as the gradient flow of the relative entropy under the Stein metric. The geodesics, on the other hand, is the curve that takes the least cost under the corresponding metric to transform a probability measure to another one. The structure of geodesics under Wasserstein distance has been well studied [49, 55]. For example, in the case of Wasserstein-2 distance, one may solve the Brenier potential through the Monge–Ampere equation so that the transport plan can be computed explicitly. As soon as one knows the transport plan, the geodesics has a clear description (see also Proposition 3.1 below), and the particles move with constant velocities and non-intersecting particle trajectories [49]. As a generalization of Wasserstein distance, WFR has also been proved to be a good metric in natural language processing to measure the distance between different documents, in waveform based earthquake location models to measure the discrepancy between true and synthetic earthquake signals, and in cellular population models to help to estimate cellular growth and death rates [50, 56, 60], etc. The structure of geodesics under WFR is however less understood compared to the Wasserstein distance. For example, how the transportation and weight change balance and how the behaviors of the particles along the geodesics are unclear. We are interested in the geodesics under spherical WFR metric and seek for solutions in particle sense.

An important application of transforming the probability measures is sample generation, which is an important research direction in data science [16, 25, 26, 44]. Compared with generative adversarial networks (GANs) [3, 16] and variational autoencoders (VAE) [25, 26], the flow-based models directly focus on original data space and tries to evolve the density distribution by transportation fields [44, 51, 54]. In the continuous normalizing flow

(CNF) [6], the probability distribution is evolved according to the mass transportation under a velocity field. An advantage of CNF is that the inverse mapping is easy to obtain through solving related ODE backward. However, the velocity field that transforms a given probability measure to a target one is not unique in general. Finlay et al. [13] first introduced optimal transportation regularization in normalizing flows and Onken et. al. [40] proposed OT-Flow, an improved version of the CNF. The OT-Flow may be preferred in applications since the particles follow straight lines and the trajectories will not cross each other, and thus such a model is expected to improve its invertibility and reduce the computational cost.

In this paper, we are interested in developing a deep learning framework to compute the geodesics under the spherical WFR metric. This framework can not only tell us how the particles and distribution evolve in the “optimal way” but also can be used as new generative models for *weighted* samples. Following the framework for the OT-Flow [40], we will use dynamic formulation of spherical WFR, where a source term is added to the transport equation. Inside this process, the particles not only are transported but also will carry an evolving weight. Using this framework, we are not only able to compute the geodesics  $\rho_t$  but also compute the velocity field and the source term that realize this measure evolution, thus the geodesics.

If we set the start distribution as the standard normal distribution and the desired distribution as the target distribution, the learned velocity field and source for the geodesics can lead to particle motions and weight change for the desired distribution, thus a generative model for the target distribution. Our framework is clearly a generalization of the CNF and OT-Flow models in the sense that it considers weight change. This framework might be promising in dealing with weighted data, which can be costly for CNF models as one needs to resample. For example, the attention mechanism provides words with different weights [11, 53]. Another particularly suited example is the Bayesian inference [2, 57], in which drawing samples from posterior is an essential task in estimation. Since directly calculating posterior with Bayesian formula to sample can be costly, we can rely on our new model to learn posterior from given weighted data and generate new samples for Bayesian inference. If one expects the transportation effect not to be significant in applications, such a framework may also be beneficial. For instance in pharmaceutical people use generative model to create and design new drug molecule and one may desire the new drug molecule to keep most already known structures.

The rest of the paper is organized as follows. Section 2 is devoted to a brief review of unbalanced optimal transport and WFR metrics that are useful to us later. We also introduce two flow-based generative models: CNF and OT-Flow, as a starting point of our framework. Then, we review some knowledge about geodesics and derive basic equations for geodesics under spherical WFR metric in Sect. 3, which guides us to construct particle algorithms. In Sect. 4, we develop a deep learning framework to compute the geodesics under spherical WFR metric. We illustrate how to use the learned geodesics to generate weighted samples efficiently in Sect. 5, especially in the Bayesian framework. In Sect. 6, we provide some numerical experiments to validate the framework. We conclude the work and make a discussion in Sect. 7.

## 2 Preliminaries

In this section, we first collect some basics of unbalanced optimal transport including the dynamic and static formulations of Wasserstein–Fisher–Rao metric, as well as spherical WFR

we will use in the rest of paper. We then give a brief introduction to continuous normalizing flows and OT-Flow model in Sect. 2.2, which are generative models based on transportation purely.

## 2.1 The Unbalanced Optimal Transport and the WFR Metrics

Optimal Transport (OT) has a long history since Monge first posed the problem in 1781 [38], which sits at the intersection of various mathematical fields including probability, geometry, PDEs and optimization [49, 55]. In recent years, optimal transport has seen an increasing amount of attention from computer science [3, 43, 46], biological sciences [50, 59], economics [14, 15], etc. The optimal transport problem (or Kantorovich problem) is, given two distribution  $\mu$  and  $\nu$  and a cost function  $c : X \times Y \rightarrow [0, \infty]$ , one is supposed to solve

$$\min \left\{ \int_{X \times Y} c d\gamma \mid \gamma \in \Pi(\mu, \nu) \right\},$$

where  $\Pi(\mu, \nu)$  is the set of *transport plans*, i.e. a joint measure on  $X \times Y$ , with marginal distribution  $\mu$  and  $\nu$ . In the case  $X = Y = \Omega \subset \mathbb{R}^d$ , the optimal value of Kantorovich problem with  $c(x, y) = |x - y|^p$  is used to define Wasserstein- $p$  distance ( $p \geq 1$ ) between  $\mu$  and  $\nu$ :

$$W_p(\mu, \nu) = \left( \inf_{\gamma \in \Pi(\mu, \nu)} \int |x - y|^p d\gamma \right)^{1/p}.$$

Denote  $\mathcal{P}(\Omega)$  be the set of all probability measures on  $\Omega$ . We can define the space  $\mathcal{W}_p := \{\mu \in \mathcal{P}(\Omega) \mid \int |x|^p \mu(dx) < \infty\}$ . Then  $(\mathcal{W}_p, W_p)$  is a complete metric space. Furthermore,  $W_p$  distance admits the following dynamic Benamou–Brenier formulation [49, Chap. 5 Theorem 5.28]: on a convex and compact domain  $\Omega$ ,  $\mu$  and  $\nu$  are two probability distribution on  $\Omega$  and are absolutely continuous with respect to the Lebesgue measure,  $v_t$  is a vector field on  $\Omega$ ,

$$W_p^p(\mu, \nu) = \min_{\rho, v} \left\{ \int_0^1 \|v_t\|_{L^p(\rho)}^p dt : \partial_t \rho_t + \nabla \cdot (\rho_t v_t) = 0, \rho_0 = \mu, \rho_1 = \nu \right\}, \quad (2.1)$$

where  $\|v_t\|_{L^p(\rho_t)}^p = \int_{\Omega} |v_t(x)|^p \rho_t(dx)$ . We will focus on  $p = 2$  in the rest of our article. The classical optimal transport theory reveals two useful facts. The first is that the optimal particle velocity under  $W_2$  is a constant along one trajectory, which implies the trajectory is a straight line. The second property is that the trajectories will not cross each other in space-time plane. These properties are beneficial for us to build an inverse mapping from  $\nu$  to  $\mu$ . A brief discussion about geodesics under Wasserstein distance will be performed in Sect. 3.

A notable restriction of optimal transport is that it is only defined between measures having the same mass, which might not be suitable for applications in image classification where measures need not be normalized [42, 45], and biophysical phenomena involving some sort of mass creation or destruction [50]. Recently, Wasserstein–Fisher–Rao (WFR) distance is proposed to generalize optimal transport to measures with different masses, which interpolates between the quadratic Wasserstein and the Fisher-Rao metrics [7, 8, 27]. We first introduce the dynamic formulation of unbalanced optimal transport designed for the WFR distance, which leads to our proposed numerical method in Sect. 4. By introducing a source term in the continuity equation, WFR distance relaxes the equality of mass constraint in the

dynamic Benamou–Brenier formulation of optimal transport

$$d_{\text{WFR},\alpha}^2(\mu, \nu) = \inf_{\rho, v, g} \left\{ \int_0^1 \int_{\Omega} \left( \frac{1}{2} |v_t(x)|^2 + \frac{\alpha}{2} g_t^2(x) \right) \rho_t(dx) dt : \right. \\ \left. \partial_t \rho_t + \nabla \cdot (\rho_t v_t) = \rho_t g_t, \rho_0 = \mu, \rho_1 = \nu \right\}, \tag{2.2}$$

where  $(\rho_t)_{t \in [0,1]}$  is a time-dependent density that interpolates between  $\rho_0$  and  $\rho_1$ ,  $(v_t)_{t \in [0,1]}$  is a velocity field that describes the movement of mass and  $(g_t)_{t \in [0,1]}$  is a scalar field (source term) associated with mass creation and destruction.  $\alpha$  is a hyper-parameter to balance the effects of transport and mass creation/destruction explicitly.

WFR also admits a static Kantorovich formulation analogously to standard optimal transport [8]. Let  $\mathcal{M}_+(X)$  be the space of nonnegative Radon measures on a compact set  $X \subset \mathbb{R}^d$ . For a measure  $\pi \in \mathcal{M}_+(X \times X)$ , its two marginals are denoted by  $(\text{Proj}_0)_\# \pi$  and  $(\text{Proj}_1)_\# \pi$  and are defined for any Borel set  $A$  via

$$(\text{Proj}_0)_\# \pi(A) = \pi(A \times X), \quad (\text{Proj}_1)_\# \pi(A) = \pi(X \times A).$$

**Definition 2.1** • (Semi-couplings) For  $\mu, \nu \in \mathcal{M}_+(\Omega)$ , the corresponding set of semi-couplings is

$$\Gamma(\mu, \nu) := \{(\pi_0, \pi_1) \in (\mathcal{M}_+(\Omega \times \Omega))^2 : (\text{Proj}_0)_\# \pi_0 = \mu, (\text{Proj}_1)_\# \pi_1 = \nu\}.$$

- (Cost function) A cost function is a function

$$c : (\Omega \times [0, \infty))^2 \rightarrow [0, \infty) \\ (x_0, m_0), (x_1, m_1) \mapsto c(x_0, m_0, x_1, m_1)$$

which is lower semi-continuous in all its arguments and jointly positively 1-homogeneous and convex in mass variables  $(m_0, m_1)$ . This function  $c(x_0, m_0, x_1, m_1)$  determines the cost of transporting a quantity of mass  $m_0$  from  $x_0$  to a (possibly different) quantity  $m_1$  at  $x_1$ .

- For a cost function  $c$  we introduce the functional

$$J_c(\pi_0, \pi_1) := \int_{\Omega \times \Omega} c\left(x, \frac{\pi_0}{\pi}, y, \frac{\pi_1}{\pi}\right) d\pi(x, y),$$

where  $\pi \in \mathcal{M}_+(\Omega \times \Omega)$  is any measure such that  $\pi_0, \pi_1 \ll \pi$ . This functional is well-defined since  $c$  is jointly 1-homogeneous w.r.t. the mass variables.

**Proposition 2.1** (Static formulation of WFR by semi-couplings [8]) *The WFR metric admits a static formulation characterized by semi-couplings:*

$$d_{\text{WFR},\alpha}^2(\mu, \nu) = \min_{(\pi_0, \pi_1) \in \Gamma(\mu, \nu)} J_c(\pi_0, \pi_1),$$

where  $c(x_0, m_0, x_1, m_1) = 2\alpha (m_0 + m_1 - 2\sqrt{m_0 m_1} \cdot \overline{\text{cos}}(|x_0 - x_1|/2\sqrt{\alpha}))$  and the truncated cosine  $\overline{\text{cos}}(z) = \cos(|z| \wedge \frac{\pi}{2})$ .

Recall the dynamic formulation (2.2). If  $g$  is restricted to the family of zero mean, then one obtains spherical Wasserstein–Fisher–Rao (SWFR) distance (a.k.a, spherical Hellinger–Kantorovich distance) [28, 29]. Such  $g$  can keep  $\rho$  as a probability measure. For convenience,

we use the short-hand notation  $\bar{g}_t = \int_{\Omega} g_t d\rho_t$ . SWFR distance used in the rest of paper has the following form:

$$d_{\text{SWFR},\alpha}^2(\mu, \nu) = \inf_{\rho, v, g} \left\{ \int_0^1 \int_{\Omega} \left( \frac{1}{2} |v_t|^2 + \frac{\alpha}{2} (g_t - \bar{g}_t)^2 \right) \rho_t(dx) dt : \right. \\ \left. \partial_t \rho_t + \nabla \cdot (\rho_t v_t) = \rho_t (g_t - \bar{g}_t), \rho_0 = \mu, \rho_1 = \nu \right\}. \tag{2.3}$$

It is remarked that the space  $(\mathcal{M}_+(\Omega), d_{\text{WFR}})$  can be identified with the cone over the space of probability measures  $(\mathcal{P}(\Omega), d_{\text{SWFR}})$ , due to the scaling property of WFR metric [5, 29]. The geodesic between two probability measures in  $(\mathcal{P}(\Omega), d_{\text{SWFR}})$  can be obtained by abstract projection from the cone  $(\mathcal{M}_+(\Omega), d_{\text{WFR}})$  to the spherical space  $(\mathcal{P}(\Omega), d_{\text{SWFR}})$ , namely by renormalizing the mass and by rescaling of arclength parameter. (see more details in [29] and references therein.)

Moreover, the WFR metrics (including SWFR metric) corresponds formally to a Riemannian structure on the space of measures, which generalizes the formulation of ordinary optimal transport. The transport, creation and destruction of mass between two measures can be described by this metric.

Finally, we remark that the dynamic formulations above require  $\mu$  and  $\nu$  to be absolutely continuous with respect to the Lebesgue measure. In applications,  $\mu$ , the data distribution, might be singular (which may concentrate on some low dimension manifold). In this case, one can always find a  $\tilde{\mu}$ , which is absolutely continuous w.r.t Lebesgue measure to approximate  $\mu$ . In specific tasks, we only have samples from  $\mu$ , the training process later using the optimal transport or the unbalanced optimal transport theory can learn a velocity field that automatically generates the approximation  $\tilde{\mu}$ .

## 2.2 Sample Generation Based on Particle Transportation

Let us briefly review the continuous normalizing flow (CNF) model [6] and OT-Flow model [40], which are sample generative models based on particle transportation purely.

