

# Variational Regularized Unbalanced Optimal Transport: Single Network, Least Action

Yuhao Sun<sup>1,†</sup>, Zhenyi Zhang<sup>2,†</sup>, Zihan Wang<sup>3,†</sup>, Tiejun Li<sup>1,2\*</sup>, and Peijie Zhou<sup>1,3,4,5\*</sup>

<sup>1</sup>Center for Machine Learning Research, Peking University, Beijing, 100871, China

<sup>2</sup>LMAM and School of Mathematical Sciences, Peking University, Beijing, 100871, China

<sup>3</sup>Center for Quantitative Biology, Peking University, Beijing, 100871, China

<sup>4</sup>National Engineering Laboratory for Big Data Analysis and Applications, Beijing, 100871, China

<sup>5</sup>AI for Science Institute, Beijing, 100080, China

<sup>†</sup>These authors contributed equally to this work.

\*Correspondence: [tieli@pku.edu.cn](mailto:tieli@pku.edu.cn) (T. Li), [pjzhou@pku.edu.cn](mailto:pjzhou@pku.edu.cn) (P. Zhou)

## ABSTRACT

Recovering the dynamics from a few snapshots of a high-dimensional system is a challenging task in statistical physics and machine learning, with important applications in computational biology. Many algorithms have been developed to tackle this problem, based on frameworks such as optimal transport and the Schrödinger bridge. A notable recent framework is Regularized Unbalanced Optimal Transport (RUOT), which integrates both stochastic dynamics and unnormalized distributions. However, since many existing methods do not explicitly enforce optimality conditions, their solutions often struggle to satisfy the principle of least action and meet challenges to converge in a stable and reliable way. To address these issues, we propose Variational RUOT (Var-RUOT), a new framework to solve the RUOT problem. By incorporating the optimal necessary conditions for the RUOT problem into both the parameterization of the search space and the loss function design, Var-RUOT only needs to learn a scalar field to solve the RUOT problem and can search for solutions with lower action. We also examined the challenge of selecting a growth penalty function in the widely used Wasserstein–Fisher–Rao metric and proposed a solution that better aligns with biological priors in Var-RUOT. We validated the effectiveness of Var-RUOT on both simulated data and real single-cell datasets. Compared with existing algorithms, Var-RUOT can find solutions with lower action while exhibiting faster convergence and improved training stability.

## 1 Introduction

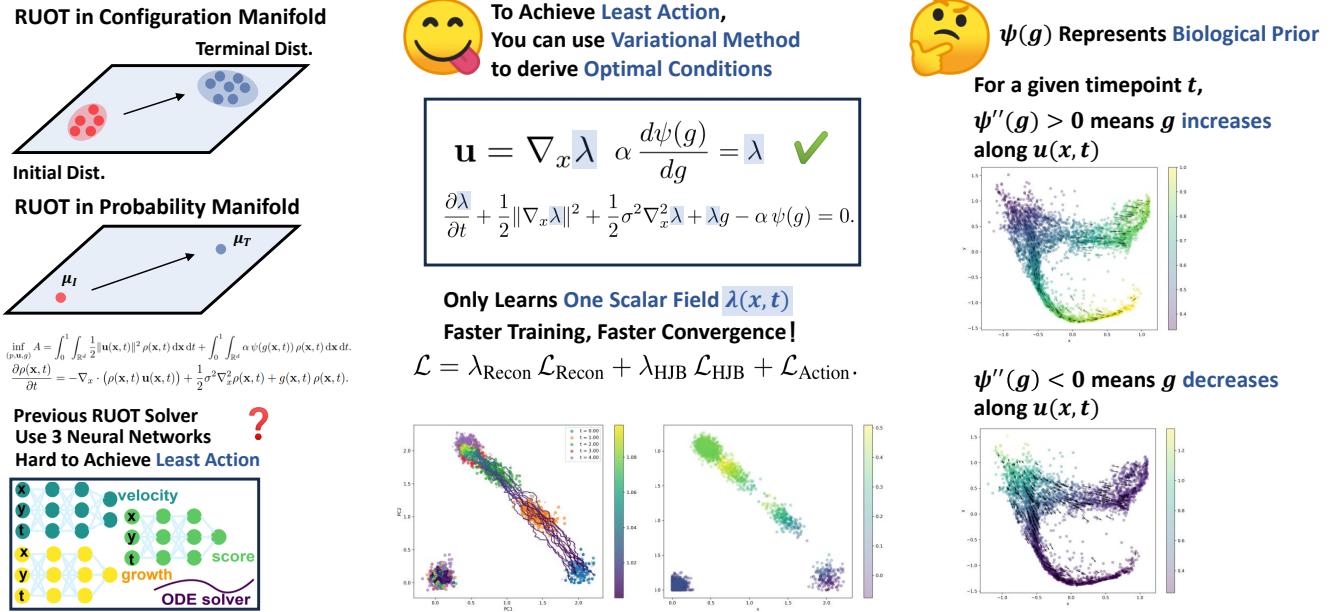
Inferring continuous dynamics from finite observations is crucial when analyzing systems with many particles (Chen et al., 2018). However, in many important applications such as single-cell RNA sequencing (scRNA-seq) experiments, only a few snapshot measurements are available, which makes recovering the underlying continuous dynamics a challenging task (Ding et al., 2022). Such a task of reconstructing dynamics from sparse snapshots is commonly referred to as *trajectory inference* in time-series scRNA-seq modeling (Zhang et al., 2025b; Ding et al., 2022; Heitz et al., 2024; Yeo et al., 2021b; Schiebinger et al., 2019a; Bunne et al., 2023b; Zhang et al., 2021) or the mathematical problem of *ensemble regression* (Yang et al., 2022).

A number of frameworks have been proposed to address this problem. For example, in dynamical optimal transport (OT), particles evolve according to the ordinary differential equations (ODEs) with the objective of minimizing the total action required to transport the initial distribution to the terminal distribution (Benamou & Brenier, 2000). Unbalanced dynamical OT further extends this framework by adding a penalty term  $\psi(g)$  on the particle growth or death processes in total transport energy (namely the Wasserstein–Fisher–Rao metric or WFR metric) to handle unnormalized distributions (Chizat et al., 2018a,b). Moreover, stochastic methods such as the Schrödinger Bridge adopt similar action principles while governing particle evolution via stochastic differential equations (SDEs) (Gentil et al., 2017; Léonard, 2014). Recently, the Regularized Unbalanced Optimal Transport (RUOT) framework generalizes these ideas by incorporating both stochasticity and particle birth-death processes (Lavenant et al., 2024; Ventre et al., 2023; Chizat et al., 2022; Pariset et al., 2023; Zhang et al., 2025a). In machine learning, generative models such as diffusion models (Ho et al., 2020; Song et al., 2021; Sohl-Dickstein et al., 2015; Song et al., 2020) and flow matching techniques (Lipman et al., 2023; Tong et al., 2024a; Liu et al., 2022) have also been adapted to solve transport problems. However, these approaches face two major challenges: 1) They usually do not explicitly enforce optimality conditions, leading to solutions that violate the principle of least action, and they meet challenges to converge reliably; 2) Selecting an appropriate penalty function  $\psi(g)$  that aligns with underlying biological priors remains challenging.

To overcome these challenges, we propose **Variational-RUOT (Var-RUOT)**. Our algorithm employs variational methods to derive the necessary conditions for action minimization within the RUOT framework. By parameterizing a single scalar field

with a neural network and incorporating these optimality conditions directly into our loss design, Var-RUOT learns dynamics with lower action. Experiments on both simulated and real datasets demonstrate that our approach achieves competitive performance with fewer training epochs and improved stability. Furthermore, we show that different choices of the penalty function for the growth rate  $g$  yield distinct biologically relevant priors in single-cell dynamics modeling. Our contributions are summarized as follows:

- We introduce a new method for solving RUOT problems by incorporating the first-order optimality conditions directly into the solution parameterization. This reduces the learning task to a single scalar potential function, which significantly simplifies the model space.
- We show how incorporating these necessary conditions into the loss function and architecture enables Var-RUOT to consistently discover transport paths with lower action, providing a more efficient and stable training process for RUOT problem.
- We address a key limitation in the classical Wasserstein-Fisher-Rao metric, which can yield biologically implausible solutions due to its quadratic growth penalty term. We propose the criterion and practical solution to modify such a penalty term, therefore enhancing the more realistic modeling of single-cell dynamics.



**Figure 1.** Overview of Variational RUOT

## 2 Related Works

**Deep Learning Solver for Trajectory Inference Problem** There are a large number of deep learning-based trajectory inference problem solvers. For example, there are solvers for Optimal Transport using static OT solver, Neural ODE or Flow matching techniques (Tong et al., 2020; Huguet et al., 2022; Wan et al., 2023; Zhang et al., 2024a; Tong et al., 2024a; Albergo et al., 2023; Palma et al., 2025; Rohbeck et al., 2025; Petrović et al., 2025; Schiebinger et al., 2019b; Klein et al., 2025), as well as solvers for the Schrödinger bridge that utilize either static or dynamic formulations (Shi et al., 2024; De Bortoli et al., 2021; Gu et al., 2025; Koshizuka & Sato, 2023; Neklyudov et al., 2023, 2024; Zhang et al., 2024b; Bunne et al., 2023a; Chen et al., 2022a; Zhou et al., 2024; Zhu et al., 2024; Maddu et al., 2024; Yeo et al., 2021a; Jiang & Wan, 2024; Lavenant et al., 2024; Ventre et al., 2023; Chizat et al., 2022; Tong et al., 2024b; Atanackovic et al., 2025; Yang, 2025; You et al., 2024). However, these methods typically **employ separate neural networks** to parameterize the velocity and growth functions, without leveraging their **optimality conditions** or the **inherent relationship** between them. This poses challenges in achieving optimal solutions that minimize the action energy.

**HJB equations in optimal transport** Methods that leverage the optimal conditions (e.g., Hamilton-Jacobi-Bellman (HJB) equations) of dynamic OT and its variants, have been proposed (Neklyudov et al., 2024; Zhang et al., 2024b; Chen et al., 2016; Benamou & Brenier, 2000; Neklyudov et al., 2023; Wu et al., 2025; Chow et al., 2020). However, these approaches typically do not simultaneously address both **unbalanced and stochastic** dynamics.

**WFR metric in time-series scRNA-seq modeling** In computational biology, several existing works model both cell state transitions and growth dynamics simultaneously in temporal scRNA-seq datasets by minimizing the action in the WFR metric i.e. solving the dynamical unbalanced optimal transport problem (Sha et al., 2024; Tong et al., 2023; Peng et al., 2024; Eyring et al., 2024) or its variants (Pariset et al., 2023; Lavenant et al., 2024; Zhang et al., 2025a). However, these works usually adopt the default growth penalty function  $\psi(g) = \frac{1}{2}g^2$  in the WFR metric and have not investigated the **biological implications of different choices** for  $\psi(g)$ .

### 3 Preliminaries and Backgrounds

**Dynamical Optimal Transport** The Dynamical Optimal Transport, also known as the Benamou–Brenier formulation, requires minimizing the following action functional (Benamou & Brenier, 2000):

$$\inf_{\rho, \mathbf{u}} \int_0^1 \int_{\mathbb{R}^d} \frac{1}{2} \|\mathbf{u}(\mathbf{x}, t)\|^2 \rho(\mathbf{x}, t) d\mathbf{x} dt,$$

where  $\rho$  and  $\mathbf{u}$  are subject to the continuity equation constraint:

$$\frac{\partial \rho(\mathbf{x}, t)}{\partial t} + \nabla_{\mathbf{x}} \cdot (\mathbf{u}(\mathbf{x}, t) \rho(\mathbf{x}, t)) = 0, \quad \rho(\cdot, 0) = \mu_0, \quad \rho(\cdot, 1) = \mu_1.$$

**Unbalanced Dynamical OT and Wasserstein–Fisher–Rao (WFR) metric** In order to handle unnormalized probability densities in practical problems (for example, to account for cell proliferation and death in computational biology), one can modify the form of the continuity equation by adding a birth-death term, and accordingly include a corresponding penalty term in the action. This leads to the optimal transport problem under the Wasserstein–Fisher–Rao (WFR) metric (Chizat et al., 2018a,b).

$$\inf_{\rho, \mathbf{u}, g} \int_0^1 \int_{\mathbb{R}^d} \left( \frac{1}{2} \|\mathbf{u}(\mathbf{x}, t)\|^2 + \alpha g^2(\mathbf{x}, t) \right) \rho(\mathbf{x}, t) d\mathbf{x} dt,$$

with  $\rho$ ,  $\mathbf{u}$ , and  $g$  subject to the unnormalized continuity equation constraint

$$\frac{\partial \rho(\mathbf{x}, t)}{\partial t} + \nabla_{\mathbf{x}} \cdot (\mathbf{u}(\mathbf{x}, t) \rho(\mathbf{x}, t)) - g(\mathbf{x}, t) \rho(\mathbf{x}, t) = 0, \quad \rho(\cdot, 0) = \mu_0, \quad \rho(\cdot, 1) = \mu_1.$$

