

# Learning High Dimensional Wasserstein Geodesics

**Shu Liu\***

Department of Mathematics  
Georgia Institute of Technology  
Atlanta, GA 30332, USA  
sliu459@gatech.edu

**Shaojun Ma\***

Department of Mathematics  
Georgia Institute of Technology  
Atlanta, GA 30332, USA  
shaojunma@gatech.edu

**Yongxin Chen**

Department of Aerospace Engineering  
Georgia Institute of Technology  
Atlanta, GA 30332, USA  
yongchen@gatech.edu

**Hongyuan Zha<sup>†</sup>**

School of Data Science  
Shenzhen Research Institute of Big Data  
The Chinese University of Hong Kong  
Shenzhen, Guangdong Province 518172, China  
zhahy@cuhk.edu.cn

**Haomin Zhou**

Department of Mathematics  
Georgia Institute of Technology  
Atlanta, GA 30332, USA  
hmzhou@math.gatech.edu

## Abstract

We propose a new formulation and learning strategy for computing the Wasserstein geodesic between two probability distributions in high dimensions. By applying the method of Lagrange multipliers to the dynamic formulation of the optimal transport (OT) problem, we derive a minimax problem whose saddle point is the Wasserstein geodesic. We then parametrize the functions by deep neural networks and design a sample based bidirectional learning algorithm for training. The trained networks enable sampling from the Wasserstein geodesic. As by-products, the algorithm also computes the Wasserstein distance and OT map between the marginal distributions. We demonstrate the performance of our algorithms through a series of experiments with both synthetic and real-world data.

## 1 Introduction

As a key concept of optimal transport (OT) (Villani, 2003), Wasserstein distance has been widely used to evaluate the distance between two distributions. Suppose we are given any two probability distributions  $\rho_a$  and  $\rho_b$  on  $\mathbb{R}^d$ , the Wasserstein distance between  $\rho_a$  and  $\rho_b$  is defined as

$$\inf_{\pi} \left\{ \int_{\mathbb{R}^d \times \mathbb{R}^d} c(x, y) d\pi(x, y) \mid \pi \in \Pi(\rho_a, \rho_b) \right\} \quad (1)$$

in the Kantorovich form. Here  $c(x, y)$  is the cost function that quantifies the effort of moving one unit of mass from location  $x$  to location  $y$ , and  $\Pi(\rho_a, \rho_b)$  is the set of joint distribution of  $\rho_a$  and  $\rho_b$ . The solution of OT refers to an optimal  $\pi^*$  to attain the Wasserstein distance between two distributions as well as an optimal map  $g$  such that  $g(x)$  and  $y$  have the same distribution when  $x \sim \rho_a$ .

\*Equal contribution

<sup>†</sup>The research of Hongyuan Zha is supported in part by a grant from Shenzhen Research Institute of Big Data

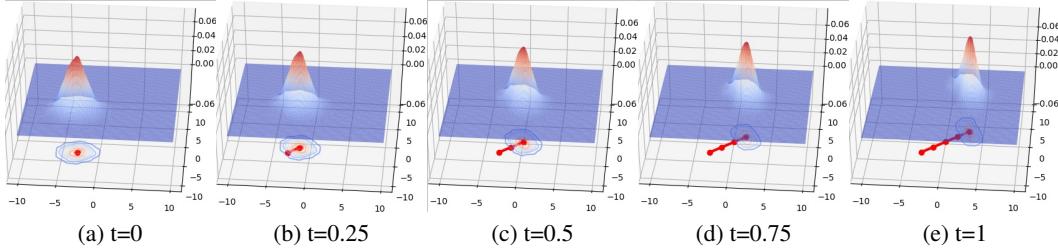


Figure 1: Wasserstein geodesic between two Gaussians

When applying OT-related computations to applications, the Kantorovich duality form has been widely studied and applied. For example, many regularized OT problems have been investigated, including entropic regularized OT (Cuturi, 2013; Seguy et al., 2017), Laplacian regularization (Flamary et al., 2014), Group-Lasso regularized OT (Courty et al., 2016), Tsallis regularized OT (Muzellec et al., 2017) and OT with  $L^2$  regularization (Dessein et al., 2018).

Based on the Kantorovich form, specially if choosing cost function as  $c(x, y) = \frac{1}{2}|x - y|^2$  which results in the well-known 2-Wasserstein distance, the OT problem can be rewritten in a fluid dynamics perspective (Benamou & Brenier, 2000a):  $W_2$  defined below is the square of 2-Wasserstein distance,

$$W_2(\mu, \nu) = \inf \left\{ \int_{\mathbb{R}^d} \int_0^1 \rho(x, t) |v(x, t)|^2 dx dt \right\},$$

subject to:  $\partial_t \rho + \nabla \cdot (\rho v) = 0, \quad \rho(\cdot, 0) = \rho_a, \quad \rho(\cdot, 1) = \rho_b.$

(2)

Solving (2) leads to the following partial differential equation (PDE) system:

$$\begin{aligned} \partial_t \rho + \nabla \cdot (\rho \nabla \Phi) &= 0, \quad \frac{\partial \Phi}{\partial t} + \frac{1}{2} |\nabla \Phi|^2 = 0, \\ \text{subject to: } \rho(\cdot, 0) &= \rho_a, \quad \rho(\cdot, 1) = \rho_b. \end{aligned}$$
(3)

The solution of Problem (2) or (3) gives the definition of Wasserstein *geodesic*: if there exists an optimal pushforward map  $\Xi^*$  between  $\rho_a$  and  $\rho_b$ , then the constant speed geodesic is defined as a continuous curve:  $\rho_t = (\text{Id} + t\Xi^*)_\# \rho_a$ <sup>3</sup>, which depicts the trajectory of  $x_0 \in \rho_a$  moving towards  $x_1 \in \rho_b$ . Here  $\text{Id}$  stands for the identity map.

Knowing Wasserstein geodesic between  $\rho_a$  and  $\rho_b$  (refer to Figure 1 as an example, trajectory of the distribution mean is indicated by a red line) provides ample information for their Wasserstein distance and optimal pushforward map. More importantly, since the Wasserstein geodesic is automatically kinetic-energy-minimizing, it offers a natural sampling mechanism without using additional artificial regularization to generate samples not only for the target distribution  $\rho_b$ , but also for all distributions along the Wasserstein geodesic. This is different from several recent OT based models for computing the optimal pushforward map, such as Jacobian and Kinetic regularized OT (Finlay et al., 2020b) and  $L^2$  regularized OT (Yang & Karniadakis, 2019).

Wasserstein geodesic also finds applications in robotics and optimal control communities. Krishnan & Martínez (2018) apply Brenier-Benamou OT to swarm control and updates the velocity of each agent. Inoue et al. (2020) study the locations of robots by minimizing the Wasserstein distance between original and target distributions. Chen et al. (2016, 2020) investigate diverse OT applications in control theories. Currently, the research that combines OT with robotics or control is still limited to low dimensions. We believe, having a method to compute the Wasserstein geodesic, especially in high dimensional setups, will be very beneficial for developing novel algorithms and applications in robotics and control, such as path planning for multi-agent systems.

Last but not least, finding an efficient method to compute the Wasserstein geodesic is important and challenging in applied mathematics as well. It is well-known that directly solving Problem (2) or (3) by the traditional numerical PDE methods, such as finite difference or finite element method which requires spatial discretization, must face the *curse of dimensionality*, meaning that the computational cost grows exponentially as the dimension increases.

<sup>3</sup>Please check Definition 1.

To this end, we first formulate the OT problem as a saddle point problem without introducing any regularizers. We also further reduce the search space for the saddle point problem by leveraging KKT conditions. We then parametrize the optimal maps as well as the Lagrange multipliers via deep neural networks, which can be trained alternately. The resulting method is a sample based algorithm that is capable of handling high dimensional Wasserstein geodesic. It is worth mentioning that there is no Lipschitz or convexity constraint on the neural networks in our computation. Those constraints become thorny issues when they can only be approximately enforced in methods for other Wasserstein related problems. Furthermore, our formulation is based on a Lagrangian framework which can be readily generalized to various alternative cost functions, including the  $L^p$ -Wasserstein distance. To the best of our knowledge, there is no method to compute high dimensional Wasserstein distance, optimal map as well as Wasserstein geodesic all at once for a general cost function. We validate our method through a series experiments on both synthetic and realistic data sets. To summarize, our contributions are:

- We develop a novel saddle point formulation so that high dimensional Wasserstein geodesic, optimal map as well as Wasserstein distance between two given distributions can be computed in one single framework.
- Our scheme is formulated to handle general convex cost functions, including the general  $L^p$ -Wasserstein distance. More importantly, it provides a method without requiring convexity or Lipschitz constraint.
- We show the effectiveness of our method through extensive numerical experiments with both synthetic and realistic data sets.

**Related work:** Traditional approaches (Benamou & Brenier, 2000a; Benamou et al., 2010; Li et al., 2018; Gangbo et al., 2019) for OT problems are designed to handle low dimensional settings but become infeasible in high dimensional cases since those methods are based on spatial discretizations. In machine learning community, researchers utilize OT to drive one distribution to another, and the Sinkhorn algorithm has been widely adopted (Altschuler et al., 2017; Genevay et al., 2018; Li et al., 2019; Xie et al., 2020) due to its convenient implementation even in high dimensional settings. However, the algorithm does not scale well to a large number of samples or can't readily handle continuous probability measures (Genevay et al., 2016).

To overcome the challenges of Sinkhorn algorithm, researchers bring neural networks to study OT problems. Seguy et al. (2017) study the regularized OT by applying neural networks such that large scale and high dimensional OT can be computed. Arjovsky et al. (2017) propose Wasserstein-1 GAN, which is a generative model to minimize Wasserstein distance. Though we have witnessed a great success of Wasserstein-1 GAN and its related work in high dimensional and large scale applications (Gulrajani et al., 2017; Tolstikhin et al., 2017; Miyato et al., 2018; Tong Lin et al., 2018; Dukler et al., 2019; Xie et al., 2019), the non-trivial Lipschitz-1 constraint of the discriminator is hard to be theoretically satisfied.

In order to avoid Lipschitz-1 constraint in Wasserstein-1 GAN, by using input convex neural networks (ICNN) (Amos et al., 2017) to approximate the potential function, Korotin et al. (2019) propose Wasserstein-2 GAN to rewrite OT as a minimization problem, Fan et al. (2020) and Makkavu et al. (2020) extend the semi-dual formulation of OT to new minimax problems.

Ruthotto et al. (2020) propose a machine learning framework for solving general Mean-Field Games in high dimensional spaces. Such method can be applied to solving Optimal Transport problems by adding penalty term to enforce the terminal constraint. Many other OT based approaches have been proposed to drive one distribution to another, including continuous normalizing flows (CNF) (Finlay et al., 2020a; Grathwohl et al., 2018) and the approaches developed by Trigila & Tabak (2016); Kuang & Tabak (2019).

Various OT models bring numerous applications in domain adaptation (Seguy et al., 2017), generative modeling (Arjovsky et al., 2017), partial differential equations (Ma et al., 2020), stochastic control (Yang, 2020), robotics (Krishnan & Martínez, 2018; Inoue et al., 2020), as well as color transfer (Korotin et al., 2019), which is also one of the experiments in this paper.

Although Wasserstein distance and optimal map have been widely studied within diverse frameworks, most of them focus on the cases that the cost function is either  $L^1$  or  $L^2$  based, and Wasserstein geodesic is seldom computed, especially in high dimensional settings. In this paper we compute all in our novel designed framework. We also note that a similar strategy formulated by Lin et al. (2020)

derives a saddle point optimization scheme for solving the mean field game equations. We should point out that our problem setting and sampling method are distinct from them.

## 2 Background of the Wasserstein Geodesic

We consider two probability distributions  $\rho_a, \rho_b$  defined on  $\mathbb{R}^d$ . For ease of discussion, we assume the density functions of these two distributions exist, then we focus on computing an interpolation curve  $\{\rho_t\}_{t=0}^1$  between  $\rho_a$  and  $\rho_b$ . To be more precise, we aim to find the length-minimizing curve joining  $\rho_a$  and  $\rho_b$ , which is also known as the *Wasserstein geodesic*. To this end, we start from the classical Optimal Transport (OT) problem (1), and assume the cost function  $c(x, y)$  is the optimal value of the following control problem,

$$c(x, y) = L(y - x) = \min_{\{v(\cdot, t)\}} \left\{ \int_0^1 L(v_t) dt \right\},$$

subject to:  $\dot{x}_t = v(x_t, t)$ ,  $x_0 = x$ ,  $x_1 = y$ .

