CSIAM Trans. Appl. Math.
doi: 10.4208/csiam-am.SO-2025-0009

Vol. **x**, No. x, pp. 1-22
xx 2025

A Neural Network Framework for High-Dimensional Dynamic Unbalanced Optimal Transport

Wei Wan[1], Jiangong Pan[2], Yuejin Zhang[2], Chenglong Bao[3,4] and Zuoqiang Shi[3,4,*]

[1] *School of Mathematics and Physics, North China Electric Power University, Beijing 102206, China.*
[2] *Department of Mathematical Science, Tsinghua University, Beijing 100084, China.*
[3] *Yau Mathematical Sciences Center, Tsinghua University, Beijing 100084, China.*
[4] *Yanqi Lake Beijing Institute of Mathematical Sciences and Applications, Beijing 101408, China.*

Received 7 February 2025; Accepted 28 July 2025

Abstract. In this paper, we introduce a neural network-based method to address the high-dimensional dynamic unbalanced optimal transport (UOT) problem. Dynamic UOT focuses on the optimal transportation between two densities with unequal total mass, however, it introduces additional complexities compared to the traditional dynamic optimal transport problem. To efficiently solve the dynamic UOT problem in high-dimensional space, we first relax the original problem by using the generalized Kullback-Leibler divergence to constrain the terminal density. Next, we adopt the Lagrangian discretization to address the unbalanced continuity equation and apply the Monte Carlo method to approximate the high-dimensional spatial integrals. Moreover, a carefully designed neural network is introduced for modeling the velocity field and source function. Numerous experiments demonstrate that the proposed framework performs excellently in high-dimensional cases. Additionally, this method can be easily extended to more general applications, such as crowd motion problem.

AMS subject classifications: 65K10, 68T05

Key words: Unbalanced optimal transport, neural network, Lagrangian discretization, crowd motion.

1 Introduction

Optimal transport (OT) [25, 33] is a significant research field, widely applied to various practical problems in machine learning [32], including domain adaptation [26], trans-

*Corresponding author. *Email addresses:* weiwan@ncepu.edu.cn (W. Wan), mathpjg@sina.com (J. Pan), zhangyj19@mails.tsinghua.edu.cn (Y. Zhang), clbao@tsinghua.edu.cn (C. Bao), zqshi@tsinghua.edu.cn (Z. Shi)

fer learning [6, 23], generative models [1, 15, 24], image processing [17, 20], and natural language processing [12]. Despite its success across various domains, the numerical resolution of the optimal transport problem still presents several challenges. The Eulerian discretization-based method suffers from the curse of dimensionality, which means that the computational complexity grows exponentially with spatial dimension. Furthermore, OT typically assumes that densities have the same total mass. However, in real-world applications, a more common scenario arises where the total mass of each density is not equal, which corresponds to so-called unbalanced optimal transport problems. In this paper, we propose an extension of our previous deep learning method designed to solve the high-dimensional OT problem to address the high-dimensional UOT problem.

The OT problem originated from the "sand moving problem" proposed by Monge [19], which studied the minimization of the total cost of material transportation. The solution to the Monge problem is called the optimal transport mapping. However, directly solving the Monge problem is extremely difficult, due to its highly non-convex nature, and the existence of its solution is not guaranteed. The Kantorovich problem [10] relaxes the transport mapping into transport plans, represented by joint probability distributions. In the discrete case, the OT problem can be relaxed into a standard linear programming problem. The Sinkhorn method [7] introduces an entropy regularization term to the original linear programming problem, which can be efficiently solved using the Sinkhorn algorithm [29], significantly accelerating the computation of the OT problem. However, as the dimensionality increases, maintaining accuracy requires an exponentially growing number of grid points, leading to higher computational costs and memory demands. An alternative formulation of the OT problem from the fluid mechanics perspective was first introduced by Benamou and Brenier [2]. Its computation requires solving a partial differential equation (PDE)-constrained optimization problem. Unlike the previous two static formulations, the dynamic OT problem can dynamically describe changes in density by involving a time variable.

The dynamic OT problem can be formulated as follows: Given two probability densities ρ_0 and ρ_1 on $\Omega\subset\mathbb{R}^d$, we want to find a time-dependent density $\rho:\Omega\times[0,1]\to\mathbb{R}$ and a time-dependent velocity field $\boldsymbol{v}:\Omega\times[0,1]\to\mathbb{R}^d$ that transport the mass of ρ_0 to the mass of ρ_1 at minimal transport cost $\mathcal{E}(\rho,\boldsymbol{v})$, i.e.

$$\min_{(\rho,\boldsymbol{v})\in\mathcal{C}(\rho_0,\rho_1)}\int_0^1\int_\Omega\|\boldsymbol{v}(\boldsymbol{x},t)\|^2\rho(\boldsymbol{x},t)\mathrm{d}\boldsymbol{x}\mathrm{d}t=:\mathcal{E}(\rho,\boldsymbol{v}),\tag{1.1}$$

where $(\rho,\boldsymbol{v})\in\mathcal{C}(\rho_0,\rho_1)$ satisfies the dynamical constraint, i.e. the continuity equation, the initial and terminal densities are given by ρ_0 and ρ_1, respectively,

$$\mathcal{C}(\rho_0,\rho_1):=\big\{(\rho,\boldsymbol{v}):\partial_t\rho(\boldsymbol{x},t)+\nabla\cdot\big(\rho(\boldsymbol{x},t)\boldsymbol{v}(\boldsymbol{x},t)\big)=0,\rho(\cdot,0)=\rho_0,\rho(\cdot,1)=\rho_1\big\},\tag{1.2}$$

where $\mathcal{E}(\rho,\boldsymbol{v})$ denotes the generalized kinetic energy and the optimal $\mathcal{E}(\rho,\boldsymbol{v})$ gives the square of the L^2 Wasserstein distance. This fluid mechanics formulation provides a natural time interpolation, which is highly beneficial for a wide range of applications, including continuous normalizing flow [30] and mean field games [13], therefore making the numerical solution for dynamic OT more attractive.

The UOT problem describes the optimal transportation between two densities with different total masses. Various versions of UOT have been developed based on different formulations of OT [27]. This work mainly focuses on solving the dynamic formulation of the UOT problem, a generalization of the dynamic OT problem proposed by Benamou and Brenier, which introduces a source function to control the creation or destruction of mass during transportation. Several dynamic UOT models have been proposed to account for mass change by introducing different source functions f into the cost functional and the continuity equation. Given two densities ρ_0 and ρ_1, the general dynamic UOT problem can be formulated as

$$\min_{(\rho,\boldsymbol{v},f)\in\mathcal{C}(\rho_0,\rho_1)}\int_0^1\int_\Omega\|\boldsymbol{v}(\boldsymbol{x},t)\|^2\rho(\boldsymbol{x},t)\mathrm{d}\boldsymbol{x}\mathrm{d}t+\alpha\mathcal{R}(\rho,f). \tag{1.3}$$

Here, α is a positive constant, and $(\rho,\boldsymbol{v},f)\in\mathcal{C}(\rho_0,\rho_1)$ satisfies the unbalanced continuity equation with initial and terminal densities ρ_0 and ρ_1, respectively,

$$\mathcal{C}(\rho_0,\rho_1):=\big\{(\rho,\boldsymbol{v},f):\partial_t\rho(\boldsymbol{x},t)+\nabla\cdot\big(\rho(\boldsymbol{x},t)\boldsymbol{v}(\boldsymbol{x},t)\big)=\mathcal{S}(\rho,f),\rho(\cdot,0)=\rho_0,\rho(\cdot,1)=\rho_1\big\}. \tag{1.4}$$

When $f\equiv 0$, the dynamic UOT problem simplifies to the dynamic OT problem. Several authors have considered density-independent source terms [8,14,21,22]. Gangbo *et al.* [8] consider $\mathcal{R}(f)$ to be density-independent and the source function $f(t)$ to be space-independent

$$\mathcal{R}(f)=\int_0^1|f(t)|^p\mathrm{d}t\cdot|\Omega|,\quad \mathcal{S}(f)=f(t).$$

They derive the corresponding unnormalized Monge problem, Monge-Ampére equation, and Kantorovich formulation, and introduce a primal-dual algorithm for the numerical solution of this dynamic UOT problem. Lee *et al.* [14] extend this UOT model in [8] by using a spatially dependent source function $f(\boldsymbol{x},t)$

$$\mathcal{R}(f)=\int_0^1\int_\Omega|f(\boldsymbol{x},t)|^p\mathrm{d}\boldsymbol{x}\mathrm{d}t,\quad \mathcal{S}(f)=f(\boldsymbol{x},t).$$

They design fast algorithms for L^1 and L^2 generalized unnormalized optimal transport using a primal-dual algorithm and a Nesterov accelerated gradient descent method, respectively. In this work, we mainly study the dynamic UOT problem as introduced by Chizat *et al.* [4, 5]. In this model, $\mathcal{R}(\rho,f)$ is density-dependent and $f(\boldsymbol{x},t)$ is space-dependent

$$\mathcal{R}(\rho,f)=\int_0^1\int_\Omega f(\boldsymbol{x},t)^2\rho(\boldsymbol{x},t)\mathrm{d}x\mathrm{d}t,\quad \mathcal{S}(\rho,f)=f(\boldsymbol{x},t)\rho(\boldsymbol{x},t).$$

This UOT model is defined on the Wasserstein-Fisher-Rao metric, also known as Hellinger-Kantorovich, initially and independently proposed by [4,11], providing an interpolation between the Wasserstein distance and the Fisher-Rao metric for measuring unequal masses. Chizat *et al.* [4], theoretically establish the existence of solutions to this variational problem. They employ staggered grid discretization techniques and solve this problem using the first-order proximal splitting method, demonstrating its application to image interpolation. In [5], the authors further generalize this model and propose a family of dynamic UOT problems. In addition, they formulate a corresponding static Kantorovich formulation and establish the equivalence between dynamic and static formulations. Building on this UOT model, Chen *et al.* [3] propose the addition of a diffusion term, $\sigma\Delta\rho$, into the source term $\mathcal{S}(\rho,f)$. This incorporates both advection and diffusion into the transport process. In their model, $f(\boldsymbol{x},t)$ is a relative source function with a specified indicator function. They have developed a detailed numerical algorithm to implement this model and applied it to modeling the fluid flow in the brain. Lombardi *et al.* [16] propose a genetic source term and consider three distinct types of source terms specifically designed for interpolating growing tumor images. Additionally, they further prove the existence and uniqueness of solutions for two of these terms, providing numerical evidence of their convergence and ability to handle mass growth in various test cases. The Eulerian discretization-based methods mentioned above are effective in low-dimensional cases but cannot be readily extended to high dimensions due to the exponential growth in the number of samples required to achieve a given tolerance as the dimensionality increases.