CNF aims to build a continuous and invertible mapping between an arbitrary distribution  $\rho_0$  and standard normal distribution  $\rho_1$ . Alternatively, for a given time  $T$ , we are trying to obtain a mapping  $z : \mathbb{R}^d \times [0, T] \rightarrow \mathbb{R}^d$ . The mapping  $z$  defines a continuous evolution  $x \mapsto z(x, t)$  of every  $x \in \mathbb{R}^d$ , which can be viewed as trajectory of particles. Then the density  $\rho(z(x, t), t)$  satisfies

$$\log \rho_0(x) = \log \rho(z(x, t), t) + \log |\det \nabla z(x, t)| \quad \text{for all } x \in \mathbb{R}^d. \tag{2.4}$$

Especially at time  $T$  we have  $\log \rho_0(x) = \log \rho_1(z(x, T), T) + \log |\det \nabla z(x, T)|$ . Define  $\ell(x, t) := \log |\det \nabla z(x, t)|$ , then  $z(x, t)$  and  $\ell(x, t)$  satisfy the following ODE system

$$\partial_t \begin{bmatrix} z(x, t) \\ \ell(x, t) \end{bmatrix} = \begin{bmatrix} v(z(x, t), t; \theta) \\ \text{tr}(\nabla v(z(x, t), t; \theta)) \end{bmatrix}, \quad \begin{bmatrix} z(x, 0) \\ \ell(x, 0) \end{bmatrix} = \begin{bmatrix} x \\ 0 \end{bmatrix}. \tag{2.5}$$

where the second ODE can be derived from the first one. For convenience we solve them together to obtain the change of  $\rho$ , which will lead to a more efficient estimation of density.

Following is the derivation of second ODE in (2.5):

$$\begin{aligned} \frac{\partial \ell(x, t)}{\partial t} &= \frac{1}{\det(\nabla z(x, t))} \frac{\partial \det(\nabla z(x, t))}{\partial t} \\ &= \frac{1}{\det(\nabla z(x, t))} \cdot \det(\nabla z(x, t)) \cdot \text{tr} \left[ (\nabla z(x, t))^{-1} \frac{\partial \nabla z(x, t)}{\partial t} \right] \\ &= \frac{1}{\det(\nabla z(x, t))} \cdot \det(\nabla z(x, t)) \cdot \text{tr} \left[ (\nabla z(x, t))^{-1} \nabla z(x, t) \nabla v(z(x, t), t) \right] \\ &= \text{tr} [(\nabla v(z(x, t), t))], \end{aligned}$$

where we have used following identities

$$\frac{\partial \det(A)}{\partial t} = \det(A) \cdot \text{tr} \left[ A^{-1} \frac{\partial A}{\partial t} \right], \quad \text{tr}(AB) = \text{tr}(BA).$$

From the ODE system 2.5 we can see that if we have a velocity field, then we can track the evolution and obtain the final distribution at time  $T$ . Thus we can set the velocity field as an output of neural network and minimize the KL divergence between target distribution and final distribution obtained from ODE. Once the velocity field is learned, one may run the ODE backward so that the transport map can be inverted. The invertibility of CNF provides us with access to estimating density of sample space, which can be used for density estimation and Bayesian inference.

In general, the velocity field that transforms a given probability measure to a target one is not unique in the formulation of CNF. To tackle this problem, Finlay et al. [13] first introduced optimal transportation regularization in normalizing flows. Onken et al. [40] also proposed an improved version of CNF: OT-Flow, which leverages optimal transport theory to regularize the CNF and enforce straight trajectories that are easier for numerical integration. More precisely, OT-Flow designs the following cost functional to train the velocity field

$$J = D_{\text{KL}} [\rho(x, T) \parallel \rho_1(x)] + \gamma_1 \mathbb{E}_{\rho_0} \left[ \int_0^T \frac{1}{2} |v(z(x, t), t)|^2 dt \right]. \tag{2.6}$$

The first part in (2.6) is a KL divergence term as a soft terminal constraint, which enforces the terminal distribution  $\rho(x, T)$  transported by velocity field to get close to  $\rho_1$ . The second term is related to  $W_2$  distance in optimal transport theory, which can also be regarded as a penalty of the squared arc-length of the trajectories. Ideally if the KL divergence term is zero, minimizing the cost function is equivalent to minimizing  $W_2$  distance and solving the optimal velocity field, which will lead to two useful properties mentioned above, encouraging straight trajectory. Here  $\gamma_1$  is a hyper-parameter to balance KL divergence and trajectory penalty.

The above cost function cannot be used to train directly since we only have discrete samples as  $\rho_0$ . We would like to use Monte-Carlo to approximate the cost function, which requires us to rewrite cost function in the form of expectation over  $\rho_0$ . According to [40], we can simplify the KL divergence term with density relationship (2.4) as following:

$$\begin{aligned} &D_{\text{KL}} [\rho(z(\mathbf{x}, T)) \parallel \rho_1(z(\mathbf{x}, T))] \\ &= \int_{\mathbb{R}^d} \log \left( \frac{\rho(z(\mathbf{x}, T))}{\rho_1(z(\mathbf{x}, T))} \right) \rho(z(\mathbf{x}, T)) \det(\nabla z(\mathbf{x}, T)) d\mathbf{x}, \\ &= \int_{\mathbb{R}^d} \log \left( \frac{\rho_0(\mathbf{x})}{\rho_1(z(\mathbf{x}, T)) \det(\nabla z(\mathbf{x}, T))} \right) \rho_0(\mathbf{x}) d\mathbf{x}, \\ &= \int_{\mathbb{R}^d} \log(\rho_0(\mathbf{x})) \rho_0(\mathbf{x}) d\mathbf{x} + \mathbb{E}_{\rho_0(x)} \left[ -\log(\rho_1(z(\mathbf{x}, T))) - \log \det(\nabla z(\mathbf{x}, T)) \right]. \end{aligned} \tag{2.7}$$

By dropping constant in the formulation and substituting  $\rho_1$ , we get final cost function  $J$  as follows:

$$\begin{aligned}
 J &= \mathbb{E}_{\rho_\theta(x)} [C(x, T) + \gamma_1 L(x, T)], \\
 C(x, T) &= -\ell(x, T) + \frac{1}{2} |z(x, T)|^2 + \frac{d}{2} \log(2\pi), \\
 L(x, T) &= \int_0^T \frac{1}{2} |v(z(x, t), t)|^2 dt.
 \end{aligned}
 \tag{2.8}$$

OT-Flow can be regarded as a model to learn geodesics under Wasserstein distance. We will adopt a similar formulation to develop our deep learning framework for geodesics under the spherical WFR metric.

### 3 Geodesics Under Spherical WFR

In this section we will first review knowledge about geodesics based on [49, Chap. 5]. Then we derive basic equations for geodesics under spherical WFR. The formulations will be used to construct particle algorithm and new regularization based on the velocity field.

#### 3.1 Geodesics

In a metric space  $(X, d)$ , we define the length of a curve  $\omega : [0, 1] \rightarrow X$  as

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

which is also the total variation of  $\omega$ . We define metric derivative of the curve at  $t$  by

$$|\omega'(t)| = \lim_{s \rightarrow t} \frac{d(\omega(s), \omega(t))}{|s - t|}$$

if the limit exists [49, Chap. 5 Box 5.1]. For all absolutely continuous functions  $\omega$  in  $X$ , we have  $\text{Length}(\omega) < +\infty$  and  $\text{Length}(\omega) = \int_0^1 |\omega'(t)| dt$ .

**Definition 3.1** • (Geodesics) A curve  $\omega : [0, 1] \rightarrow X$  is said to be a geodesic between  $x_0$  and  $x_1 \in X$  if it minimizes the length among all curves such that  $\omega(0) = x_0$  and  $\omega(1) = x_1$ .

- (Length space) A space  $(X, d)$  is said to be a length space if it holds:

$$d(x, y) = \inf \{ \text{Length}(\omega) : \omega \text{ is absolute continuous in } X, \omega(0) = x, \omega(1) = y \}.$$

- (Constant-speed geodesic) In a length space, a curve  $\omega : [0, 1] \rightarrow X$  is said to be a constant-speed geodesic between  $\omega(0)$  and  $\omega(1)$  if it satisfies:

$$d(\omega(t), \omega(s)) = |t - s| d(\omega(0), \omega(1)) \quad \text{for all } t, s \in [0, 1].
 \tag{3.1}$$

The following result describes the structure of geodesics under Wasserstein distance.

**Proposition 3.1** (Theorem 5.27 in [49]) *Suppose  $\Omega$  is convex.  $(\mathcal{W}_p, W_p)$  is the metric space.  $\mu, \nu \in \mathcal{W}_p$  and  $\gamma \in \Pi(\mu, \nu)$  is an optimal transport plan for the cost  $c(x, y) = |x - y|^p (p \geq 1)$ . Let  $\pi_t(x, y) = (1 - t)x + ty$ . Then the curve  $\mu_t = (\pi_t)_\# \gamma$  is a constant-speed geodesics in  $\mathcal{W}_p$  connecting  $\mu_0 = \mu$  and  $\mu_1 = \nu$ . As a consequence, the space  $\mathcal{W}_p(\Omega)$  is a length space.*

According to the description of the geodesics, a particle that is transported from  $x$  to  $y$  moves under the evolution  $\pi_t(x, y) = (1 - t)x + ty$ , which indicates that the particle is moving with constant velocity  $y - x$ . Moreover, in the optimal transport plan, the particles will not meet each other. The trajectories are straight lines that are non-intersecting in space-time plane.

Though WFR is a generalization of Wasserstein distance, the structure of the geodesics under spherical WFR metric and corresponding particle motions are not as clear as the Wasserstein case. In particular, while the geodesics for Wasserstein distance can be obtained easily using the optimal plan in the static formulation by Proposition 3.1, the one for spherical WFR cannot be obtained directly if one knows the optimal solution to the static formulation in Proposition 2.1.

As a start, we note the following simple observation for the dynamic formulation.

**Proposition 3.2** *The optimal  $(\rho_t)_{t \in [0,1]}$  satisfying dynamic description of spherical WFR (2.3) is exactly the constant speed geodesics under spherical WFR.*

**Proof** Consider the optimal solution  $\rho_t^*, g_t^*, v_t^*, t \in [0, 1]$  satisfying minimization problem (2.3). For  $\beta_1, \beta_2 \in [0, 1], \beta_1 < \beta_2$ , we have

$$\begin{aligned} \int_{\beta_1}^{\beta_2} \int_{\Omega} \left( \frac{1}{2} |v_t^*|^2 + \frac{\alpha}{2} (g_t^* - \bar{g}_t^*)^2 \right) \rho_t(dx) dt &= \min_{\rho, v, g} \left\{ \int_{\beta_1}^{\beta_2} \int_{\Omega} \left( \frac{1}{2} |v|^2 + \frac{\alpha}{2} (g - \bar{g})^2 \right) \rho(dx) dt : \right. \\ &\quad \left. \partial_t \rho + \nabla \cdot (\rho v) = \rho(g - \bar{g}), \rho(\beta_1) = \rho_{\beta_1}, \rho(\beta_2) = \rho_{\beta_2} \right\} \\ &= \frac{1}{\beta_2 - \beta_1} \min_{\rho_{\tau}, v_{\tau}, g_{\tau}} \left\{ \int_0^1 \int_{\Omega} \left( \frac{1}{2} |v_{\tau}|^2 + \frac{\alpha}{2} (g_{\tau} - \bar{g}_{\tau})^2 \right) \rho_{\tau}(dx) d\tau : \right. \\ &\quad \left. \partial_{\tau} \rho_{\tau} + \nabla \cdot (\rho_{\tau} v_{\tau}) = \rho_{\tau}(g_{\tau} - \bar{g}_{\tau}), \rho_{\tau}(0) = \rho_{\beta_1}, \rho_{\tau}(1) = \rho_{\beta_2} \right\} \\ &= \frac{1}{\beta_2 - \beta_1} d_{\text{SWFR}}^2(\rho_{\beta_1}, \rho_{\beta_2}). \end{aligned}$$

The first equality follows directly from minimization problem (2.3). In fact, if it has a better solution  $(\tilde{\rho}_t, \tilde{v}_t, \tilde{g}_t)_{\beta_1 \leq t \leq \beta_2}$  such that the objective function is smaller, then we may concatenate with  $(\rho_t^*, v_t^*, g_t^*)_{t \in [0, \beta_1] \cup [\beta_2, 1]}$  to get a better solution for the original minimization problem. Note that the concatenated solution is still continuous due to the boundary condition, so the continuity equation still holds in weak sense. The second equality is a simple time rescaling  $\tau = \frac{1}{\beta_2 - \beta_1}(t - \beta_1)$ . Correspondingly, one has the following correspondence:  $\rho_{\tau}(x, \tau) = \rho(t, x), v_{\tau}(x, \tau) = (\beta_2 - \beta_1)v(x, t), g_{\tau}(x, \tau) = (\beta_2 - \beta_1)g(x, t)$  for  $\tau \in [0, 1]$ . Thus we have

$$\frac{1}{\beta_2} d_{\text{SWFR}}^2(\rho_0, \rho_{\beta_2}) = \frac{1}{\beta_1} d_{\text{SWFR}}^2(\rho_0, \rho_{\beta_1}) + \frac{1}{\beta_2 - \beta_1} d_{\text{SWFR}}^2(\rho_{\beta_1}, \rho_{\beta_2}). \tag{3.2}$$

Note that  $d_{\text{SWFR}}(\rho_0, \rho_{\beta_2}) \leq d_{\text{SWFR}}(\rho_0, \rho_{\beta_1}) + d_{\text{SWFR}}(\rho_{\beta_1}, \rho_{\beta_2})$ , thus we have

$$\begin{aligned} \frac{\beta_2 - \beta_1}{\beta_1 \beta_2} d_{\text{SWFR}}^2(\rho_0, \rho_{\beta_1}) + \frac{\beta_1}{(\beta_2 - \beta_1) \beta_2} d_{\text{SWFR}}^2(\rho_{\beta_1}, \rho_{\beta_2}) - \\ \frac{2}{\beta_2} d_{\text{SWFR}}(\rho_0, \rho_{\beta_1}) \cdot d_{\text{SWFR}}(\rho_{\beta_1}, \rho_{\beta_2}) \leq 0. \end{aligned}$$

In fact the left side  $\geq 0$ . Thus the equal sign holds only when

$$\frac{d_{\text{SWFR}}^2(\rho_0, \rho_{\beta_1})}{\beta_1^2} = \frac{d_{\text{SWFR}}^2(\rho_{\beta_1}, \rho_{\beta_2})}{(\beta_2 - \beta_1)^2}. \tag{3.3}$$

Setting  $\beta_2 = 1$  in (3.2) and using (3.3), one finds that  $d_{\text{SWFR}}(\rho_0, \rho_{\beta_1}) = \beta_1 d_{\text{SWFR}}(\rho_0, \rho_1)$ . It follows then by (3.3) that  $d_{\text{SWFR}}(\rho_{\beta_2}, \rho_{\beta_1}) = (\beta_2 - \beta_1)d_{\text{SWFR}}(\rho_0, \rho_1)$  for  $\forall \beta_1, \beta_2 \in [0, 1], \beta_1 < \beta_2$ . By definition (3.1) we know that the optimal  $(\rho_t)_{t \in [0,1]}$  is the constant speed geodesic under spherical WFR.  $\square$