Here we assume the Lagrangian  $L(\cdot) \in C^1(\mathbb{R}^d)$  satisfies  $L(-u) = L(u)$  for arbitrary  $u \in \mathbb{R}^d$  and is strictly convex and superlinear.<sup>4</sup> Then  $\nabla L : \mathbb{R}^d \rightarrow \mathbb{R}^d$  is an invertible map, we denote  $\nabla L^{-1}$  the inverse of  $\nabla L$  in the sequel. We define the Hamiltonian  $H(\cdot)$  associated with Lagrangian  $L(\cdot)$  as

$$H(p) = \max_{v \in \mathbb{R}^d} \{v \cdot p - L(v)\} = \nabla L^{-1}(p) \cdot p - L(\nabla L^{-1}(p)), \quad (4)$$

This is useful in our future discussion.

**Remark 1.** One important instance of  $L$  is  $L(v) = \frac{|v|^p}{p}$ ,  $p > 1$ , which leads to  $p$ -Wasserstein distance. When  $p = 2$ , it recovers the classical 2-Wasserstein distance.

Since (1) doesn't involve time, we denote it as **Static OT** problem and we denote its optimal value as  $W_{\text{Static}}(\rho_a, \rho_b)$ . Before introducing the dual problem of (1), we introduce the following definition.

**Definition 1.** Given measurable map  $T : \mathbb{R}^d \rightarrow \mathbb{R}^d$  and  $\lambda$  as a probability distribution on  $\mathbb{R}^d$ . We denote the pushforward of  $\lambda$  by  $T$  as  $T_\# \lambda$ , which is defined as

$$T_\# \lambda(E) = \lambda(T^{-1}(E)) \quad \text{for all measurable set } E \subset \mathbb{R}^d.$$

Problem (1) has the Kantorovich dual form (Villani, 2008)

$$\max_{\phi(y) - \psi(x) \leq c(x, y)} \left\{ \int \phi(y) \rho_b(y) dy - \int \psi(x) \rho_a(x) dx \right\}. \quad (5)$$

Here and in the sequel, we will denote  $\int$  as  $\int_{\mathbb{R}^d}$  for conciseness. One can show that the optimal value of (5) equals  $W_{\text{Static}}(\rho_a, \rho_b)$ . Let us denote the optimizer to (5) as  $(\phi^*, \psi^*)$ . Then  $\nabla \psi^*$  (as well as  $\nabla \phi^*$ ) provides optimal transport maps from  $\rho_a$  to  $\rho_b$  (as well as from  $\rho_b$  to  $\rho_a$ ) in the sense that

$$(\text{Id} + \nabla L^{-1}(\nabla \psi^*))_\# \rho_a = \rho_b, \quad (\text{Id} - \nabla L^{-1}(\nabla \phi^*))_\# \rho_b = \rho_a. \quad (6)$$

We then consider the dynamic version of (1)

$$W_{\text{Dym}}(\rho_a, \rho_b) = \min_{\rho, v} \left\{ \int_0^1 \int L(v(x, t)) \rho(x, t) dx dt \right\}, \quad (7)$$

$$\text{subject to: } \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0, \quad \rho(\cdot, 0) = \rho_a, \quad \rho(\cdot, 1) = \rho_b. \quad (8)$$

This problem also has an equivalent particle control version

$$\min_v \left\{ \int_0^1 \mathbb{E}[L(v(\mathbf{X}_t, t))] dt \right\}, \quad \text{subject to } \frac{d}{dt} \mathbf{X}_t = v(\mathbf{X}_t, t), \quad \mathbf{X}_0 \sim \rho_a, \quad \mathbf{X}_1 \sim \rho_b. \quad (9)$$

Such particle control version is more helpful for designing sample based formulation than its PDE counterpart, (7) as we will stated in Section 3.2. Since we introduce the dynamics of  $\{\rho_t\}$  and  $\{\mathbf{X}_t\}$

---

<sup>4</sup>  $L(\cdot)$  is super linear if  $\lim_{u \rightarrow \infty} \frac{L(u)}{|u|} = \infty$ .

into the new definition and reformulate the original problem (1) as an optimal control problem (7) and (9), we thus denote (7), (9) as **Dynamical OT** problem. The optimal solution of Dynamical OT is given by the following coupled PDE system, c.f. Chapter 13 of Villani (2008)

$$\frac{\partial \rho(x, t)}{\partial t} + \nabla \cdot (\rho(x, t) \nabla L^{-1}(\nabla \Phi(x, t))) = 0, \quad \frac{\partial \Phi(x, t)}{\partial t} + H(\nabla \Phi(x, t)) = 0, \quad (10)$$

subject to:  $\rho(\cdot, 0) = \rho_a, \rho(\cdot, 1) = \rho_b.$

We denote the solution to (10) as  $(\rho^*, \Phi^*)$ . Then the optimal vector field is

$$v^*(x, t) = \nabla L^{-1}(\nabla \Phi^*(x, t)). \quad (11)$$

From a geometric perspective, Problem (7) can be treated as a computing scheme for the geodesic on the probability manifold equipped with Wasserstein distance  $W_{\text{Dym}}(\cdot, \cdot)$ . Following this point of view, we also treat (7) as the problem for evaluating the **Wasserstein geodesic** joining  $\rho_a$  and  $\rho_b$ . Meanwhile, the PDE system (10) serves as the geodesic equation for the Wasserstein geodesic.

Static OT problems and Dynamical OT problems are closely related. We summarize some of their connections, which are useful for our future derivations in section 3.

Recall  $(\phi^*, \psi^*)$  as optimal solution to (5), and  $\Phi^*$  as solution of (10), then

$$\Phi^*(x, 0) = \psi^*(x) + C_0, \quad \Phi^*(x, 1) = \phi^*(x) + C_1. \quad (12)$$

Here  $C_0, C_1$  are constants. Furthermore, from (6) we have

$$(\text{Id} + \nabla L^{-1}(\nabla \Phi^*(\cdot, 0))) \sharp \rho_a = \rho_b, \quad (\text{Id} - \nabla L^{-1}(\nabla \Phi^*(\cdot, 1))) \sharp \rho_b = \rho_a. \quad (13)$$

In addition, Static and Dynamical OT produce the same distance, i.e.  $W_{\text{Static}}(\rho_a, \rho_b) = W_{\text{Dym}}(\rho_a, \rho_b)$ .

### 3 Proposed Methods

#### 3.1 Primal-Dual based saddle point scheme

In this section, we present an approach of solving **Dynamical OT** problem (7) by applying Lagrange Multiplier method. We introduce Lagrange Multiplier  $\Phi(x, t)$  for the PDE constraint  $\frac{\partial \rho(x, t)}{\partial t} + \nabla \cdot (\rho(x, t)v(x, t)) = 0$  and  $\Psi(x)$  for one of the boundary constraints  $\rho(\cdot, 1) = \rho_b$ . (The constraint  $\rho(\cdot, 0) = \rho_a$  can be naturally treated as the initial condition of  $\rho_t$ .) Then we consider the functional

$$\begin{aligned} \mathcal{L}(\rho, v, \Phi, \Psi) &= \int_0^1 \int L(v) \rho + \left( \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) \right) \Phi(x, t) dx dt + \int \Psi(x)(\rho(x, 1) - \rho_b(x)) dx \\ &= \int_0^1 \int \left( L(v) - \frac{\partial \Phi}{\partial t} - \nabla \Phi \cdot v \right) \rho(x, t) dx dt + \int \Phi(x, 1) \rho(x, 1) dx \\ &\quad - \int \Phi(x, 0) \rho_a(x) dx + \int \Psi(x)(\rho(x, 1) - \rho_b(x)) dx. \end{aligned} \quad (14)$$

For the second equality, we apply integration by parts on  $[0, 1]$  and use the initial condition  $\rho(\cdot, 0) = \rho_a$ . Solving the constrained optimization problem (7) is equivalent to investigating the following saddle point optimization problem

$$\min_{\rho, v} \max_{\Phi, \Psi} \mathcal{L}(\rho, v, \Phi, \Psi). \quad (15)$$

Problem (15) contains  $(\rho, v, \Phi, \Psi)$  as variables. We eliminate some of the variables by leveraging the Karush–Kuhn–Tucker (KKT) conditions (Kuhn & Tucker, 1951)

$$\frac{\partial \mathcal{L}}{\partial \Phi(x, t)} = 0, \quad \frac{\partial \mathcal{L}}{\partial \Psi(x)} = 0, \quad \frac{\partial \mathcal{L}}{\partial \rho(x, t)} = 0, \quad \frac{\partial \mathcal{L}}{\partial v(x, t)} = 0. \quad (16)$$

The first two conditions lead to the constraints (8). The third condition in (16) yields

$$-\frac{\partial \Phi}{\partial t} - (\nabla \Phi(x, t) \cdot v(x, t) - L(v(x, t))) = 0, \quad (17)$$

$$\Phi(x, 1) + \Psi(x) = 0. \quad (18)$$

The fourth condition in (16) yields  $\nabla L(v(x, t)) - \nabla \Phi(x, t) = 0$ , which can be rewritten as

$$v(x, t) = \nabla L^{-1}(\nabla \Phi(x, t)). \quad (19)$$

The KKT conditions (18) and (19) reveal explicit relations among  $v$ ,  $\Phi$  and  $\Psi$ , which can be incorporated in (15) by plugging  $\Psi(x) = -\Phi(x, 1)$  and  $v(x, t) = \nabla L^{-1}(\nabla \Phi(x, t))$  back into (14). Recall definition of  $H$  in (4), the first term of (14) then becomes  $-\left(\frac{\partial \Phi(x, t)}{\partial t} + H(\nabla \Phi(x, t))\right)$ . Our new optimization problem can be formulated as

$$\min_{\rho} \max_{\Phi} \int_0^1 \int -\left(\frac{\partial \Phi}{\partial t} + H(\nabla \Phi)\right) \rho(x, t) dx dt + \int \Phi(x, 1) \rho_b(x) - \Phi(x, 0) \rho_a(x) dx. \quad (20)$$

### 3.2 Simplification via geodesic pushforward map

In the saddle point problem (20), both variables  $\rho(\cdot, t)$  and  $\Phi(\cdot, t)$  are time-varying functions. It is a weak formulation of the OT problem that may accommodate general cost  $L$ . On the other hand, it requires to optimize over rather large space of time-varying functions, which may increase the computational cost as well as the chance of encountering local optima. To mitigate the challenge, we reduce the search space by leveraging the the following geodesic property of optimal transporting trajectory if  $L$  is strictly convex.

**Theorem 1.** Suppose  $\{\mathbf{X}_t^*\}_{t=0}^1$  is the trajectory obeying the optimal vector field of (9) with strictly convex Lagrangian  $L$ , then  $\frac{d^2 \mathbf{X}_t^*}{dt^2} = 0$ .

This result was discussed by Benamou & Brenier (2000b) for 2-Wasserstein case ( $L(\cdot) = \frac{|\cdot|^2}{2}$ ). The more general result was presented in Theorem 5.5 of Villani (2003). This theorem illustrates that, from the particle point of view, the optimal transport process can be treated as a pushforward operation along the geodesics (straight lines) in  $\mathbb{R}^d$ . To be more precise, we denote  $\{\rho^*(\cdot, t)\}_{t=0}^1$  as the optimal solution to (7). Denote  $\{v^*(\cdot, t)\}_{t=0}^1$  as the optimal vector field in (9). Then  $\{\mathbf{X}_t^*\}_{t=0}^1$  solves  $\frac{d}{dt} \mathbf{X}_t^* = v^*(\mathbf{X}_t, t)$ . Since  $\frac{d^2}{dt^2} \mathbf{X}_t^* = 0$ , this implies  $\mathbf{X}_t^* = \mathbf{X}_0^* + tv^*(\mathbf{X}_0^*, 0)$ ,  $t \in [0, 1]$ . Finally, due to the equivalence between (7) and (9), we are able to verify that  $\mathbf{X}_t^* \sim \rho^*(\cdot, t)$ , which yields  $\mathbf{X}_0^* + tv^*(\mathbf{X}_0^*, 0) \sim \rho^*(\cdot, t)$ . Since  $\mathbf{X}_0^*$  follows the distribution with density  $\rho_a$ , this leads to the following relation between optimal density  $\rho^*(\cdot, t)$  and optimal vector field  $v^*(\cdot, 0)$  at  $t = 0$

$$\rho^*(\cdot, t) = (\text{Id} + tv^*(\cdot, 0)) \# \rho_a. \quad (21)$$

Equation (21) justifies that the optimal  $\rho^*(\cdot, t)$  can be obtained by pushforwarding the initial distribution  $\rho_a$  along certain straight lines with initial direction  $v^*(\cdot, 0)$ . This observation motivates us to restrict the search space of  $\{\rho_t\}_{t=0}^1$  on the following space  $\mathcal{P}_{\text{restrict}}$

$$\{ \{\rho(\cdot, t)\}_{t=0}^1 \mid \rho(\cdot, t) = (\text{Id} + tF) \# \rho_a \text{ for } t \in [0, 1] \}.$$

Here  $F$  is an arbitrary vector field defined on  $\mathbb{R}^d$ . Combining the previous discussions together, we propose the scheme  $\min_{\rho \in \mathcal{P}_{\text{restrict}}} \max_{\Phi} \hat{\mathcal{L}}(\rho, \Phi)$ . In particular, since  $\rho$  is uniquely determined by  $F$ , we reformulate our scheme as

$$\min_F \max_{\Phi} \mathcal{L}(F, \Phi), \quad \mathcal{L}(F, \Phi) = \hat{\mathcal{L}}((\text{Id} + tF) \# \rho_a, \Phi). \quad (22)$$

We have the following theoretical property for scheme (22).