In recent years, deep learning has demonstrated remarkable success on a variety of computational problems. Jing *et al.* [9] develop a deep learning framework to compute the geodesics under the spherical Wasserstein-Fisher-Rao metric, where f is restricted to the family of zero mean functions. The learned geodesics can be adopted to generate weighted samples from some target distribution. Recently, TrajectoryNet [31] was proposed for learning cellular dynamics. Its loss function is carefully designed by incorporating multiple biological priors to perform an unbalanced optimal transport. Using a neural network model, TrajectoryNet captures continuous changes in cellular states over time and simulates cell proliferation and death processes through unbalanced optimal transport. Sha *et al.* [28] introduce TIGON, a novel dynamic unbalanced optimal transport algorithm that integrates deep learning with the Wasserstein-Fisher-Rao metric. This method simultaneously reconstructs cellular dynamic trajectories, models population growth, and infers the underlying gene regulatory network from time-series single-cell RNA sequencing snapshots. These approaches offer new insights into the understanding of cellular differentiation and development. In [34], we propose a novel deep learning approach for addressing the high-dimensional dynamic OT problem. We adopt the Lagrangian discretization to solve the continuity equation and use Monte Carlo sampling to approximate the high-dimensional integrals rather than Eulerian discretization-based methods. Moreover, we carefully design neural networks to parametrize the velocity field, achieving more accurate results in high-dimensional cases with excellent scalability.

In this paper, our first contribution is the introduction of a novel neural network approach to address the high-dimensional dynamic UOT problem used with the Wasserstein-Fisher-Rao metric. In contrast to our previous method [34], which was designed for high-dimensional dynamic OT problems, the UOT problem introduces additional complexities by incorporating a source term into both the objective function and the continuity equation. There are three main differences compared to the OT problem:

(i) To alleviate the dynamic formulation, we employ the generalized Kullback-Leibler (GKL) divergence to constrain the terminal density, rather than the standard Kullback-Leibler (KL) divergence, which is ineffective for the UOT problem.

(ii) To overcome the curse of dimensionality, we utilize Lagrangian discretization to solve the unbalanced PDE constraint and employ Monte Carlo sampling to approximate the high-dimensional integral. However, this introduces the need to solve an additional ordinary differential equation (ODE) compared to the OT problem.

(iii) In addition to parameterizing the velocity field, the source function also requires parameterization which increase the number of parameters.

Our second contribution is the good scalability of our proposed framework in high dimensions with constant number of samples. Furthermore, it can be easily extended to more general applications, such as Gaussian mixtures and crowd motion problems with different total masses.

The structure of this paper is as follows. In Section 2, we introduce the dynamic UOT problem and the GKL divergence. Section 3 presents the Lagrangian discretization approach. The neural network architecture is described in Section 4. Section 5 provides the experimental results along with implementation details. Finally, we conclude this work in Section 6.

2 Dynamic UOT and generalized Killback-Leibler divergence

In this work, we mainly study the dynamic UOT problem as introduced by Chizat *et al.* [4, 5]. Given two densities ρ_0 and ρ_1, the fundamental problem can be formulated as minimizing the Wasserstein-Fisher-Rao metric

$$\begin{aligned}&\min_{(\rho,v,f)\in\mathcal{C}(\rho_0,\rho_1)}\int_0^1\int_\Omega\|\boldsymbol{v}(\boldsymbol{x},t)\|^2\rho(\boldsymbol{x},t)\mathrm{d}\boldsymbol{x}\mathrm{d}t+\alpha\int_0^1\int_\Omega f(\boldsymbol{x},t)^2\rho(\boldsymbol{x},t)\mathrm{d}\boldsymbol{x}\mathrm{d}t\\&=:\mathcal{E}(\rho,\boldsymbol{v})+\alpha\mathcal{R}(\rho,f),\end{aligned}\tag{2.1}$$

where $\rho:\Omega\times[0,1]\to\mathbb{R}$ is a time-dependent density, $\boldsymbol{v}:\Omega\times[0,1]\to\mathbb{R}^d$ is a velocity field that describes the movement of mass and $f:\Omega\times[0,1]\to\mathbb{R}$ represents a spatially dependent source function that captures local growth and destruction of mass. In addition, α is a positive constant, and $(\rho,v,f)\in\mathcal{C}(\rho_0,\rho_1)$ satisfies the unbalanced continuity equation with the initial and terminal densities denoted as ρ_0 and ρ_1, respectively,

$$\begin{aligned}\mathcal{C}(\rho_0,\rho_1):=\big\{(\rho,v,f):&\partial_t\rho(\boldsymbol{x},t)+\nabla\cdot\big(\rho(\boldsymbol{x},t)\boldsymbol{v}(\boldsymbol{x},t)\big)=f(\boldsymbol{x},t)\rho(\boldsymbol{x},t),\\&\rho(\cdot,0)=\rho_0,\rho(\cdot,1)=\rho_1\big\}.\end{aligned}\tag{2.2}$$

In particular, when $f(x,t)\equiv 0$, the dynamic UOT problem reduces to the dynamic OT problem.

To efficiently solve the dynamic UOT problem, we first relax the constraint on the terminal density $\rho(\cdot,1)=\rho_1$ by transforming it into an implicit condition. Unlike the standard OT problem, where the total mass is conserved and $\rho(\cdot,1)$ represents a normalized probability distribution, the UOT problem allows for the creation or destruction of mass. This means that during the transportation process, the total mass can change, leading to a terminal density $\rho(\cdot,1)$ that may not by a normalized probability distribution. To address this issue, instead of using the original KL divergence to enforce the terminal density constraint in the OT problem, we apply the generalized Kullback-Leibler (GKL) divergence for the dynamic UOT problem.

Let $p(x)$ and $q(x)$ be unnormalized probability distributions. The GKL divergence [18] between $p(x)$ and $q(x)$ is defined by

$$\mathbf{GKL}\big(p(x)||q(x)\big):=\int_\Omega \phi\left(\frac{q(x)}{p(x)}\right)p(x)\mathrm{d}x, \quad \phi(r):=-\log r+r-1.$$

The key properties of the GKL divergence are as follows: (a) Gibbs' inequality holds: $\mathbf{GKL}(p(x)||q(x))\geq 0$, and (b) $\mathbf{GKL}(p(x)||q(x))=0$ if and only if $p(x)=q(x)$ holds almost everywhere for all x. The GKL divergence is general because for normalized probability distributions $p(x)$ and $q(x)$, it reduces to the original KL divergence

$$\mathbf{GKL}\big(p(x)||q(x)\big)=\int_\Omega p(x)\log\frac{p(x)}{q(x)}\mathrm{d}x=:\mathbf{KL}\big(p(x)||q(x)\big).$$

Therefore, by relaxing the terminal density constraint $\rho(\cdot,1)=\rho_1$ as an implicit condition through the introduction of the GKL divergence term

$$\mathcal{P}(\rho):=\mathbf{GKL}\big(\rho(x,1)||\rho_1(x)\big)=\int_\Omega \rho(x,1)\log\frac{\rho(x,1)}{\rho_1(x)}-\rho(x,1)+\rho_1(x)\mathrm{d}x,$$

we obtain the following optimal control formulation:

$$\min_{(\rho,v,f)\in\mathcal{C}(\rho_0)} \mathcal{F}(\rho,v,f)=\mathcal{E}(\rho,v)+\alpha\mathcal{R}(\rho,f)+\lambda\mathcal{P}(\rho), \tag{2.3}$$

where

$$\mathcal{C}(\rho_0):=\big\{(\rho,v,f):\partial_t\rho(x,t)+\nabla\cdot\big(\rho(x,t)v(x,t)\big)=f(x,t)\rho(x,t),\rho(\cdot,0)=\rho_0\big\}.$$

Additionally, λ serves as the penalty parameter.

3 Lagrangian formulation

To solve UOT in high-dimensional space, we solve the transport equation

$$\partial_t\rho(x,t)+\nabla\cdot\big(\rho(x,t)v(x,t)\big)=f(x,t)\rho(x,t)$$

along the characteristics. The characteristics can be obtained by solving an ODE system

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{z}(x,t)=v\big(\mathbf{z}(x,t),t\big),\quad \mathbf{z}(x,0)=x. \tag{3.1}$$

Then the original PDE can be transformed to an ODE system along the characteristics

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}t}\rho\big(\mathbf{z}(x,t),t\big)=-\rho\big(\mathbf{z}(x,t),t\big)\nabla\cdot v\big(\mathbf{z}(x,t),t\big)+f\big(\mathbf{z}(x,t),t\big)\rho\big(\mathbf{z}(x,t),t\big),\\&\rho\big(\mathbf{z}(x,0),0\big)=\rho_0(x).\end{aligned} \tag{3.2}$$

Diving ρ on both sides, above ODE can be simplified further

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}t}\ln\rho\big(\mathbf{z}(x,t),t\big)=-\nabla\cdot v\big(\mathbf{z}(x,t),t\big)+f\big(\mathbf{z}(x,t),t\big),\\&\ln\rho\big(\mathbf{z}(x,0),0\big)=\ln\rho_0(x).\end{aligned} \tag{3.3}$$

Note that (3.2) and (3.3) are equivalent, however (3.3) offers some advantages in numerical computation. In high-dimensional cases, since ρ_0 is a positive number very close to zero, solving Eq. (3.2) can lead to numerical overflow issues. Therefore, in practice, we solve Eq. (3.3) instead.

In addition, we define an auxiliary variable $\mu(x,t)$ which satisfies following PDE:

$$\partial_t\mu(x,t)+\nabla\cdot\big(\mu(x,t)v(x,t)\big)=0,\quad \mu(x,0)=\mu_0(x).$$

μ_0 is a given density. Then μ preserves to be density for $t>0$. Similarly to ρ,μ can be solved along characteristic also.

$$\frac{\mathrm{d}}{\mathrm{d}t}\ln\mu\big(\mathbf{z}(x,t),t\big)=-\nabla\cdot v\big(\mathbf{z}(x,t),t\big),\quad \ln\mu\big(\mathbf{z}(x,0),0\big)=\ln\mu_0(x). \tag{3.4}$$

Here, $\mathbf{z}(x,t)$ represents the characteristics solved in (3.1).