**Schrödinger Bridge Problem and Dynamical Formulation** Schrödinger bridges aims to find the most likely way for a system to evolve from an initial distribution  $\mu_0$  to a terminal distribution  $\mu_1$ . Formally, let  $\mu_{[0,1]}^{\mathbf{X}}$  denote the probability measure induced by the stochastic process  $\mathbf{X}(t)$ ,  $0 \leq t \leq 1$ , and let  $\mu_{[0,1]}^{\mathbf{Y}}$  denote the probability measure induced by a given reference process  $\mathbf{Y}(t)$ ,  $0 \leq t \leq 1$ . The Schrödinger bridge seeks to solve  $\min_{\mu_{[0,1]}^{\mathbf{X}}} \mathcal{D}_{\text{KL}}(\mu_{[0,1]}^{\mathbf{X}} \parallel \mu_{[0,1]}^{\mathbf{Y}})$ . In particular, if  $\mathbf{X}_t$  follows the SDE  $d\mathbf{X}_t = \mathbf{u}(\mathbf{X}_t, t) dt + \boldsymbol{\sigma}(\mathbf{X}_t, t) d\mathbf{W}_t$ , where  $\mathbf{W}_t \in \mathbb{R}^d$  is a standard Brownian motion,  $\boldsymbol{\sigma}(\mathbf{x}, t) \in \mathbb{R}^{d \times d}$  is a given diffusion matrix, and the reference process is defined as  $d\mathbf{Y}_t = \boldsymbol{\sigma}(\mathbf{Y}_t, t) d\mathbf{W}_t$ , then the Schrödinger bridge problem is equivalent to the following stochastic optimal control problem (Chen et al., 2016; Gentil et al., 2017):

$$\inf_{\rho, \mathbf{u}} \int_0^1 \int_{\mathbb{R}^d} \left( \frac{1}{2} \mathbf{u}^T(\mathbf{x}, t) \mathbf{a}^{-1}(\mathbf{x}, t) \mathbf{u}(\mathbf{x}, t) \right) \rho(\mathbf{x}, t) d\mathbf{x} dt,$$

where  $\rho$  and  $\mathbf{u}$  are subject to the Fokker–Planck equation constraint

$$\frac{\partial \rho(\mathbf{x}, t)}{\partial t} + \nabla_{\mathbf{x}} \cdot (\mathbf{u}(\mathbf{x}, t) \rho(\mathbf{x}, t)) - \frac{1}{2} \nabla_{\mathbf{x}}^2 : (\mathbf{a}(\mathbf{x}, t) \rho(\mathbf{x}, t)) = 0, \quad \rho(\cdot, 0) = \mu_0, \quad \rho(\cdot, 1) = \mu_1.$$

Here,  $\mathbf{a}(\mathbf{x}, t) = \boldsymbol{\sigma}(\mathbf{x}, t) \boldsymbol{\sigma}^T(\mathbf{x}, t)$  and  $\nabla_{\mathbf{x}}^2 : (\mathbf{a}(\mathbf{x}, t) \rho(\mathbf{x}, t)) = \sum_{ij} \partial_{ij} (a_{ij} \rho(\mathbf{x}, t))$ .

**Regularized Unbalanced Optimal Transport** If we consider both unnormalized probability densities and stochasticity simultaneously, we arrive at the Regularized Unbalanced Optimal Transport (RUOT) problem (Chen et al., 2022b; Baradat & Lavenant, 2021; Zhang et al., 2025a).

**Definition 3.1** (Regularized Unbalanced Optimal Transport (RUOT) Problem). *Consider minimizing the following action:*

$$\inf_{\rho, \mathbf{u}, g} \int_0^1 \int_{\mathbb{R}^d} \left( \frac{1}{2} \mathbf{u}^T(\mathbf{x}, t) \mathbf{a}^{-1}(\mathbf{x}, t) \mathbf{u}(\mathbf{x}, t) + \alpha \psi(g) \right) \rho(\mathbf{x}, t) d\mathbf{x} dt,$$

where  $\psi : \mathbb{R} \rightarrow [0, +\infty)$  is a growth penalty function, and the quantities  $\rho$ ,  $\mathbf{u}$ , and  $g$  are subject to the following constraint, which is an unnormalized continuity equation:

$$\frac{\partial \rho}{\partial t} + \nabla_{\mathbf{x}} \cdot (\mathbf{u}(\mathbf{x}, t) \rho) - \frac{1}{2} \nabla_{\mathbf{x}}^2 : (\mathbf{a}(\mathbf{x}, t) \rho) - g(\mathbf{x}, t) \rho = 0, \quad \rho(\cdot, 0) = \mu_0, \quad \rho(\cdot, 1) = \mu_1.$$

## 4 Optimal Necessary Conditions for RUOT

To simplify our problem we adopt the assumption of isotropic time-invariant diffusion, i.e.,  $\mathbf{a}(\mathbf{x}, t) = \sigma^2 \mathbf{I}$ . We refer to the RUOT problem in this scenario as the *isotropic time-invariant RUOT problem*.

**Definition 4.1** (Isotropic Time-Invariant (ITI) RUOT Problem). *Consider the following minimum-action problem with the action functional given by*

$$\inf_{(\rho, \mathbf{u}, g)} \mathcal{T} = \int_0^1 \int_{\mathbb{R}^d} \frac{1}{2} \|\mathbf{u}(\mathbf{x}, t)\|^2 \rho(\mathbf{x}, t) d\mathbf{x} dt + \int_0^1 \int_{\mathbb{R}^d} \alpha \psi(g(\mathbf{x}, t)) \rho(\mathbf{x}, t) d\mathbf{x} dt. \quad (1)$$

Here,  $\psi : \mathbb{R} \rightarrow [0, +\infty)$  is the growth penalty function, and the triplet  $(\rho, \mathbf{u}, g)$  is subject to the constraint of the Fokker–Planck equation

$$\frac{\partial \rho(\mathbf{x}, t)}{\partial t} = -\nabla_{\mathbf{x}} \cdot (\rho(\mathbf{x}, t) \mathbf{u}(\mathbf{x}, t)) + \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 \rho(\mathbf{x}, t) + g(\mathbf{x}, t) \rho(\mathbf{x}, t). \quad (2)$$

Additionally,  $\rho$  satisfies the initial and terminal conditions  $\rho(\cdot, 0) = \mu_0$ ,  $\rho(\cdot, 1) = \mu_1$ .

In particular, if  $\psi(g(\mathbf{x}, t)) = \frac{1}{2} g^2(\mathbf{x}, t)$ , then this problem is referred to as the *unbalanced dynamic optimal transport with WFR metric*. We can derive the necessary conditions for the action functional to achieve a minimum using variational methods.

**Theorem 4.1** (Necessary Conditions for Achieving the Optimal Solution in the ITI-RUOT Problem). *In the problem defined in Definition 4.1, the necessary conditions for the action  $\mathcal{T}$  to attain a minimum are*

$$\mathbf{u} = \nabla_{\mathbf{x}} \lambda, \quad \alpha \frac{d\psi(g)}{dg} = \lambda, \quad \frac{\partial \lambda}{\partial t} + \frac{1}{2} \|\nabla_{\mathbf{x}} \lambda\|^2 + \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 \lambda + \lambda g - \alpha \psi(g) = 0. \quad (3)$$

Here,  $\lambda(\mathbf{x}, t)$  is a scalar field. The proof of this theorem can be found in Appendix A.1.1.

**Remark 4.1.** Substituting the necessary conditions satisfied by  $\mathbf{u}$  and  $g$  into the Fokker–Planck equation, the evolution of the probability density  $\rho(\mathbf{x}, t)$  is determined by  $\frac{\partial \rho(\mathbf{x}, t)}{\partial t} = -\nabla_{\mathbf{x}} \cdot (\rho(\mathbf{x}, t) \nabla_{\mathbf{x}} \lambda(\mathbf{x}, t)) + \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 \rho(\mathbf{x}, t) + (\psi')^{-1}\left(\frac{\lambda(\mathbf{x}, t)}{\alpha}\right) \rho(\mathbf{x}, t)$ , where  $\psi' = \frac{d\psi(g)}{dg}$ , and  $(\psi')^{-1}$  denotes the inverse function of  $\psi'$ .

**Remark 4.2.** If we choose the growth penalty function to take the form used in the WFR metric, i.e.,  $\psi(g) = \frac{1}{2} g^2$ , and set  $\alpha = 1$ ,  $\sigma = 0$ , then the above optimal necessary conditions immediately degenerate to  $\mathbf{u} = \nabla_{\mathbf{x}} \lambda$ ,  $g = \lambda$ ,  $\frac{\partial \lambda}{\partial t} + \frac{1}{2} \|\nabla_{\mathbf{x}} \lambda\|^2 + \frac{1}{2} \lambda^2 = 0$ , which is same as the form derived in (Neklyudov et al., 2024) under the WFR metric. If we let  $g = 0$  and  $\psi(0) = 0$  it becomes  $\mathbf{u} = \nabla_{\mathbf{x}} \lambda$ ,  $\frac{\partial \lambda}{\partial t} + \frac{1}{2} \|\nabla_{\mathbf{x}} \lambda\|^2 + \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 \lambda = 0$ , which is same as the form derived in (Neklyudov et al., 2024; Zhang et al., 2024b; Chen et al., 2016) under the Schrödinger Bridge problem.

From Theorem 4.1 and Remark 4.1, the vector field  $\mathbf{u}(\mathbf{x}, t)$  and the growth rate  $g(\mathbf{x}, t)$  can be directly obtained from the scalar field  $\lambda(\mathbf{x}, t)$ . Moreover, since the initial density  $\rho(\cdot, 0)$  is known, once the necessary conditions are satisfied the evolution equation (i.e., the Fokker–Planck equation) is completely determined by  $\lambda(\mathbf{x}, t)$ . Thus, the scalar field  $\lambda(\mathbf{x}, t)$  fully determines the system’s evolution; we only need to solve for one  $\lambda(\mathbf{x}, t)$ , which simplifies the problem. However, these necessary conditions

introduce a coupling between  $\mathbf{u}(\mathbf{x}, t)$  and  $g(\mathbf{x}, t)$ , and this coupling could contradict biological prior knowledge. In biological data, it is generally believed that cells located at the upstream of a trajectory are stem cells with the highest proliferation and differentiation capabilities, and thus the corresponding  $g$  values should be maximal. Along the trajectory, as the cells gradually lose their "stemness," the  $g$  values should decrease. Under the necessary conditions, however, whether  $g(\mathbf{x}, t)$  increases or decreases along  $\mathbf{u}(\mathbf{x}, t)$  at a given time  $t$  depends on the form of the growth penalty function.

**Theorem 4.2** (The relationship between  $\mathbf{u}$  and  $g$ ; Biological prior). *At a fixed time  $t$ , if  $\frac{d\psi^2(g)}{dg^2} > 0$ , then  $g(\mathbf{x}, t)$  ascends in the direction of the velocity field  $\mathbf{u}(\mathbf{x}, t)$  (i.e.,  $\mathbf{u}(\mathbf{x}, t)^T (\nabla_{\mathbf{x}} g(\mathbf{x}, t)) > 0$ ); otherwise, it descends.*

The proof is given in [Appendix A.1.2](#). According to this theorem, to ensure the solution complies with biological prior, i.e., that at a given time the cells upstream in the trajectory exhibit the higher  $g$  value, it is necessary that  $\frac{d^2\psi(g)}{dg^2} < 0$ .

## 5 Solving ITI ROUT Problem Through Neural Network

Given samples from distributions  $\rho_t$  at  $K$  discrete time points,  $t \in \{T_1, \dots, T_K\}$ , we aim to recover the continuous evolution process of the distributions by solving the ITI RUOT problem, that is, by minimizing the action functional while ensuring that  $\rho(\mathbf{x}, t)$  matches the distributions  $\rho_t$  at the corresponding time points. Since the values of  $\mathbf{u}(\mathbf{x}, t)$  and  $g(\mathbf{x}, t)$ , as well as the evolution of  $\rho(\mathbf{x}, t)$  over time, are fully determined by the scalar field  $\lambda(\mathbf{x}, t)$  in variational form ([Appendix A.1.1](#)), we approximate this scalar field using a single neural network. Specifically, we parameterize  $\lambda(\mathbf{x}, t)$  as  $\lambda_{\theta}(\mathbf{x}, t)$ , where  $\theta$  represents the neural network parameters.

### 5.1 Simulating SDEs Using the Weighted Particle Method

Directly solving the high-dimensional RUOT with PDE constraints is challenging; therefore, we reformulate the problem by simulating the trajectories of a number of weighted particles.

**Theorem 5.1.** *Consider a weighted particle system consisting of  $N$  particles, where the position of particle  $i$  at time  $t$  is given by  $\mathbf{X}_i^t \in \mathbb{R}^d$  and its weight by  $w_i(t) > 0$ . The dynamics of each particle are described by*

$$\begin{aligned} d\mathbf{X}_i^t &= \mathbf{u}(\mathbf{X}_i^t, t) dt + \sigma(t) d\mathbf{W}_i, \\ dw_i &= g(\mathbf{X}_i^t, t) w_i dt, \end{aligned} \tag{4}$$

where  $\mathbf{u} : \mathbb{R}^d \times [0, T] \rightarrow \mathbb{R}^d$  is a time-varying vector field,  $g : \mathbb{R}^d \times [0, T] \rightarrow \mathbb{R}$  is a growth rate function,  $\sigma : [0, T] \rightarrow [0, +\infty)$  is a time-varying diffusion coefficient, and  $\mathbf{W}_i$  is an  $N$ -dimensional standard Brownian motion with independent components in each coordinate. The initial conditions are  $\mathbf{X}_i^0 \sim \rho(\mathbf{x}, 0)$  and  $w_i(0) = 1$ . In the limit as  $N \rightarrow \infty$ , the empirical measure  $\mu^N = \frac{1}{N} \sum_{i=1}^N w_i(t) \delta(\mathbf{x} - \mathbf{X}_i^t)$  converges to the solution of the following Fokker–Planck equation:

$$\frac{\partial \rho(\mathbf{x}, t)}{\partial t} = -\nabla_{\mathbf{x}} \cdot (\mathbf{u}(\mathbf{x}, t) \rho(\mathbf{x}, t)) + \frac{1}{2} \sigma^2(t) \nabla_{\mathbf{x}}^2 \rho(\mathbf{x}, t) + g(\mathbf{x}, t) \rho(\mathbf{x}, t), \tag{5}$$

with the initial condition  $\rho(\mathbf{x}, 0) = \rho_0(\mathbf{x})$ .