**Theorem 2.** Denote the solution (10) as  $\rho^*(x, t)$  and  $\Phi^*(x, t)$ , and set  $\Phi_0^*(\cdot) = \Phi^*(\cdot, 0)$ . Assume  $\Phi^*(\cdot, t) \in C^2(\mathbb{R}^d)$ , then  $(\nabla L^{-1}(\nabla \Phi_0^*), \Phi^*)$  is a critical point to the functional  $\mathcal{L}$ , i.e.

$$\frac{\partial \mathcal{L}}{\partial F}(\nabla L^{-1}(\nabla \Phi_0^*), \Phi^*) = 0, \quad \frac{\partial \mathcal{L}}{\partial \psi}(\nabla L^{-1}(\nabla \Phi_0^*), \Phi^*) = 0.$$

Furthermore,  $\mathcal{L}(\nabla L^{-1}(\nabla \Phi_0^*), \Phi^*) = W_{\text{Dym}}(\rho_a, \rho_b)$ .

Theorem 2 shows that the optimal solution of Dynamical OT is also a critical point of the functional used in our saddle point optimization scheme (22). The optimal value of (22) is exactly the optimal transport distance. The theorem is proved in the supplemental materials (Appendix B).

### 3.3 Bidirectional dynamical formulation

To improve the stability and avoid local traps in the training processing, we propose a *bidirectional* scheme by exploiting the symmetric status of  $\rho_a$  and  $\rho_b$  in (1). Let's consider two OT problems

$$\min_F \max_{\Phi_F} \mathcal{L}^{ab}(F, \Phi_F), \quad \min_G \max_{\Phi_G} \mathcal{L}^{ba}(G, \Phi_G),$$

where  $\mathcal{L}^{ab}$  is defined in (22), and  $\mathcal{L}^{ba}$  is defined by switching  $\rho_a$  and  $\rho_b$  in (22). Clearly, the first one is for  $W(\rho_a, \rho_b)$  while the second one is for  $W(\rho_b, \rho_a)$ . When reaching optima, the vector fields  $F$  and  $G$  are transport vectors in the opposite directions. At a specific point  $x \in \mathbb{R}^d$ , moving along straight line in the direction  $F$  ends up at  $x + F(x)$ . The direction of  $G$  at  $x + F(x)$  must point to the opposite direction of  $F(x)$ , which leads to  $G(x + F(x)) = -F(x)$ . Similarly, we also have  $F(x + G(x)) = -G(x)$ . Thus we introduce two constraints for  $F$  and  $G$

$$\begin{aligned} \mathcal{R}^{ab}(F, G) &= \int |G(x + F(x)) + F(x)|^2 \rho_a(x) dx, \\ \mathcal{R}^{ba}(F, G) &= \int |F(x + G(x)) + G(x)|^2 \rho_b(x) dx. \end{aligned}$$

Our final saddle-point problem becomes

$$\min_{F, G} \max_{\Phi_F, \Phi_G} \mathcal{L}^{ab}(F, \Phi_F) + \mathcal{L}^{ba}(G, \Phi_G) + \lambda(\mathcal{R}^{ab}(F, G) + \mathcal{R}^{ba}(F, G)), \quad (23)$$

where  $\lambda$  is a tunable coefficient of our constraint terms.

### 3.4 Overview of the algorithm

To solve (23), we propose an algorithm that is summarized in the following steps. Please also check its detailed discussions in Appendix A.

- **Preconditioning** We can apply preconditioning techniques to 2-Wasserstein cases in order to make our computation more efficient.
- **Main Algorithm** We set  $F_{\theta_1}, G_{\theta_2}$  and  $\Phi_{\omega_1}^F, \Phi_{\omega_2}^G$  as fully connected neural networks and optimize over their parameters  $\omega_1, \omega_2$  and  $\theta_1, \theta_2$  alternatively via stochastic gradient ascend and descend.
- **Stopping Criteria** When computed  $F$  (or  $G$ ) is close to the optimal solution, the Wasserstein distance  $W(\rho_a, \rho_b)$  (or  $W(\rho_b, \rho_a)$ ) can be approximated by

$$\widehat{W}^{ab} = \int L(F(x)) \rho_a(x) dx, \quad \widehat{W}^{ba} = \int L(G(x)) \rho_b(x) dx.$$

For a chosen threshold  $\epsilon > 0$ , we treat  $|\widehat{W}^{ab} - \widehat{W}^{ba}| < \epsilon$  as the stopping criteria of our algorithm.

## 4 Experiments

**Experiment Setup:** We test our algorithm through a series of synthetic data sets and compare our numerical results with the ground truths (only for the Gaussian cases) or with the computational methods introduced in the Python library (Python Optimal Transport (POT)) (Flamary et al., 2021). We also test our algorithm for realistic data sets including color transfer (Reinhard et al., 2001) and transportation between MNIST digits (Lecun et al., 1998).

For low dimensional cases, namely, 2 and 10 dimensional cases, we set  $\Phi_F, \Phi_G$  and  $F, G$  as fully connected neural networks, where  $\Phi_F, \Phi_G$  have 6 hidden layers and  $F$  and  $G$  have 5 hidden layers. Each layer has 48 nodes, the activation function is chosen as Tanh. For high dimensional cases, where we deal with MNIST handwritten digits data set, we adopt similar structures of neural networks, the only difference is that in each layer we extend the number of nodes from 48 to 512. In terms of training process, for all synthetic and realistic cases we use the Adam optimizer (Kingma & Ba, 2014) with learning rate  $10^{-4}$ . Notice that we are computing Wasserstein geodesic, namely, starting with an initial distribution  $\rho_{t_0}$ , in most cases we generate evolving distributions for next ten time steps, from  $t_1 = 0.1$  to  $t_{10} = 1$ . The cost functions are chosen as  $L(v) = |v|^{\frac{3}{2}}$  in Synthetic-2 and

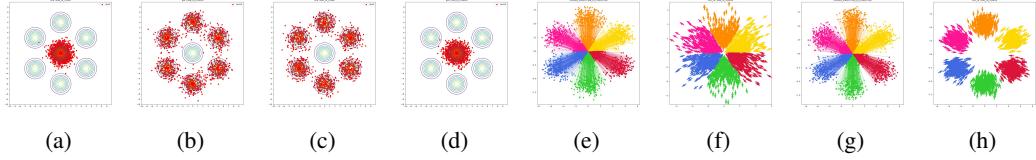


Figure 2: Syn-1. (a)(b) true  $\rho_a$  and generated  $\rho_b$ , (c)(d) true  $\rho_b$  and generated  $\rho_a$ , (e)(g) tracks of sample points from  $\rho_a(\rho_b)$  to  $\rho_b(\rho_a)$ , (f)(h) vector fields from  $\rho_a(\rho_b)$  to  $\rho_b(\rho_a)$ .

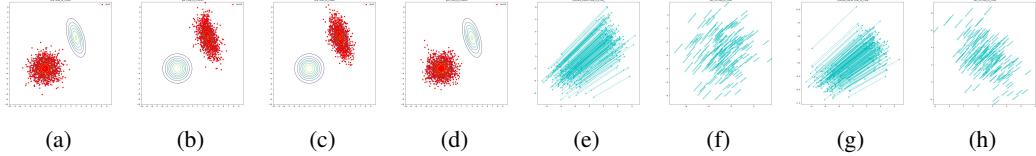


Figure 3: Syn-2. (a)(b) true  $\rho_a$  and generated  $\rho_b$ , (c)(d) true  $\rho_b$  and generated  $\rho_a$ , (e)(g) tracks of sample points from  $\rho_a(\rho_b)$  to  $\rho_b(\rho_a)$ , (f)(h) vector fields from  $\rho_a(\rho_b)$  to  $\rho_b(\rho_a)$ .

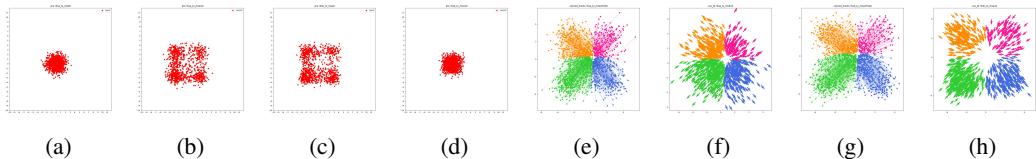


Figure 4: Syn-3. (a)(b) true  $\rho_a$  and generated  $\rho_b$ , (c)(d) true  $\rho_b$  and generated  $\rho_a$ , (e)(g) tracks of sample points from  $\rho_a(\rho_b)$  to  $\rho_b(\rho_a)$ , (f)(h) vector fields from  $\rho_a(\rho_b)$  to  $\rho_b(\rho_a)$ .

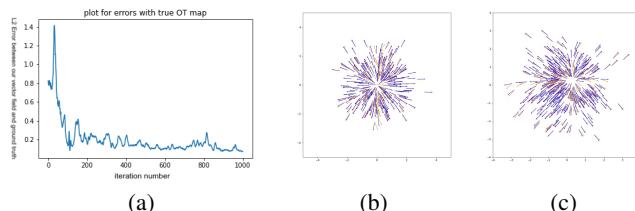


Figure 5: Left: Syn2:  $L^2(\rho_a)$  error between our computed  $F$  and the real OT map vs iteration number; Middle: Syn1: Plot of our computed  $F$  (blue) and the OT vector field computed by POT (orange); Right: Syn3: Plot of our computed  $F$  and the OT vector field computed by POT (orange).

$L(v) = |v|^2$  in all other tests. We only show the final state of the generated distribution due to space limitation, more experiments and details are included in the last part of Appendix.

**Synthetic-1:** This is a 2-dimensional case, we set  $\rho_a$  as a standard Gaussian distribution  $N(\mu_0, \Sigma_0)$  while  $\rho_b$  as six surrounding Gaussian distributions with the same  $\Sigma_0$ . In Figure 2, we show the generated distribution that follows  $\rho_b$  as well as the one that follows  $\rho_a$ . We also show the start-end tracks of points and vector field.

**Synthetic-2:** In this 5-dimensional case we treat  $\rho_a$  and  $\rho_b$  both as two Gaussian distributions. We show the results of two dimensional projection in Figure 3.

**Synthetic-3:** As a 10-dimensional case, here we set  $\rho_a$  as a standard Gaussian distribution and  $\rho_b$  as a special distribution where samples are unevenly distributed around four corners. We show the results of two dimensional projection in Figure 4.

**Training and Results:** For synthetic data sets, in the training process we set the batch size  $N_t = 2000$  and sample size for prediction  $N_p = 1000$ . From Figures 2, 3 and 4 we see that in all cases, within various dimensional settings, the generated samples closely follow the ground-truth distributions.

**Comparisons:** We now compare the numerical results listed in the Synthetic cases with either the ground truth (Synthetic-2) or with the results computed by POT (Synthetic-1, Synthetic-3) in Figure 5. Here all the examples are computed under quadratic cost  $L(v) = |v|^2/2$ .

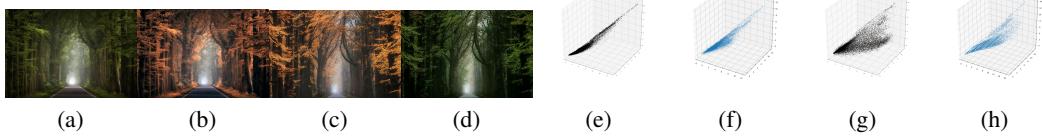


Figure 6: Real-1, Forest views. (a)(c) true summer(autumn) view, (b)(d) generated autumn(summer) view when true summer(autumn) view is given, (e)(g) true palette distribution of summer(autumn) view, (f)(h) generated palette distribution of summer(autumn) view.