From (3.1) and (3.4), it is well known that

$$\mu\big(\mathbf{z}(x,t),t\big)\det\big(\nabla\mathbf{z}(x,t)\big)=\mu_0(x),\quad \forall t\in[0,1]. \tag{3.5}$$

Now, we can rewrite the objective function in Lagrangian form. First,

$$\begin{aligned}\mathcal{E}(\rho,v)&=\int_0^1\int_\Omega\|v(x,t)\|^2\rho(x,t)\mathrm{d}x\mathrm{d}t\\&=\int_0^1\int_\Omega\|v(x,t)\|^2\frac{\rho(x,t)}{\mu(x,t)}\mu(x,t)\mathrm{d}x\mathrm{d}t\\&=\int_0^1\int_\Omega\big\|v\big(\mathbf{z}(x,t),t\big)\big\|^2\frac{\rho\big(\mathbf{z}(x,t),t\big)}{\mu\big(\mathbf{z}(x,t),t\big)}\mu\big(\mathbf{z}(x,t),t\big)\det\big(\nabla\mathbf{z}(x,t)\big)\mathrm{d}x\mathrm{d}t\\&=\int_0^1\int_\Omega\big\|v\big(\mathbf{z}(x,t),t\big)\big\|^2\frac{\rho\big(\mathbf{z}(x,t),t\big)}{\mu\big(\mathbf{z}(x,t),t\big)}\mu_0(x)\mathrm{d}x\mathrm{d}t\\&=\int_0^1\mathbb{E}_{x\sim\mu_0}\left[\big\|v\big(\mathbf{z}(x,t),t\big)\big\|^2\frac{\rho\big(\mathbf{z}(x,t),t\big)}{\mu\big(\mathbf{z}(x,t),t\big)}\right]\mathrm{d}t.\end{aligned}$$

Similarly, we can rewrite other two terms in the objective function

$$
\begin{aligned}
\mathcal{R}(\rho,f) &= \int_0^1 \int_\Omega f(x,t)^2 \rho(x,t) \mathrm{d}x\mathrm{d}t \\
&= \int_0^1 \int_\Omega f(x,t)^2 \frac{\rho(x,t)}{\mu(x,t)} \mu(x,t) \mathrm{d}x\mathrm{d}t \\
&= \int_0^1 \int_\Omega f\big(\mathbf{z}(x,t),t\big)^2 \frac{\rho\big(\mathbf{z}(x,t),t\big)}{\mu\big(\mathbf{z}(x,t),t\big)} \mu\big(\mathbf{z}(x,t),t\big) \det\big(\nabla \mathbf{z}(x,t)\big) \mathrm{d}x\mathrm{d}t \\
&= \int_0^1 \int_\Omega f\big(\mathbf{z}(x,t),t\big)^2 \frac{\rho\big(\mathbf{z}(x,t),t\big)}{\mu\big(\mathbf{z}(x,t),t\big)} \mu_0(x) \mathrm{d}x\mathrm{d}t \\
&= \int_0^1 \mathbb{E}_{x\sim\mu_0} \left[f\big(\mathbf{z}(x,t),t\big)^2 \frac{\rho\big(\mathbf{z}(x,t),t\big)}{\mu\big(\mathbf{z}(x,t),t\big)} \right] \mathrm{d}t
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{P}(\rho) &= \int_\Omega \rho(x,1) \log \frac{\rho(x,1)}{\rho_1(x)} - \rho(x,1) + \rho_1(x) \mathrm{d}x \\
&= \int_\Omega \left[\frac{\rho(x,1)}{\mu(x,1)} \log \frac{\rho(x,1)}{\rho_1(x)} - \frac{\rho(x,1)}{\mu(x,1)} + \frac{\rho_1(x)}{\mu(x,1)} \right] \mu(x,1) \mathrm{d}x \\
&= \int_\Omega \left[\frac{\rho\big(\mathbf{z}(x,1),1\big)}{\mu\big(\mathbf{z}(x,1),1\big)} \log \frac{\rho\big(\mathbf{z}(x,1),1\big)}{\rho_1\big(\mathbf{z}(x,1)\big)} - \frac{\rho\big(\mathbf{z}(x,1),1\big)}{\mu\big(\mathbf{z}(x,1),1\big)} + \frac{\rho_1\big(\mathbf{z}(x,1)\big)}{\mu\big(\mathbf{z}(x,1),1\big)} \right] \\
&\quad \times \mu\big(\mathbf{z}(x,1),1\big) \det\big(\nabla \mathbf{z}(x,1)\big) \mathrm{d}x \\
&= \int_\Omega \left[\frac{\rho\big(\mathbf{z}(x,1),1\big)}{\mu\big(\mathbf{z}(x,1),1\big)} \log \frac{\rho\big(\mathbf{z}(x,1),1\big)}{\rho_1\big(\mathbf{z}(x,1)\big)} - \frac{\rho\big(\mathbf{z}(x,1),1\big)}{\mu\big(\mathbf{z}(x,1),1\big)} + \frac{\rho_1\big(\mathbf{z}(x,1)\big)}{\mu\big(\mathbf{z}(x,1),1\big)} \right] \mu_0(x) \mathrm{d}x \\
&= \mathbb{E}_{x\sim\mu_0} \left[\frac{\rho\big(\mathbf{z}(x,1),1\big)}{\mu\big(\mathbf{z}(x,1),1\big)} \log \frac{\rho\big(\mathbf{z}(x,1),1\big)}{\rho_1\big(\mathbf{z}(x,1)\big)} - \frac{\rho\big(\mathbf{z}(x,1),1\big)}{\mu\big(\mathbf{z}(x,1),1\big)} + \frac{\rho_1\big(\mathbf{z}(x,1)\big)}{\mu\big(\mathbf{z}(x,1),1\big)} \right].
\end{aligned}
$$

With the help of Lagrangian formulation, we write the objective function in the form of expectation. This is important to calculate the objective function especially in high-dimensional space, since Monte-Carlo method is almost the only effective method to calculate high-dimensional integrals.

Using Monte-Carlo method, we get the following semi-discretized version of the dynamic UOT problem:

$$
\min_{v,f} \overline{\mathcal{F}}(\rho,v,f) := \overline{\mathcal{E}}(\rho,v) + \alpha \overline{\mathcal{R}}(\rho,f) + \lambda \overline{\mathcal{P}}(\rho), \tag{3.6}
$$

where $\overline{\mathcal{E}}(\rho,v), \overline{\mathcal{P}}(\rho)$ and $\overline{\mathcal{R}}(\rho,f)$ denote the discretized versions of $\mathcal{E}(\rho,v), \mathcal{P}(\rho)$ and $\mathcal{R}(\rho,f)$ in the spatial direction, respectively, i.e.

$$
\overline{\mathcal{E}}(\rho,v) := \int_0^1 \sum_{i=1}^r \frac{1}{r} \big\| v\big(\mathbf{z}(x_i,t),t\big) \big\|^2 \frac{\rho\big(\mathbf{z}(x_i,t),t\big)}{\mu\big(\mathbf{z}(x_i,t),t\big)} \mathrm{d}t, \tag{3.7}
$$

$$\overline{\mathcal{R}}(\rho,f):=\int_0^1\sum_{i=1}^r\frac{1}{r}f\big(\mathbf{z}(\boldsymbol{x}_i,t),t\big)^2\frac{\rho\big(\mathbf{z}(\boldsymbol{x}_i,t),t\big)}{\mu\big(\mathbf{z}(\boldsymbol{x}_i,t),t\big)}\mathrm{d}t, \tag{3.8}$$

$$\overline{\mathcal{P}}(\rho):=\sum_{i=1}^r\frac{1}{r}\left[\frac{\rho\big(\mathbf{z}(\boldsymbol{x}_i,1),1\big)}{\mu\big(\mathbf{z}(\boldsymbol{x}_i,1),1\big)}\log\frac{\rho\big(\mathbf{z}(\boldsymbol{x}_i,1),1\big)}{\rho_1\big(\mathbf{z}(\boldsymbol{x}_i,1)\big)}-\frac{\rho\big(\mathbf{z}(\boldsymbol{x}_i,1),1\big)}{\mu\big(\mathbf{z}(\boldsymbol{x}_i,1),1\big)}+\frac{\rho_1\big(\mathbf{z}(\boldsymbol{x}_i,1)\big)}{\mu\big(\mathbf{z}(\boldsymbol{x}_i,1),1\big)}\right], \tag{3.9}$$

where $\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_r\in\mathbb{R}^d$ are random samples independently drawn from μ_0. $\rho(\mathbf{z}(\boldsymbol{x}_i,t),t)$ and $\mu(\mathbf{z}(\boldsymbol{x}_i,t),t)$ can be obtained from the solution of (3.3) and (3.4) along the characteristics $\mathbf{z}(\boldsymbol{x}_i,t)$. Then, (3.1), (3.3) and (3.4) can be reformulated as the following constraints:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{z}(\boldsymbol{x}_i,t)=\boldsymbol{v}\big(\mathbf{z}(\boldsymbol{x}_i,t),t\big),\quad \mathbf{z}(\boldsymbol{x}_i,0)=\boldsymbol{x}_i,\quad \boldsymbol{x}_i\sim\mu_0(\boldsymbol{x}), \tag{3.10}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\ln\rho\big(\mathbf{z}(\boldsymbol{x}_i,t),t\big)=-\nabla\cdot\boldsymbol{v}\big(\mathbf{z}(\boldsymbol{x}_i,t),t\big)+f\big(\mathbf{z}(\boldsymbol{x}_i,t),t\big),\quad \ln\rho\big(\mathbf{z}(\boldsymbol{x}_i,0),0\big)=\ln\rho_0(\boldsymbol{x}_i), \tag{3.11}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\ln\mu\big(\mathbf{z}(\boldsymbol{x}_i,t),t\big)=-\nabla\cdot\boldsymbol{v}\big(\mathbf{z}(\boldsymbol{x}_i,t),t\big),\quad \ln\mu\big(\mathbf{z}(\boldsymbol{x}_i,0),0\big)=\ln\mu_0(\boldsymbol{x}_i). \tag{3.12}$$

In our numerical implementation, we simply set $\mu_0=\rho_0$. To get the fully discretized model, we need to solve ODEs along time and compute integrals in time. The above three ODE constraints, (3.10)-(3.12), are solved using the classical fourth-order explicit Runge-Kutta scheme (RK4). The time integral in (3.6) is computed using the standard composite Simpson's formula.

4 Neural network architectures for the velocity field and source function

To address high-dimensional UOT problems, we parameterize both the velocity field $\boldsymbol{v}$ and the source function f using neural networks, leveraging the ability of neural networks to approximate high dimensional functions.