### 3.2 Equations for the Density and Potential for the Spherical WFR Metric

Below, we make use of Proposition 3.2 to investigate the geodesics under spherical WFR metric. Consider the following optimization problem

$$\min_{\rho, v, g} \left\{ \frac{1}{2} \int_0^1 \int_{\mathbb{R}^d} \rho |v|^2 + \alpha \rho g^2 dx dt, \partial_t \rho + \nabla \cdot (\rho v) = \rho g, \int_{\mathbb{R}^d} \rho g dx = 0 \right\}, \tag{3.4}$$

where  $\alpha$ , the source coefficient, is to balance the effects of transport and creation/destruction of mass explicitly. The Lagrangian function for (3.4) is

$$\begin{aligned} \mathcal{L} = & \frac{1}{2} \int_0^1 \int_{\mathbb{R}^d} \rho |v|^2 + \alpha \rho g^2 dx dt - \int_0^1 \int_{\mathbb{R}^d} \Phi(x, t) (\partial_t \rho + \nabla \cdot (\rho v) - \rho g) dx dt \\ & - \int_0^1 \gamma(t) \left( \int_{\mathbb{R}^d} \rho g dx \right) dt, \end{aligned}$$

where  $\Phi(x, t)$  and  $\gamma(t)$  are all Lagrange multipliers. Taking the variation, by the first order optimality conditions, one has

$$\begin{cases} \frac{\delta \mathcal{L}}{\delta v} = 0, \\ \frac{\delta \mathcal{L}}{\delta g} = 0, \\ \frac{\delta \mathcal{L}}{\delta \rho} = 0. \end{cases} \implies \begin{cases} v = -\nabla \Phi, \\ g = -\frac{1}{\alpha}(\Phi - \gamma(t)), \\ \partial_t \Phi = \frac{1}{2}|\nabla \Phi|^2 + \frac{1}{2\alpha}(\Phi - \gamma(t))^2. \end{cases} \tag{3.5}$$

Using the fact  $\int \rho g dx = 0$ , one can determine  $\gamma(t) = \int \Phi d\rho =: \bar{\Phi}(t)$ . Then the evolution of  $\rho$  is governed by

$$\partial_t \rho - \nabla \cdot (\rho \nabla \Phi) = -\frac{1}{\alpha} \rho (\Phi - \bar{\Phi}). \tag{3.6}$$

To summarize, we get the equations of  $\rho$  and  $\Phi$

$$\begin{cases} \partial_t \rho - \nabla \cdot (\rho \nabla \Phi) = -\frac{1}{\alpha} \rho (\Phi - \bar{\Phi}), \\ \partial_t \Phi = \frac{1}{2}|\nabla \Phi|^2 + \frac{1}{2\alpha}(\Phi - \bar{\Phi})^2. \end{cases} \tag{3.7}$$

We remark that if we define a Hamiltonian

$$H(\rho, \Phi) = \frac{1}{2} \int_0^1 \int_{\mathbb{R}^d} \rho |\nabla \Phi|^2 + \frac{1}{\alpha} \rho (\Phi - \bar{\Phi})^2 dx dt,$$

then system (3.7) can be rewritten as

$$\begin{cases} \dot{\rho} = -\frac{\delta H}{\delta \Phi}, \\ \dot{\Phi} = \frac{\delta H}{\delta \rho}. \end{cases}$$

The Hamiltonian structure for the minimizer is in fact a consequence of the well-known Pontryagin Maximum Principle [12], if we regard the minimization problem (2.3) of SWFR distance as an infinite-dimensional (mean-field) control problem. The equation that governs the evolution of  $\Phi$  is also called Hamilton–Jacobi equation, i.e. the second equation in (3.7).

### 3.3 The Evolution of Distribution and Particles Along the Geodesics

In order to learn the geodesics under spherical WFR, we will derive related equations of weight evolution, density formula and velocity field evolution first. We use  $z : \mathbb{R}^d \times [0, T] \rightarrow \mathbb{R}^d$  to represent the trajectory of particles. Consider a weighted formulation of  $\rho$ :

$$\rho(z, t) = \int_{\mathbb{R}^d} w(x, t)\delta(z - z(x, t))p(x)dx \tag{3.8}$$

for some measure  $p$  so that  $\rho_0(x) = p(x)w(x, 0)$ . Expression (3.8) can be rigorously understood in the weak sense:

$$\int_{\mathbb{R}^d} \eta(z)d\rho(z, t) = \int_{\mathbb{R}^d} \eta(z(x, t))w(x, t)p(x)dx, \quad \forall \eta \in C_b(\mathbb{R}^d). \tag{3.9}$$

Here  $C_b(\mathbb{R}^d)$  refers to the class of bounded continuous functions. We have the following claims about density and particle velocity along the trajectories.

**Theorem 3.1** (i) *If the positions  $z$  and weights  $w$  of the particles satisfy the following ODE system:*

$$\begin{cases} \frac{d}{dt}z(x, t) = -\nabla\Phi(z(x, t), t), \\ \frac{d}{dt}w(x, t) = -\frac{1}{\alpha}(\Phi(z(x, t), t) - \bar{\Phi}(t))w(x, t), \end{cases} \tag{3.10}$$

*then the unique solution  $\rho$  of (3.6) satisfies the following density formula under spherical WFR*

$$\rho_0(x)e^{-\frac{1}{\alpha}\int_0^t(\Phi(z(x,s),s) - \bar{\Phi}(s))ds} = \rho(z(x, t), t) \cdot \det(\nabla z(x, t)). \tag{3.11}$$

(ii) *For the particle velocity under spherical WFR, it holds that*

$$v(z(x, t), t) = v(x, 0)e^{\frac{1}{\alpha}\int_0^t(\Phi(z(x,s),s) - \bar{\Phi}(s))ds} = \frac{v(x, 0)w(x, 0)}{w(x, t)}. \tag{3.12}$$

*Consequently, the particles move in straight lines.*

**Proof** (i) Considering an ODE system given by

$$\begin{cases} \frac{d}{dt}z(x, t) = v(z(x, t), t), \\ \frac{d}{dt}w(x, t) = g(z(x, t), t)w(x, t). \end{cases} \tag{3.13}$$

Consider  $\rho(z, t)$  in (3.8) (or equivalently (3.9)) with  $z$  and  $w$  given by (3.13), then  $\rho$  solves following PDE in the sense of distribution:

$$\partial_t \rho + \nabla \cdot (\rho v) = \rho g. \tag{3.14}$$

If  $v$  and  $g$  are Lipschitz continuous, the solution of (3.14) must be unique and so it must be the one considered above. Plugging  $v = -\nabla\Phi$  and  $g = \Phi - \bar{\Phi}$ , we have the ODE system (3.10). Thus the density  $\rho$  (3.8) with the weight and position given by ODE system (3.10) solves (3.6) uniquely. One can solve  $w(x, t)$  as

$$w(x, t) = w(x, 0)e^{-\frac{1}{\alpha} \int_0^t (\Phi(z(x,s),s) - \bar{\Phi}(s)) ds}.$$

The definition of  $\rho$  in (3.9) immediately yields (3.11).

- (ii) Recall (3.5) from Lagrangian equation of spherical WFR, taking the gradient of the Hamilton–Jacobi equation, one has

$$\partial_t v + v \cdot \nabla v - \frac{1}{\alpha} (\Phi - \bar{\Phi})v = 0.$$

We use the classical argument of characteristic lines. Let  $\gamma(s; x, t)$  be the characteristic line which satisfies

$$\begin{cases} \frac{d\gamma(s; x, t)}{ds} = v(\gamma(s; x, t), s), \\ \gamma(t; x, t) = x, \end{cases}$$

then  $U(s) := v(\gamma(s; x, t), s)$  satisfies

$$U'(s) = \partial_t v + v \cdot \nabla v = \frac{1}{\alpha} (\Phi - \bar{\Phi})v(\gamma(s; x, t), s) = \frac{1}{\alpha} (\Phi - \bar{\Phi})U(s).$$

It follows that

$$v(x, t) = U(t) = U(0)e^{\frac{1}{\alpha} \int_0^t (\Phi(\gamma(s;x,t),s) - \bar{\Phi}(s)) ds}.$$

Note that  $\gamma(s; z(x, t), t) = z(x, s)$  and  $\gamma(0; z(x, t), t) = x$ . Hence

$$v(z(x, t), t) = v(x, 0)e^{\frac{1}{\alpha} \int_0^t (\Phi(z(x,s),s) - \bar{\Phi}(s)) ds} = \frac{v(x, 0)w(x, 0)}{w(x, t)}.$$

Since the direction of the velocity does not change, the particles move in straight lines, which concludes the proof. □

**Remark 3.1** In general,  $z$  and  $w$  do not have to satisfy (3.13), in order for the measure (3.8) to be a solution of (3.14). In particular, if we replace first equation in (3.13) with  $\dot{z} = v(z, t) + q(z, t)$ , where  $q$  satisfies  $\nabla \cdot (q\rho) = 0$ , the measure (3.8) also solves (3.14) and it must be the same measure as we considered in the proof. That means the same unique solution  $\rho$  may correspond to several  $(z, w)$  pairs in the form of (3.8). However, if we restrict  $\rho$  to be an empirical measure, then ODE system (3.13) becomes necessary for solving (3.14) since  $q$  has to be zero along the trajectory.

The formula (3.12) indicates that, in the optimal case, the magnitude of particle velocity is inversely proportional to its weight while the direction of velocity remains the same along each trajectory.

### 4 Learning Geodesics Under Spherical WFR

In this section, we propose a deep learning framework to learn the geodesics under spherical WFR metric. A KL divergence term based on the inverse map is used for the terminal

condition. Moreover, a new regularization term based on particle velocity and weight is introduced in Sect. 4.2 as a substitute for the Hamilton–Jacobi equation for the potential in dynamic formulation. Then, in Sect. 4.3, we obtain the total cost for our model and make some discussion on the hyper-parameter  $\gamma_1$ . Detailed implementation of the algorithm is showed in Sect. 4.4.

### 4.1 Imposing the Terminal Condition

Suppose we are given a starting distribution  $\rho_0$ , and we may know or may not know the expression. In the latter case, we may assume some samples drawn from  $\rho_0$ . We are also given the terminal distribution  $\rho_1$  up to potentially an unknown normalizing constant.

To compute the geodesics, we may impose the terminal condition  $\rho(\cdot, t = 1) = \rho_1$ . We propose in this work to construct the KL divergence using the inverse map. In particular, we consider the probability distribution evolved from the terminal distribution  $\rho_1$ , denoted by  $\tilde{\rho}_0$ . Then, compare it to the initial distribution  $\rho_0$ .

Let  $\tilde{\rho}(z, T - t) := \rho(z, t)$ . One can deduce from (3.7) that

$$\begin{cases} \partial_t \tilde{\rho}(z, t) + \nabla \cdot (\tilde{\rho}(z, t) \nabla \Phi(z, T - t)) = \frac{1}{\alpha} \tilde{\rho}(z, t) (\Phi(z, T - t) - \hat{\Phi}(T - t)), \\ \tilde{\rho}(z, 0) = \rho_1(z), \quad \hat{\Phi}(T - t) = \int \Phi(z, T - t) \tilde{\rho}(z, t) dz. \end{cases} \tag{4.1}$$

Similarly, using  $x(z, t) = z(x, T - t)$ , it holds that

$$\rho_1(z) e^{\frac{1}{\alpha} \int_0^T (\Phi(x(z,t), T-t) - \hat{\Phi}(T-t)) dt} = \tilde{\rho}(x(z, T), T) \det(\nabla_z x(z, T)), \tag{4.2}$$

which is

$$\tilde{\rho}(x, T) e^{-\frac{1}{\alpha} \int_0^T (\Phi(z(x,t), t) - \hat{\Phi}(t)) dt} = \rho_1(z(x, T)) \det(\nabla_x z(x, T)). \tag{4.3}$$

We turn to minimize the KL divergence between  $\tilde{\rho}(x, T)$  and  $\rho_0(x)$

$$\begin{aligned} D_{\text{KL}}[\rho_0(x) \parallel \tilde{\rho}(x, T)] &= \int_{\mathbb{R}^d} \log \left( \frac{\rho_0(x)}{\tilde{\rho}(x, T)} \right) \rho_0(x) dx \\ &= \int_{\mathbb{R}^d} \log \left( \frac{\rho_0(x) e^{-\frac{1}{\alpha} \int_0^T (\Phi(z(x,t), t) - \hat{\Phi}(t)) dt}}{\rho_1(z(x, T)) \cdot \det(\nabla_x z(x, T))} \right) \rho_0(x) dx \\ &= \int_{\mathbb{R}^d} [-\log(\rho_1(z(x, T))) - \log \det(\nabla_x z(x, T))] \rho_0(x) dx \\ &\quad - \frac{1}{\alpha} \int_{\mathbb{R}^d} \left( \int_0^T (\Phi(z(x, t), t) - \hat{\Phi}(t)) dt \right) \rho_0(x) dx + \int_{\mathbb{R}^d} \log(\rho_0(x)) \rho_0(x) dx. \end{aligned} \tag{4.4}$$

The term  $\int \log(\rho_0(x)) \rho_0(x) dx$  can be dropped, since it is unrelated to the parameters to optimize.

We emphasize that we use here a different KL divergence object compared with CNFs rather than adopting a new method to calculate the original KL object. In OT-Flow [40], they use  $D_{\text{KL}}[\rho(x, T) \parallel \rho_1(x)]$ , while we use  $D_{\text{KL}}[\rho_0(x) \parallel \tilde{\rho}(x, T)]$  in our framework. The above two KL divergence formulations are equivalent in CNFs case, but they are different after introducing weight change. We now explain why we use a new KL divergence based on the inverse map to impose terminal condition, instead of directly using KL divergence between

$\rho(\cdot, t = 1)$  and  $\rho_1$ . Mimicking the KL divergence approach in the CNF framework and using (3.11), we have

$$\begin{aligned}
 D_{\text{KL}}[\rho(z(x, T))\|\rho_1(z)] &= \int_{\mathbb{R}^d} \log\left(\frac{\rho(z(x, T))}{\rho_1(z(x, T))}\right) \rho(z(x, T)) dz \\
 &= \int_{\mathbb{R}^d} [-\log(\rho_1(z(x, T))) - \log(\det(\nabla z(x, T)))] \rho_0(x) e^{-\frac{1}{\alpha} \int_0^T (\Phi(z(x,t),t) - \bar{\Phi}(t)) dt} dx \\
 &\quad - \frac{1}{\alpha} \int_{\mathbb{R}^d} \left( \int_0^T (\Phi(z(x,t),t) - \bar{\Phi}(t)) dt \right) \rho_0(x) e^{-\frac{1}{\alpha} \int_0^T (\Phi(z(x,t),t) - \bar{\Phi}(t)) dt} dx \quad (4.5) \\
 &\quad + \int_{\mathbb{R}^d} \log(\rho_0(x)) \rho_0(x) e^{-\frac{1}{\alpha} \int_0^T (\Phi(z(x,t),t) - \bar{\Phi}(t)) dt} dx \\
 &=: I_1 + I_2 + I_3.
 \end{aligned}$$

Clearly,  $I_1$  and  $I_2$  can be approximated using the standard Monte Carlo approximation. However,  $I_3$  cannot be computed easily if the expression of  $\rho_0$  is unknown or complicated, and we cannot drop it since it is not a constant with respect to model parameters.