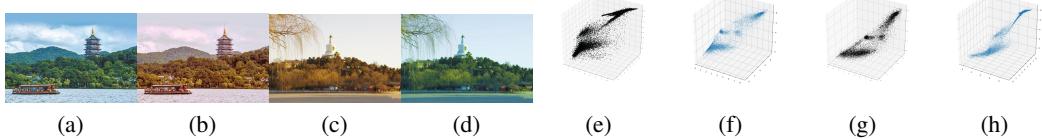


Figure 7: Real-2, Color transfer. (a)(b) true summer(generated autumn) view of the West Lake, (c)(d) true autumn(generated summer) view of the White Tower, (e)(f) palette distribution of the true summer West Lake(generated summer White Tower), (g)(h) palette distribution of the true autumn White Tower(generated autumn West Lake).

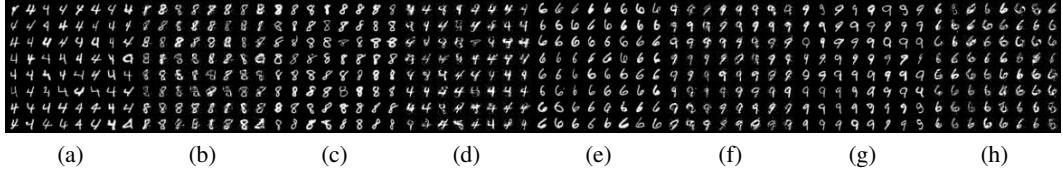


Figure 8: Real-3, Digits transformation. (a)(b) true(generated) digit 4(8), (c)(d) true(generated) digit 8(4), (e)(f) true(generated) digit 6(9), (g)(h)true(generated) digit 9(6).

**Realistic-1:** Two given pictures describe the summer view( $\rho_a$ ) and autumn view( $\rho_b$ ) of a forest. We generate the autumn view starting with the summer view and generate the summer view starting with the autumn view. We also show the ground-truth and generated palette distributions in Figure 6.

**Realistic-2:** In this case the view of West Lake in summer and the view of White Tower in autumn are given, then we aim to do a color transfer and simulate the summer view of White Tower and the autumn view of West Lake, the results are shown in Figure 7, the ground-truth and generated palette distributions are also included.

**Realistic-3:** We choose MNIST as our data set ( $28 \times 28$  dimensional) and study the Wasserstein mappings as well as geodesic between digit 0( $\rho_a$ ) and digit 1( $\rho_b$ ), digit 4( $\rho_a$ ) and digit 8( $\rho_b$ ), digit 6( $\rho_a$ ) and digit 9( $\rho_b$ ). Only partial results are presented in Figure 8 due to space limit.

**Training and Results:** For realistic-1 and realistic-2 we set the batch size  $N_t = 1000$ , for realistic-3 in each iteration we take  $N_t = 500$  pictures for training, especially for realistic-3 we also add small noise to samples during the training process. From Figure 6 and 7, the generated autumn(summer) views are very close to the true autumn(summer) views, moreover, the similarity between true and generated palette distributions also demonstrate that our algorithm works well in these cases. In realistic-3 we study the Wasserstein geodesic and mappings in original space without any dimension reduction techniques. Our generated handwritten digits follow similar patterns with the ground truth.

## 5 Conclusion

OT problem has been drawing more attention in machine learning recently. Though many algorithms have been proposed during the past several years for efficient computations, most of them do not consider the Wasserstein geodesics, neither be suitable for estimating optimal transport map with general cost in high dimensions. In this paper we present a novel method to compute Wasserstein geodesic between two given distributions with general convex cost. In particular, we consider the primal-dual scheme of the dynamical OT problem and simplify the scheme via the KKT conditions

as well as the geodesic transporting properties. By further introducing preconditioning techniques and bidirectional dynamics into our optimization, we obtain a stable and effective algorithm that is capable of high dimensional computation, which is one of the very first scalable algorithms for directly computing geodesics with general cost, including  $L-p$ . Our method not only computes sample based Wasserstein geodesics, but also provides Wasserstein distance and optimal map. We demonstrate the effectiveness of our scheme through a series of experiments with both low dimensional and high dimensional settings. It is worth noting that our model can be applied not only in machine learning such as image transfer, density estimation, but also in optimal control and robotics, where one needs to study the distribution of mobile agents. We should also be aware of the malicious usage for our method, for instance, it could be potentially used in some activities involving generating misleading data distributions.

## References

- Altschuler, J., Niles-Weed, J., and Rigollet, P. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. *Advances in neural information processing systems*, pp. 1964–1974, 2017.
- Amos, B., Xu, L., and Kolter, J. Sinkhorn distances: Lightspeed computation of optimal transport. In *International Conference on Machine Learning*, pp. 146–155, 2017.
- Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein gan. In *arXiv preprint arXiv:1701.07875*, 2017.
- Benamou, J. and Brenier, Y. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000a.
- Benamou, J. and Brenier, Y. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. In *Numerische Mathematik*, 2000b.
- Benamou, J., Froese, B., and Oberman, A. Two numerical methods for the elliptic monge-ampere equation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44(4):737–758, 2010.
- Chen, Y., Georgiou, T., and Pavon, M. On the relation between optimal transport and schrödinger bridges: A stochastic control viewpoint. In *Journal of Optimization Theory and Applications*, 2016.
- Chen, Y., Georgiou, T., and Pavon, M. Optimal transport in systems and control. *Annual Review of Control, Robotics, and Autonomous Systems*, 4, 2020.
- Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2016.
- Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. In *Neural Information Processing Systems*, 2013.
- Dessein, A., Papadakis, N., and Rouas, J. Regularized optimal transport and the rot mover’s distance. *The Journal of Machine Learning Research*, 19(1):590–642, 2018.
- Dukler, Y., Li, W., Tong Lin, A., and Montúfar, G. Wasserstein of wasserstein loss for learning generative models. 2019.
- Fan, J., Taghvaei, A., and Chen, Y. Scalable computations of wasserstein barycenter via input convex neural networks. *arXiv preprint arXiv:2007.04462*, 2020.
- Finlay, C., Gerolin, A., Oberman, A., and Pooladian, A. Learning normalizing flows from entropy-kantorovich potentials. In *arXiv preprint arXiv:2006.06033*, 2020a.
- Finlay, C., Jacobsen, J.-H., Nurbekyan, L., and Oberman, A. M. How to train your neural ode: the world of jacobian and kinetic regularization. In *arXiv preprint arXiv:2002.02798*, 2020b.
- Flamary, R., Courty, N., Rakotomamonjy, A., and Tuia, D. Optimal transport with laplacian regularization. In *Advances in Neural Information Processing Systems, Workshop on Optimal Transport and Machine Learning*, 2014.

- Flamary, R., Courty, N., Gramfort, A., Alaya, M. Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., Gautheron, L., Gayraud, N. T., Janati, H., Rakotomamonjy, A., Redko, I., Rolet, A., Schutz, A., Seguy, V., Sutherland, D. J., Tavenard, R., Tong, A., and Vayer, T. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. URL <http://jmlr.org/papers/v22/20-451.html>.
- Gangbo, W., Li, W., Osher, S., and Puthawala, M. Unnormalized optimal transport. *Journal of Computational Physics*, 399:108940, 2019.
- Genevay, A., Cuturi, M., Peyré, G., and Bach, F. Stochastic optimization for large-scale optimal transport. In *Advances in neural information processing systems*, pp. 3440–3448, 2016.
- Genevay, A., Peyré, G., and Cuturi, M. Learning generative models with sinkhorn divergences. *International Conference on Artificial Intelligence and Statistics*, pp. 1608–1617, 2018.
- Grathwohl, W., Chen, R., Bettencourt, J., Sutskever, I., and Duvenaud, D. Ffjord: Free-form continuous dynamics for scalable reversible generative models. In *arXiv preprint arXiv:1810.01367*, 2018.
- Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In *Neural Information Processing Systems*, 2017.
- Inoue, D., Ito, Y., and Yoshida, H. Optimal transport-based coverage control for swarm robot systems: Generalization of the voronoi tessellation-based method. *IEEE Control Systems Letters*, 5(4):1483–1488, 2020.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2014.
- Korotin, A., Egiazarian, V., Asadulaev, A., Safin, A., and Burnaev, E. Wasserstein-2 generative networks. *arXiv preprint arXiv:1909.13082*, 2019.
- Krishnan, V. and Martínez, S. Distributed optimal transport for the deployment of swarms. *IEEE Conference on Decision and Control (CDC)*, pp. 4583–4588, 2018.
- Kuang, M. and Tabak, E. Sample-based optimal transport and barycenter problems. *Communications on Pure and Applied Mathematics*, 72(8):1581–1630, 2019.
- Kuang, M. and Tabak, E. G. Preconditioning of optimal transport. *SIAM Journal on Scientific Computing*, 39(4):A1793–A1810, 2017.
- Kuhn, H. W. and Tucker, A. W. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pp. 481–492, Berkeley, Calif., 1951. University of California Press. URL <https://projecteuclid.org/euclid.bsmsp/1200500249>.
- Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.
- Li, R., Ye, X., Zhou, H., and Zha, H. Learning to match via inverse optimal transport. In *J. Mach. Learn. Res.*, pp. 20, pp.80–1, 2019.
- Li, W., Ryu, E., Osher, S., Yin, W., and Gangbo, W. A parallel method for earth mover’s distance. *Journal of Scientific Computing*, 75(1):182–197, 2018.
- Lin, A., Fung, S., Li, W., Nurbekyan, L., and Osher, S. Apac-net: Alternating the population and agent control via two neural networks to solve high-dimensional stochastic mean field games. In *arXiv preprint arXiv:2002.10113*, 2020.
- Ma, S., Liu, S., Zha, H., and Zhou, H. Learning stochastic behaviour of aggregate data. In *arXiv preprint arXiv:2002.03513*, 2020.
- Makkluva, A. and Taghvaei, A., Oh, S., and Lee, J. Optimal transport mapping via input convex neural networks. In *International Conference on Machine Learning*, pp. 6672–6681, 2020.
- Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

- Muzellec, B., Nock, R., Patrini, G., and Nielsen, F. Tsallis regularized optimal transport and ecological inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), 2017.
- Reinhard, E., Ashikhmin, M., Gooch, B., and Shirley, P. Color transfer between images. *IEEE Comput. Graph. Appl.*, 21(5):34–41, September 2001. ISSN 0272-1716. doi: 10.1109/38.946629. URL <https://doi.org/10.1109/38.946629>.
- Ruthotto, L., Osher, S. J., Li, W., Nurbekyan, L., and Fung, S. W. A machine learning framework for solving high-dimensional mean field game and mean field control problems. *Proceedings of the National Academy of Sciences*, 117(17):9183–9193, 2020.
- Seguy, V., Damodaran, B., Flamary, R., Courty, N., R., A., and Blondel, M. Large-scale optimal transport and mapping estimation. *arXiv preprint arXiv:1711.02283*, 2017.
- Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. Wasserstein auto-encoders. In *arXiv preprint arXiv:1711.01558*, pp. 3440–3448, 2017.
- Tong Lin, A., Li, W., Osher, S., and Montúfar, G. Wasserstein proximal of gans. 2018.
- Trigila, G. and Tabak, E. Data-driven optimal transport. *Communications on Pure and Applied Mathematics*, 69(4):613–648, 2016.
- Villani, C. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.
- Villani, C. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.
- Xie, Y., Chen, M., Jiang, H., Zhao, T., and Zha, H. On scalable and efficient computation of large scale optimal transport. In *In International Conference on Machine Learning*, pp. 6882–6892, 2019.
- Xie, Y., Wang, X., Wang, R., and Zha, H. A fast proximal point method for computing exact wasserstein distance. In *Uncertainty in Artificial Intelligence*, pp. 433–453, 2020.
- Yang, I. Wasserstein distributionally robust stochastic control: A data-driven approach. *IEEE Transactions on Automatic Control*, 2020.
- Yang, L. and Karniadakis, G. Potential flow generator with  $l_2$  optimal transport regularity for generative models. In *arXiv preprint arXiv:1908.11462*, 2019.