The proof is provided in [Appendix A.1.3](#). This theorem implies that we can approximate the evolution of  $\rho(\mathbf{x}, t)$  by simulating  $N$  particles, where each particle's weight  $w_i$  is governed by an ODE and its position  $\mathbf{X}_i$  is governed by an SDE. The evolution of the empirical measure  $\mu^N$  thereby approximates the evolution of  $\rho(\mathbf{x}, t)$ .

### 5.2 Reformulating the Loss in Weighted Particle Form

The total loss function consists of three components such that  $\mathcal{L} = \mathcal{L}_{\text{Recon}} + \gamma_{\text{HJB}} \mathcal{L}_{\text{HJB}} + \gamma_{\text{Action}} \mathcal{L}_{\text{Action}}$ . Here,  $\mathcal{L}_{\text{Recon}}$  ensures that the distribution generated by the model closely matches the true data distribution,  $\mathcal{L}_{\text{HJB}}$  enforces that the learned  $\lambda_{\theta}(\mathbf{x}, t)$  satisfies the HJB equation in the necessary conditions, and  $\mathcal{L}_{\text{Action}}$  minimizes the action as much as possible.

**Reconstruction Loss** Minimizing the reconstruction loss guarantees that the distribution generated by the model is consistent with the real data distribution. Since in the ITI RUOT problem the probability density  $\rho(\mathbf{x}, t)$  is not normalized, we need to match both the total mass and the discrepancy between the two distributions. Our reconstruction loss is given by  $\mathcal{L}_{\text{Recon}} = \gamma_{\text{Mass}} \mathcal{L}_{\text{Mass}} + \mathcal{L}_{\text{OT}}$ , where, at time point  $k$ , the true mass is  $\int_{\mathbb{R}^d} \rho(\mathbf{x}, T_k) d\mathbf{x} = M(T_k)$ , and the weight of particle  $i$  is  $w_i(T_k)$ . Thus, the total mass of the model-generated distribution is  $\hat{M}(T_k) = \frac{1}{N} \sum_{i=1}^N w_i(T_k)$ . The mass reconstruction loss is then defined as  $\mathcal{L}_{\text{Mass}} = \sum_{k=1}^K (M(T_k) - \hat{M}(T_k))^2$ . Let the true distribution at time point  $k$  be  $\rho(\mathbf{x}, T_k)$ . Its normalized version is given by  $\tilde{\rho}(\mathbf{x}, T_k) = \frac{\rho(\mathbf{x}, T_k)}{\int_{\mathbb{R}^d} \rho(\mathbf{x}, T_k) d\mathbf{x}}$ , while the normalized model-generated distribution is  $\hat{\rho}(\mathbf{x}, T_k) = \frac{\frac{1}{N} \sum_{i=1}^N w_i(T_k) \delta(\mathbf{x} - \mathbf{X}_i)}{\hat{M}(T_k)}$ . The distribution reconstruction loss is then defined as  $\mathcal{L}_{\text{OT}} = \sum_{k=1}^K \mathcal{W}_2(\tilde{\rho}(\cdot, T_k), \hat{\rho}(\cdot, T_k))$ , where  $\gamma_{\text{Mass}}$  is a hyperparameter that controls the importance of the mass reconstruction loss.

**HJB Loss** Minimizing the HJB loss ensures that the learned  $\lambda_\theta(\mathbf{x}, t)$  obeys the HJB equation constraints specified in the necessary conditions. Since the gradient operator in the HJB equation is a local operator, we compute the HJB loss by integrating the extent to which  $\lambda_\theta(\mathbf{x}, t)$  violates the HJB equation along the trajectories. When using  $N$  particles, the HJB loss is given by:  $\mathcal{L}_{\text{HJB}}^N = \sum_{i=1}^N \left[ \int_0^{T_K} \frac{w_i(t)}{\sum_{i=1}^N w_i(t)} \left( \frac{\partial \lambda_\theta(\mathbf{X}_i^t, t)}{\partial t} + \frac{1}{2} \|\nabla_{\mathbf{x}} \lambda_\theta\|^2 + \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 \lambda_\theta + g_\theta(\mathbf{X}_i^t, t) - \alpha \psi(g_\theta) \right)^2 dt \right]$ . Here  $g_\theta$  is obtained from the necessary condition  $\alpha \frac{d\psi(g_\theta)}{dg_\theta} = \lambda_\theta$ .

**Remark 5.1.** The expectation of HJB Loss is  $\mathbb{E}[\mathcal{L}_{\text{HJB}}^N] = \int_0^{T_K} \int_{\mathbb{R}^d} \hat{\rho}(\mathbf{x}, t) \left( \frac{\partial \lambda(\mathbf{x}, t)}{\partial t} + \frac{1}{2} \|\nabla_{\mathbf{x}} \lambda\|^2 + \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 \lambda + g(\mathbf{x}, t) - \alpha \psi(g) \right)^2 d\mathbf{x} dt$ , where  $\hat{\rho}(\mathbf{x}, t) = \frac{\rho(\mathbf{x}, t)}{\int_{\mathbb{R}^d} \rho(\mathbf{x}, t) d\mathbf{x}}$  is the probability density by normalizing  $\rho(\mathbf{x}, t)$ . The proof is left in [Appendix A.1.4](#).

**Action Loss** Since the variational method provides only the necessary conditions for achieving minimal action rather than sufficient ones, we need to incorporate the action into the loss so that the action is minimized as much as possible. The action loss is also computed via simulating weighted particles. When using  $N$  particles, it is given by  $\mathcal{L}_{\text{Action}}^N = \frac{1}{N} \sum_{i=1}^N \left( \int_0^1 \frac{1}{2} \|\mathbf{u}_\theta(\mathbf{X}_i^t, t)\|^2 w_i(t) dt + \int_0^1 \alpha \psi(g_\theta(\mathbf{X}_i^t, t)) w_i(t) dt \right)$ . Here  $\mathbf{u}$  and  $g$  is obtained from the necessary condition  $\mathbf{u}_\theta = \nabla_{\mathbf{x}} \lambda_\theta(\mathbf{x}, t), \alpha \frac{d\psi(g_\theta)}{dg_\theta} = \lambda_\theta$ .

**Remark 5.2.** The expectation of action loss is exactly the action defined in the ITI RUOT problem ([Definition 4.1](#)):  $\mathbb{E}[\mathcal{L}_{\text{Action}}^N] = \mathcal{T}$ . The proof is left in [Appendix A.1.5](#).

Overall, the training process of Var-RUOT involves minimizing the total of three loss terms described above to fit  $\lambda_\theta$ . The training procedure is provided in [Algorithm 1](#).

### 5.3 Adjusting the Growth Penalty Function to Match Biological Priors

As discussed in [Theorem 4.2](#), the second-order derivative of  $\psi(g)$  encodes the biological prior: if  $\psi''(g) > 0$ , then at any given time  $t$ ,  $g$  increases in the direction of the velocity field, and vice versa. Therefore, we choose two representative forms of  $\psi(g)$  for our solution. Given that  $\psi(g)$  penalizes nonzero  $g$ , it should satisfy the following properties: (1) The further  $g$  deviates from 0, the larger  $\psi(g)$  becomes, i.e.,  $\frac{d\psi(g)}{dg} > 0$ . (2) Birth and death are penalized equally when prior knowledge is absent, i.e.,  $\psi(g) = \psi(-g)$ .

**Case 1:**  $\psi''(g) > 0$  In the case where  $\psi''(g) > 0$ , a typical form that meets the requirements is  $\psi(g) = Cg^{2p}$ ,  $p \in \mathbb{Z}^+$ ,  $C > 0$ . We select the form used in the WFR Metric, namely,  $\psi_1(g) = \frac{1}{2}g^2$ . The optimal conditions are presented in [Appendix A.2](#).

**Case 2:**  $\psi''(g) < 0$  For the case where  $\psi''(g) < 0$ , a typical form that meets the conditions is  $\psi(g) = Cg^{(2p)/(2q+1)}$ , where  $p, q \in \mathbb{Z}^+$  and  $2p < 2q + 1$ . In order to obtain a smoother  $g(\lambda)$  relationship from the necessary conditions, and as a illustrative example, we choose  $\psi_2(g) = g^{2/15}$ . The optimal conditions are also presented in [Appendix A.2](#).

## 6 Numerical Results

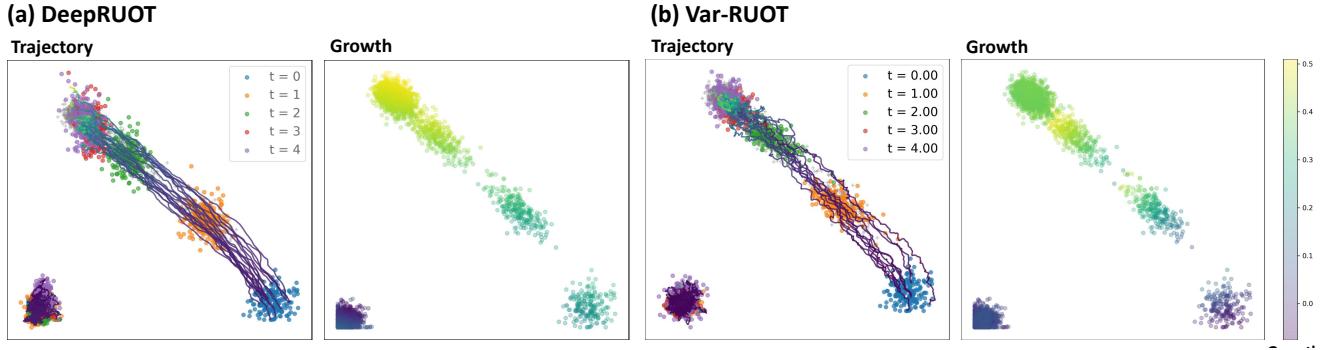
In the experiments presented below, unless the use of the modified metric is explicitly stated, we utilize the standard WFR metric, namely,  $\psi_1(g) = \frac{1}{2}g^2$ .

### 6.1 Var-RUOT Minimizes Path Action

To evaluate the ability of Var-RUOT to capture the minimum-action trajectory, we first conducted experiments on a three-gene simulation dataset ([Zhang et al., 2025a](#)). The dynamics of the three-gene simulation data are governed by stochastic differential equations that incorporate self-activation, mutual inhibition, and external activation. The detailed specifications of the dataset are provided in [Appendix B.1](#). The trajectories learned by DeepRUOT and Var-RUOT are illustrated in [Fig. 2](#), and the  $\mathcal{W}_1$  and  $\mathcal{W}_2$  losses between the generated distributions and the ground truth distribution, as well as the corresponding action values, are reported in [Table 1](#). In the table, we report the action of the method that utilizes the WFR Metric. The experimental results demonstrate that Var-RUOT accurately recovers the desired trajectories, achieving a lower action while maintaining distribution matching accuracy. To further assess the performance of Var-RUOT on high-dimensional data, we also conducted experiments on an Epithelial Mesenchymal Transition (EMT) dataset([Sha et al., 2024](#); [Cook & Vanderhyden, 2020](#)). This dataset was reduced to a 10-dimensional feature space, and the trajectories obtained after applying PCA for dimensionality reduction are shown in [Fig. 3](#). Both Var-RUOT and DeepRUOT learn dynamics that can transform the distribution at  $t = 0$  into the distributions at  $t = 1, 2, 3$ . Var-RUOT learns the nearly straight-line trajectory corresponding to the minimum action, whereas DeepRUOT learns a curved trajectory. The results of  $\mathcal{W}_1, \mathcal{W}_2$  distance and action, summarized in [Table 2](#), Var-RUOT also learns trajectories with smaller action while achieving matching accuracy comparable to that of other algorithms.