There are also two alternative options to address the problem of estimating  $I_3$ . We tried them in solving above problems brought by weight change, while we decided not to follow these approaches due to the drawbacks discussed below.

- *Estimating initial density* One possible strategy is to use the variational expression of  $\log(\rho_0(x)/\rho(x))$  introduced in [22] to approximate  $\log(\rho_0)$ . In fact,

$$\log(\rho_0/\rho) = \operatorname{argmin}_{D'} [\mathbb{E}_{x \sim \rho_0} \log(1 + e^{-D'(x)}) + \mathbb{E}_{x \sim \rho} \log(1 + e^{D'(x)})],$$

where the argument of the minimization problem  $D'$  is a function of  $x$ . In practice, one may take some class of functions for the optimization and use the optimal one  $\mathcal{D}(x)$  from this class to approximate the theoretical optimum. Taking  $\rho(x)$  as the standard normal distribution,  $\log(\rho_0)$  can be then computed as

$$\log(\rho_0(x)) = -\frac{1}{2}|x|^2 - \frac{d}{2} \log(2\pi) + \mathcal{D}(x). \quad (4.6)$$

- *Kernelized Stein Discrepancy (KSD)* The second potential approach is replacing KL divergence with another weak metric such as the kernelized Stein discrepancy [9, 17, 35] or the Wasserstein distances, to avoid the density estimation.

The approach to approximate  $\log(\rho_0)$  by (4.6) works in some toy cases but is not appealing as a building block in our algorithm. First, in high dimension case such estimation can be tricky and costly. Second, estimating data density is one of essential applications of CNF models. Though approximating  $\rho_0$  out first is feasible, we would prefer solutions without estimating  $\rho_0$  to build our framework. For the alternative KSD metric, we were not able to obtain satisfactory result when the initial density's support is not connected. [21] argues that when KSD is small, it means that within the region of generated samples, the score function of  $s_p$  matches the target score function  $s_q$  well. An almost-zero empirical KSD does not necessarily imply capturing all the modes or recovering all the support of the true density. Considering these reasons and results of numerical experiments we did, we choose to construct KL divergence of  $\rho_0$  using inverse map in our framework eventually.

### 4.2 Regularization

For OT-Flow, Onken et al. [40] used HJB equation of  $\Phi$  to construct a regularization to help training and accelerate convergence. We hope to construct a regularization similarly to obtain a better velocity field in training. Instead of using the Hamilton–Jacobi equation, we make use of the relationship between velocity field and weight in spherical WFR (3.12) to impose

$$\begin{aligned} \mathbf{R}_v &:= \int_0^T \int_{\mathbb{R}^d} |v(z(x, t), t)w(x, t) - v(x, 0)w(x, 0)|^2 \rho_0(x) e^{-\frac{1}{\alpha} \int_0^t (\Phi(z(x, s), s) - \bar{\Phi}(s)) ds} dx dt \\ &= \int_0^T \int_{\mathbb{R}^d} |v(z(x, t), t)w(x, t) - v(x, 0)w(x, 0)|^2 \rho_0(x) \frac{w(x, t)}{w(x, 0)} dx dt \end{aligned} \tag{4.7}$$

as regularization. Such a term penalizes the velocity field along the trajectory, which can lead to a better velocity field suited for our framework in training.

### 4.3 Total Cost

We are supposed to learn the geodesics under spherical WFR. In our framework, once the potential  $\Phi$  is known, the velocity field, the source field and thus geodesic curve can be computed. Thus we parameterize  $\Phi$  as an output of a neural network, and total cost function with regularization can be written as:

$$\begin{aligned} J(\Phi) &= \text{D}_{\text{KL}}[\rho_0(x) \|\tilde{\rho}(x, T)] + \gamma_1 \cdot \int_0^T \int_{\mathbb{R}^d} \left( \frac{1}{2} |\nabla \Phi|^2 + \frac{1}{2\alpha} (\Phi - \bar{\Phi})^2 \right) \rho(dz) dt \\ &\quad + \gamma_2 \cdot \mathbf{R}_v, \end{aligned} \tag{4.8}$$

where  $\gamma_1$  and  $\gamma_2$  are the hyper-parameters to balance the terms in cost function  $J$ . Since we will sample from data space later to approximate the value by Monte-Carlo, we rewrite the integral into the form of expectation over initial density for convenience. By 4.4 the KL term can be rewritten as

$$\begin{aligned} \text{D}_{\text{KL}}[\rho_0(x) \|\tilde{\rho}(x, T)] &= \mathbb{E}_{\rho_0(x)} \left[ -\log(\rho_1(z(x, T))) - \log \det(\nabla z(x, T)) \right. \\ &\quad \left. - \frac{1}{\alpha} \int_0^T (\Phi(z(x, t), t) - \hat{\Phi}(t)) dt \right] + \mathbb{E}_{\rho_0(x)} \left[ \log(\rho_0(x)) \right]. \end{aligned}$$

where  $\hat{\Phi}(t)$  is defined in (4.1) and the detail of the implementation will be explained in Sect. 4.4. The second part can be rewritten as:

$$\begin{aligned} \gamma_1 \cdot \int_0^T \int_{\mathbb{R}^d} \left( \frac{1}{2} |\nabla \Phi|^2 + \frac{1}{2\alpha} (\Phi - \bar{\Phi})^2 \right) \rho(dz) dt \\ = \frac{\gamma_1}{2} \mathbb{E}_{\rho_0(x)} \left[ \int_0^T \left( |\nabla \Phi(z(x, t), t)|^2 + \frac{1}{\alpha} (\Phi(z(x, t), t) - \bar{\Phi}(t))^2 \right) \cdot e^{-\frac{1}{\alpha} \int_0^t (\Phi(z(x, s), s) - \bar{\Phi}(s)) ds} dt \right]. \end{aligned}$$

The third term is obvious from (4.7):

$$\gamma_2 \cdot \mathbf{R}_v = \gamma_2 \mathbb{E}_{\rho_0(x)} \left[ \int_0^T |\nabla \Phi(z(x, t), t)w(x, t) - \nabla \Phi(x, 0)w(x, 0)|^2 \frac{w(x, t)}{w(x, 0)} dt \right].$$

### Discussion on the Hyper-Parameter $\gamma_1$

The bigness of the parameter  $\gamma_1$  is important. As we hope the constraint  $\rho(\cdot, t = 1) = \rho_1$  to be a hard constraint (i.e. the KL divergence should be zero), we choose  $\gamma_1$  to be relatively small. In fact, if we ignore the regularization term and consider parameter  $\gamma_1$  alone. It may be shown theoretically that if  $\gamma_1 \rightarrow 0$ , the optimal solution will converge to the geodesics. Alexander Vidal et al. [54] gives similar conclusions on the choice of  $\gamma_1$  and develops methods to tame hyper-parameter tuning using Jordan-Kinderlehrer-Otto (JKO) scheme [23]. In particular, the following proposition tells us why this is a reasonable approach.

**Proposition 4.1** *Consider the optimization problem*

$$\min_{u \in X} E(u) + \gamma F(u).$$

*If both functionals are convex, lower semi-continuous and  $U := \operatorname{argmin}(E)$  is nonempty, then as  $\gamma \rightarrow 0^+$ , a cluster point of the minimizers  $\{u(\gamma)\}$  is a solution to the following problem*

$$\min_{u \in U} F(u).$$

For the proof of this proposition, one may set  $E_* = \inf E(u) > -\infty$  since  $U$  is assumed to be nonempty. Then, consider  $J_\alpha = F(u) + \alpha(E(u) - E_*)$  and  $J_\infty = F(u) + \mathbf{1}(U)$ , where the indicator function

$$\mathbf{1}(U) = \begin{cases} 0 & x \in U \\ +\infty & x \notin U. \end{cases}$$

Then, one may verify the  $\Gamma$ -convergence of  $J_\alpha$  to  $J_\infty$  as  $\alpha \rightarrow \infty$  [4, 8]. This proposition suggests that if we choose  $\gamma_1$  small enough, our framework can give a good approximation to the geodesics under the spherical WFR metric.

As an additional remark, if the parameter  $\gamma_1$  is big, a single step is like an implicit Euler step for the gradient descent of the KL divergence, which is the generalization of Jordan-Kinderlehrer-Otto (JKO) scheme [10, 23, 30].

### 4.4 Detailed Algorithm and Implementation

In practical computations, we can only use discrete particles to approximate the high dimensional distribution. Consider empirical measure for particle system:

$$\rho(x, t) = \sum_{i=1}^n w_i(t) \delta(x - x_i(t)),$$

where  $w_i(t)$  denotes the weight of particle  $x_i$  at time  $t$ . Taking the empirical measure into (3.6) and considering the second ODE in (2.5), one obtains following ODE system

$$\partial_t \begin{bmatrix} z_i(t) \\ w_i(t) \\ \ell(x_i, t) \end{bmatrix} = \begin{bmatrix} -\nabla \Phi(z_i(t), t) \\ -\frac{1}{\alpha} (\Phi(z_i(t), t) - \bar{\Phi}(t)) w_i(t) \\ -\operatorname{tr}(\nabla^2 \Phi(z_i(t), t)) \end{bmatrix}, \quad \begin{bmatrix} z_i(0) \\ w_i(0) \\ \ell(x_i, 0) \end{bmatrix} = \begin{bmatrix} x_i \\ w_0(x_i) \\ 0 \end{bmatrix}. \tag{4.9}$$

where  $\bar{\Phi}$  is approximated by

$$\bar{\Phi}(t) = \frac{1}{n} \sum_{i=1}^n w_i(t) \Phi(z_i(t), t), \tag{4.10}$$

and we identify  $z_i(t)$  with  $z(x_i, t)$  as the  $i^{\text{th}}$  particle’s trajectory and  $w_0(x_i)$  as  $i^{\text{th}}$  particle’s initial weight. Assume that  $\sum_{i=1}^n w_i(0) = n$ , then  $\sum_{i=1}^n w_i(t) \equiv n$  for any  $t > 0$ . Alternatively, one may choose any convenient  $\bar{\Phi}(t)$  to integrate (4.9) and then normalize  $\{w_i(t)\}$  for the sum to be  $n$  so that the true  $\bar{\Phi}(t)$  can be approximated using (4.10).

The first term in (4.8) is the KL divergence

$$D_{\text{KL}}[\rho_0(x) \|\tilde{\rho}(x, T)] = \mathbb{E}_{\rho_0(x)} \left[ -\log(\rho_1(z(x, T))) - \log \det(\nabla z(x, T)) \right. \\ \left. - \frac{1}{\alpha} \int_0^T (\Phi(z(x, t), t) - \hat{\Phi}(t)) dt \right] + \mathbb{E}_{\rho_0(x)} \left[ \log(\rho_0(x)) \right].$$

We can drop the term  $\mathbb{E}_{\rho_0(x)} \left[ \log(\rho_0(x)) \right]$  which does not affect the training, thus

$$J_{\text{KL}} := \mathbb{E}_{\rho_0(x)} \left[ -\log \rho_1(z(x, T)) - \ell(x, T) + \frac{1}{\alpha} \int_0^T (\Phi(z(x, t), t) - \hat{\Phi}(t)) dt \right] \\ \approx \frac{1}{n} \sum_{i=1}^n \left[ -\ell(x_i, T) - \log \rho_1(z(x_i, T)) + \frac{1}{\alpha} \int_0^T \Phi(z(x_i, t), t) dt \right] - \frac{1}{\alpha} \int_0^T \hat{\Phi}(t) dt.$$

To compute  $\int_0^T \hat{\Phi}(t) dt = \int_0^T \hat{\Phi}(T-t) dt$ , assuming  $\{\hat{Z}_j\}_{j=1}^N$  are samples from  $\rho_1$ , we solve

$$\partial_t \begin{bmatrix} \hat{x}_j(t) \\ \hat{w}_j(t) \end{bmatrix} = \begin{bmatrix} \nabla \Phi(\hat{x}_j(t), T-t) \\ \frac{1}{\alpha} (\Phi(\hat{x}_j(t), T-t) - \hat{\Phi}(T-t)) \hat{w}_j(t) \end{bmatrix}, \quad \begin{bmatrix} \hat{x}_j(0) \\ \hat{w}_j(0) \end{bmatrix} = \begin{bmatrix} \hat{Z}_j \\ 1 \end{bmatrix}. \quad (4.11)$$

Here  $\hat{\Phi}$  can be calculated as

$$\hat{\Phi}(T-t) = \frac{1}{N} \sum_{j=1}^N \hat{w}_j(t) \Phi(\hat{x}_j(t), T-t). \quad (4.12)$$

Alternatively, one may solve the ODE system (4.9) backward in time with terminal conditions to approximate  $\hat{\Phi}$ . Again in (4.11), the function  $\hat{\Phi}(T-t)$  can be replaced by any other convenient function (e.g. 0) and one needs only to normalize  $\hat{w}_j(t)$  and then use (4.12) for the approximation.

The second term in (4.8) characterizes spherical WFR distance

$$J_{\text{SWFR}} := \int_0^T \int_{\mathbb{R}^d} \left( \frac{1}{2} \rho |v|^2 + \frac{1}{2} \alpha \rho g^2 \right) dz dt \\ = \frac{1}{2} \mathbb{E}_{\rho_0(x)} \left[ \int_0^T \left( |\nabla \Phi(z(x, t), t)|^2 + \frac{1}{\alpha} (\Phi(z(x, t), t) - \bar{\Phi}(t))^2 \right) e^{-\frac{1}{\alpha} \int_0^t (\Phi(z(x, s), s) - \bar{\Phi}(s)) ds} dt \right] \\ \approx \frac{1}{2n} \sum_{i=1}^n \left[ \int_0^T \left( |\nabla \Phi(z(x_i, t), t)|^2 + \frac{1}{\alpha} (\Phi(z(x_i, t), t) - \bar{\Phi}(t))^2 \right) \frac{w_i(t)}{w_i(0)} dt \right],$$

where  $\bar{\Phi}(t)$  is approximated by (4.10).

The regularization term in (4.8) is:

$$J_{\text{R}} := \mathbf{R}_v = \mathbb{E}_{\rho_0(x)} \left[ \int_0^T |\nabla \Phi(z(x, t), t) w(x, t) - \nabla \Phi(x, 0) w(x, 0)|^2 \frac{w(x, t)}{w(x, 0)} dt \right] \\ \approx \frac{1}{n} \sum_{i=1}^n \left[ \int_0^T |\nabla \Phi(z(x_i, t), t) w_i(t) - \nabla \Phi(x_i, 0) w_i(0)|^2 \frac{w_i(t)}{w_i(0)} dt \right].$$

We parameterize  $\Phi$  with a neural network and use ADAM optimizer [24] to minimize the cost function in (4.8), which is  $J = J_{\text{KL}} + \gamma_1 J_{\text{SWFR}} + \gamma_2 J_{\text{R}}$  indeed. Then we formulate our deep learning framework for computing geodesics under spherical WFR in Algorithm 1.