## A Algorithm

### A.1 Preconditioning technique for 2-Wasserstein case

It's worth mentioning that we can apply preconditioning technique under the 2-Wasserstein cases, i.e.,  $L(\cdot) = \frac{|\cdot|^2}{2}$ . When the support of distributions  $\rho_a$  and  $\rho_b$  are far away from each other, the computational process might get much more sensitive with respect to vector field  $F$ . In order to deal with this situation, we consider preconditioning to our initial distribution  $\rho_a$ . In our implementation, we treat  $P : \mathbb{R}^d \rightarrow \mathbb{R}^d$  as our preconditioning map. We fix its structure as  $P(x) = \sigma x + \mu$  with  $\sigma \in \mathbb{R}_+$ ,  $\mu \in \mathbb{R}^d$ . Such preconditioning process can be treated as an operation aiming at relocating and rescaling the initial distribution  $\rho_a$  so that the support of  $P_{\sharp}\rho_a$  matches with the support of  $\rho_b$  in a better way, which in turn facilitates the training process of our OT problem. A similar technique is also carried out by Kuang & Tabak (2017).

Let us denote the optimal vector field of OT problem between  $P_{\sharp}\rho_a$  and  $\rho_b$  as  $\nabla\widehat{\Phi}_0$ , then for the vector field  $F^*(x) = \nabla\widehat{\Phi}_0 \circ P(x) + P(x) - x$ , the following theorem guarantees the optimality of  $F^*$ .

**Theorem 3.** Suppose  $L(\cdot) = \frac{|\cdot|^2}{2}$ . We define the map  $P(x) = \sigma x + \mu$  with  $\sigma \in \mathbb{R}_+$ ,  $\mu \in \mathbb{R}^d$ . Recall (11), we denote  $v(x, t) = \nabla\widehat{\Phi}(x, t)$  as the optimal solution to dynamical OT problem (7) from  $P_{\sharp}\rho_a$  to  $\rho_b$ . We set  $\widehat{\Phi}_0 = \widehat{\Phi}(\cdot, 0)$ . Furthermore, we denote  $v(x, t) = \nabla\Phi^*(x, t)$  as the optimal solution to dynamical OT problem (7) from  $\rho_a$  to  $\rho_b$ , and set  $\Phi_0^* = \Phi^*(\cdot, 0)$ . Then we have

$$\nabla\Phi_0^*(x) = \nabla\widehat{\Phi}_0 \circ P(x) + P(x) - x, \quad \Phi_0^*(x) = \frac{1}{\sigma}\widehat{\Phi}_0(\sigma x + \mu) + \frac{\sigma - 1}{2}|x|^2 + \mu^T x + \text{Const.}$$

This theorem indicates that our constructed  $F^*$  is exactly the optimal transport field  $\nabla\Phi_0^*$  for the original OT problem from  $\rho_a$  to  $\rho_b$ . The proof is in Appendix C.

Our computation procedure is summarized in Algorithm 1. We set  $F_{\theta_1}, G_{\theta_2}$  and  $\Phi_{\omega_1}^F, \Phi_{\omega_2}^G$  as fully connected neural networks and optimize over their parameters.

---

**Algorithm 1** Computing Wasserstein geodesic from  $\rho_a$  to  $\rho_b$  via bidirectional scheme (23) and preconditioning

---

1: Choose our preconditioning map  $P(x) = \sigma x + \mu$ . Denote  $\hat{\rho}_a = P_{\sharp}\rho_a$  (This step is only applicable for 2-Wasserstein case. If we do not need preconditioning, we treat  $P = \text{Id}$ .)

2: Set up the threshold  $\epsilon > 0$  as the stopping criteria

3: Initialize  $F_{\theta_1}, G_{\theta_2}, \Phi_{\omega_1}^F, \Phi_{\omega_2}^G$

4: **for**  $F_{\theta_1}, G_{\theta_2}$  steps **do**

5:     Sample  $\{(z_k^a, t_k^a)\}_{k=1}^N$  from  $\hat{\rho}_a \otimes U(0, 1)$  and  $\{(z_k^b, t_k^b)\}_{k=1}^N$  from  $\rho_b \otimes U(0, 1)$ ;

6:     Set  $x_k^a = z_k^a + t_k^a F_{\theta_1}(z_k^a)$ ,  $x_k^b = z_k^b + t_k^b G_{\theta_2}(z_k^b)$ ;

7:     Sample  $\{w_k^a\}_{k=1}^M$  from  $\hat{\rho}_a$  and  $\{w_k^b\}$  from  $\rho_b$ ;

8:     **for**  $\Phi_{\omega_1}^F, \Phi_{\omega_2}^G$  steps **do**

9:         Update (via gradient ascent)  $\Phi_{\omega_1}^F, \Phi_{\omega_2}^G$  by:

$$\nabla_{\omega_1, \omega_2} (\mathcal{L}^{ab}(\Phi_{\omega_1}^F) + \mathcal{L}^{ba}(\Phi_{\omega_2}^G))$$

10:     **end for**

11:     Sample  $\{\xi_k^a\}_{k=1}^K$  from  $\hat{\rho}_a$  and  $\{\xi_k^b\}_{k=1}^K$  from  $\rho_b$

12:     Update (grad descent)  $F_{\theta_1}, G_{\theta_2}$  by:

$$\nabla_{\theta_1, \theta_2} (\mathcal{L}^{ab}(\Phi_{\omega_1}^F) + \mathcal{L}^{ba}(\Phi_{\omega_2}^G) + \mathcal{K}(F_{\theta_1}, G_{\theta_2}))$$

13:     Whenever  $|\widehat{W}^{ab} - \widehat{W}^{ba}| < \epsilon$ , skip out of the loop.

14: **end for**

15: Set  $F^* = F_{\theta_1} \circ P + P - \text{Id}$  and  $G^* = G_{\theta_2} \circ P + P - \text{Id}$ .

16: Wasserstein geodesic from  $\rho_a$  to  $\rho_b$  is given by  $\{(\text{Id} + tF_{\theta_1})_{\sharp}\rho_a\}$ ; Wasserstein geodesic from  $\rho_b$  to  $\rho_a$  is given by  $\{(\text{Id} + tG_{\theta_2})_{\sharp}\rho_b\}$ .

---

**Remark 2.** In Algorithm 1, we need to sample points  $\{z_k^a\}$  from the distribution  $\hat{\rho}_a = P_{\sharp}\rho_a$ . To achieve this, we first sample  $\{u_k\}$  from  $\rho_a$ . Then  $\{P(u_k)\}$  are our desired samples from  $\hat{\rho}_a$ .

In Algorithm 1 we define

$$\begin{aligned}\mathcal{L}^{ab}(\Phi_{\omega_1}^F) &= -\frac{1}{N} \sum_{k=1}^N \left[ \frac{\partial}{\partial t} \Phi_{\omega_1}^F(x_k^a, t_k^a) + H(\nabla \Phi_{\omega_1}^F(x_k^a, t_k^a)) \right] + \frac{1}{M} \sum_{k=1}^M (\Phi_{\omega_1}^F(w_k^b, 1) - \Phi_{\omega_1}^F(w_k^a, 0)), \\ \mathcal{L}^{ba}(\Phi_{\omega_2}^G) &= -\frac{1}{N} \sum_{k=1}^N \left[ \frac{\partial}{\partial t} \Phi_{\omega_2}^G(x_k^b, t_k^b) + H(\nabla \Phi_{\omega_2}^G(x_k^b, t_k^b)) \right] + \frac{1}{M} \sum_{k=1}^M (\Phi_{\omega_2}^G(w_k^b, 1) - \Phi_{\omega_2}^G(w_k^a, 0)), \\ \mathcal{K}(F_{\theta_1}, G_{\theta_2}) &= \frac{\lambda}{K} \sum_{k=1}^K |G_{\theta_2}(\xi_k^a + F_{\theta_1}(\xi_k^a)) + F_{\theta_1}(\xi_k^a)|^2 + \frac{\lambda}{K} \sum_{k=1}^K |F_{\theta_1}(\xi_k^b + G_{\theta_2}(\xi_k^b)) + G_{\theta_2}(\xi_k^b)|^2, \\ \widehat{W}^{ab} &= \frac{1}{M} \sum_{k=1}^M L(F_{\theta_1}(w_k^a)), \quad \widehat{W}^{ba} = \frac{1}{M} \sum_{k=1}^M L(G_{\theta_2}(w_k^b)).\end{aligned}$$

## B Proof of Theorem 2

Let us denote  $F : \mathbb{R}^d \rightarrow \mathbb{R}^d$  as the transporting vector field. Recall that we are computing for the Wasserstein geodesic interpolating  $\rho_a$  and  $\rho_b$ . We denote  $\hat{\rho}_t = (I + tF)_{\sharp}\rho_a$ . Then we introduce the following functional of  $F$  and  $\Phi$ :

$$\begin{aligned}\mathcal{L}(F, \Phi) &= \int_0^1 \int \left( -\frac{\partial \Phi(x, t)}{\partial t} - H(\nabla \Phi(x, t)) \right) \hat{\rho}(x, t) dx dt + \int \Phi(x, 1) \rho_b(x) - \Phi(x, 0) \rho_a(x) dx \\ &= \int_0^1 \int \left( -\frac{\partial \Phi(x + tF(x), t)}{\partial t} - H(\nabla \Phi(x + tF(x), t)) \right) \rho_a(x, t) dx dt + \int \Phi(x, 1) \rho_b(x) \\ &\quad - \Phi(x, 0) \rho_a(x) dx\end{aligned}\tag{24}$$

As mentioned in (22), our numerical method is to solve the following saddle point problem

$$\min_F \max_{\Phi} \mathcal{L}(F, \Phi)\tag{25}$$

As stated in Theorem 2, the optimal solution obtained from dynamical OT problem (Brenier-Benamou formulation) (7) is a critical point to the functional  $\mathcal{L}(F, \Phi)$ . Before we prove this result, we need the following lemmas:

**Lemma 1.** Given a distribution with density  $\rho$  defined on  $\mathbb{R}^d$ , consider vector field  $F : \mathbb{R}^d \rightarrow \mathbb{R}^d$ . Define time-varying density  $\{\rho(\cdot, t)\}_{t \in [0, 1]}$  as  $\rho(\cdot, t) = (Id + tF)_{\sharp}\rho_0$ . Suppose for a given  $f \in C^1(\mathbb{R}^d)$ ,  $f(x)\rho(x, t)$  is integrable on  $\mathbb{R}^d$ . Then

$$\int f(x) \frac{\partial}{\partial t} \rho(x, t) = \int \nabla f(x + tF(x)) \cdot F(x) \rho_a(x) dx$$

*Proof.* We have

$$\begin{aligned}\int f(x) \frac{\partial}{\partial t} \rho(x, t) &= \frac{d}{dt} \left( \int f(x) \rho(x, t) dx \right) = \frac{d}{dt} \left( \int f(x + tF(x)) \rho_a(x) dx \right) \\ &= \int \nabla f(x + tF(x)) \cdot F(x) \rho_a(x) dx\end{aligned}$$

□

**Lemma 2.** Suppose  $\Phi^*(x, t)$  is solved from (10) in the paper with initial condition  $\Phi^*(\cdot, 0) = \Phi_0^*(\cdot)$ , we further assume  $\Phi^*(\cdot, t) \in C^2(\mathbb{R}^d)$ . Then we have

$$\nabla \Phi^*(x + t \nabla L^{-1}(\nabla \Phi_0^*(x)), t) = \nabla \Phi_0^*(x).\tag{26}$$

*Proof.* Now consider Hamilton-Jacobi equation of (10) in the paper:

$$\frac{\partial \Phi^*(y, t)}{\partial t} + H(\nabla \Phi^*(y, t)) = 0 \quad \Phi^*(\cdot, 0) = \Phi_0^*.$$

We take gradient with respect to  $x$  on both sides, we have

$$\frac{\partial}{\partial t}(\nabla \Phi^*(y, t)) + \nabla^2 \Phi^*(y, t) \nabla H(\nabla \Phi^*(y, t)) = 0. \quad (27)$$

Let us denote  $T_t(x) = x + t \nabla H(\nabla \Phi_0^*(x))$  for simplicity. We now compute

$$\frac{d}{dt} \nabla \Phi^*(T_t(x), t) = \frac{\partial}{\partial t} \nabla \Phi^*(T_t(x), t) + \nabla^2 \Phi^*(T_t(x), t) \nabla H(\nabla \Phi_0^*(x))$$

By plugging  $y = T_t(x)$  into (27), we are able to verify  $\frac{d}{dt} \nabla \Phi^*(T_t(x), t) = 0$ . Thus

$$\nabla \Phi^*(T_t(x), t) = \nabla \Phi^*(T_0(x), 0) = \nabla \Phi_0^*(x) \quad \text{for } t \in [0, 1] \quad (28)$$