**Table 1.** On the three-gene simulated dataset, the Wasserstein distances ( $\mathcal{W}_1$  and  $\mathcal{W}_2$ ) between the predicted distributions of each algorithm and the true distribution at various time points. Each experiment was run five times to compute the mean and standard deviation.

| Model                                    | $t = 1$                    |                            | $t = 2$                    |                            | $t = 3$                    |                            | $t = 4$                    |                            | Path Action                |
|--|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|
|  | $\mathcal{W}_1$            | $\mathcal{W}_2$            | $\mathcal{W}_1$            | $\mathcal{W}_2$            | $\mathcal{W}_1$            | $\mathcal{W}_2$            | $\mathcal{W}_1$            | $\mathcal{W}_2$            |                            |
| SF2M (Tong et al., 2024b)                | 0.1914 $\pm$ 0.0051        | 0.3253 $\pm$ 0.0059        | 0.4706 $\pm$ 0.0200        | 0.7648 $\pm$ 0.0059        | 0.7648 $\pm$ 0.0260        | 1.0750 $\pm$ 0.0267        | 2.1879 $\pm$ 0.0451        | 2.8830 $\pm$ 0.0741        | —                          |
| PISDE (Jiang & Wan, 2024)                | 0.1313 $\pm$ 0.0023        | 0.3232 $\pm$ 0.0013        | 0.2311 $\pm$ 0.0015        | 0.5356 $\pm$ 0.0015        | 0.4103 $\pm$ 0.0006        | 0.7913 $\pm$ 0.0035        | 0.5418 $\pm$ 0.0015        | 0.9579 $\pm$ 0.0037        | —                          |
| MIO Flow (Huguet et al., 2022)           | 0.1290 $\pm$ 0.0000        | 0.2087 $\pm$ 0.0000        | 0.2963 $\pm$ 0.0000        | 0.4565 $\pm$ 0.0000        | 0.6461 $\pm$ 0.0000        | 1.0165 $\pm$ 0.0000        | 1.1473 $\pm$ 0.0000        | 1.7827 $\pm$ 0.0000        | —                          |
| Action Matching (Neklyudov et al., 2023) | 0.3801 $\pm$ 0.0000        | 0.5033 $\pm$ 0.0000        | 0.5028 $\pm$ 0.0000        | 0.5637 $\pm$ 0.0000        | 0.6288 $\pm$ 0.0000        | 0.6822 $\pm$ 0.0000        | 0.8480 $\pm$ 0.0000        | 0.9034 $\pm$ 0.0000        | 1.5491                     |
| TIGON (Sha et al., 2024)                 | 0.0519 $\pm$ 0.0000        | <b>0.0731</b> $\pm$ 0.0000 | 0.0763 $\pm$ 0.0000        | 0.1559 $\pm$ 0.0000        | 0.1387 $\pm$ 0.0000        | 0.2436 $\pm$ 0.0000        | 0.1908 $\pm$ 0.0000        | 0.2203 $\pm$ 0.0000        | 1.2442                     |
| DeepRUOT (Zhang et al., 2025a)           | 0.0569 $\pm$ 0.0019        | 0.1125 $\pm$ 0.0033        | 0.0811 $\pm$ 0.0037        | 0.1578 $\pm$ 0.0079        | 0.1246 $\pm$ 0.0040        | 0.2158 $\pm$ 0.0081        | 0.1538 $\pm$ 0.0056        | 0.2588 $\pm$ 0.0088        | 1.4058                     |
| Var-RUOT (Ours)                          | <b>0.0452</b> $\pm$ 0.0024 | 0.1181 $\pm$ 0.0064        | <b>0.0385</b> $\pm$ 0.0022 | <b>0.1270</b> $\pm$ 0.0121 | <b>0.0445</b> $\pm$ 0.0033 | <b>0.1144</b> $\pm$ 0.0160 | <b>0.0572</b> $\pm$ 0.0034 | <b>0.2140</b> $\pm$ 0.0067 | <b>1.1105</b> $\pm$ 0.0515 |



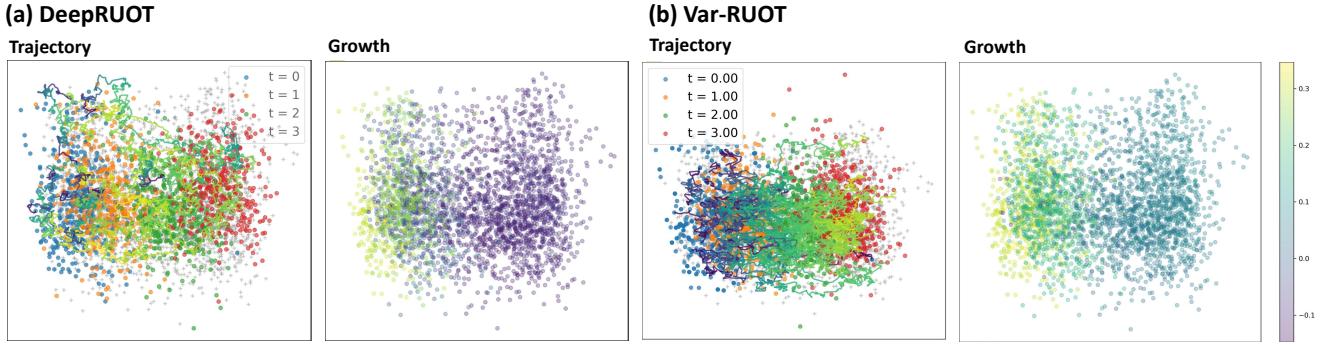
**Figure 2.** a) The trajectory and growth learned by DeepRUOT on the three-gene simulated dataset; b) The trajectory and growth learned by Var-RUOT on the same dataset.

**Table 2.** On the EMT dataset, the Wasserstein distances ( $\mathcal{W}_1$  and  $\mathcal{W}_2$ ) between the predicted distributions of each algorithm and the true distribution at various time points. Each experiment was run five times to compute the mean and standard deviation.

| Model                                    | $t = 1$                    |                            | $t = 2$                    |                            | $t = 3$                    |                            | $t = 4$                    |                 | Path Action |
|--|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|-----------------|-------------|
|  | $\mathcal{W}_1$            | $\mathcal{W}_2$            | $\mathcal{W}_1$            | $\mathcal{W}_2$            | $\mathcal{W}_1$            | $\mathcal{W}_2$            | $\mathcal{W}_1$            | $\mathcal{W}_2$ |             |
| S2FM (Tong et al., 2024b)                | 0.2566 $\pm$ 0.0016        | 0.2646 $\pm$ 0.0016        | 0.2811 $\pm$ 0.0016        | 0.2897 $\pm$ 0.0012        | 0.2900 $\pm$ 0.0010        | 0.3005 $\pm$ 0.0010        | —                          | —               | —           |
| PISDE (Jiang & Wan, 2024)                | 0.2694 $\pm$ 0.0016        | 0.2785 $\pm$ 0.0016        | 0.2860 $\pm$ 0.0013        | 0.2954 $\pm$ 0.0012        | 0.2790 $\pm$ 0.0015        | 0.2920 $\pm$ 0.0016        | —                          | —               | —           |
| MIO Flow (Huguet et al., 2022)           | <b>0.2439</b> $\pm$ 0.0000 | <b>0.2529</b> $\pm$ 0.0000 | <b>0.2665</b> $\pm$ 0.0000 | 0.2770 $\pm$ 0.0000        | 0.2841 $\pm$ 0.0000        | 0.2984 $\pm$ 0.0000        | —                          | —               | —           |
| Action Matching (Neklyudov et al., 2023) | 0.4723 $\pm$ 0.0000        | 0.4794 $\pm$ 0.0000        | 0.6382 $\pm$ 0.0000        | 0.6454 $\pm$ 0.0000        | 0.8453 $\pm$ 0.0000        | 0.8524 $\pm$ 0.0000        | 0.8583                     | 0.8583          | —           |
| TIGON (Sha et al., 2024)                 | 0.2433 $\pm$ 0.0000        | 0.2523 $\pm$ 0.0000        | 0.2661 $\pm$ 0.0000        | 0.2766 $\pm$ 0.0000        | 0.2847 $\pm$ 0.0000        | 0.2989 $\pm$ 0.0000        | 0.4672                     | 0.4672          | —           |
| DeepRUOT (Zhang et al., 2025a)           | 0.2902 $\pm$ 0.0009        | 0.2987 $\pm$ 0.0012        | 0.3193 $\pm$ 0.0006        | 0.3293 $\pm$ 0.0008        | 0.3291 $\pm$ 0.0018        | 0.3410 $\pm$ 0.0023        | 0.4857                     | 0.4857          | —           |
| Var-RUOT(Ours)                           | 0.2540 $\pm$ 0.0016        | 0.2623 $\pm$ 0.0017        | 0.2670 $\pm$ 0.0013        | <b>0.2756</b> $\pm$ 0.0014 | <b>0.2683</b> $\pm$ 0.0014 | <b>0.2796</b> $\pm$ 0.0015 | <b>0.3544</b> $\pm$ 0.0019 | —               | —           |

## 6.2 Var-RUOT Stabilizes and Accelerates Training Process

To demonstrate that Var-RUOT converges faster and exhibits improved training stability, we further tested it on both the simulated and the EMT dataset. We trained the neural networks for the various algorithms using the same learning rate and optimizer, running each dataset five times. For each training, we recorded the number of epochs and wall-clock time required for the OT loss related to the distribution matching accuracy to decrease below a specified threshold (set to 0.30 in this study). Each training session was capped at a maximum of 500 epochs; if an algorithm’s OT loss did not reach the threshold within 500 epochs, the required epoch was recorded as 500, and the wall-clock time was noted as the total duration of the training session. The experimental results are summarized in Table 3, which lists the mean and standard deviation of both the epochs and wall-clock times required for each algorithm on each dataset. The mean values reflect the convergence speed, while the standard deviations indicate the training stability. Our algorithm demonstrated both a faster convergence speed and better stability compared to the other methods. In Appendix C.1, we further demonstrate our training speed and stability by plotting the loss decay curves.



**Figure 3.** a) The trajectory and growth learned by DeepRUOT on the EMT dataset; b) The trajectory and growth learned by Var-RUOT on the same dataset.

**Table 3.** The number of epochs and wall time required for the OT Loss to drop below the threshold for each algorithm. We trained each algorithm five times to compute the mean and standard deviation.

| Model   | Simulation Gene               |                               | EMT                          |                              |
|---|-------------------------------|-------------------------------|------------------------------|------------------------------|
|   | Epoch                         | Wall Time                     | Epoch                        | Wall Time                    |
| TIGON (Sha et al., 2024)                            | 228.40 <sub>±223.71</sub>     | 1142.79 <sub>±1345.21</sub>   | 110.40 <sub>±193.37</sub>    | 365.54 <sub>±639.86</sub>    |
| RUOT w/o Pretraining (Zhang et al., 2025a)          | 172.00 <sub>±229.11</sub>     | 578.67 <sub>±768.52</sub>     | 228.20 <sub>±223.88</sub>    | 819.31 <sub>±804.05</sub>    |
| RUOT with 3 Epoch Pretraining (Zhang et al., 2025a) | 204.40 <sub>±238.29</sub>     | 653.33 <sub>±761.35</sub>     | 221.60 <sub>±226.52</sub>    | 801.18 <sub>±819.46</sub>    |
| Var-RUOT (Ours)                                     | <b>27.60</b> <sub>±5.75</sub> | <b>33.98</b> <sub>±6.37</sub> | <b>5.20</b> <sub>±1.26</sub> | <b>7.37</b> <sub>±1.89</sub> |

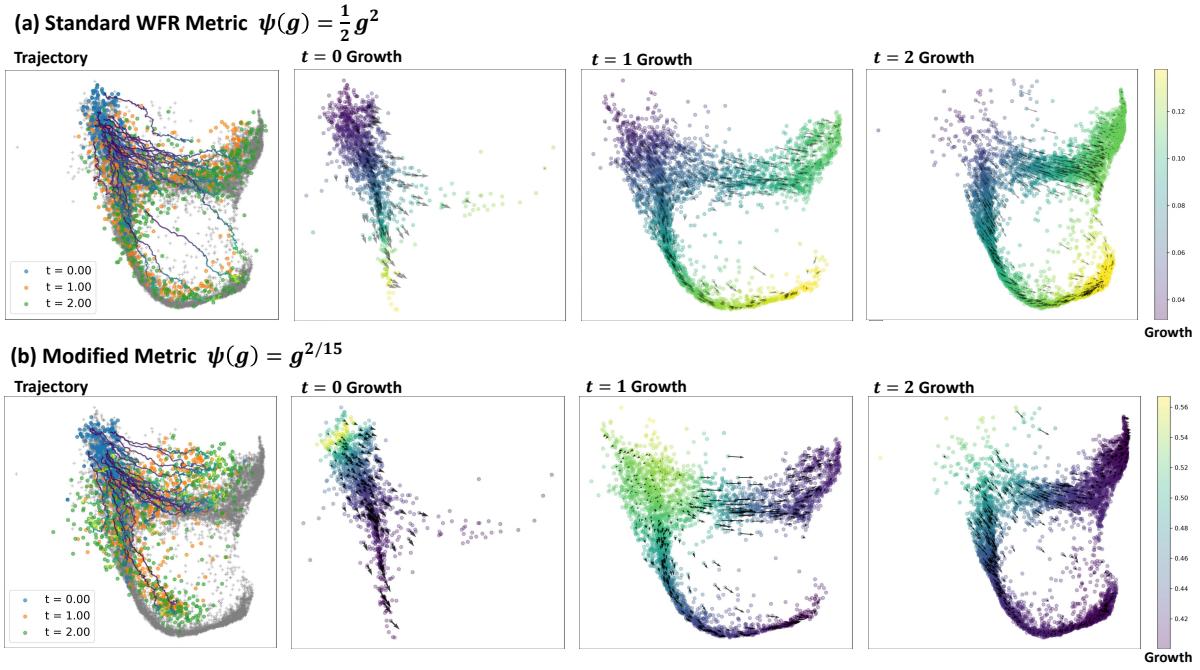
### 6.3 Different $\psi(g)$ Represents Different Biological Prior

To illustrate that the choice of  $\psi(g)$  represents different biological priors, we present the learned dynamics under two selections of  $\psi(g)$ . We apply our algorithm on the Mouse Blood Hematopoiesis dataset (Weinreb et al., 2020; Sha et al., 2024). In Fig. 4(a), the standard WFR metric is applied, i.e.,  $\psi_1(g) = \frac{1}{2}g^2$ , from which it can be observed that at time points  $t = 0, 1, 2$ , along the direction of the drift vector field  $\mathbf{u}(\mathbf{x}, t)$ ,  $g(\mathbf{x}, t)$  gradually increases; in Fig. 4(b), on the other hand, the alternative selection  $\psi_2(g) = g^{2/15}$  mentioned in Section 5.3 is used, and it is evident that at each time point,  $g(\mathbf{x}, t)$  gradually decreases along the direction of  $\mathbf{u}(\mathbf{x}, t)$ . The distribution matching accuracy and the action are reported in Table 4. When employing the modified metric, the corresponding action quantity is not directly comparable to those obtained using the WFR metric; therefore, we do not report its action here.