---

### Algorithm 1 Framework for computing geodesics under spherical WFR

---

**Require:** particles  $\{x_i\}_{i=1}^n$  drawn from  $\rho_0$ , source parameter  $\alpha$ ,  $\log \rho_1$  up to a constant, time interval  $[0, T]$ , initializing network  $\Phi$ , hyper-parameters  $\gamma_1, \gamma_2$

- 1: **for** number of training iterations **do**
- 2: Use  $\Phi$  to solve the ODE system (4.9) to obtain position  $z(x_i, t)$  and weight  $w_i(t)$
- 3: Draw samples from  $\rho_1$  and use  $\Phi$  to solve the ODE system (4.11) to calculate  $\hat{\Phi}(t)$ .
- 4: Calculate cost function  $J = J_{\text{KL}} + \gamma_1 J_{\text{SWFR}} + \gamma_2 J_{\text{R}}$  using  $z(x_i, t)$ ,  $w_i(t)$  and  $\hat{\Phi}(t)$ .
- 5: Use ADAM optimizer to update network parameter of  $\Phi$
- 6: **end for**

---

In previous OT-Flow, one is supposed to solve (2.5) forward and then loss function (2.8) can be computed to optimize parameters. If  $N$  denotes the number of particles for simulation,  $2N$  ODE equations need to be solved in OT-Flow. The difference of computational cost between our algorithm and previous OT-Flow mainly dues to solving one more ODE in system (4.9) and the computation of  $\hat{\Phi}$  in inverse process after introducing weight change. The latter requires to solve ODE system (4.11) with samples from  $\rho_1$ . Under our framework,  $5N$  ODE equations need to be solved, which is 2.5 times more than the OT-Flow. In general, the computational cost of our framework is comparable to that of OT-Flow.

## 5 Using the Geodesics to Generate Weighted Samples

As we know, OT-Flow is developed based on previous CNF model, and can be viewed as a model using geodesics under Wasserstein distance between standard normal distribution and data distribution for sample generation. Similarly, by setting  $\rho_1$  in our framework as a given distribution which is easy to sample (standard normal distribution for instance), we can use the learned geodesics to build a new generative model for weighted samples. In particular, this new generative model can be applied to given weighted data samples, gaining an advantage over previous CNF models. We will call our model UOT-gen (short for “unbalanced OT-generation”) and present it in Algorithm 2.

---

### Algorithm 2 Framework of UOT-gen

---

**Require:** particles  $\{x_i\}_{i=1}^n$  with corresponding initial weights  $\{w_i\}_{i=1}^n$  drawn from  $\rho_0$ , source parameter  $\alpha$ ,  $\log \rho_1$  up to a constant, time interval  $[0, T]$ , initializing network  $\Phi$ , hyper-parameters  $\gamma_1, \gamma_2$

- 1: **for** number of training iterations **do**
- 2: Use  $\Phi$  to solve the ODE system (4.9) to obtain position  $z(x_i, t)$  and weight  $w_i(t)$ .
- 3: Draw samples from  $\rho_1$  and use  $\Phi$  to solve the ODE system (4.11) to calculate  $\hat{\Phi}(t)$ .
- 4: Calculate cost function  $J = J_{\text{KL}} + \gamma_1 J_{\text{SWFR}} + \gamma_2 J_{\text{R}}$  using  $z(x_i, t)$ ,  $w_i(t)$  and  $\hat{\Phi}(t)$ .
- 5: Use ADAM optimizer to update network parameter of  $\Phi$ .
- 6: **end for**
- 7: Draw equally weighted samples  $\{\hat{Z}_j\}_{j=1}^N$  from  $\rho_1$ .
- 8: Use well-trained  $\Phi$  to solve the ODE system (4.11) and get position  $\hat{x}_j(T)$  and weight  $\hat{w}_j(T)$ .  $\{\hat{x}_j(T)\}_{j=1}^N$  with weight  $\{\hat{w}_j(T)\}_{j=1}^N$  are the generated weighted samples satisfying  $\rho_0$ .

---

The above generative model is particularly suited in the Bayesian framework, where the objective is to infer and sample the posterior distribution of unknown parameter  $\theta \in \mathbb{R}^d$  given some observed data  $\mathcal{D}$ . By Bayes' theorem, the target distribution (i.e. the posterior distribution) is given by

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})}, \tag{5.1}$$

through which one can draw samples and do density estimation for  $\theta$ .  $p(\mathcal{D}|\theta)$  is the law of data given the parameters, or so-called likelihood of  $\theta$ ,  $p(\theta)$  is the prior distribution, and  $Z := p(\mathcal{D})$  is the normalization constant or the so-called evidence. In general, the normalization constant,

$$Z = \int p(\mathcal{D}|\theta)p(\theta)d\theta$$

is an intractable integral and one can only evaluate the numerator of (5.1). Often times calculating  $p(\mathcal{D}|\theta)$  is costly. For instance in reinforcement learning, estimating  $p(\mathcal{D}|\theta)$  requires running strategy, which can involve massive calculation.

We can sample from prior  $p(\theta)$  with normalized  $p(\mathcal{D}|\theta)$  as the weights to generate new weighted samples satisfying posterior, i.e. we take  $p$  in (3.9) as prior and  $w(\theta, 0)$  as normalized  $p(\mathcal{D}|\theta)$ :

$$\rho_0(\theta) = w(\theta, 0)p(\theta). \tag{5.2}$$

We set  $\rho_0$  as weighted data mentioned above and  $\rho_1$  as standard normal distribution. By training our UOT-gen model we can draw new weighted samples from posterior.

In most cases, the observation is continued and sequential. We are supposed to update the posterior and draw new posterior samples after obtaining new data. We would like to use previous estimation to help updating. An online algorithm can be developed based on UOT-gen model. If new data are observed, we can use generated samples with weight from last well-trained UOT-gen as prior and calculate corresponding likelihood. Then we multiply weight in generated samples with likelihood and normalized them as new weight. The last generated samples with new weight should satisfy updated posterior, which can be used for training to update parameters of UOT-gen model.

Suppose we have already trained an UOT-gen model for the latest posterior based on given observations. If new sequential data are observed, then the online algorithm can be as following in Algorithm 3:

---

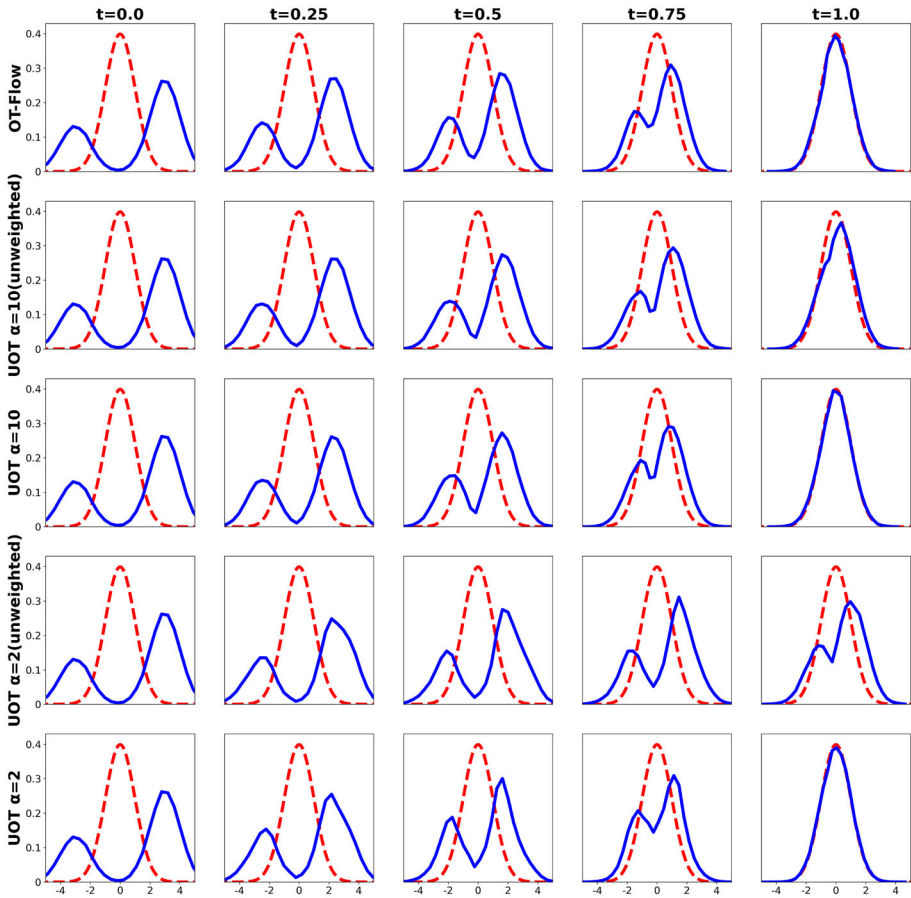
**Algorithm 3** Online algorithm for sequential data

---

**Require:** a well-trained UOT-gen model based on observed data so far

- 1: **if** new observations during a time period  $m\Delta t$  are made **then**
  - 2: drawing weighted samples from well-trained model: position  $\theta_i$  and weight  $w_i$
  - 3: Calculate corresponding likelihood  $p(\mathcal{D}|\theta_i)$  during new observation time period
  - 4: Normalize  $p(\mathcal{D}|\theta_i)w_i$  as new weight  $\bar{w}_i$
  - 5: Using  $\theta_i$  and  $\bar{w}_i$  to train UOT-gen model with Algorithm 1. The initial values of parameters in UOT-gen model can be set as the latest well-trained ones before observations are made.
  - 6: **end if**
- 

One advantage of online algorithm is that we do not need to calculate new likelihood by accumulating the whole time process. We just need to calculate the likelihood in the new time period of observation and multiply it with generated sample's weight. Meanwhile since



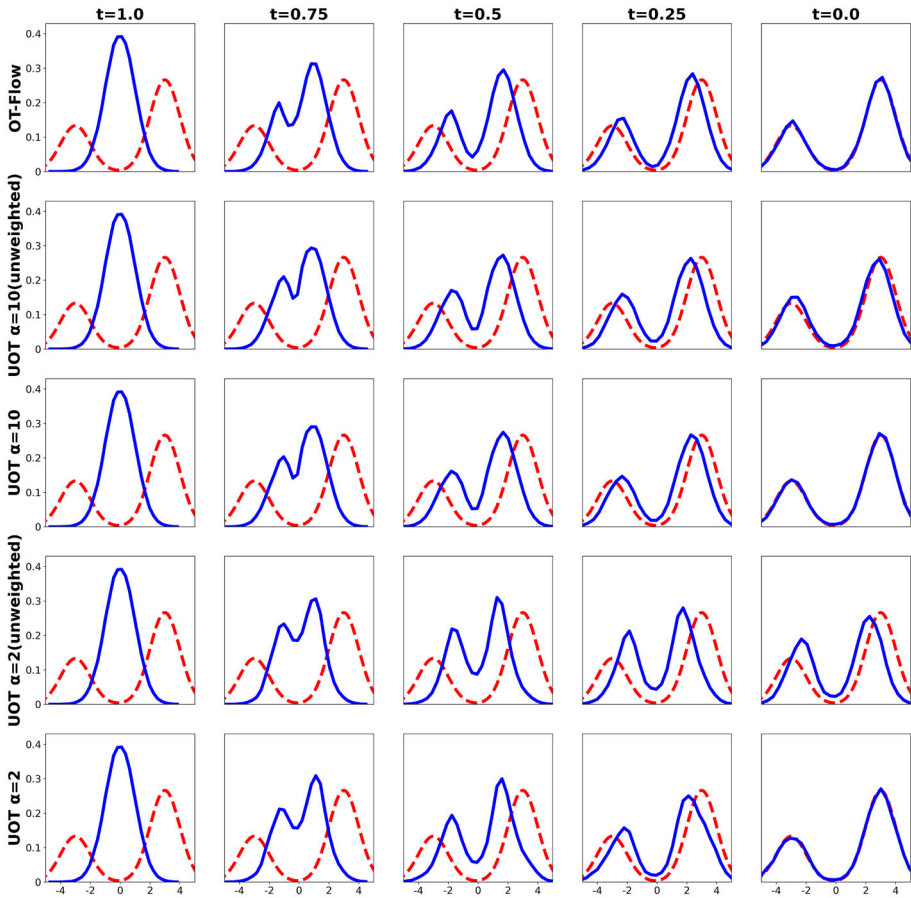
**Fig. 1** Evolution of particle distribution in forward transform with different  $\alpha$ . The first row reproduces results of OT-Flow model. The 2nd and 3rd rows show the particle distributions without weight (only transport is considered) and with weight respectively when  $\alpha = 10$ . The 4th and 5th rows show the results with  $\alpha = 2$  and other settings are the same as the 2nd and 3rd rows

the latest posterior is not far away from the previous one, we can set the initial values of parameters in UOT-gen model as last well-trained ones. Thus the new UOT-gen model can be trained with ease.

### 6 Numerical Experiments

In this section, we consider some test examples to demonstrate our deep learning framework and UOT-gen model. Our code is available at <https://github.com/sharkjingyang/WFR>.

We first test our method on a 1-D Gaussian mixture toy example to illustrate features of the geodesics under spherical WFR metric. The result also shows the effectiveness of the method in the sense that the particle system converges to the expected distribution precisely. We then perform density estimation on several two dimensional toy problems as done in [40]. We also apply the UOT-gen model to Bayesian problem to generate weighted samples



**Fig. 2** Evolution of particle distribution in inverse transform with different  $\alpha$ . Parameters and other settings are the same as in Fig. 1

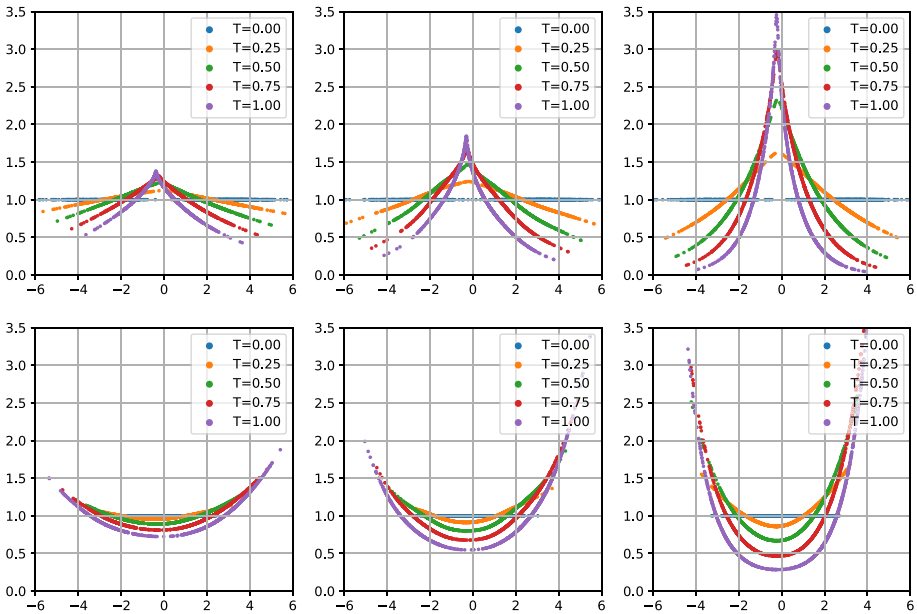
satisfying posterior. Meanwhile, we test the online Algorithm 3 on the same Bayesian problem with sequential data, in which one is required to update our UOT-gen iteratively to adjust posterior estimation after obtaining new observations. Our UOT-gen and online algorithm achieve competitive prediction accuracy in numerical experiments.