The representation is split into spatial and temporal components. For the temporal component, we use simple linear finite element basis functions, while for the spatial component, we employ a standard multi-layer perceptron (MLP) network. More specifically, for the velocity field $\boldsymbol{v}$, our model architecture is as follows:

$$\boldsymbol{v}(\boldsymbol{x},t;\theta)=\sum_{i=0}^M\boldsymbol{v}_i(\boldsymbol{x};\theta)\phi_i(t), \tag{4.1}$$

where $(\boldsymbol{x},t)\in\mathbb{R}^{d+1}$ represents the input feature, and the basis functions $\phi_i(t)$ are defined as

$$\phi_i(t)=\begin{cases}0, & t\le t_{i-1},\\ \dfrac{t-t_{i-1}}{t_i-t_{i-1}}=M(t-t_{i-1}), & t_{i-1}<t\le t_i,\\ \dfrac{t_{i+1}-t}{t_{i+1}-t_i}=M(t_{i+1}-t), & t_i<t\le t_{i+1},\\ 0, & t>t_{i+1}.\end{cases}$$

Here, the number of the basis functions is $M+1$. Once M is determined, we can set the number of time steps N to kM, such as $k=2$, ensuring that the forward ODEs (3.1), (3.3) and (3.4) can be solved stably and accurately.

Additionally, $\boldsymbol{v}_i(\boldsymbol{x};\theta)$ is the sum of some two-layer MLP networks, defined by

$$\boldsymbol{v}_i(\boldsymbol{x};\theta)=\sum_{l=1}^{L}\left[W_{2i}^{(l)}\sigma\left(W_{1i}^{(l)}\boldsymbol{x}+b_{1i}^{(l)}\right)+b_{2i}^{(l)}\right].$$

Here, the network parameters are denoted by $\theta=\{W_{1i}^{(l)},W_{2i}^{(l)},b_{1i}^{(l)},b_{2i}^{(l)}\}$, where $W_{1i}^{(l)}\in\mathbb{R}^{H\times d}$, $W_{2i}^{(l)}\in\mathbb{R}^{d\times H},b_{1i}^{(l)}\in\mathbb{R}^{H},b_{2i}^{(l)}\in\mathbb{R}^{d}$. H represents the number of hidden units, L denotes the width of the network, and $\sigma(\boldsymbol{x})=\tanh(\boldsymbol{x})$ serves as the activation function.

The advantage of the proposed architecture is that it enhances representation ability and reduces oscillations in the time direction by utilizing piecewise linear function as the basis functions. A detailed analysis of this effect can be found in our previous work [34].

Unlike the standard OT problem, in the dynamic UOT problem, the additional source function $f(\boldsymbol{x},t;\tilde{\theta})$ is also parameterized using basis functions as follows:

$$f(\boldsymbol{x},t;\tilde{\theta})=\mathbf{w}^{\top}\boldsymbol{u}(\boldsymbol{x},t;\hat{\theta})+b,$$

where $\boldsymbol{u}(\boldsymbol{x},t;\hat{\theta})$ follows the same architecture as $\boldsymbol{v}$ in (4.1). The parameters of the network f are given by $\tilde{\theta}=\{\hat{\theta},\mathbf{w},b\}$, with $\mathbf{w}\in\mathbb{R}^{d}$, $b\in\mathbb{R}$, and $\hat{\theta}=\{\hat{W}_{1i}^{(l)},\hat{W}_{2i}^{(l)},\hat{b}_{1i}^{(l)},\hat{b}_{2i}^{(l)}\}$.

As a result, the original optimization problem (3.6) concerning the velocity field $\boldsymbol{v}$ and the source function f can be reformulated into an optimization problem focused on the network parameters. To compute the gradient of the objective function with respect to the parameters θ and $\tilde{\theta}$, we employ the standard back propagation method enabled by automatic differentiation. Consequently, the parameters can be optimized using gradient descent and its variant. We summarize the proposed UOT algorithm in Algorithm 1.

5 Numerical experiments

In this section, we illustrate the effectiveness of the proposed method. The numerical examples include Gaussian examples, Gaussian mixture examples and crowd motion examples. Note that when the means of the densities ρ_0 and ρ_1 are not identical, we first set the source function $f=0$ and employ the KL divergence to find the target mean, a process referred to as initialization.

In our experiments, unless otherwise stated, we adopt the following default settings: the time steps $N=10$, the GKL penalty parameter $\lambda=10000$, and the unbalanced coefficient $\alpha=100$. The parameters of the network structure are as follows: the number of the basis $M=5$, the width of neural network $L=2$, and the number of hidden units $H=10$. All models are trained for 1000 epochs (training iterations) using the Adam optimizer and a consistent batch size of 1024 training samples for all high-dimensional cases. The initial learning rate is set to 0.01, with a decay factor of 0.98 applied every 10 epochs. All

Algorithm 1. UOT Algorithm.

```
Input: Initial density $\rho_0$, terminal density $\rho_1$, time steps $N$, network structure parameters $L,M,H$, hyperparameters $\lambda$ and $\alpha$.
Initialize: Parameters $\theta$ of the velocity field $v$ and $\tilde{\theta}$ of the source function $f$.
for $epoch = 1,2,\dots,1000$ do
  Randomly draw samples $x_1,x_2,\dots,x_r$ from $\mu_0$.
  Set $t_j = j/N$.
  for $j = 0$ to $N$ do
    Solve $\mathbf{z}(x_i,t_j)$ using RK4 by Eq. (3.10).
    Solve $\rho(\mathbf{z}(x_i,t_j),t_j)$ using RK4 by Eq. (3.11).
    Solve $\mu(\mathbf{z}(x_i,t_j),t_j)$ using RK4 by Eq. (3.12).
  end for
  Compute $\overline{\mathcal{E}}(\rho,v)$, $\overline{\mathcal{R}}(\rho,f)$ and $\overline{\mathcal{P}}(\rho)$ using standard composite Simpson formula by Eqs. (3.7), (3.8), and (3.9), respectively.
  Compute gradient of the objective function $\overline{\mathcal{F}}(\rho,v,f)$ with respect to parameters $\theta$ and $\tilde{\theta}$ using automatic differentiation.
  Update $\theta$ and $\tilde{\theta}$ by using the gradient descent method.
end for
Output: Optimized parameters $\theta$ and $\tilde{\theta}$.
```

the experiments are conducted in PyTorch on an NVIDIA H800 PCIe GPU with 80 GB memory.

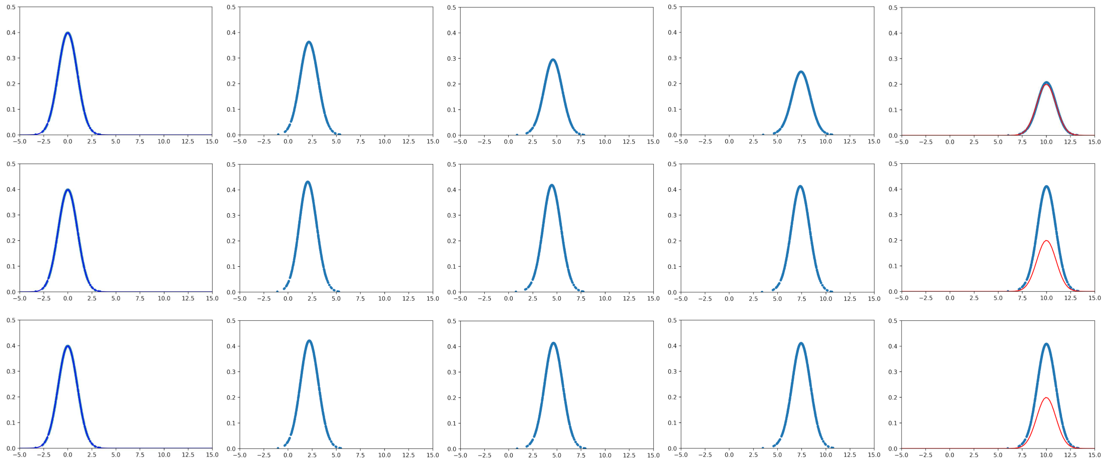
5.1 Examples of UOT with different weight

In this section, we discuss the coefficient α governing the source term in the dynamic UOT problem (2.1). This coefficient parameter regulates the creation and destruction of mass during transportation. When α approaches infinity, the UOT solution converges towards the classical OT solution in (1.1).

Figs. 1-3 numerically illustrate the density evolution using the proposed method, moving from the initial density (shown in blue in (a)) to the target density (shown in red in (e)) at one dimension ($d=1$). The first row illustrates the transportation process for UOT with $\alpha=100$, while the second row shows the process for UOT with $\alpha=10^6$. Lastly, the third row presents the transportation process for OT. We can clearly see that when the unbalanced coefficient α is sufficiently large, UOT performs similarly to OT, indicating a preference for transporting with equal mass.

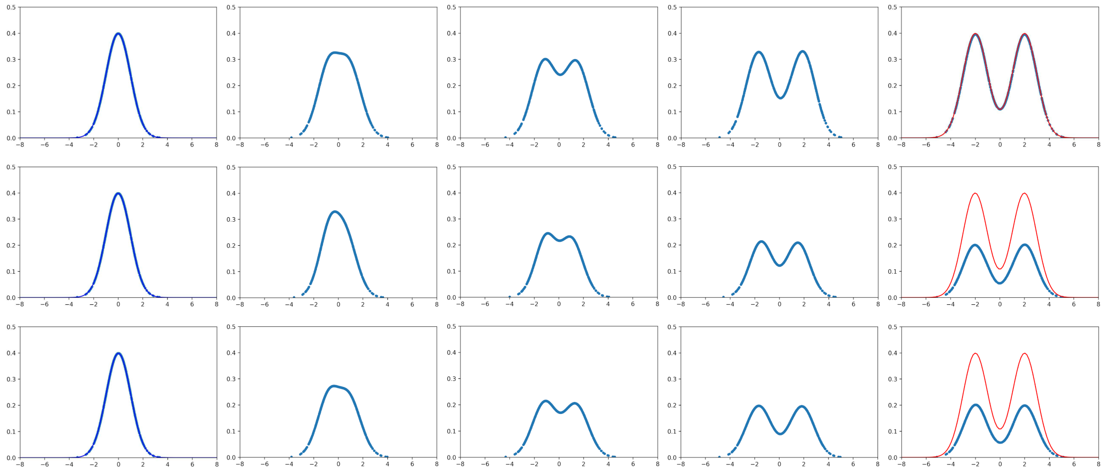
5.2 Gaussian examples

In this section, we consider the UOT problem on several typical Gaussian examples with different means and covariance matrices. Here, $\rho_G(\cdot,\mathbf{m},\mathbf{\Sigma})$ represents the probability den-



(a) $\rho_0(\boldsymbol{x}_0)$ (b) $\tilde{\rho}_{0.25}(\tilde{\boldsymbol{x}}_{0.25})$ (c) $\tilde{\rho}_{0.5}(\tilde{\boldsymbol{x}}_{0.5})$ (d) $\tilde{\rho}_{0.75}(\tilde{\boldsymbol{x}}_{0.75})$ (e) $\tilde{\rho}_1(\tilde{\boldsymbol{x}}_1)$

Figure 1: Top: UOT with $\alpha=100$, middle: UOT with $\alpha=10^6$, bottom: OT.