**Table 4.** Wasserstein distances ( $\mathcal{W}_1$  and  $\mathcal{W}_2$ ) between the predicted distributions of each algorithm and the true distribution at various time points on mouse blood hematopoiesis. Each experiment was run five times to compute the mean and standard deviation.

| Model                                    | $t = 1$                          |                                  | $t = 2$                          |                                  | Path Action                      |
|--|----------------------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|
|  | $\mathcal{W}_1$                  | $\mathcal{W}_2$                  | $\mathcal{W}_1$                  | $\mathcal{W}_2$                  |                                  |
| Action Matching (Neklyudov et al., 2023) | 0.4719 <sub>±0.0000</sub>        | 0.5673 <sub>±0.0000</sub>        | 0.8350 <sub>±0.0000</sub>        | 0.8936 <sub>±0.0000</sub>        | 4.3517                           |
| TIGON (Sha et al., 2024)                 | 0.4498 <sub>±0.0000</sub>        | 0.5139 <sub>±0.0000</sub>        | 0.4368 <sub>±0.0000</sub>        | 0.4852 <sub>±0.0000</sub>        | 3.7438                           |
| DeepRUOT (Zhang et al., 2025a)           | 0.1456 <sub>±0.0016</sub>        | 0.1807 <sub>±0.0019</sub>        | 0.1469 <sub>±0.0046</sub>        | 0.1791 <sub>±0.0061</sub>        | 5.5887                           |
| Var-RUOT (Standard WFR)                  | <b>0.1200</b> <sub>±0.0038</sub> | <b>0.1459</b> <sub>±0.0038</sub> | <b>0.1431</b> <sub>±0.0092</sub> | <b>0.1764</b> <sub>±0.0135</sub> | <b>3.1491</b> <sub>±0.0837</sub> |
| Var-RUOT (Modified Metric)               | 0.2953 <sub>±0.0357</sub>        | 0.3117 <sub>±0.0323</sub>        | 0.1917 <sub>±0.0140</sub>        | 0.2226 <sub>±0.0170</sub>        | -                                |

In addition to the three experiments presented here, we also conducted ablation studies on the weights of the HJB loss and the action loss to verify the effectiveness of these loss terms in learning dynamics with smaller action (Appendix C.2). We also performed a hold-one-out experiment, and the results indicate that the Var-RUOT algorithm can effectively perform both interpolation and extrapolation, with learning minimum-action dynamics leading to more accurate extrapolation outcomes



**Figure 4.** a) The trajectory and growth at time points  $t = 0, 1, 2$  learned using the standard WFR metric; b) The trajectory and growth at time points  $t = 0, 1, 2$  learned using the modified metric.

(Appendix C.3). Furthermore, we carried out experiments on several high-dimensional datasets, which further validate the effectiveness of Var-RUOT in high-dimensional settings (Appendix C.4).

## 7 Conclusion

In this paper, we propose a new algorithm for solving the RUOT problem called Variational RUOT. By employing variational methods to derive the necessary conditions for the minimum action solution of RUOT, we solve the problem by learning a single scalar field. Compared to other algorithms, Var-RUOT can find solutions with lower action values while achieving the same level of fitting performance, and it offers faster training and convergence speeds. Finally, we emphasize that the selection of  $\psi(g)$  in the action is crucial and directly linked to biological priors. We also discussed the limitations of our work and potential directions for future research in Appendix D.1.

## Acknowledgements

This work was supported by the National Key R&D Program of China (No. 2021YFA1003301 to T.L.) and National Natural Science Foundation of China (NSFC No. 12288101 to T.L. & P.Z., and 8206100646, T2321001 to P.Z.). We acknowledge the support from the High-performance Computing Platform of Peking University for computation.

## References

- Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.
- Lazar Atanackovic, Xi Zhang, Brandon Amos, Mathieu Blanchette, Leo J Lee, Yoshua Bengio, Alexander Tong, and Kirill Neklyudov. Meta flow matching: Integrating vector fields on the wasserstein manifold. In *The Thirteenth International Conference on Learning Representations*, 2025.
- Aymeric Baradat and Hugo Lavenant. Regularized unbalanced optimal transport as entropy minimization with respect to branching brownian motion. *arXiv preprint arXiv:2111.01666*, 2021.
- Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

- Charlotte Bunne, Ya-Ping Hsieh, Marco Cuturi, and Andreas Krause. The schrödinger bridge between gaussian measures has a closed form. In *International Conference on Artificial Intelligence and Statistics*, pp. 5802–5833. PMLR, 2023a.
- Charlotte Bunne, Stefan G Stark, Gabriele Gut, Jacobo Sarabia Del Castillo, Mitch Levesque, Kjong-Van Lehmann, Lucas Pelkmans, Andreas Krause, and Gunnar Rätsch. Learning single-cell perturbation responses using neural optimal transport. *Nature methods*, 20(11):1759–1768, 2023b.
- Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.
- Tianrong Chen, Guan-Horng Liu, and Evangelos Theodorou. Likelihood training of schrödinger bridge using forward-backward SDEs theory. In *International Conference on Learning Representations*, 2022a.
- Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. On the relation between optimal transport and schrödinger bridges: A stochastic control viewpoint. *Journal of Optimization Theory and Applications*, 169:671–691, 2016.
- Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. The most likely evolution of diffusing and vanishing particles: Schrodinger bridges with unbalanced marginals. *SIAM Journal on Control and Optimization*, 60(4):2016–2039, 2022b.
- Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. An interpolating distance between optimal transport and fisher-rao metrics. *Foundations of Computational Mathematics*, 18:1–44, 2018a.
- Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Unbalanced optimal transport: Dynamic and kantorovich formulations. *Journal of Functional Analysis*, 274(11):3090–3123, 2018b.
- Lénaïc Chizat, Stephen Zhang, Matthieu Heitz, and Geoffrey Schiebinger. Trajectory inference via mean-field langevin in path space. *Advances in Neural Information Processing Systems*, 35:16731–16742, 2022.
- Shui-Nee Chow, Wuchen Li, and Haomin Zhou. Wasserstein hamiltonian flows. *Journal of Differential Equations*, 268(3): 1205–1219, 2020.
- David P Cook and Barbara C Vanderhyden. Context specificity of the emt transcriptional response. *Nature communications*, 11 (1):2142, 2020.
- Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.
- Jun Ding, Nadav Sharon, and Ziv Bar-Joseph. Temporal modelling using single-cell transcriptomics. *Nature Reviews Genetics*, 23(6):355–368, 2022.
- Luca Eyring, Dominik Klein, Théo Uscidda, Giovanni Palla, Niki Kilbertus, Zeynep Akata, and Fabian J Theis. Unbalancedness in neural monge maps improves unpaired domain translation. In *The Twelfth International Conference on Learning Representations*, 2024.
- Ivan Gentil, Christian Léonard, and Luigia Ripani. About the analogy between optimal transport and minimal entropy. In *Annales de la Faculté des sciences de Toulouse: Mathématiques*, volume 26, pp. 569–600, 2017.
- Anming Gu, Edward Chien, and Kristjan Greenewald. Partially observed trajectory inference using optimal transport and a dynamics prior. In *The Thirteenth International Conference on Learning Representations*, 2025.
- Matthieu Heitz, Yujia Ma, Sharvaj Kubal, and Geoffrey Schiebinger. Spatial transcriptomics brings new challenges and opportunities for trajectory inference. *Annual Review of Biomedical Data Science*, 8, 2024.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- Guillaume Huguet, Daniel Sumner Magruder, Alexander Tong, Oluwadamilola Fasina, Manik Kuchroo, Guy Wolf, and Smita Krishnaswamy. Manifold interpolating optimal-transport flows for trajectory inference. *Advances in neural information processing systems*, 35:29705–29718, 2022.
- Qi Jiang and Lin Wan. A physics-informed neural SDE network for learning cellular dynamics from time-series scRNA-seq data. *Bioinformatics*, 40:ii120–ii127, 09 2024. ISSN 1367-4811.

- Dominik Klein, Giovanni Palla, Marius Lange, Michal Klein, Zoe Piran, Manuel Gander, Laetitia Meng-Papaxanthos, Michael Stern, Lama Saber, Changying Jing, et al. Mapping cells through time and space with moscot. *Nature*, pp. 1–11, 2025.
- Takeshi Koshizuka and Issei Sato. Neural lagrangian schrödinger bridge: Diffusion modeling for population dynamics. In *The Eleventh International Conference on Learning Representations*, 2023.
- Hugo Lavenant, Stephen Zhang, Young-Heon Kim, Geoffrey Schiebinger, et al. Toward a mathematical theory of trajectory inference. *The Annals of Applied Probability*, 34(1A):428–500, 2024.
- Christian Léonard. A survey of the schrödinger problem and some of its connections with optimal transport. *Discrete and Continuous Dynamical Systems-Series A*, 34(4):1533–1574, 2014.
- Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.
- Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.
- Suryanarayana Maddu, Victor Chardès, Michael Shelley, et al. Inferring biological processes with intrinsic noise from cross-sectional data. *arXiv preprint arXiv:2410.07501*, 2024.
- Kirill Neklyudov, Rob Brekelmans, Daniel Severo, and Alireza Makhzani. Action matching: Learning stochastic dynamics from samples. In *International conference on machine learning*, pp. 25858–25889. PMLR, 2023.
- Kirill Neklyudov, Rob Brekelmans, Alexander Tong, Lazar Atanackovic, Qiang Liu, and Alireza Makhzani. A computational framework for solving wasserstein lagrangian flows. In *Forty-first International Conference on Machine Learning*, 2024.
- Alessandro Palma, Till Richter, Hanyi Zhang, Manuel Lubetzki, Alexander Tong, Andrea Dittadi, and Fabian J Theis. Multi-modal and multi-attribute generation of single cells with CFGen. In *The Thirteenth International Conference on Learning Representations*, 2025.
- Matteo Pariset, Ya-Ping Hsieh, Charlotte Bunne, Andreas Krause, and Valentin De Bortoli. Unbalanced diffusion schrödinger bridge. In *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*, 2023.
- Qiangwei Peng, Peijie Zhou, and Tiejun Li. stver: Reconstructing spatio-temporal dynamics of cell development using optimal transport. *bioRxiv*, pp. 2024–06, 2024.
- Katarina Petrović, Lazar Atanackovic, Kacper Kapusniak, Michael M. Bronstein, Joey Bose, and Alexander Tong. Curly flow matching for learning non-gradient field dynamics. In *Learning Meaningful Representations of Life (LMRL) Workshop at ICLR 2025*, 2025.
- Martin Rohbeck, Charlotte Bunne, Edward De Brouwer, Jan-Christian Huetter, Anne Biton, Kelvin Y. Chen, Aviv Regev, and Romain Lopez. Modeling complex system dynamics with flow matching across time and conditions. In *The Thirteenth International Conference on Learning Representations*, 2025.
- Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, et al. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943, 2019a.
- Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, et al. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943, 2019b.
- Yutong Sha, Yuchi Qiu, Peijie Zhou, and Qing Nie. Reconstructing growth and dynamic trajectories from single-cell transcriptomics data. *Nature Machine Intelligence*, 6(1):25–39, 2024.
- Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. *Advances in Neural Information Processing Systems*, 36, 2024.
- Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Alexander Tong, Jessie Huang, Guy Wolf, David Van Dijk, and Smita Krishnaswamy. Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics. In *International conference on machine learning*, pp. 9526–9536. PMLR, 2020.

Alexander Tong, Manik Kuchroo, Shabarni Gupta, Aarthi Venkat, Beatriz P San Juan, Laura Rangel, Brandon Zhu, John G Lock, Christine L Chaffer, and Smita Krishnaswamy. Learning transcriptional and regulatory dynamics driving cancer cell plasticity using neural ode-based optimal transport. *bioRxiv*, pp. 2023–03, 2023.

Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024a. ISSN 2835-8856. Expert Certification.

Alexander Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-free schrödinger bridges via score and flow matching. In *International Conference on Artificial Intelligence and Statistics*, pp. 1279–1287. PMLR, 2024b.

Elias Ventre, Aden Forrow, Nitya Gadhiwala, Parijat Chakraborty, Omer Angel, and Geoffrey Schiebinger. Trajectory inference for a branching sde model of cell differentiation. *arXiv preprint arXiv:2307.07687*, 2023.