### Network Architecture

We adopt the same network architecture as in [40]. The potential function  $\Phi$  is parameterized as

$$\Phi(s; \theta) = \omega^\top N(s; \theta_N) + \frac{1}{2} s^\top (A^\top A) s + b^\top s + c,$$

where  $\theta = (\omega, \theta_N, A, b, c)$ ,  $s = (x, t) \in \mathbb{R}^{d+1}$  are the input features corresponding to space-time.  $N(s; \theta_N) : \mathbb{R}^{d+1} \rightarrow \mathbb{R}^m$  is a Residual Neural Network (ResNet) [20].  $\theta$  consists of all the trainable weights:  $\omega \in \mathbb{R}^m, \theta_N \in \mathbb{R}^p, A \in \mathbb{R}^{r \times (d+1)}, b \in \mathbb{R}^{d+1}, c \in \mathbb{R}$ . In our



**Fig. 3** Evolution of particle weights with different  $\alpha = 10, 5, 2$  from left to right. In all figures, x-axis indicates the current position of particles and y-axis represents the weight of particles. The first and second row compare the weight change over time in forward and inverse flow respectively

experiments, we set  $r = d$ .  $A, b$  and  $c$  model quadratic potentials (linear dynamics) whereas  $N$  models the nonlinear dynamics [47].

### ResNet

For the nonlinear  $N(s; \theta_N)$ , we adopt a simple two layer ResNet in our experiments:

$$u_0 = \sigma(K_0s + b_0),$$

$$N(s; \theta_N) = u_1 = u_0 + \sigma(K_1u_0 + b_1).$$

Here,  $K_0 \in \mathbb{R}^{m \times (d+1)}$ ,  $K_1 \in \mathbb{R}^{m \times m}$ ,  $b_0, b_1 \in \mathbb{R}^m$ . We select the element-wise activation function  $\sigma(x) = \log(\exp(x) + \exp(-x))$  as in [40], which is the antiderivative of the hyperbolic tangent, i.e.,  $\sigma'(x) = \tanh(x)$ . Therefore, hyperbolic tangent is the activation function of the flow  $\nabla\Phi$ .

Also, we can compute the  $\nabla_s \Phi(s; \theta)$  and  $\text{tr}(\nabla^2 \Phi(s; \theta))$  explicitly using the methods in [40]. For the forward propagation of flow, we use Runge-Kutta 4 with equidistant time steps to solve (2.5) and time integrals in cost function  $J$ .

### Hyper-Parameters

In all experiments, we consider the time interval as  $[0, 1]$ , i.e.  $T = 1$ . The total time step is set as 8. The width of neural network  $m$  is 32 in the experiments. We take  $\gamma_1 = \gamma_2 = 0.01$  as a default choice to make a balance for terms in the cost function.

### Metrics

To evaluate the goodness of approximating  $\rho_0$ , we follow the techniques applied to generative models as in [40]. We compare the known samples  $X = \{x_i\}_{i=1}^N$  to the generated synthetic samples  $Q = \{q_i\}_{i=1}^M$  via maximum mean discrepancy (MMD) [18, 32, 52]

$$\text{MMD}(X, Q) = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N k(x_i, x_j) + \frac{1}{M^2} \sum_{i=1}^M \sum_{j=1}^M k(q_i, q_j) - \frac{2}{NM} \sum_{i=1}^N \sum_{j=1}^M k(x_i, q_j), \tag{6.1}$$

where Gaussian kernel  $k(x_i, q_j) = \exp\left(-\frac{1}{2} |x_i - q_j|^2\right)$ . To adapt to the case of weighted generated samples  $Q = \{(w_i, q_i)\}_{i=1}^M$ , we use the following modified version of MMD

$$\text{MMD}_w(X, Q) = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N k(x_i, x_j) + \sum_{i=1}^M \sum_{j=1}^M w_i w_j k(q_i, q_j) - \frac{2}{N} \sum_{i=1}^N \sum_{j=1}^M w_j k(x_i, q_j). \tag{6.2}$$

MMD tests the difference between two distributions on the basis of samples drawn from each ( $X$  and  $Q$ ). A low MMD value means that the two sets of samples are likely to have been drawn from the same distribution [18]. Moreover, MMD provides an external and impartial metric to evaluate our model since it is not used in the training.

### 6.1 1-D Gaussian Mixture

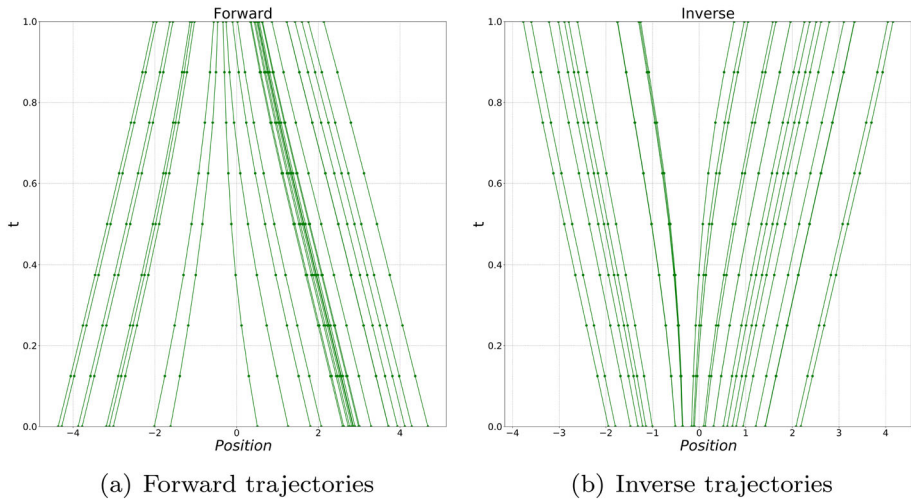
As a first example, we use the 1-D Gaussian mixture problem for our deep learning framework. We sample from the Gaussian mixture

$$\rho_0(x) = \frac{1}{3} \cdot \frac{1}{\sqrt{2\pi}} e^{-(x+3)^2/2} + \frac{2}{3} \cdot \frac{1}{\sqrt{2\pi}} e^{-(x-3)^2/2}, \tag{6.3}$$

and use our framework to learn the geodesics from  $\rho_0$  to standard normal distribution  $\rho_1$ .

Figures 1 and 2 illustrate the evolution of particle distribution in the geodesics for forward and inverse transformations respectively, against different parameter  $\alpha$ . In all figures, red dash curves indicate target density function whereas blue curves are distributions of particles at each time  $t$ . By Fig. 1, our deep learning framework shows compelling results with  $\alpha = 10, 2$  (the 3<sup>rd</sup> and 5<sup>th</sup> rows) compared to OT-Flow (the 1<sup>st</sup> row) in the sense that our framework captures the target distribution  $\rho_1$  precisely. The figures in 2<sup>nd</sup> and 4<sup>th</sup> rows show the particle distribution evolution without weight change. The differences between the 2<sup>nd</sup> and 3<sup>rd</sup> rows (as well as the 4<sup>th</sup> and 5<sup>th</sup> rows) show the effect brought by weight change. Compared with OT-Flow, the transportation of particles in our UOT-gen is weaker. The particles seem to be “lazier” for that they move to the location close to target distribution and the weight change compensate the remained. Compared to  $\alpha = 10$ , the result with  $\alpha = 2$  shows a stronger birth-death effect (weight change) whereas weaker transport effect, which is consistent with the guiding PDE (3.6). Figure 2 shows the results of inverse transformation (starting with  $\rho_1$ ) with other settings being the same as in Fig. 1, which validates the efficiency of sample generation of our model.

Then, we plot the weight change over time with different  $\alpha$  in forward and inverse transformation respectively in Fig. 3. The different weight changing trends between particles verify the property of unbalanced optimal transport. Especially the particles in the region



**Fig. 4** Trajectories of 30 random particles in our framework. The x-axis indicates the position of particles and y-axis indicates the time interval. **a** initial positions are drawn from  $\rho_0$ . **b** initial positions are drawn from  $\rho_1$

corresponding to high target density have greater weight, in the sense that the weight change in UOT can play similar role of transport in the whole transformation. We also observe that as  $\alpha$  decreases, the mass creation/destruction effect becomes more evident which agrees with (3.6). To see the particle trajectories more clearly, we randomly choose each 30 particles from  $\rho_0$  and  $\rho_1$  to plot their trajectories for forward and inverse transform in Fig. 4. The slope of trajectories can roughly indicate the velocity change, which is consistent with the relationship between particle velocity and weight (3.12). Particles with increasing weight slow down when being about to reach their destinations. The velocity of particles with decreasing weight increases, which indicates that they tend to leave present location.

Summarily, unbalanced optimal transport introduce a more general particle transformation involved both location and weight. The coefficient  $\alpha$  controls the effect of mass creation/destruction, by adjusting which we can decide how weight change influences the whole transformation.

### The Effects of $\gamma_1$ and $\alpha$ on $J_{SWFR}$

In this part we set  $\gamma_2 = 0$  for convenience to see how  $J_{SWFR}$  varies against different  $\gamma_1$  and  $\alpha$ .

We fix  $\alpha = 10$  and draw 10,000 random particles following the distribution  $\rho_0$  as training set. Figure 5a shows  $J_{SWFR}$  against different  $\gamma_1$ . We observe that as  $\gamma_1$  goes to zero,  $J_{SWFR}$  increases and converges to some limit which agrees with Proposition 4.1.

Furthermore, Proposition 4.1 suggests that if we take a sufficiently small  $\gamma_1$ , the model can give a good approximation to the geodesics under spherical WFR metric. In this small  $\gamma_1$  regime, as training process goes on, one can expect  $D_{KL}[\rho_0(x) \|\tilde{\rho}(x, T)] \approx 0$  and  $J_{SWFR} \approx d_{SWFR, \alpha}^2(\rho_0, \rho_1)$ . In Fig. 5b, we compare  $J_{SWFR}$  against different  $\alpha$  with  $\gamma_1 = 0.01$  fixed when training reaches the equilibrium state. The experiment is repeated independently for 5 times. We can find that as  $\alpha$  increases,  $J_{SWFR}$  has a slower increasing rate, which is expected to converge to  $W_2^2(\rho_0, \rho_1)$  according to (3.6). To make a comparison, we use the same training

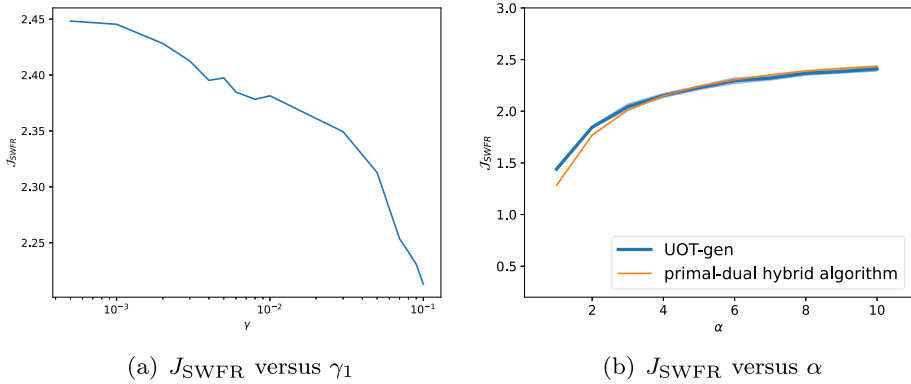


Fig. 5  $J_{SWFR}$  with different  $\gamma_1$  or  $\alpha$

set to compute  $\mathbb{E}_{\rho_0} L(x, T)$  in (2.8) for OT-Flow model, which turns out to be 2.51 and larger than  $J_{SWFR}$  in our framework. We also compare our results with a traditional optimization method, as shown in Fig. 5b. We employ a similar discrete scheme and a primal-dual hybrid algorithm described in [31] to solve the optimization problem (3.4). Our UOT-gen algorithm agrees with the  $\Gamma$ -convergence results solved by the traditional optimization algorithm. The details of numerical implementation of the primal-dual algorithm can be found in Appendix A.

### 6.2 2-D Toy Problems

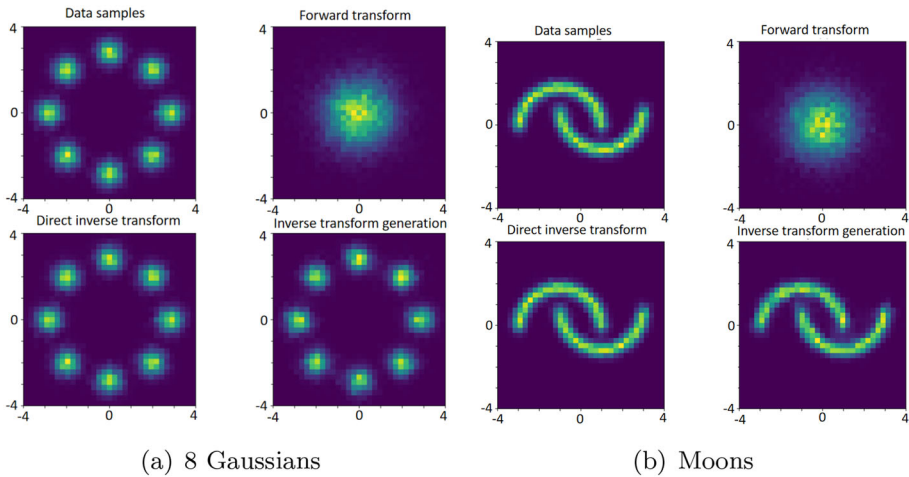
In this section, we demonstrate the accuracy and generation quality of our model on two 2-D toy problems also used in [40]. We use trained  $\Phi$  to simulate an inverse transformation (sampling from 2-D standard normal distribution and pushing particles back to the data distribution), from which one can compare the similarity of original data with generative distribution. The high similarity indicates that our model can generate weighted samples to approximate  $\rho_0$  with satisfactory accuracy even though  $\rho_0$  has separate supports. Numerical results in type of distribution heatmaps are illustrated in Fig. 6. The transformation we mentioned refers to considering both location and weight change of particles.

We also use MMD metric to evaluate the performance of UOT-gen and OT-Flow [40] on 1-D Gaussian mixture problem in Table 1 and 2-D toy problems in Table 2. In the experiments, we use 1000 samples for training and generate 1000 samples for testing. Forward MMD is computed between forward samples along the flow and samples from standard Gaussian distribution. Inverse MMD is computed between the generated samples and samples from target distribution  $\rho_0$ . We use (6.1) for OT-Flow and (6.2) for UOT-gen model.