(a) $\rho_0(\boldsymbol{x}_0)$ (b) $\tilde{\rho}_{0.25}(\tilde{\boldsymbol{x}}_{0.25})$ (c) $\tilde{\rho}_{0.5}(\tilde{\boldsymbol{x}}_{0.5})$ (d) $\tilde{\rho}_{0.75}(\tilde{\boldsymbol{x}}_{0.75})$ (e) $\tilde{\rho}_1(\tilde{\boldsymbol{x}}_1)$

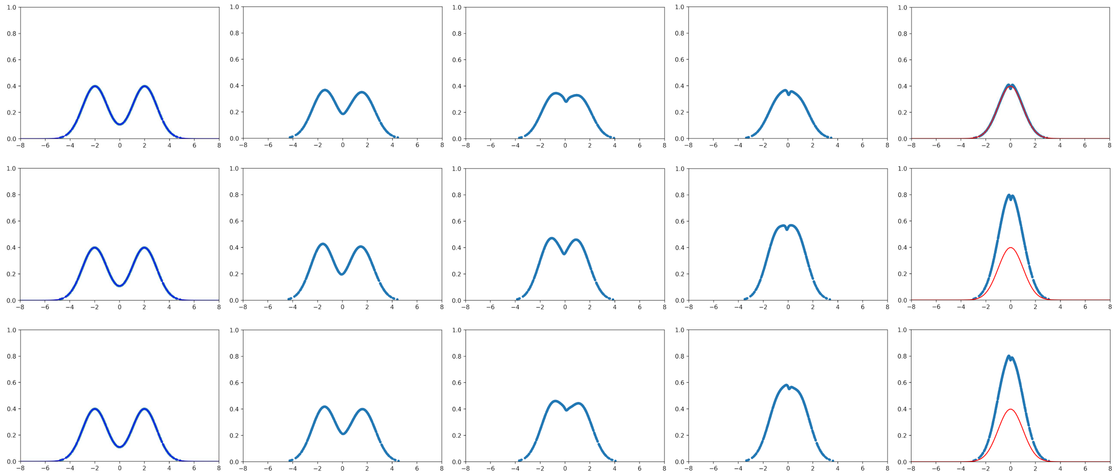
Figure 2: Top: UOT with $\alpha=100$, middle: UOT with $\alpha=10^6$, bottom: OT.

sity function of a d-dimensional Gaussian distribution with a mean vector $\mathbf{m}\in\mathbb{R}^d$ and covariance matrix $\mathbf{\Sigma}\in\mathbb{R}^{d\times d}$, i.e.

$$\rho_G(\boldsymbol{x},\mathbf{m},\mathbf{\Sigma})=\frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\mathbf{m})^\top\mathbf{\Sigma}^{-1}(\boldsymbol{x}-\mathbf{m})\right).$$

Below, we first provide specific settings for four unbalanced synthetic tests in $d=1$.

- Test 1: $\rho_0(x)=\rho_G(x,0,1)$, $\rho_1(x)=2\rho_G(x,0,1)$.
- Test 2: $\rho_0(x)=\rho_G(x,0,1)$, $\rho_1(x)=\dfrac{1}{2}\rho_G(x,0,1)$.



(a) $\rho_0(x_0)$ (b) $\tilde{\rho}_{0.25}(\tilde{x}_{0.25})$ (c) $\tilde{\rho}_{0.5}(\tilde{x}_{0.5})$ (d) $\tilde{\rho}_{0.75}(\tilde{x}_{0.75})$ (e) $\tilde{\rho}_1(\tilde{x}_1)$

Figure 3: Top: UOT with $\alpha=100$, middle: UOT with $\alpha=10^6$, bottom: OT.

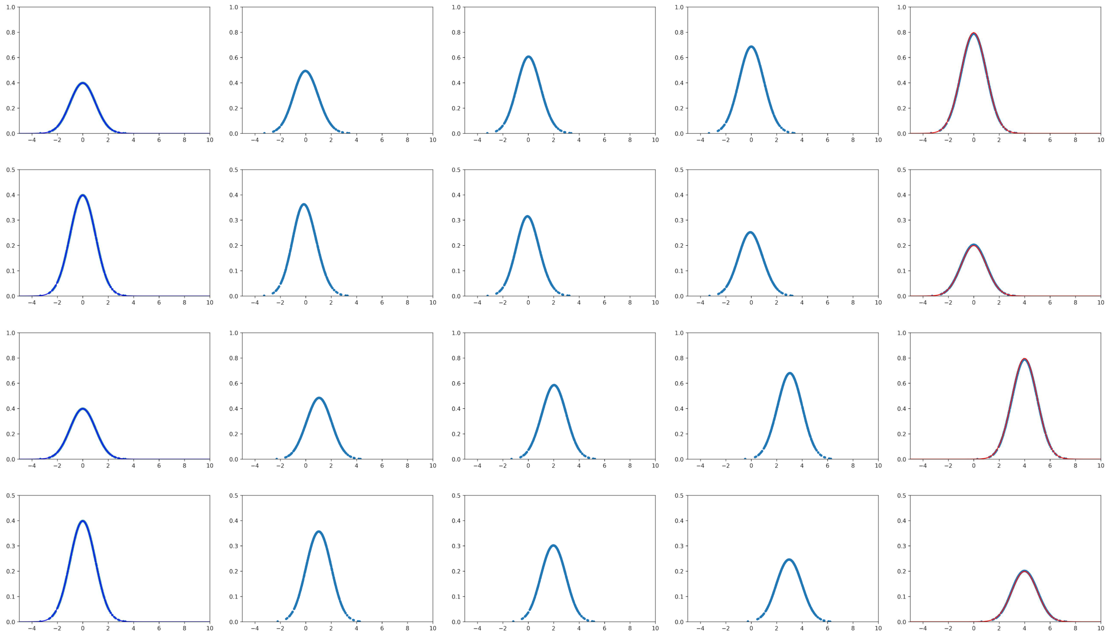
- Test 3: $\rho_0(x)=\rho_G(x,0,1)$, $\rho_1(x)=2\rho_G(x,4,1)$.
- Test 4: $\rho_0(x)=\rho_G(x,0,1)$, $\rho_1(x)=\dfrac{1}{2}\rho_G(x,4,1)$.

In Fig. 4, we illustrate the evolution of the density transition from the initial density ρ_0 in (a) to the target density ρ_1 in (e). We observe that the unbalanced movement is reasonable, and the target density estimated by our trained network (depicted in blue in (e)) closely matches the true target density ρ_1 (shown in red in (e)). This result demonstrates the effectiveness of our training procedure.

Secondly, we consider the UOT problem on four typical Gaussian examples in $d\geq 2$ as listed below.

- Test 5: $\rho_0(\boldsymbol{x})=\rho_G(\boldsymbol{x},\mathbf{0},\mathbf{I})$, $\rho_1(\boldsymbol{x})=\dfrac{1}{2}\rho_G(\boldsymbol{x},-4\cdot\mathbf{e_1},\mathbf{I})$.
- Test 6: $\rho_0(\boldsymbol{x})=\rho_G(\boldsymbol{x},\mathbf{0},\mathbf{I})$, $\rho_1(\boldsymbol{x})=2\rho_G(\boldsymbol{x},4\cdot\mathbf{e_1},\mathbf{I})$.
- Test 7: $\rho_0(\boldsymbol{x})=\rho_G(\boldsymbol{x},\mathbf{0},0.3\cdot\mathbf{I})$, $\rho_1(\boldsymbol{x})=2\rho_G(\boldsymbol{x},4\cdot\mathbf{e_1},0.3\cdot\mathbf{I})$.
- Test 8: $\rho_0(\boldsymbol{x})=\rho_G(\boldsymbol{x},-4\cdot\mathbf{e_1}-4\cdot\mathbf{e_2},\mathbf{I})$, $\rho_1(\boldsymbol{x})=\dfrac{1}{2}\rho_G(\boldsymbol{x},4\cdot\mathbf{e_1}+4\cdot\mathbf{e_2},\mathbf{I})$.

In Fig. 5, we present the transmission process in the two-dimensional case ($d=2$). The first column (a) and the last column (f) show the initial density and the target density, respectively. The other columns correspond to the push-forward of the initial density at intermediate times $t=0.25,0.5,0.75$, and 1.0, respectively. We can clearly observe that the



(a) $\rho_0(x_0)$ (b) $\tilde{\rho}_{0.25}(\tilde{x}_{0.25})$ (c) $\tilde{\rho}_{0.5}(\tilde{x}_{0.5})$ (d) $\tilde{\rho}_{0.75}(\tilde{x}_{0.75})$ (e) $\tilde{\rho}_1(\tilde{x}_1)$

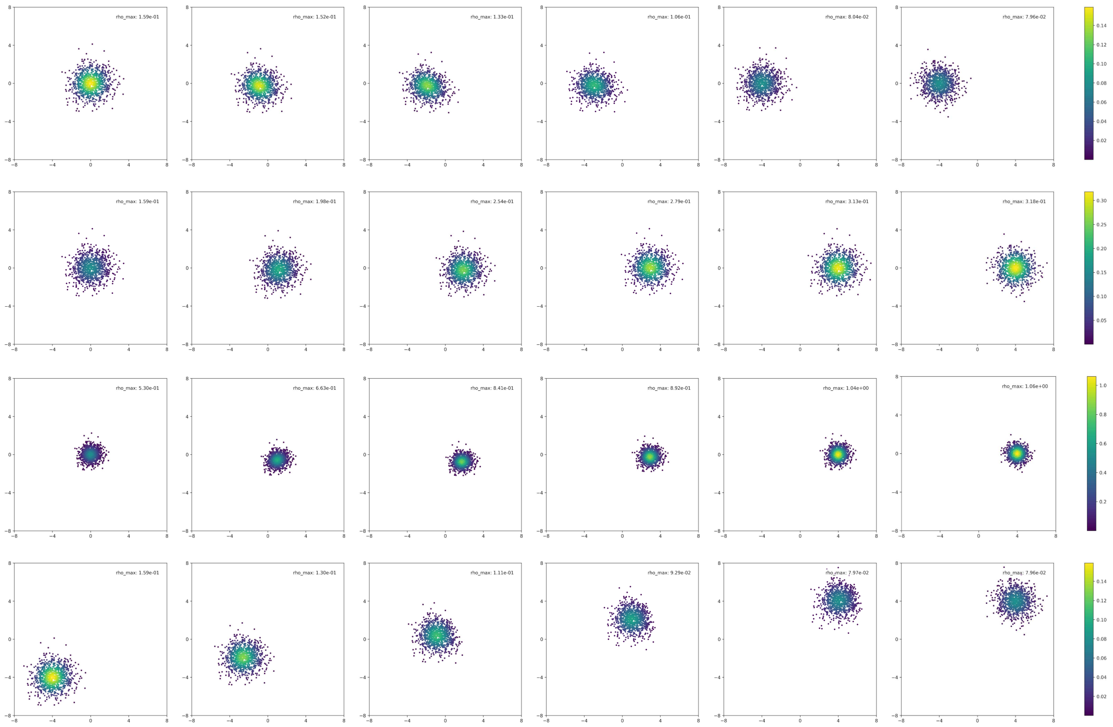
Figure 4: Illustration of these four Gaussian problems at $d=1$.