Recall  $H$  defined in the paper, we can verify that  $\nabla H = \nabla L^{-1}$ . Thus (28) leads to

$$\nabla \Phi^*(x + t \nabla L^{-1}(\nabla \Phi_0^*(x)), t) = \nabla \Phi_0^*(x).$$

□

**Lemma 3.** Suppose  $\Phi^*(x, t)$  is solved from (10) in the paper with initial condition  $\Phi^*(\cdot, 0) = \Phi_0^*(\cdot)$ , we further assume  $\Phi^*(\cdot, t) \in C^2(\mathbb{R}^d)$ . Now denote  $\hat{\rho}(\cdot, t) = (Id + t \nabla L^{-1}(\nabla \Phi_0^*)) \# \rho_a$ . Then  $\hat{\rho}(\cdot, t)$  solves

$$\frac{\partial \hat{\rho}(x, t)}{\partial t} + \nabla \cdot (\hat{\rho}(x, t) \nabla L^{-1}(\nabla \Phi^*(x, t))) = 0.$$

*Proof.* For arbitrary  $f \in C_0^\infty(\mathbb{R}^d)$ , we consider:

$$\begin{aligned} & \int f(x) \left( \frac{\partial \hat{\rho}(x, t)}{\partial t} + \nabla \cdot (\hat{\rho}(x, t) \nabla L^{-1}(\nabla \Phi^*(x, t))) \right) dx \\ &= \int f(x) \frac{\partial \hat{\rho}(x, t)}{\partial t} dx - \int \nabla f(x) \cdot \nabla L^{-1}(\nabla \Phi^*(x, t)) \hat{\rho}(x, t) dx \end{aligned}$$

By Lemma 1, the first term equals

$$\int \nabla f(x + t \nabla L^{-1}(\nabla \Phi_0^*(x))) \cdot \nabla L^{-1}(\nabla \Phi_0^*(x)) \rho_a(x) dx \quad (29)$$

The second term equals

$$\int \nabla f(x + t \nabla L^{-1}(\nabla \Phi_0^*(x))) \cdot \nabla L^{-1}(\nabla \Phi^*(x + t \nabla L^{-1}(\nabla \Phi_0^*(x)), t)) \rho_a(x) dx \quad (30)$$

Using Lemma 2, we know the integrals (29) and (30) are the same. Thus we have

$$\int f(x) \left( \frac{\partial \hat{\rho}(x, t)}{\partial t} + \nabla \cdot (\hat{\rho}(x, t) \nabla L^{-1}(\nabla \Phi^*(x, t))) \right) dx = 0 \quad \forall f \in C_0^\infty(\mathbb{R}^d).$$

This leads to our result. □

**Lemma 4.** Suppose  $\Phi^*(x, t)$  is solved from (10) in the paper with initial condition  $\Phi^*(\cdot, 0) = \Phi_0^*(\cdot)$ , then

$$W_{Dym}(\rho_a, \rho_b) = \int L(\nabla L^{-1}(\nabla \Phi_0^*(x))) \rho_a(x) dx. \quad (31)$$

*Proof.* Consider particle dynamical OT with its optimal solution  $v^*(x, t) = \nabla L^{-1}(\nabla \Phi^*(x, t))$  as stated in (11) in the paper. Recall Theorem 1 stating that the optimal plan is transporting each particle  $\mathbf{X}_t$  along straight lines with constant velocity  $v^*(\mathbf{X}_0, 0) = \nabla L^{-1}(\nabla \Phi_0^*(\mathbf{X}_0))$ , i.e.  $v^*(\mathbf{X}_t, t) = v^*(\mathbf{X}_0, 0)$  for any  $t \in [0, 1]$ . Combining these, we have

$$W_{Dym OT}(\rho_a, \rho_b) = \int_0^1 \mathbb{E} L(v^*(\mathbf{X}_t, t)) dt = \mathbb{E} \left( \int_0^1 L(v^*(\mathbf{X}_t, t)) dt \right) = \mathbb{E} L(\nabla L^{-1}(\nabla \Phi_0^*(\mathbf{X}_0))).$$

Notice that we require  $\mathbf{X}_0 \sim \rho_a$ . This will lead to (31). □

**Theorem 2.** Recall the solution to the equation system (10) in the paper is  $\rho(x, t)$  and  $\Phi^*(x, t)$ , suppose  $\Phi^*(\cdot, t) \in C^2(\mathbb{R}^d)$ . Denote  $\Phi_0^*(\cdot) = \Phi^*(\cdot, 0)$ , then  $(\nabla L^{-1}(\nabla \Phi_0^*), \Phi^*)$  is a critical point to the functional  $\mathcal{L}$ , i.e.,

$$\frac{\partial \mathcal{L}}{\partial F}(\nabla L^{-1}(\nabla \Phi_0^*), \Phi^*) = 0, \quad \frac{\partial \mathcal{L}}{\partial \psi}(\nabla L^{-1}(\nabla \Phi_0^*), \Phi^*) = 0.$$

Furthermore,  $\mathcal{L}(\nabla L^{-1}(\nabla \Phi_0^*), \Phi^*) = W_{\text{Dym OT}}(\rho_a, \rho_b)$ .

*Proof.* Since we have assumed  $\Phi^*(\cdot, t) \in C^2(\mathbb{R}^d)$ , we restrict our  $\Phi \in C^2(\mathbb{R}^d)$  as well. We first rewrite  $\mathcal{L}(F, \Phi)$  by using integration by parts as:

$$\int_0^1 \int \Phi(x, t) \frac{\partial \hat{\rho}(x, t)}{\partial t} - H(\nabla \Phi(x, t)) \hat{\rho}(x, t) dx dt + \int \Phi(x, 1)(\rho_a(x) - \hat{\rho}(x, 1)) dx. \quad (32)$$

By Lemma 1, (32) can be written as

$$\begin{aligned} \mathcal{L}(F, \Phi) &= \int_0^1 \int_{\mathbb{R}^d} [\nabla \Phi(x + tF(x), t) \cdot F(x) - H(\nabla \Phi(x + tF(x), t))] \rho_a(x) dx dt \\ &\quad + \int \Phi(x, 1) \rho_b(x) dx - \int \Phi(x + F(x), 1) \rho_a(x) dx. \end{aligned} \quad (33)$$

Now based on (33) here, we are able to compute  $\frac{\partial \mathcal{L}(F, \Phi)}{\partial F}(x)$  as

$$\begin{aligned} \frac{\partial \mathcal{L}(F, \Phi)}{\partial F} &= \int_0^1 t \nabla^2 \Phi(x + tF(x), t) \cdot \underbrace{[F(x) - \nabla H(\nabla \Phi(x + tF(x), t))] \rho_a(x)}_{(A)} dt \\ &\quad + \underbrace{\left( \int_0^1 \nabla \Phi(x + tF(x), t) dt - \nabla \Phi(x + F(x), 1) \right) \rho_a(x)}_{(B)}. \end{aligned} \quad (34)$$

Now we plug  $F = \nabla L^{-1}(\nabla \Phi_0^*)$ ,  $\Phi = \Phi^*$  into (34), by Lemma 2, we have

$$\nabla \Phi^*(x + t \nabla L^{-1}(\nabla \Phi_0^*(x)), t) = \nabla \Phi_0^*(x). \quad (35)$$

Then using (35) and recall that  $\nabla H = \nabla L^{-1}$ , one can verify that  $(A) = 0$ , similarly, for  $(B)$ , we have  $\nabla \Phi(x + tF(x), t) = \nabla \Phi_0^*$  for all  $t \in [0, 1]$ . Thus  $(B) = 0$  and we are able to verify  $\frac{\partial \mathcal{L}}{\partial F}(\nabla L^{-1}(\nabla \Phi_0^*), \Phi^*) = 0$ .

On the other hand, we can compute  $\frac{\partial \mathcal{L}(F, \Phi)}{\partial \Phi}(x, t)$  as

$$\frac{\delta \mathcal{L}(F, \Phi)}{\delta \Phi}(x, t) = \underbrace{\left[ \frac{\partial \hat{\rho}(x, t)}{\partial t} + \nabla \cdot (\hat{\rho}(x, t) \nabla H(\nabla \Phi(x, t))) \right]}_{(C)} + \underbrace{[\rho_b(x) - \hat{\rho}(x, 1)] \delta_1(t)}_{(D)}$$

Now by Lemma 3, we know  $(C) = 0$ . Furthermore, since  $\Phi^*$  solves Dynamical OT problem associated to the optimal transport problem between  $\rho_a$  and  $\rho_b$ , by (13), we have  $\hat{\rho}(x, 1) = (Id + \nabla \Phi_0^*) \# \rho_a = \rho_b$ , this verifies  $(D) = 0$ . Thus, we are able to verify  $\frac{\partial \mathcal{L}(F, \Phi)}{\partial \Phi}(\nabla L^{-1}(\nabla \Phi_0^*), \Phi^*) = 0$ .

At last, we plug  $F = \nabla L^{-1}(\nabla \Phi_0^*)$ ,  $\Phi = \Phi^*$  in (33) to obtain:

$$\begin{aligned} \mathcal{L}(\nabla L^{-1}(\nabla \Phi_0^*), \Phi^*) &= \int_0^1 \int \nabla \Phi_0^*(x) \cdot \nabla L^{-1}(\nabla \Phi_0^*(x)) - H(\nabla \Phi_0^*(x)) \rho_a(x) dx dt \\ &= \int L(\nabla L^{-1}(\nabla \Phi_0^*(x))) \rho_a(x) dx. \end{aligned}$$

Now by Lemma 4, we have verified  $\mathcal{L}(\nabla L^{-1}(\nabla \Phi_0^*), \Phi^*) = W_{\text{Dym OT}}(\rho_a, \rho_b)$ .  $\square$

## C Proof of Theorem 3

**Theorem 3.** Suppose  $L(\cdot) = \frac{|\cdot|^2}{2}$ . We define the map  $P(x) = \sigma x + \mu$  with  $\sigma \in \mathbb{R}_+$ ,  $\mu \in \mathbb{R}^d$ . Recall (11) in the paper, we denote  $v(x, t) = \nabla \widehat{\Phi}(x, t)$  as the optimal solution to dynamical OT problem (7) in the paper from  $P_\sharp \rho_a$  to  $\rho_b$ . We set  $\widehat{\Phi}_0 = \widehat{\Phi}(\cdot, 0)$ . Furthermore, we denote  $v(x, t) = \nabla \Phi^*(x, t)$  as the optimal solution to dynamical OT problem (7) from  $\rho_a$  to  $\rho_b$ . We set  $\Phi_0^* = \Phi^*(\cdot, 0)$ . Then we have

$$\nabla \Phi_0^*(x) = \nabla \widehat{\Phi}_0 \circ P(x) + P(x) - x,$$

$$\text{and } \Phi_0^*(x) = \frac{1}{\sigma} \widehat{\Phi}_0(\sigma x + \mu) + \frac{\sigma-1}{2} |x|^2 + \mu^T x + \text{Const.}$$

*Proof.* According to (13) in the paper, we have

$$(\text{Id} + \nabla \widehat{\Phi}_0)_\sharp(P_\sharp \rho_a) = \rho_b$$

This yields

$$(P + \nabla \Phi_0 \circ P)_\sharp \rho_a = \rho_b$$

We rewrite this as

$$(\text{Id} + \nabla \widehat{\Phi}_0 \circ P + P - \text{Id})_\sharp \rho_a = \rho_b \quad (36)$$

$$\text{We denote } u(x) = \frac{1}{\sigma} \widehat{\Phi}_0(\sigma x + \mu) + \frac{\sigma-1}{2} |x|^2 + \mu^T x.$$

Then we can directly verify that

$$\nabla u(x) = \nabla \widehat{\Phi}_0(\sigma x + \mu) + (\sigma x + \mu) - x = \nabla \widehat{\Phi}_0 \circ P(x) + P(x) - x$$

Plug this into (36) above we get:

$$(\text{Id} + \nabla u)_\sharp \rho_a = \rho_b$$

Using the uniqueness of the solution to Monge-Ampere equation, we have  $\Phi_0^* = u + \text{Const}$ , or equivalently,  $\nabla \Phi_0^*(x) = \nabla u(x) = \nabla \widehat{\Phi}_0 \circ P(x) + P(x) - x$   $\square$

## D Complete Geodesics in Experiments

### D.1 Synthetic Data

Syn-1:

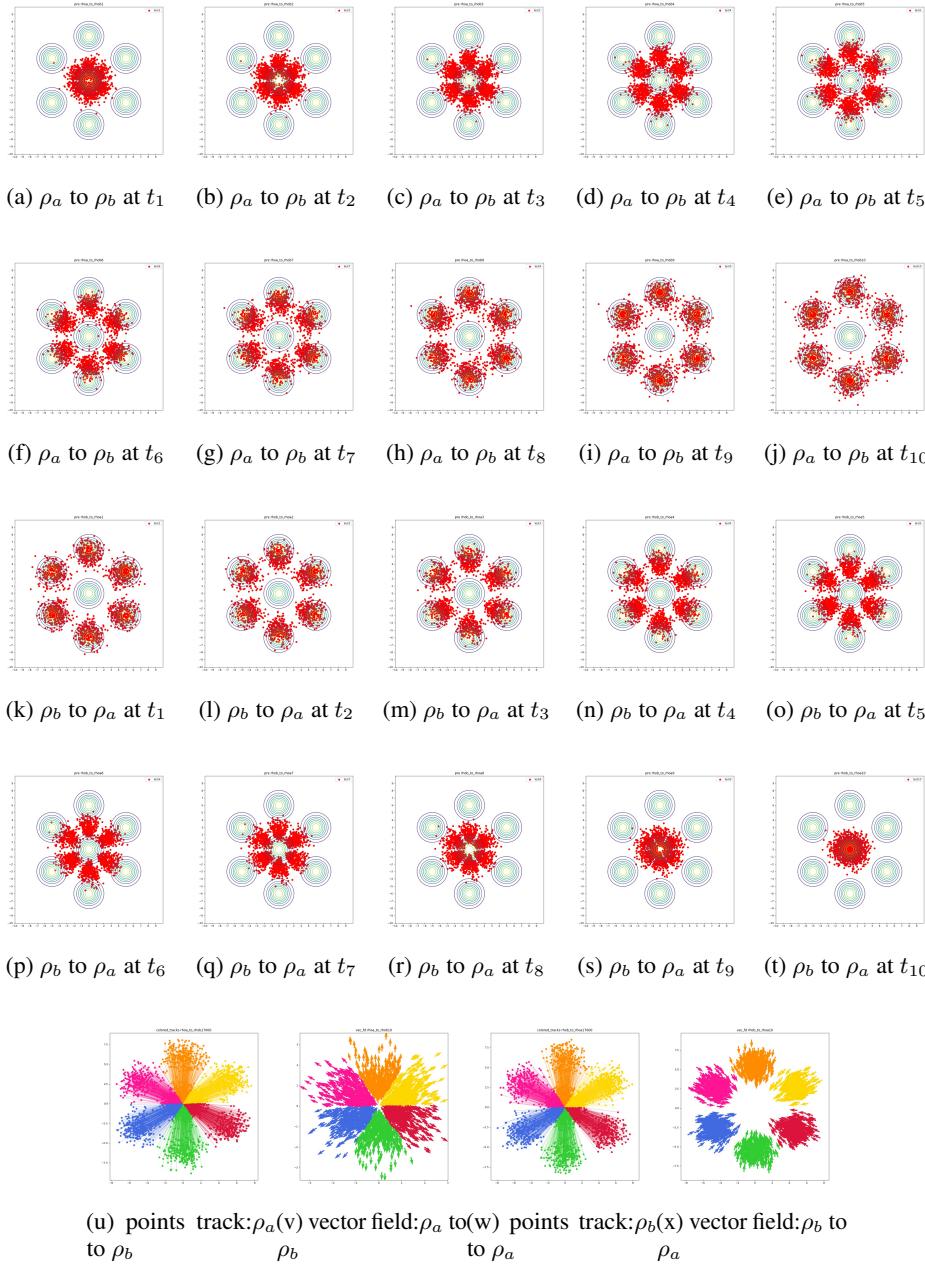


Figure 9: 2-dimensional Gaussian to Gaussian

### Syn-2: Wasserstein-1.5

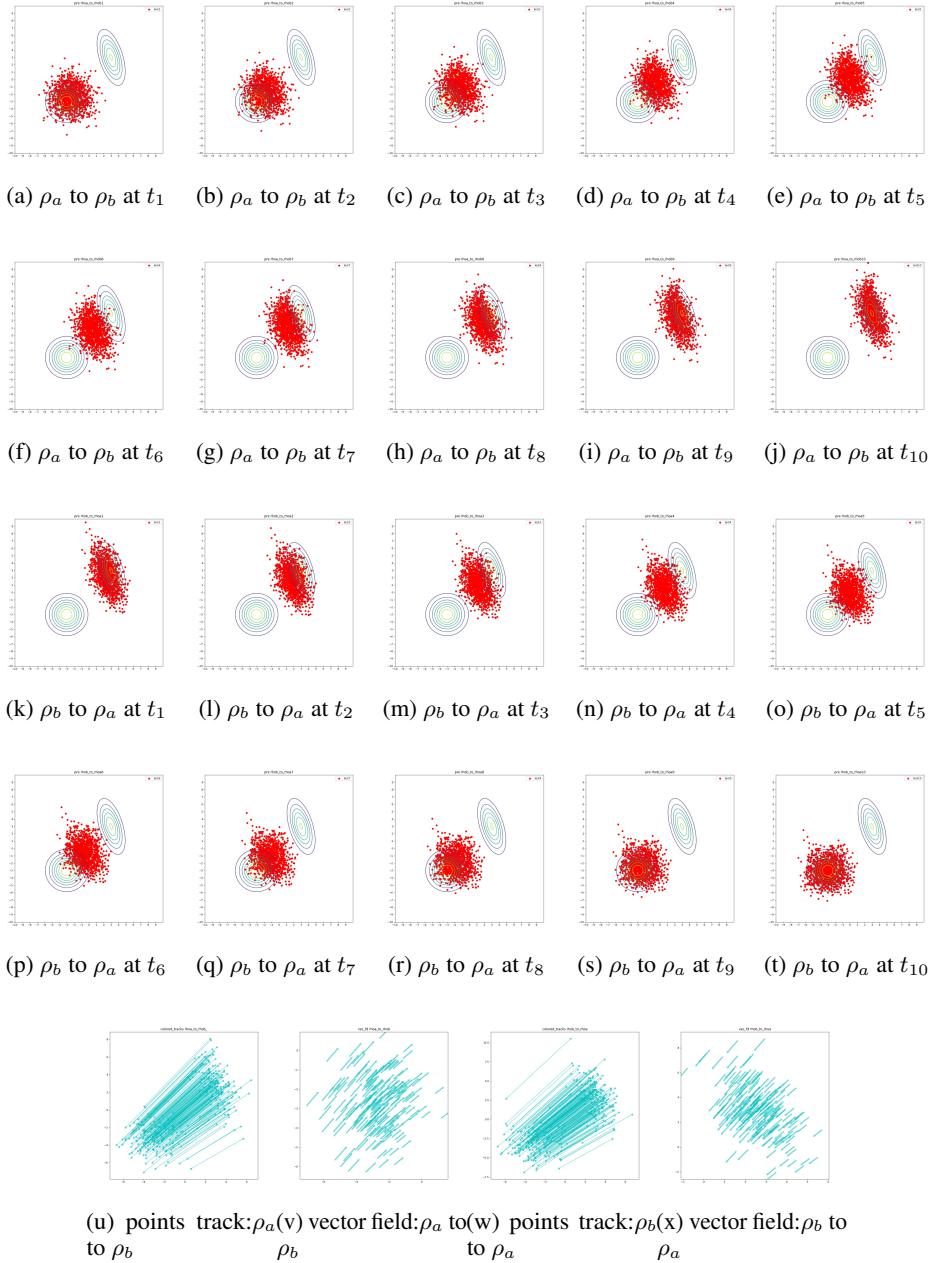


Figure 10: 5-dimensional Gaussian to Gaussian

## Syn-2: Wasserstein-2

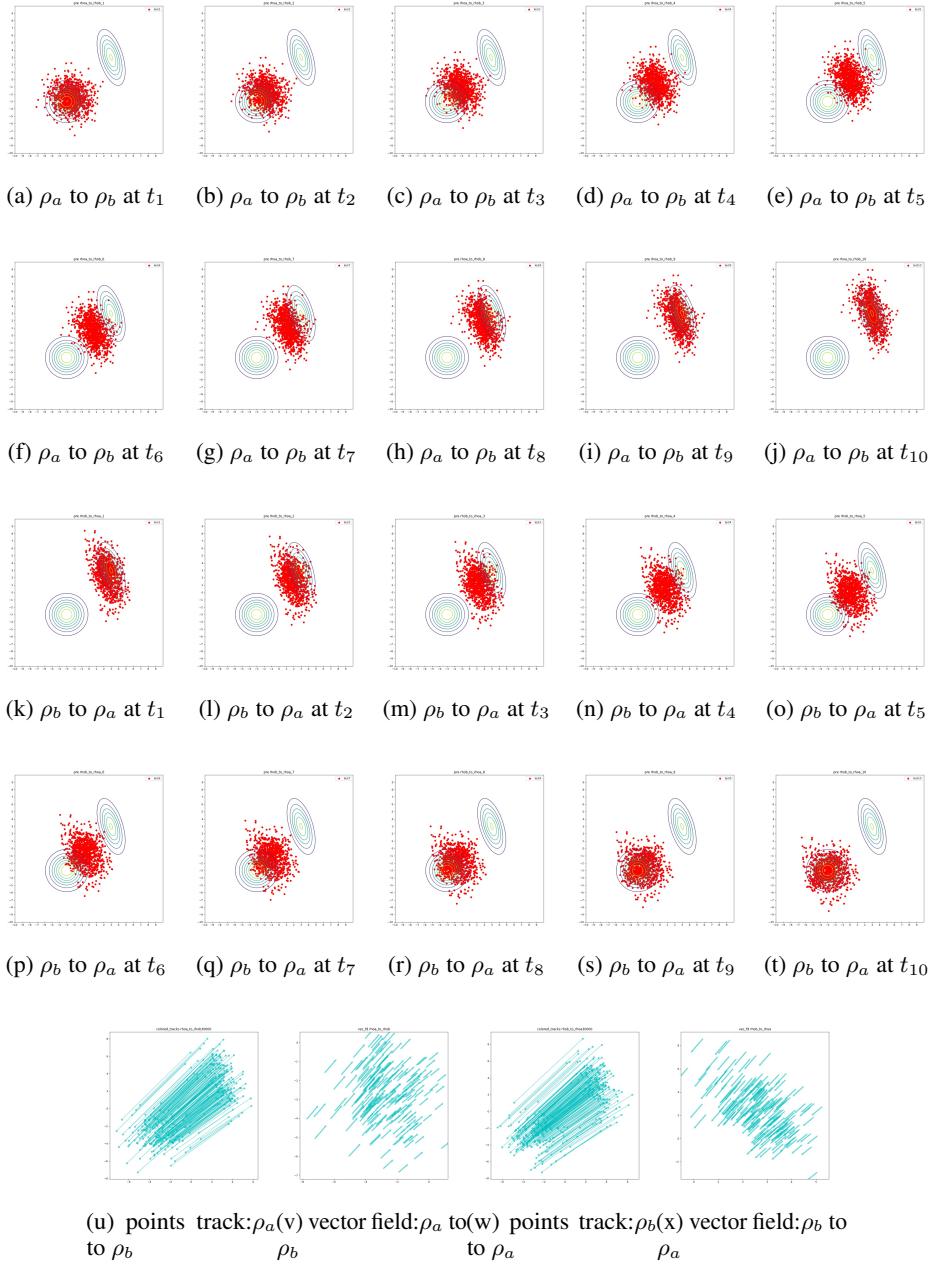


Figure 11: 5-dimensional Gaussian to Gaussian

### Syn-3:



Figure 12: 5-dimensional Gaussian Mixture

### Syn-4:



Figure 13: 10-dimensional unbalanced Gaussian

## D.2 Realistic Data

**Real-1:**

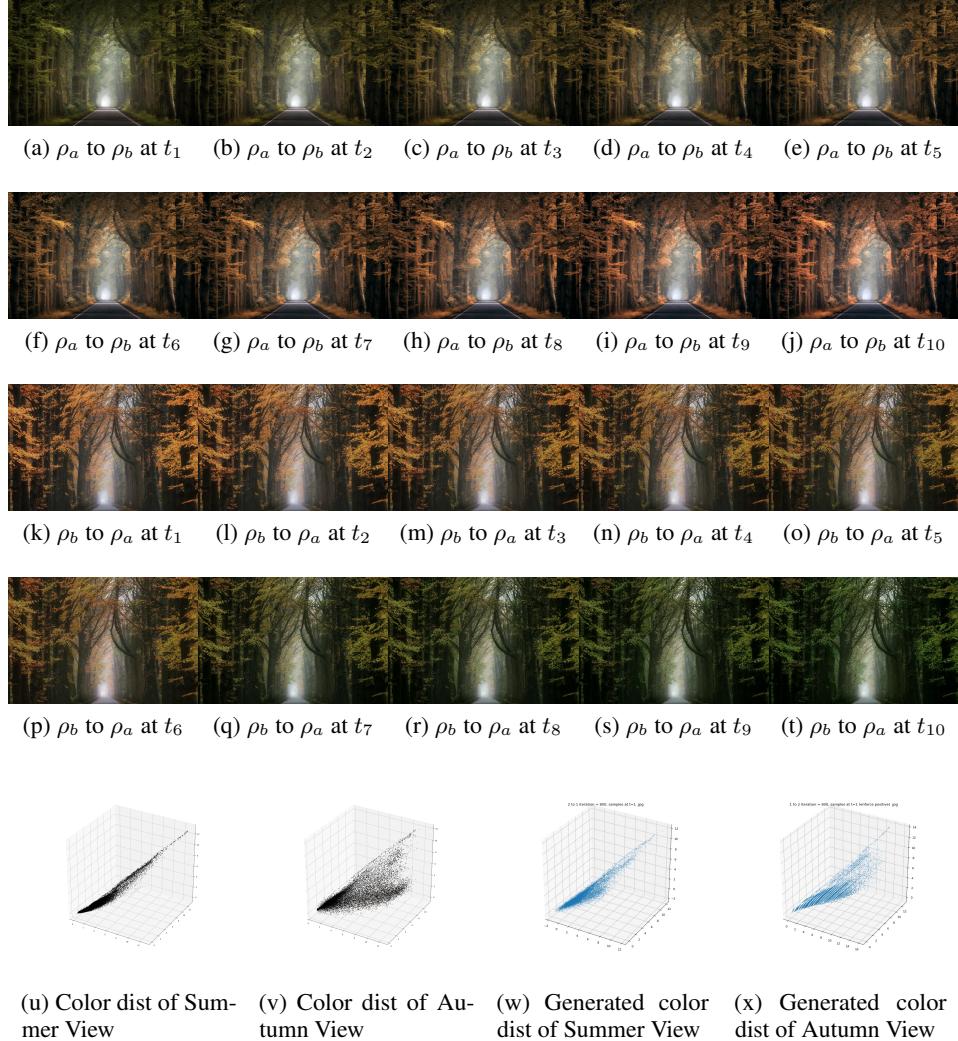


Figure 14: Forest summer view and autumn view

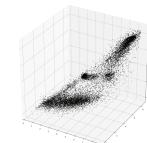
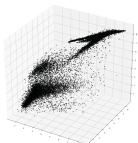
**Real-2:**



(a) Summer West Lake



(b) Autumn White Tower (c) Color dist of West Lake



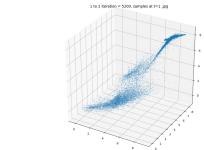
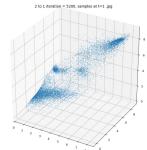
(d) Color dist of White Tower



(e) Autumn West Lake



(f) Summer White Tower (g) Generated color dist of West Lake



(h) Generated color of White Tower

Figure 15: Color transfer

**Real-3:**

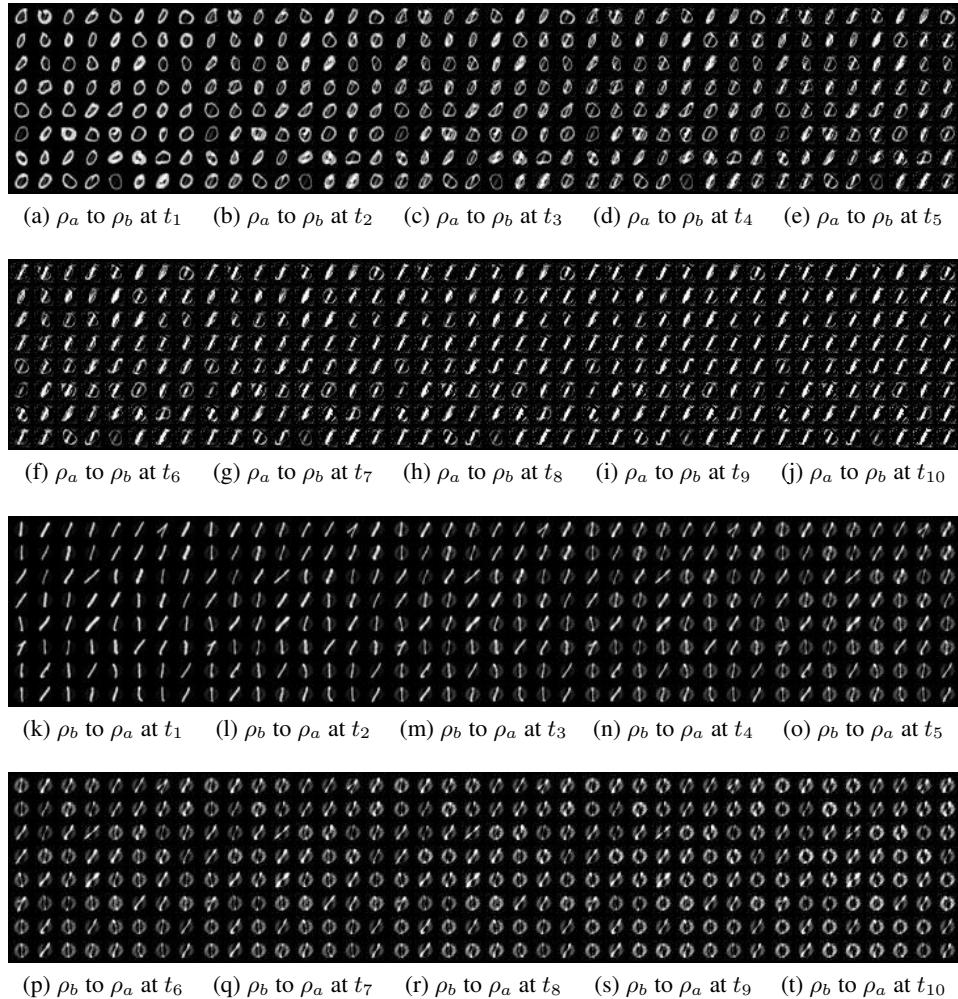
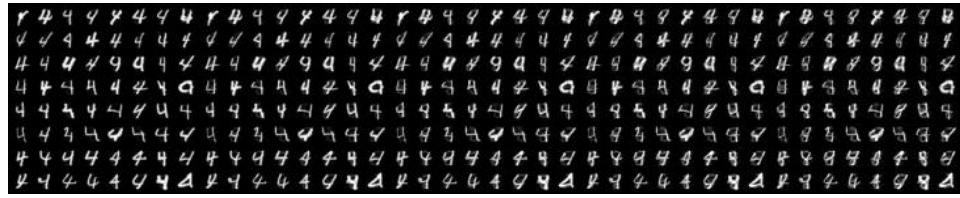


Figure 16: Geodesics between "0" and "1"



A 10x10 grid of characters showing a path from '4' to '8'. The path starts at the bottom-left '4', moves right to another '4', then up to an '8', and continues in a zig-zag pattern across the grid.

(a)  $\rho_a$  to  $\rho_b$  at  $t_1$  (b)  $\rho_a$  to  $\rho_b$  at  $t_2$  (c)  $\rho_a$  to  $\rho_b$  at  $t_3$  (d)  $\rho_a$  to  $\rho_b$  at  $t_4$  (e)  $\rho_a$  to  $\rho_b$  at  $t_5$



A 10x10 grid of characters showing a path from '4' to '8'. The path starts at the bottom-left '4', moves right to another '4', then up to an '8', and continues in a zig-zag pattern across the grid.

(f)  $\rho_a$  to  $\rho_b$  at  $t_6$  (g)  $\rho_a$  to  $\rho_b$  at  $t_7$  (h)  $\rho_a$  to  $\rho_b$  at  $t_8$  (i)  $\rho_a$  to  $\rho_b$  at  $t_9$  (j)  $\rho_a$  to  $\rho_b$  at  $t_{10}$