Adrian Veres, Aubrey L Faust, Henry L Bushnell, Elise N Engquist, Jennifer Hyoje-Ryu Kenty, George Harb, Yeh-Chuin Poh, Elad Sintov, Mads Gürtler, Felicia W Pagliuca, et al. Charting cellular identity during human in vitro  $\beta$ -cell differentiation. *Nature*, 569(7756):368–373, 2019.

Wei Wan, Yuejin Zhang, Chenglong Bao, Bin Dong, and Zuoqiang Shi. A scalable deep learning approach for solving high-dimensional dynamic optimal transport. *SIAM Journal on Scientific Computing*, 45(4):B544–B563, 2023.

Caleb Weinreb, Alejo Rodriguez-Fraticelli, Fernando D Camargo, and Allon M Klein. Lineage tracing on transcriptional landscapes links state to fate during differentiation. *Science*, 367(6479):eaaw3381, 2020.

Hao Wu, Shu Liu, Xiaojing Ye, and Haomin Zhou. Parameterized wasserstein hamiltonian flow. *SIAM Journal on Numerical Analysis*, 63(1):360–395, 2025.

Liu Yang, Constantinos Daskalakis, and George E Karniadakis. Generative ensemble regression: Learning particle dynamics from observations of ensembles with physics-informed deep generative models. *SIAM Journal on Scientific Computing*, 44(1):B80–B99, 2022.

Maosheng Yang. Topological schrödinger bridge matching. In *The Thirteenth International Conference on Learning Representations*, 2025.

Grace Hui Ting Yeo, Sachit D Saksena, and David K Gifford. Generative modeling of single-cell time series with prescient enables prediction of cell trajectories with interventions. *Nature communications*, 12(1):3222, 2021a.

Grace Hui Ting Yeo, Sachit D Saksena, and David K Gifford. Generative modeling of single-cell time series with prescient enables prediction of cell trajectories with interventions. *Nature communications*, 12(1):3222, 2021b.

Yuning You, Ruida Zhou, and Yang Shen. Correlational lagrangian schrödinger bridge: Learning dynamics with population-level regularization. *arXiv preprint arXiv:2402.10227*, 2024.

Jiaqi Zhang, Erica Larschan, Jeremy Bigness, and Ritambhara Singh. scNODE: generative model for temporal single cell transcriptomic data prediction. *Bioinformatics*, 40(Supplement\_2):ii146–ii154, 09 2024a. ISSN 1367-4811.

Peng Zhang, Ting Gao, Jin Guo, and Jinqiao Duan. Action functional as early warning indicator in the space of probability measures. *arXiv preprint arXiv:2403.10405*, 2024b.

Stephen Zhang, Anton Afanassiev, Laura Greenstreet, Tetsuya Matsumoto, and Geoffrey Schiebinger. Optimal transport analysis reveals trajectories in steady-state systems. *PLoS computational biology*, 17(12):e1009466, 2021.

Zhenyi Zhang, Tiejun Li, and Peijie Zhou. Learning stochastic dynamics from snapshots through regularized unbalanced optimal transport. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Zhenyi Zhang, Yuhao Sun, Qiangwei Peng, Tiejun Li, and Peijie Zhou. Integrating dynamical systems modeling with spatiotemporal scRNA-seq data analysis. *Entropy*, 27(5), 2025b. ISSN 1099-4300.

Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. In *The Twelfth International Conference on Learning Representations*, 2024.

Qunxi Zhu, Bolin Zhao, Jingdong Zhang, Peiyang Li, and Wei Lin. Governing equation discovery of a complex system from snapshots. *arXiv preprint arXiv:2410.16694*, 2024.

## A Technical Details

### A.1 Proof of Theorems

#### A.1.1 Proof for Theorem 4.1

**Theorem A.1.** *The RUOT problem with isotropic and time-invariant diffusion intensity is formulated as:*

$$\min_{\rho, g} A = \int_0^1 \int_{\mathbb{R}^d} \left( \frac{1}{2} \|\mathbf{u}(\mathbf{x}, t)\|^2 + \alpha \psi(g(\mathbf{x}, t)) \right) \rho(\mathbf{x}, t) d\mathbf{x} dt \quad (6)$$

$$s.t. \quad \frac{\partial \rho(\mathbf{x}, t)}{\partial t} = -\nabla \cdot (\rho(\mathbf{x}, t) \mathbf{u}(\mathbf{x}, t)) + \frac{1}{2} \sigma^2 \nabla^2 \rho(\mathbf{x}, t) + g(\mathbf{x}, t) \rho(\mathbf{x}, t) \quad (7)$$

In this problem, the necessary conditions for the action  $A$  to achieve a minimum are given by:

$$\mathbf{u} = \nabla_{\mathbf{x}} \lambda, \quad \alpha \frac{d\psi(g)}{dg} = \lambda, \quad \frac{\partial \lambda}{\partial t} + \frac{1}{2} \|\nabla_{\mathbf{x}} \lambda\|^2 + \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 \lambda + \lambda g - \alpha \psi(g) = 0, \quad (8)$$

where  $\lambda(\mathbf{x}, t)$  is a scalar field.

*Proof.* In order to incorporate the constraints of the Fokker–Planck equation, we construct an augmented action functional:

$$\begin{aligned} A^\dagger &= \int_0^1 \int_{\mathbb{R}^d} \left( \frac{1}{2} \|\mathbf{u}(\mathbf{x}, t)\|^2 + \alpha \psi(g(\mathbf{x}, t)) \right) \rho(\mathbf{x}, t) d\mathbf{x} dt \\ &\quad + \int_0^1 \int_{\mathbb{R}^d} \lambda(\mathbf{x}, t) \left( \frac{\partial \rho(\mathbf{x}, t)}{\partial t} + \nabla \cdot (\rho(\mathbf{x}, t) \mathbf{u}(\mathbf{x}, t)) - \frac{1}{2} \sigma^2 \nabla^2 \rho(\mathbf{x}, t) - g(\mathbf{x}, t) \rho(\mathbf{x}, t) \right) d\mathbf{x} dt \end{aligned}$$

We take variations with respect to  $\mathbf{u}$ ,  $g$ , and  $\rho$ . At the stationary point of the functional, the variation of the augmented action functional must vanish.

#### Step 1: Variation with respect to $\mathbf{u}$ .

Let  $\mathbf{u} \rightarrow \mathbf{u} + \delta \mathbf{u}$ . The variation of the augmented action is

$$\begin{aligned} \delta A^\dagger &= \int_0^1 \int_{\mathbb{R}^d} (\mathbf{u}^T \delta \mathbf{u}) \rho + \lambda \nabla_{\mathbf{x}} \cdot (\rho \delta \mathbf{u}) d\mathbf{x} dt \\ &= \int_0^1 \int_{\mathbb{R}^d} (\mathbf{u}^T \delta \mathbf{u}) \rho + \int_0^1 \int_{\mathbb{R}^d} (\nabla_{\mathbf{x}} \cdot (\lambda \rho \delta \mathbf{u}) - \rho (\nabla_{\mathbf{x}} \lambda)^T \delta \mathbf{u}) d\mathbf{x} dt \\ &= \int_0^1 \int_{\mathbb{R}^d} (\mathbf{u}^T \delta \mathbf{u}) \rho + \int_0^1 \int_{S^\infty} \lambda \rho (\delta \mathbf{u})^T d\mathbf{S} - \int_0^1 \int_{\mathbb{R}^d} \rho (\nabla_{\mathbf{x}} \lambda)^T \delta \mathbf{u} d\mathbf{x} dt \\ &= \int_0^1 \int_{\mathbb{R}^d} (\mathbf{u}^T - (\nabla_{\mathbf{x}} \lambda)^T) \rho \delta \mathbf{u} d\mathbf{x} dt \end{aligned}$$

Here,  $S^\infty$  denotes the boundary at infinity in  $\mathbb{R}^d$  and  $d\mathbf{S}$  is the surface element. Based on the assumption that

$$\int_{S^\infty} \lambda \rho (\delta \mathbf{u})^T d\mathbf{S} = 0,$$

and using the arbitrariness of  $\delta \mathbf{u}$ , we obtain the optimality condition

$$\mathbf{u} = \nabla_{\mathbf{x}} \lambda.$$

**Step 2: Variation with respect to  $g$ .**

Let  $g \rightarrow g + \delta g$ , then the variation of the augmented action becomes

$$\delta A^\dagger = \int_0^1 \int_{\mathbb{R}^d} \left( \alpha \frac{d\psi(g)}{dg} - \lambda \right) \rho \delta g d\mathbf{x} dt.$$

Since  $\delta g$  is arbitrary, we immediately obtain the optimality condition

$$\alpha \frac{d\psi(g)}{dg} = \lambda.$$

**Step 3: Variation with respect to  $\rho$ .**

Let  $\rho \rightarrow \rho + \delta \rho$ . Then the variation of the augmented action is given by

$$\begin{aligned} \delta A^\dagger &= \int_0^1 \int_{\mathbb{R}^d} \left[ \left( \frac{1}{2} \|\mathbf{u}\|^2 + \alpha \psi(g) \right) \delta \rho + \lambda \left( \frac{\partial \delta \rho}{\partial t} + \nabla_{\mathbf{x}} \cdot (\mathbf{u} \delta \rho) - \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 (\delta \rho) - g \delta \rho \right) \right] d\mathbf{x} dt \\ &= \int_0^1 \int_{\mathbb{R}^d} \left( \frac{1}{2} \|\mathbf{u}\|^2 + \alpha \psi(g) - \lambda g \right) \delta \rho d\mathbf{x} dt + \int_{\mathbb{R}^d} \int_0^1 \left( \frac{\partial(\lambda \delta \rho)}{\partial t} - \delta \rho \frac{\partial \lambda}{\partial t} \right) dt d\mathbf{x} \\ &\quad + \int_0^1 \int_{\mathbb{R}^d} (\nabla_{\mathbf{x}} \cdot (\mathbf{u} \lambda \delta \rho) d\mathbf{x} dt - \mathbf{u}^T (\nabla_{\mathbf{x}} \lambda) \delta \rho) d\mathbf{x} dt \\ &\quad - \frac{1}{2} \sigma^2 \int_0^1 \int_{\mathbb{R}^d} (\nabla_{\mathbf{x}} \cdot (\lambda \nabla_{\mathbf{x}} \delta \rho) - (\nabla_{\mathbf{x}} \lambda)^T (\nabla_{\mathbf{x}} \delta \rho)) d\mathbf{x} dt \\ &= \int_0^1 \int_{\mathbb{R}^d} \left( \frac{1}{2} \|\mathbf{u}\|^2 + \alpha \psi(g) - \lambda g - \frac{\partial \lambda}{\partial t} - \mathbf{u}^T (\nabla_{\mathbf{x}} \lambda) \right) \delta \rho d\mathbf{x} dt + \int_{\mathbb{R}^d} [\lambda \delta \rho]_{t=0}^{t=1} d\mathbf{x} \\ &\quad + \int_0^1 \int_{S^\infty} \nabla_{\mathbf{x}} \cdot (\mathbf{u} \lambda \delta \rho)^T d\mathbf{S} dt - \frac{1}{2} \sigma^2 \int_0^1 \int_{S^\infty} (\lambda \nabla_{\mathbf{x}} \delta \rho)^T d\mathbf{S} dt \\ &\quad + \frac{1}{2} \sigma^2 \int_0^1 \int_{\mathbb{R}^d} (\nabla_{\mathbf{x}} \lambda)^T (\nabla_{\mathbf{x}} \delta \rho) d\mathbf{x} dt \\ &= \int_0^1 \int_{\mathbb{R}^d} \left( \frac{1}{2} \|\mathbf{u}\|^2 + \alpha \psi(g) - \lambda g - \frac{\partial \lambda}{\partial t} - \mathbf{u}^T (\nabla_{\mathbf{x}} \lambda) \right) \delta \rho d\mathbf{x} dt \\ &\quad + \frac{1}{2} \sigma^2 \int_0^1 \int_{\mathbb{R}^d} \left( \nabla_{\mathbf{x}} \cdot ((\nabla_{\mathbf{x}} \lambda) \delta \rho) - (\nabla_{\mathbf{x}}^2 \lambda) \delta \rho \right) d\mathbf{x} dt \\ &= \int_0^1 \int_{\mathbb{R}^d} \left( \frac{1}{2} \|\mathbf{u}\|^2 + \alpha \psi(g) - \lambda g - \frac{\partial \lambda}{\partial t} - \mathbf{u}^T (\nabla_{\mathbf{x}} \lambda) - \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 \lambda \right) \delta \rho d\mathbf{x} dt \\ &\quad - \frac{1}{2} \sigma^2 \int_0^1 \int_{S^\infty} ((\nabla_{\mathbf{x}} \lambda) \delta \rho)^T d\mathbf{S} dt \\ &= \int_0^1 \int_{\mathbb{R}^d} \left( \frac{1}{2} \|\mathbf{u}\|^2 + \alpha \psi(g) - \lambda g - \frac{\partial \lambda}{\partial t} - \mathbf{u}^T (\nabla_{\mathbf{x}} \lambda) - \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 \lambda \right) \delta \rho d\mathbf{x} dt. \end{aligned}$$

Since  $\delta \rho$  is arbitrary, the corresponding optimality condition is

$$\frac{1}{2} \|\mathbf{u}\|^2 + \alpha \psi(g) - \lambda g - \frac{\partial \lambda}{\partial t} - \mathbf{u}^T (\nabla_{\mathbf{x}} \lambda) - \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 \lambda = 0.$$

Substituting the previously obtained condition  $\mathbf{u} = \nabla_{\mathbf{x}} \lambda$ , we arrive at the final optimality condition:

$$\frac{\partial \lambda}{\partial t} + \frac{1}{2} \|\nabla_{\mathbf{x}} \lambda\|^2 + \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 \lambda + \lambda g - \alpha \psi(g) = 0.$$

□

### A.1.2 Proof for Theorem 4.2

**Theorem A.2.** The choice of  $\psi(g)$  affects whether  $g$  ascends or descends along the direction of the velocity field  $\mathbf{u}$  at a given time. Specifically, at a fixed time  $t$ , if

$$\frac{d\psi^2(g)}{dg^2} > 0, \quad (9)$$

then  $g(\mathbf{x}, t)$  ascends in the direction of the velocity field  $\mathbf{u}(\mathbf{x}, t)$  (i.e.,  $\mathbf{u}(\mathbf{x}, t)^T (\nabla_{\mathbf{x}} g(\mathbf{x}, t)) > 0$ ); otherwise, it descends.