target density estimated by our trained network in (e) is very similar to the true target density in (f). Additionally, the changes in mass during the transmission process can be observed through the maximum density values of the samples at the top-right corner across different time points. Specifically, in Test 5 and Test 8, the maximum density value of the samples at $t=1$ is approximately half of that at $t=0$. In contrast, for Test 6 and Test 7, the maximum density value at $t=1$ is nearly double that at $t=0$. These observed unbalanced movements are both expected and reasonable.

Similar results can be observed across high-dimensional cases, as shown in Fig. 6 for $d=100$. Furthermore, we present the running time per epoch and memory usage for dimensions ranging from 2 to 100 in Fig. 7. As shown in the figure, memory usage increases linearly with dimensionality.

Moreover, we conduct experiments in higher dimensions with $d=300$ and $d=400$. For these cases, we find that increasing the batch size is necessary, as a batch size of 1024 samples is insufficient for higher dimensions. However, this results in memory constraints. Therefore, we set $L=1$ and use 2048 samples for $d=300$ and 3096 samples for $d=400$, respectively. The corresponding results are presented in Figs. 8 and 9.

For better illustration, we report the GKL divergence values across dimensions ranging from 2 to 400 in Table 1. It is evident that the proposed method consistently yields promising quantitative results, even as dimensionality increases significantly.



(a) $\rho_0(\boldsymbol{x}_0)$ (b) $\tilde{\rho}_{0.25}(\tilde{\boldsymbol{x}}_{0.25})$ (c) $\tilde{\rho}_{0.5}(\tilde{\boldsymbol{x}}_{0.5})$ (d) $\tilde{\rho}_{0.75}(\tilde{\boldsymbol{x}}_{0.75})$ (e) $\tilde{\rho}_1(\tilde{\boldsymbol{x}}_1)$ (f) $\rho_1(\boldsymbol{x}_1)$

Figure 5: Illustration of these Gaussian problems at $d=2$.

Table 1: Results of the GKL divergence term for each dimension.

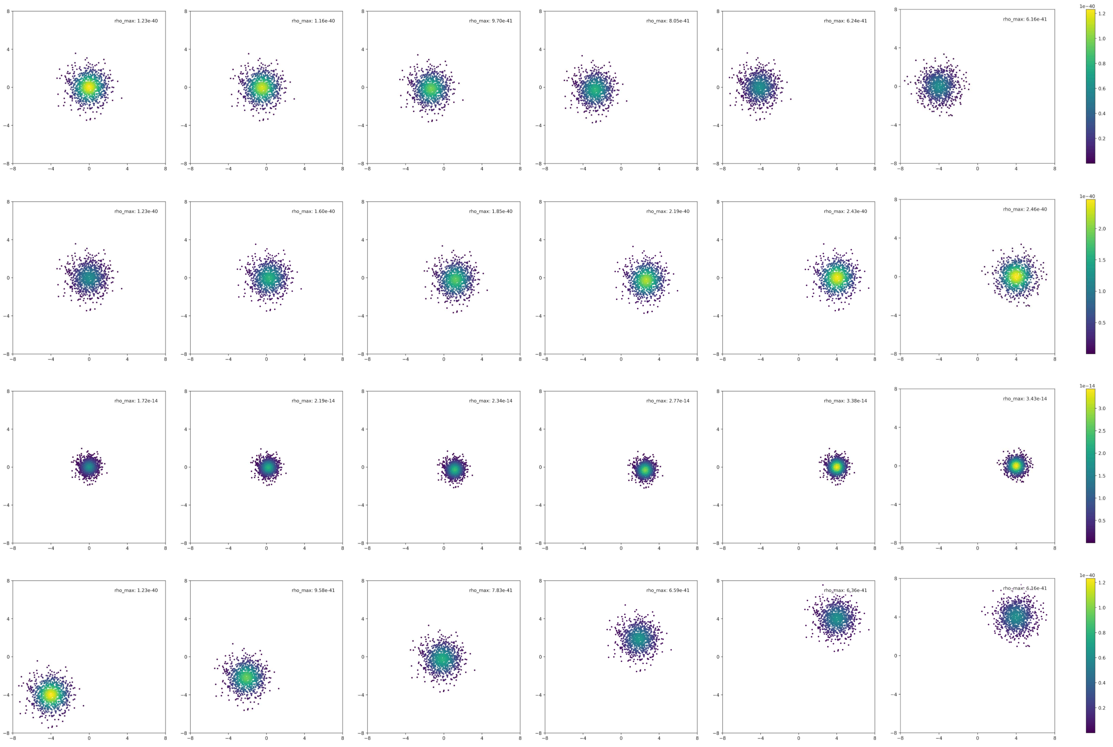
d	Samples	Test 5	Test 6	Test 7	Test 8	Time (s)
2	1024	1.79e-04	3.59e-04	6.61e-04	3.48e-03	1.1598
100	1024	1.30e-03	3.19e-03	2.77e-03	2.23e-03	1.4172
300	2048	1.49e-03	3.80e-03	3.83e-03	5.22e-03	1.6513
400	3096	2.36e-03	5.66e-03	3.43e-03	3.47e-03	2.4753

5.3 Gaussian mixtures

In this section, we consider UOT problem on two Gaussian mixtures examples as listed below.

- Test 9: $\rho_0(\boldsymbol{x})=\rho_G(\boldsymbol{x},\mathbf{0},\mathbf{I})$, $\rho_1(\boldsymbol{x})=\rho_G(\boldsymbol{x},-2\cdot\mathbf{e_1},\mathbf{I})+\rho_G(\boldsymbol{x},2\cdot\mathbf{e_1},\mathbf{I})$.
- Test 10: $\rho_0(\boldsymbol{x})=\rho_G(\boldsymbol{x},-2\cdot\mathbf{e_1},\mathbf{I})+\rho_G(\boldsymbol{x},2\cdot\mathbf{e_1},\mathbf{I})$, $\rho_1(\boldsymbol{x})=\rho_G(\boldsymbol{x},\mathbf{0},\mathbf{I})$.

In Figs. 10 and 11, we demonstrate that the proposed method effectively achieves the splitting from one to two (Test 9) and the merging from two to one (Test 10), along with



(a) $\rho_0(x_0)$ (b) $\tilde{\rho}_{0.25}(\tilde{x}_{0.25})$ (c) $\tilde{\rho}_{0.5}(\tilde{x}_{0.5})$ (d) $\tilde{\rho}_{0.75}(\tilde{x}_{0.75})$ (e) $\tilde{\rho}_1(\tilde{x}_1)$ (f) $\rho_1(x_1)$

Figure 6: Illustration of these Gaussian problems at $d=100$.

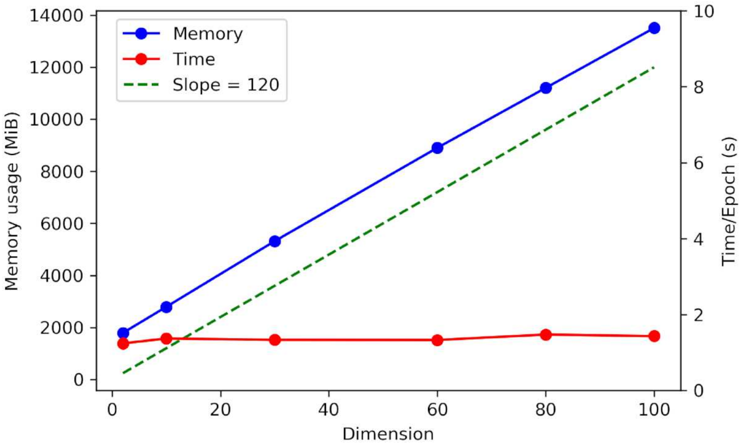
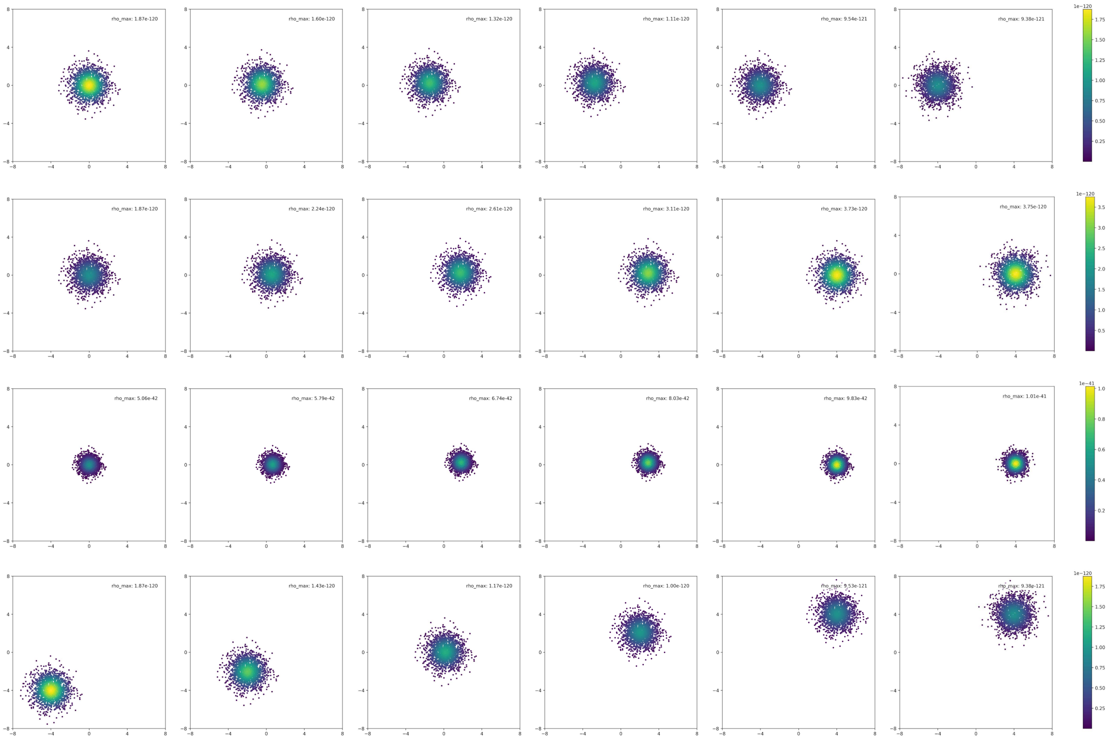


Figure 7: Memory usage across dimensions (2-100) with 1024 samples.

the creation and destruction in mass. Furthermore, we can clearly observe that the target density estimated by our trained network in (e) closely resembles the target density ρ_1 in (f), even in high-dimensional instances.