### 6.3 Generating Weighted Samples from Bayesian Posterior

We apply our UOT-gen model to a Bayesian problem [57]. Consider Bernoulli equation:

$$\frac{dv}{dt} - v = -v^3, \quad v(0) = x, \tag{6.4}$$



**Fig. 6** Performance heatmaps for 2-D toy problems. In each example: (1) top left: samples from the unknown distribution  $\rho_0$  (2) top right: the forward transformation of particles from  $\rho_0$  (3) bottom left: direct inverse transformation of weighted particles in top right figure (i.e. samples generated by inverse transformation using samples in (2)). (4) bottom right: samples generated by inverse transformation from standard normal distribution

**Table 1** Comparison of UOT-gen model and OT-Flow in 1-D Gaussian mixture problem

Dataset	Model	Training time/itr (s)	Forward MMD	Inverse MMD
1d Gaussian mixture	UOT-gen	0.297	1.3e−3	2.8e−3
	OT-Flow	0.133	3.3e−3	6.1e−3

**Table 2** Comparison of UOT-gen model and OT-Flow one 2-D toy problems

Dataset	Model	Training time/itr (s)	Forward MMD	Inverse MMD
8 Gaussians	UOT-gen	0.274	2.1e−3	2.9e−3
	OT-Flow	0.123	1.3e−3	2.5e−3
Moons	UOT-gen	0.289	1.9e−3	1.7e−3
	OT-Flow	0.131	1.9e−3	4.3e−3

whose analytic solution is given as follows:

$$v(t) = G(x, t) = x (x^2 + (1 - x^2) e^{-2t})^{-1/2}. \tag{6.5}$$

Suppose we can collect the observations of  $v(t)$  at different time  $t = n\Delta t, n = 1, 2, \dots, T$ . Assume the observation will introduce a zero-mean Gaussian noise with standard deviation  $\sigma$ . We are supposed to estimate the initial position from the sequential observed data. This model is a typical problem for data assimilation methods since it exhibits certain non-Gaussian behavior [2]. In our experiment we set  $T = 50, \Delta t = 1, \sigma = 0.4$ . The ground truth initial position is  $x = 0.2$ . First we show a sequence of observed data and analytic solution in Fig. 7.

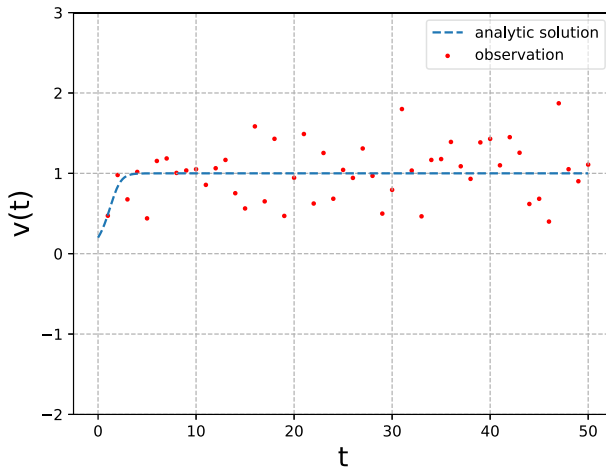


Fig. 7 Analytic solution and simulated observation for  $\sigma = 0.4$

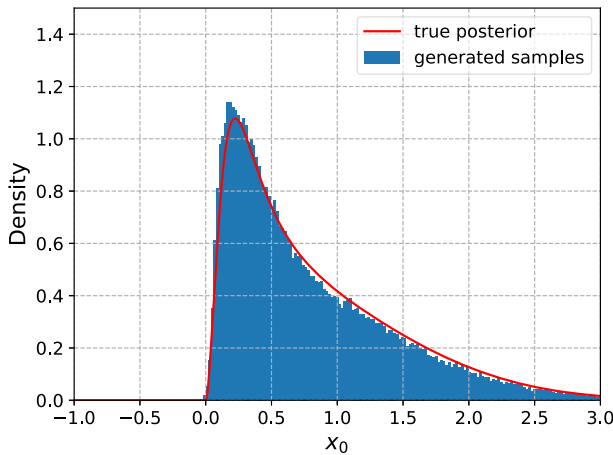


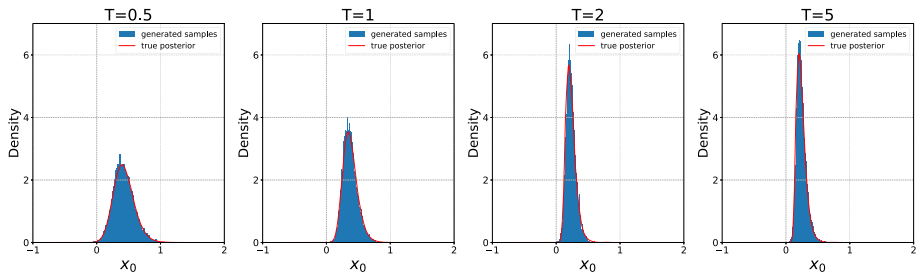
Fig. 8 Comparison of histogram of generated samples and true posterior

We can use Bayesian formula to get weighted samples satisfying posterior. Without loss of generality, we set prior  $p(w)$  as  $\mathcal{N}(0.5, 1)$ . Then we draw samples from prior and calculate corresponding likelihood  $p(\mathcal{D}|w)$ , which will be normalized as weights.  $D_t$  denotes the observation data at time  $t$ , then the exact  $p(\mathcal{D}|w)$  is

$$p(\mathcal{D}|x_0) = \prod_{t=1}^{50} \frac{1}{\sqrt{2\pi}\sigma^2} \exp\left(-\frac{(D_t - G(x_0, t))^2}{2\sigma^2}\right). \tag{6.6}$$

Then posterior is

$$p(x_0|\mathcal{D}) = \frac{1}{p(\mathcal{D})} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x_0 - 0.5)^2}{2}\right) \prod_{t=1}^{50} \frac{1}{\sqrt{2\pi}\sigma^2} \exp\left(-\frac{(D_t - G(x_0, t))^2}{2\sigma^2}\right). \tag{6.7}$$



**Fig. 9** Using online algorithm to generate weighted samples for Bernoulli example above

We plot the unnormalized posterior for the convenience for comparison with weighted samples generated by our UOT-gen model later. Our UOT-gen model is supposed to use given weighted samples to generative weighted sample satisfying posterior. We draw 2048 samples from the prior in our experiment. We set above 2048 samples with corresponding weights as discrete  $\rho_0, \mathcal{N}(0, 1)$  as  $\rho_1$  to train our UOT-gen model. Figure 8 shows the comparison of the new generated samples with true posterior. The histogram of new generated samples fit posterior well.

We also apply the online algorithm 3 on the above Bernoulli example. We set  $\Delta t = 0.1, m = 5$  with other settings unchanged, which means we collected observations in each  $5\Delta t$  and then update UOT-gen iteratively. Figure 9 shows the comparison of true posterior with generated samples from updated UOT-gen in four different time points. The well-fitted density estimations indicate that our online algorithm can generate weighted samples satisfying updated posterior after obtaining new observations.

### 6.4 High Dimensional Experiment

Computing expectations under the posterior is a common problem in Bayesian inference [41]. In general, consider the computation of analytically intractable integrals:  $\mathbb{E}_\pi[f(x)] = \int \pi(\eta) f(\eta) d\eta$ , where  $\pi$  is a posterior distribution known up to a normalizing constant. Importance sampling is used to converting it to an expectation under an auxiliary distribution. Mller, Thomas, et al. [39] proposed implementing the auxiliary distribution as a normalizing flow to generate samples for calculation:

$$\int \pi(\eta) f(\eta) d\eta = \int q(\eta) \frac{\pi(\eta)}{q(\eta)} f(\eta) d\eta = E_{q(\eta)} \left[ \frac{\pi(\eta)}{q(\eta)} f(\eta) \right] \approx \frac{1}{S} \sum_{s=1}^S \frac{\pi(\hat{\eta}_s)}{q(\hat{\eta}_s)} f(\eta_s),$$

where  $q(\eta)$  is a well-trained normalizing flow and  $\hat{\eta}_s$  is a sample from  $q(\eta)$ . However, we still need to compute  $\frac{\pi(\hat{\eta}_s)}{q(\hat{\eta}_s)}$  for the newly generated samples, which always involves many consecutive multiplications, resulting in a significant computational burden. In our UOT, the weight  $w_i$  is generated together with sample  $\eta_i$  in inference, thus they can be directly used to compute  $\int \pi(\eta) f(\eta) d\eta \approx \frac{1}{S} \sum_{i=1}^S w_i f(\eta_i)$ . The information of the weight has already been incorporated into the neural network. Compared with the inference process of using normalizing flows, we no longer need to calculate corresponding coefficients  $\frac{\pi(\hat{\eta}_s)}{q(\hat{\eta}_s)}$  for the newly generated samples, thus saves the computational cost in inference. We demonstrate it in our high dimensional experiment of Bayesian Inference.

**Table 3** Comparison of UOT-gen model and OT-flow for the Bayesian logistic regression

	Model	Training time/itr (s)	Inference time (s)	Accuracy (%)
Warm-up samples	UOT-gen	0.845	29.2	73.80
	OT-Flow	0.334	57.1	71.00
Random samples	UOT-gen	0.915	31.2	66.23
	OT-Flow	0.346	60.3	66.94

We apply our UOT-gen model to the Bayesian Logistic Regression problem in [36], which tests the binary Covertype dataset<sup>1</sup> with 581,012 data points and 54 features. Denote the observed data as  $\{x_t, y_t\}_{t=1}^N$ , where labels  $y_t \in \{-1, 1\}$  and features  $x_t \in \mathbb{R}^{54}$ . The regression coefficient is  $\theta$ , which satisfies the predicting model:

$$p(y_t = 1|x_t, \theta) = \frac{1}{1 + \exp(-\theta^T x_t)} =: P(x_t), \quad (6.8)$$

and we have:

$$p(\mathcal{D}|\theta) \propto \prod_{t=1}^N [P(x_t)]^{y_t} [1 - P(x_t)]^{1-y_t}. \quad (6.9)$$

We would like to use normalizing flow and UOT-gen to generate new  $\theta$  for inference. When applying continuous normalizing flows to the above problem, the coefficient  $\frac{\pi(\eta)}{q(\eta)}$  reduces to likelihood if we set the well-trained CNFs  $q(\eta)$  as prior.

We compare the performance on two training data sets. The first data set includes warm-up samples, whose distribution is not far away from the ground truth posterior. It simulates the situation that we have a good prior in Bayesian Inference. The warm-up samples can be obtained from some posterior traditional sampling algorithms [36, 37] (which can be computational expensive). The second data set contains some samples randomly generated from a trivial prior distribution, a multivariate normal distribution. Table 3 shows the comparison between our UOT-gen and OT-Flow. The training time per iteration of UOT-gen is comparable to OT-Flow (2–3 times as estimated in Sec 4.4). When it comes to inference stage, our UOT-gen exhibits an advantage over OT-Flow, and this advantage will become more pronounced with larger data sizes due to the increased computational cost of likelihood estimation. UOT-gen also achieves comparable accuracy results in inference, which demonstrates the feasibility of the algorithm.

## 7 Conclusion and Discussion

In summary, we have developed a deep learning framework to compute the geodesics under the spherical WFR metric based on a Benamou–Brenier type dynamic formulation. A KL divergence term based on inverse mapping was introduced into cost function as a soft boundary constraint to tackle the problem arising from weight change. Also, we leveraged the relationship between particle velocity and weight to introduce a new regularization term into our model. Then we demonstrated that the learned geodesics can be used to generate weighted samples from some target distribution. Numerical results have shown the accuracy and effi-

<sup>1</sup> <https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html>.

ciency of our model, especially beneficial for applications with given weighted samples in Bayesian inference.

Our framework is promising in dealing with weighted data, which can be costly or even infeasible for previous flow models. Future topics include applying the UOT-gen model in other fields with weighted samples, such as general Bayesian inference tasks and new drug molecule design, where one may hope to keep the most already known structures.

**Author Contributions** All authors contributed equally.

**Funding** This work is partially supported by the National Key R&D Program of China Nos. 2020YFA0712000 and 2021YFA1002800. The work of L. Li was partially supported by Shanghai Municipal Science and Technology Major Project 2021SHZDZX0102, NSFC 12371400 and 12031013, and Shanghai Science and Technology Commission Grant No. 21JC1402900.

**Data Availability** The datasets generated during the current study are available from the corresponding author on reasonable request.

### Declarations

**Competing interests** The authors have no relevant financial or non-financial interests to disclose.

## Appendix A: Implementation Details for Solving SWFR Distance with Optimization Method

We can apply a primal-dual hybrid algorithm to compute the SWFR distance. The algorithm is also used to compute the earth mover’s distance in [31]. To begin with, we reformulate the optimization problem (3.4) as a convex form:

$$\min_{\rho, m, \xi} \left\{ \frac{1}{2} \int_0^1 \int_{\mathbb{R}^d} \left( \frac{|m|^2}{\rho} + \alpha \frac{\xi^2}{\rho} \right) dx dt, \partial_t \rho + \nabla \cdot m = \xi, \int_{\mathbb{R}^d} \xi dx = 0 \right\}. \quad (A.1)$$

Then we solve the corresponding min-max problem:

$$\begin{aligned} & \max_{\phi, \lambda} \min_{\rho, m, \xi} \int_0^1 \int_{\Omega} \left( \frac{|m|^2}{\rho} + \alpha \frac{\xi^2}{\rho} \right) dx dt \\ & + \int_0^1 \int_{\Omega} \phi(x, t) (\partial_t \rho + \nabla \cdot m - \xi) dx dt + \int_0^1 \lambda(t) \left( \int_{\Omega} \xi dx \right) dt. \end{aligned} \quad (A.2)$$

We consider the space domain  $\Omega = [0, 1]^d$  and the time domain  $[0, 1]$  for simplicity. Let  $\Omega_h$  be the discrete space mesh-grid of  $\Omega$  with step size  $h$ , i.e.  $\Omega_h = \{0, h, 2h, \dots, 1\}^d$ . The time domain  $[0, 1]$  is discretized with step size  $\Delta t$ . Let  $N_x = 1/h$  be the space grid size and  $N_t = 1/\Delta t$  be the time grid size. All optimization variables ( $\rho, \xi, m, \phi$  and  $\lambda$ ) are defined on the grid.