*Proof.* Let

$$\frac{d\psi(g)}{dg} = \psi'(g) \quad \text{and} \quad \frac{d^2\psi(g)}{dg^2} = \psi''(g).$$

Using the optimality condition for  $g$  from [Appendix A.1.1](#),

$$\alpha \frac{d\psi(g)}{dg} = \lambda,$$

taking the gradient with respect to  $x$  on both sides yields

$$\nabla_{\mathbf{x}} g(\mathbf{x}, t) = \frac{1}{\alpha \psi''(g)} \nabla_{\mathbf{x}} \lambda(\mathbf{x}, t).$$

The condition for  $g$  to increase along the velocity field is that the inner product between  $\nabla_{\mathbf{x}} g(\mathbf{x}, t)$  and  $\mathbf{u}(\mathbf{x}, t)$  is positive everywhere. Using the optimality condition for the velocity,

$$\mathbf{u}(\mathbf{x}, t) = \nabla_{\mathbf{x}} \lambda(\mathbf{x}, t),$$

we have

$$\mathbf{u}(\mathbf{x}, t)^T \nabla_{\mathbf{x}} g(\mathbf{x}, t) = \frac{1}{\alpha \psi''(g)} \|\nabla_{\mathbf{x}} \lambda(\mathbf{x}, t)\|^2.$$

Since  $\alpha > 0$ , the condition

$$\mathbf{u}(\mathbf{x}, t)^T \nabla_{\mathbf{x}} g(\mathbf{x}, t) > 0$$

is equivalent to requiring that

$$\psi''(g) > 0 \quad \forall g.$$

□

### A.1.3 Proof for Theorem 5.1

**Theorem A.3.** Consider a weighted particle system consisting of  $N$  particles, where the position of particle  $i$  is given by  $\mathbf{X}_i^t \in \mathbb{R}^d$  and its weight by  $w_i(t) > 0$ . The dynamics of each particle are described by

$$\begin{aligned} \frac{d\mathbf{X}_i^t}{dt} &= \mathbf{u}(\mathbf{X}_i^t, t) dt + \sigma(t) d\mathbf{W}_i, \\ \frac{dw_i}{dt} &= g(\mathbf{X}_i^t, t) w_i dt, \end{aligned} \quad (10)$$

where  $\mathbf{u} : \mathbb{R}^d \times [0, T] \rightarrow \mathbb{R}^d$  is a time-varying vector field,  $g : \mathbb{R}^d \times [0, T] \rightarrow \mathbb{R}$  is a growth rate function,  $\sigma : [0, T] \rightarrow [0, +\infty)$  is a time-varying diffusion coefficient, and  $\mathbf{W}_i$  is an  $N$ -dimensional standard Brownian motion with independent components in each coordinate. The initial conditions are  $\mathbf{X}_i^0 \sim \rho(\mathbf{x}, 0)$  and  $w_i(0) = 1$ . In the limit as  $N \rightarrow \infty$ , the empirical measure

$$\mu^N = \frac{1}{N} \sum_{i=1}^N w_i(t) \delta(\mathbf{x} - \mathbf{X}_i^t) \quad (11)$$

converges to the solution of the following Fokker–Planck equation:

$$\frac{\partial \rho(\mathbf{x}, t)}{\partial t} = -\nabla_{\mathbf{x}} \cdot \left( \mathbf{u}(\mathbf{x}, t) \rho(\mathbf{x}, t) \right) + \frac{1}{2} \sigma^2(t) \nabla_{\mathbf{x}}^2 \rho(\mathbf{x}, t) + g(\mathbf{x}, t) \rho(\mathbf{x}, t), \quad (12)$$

with the initial condition  $\rho(\mathbf{x}, 0) = \rho_0(\mathbf{x})$ .

*Proof.* Consider a smooth test function  $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$ . We study the evolution of the expectation

$$\int_{\mathbb{R}^d} \phi(\mathbf{x}) \mu^N(\mathbf{x}, t) d\mathbf{x} = \frac{1}{N} \sum_{i=1}^N w_i(t) \phi(\mathbf{X}_i^t).$$

By applying Itô's formula, we have

$$d\left(w_i(t) \phi(\mathbf{X}_i^t)\right) = w_i(t) d\phi(\mathbf{X}_i^t) + \phi(\mathbf{X}_i^t) dw_i(t) + dw_i(t) d\phi(\mathbf{X}_i^t).$$

Using Itô's formula to compute  $d\phi(\mathbf{X}_i^t)$ , we obtain

$$d\phi(\mathbf{X}_i^t) = (\nabla_{\mathbf{x}} \phi(\mathbf{X}_i^t))^T d\mathbf{X}_i^t + \frac{1}{2} \nabla_{\mathbf{x}}^2 \phi(\mathbf{X}_i^t) \sigma^2(t) dt.$$

Since  $dw_i(t) = g(\mathbf{X}_i^t, t) w_i(t) dt$  contains no stochastic term (i.e., there is no  $d\mathbf{W}$ ), the term  $dw_i(t) d\phi(\mathbf{X}_i^t)$  is of higher order and can be neglected. Therefore, we have:

$$\begin{aligned} d\left(w_i(t) \phi(\mathbf{X}_i^t)\right) &= \phi(\mathbf{X}_i^t) g(\mathbf{X}_i^t, t) w_i(t) dt \\ &\quad + w_i(t) (\nabla_{\mathbf{x}} \phi(\mathbf{X}_i^t))^T \left( \mathbf{u}(\mathbf{X}_i^t, t) dt + \sigma(t) d\mathbf{W}_t \right) \\ &\quad + \frac{1}{2} \nabla_{\mathbf{x}}^2 \phi(\mathbf{X}_i^t) \sigma^2(t) dt. \end{aligned}$$

Next, we compute

$$\begin{aligned} \mathbb{E}\left[\frac{d}{dt} \left( \frac{1}{N} \sum_{i=1}^N w_i(t) \phi(\mathbf{X}_i^t) \right)\right] &= \mathbb{E}\left[ \frac{1}{N} \sum_{i=1}^N \left( \phi(\mathbf{X}_i^t) g(\mathbf{X}_i^t, t) w_i(t) \right. \right. \\ &\quad \left. \left. + w_i(t) (\nabla_{\mathbf{x}} \phi(\mathbf{X}_i^t))^T \mathbf{u}(\mathbf{X}_i^t, t) \right. \right. \\ &\quad \left. \left. + \frac{1}{2} \nabla_{\mathbf{x}}^2 \phi(\mathbf{X}_i^t) \sigma^2(t) \right) \right]. \end{aligned}$$

Thus, in the limit as  $N \rightarrow \infty$ , and let  $\rho(\mathbf{x}, t) = \mu^\infty(\mathbf{x}, t)$ , we have

$$\begin{aligned} \frac{d}{dt} \int_{\mathbb{R}^d} \phi(\mathbf{x}) \rho(\mathbf{x}, t) d\mathbf{x} &= \int_{\mathbb{R}^d} \left( g(\mathbf{x}, t) \rho(\mathbf{x}, t) \phi(\mathbf{x}) + \rho(\mathbf{x}, t) (\nabla_{\mathbf{x}} \phi(\mathbf{x}))^T \mathbf{u}(\mathbf{x}, t) \right. \\ &\quad \left. + \frac{1}{2} \sigma^2(t) \rho(\mathbf{x}, t) \nabla_{\mathbf{x}}^2 \phi(\mathbf{x}) \right) d\mathbf{x}. \end{aligned}$$

By integrating by parts, we obtain

$$\int_{\mathbb{R}^d} \rho(\mathbf{x}, t) (\nabla_{\mathbf{x}} \phi(\mathbf{x}))^T \mathbf{u}(\mathbf{x}, t) d\mathbf{x} = - \int_{\mathbb{R}^d} \phi(\mathbf{x}, t) \nabla_{\mathbf{x}} \cdot (\mathbf{u}(\mathbf{x}, t) \rho(\mathbf{x}, t)) d\mathbf{x},$$

and

$$\int_{\mathbb{R}^d} \rho(\mathbf{x}, t) \nabla_{\mathbf{x}}^2 \phi(\mathbf{x}) d\mathbf{x} = \int_{\mathbb{R}^d} \phi(\mathbf{x}, t) \nabla_{\mathbf{x}}^2 \rho(\mathbf{x}, t) d\mathbf{x}.$$

Hence, we deduce that

$$\frac{d}{dt} \int_{\mathbb{R}^d} \phi(\mathbf{x}, t) \rho(\mathbf{x}, t) d\mathbf{x} = \int_{\mathbb{R}^d} \phi(\mathbf{x}, t) \left[ -\nabla_{\mathbf{x}} \cdot (\mathbf{u}(\mathbf{x}, t) \rho(\mathbf{x}, t)) + \frac{1}{2} \sigma^2(t) \nabla_{\mathbf{x}}^2 \rho(\mathbf{x}, t) + g(\mathbf{x}, t) \rho(\mathbf{x}, t) \right] d\mathbf{x}.$$

Since  $\phi(\mathbf{x})$  is arbitrary, we obtain the Fokker–Planck equation:

$$\frac{\partial \rho(\mathbf{x}, t)}{\partial t} = -\nabla_{\mathbf{x}} \cdot (\mathbf{u}(\mathbf{x}, t) \rho(\mathbf{x}, t)) + \frac{1}{2} \sigma^2(t) \nabla_{\mathbf{x}}^2 \rho(\mathbf{x}, t) + g(\mathbf{x}, t) \rho(\mathbf{x}, t).$$

□

#### A.1.4 Proposition : the Expectation of HJB Loss

**Proposition A.1.** Consider the following HJB loss:

$$\mathcal{L}_{HJB}^N = \sum_{i=1}^N \left[ \int_0^{T_K} \frac{w_i(t)}{\sum_{j=1}^N w_j(t)} \left( \frac{\partial \lambda(\mathbf{X}_i^t, t)}{\partial t} + \frac{1}{2} \|\nabla_{\mathbf{x}} \lambda\|^2 + \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 \lambda + g(\mathbf{X}_i^t, t) - \alpha \psi(g) \right)^2 dt \right] \quad (13)$$

where

$$d\mathbf{X}_i^t = \mathbf{u}(\mathbf{X}_i^t, t) dt + \sigma(t) d\mathbf{W}_t, \quad dw_i = g(\mathbf{X}_i^t, t) w_i dt,$$

The expectation of HJB loss is

$$\mathbb{E}[\mathcal{L}_{HJB}^N] = \int_0^{T_K} \int_{\mathbb{R}^d} \hat{\rho}(\mathbf{x}, t) \left( \frac{\partial \lambda(\mathbf{x}, t)}{\partial t} + \frac{1}{2} \|\nabla_{\mathbf{x}} \lambda\|^2 + \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 \lambda + g(\mathbf{x}, t) - \alpha \psi(g) \right)^2 d\mathbf{x} dt \quad (14)$$

where  $\hat{\rho}(\mathbf{x}, t) = \frac{\rho(\mathbf{x}, t)}{\int_{\mathbb{R}^d} \rho(\mathbf{x}, t) d\mathbf{x}}$  is the normalized probability density.