A 10x10 grid of characters showing a path from '4' to '8'. The path starts at the bottom-left '4', moves right to another '4', then up to an '8', and continues in a zig-zag pattern across the grid.

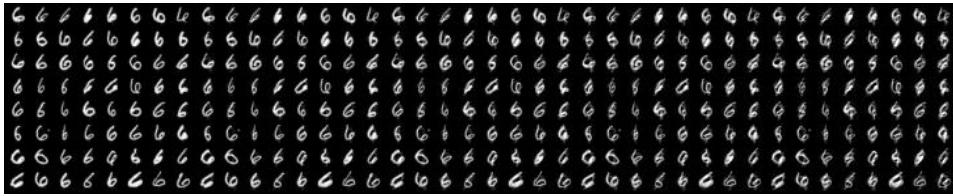
(k)  $\rho_b$  to  $\rho_a$  at  $t_1$  (l)  $\rho_b$  to  $\rho_a$  at  $t_2$  (m)  $\rho_b$  to  $\rho_a$  at  $t_3$  (n)  $\rho_b$  to  $\rho_a$  at  $t_4$  (o)  $\rho_b$  to  $\rho_a$  at  $t_5$



A 10x10 grid of characters showing a path from '4' to '8'. The path starts at the bottom-left '4', moves right to another '4', then up to an '8', and continues in a zig-zag pattern across the grid.

(p)  $\rho_b$  to  $\rho_a$  at  $t_6$  (q)  $\rho_b$  to  $\rho_a$  at  $t_7$  (r)  $\rho_b$  to  $\rho_a$  at  $t_8$  (s)  $\rho_b$  to  $\rho_a$  at  $t_9$  (t)  $\rho_b$  to  $\rho_a$  at  $t_{10}$

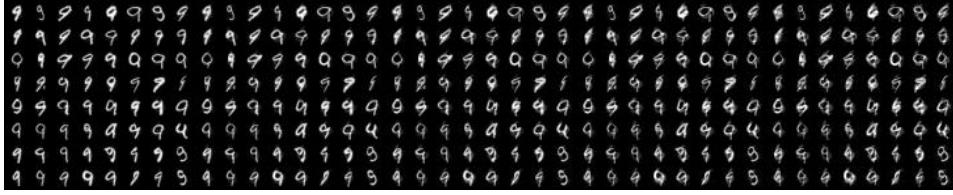
Figure 17: Geodesics between "4" and "8"



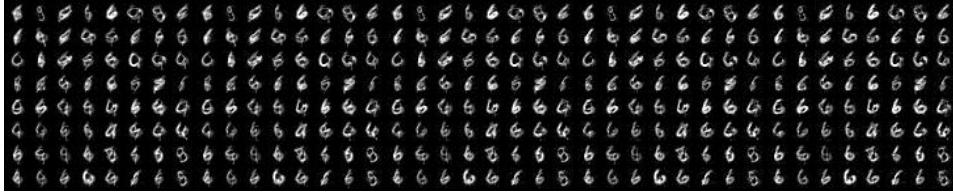
(a)  $\rho_a$  to  $\rho_b$  at  $t_1$  (b)  $\rho_a$  to  $\rho_b$  at  $t_2$  (c)  $\rho_a$  to  $\rho_b$  at  $t_3$  (d)  $\rho_a$  to  $\rho_b$  at  $t_4$  (e)  $\rho_a$  to  $\rho_b$  at  $t_5$



(f)  $\rho_a$  to  $\rho_b$  at  $t_6$  (g)  $\rho_a$  to  $\rho_b$  at  $t_7$  (h)  $\rho_a$  to  $\rho_b$  at  $t_8$  (i)  $\rho_a$  to  $\rho_b$  at  $t_9$  (j)  $\rho_a$  to  $\rho_b$  at  $t_{10}$



(k)  $\rho_b$  to  $\rho_a$  at  $t_1$  (l)  $\rho_b$  to  $\rho_a$  at  $t_2$  (m)  $\rho_b$  to  $\rho_a$  at  $t_3$  (n)  $\rho_b$  to  $\rho_a$  at  $t_4$  (o)  $\rho_b$  to  $\rho_a$  at  $t_5$



(p)  $\rho_b$  to  $\rho_a$  at  $t_6$  (q)  $\rho_b$  to  $\rho_a$  at  $t_7$  (r)  $\rho_b$  to  $\rho_a$  at  $t_8$  (s)  $\rho_b$  to  $\rho_a$  at  $t_9$  (t)  $\rho_b$  to  $\rho_a$  at  $t_{10}$

Figure 18: Geodesics between "6" and "9"