We employ the same discrete scheme for divergence operator and boundary conditions as [58]. We give some definitions on the discrete space  $\Omega_h$ :

$$\begin{aligned} \int_{\Omega_h} f(x) dx & := \sum_{x \in \Omega_h, x_i \neq 0} f(x) h^d, \quad \langle f, g \rangle_h := \int_{\Omega_h} f(x) g(x) dx, \\ \nabla_h \cdot m(x) & := \sum_{i=1}^d D_{h,i} m(x), \quad x \in \Omega_h, \end{aligned}$$

where  $D_{h,i}$  denotes discrete differential operator for  $i$ -th component in  $i$ -th dimension with step size  $h$ :

$$D_{h,i}m(x) = (m_i(x_1, \dots, x_i, \dots, x_d) - m_i(x_1, \dots, x_i - h, \dots, x_d)) / h, \quad h \leq x_i \leq 1.$$

Then the discretization of (A.2) can be written as:

$$\begin{aligned} \max_{\phi, \lambda} \min_{\rho, m, \xi} L(m, \xi, \rho, \phi, \lambda) &= \sum_{t=1}^{N_t} \left\{ \int_{\Omega_h} \left( \frac{|m_t|^2}{2\rho_t} + \alpha \frac{\xi_t^2}{2\rho_t} \right) dx + \left\langle \phi_t, \frac{\rho_{t+1} - \rho_t}{\Delta t} + \nabla_h \cdot m_t - \xi_t \right\rangle_h \right. \\ &\quad \left. + \lambda_t \left( \int_{\Omega_h} \xi_t dx \right) \right\}. \end{aligned} \tag{A.3}$$

From the discretized form, we can describe the sizes of the optimization variables individually. The sizes of discretized variables are:  $(N_t + 1) \times (N_x + 1)^d$  for  $\rho$ ,  $N_t \times (N_x + 1)^d \times d$  for  $m$ ,  $N_t \times (N_x)^d$  for  $\xi$  and  $\phi$ , and  $N_t$  for  $\lambda$ . The boundary conditions are given as  $\rho_1 = \mu$ ,  $\rho_{N_t+1} = \nu$  and  $m(x) = 0$  for all  $x \in \partial\Omega_h$ . Then the primal-dual hybrid algorithm gives as follows for variables with superscripts  $k$ :

$$\begin{aligned} (m_t^{k+1}, \xi_t^{k+1}, \rho_t^{k+1}) &= \arg \min_{m^*, \xi^*, \rho^*} L(m^*, \xi^*, \rho^*, \phi^k, \lambda^k) \\ &\quad + \frac{1}{2\mu} \left( \|m^* - m_t^k\|_2^2 + \alpha \|\xi^* - \xi_t^k\|_2^2 + \|\rho^* - \rho_t^k\|_2^2 \right), \\ \tilde{m}_t^{k+1} &= 2m_t^{k+1} - m_t^k, \quad \tilde{\xi}_t^{k+1} = 2\xi_t^{k+1} - \xi_t^k, \quad \tilde{\rho}_t^{k+1} = 2\rho_t^{k+1} - \rho_t^k, \\ (\phi_t^{k+1}, \lambda_t^{k+1}) &= \arg \max_{\phi^*, \lambda^*} L(\tilde{m}_t^{k+1}, \tilde{\xi}_t^{k+1}, \tilde{\rho}_t^{k+1}, \phi^*, \lambda^*) - \frac{1}{2\tau} \left( \|\phi^* - \phi_t^k\|_2^2 + \|\lambda^* - \lambda_t^k\|_2^2 \right). \end{aligned} \tag{A.4}$$

The first step of the algorithm is equivalent to solving the following system:

$$\begin{cases} m^* = \frac{\rho^*(m_t^k - \mu \operatorname{div}_h^* \phi_t^k)}{\rho^* + \mu}, \\ \xi^* = \frac{\rho^*(\xi_t^k + \frac{1}{\alpha} \mu (\phi_t^k - \lambda_t^k))}{\rho^* + \mu}, \\ -\frac{m^{*2}}{2\rho^{*2}} - \alpha \frac{\xi^{*2}}{2\rho^{*2}} + \frac{\phi_{t-1}^k - \phi_t^k}{\Delta t} + \frac{1}{\mu} (\rho^* - \rho_t^k) = 0, \end{cases} \tag{A.5}$$

where  $\operatorname{div}_h^*$  represents the conjugate operator of divergence operator. By the definition of conjugate operator, we have

$$\langle \nabla_h \cdot f, g \rangle_h = \langle f, \operatorname{div}_h^* g \rangle_h = \langle f, -\nabla_h g \rangle_h.$$

Thus  $\operatorname{div}_h^* = -\nabla_h$  and

$$\nabla_h u = (\partial_{h,1} u, \partial_{h,2} u, \dots, \partial_{h,d} u),$$

where each  $\partial_{h,i}$  denotes discrete differential operator in  $i$ -th dimension with step size  $h$ :

$$\partial_{h,i} u(x) = \begin{cases} (u(x_1, \dots, x_i + h, \dots, x_d) - u(x_1, \dots, x_i, \dots, x_d)) / h, & 0 \leq x_i < 1, \\ 0, & x_i = 1. \end{cases}$$

We also introduce a natural boundary condition  $\phi_0 = 0$ , which comes from deriving optimal value condition for updating  $\rho$ . Solving above system (A.5) requires us to solve the roots

for a third order polynomial, where  $\rho^*$  should be the largest real root. The third step of the algorithm is to update dual variables:

$$\begin{cases} \phi_t^{k+1} = \phi_t^k + \tau \left( \frac{\rho_t^{k+1} - \rho_t^k}{\Delta t} - \nabla \cdot \tilde{m}_t^{k+1} - \tilde{\xi}_t^{k+1} \right), \\ \lambda_t^{k+1} = \lambda_t^k + \tau \left( \int_{\Omega} \tilde{\xi}_t^{k+1} dx \right). \end{cases} \quad (\text{A.6})$$

## References

1. Ambrosio, L., Gigli, N., Savaré, G.: Gradient Flows: in Metric Spaces and in the Space of Probability Measures. Springer (2005)
2. Apte, A., Hairer, M., Stuart, A.M., Voss, J.: Sampling the Posterior: An Approach to Non-Gaussian Data Assimilation. *Physica D: Nonlinear Phenomena*, **230**(1–2), 50–64 (2007)
3. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International Conference on Machine Learning, pp. 214–223 (2017)
4. Braides, A.: Gamma-Convergence for Beginners, vol. 22. Clarendon Press (2002)
5. Brenier, Y., Vorotnikov, D.: On optimal transport of matrix-valued measures. *SIAM J. Math. Anal.* **52**(3), 2849–2873 (2020)
6. Chen, R.T.Q., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. *Adv. Neural Inf. Process. Syst.* **31** (2018)
7. Chizat, L., Peyré, G., Schmitzer, B., Vialard, F.-X.: An interpolating distance between optimal transport and Fisher–Rao metrics. *Found. Comput. Math.* **18**(1), 1–44 (2018)
8. Chizat, L., Peyré, G., Schmitzer, B., Vialard, F.-X.: Unbalanced optimal transport: dynamic and Kantorovich formulations. *J. Funct. Anal.* **274**(11), 3090–3123 (2018)
9. Chwialkowski, K., Strathmann, H., Gretton, A.: A kernel test of goodness of fit. In: International Conference on Machine Learning, pp. 2606–2615 (2016)
10. De Giorgi, E.: New Problems on Minimizing Movements. *Ennio de Giorgi: Selected Papers*, pp. 699–713 (1993)
11. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics (2019)
12. Evans, L.C.: An Introduction to Mathematical Optimal Control Theory Version 0.2. Lecture Notes available at <http://math.berkeley.edu/~evans/control.course.pdf> (1983)
13. Finlay, C., Jacobsen, J.-H., Nurbekyan, L., Oberman, A.: How to train your neural ODE: the world of Jacobian and kinetic regularization. In: International Conference on Machine Learning, pp. 3154–3164 (2020)
14. Galichon, A.: A survey of some recent applications of optimal transport methods to econometrics. *Econom. J.* **20**(2), C1–C11 (2017)
15. Galichon, A.: Optimal Transport Methods in Economics. Princeton University Press (2018)
16. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **27** (2014)
17. Gorham, J., Mackey, L.: Measuring sample quality with kernels. In: International Conference on Machine Learning, pp. 1292–1301 (2017)
18. Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. *J. Mach. Learn. Res.* **13**(1), 723–773 (2012)
19. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved Training of Wasserstein GANs. *Adv. Neural Inf. Process. Syst.* **30** (2017)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
21. Hu, T., Chen, Z., Sun, H., Bai, J., Ye, M., Cheng, G.: Stein neural sampler. *arXiv preprint arXiv:1810.03545* (2018)
22. Johnson, R., Zhang, T.: A framework of composite functional gradient methods for generative adversarial models. *IEEE Trans. Pattern Anal. Mach. Intell.* **43**(1), 17–32 (2019)
23. Jordan, R., Kinderlehrer, D., Otto, F.: The variational formulation of the Fokker–Planck equation. *SIAM J. Math. Anal.* **29**(1), 1–17 (1998)

24. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations (2015)
25. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: International Conference on Learning Representations (2014)
26. Kingma, D.P., Welling, M., et al.: An introduction to variational autoencoders. *Found. Trends@ Mach. Learn.* **12**(4), 307–392 (2019)
27. Kondratyev, S., Monsaingeon, L., Vorotnikov, D.: A new optimal transport distance on the space of finite Radon measures. *Adv. Differ. Equ.* **21**(11/12), 1117–1164 (2016)
28. Kondratyev, S., Vorotnikov, D.: Spherical Hellinger–Kantorovich gradient flows. *SIAM J. Math. Anal.* **51**(3), 2053–2084 (2019)
29. Laschos, V., Mielke, A.: Geometric properties of cones with applications on the Hellinger–Kantorovich space, and a new distance on the space of probability measures. *J. Funct. Anal.* **276**(11), 3529–3576 (2019)
30. Li, W., Lee, W., Osher, S.: Computational mean-field information dynamics associated with reaction–diffusion equations. *J. Comput. Phys.*, p. 111409 (2022)
31. Li, W., Ryu, E.K., Osher, S., Yin, W., Gangbo, W.: A parallel method for Earth Mover’s distance. *J. Sci. Comput.* **75**(1), 182–197 (2018)
32. Li, Y., Swersky, K., Zemel, R.: Generative moment matching networks. In: International Conference on Machine Learning, pp. 1718–1727 (2015)
33. Liero, M., Mielke, A., Savaré, G.: Optimal entropy-transport problems and a new Hellinger–Kantorovich distance between positive measures. *Inventiones Math.* **211**(3), 969–1117 (2018)
34. Liu, Q.: Stein variational gradient descent as gradient flow. *Adv. Neural Inf. Process. Syst.* **30** (2017)
35. Liu, Q., Lee, J., Jordan, M.: A kernelized Stein discrepancy for goodness-of-fit tests. In: International Conference on Machine Learning, pp. 276–284 (2016)
36. Liu, Q., Wang, D.: Stein variational gradient descent: a general purpose Bayesian inference algorithm. *Adv. Neural Inf. Process. Syst.* **29** (2016)
37. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21**(6), 1087–1092 (1953)
38. Monge, G.: Mémoire sur la théorie des déblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.*, pp. 666–704 (1781)
39. Müller, T., McWilliams, B., Rousselle, F., Gross, M., Novák, J.: Neural importance sampling. *ACM Trans. Graphics (ToG)* **38**(5), 1–19 (2019)
40. Onken, D., Fung, S.W., Li, X., Ruthotto, L.: OT-Flow: Fast and accurate continuous normalizing flows via optimal transport. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 9223–9232 (2021)
41. Papamakarios, G., Nalisnick, E., Rezende, D.J., Mohamed, S., Lakshminarayanan, B.: Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.* **22**(1), 2617–2680 (2021)
42. Pele, O., Werman, M.: A linear time histogram metric for improved sift matching. In: European Conference on Computer Vision, pp. 495–508. Springer (2008)
43. Peyré, G., Cuturi, M.: Computational optimal transport: with applications to data science. *Found. Trends@ Mach. Learn.* **11**(5–6), 355–607 (2019)
44. Rezende, D., Mohamed, S.: Variational inference with normalizing flows. In: International Conference on Machine Learning, pp. 1530–1538 (2015)
45. Rubner, Y., Guibas, L.J., Tomasi, C.: The Earth Mover’s distance, multi-dimensional scaling, and color-based image retrieval. In: Proceedings of the ARPA Image Understanding Workshop, vol. 661, p. 668 (1997)
46. Rubner, Y., Tomasi, C., Guibas, L.J.: The Earth Mover’s distance as a metric for image retrieval. *Int. J. Comput. Vis.* **40**(2), 99–121 (2000)
47. Ruthotto, L., Osher, S.J., Li, W., Nurbekyan, L., Fung, S.W.: A machine learning framework for solving high-dimensional mean field game and mean field control problems. *Proc. Natl. Acad. Sci.* **117**(17), 9183–9193 (2020)
48. Salimans, T., Zhang, H., Radford, A., Metaxas, D.: Improving GANs using optimal transport. In: International Conference on Learning Representations (2018)
49. Santambrogio, F.: Optimal Transport for Applied Mathematicians. Birkäuser, NY **55**(58–63), 94 (2015)
50. Schiebinger, G., Shu, J., Tabaka, M., Cleary, B., Subramanian, V., Solomon, A., Gould, J., Liu, S., Lin, S., Berube, P., et al.: Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell* **176**(4), 928–943 (2019)
51. Tabak, E.G., Vanden-Eijnden, E.: Density estimation by dual ascent of the log-likelihood. *Commun. Math. Sci.* **8**(1), 217–233 (2010)

52. Theis, L., Oord, A.V.D., Bethge, M.: A note on the evaluation of generative models. In: International Conference on Learning Representations (2016)
53. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30** (2017)
54. Vidal, A., Wu Fung, S., Tenorio, L., Osher, S., Nurbekyan, L.: Taming hyperparameter tuning in continuous normalizing flows using the JKO scheme. *Sci. Rep.* **13**(1), 4501 (2023)
55. Villani, C.: *Optimal Transport: Old and New*, vol. 338. Springer (2009)
56. Wang, Z., Zhou, D., Yang, M., Zhang, Y., Rao, C., Wu, H.: Robust document distance with Wasserstein–Fisher–Rao metric. In: Asian Conference on Machine Learning, pp. 721–736 (2020)
57. Wu, J., Wen, L., Green, P.L., Li, J., Maskell, S.: Ensemble Kalman filter based sequential Monte Carlo sampler for sequential Bayesian inference. *Stat. Comput.* **32**(1), 1–14 (2022)
58. Xiong, Z., Li, L., Zhu, Y.-N., Zhang, X.: On the convergence of continuous and discrete unbalanced optimal transport models. *SIAM J. Numer. Anal.* To appear. arXiv preprint [arXiv:2303.17267](https://arxiv.org/abs/2303.17267) (2023)
59. Yang, K.D., Damodaran, K., Venkatachalapathy, S., Soylemezoglu, A.C., Shivashankar, G.V., Uhler, C.: Predicting cell lineages using autoencoders and optimal transport. *PLoS Comput. Biol.* **16**(4), e1007828 (2020)
60. Zhou, D., Chen, J., Wu, H., Yang, D., Qiu, L.: The Wasserstein–Fisher–Rao metric for waveform based earthquake location. *J. Comput. Math.* **41**(3), 417–438 (2023)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.