(a) $\rho_0(\boldsymbol{x}_0)$ (b) $\tilde{\rho}_{0.25}(\tilde{\boldsymbol{x}}_{0.25})$ (c) $\tilde{\rho}_{0.5}(\tilde{\boldsymbol{x}}_{0.5})$ (d) $\tilde{\rho}_{0.75}(\tilde{\boldsymbol{x}}_{0.75})$ (e) $\tilde{\rho}_1(\tilde{\boldsymbol{x}}_1)$ (f) $\rho_1(\boldsymbol{x}_1)$

Figure 8: Illustration of these Gaussian problems at $d=300$.

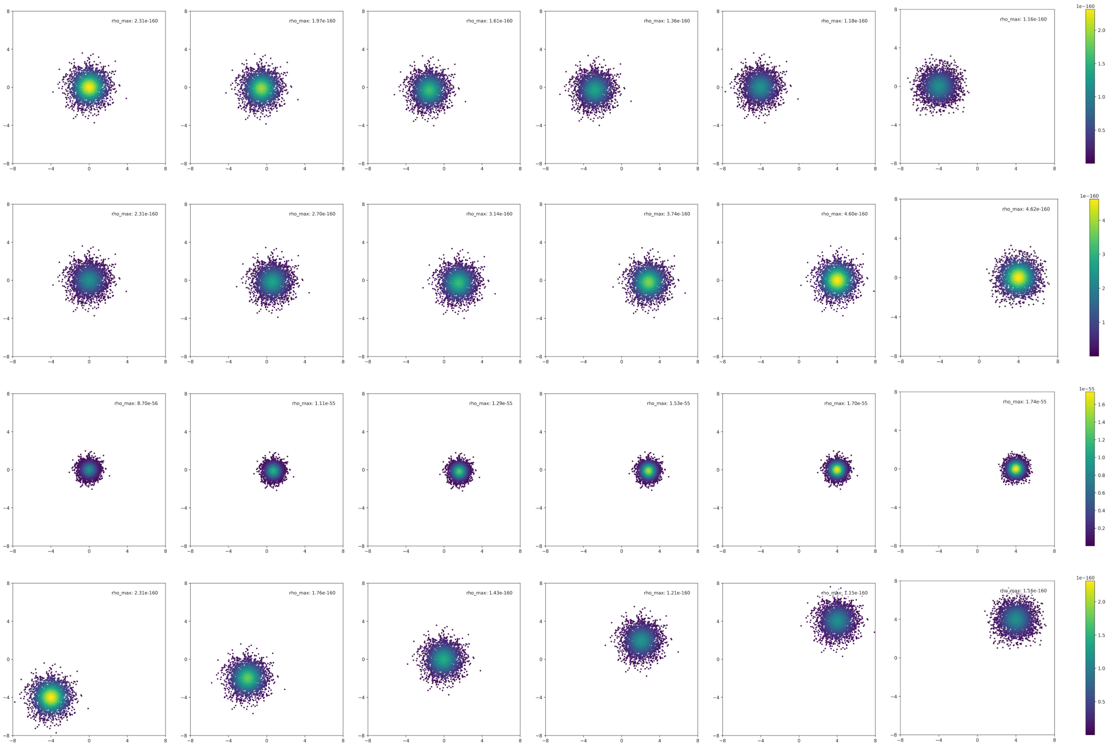
5.4 Generalization on crowd motion

The proposed method of UOT can be easily extended to address crowd motion problems with different total mass. For example, in the application of traffic flow, some vehicles enter and exit the flow while traveling from one location to another, resulting in different initial and terminal vehicle counts. Similarly, in the population migration of animals, there are births of new life and deaths of weaker individuals, resulting in differences between the initial and final population counts.

In contrast to the optimal transport problems, the crowd motion problem involves not only approximating the transition from the initial density ρ_0 to the target density ρ_1 but also ensuring that obstacles are avoided during dynamic movement. These obstacles can be incorporated into the objective function by introducing an additional term, referred to as the preference term

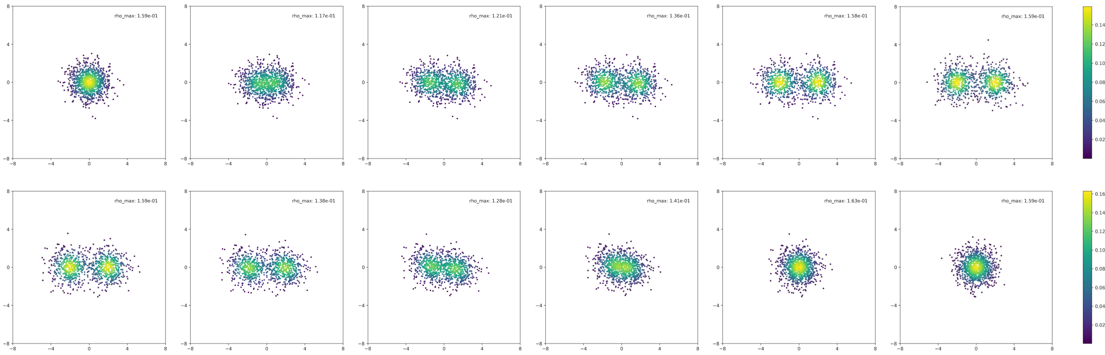
$$\mathcal{Q}(\rho)=\int_0^1\int_\Omega Q(\boldsymbol{x})\rho(\boldsymbol{x},t)\mathrm{d}\boldsymbol{x}\mathrm{d}t, \tag{5.1}$$

where $Q:\Omega\to\mathbb{R}$ models the spatial preferences of agents. A higher value of $Q(\boldsymbol{x})$ indicates a position closer to obstacles.



(a) $\rho_0(\boldsymbol{x}_0)$ (b) $\tilde{\rho}_{0.25}(\tilde{\boldsymbol{x}}_{0.25})$ (c) $\tilde{\rho}_{0.5}(\tilde{\boldsymbol{x}}_{0.5})$ (d) $\tilde{\rho}_{0.75}(\tilde{\boldsymbol{x}}_{0.75})$ (e) $\tilde{\rho}_1(\tilde{\boldsymbol{x}}_1)$ (f) $\rho_1(\boldsymbol{x}_1)$

Figure 9: Illustration of these Gaussian problems at $d=400$.

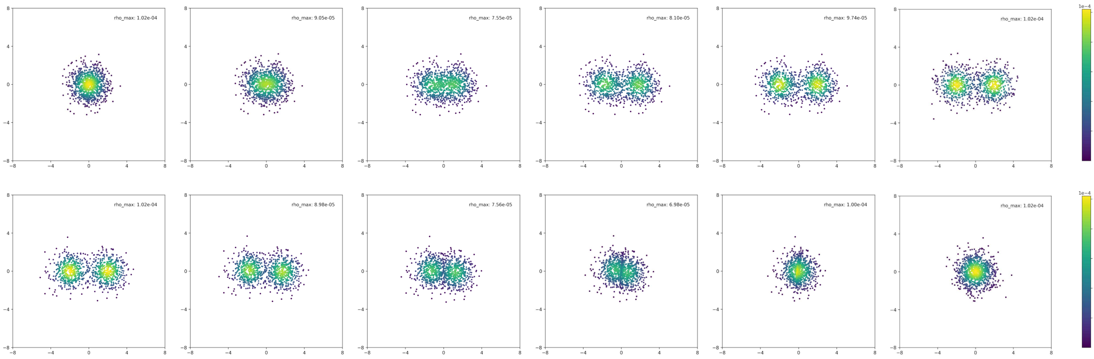


(a) $\rho_0(\boldsymbol{x}_0)$ (b) $\tilde{\rho}_{0.25}(\tilde{\boldsymbol{x}}_{0.25})$ (c) $\tilde{\rho}_{0.5}(\tilde{\boldsymbol{x}}_{0.5})$ (d) $\tilde{\rho}_{0.75}(\tilde{\boldsymbol{x}}_{0.75})$ (e) $\tilde{\rho}_1(\tilde{\boldsymbol{x}}_1)$ (f) $\rho_1(\boldsymbol{x}_1)$

Figure 10: Illustration of these Gaussian mixtures problems at $d=2$.

The optimization problem can then be formulated as

$$\min_{(\rho,v,f)\in\mathcal{C}(\rho_0,\rho_1)} \mathcal{E}(\rho,v)+\frac{1}{\alpha}\mathcal{R}(\rho,f)+\lambda_P\mathcal{Q}(\rho),$$



(a) $\rho_0(\boldsymbol{x}_0)$ (b) $\tilde{\rho}_{0.25}(\tilde{\boldsymbol{x}}_{0.25})$ (c) $\tilde{\rho}_{0.5}(\tilde{\boldsymbol{x}}_{0.5})$ (d) $\tilde{\rho}_{0.75}(\tilde{\boldsymbol{x}}_{0.75})$ (e) $\tilde{\rho}_1(\tilde{\boldsymbol{x}}_1)$ (f) $\rho_1(\boldsymbol{x}_1)$

Figure 11: Illustration of these Gaussian mixtures problems at $d=10$.

where λ_P is a penalty parameter, and $\mathcal{E}(\rho,v),\mathcal{R}(\rho,f)$ and $\mathcal{C}(\rho_0,\rho_1)$ are defined in (2.1) and (2.2). The proposed method can naturally be applied to solve the crowd motion problem. The semi-discrete version of $\mathcal{Q}(\rho)$ in the spatial direction is expressed as

$$\overline{\mathcal{Q}}(\rho):=\int_0^1\sum_{i=1}^r\frac{1}{r}Q\big(\mathbf{z}(\boldsymbol{x}_i,t)\big)\frac{\rho\big(\mathbf{z}(\boldsymbol{x}_i,t),t\big)}{\mu\big(\mathbf{z}(\boldsymbol{x}_i,t),t\big)}\mathrm{d}t.$$

Our crowd motion experiment involves two maze examples, where the initial density ρ_0 and target density ρ_1 have different total masses.

- Test 11: $\rho_0(\boldsymbol{x})=\rho_G(\boldsymbol{x},8\cdot\mathbf{e_1}+4\cdot\mathbf{e_2},\mathbf{I})$, $\rho_1(\boldsymbol{x})=\dfrac{1}{2}\rho_G(\boldsymbol{x},8\cdot\mathbf{e_1}+12\cdot\mathbf{e_2},0.3\cdot\mathbf{I})$.
- Test 12: $\rho_0(\boldsymbol{x})=\rho_G(\boldsymbol{x},-4\cdot\mathbf{e_1}-4\cdot\mathbf{e_2},\mathbf{I})$, $\rho_1(\boldsymbol{x})=\dfrac{1}{2}\rho_G(\boldsymbol{x},4\cdot\mathbf{e_1}+4\cdot\mathbf{e_2},0.3\cdot\mathbf{I})$.

The obstacles in the mazes are represented by an indicator function, with the regions defined as regular rectangles. To ensure that obstacles are successfully avoided, we initially apply a method similar to a Gaussian function to blur the rectangular obstacles, as illustrated in the last column (g) of Fig. 12. The density evolution at times $t=0,0.25,0.5,0.75$ and 1 is shown in (a)-(e). Furthermore, the red lines in (g) represent the characteristics (i.e. the learned trajectories) starting from randomly sampled points based on the initial density ρ_0. Besides, the push-forward of ρ_0 in (e) closely resembles the target density ρ_1 in (f), both in terms of position and colorbar values. These results show that the mass can effectively avoid the obstacles, which demonstrates the effectiveness of our UOT algorithm in handing the crowd motion problem with varying total masses.

6 Conclusion

In this paper, we propose a novel neural network-based method for solving high-dimensional dynamic UOT problems. To tackle the curse of dimensionality inherent in the

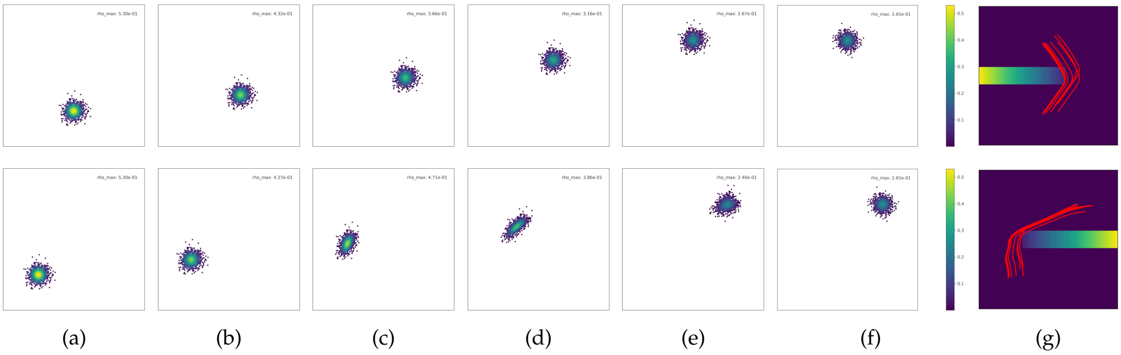


Figure 12: Illustration of crowd motion problems at $d=2$.

traditional mesh-based discretization methods, we employ Lagrangian discretization to solve the continuity equation with the source term along the characteristics, while utilizing the Monte Carlo method to approximate the high-dimensional integrals. Additionally, we carefully designed neural networks to parametrize both the velocity field and the source function. Our method is easy to implement through back-propagation and yields highly accurate results with a constant number of samples in the numerical experiments. In the future, we aim to explore the applications of the proposed framework in inverse problems and machine learning.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant Nos. 12301538 and 92370125) and by the National Key R&D Program of China (Grant No. 2021YFA1001300).

References

[1] M. Arjovsky, S. Chintala, and L. Bottou, *Wasserstein GAN*, arXiv:1701.07875, 2017.

[2] J.-D. Benamou and Y. Brenier, *A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem*, Numer. Math., 84:375–393, 2000.

[3] X. Chen, H. Benveniste, and A. R. Tannenbaum, *Unbalanced regularized optimal mass transport with applications to fluid flows in the brain*, Sci. Rep., 14:1111–1126, 2024.

[4] L. Chizat, G. Peyre, B. Schmitzer, and F.-X. Vialard, *An interpolating distance between optimal transport and Fisher-Rao metrics*, Found. Comput. Math., 18:1–44, 2018.

[5] L. Chizat, G. Peyre, B. Schmitzer, and F.-X. Vialard, *Unbalanced optimal transport: Dynamic and Kantorovich formulations*, J. Funct. Anal., 274:3090–3123, 2018.

[6] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, *Optimal transport for domain adaptation*, IEEE Trans. Pattern Anal. Mach. Intell., 39:1853–1865, 2017.

[7] M. Cuturi, *Sinkhorn distances: Lightspeed computation of optimal transport*, in: Advances in Neural Information Processing Systems, Curran Associates, Inc., 26:2294–2302, 2013.

[8] W. Gangbo, W. Li, S. Osher, and M. Puthawala, *Unnormalized optimal transport*, J. Comput. Phys., 399:108940, 2019.

[9] Y. Jing, J. Chen, L. Li, and J. Lu, *A deep learning framework for geodesics under spherical Wasserstein-Fisher-Rao metric and its application for weighted sample generation*, arXiv:2208.12145, 2022.

[10] L. Kantorovich, *On the translocation of masses*, Manag. Sci., 5:1–4, 1958.

[11] S. Kondratyev, L. Monsaingeon, and D. Vorotnikov, *A new optimal transport distance on the space of finite Radon measures*, Adv. Differential Equations, 21:1117–1164, 2016.

[12] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, *From word embeddings to document distances*, in: Proceedings of the 32nd International Conference on Machine Learning, PMLR, 957–966, 2015.

[13] J.-M. Lasry and P.-L. Lions, *Mean field games*, Jpn. J. Math., 2:229–260, 2007.

[14] W. Lee, R. Lai, W. Li, and S. Osher, *Generalized unnormalized optimal transport and its fast algorithms*, J. Comput. Phys., 436:110041, 2021.

[15] N. Lei, K. Su, L. Cui, S.-T. Yau, and X. D. Gu, *A geometric view of optimal transportation and generative model*, Comput. Aided Geom. Design, 68:1–21, 2019.

[16] D. Lombardi and E. Maitre, *Eulerian models and algorithms for unbalanced optimal transport*, ESAIM Math. Model. Numer. Anal., 49:1717–1744, 2015.

[17] J. Maas, M. Rumpf, C. Schonlieb, and S. Simon, *A generalized model for optimal transport of images including dissipation and density modulation*, ESAIM Math. Model. Numer. Anal., 49:1745–1769, 2015.

[18] B. K. Miller, M. Federici, C. Weniger, and P. Forrén, *Simulation-based inference with the generalized Kullback-Leibler divergence*, in: ICML 2023 Workshop on Synergy of Scientific and Machine Learning Modeling, 2023.

[19] G. Monge, *Mémoire sur la théorie des déblais et des remblais*, Histoire de l'Académie Royale des Sciences, 1781.

[20] N. Papadakis, *Optimal Transport for Image Processing*, PhD Thesis, Université de Bordeaux, 2015.

[21] B. Piccoli and F. Rossi, *Generalized Wasserstein distance and its application to transport equations with source*, Arch. Ration. Mech. Anal., 211:335–358, 2014.

[22] B. Piccoli and F. Rossi, *On properties of the generalized Wasserstein distance*, Arch. Ration. Mech. Anal., 222:1339–1365, 2016.

[23] I. Redko, N. Courty, R. Flamary, and D. Tuia, *Optimal transport for multi-source domain adaptation under target shift*, arXiv:1803.04899, 2019.

[24] T. Salimans, H. Zhang, A. Radford, and D. Metaxas, *Improving gans using optimal transport*, arXiv:1803.05573, 2018.

[25] F. Santambrogio, *Optimal Transport for Applied Mathematicians: Calculus of Variations, PDEs, and Modeling*, Birkäuser, 2015.

[26] V. Seguy, B. B. Damodaran, R. Flamary, N. Courty, A. Rolet, and M. Blondel, *Large scale optimal transport and mapping estimation*, in: 6th International Conference on Learning Representations, Curran Associates, Inc., 1006–1020, 2018.

[27] T. Sejourne, G. Peyre, and F.-X. Vialard, *Unbalanced optimal transport, from theory to numerics*, Handb. Numer. Anal., 24:407–471, 2023.

[28] Y. Sha, Y. Qiu, P. Zhou, and Q. Nie, *Reconstructing growth and dynamic trajectories from single-cell transcriptomics data*, Nat. Mach. Intell., 6:25–39, 2024.

[29] R. Sinkhorn, *A relationship between arbitrary positive matrices and doubly stochastic matrices*, Annals of Mathematical Statistics, 35:876–879, 1964.

[30] A. Tong, J. Huang, G. Wolf, D. V. Dijk, and S. Krishnaswamy, *Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics*, in: Proceedings of the 37th International Conference on Machine Learning, Curran Associates, Inc., 9468–9478, 2020.

[31] A. Tong, M. Kuchroo, S. Gupta, A. Venkat, B. P. San Juan, L. Rangel, B. Zhu, J. G. Lock, C. L. Chaffer, and S. Krishnaswamy, *Learning transcriptional and regulatory dynamics driving cancer cell plasticity using neural ODE-based optimal transport*, bioRxiv, 2023, doi: 10.1101/2023.03.28.534644.

[32] L. C. Torres, L. M. Pereira, and M. H. Amini, *A Survey on optimal transport for machine learning: Theory and applications*, arXiv:2106.01963, 2021.

[33] C. Villani, *Topics in Optimal Transportation*, in: Graduate Studies in Mathematics, Vol. 58, AMS, 2003.

[34] W. Wan, Y. Zhang, C. Bao, B. Dong, and Z. Shi, *A scalable deep learning approach for solving high-dimensional dynamic optimal transport*, SIAM J. Sci. Comput., 45:B544–B563, 2023.