*Proof.* Taking the expectation of  $\mathcal{L}_{HJB}^N$  is equivalent to drawing  $N$  particles each time to obtain  $\mathcal{L}_{HJB}^N$ , repeating this process infinitely many times, and computing the average of the  $\mathcal{L}_{HJB}^N$  obtained in each instance. Since the particles are independent, this operation is directly equivalent to taking the number of particles  $N \rightarrow \infty$ , thus:

$$\begin{aligned} \mathbb{E}[\mathcal{L}_{HJB}^N] &= \lim_{N \rightarrow \infty} \sum_{i=1}^N \left[ \int_0^{T_K} \frac{w_i(t)}{\sum_{j=1}^N w_j(t)} \left( \frac{\partial \lambda}{\partial t} + \frac{1}{2} \|\nabla_{\mathbf{x}} \lambda\|^2 + \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 \lambda + g(\mathbf{X}_i^t, t) - \alpha \psi(g) \right)^2 dt \right] \\ &= \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \left[ \int_0^{T_K} \frac{1}{\frac{1}{N} \sum_{j=1}^N w_j(t)} \left( \frac{\partial \lambda}{\partial t} + \frac{1}{2} \|\nabla_{\mathbf{x}} \lambda\|^2 + \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 \lambda + g - \alpha \psi(g) \right)^2 dt \right] \\ &= \lim_{N \rightarrow \infty} \left[ \int_0^{T_K} \int_{\mathbb{R}} \frac{1}{\int_{\mathbb{R}^d} \mu^N d\mathbf{x}} \mu^N(\mathbf{x}, t) \left( \frac{\partial \lambda}{\partial t} + \frac{1}{2} \|\nabla_{\mathbf{x}} \lambda\|^2 + \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 \lambda + g - \alpha \psi(g) \right)^2 d\mathbf{x} dt \right] \\ &= \int_0^{T_K} \int_{\mathbb{R}} \frac{1}{\lim_{N \rightarrow \infty} \int_{\mathbb{R}^d} \mu^N d\mathbf{x}} \lim_{N \rightarrow \infty} \left( \mu^N \left( \frac{\partial \lambda}{\partial t} + \frac{1}{2} \|\nabla_{\mathbf{x}} \lambda\|^2 + \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 \lambda + g - \alpha \psi(g) \right)^2 \right) d\mathbf{x} dt \\ &= \int_0^{T_K} \int_{\mathbb{R}^d} \hat{\rho}(\mathbf{x}, t) \left( \frac{\partial \lambda(\mathbf{x}, t)}{\partial t} + \frac{1}{2} \|\nabla_{\mathbf{x}} \lambda\|^2 + \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 \lambda + g(\mathbf{x}, t) - \alpha \psi(g) \right)^2 d\mathbf{x} dt \end{aligned}$$

In the final equality of the proof, we employed the previously proven [Appendix A.1.3](#).  $\square$

#### A.1.5 Proposition : the Expectation of Action Loss

**Proposition A.2.** Consider the following action loss:

$$\mathcal{L}_{Action}^N = \frac{1}{N} \sum_{i=1}^N \left( \int_0^1 \frac{1}{2} \|\mathbf{u}(\mathbf{X}_i^t, t)\|^2 w_i(t) dt + \int_0^1 \alpha \psi(g(\mathbf{X}_i^t, t)) w_i(t) dt \right) \quad (15)$$

where

$$d\mathbf{X}_i^t = \mathbf{u}(\mathbf{X}_i^t, t) dt + \sigma(t) d\mathbf{W}_t, \quad dw_i = g(\mathbf{X}_i^t, t) w_i dt,$$

The expectation of action loss is equal to the action defined in the RUOT formulation, namely,

$$\mathbb{E}[\mathcal{L}_{Action}^N] = \int_0^{T_K} \int_{\mathbb{R}^d} \left( \frac{1}{2} \|\mathbf{u}(\mathbf{x}, t)\|^2 + \alpha \psi(g(\mathbf{x}, t)) \right) \rho(\mathbf{x}, t) d\mathbf{x} dt. \quad (16)$$

*Proof.* Taking the expectation of  $\mathcal{L}_{Action}^N$  is equivalent to drawing  $N$  particles each time to obtain  $\mathcal{L}_{Action}^N$ , repeating this process infinitely many times, and computing the average of the  $\mathcal{L}_{Action}^N$  obtained in each instance. Since the particles are independent, this operation is directly equivalent to taking the number of particles  $N \rightarrow \infty$ , thus:

$$\begin{aligned}
\mathbb{E}[\mathcal{L}_{\text{Action}}^N] &= \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \left[ \int_0^{T_K} \frac{1}{2} \|\mathbf{u}(\mathbf{X}_i^t, t)\|^2 w_i(t) dt + \int_0^{T_K} \alpha \psi(g(\mathbf{X}_i^t, t)) w_i(t) dt \right] \\
&= \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \left[ \int_0^{T_K} \int_{\mathbb{R}^d} \frac{1}{2} \|\mathbf{u}(\mathbf{x}, t)\|^2 \mu^N(\mathbf{x}) dt \right. \\
&\quad \left. + \int_0^{T_K} \int_{\mathbb{R}^d} \alpha \psi(g(\mathbf{x}, t)) \mu^N(\mathbf{x}) dt \right] \\
&= \int_0^{T_K} \int_{\mathbb{R}^d} \left( \frac{1}{2} \|\mathbf{u}(\mathbf{x}, t)\|^2 + \alpha \psi(g(\mathbf{x}, t)) \right) \rho(\mathbf{x}, t) d\mathbf{x} dt.
\end{aligned}$$

In the final equality of the proof, we employed the previously proven [Appendix A.1.3](#).  $\square$

## A.2 Optimal Conditions Under Different $\psi(g)$

In our Experiment, we use two different  $\psi(g)$  as examples. When  $\psi_1(g) = \frac{1}{2}g^2$ , the optimal conditions are :

$$\mathbf{u} = \nabla_{\mathbf{x}} \lambda, \quad g = \frac{\lambda}{\alpha}, \quad \frac{\partial \lambda}{\partial t} + \frac{1}{2} \|\nabla_{\mathbf{x}} \lambda\|^2 + \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 \lambda + \frac{1}{2} \frac{\lambda^2}{\alpha} = 0.$$

When  $\psi_2(g) = g^{2/15}$ , the optimal conditions are :

$$\mathbf{u} = \nabla_{\mathbf{x}} \lambda, \quad g = \left( \frac{2\alpha}{15\lambda} \right)^{\frac{15}{13}}, \quad \frac{\partial \lambda}{\partial t} + \frac{1}{2} \|\nabla_{\mathbf{x}} \lambda\|^2 + \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 \lambda - \frac{13}{15} \alpha \left( \frac{2\alpha}{15\lambda} \right)^{\frac{3}{13}} = 0.$$

Note that in this case the function  $g(\lambda)$  exhibits a singularity at  $\lambda = 0$ . In fact, given the two properties we imposed on  $\psi(g)$  ( $\frac{d\psi(g)}{d|g|} > 0$  and  $\psi(g) = \psi(-g)$ ) along with the constraint  $\psi''(g) < 0$ , it follows that  $\psi'(g)$  must be discontinuous at 0, and hence  $g(\lambda) = (\psi')^{-1}\left(\frac{\lambda(x,t)}{\alpha}\right)$  necessarily has a singularity at  $\lambda = 0$ . For the sake of training stability, we slightly modify  $g(\lambda)$  to remove this singularity. We redefine  $g(\lambda)$  as:

$$g^\dagger(\lambda) = \begin{cases} g(\lambda), & \lambda \geq \delta, \\ \frac{1}{2\delta} (g(\delta) - g(-\delta))(\lambda + \delta), & -\delta \leq \lambda < \delta, \\ g(\lambda), & \lambda < -\delta, \end{cases}$$

where  $\delta$  is a small positive constant. In our computations, we set  $\delta = 0.1$ .

## A.3 Training Algorithm

The Var-RUOT training algorithm is shown in [Algorithm 1](#).

## B Experiential Details

### B.1 Additional Information for Datasets

**Simulation Dataset** In the main text, we utilize a simulated dataset derived from a three-gene regulatory network ([Zhang et al., 2025a](#)). The dynamics of this system are governed by stochastic differential equations that incorporate self-activation, mutual inhibition, and external activation. The dynamics of the three genes are described by the following equations:

$$\begin{aligned}
\frac{dX_1^i}{dt} &= \frac{\alpha_1 (X_1^i)^2 + \beta}{1 + \gamma_1 (X_1^i)^2 + \alpha_2 (X_2^i)^2 + \gamma_3 (X_3^i)^2 + \beta} - \delta_1 X_1^i + \eta_1 \xi_t, \\
\frac{dX_2^i}{dt} &= \frac{\alpha_2 (X_2^i)^2 + \beta}{1 + \gamma_1 (X_1^i)^2 + \alpha_2 (X_2^i)^2 + \gamma_3 (X_3^i)^2 + \beta} - \delta_2 X_2^i + \eta_2 \xi_t, \\
\frac{dX_3^i}{dt} &= \frac{\alpha_3 (X_3^i)^2}{1 + \alpha_3 (X_3^i)^2} - \delta_3 X_3^i + \eta_3 \xi_t,
\end{aligned}$$

---

**Algorithm 1** Training Var-RUOT

---

**Require:** Datasets  $D_1, \dots, D_K$ , batch size  $N$ , training epochs  $N_{\text{Epoch}}$ , initialized network  $\lambda_\theta(\mathbf{x}, t)$ .

**Ensure:** Trained scalar field  $\lambda_\theta(\mathbf{x}, t)$ .

```

1: for  $i = 1$  to  $N_{\text{Epoch}}$  do
2:   From the data at the first time point,  $D_1$ , sample  $N$  particles and set all their weights to  $w_i(0) = 1$ , for  $i = \{1, 2, \dots, N\}$ .
3:   for  $j = 1$  to  $K$  do
4:     use optimal conditions  $\mathbf{u}_\theta(\mathbf{x}, t) = \nabla_{\mathbf{x}} \lambda_\theta(\mathbf{x}, t)$ ,  $\alpha \frac{d\psi(g_\theta)}{dg_\theta} = \lambda_\theta(\mathbf{x}, t)$  to calculate  $\mathbf{u}_\theta(\mathbf{x}, t)$  and  $g_\theta(\mathbf{x}, t)$ , where  $t \in [T_i, T_{i+1})$ 
       $\mathcal{L}_{\text{Action}}^N \leftarrow \mathcal{L}_{\text{Action}}^N +$ 
      
$$\frac{1}{N} \sum_{i=1}^N \left( \int_{T_i}^{T_{i+1}} \frac{1}{2} \|\mathbf{u}_\theta(\mathbf{X}_i^t, t)\|^2 w_i(t) dt + \int_{T_i}^{T_{i+1}} \alpha \psi(g_\theta(\mathbf{X}_i^t, t)) w_i(t) dt \right)$$

      
$$\mathcal{L}_{\text{HJB}}^N \leftarrow \mathcal{L}_{\text{HJB}}^N + \sum_{i=1}^N \left[ \int_{T_i}^{T_{i+1}} \frac{w_i(t)}{\sum_{j=1}^N w_j(t)} \right.$$

      
$$\left. \times \left( \frac{\partial \lambda_\theta(\mathbf{X}_i^t, t)}{\partial t} + \frac{1}{2} \|\nabla_{\mathbf{x}} \lambda_\theta\|^2 + \frac{1}{2} \sigma^2 \nabla_{\mathbf{x}}^2 \lambda_\theta + g_\theta(\mathbf{X}_i^t, t) - \alpha \psi(g_\theta) \right)^2 dt \right]$$

7:    $\mathcal{L}_{\text{Recon}} \leftarrow \mathcal{L}_{\text{Recon}} + (M(T_{i+1}) - \hat{M}(T_{i+1}))^2 + \mathcal{W}_2(\tilde{p}(\cdot, T_{i+1}), \hat{p}(\cdot, T_{i+1}))$ 
8:    $\mathcal{L}_{\text{Total}} = \gamma_{\text{Action}} \mathcal{L}_{\text{Action}} + \gamma_{\text{HJB}} \mathcal{L}_{\text{HJB}} + \mathcal{L}_{\text{Recon}}$ 
9:   update  $\lambda_\theta(\mathbf{x}, t)$  w.r.t.  $\mathcal{L}_{\text{Total}}$ 

```

---

where  $\mathbf{X}^i(t)$  represents the gene expression levels of the  $i$ th cell at time  $t$ . The coefficients  $\alpha_i$ ,  $\gamma_i$ , and  $\beta$  control the strengths of self-activation, inhibition, and the external stimulus, respectively. The parameters  $\delta_i$  indicate the rates of gene degradation, and the terms  $\eta_i \xi_t$  account for stochastic influences using additive white noise.

The probability of cell division is linked to the expression level of  $X_2$  and is given by

$$g = \alpha_g \frac{X_2^2}{1 + X_2^2}.$$

When a cell divides, the resulting daughter cells are created with each gene perturbed by an independent random noise term,  $\eta_d N(0, 1)$ , around the parent cell's gene expression profile  $(X_1(t), X_2(t), X_3(t))$ . Detailed hyper-parameters are provided in Table 5. The initial population of cells is independently sampled from two normal distributions,  $\mathcal{N}([2, 0.2, 0], 0.1)$  and  $\mathcal{N}([0, 0, 2], 0.1)$ . At every time step, any negative expression values are set to zero.

**Other Datasets Used in Main Text** In addition to the three-gene simulated dataset, our main text also utilizes the EMT dataset and the Mouse Blood Hematopoiesis dataset. The EMT dataset is sourced from (Sha et al., 2024; Cook & Vanderhyden, 2020) and is derived from A549 cancer cells undergoing TGFB1-induced epithelial-mesenchymal transition (EMT). It comprises data from four distinct time points, containing a total of 3133 cells, with each cell represented by 10 features obtained through PCA dimensionality reduction. Meanwhile, the Mouse Blood Hematopoiesis dataset covers 3 time points and includes 10,998 cells in total (Weinreb et al., 2020; Sha et al., 2024) and was reduced to 2-dimensional space using nonlinear dimensionality reduction .

**High Dimensional Gaussian Dataset** To validate the capability of our model to capture the dynamics of high-dimensional data, we used two high-dimensional gaussian datasets (a 50-dimensional set and a 100-dimensional set) from (Zhang et al., 2025a). The two-dimensional PCA visualizations of these datasets are shown in Fig. 5. The datasets were constructed as follows: for the initial distribution, 100 samples were drawn from a Gaussian distribution at location A and 400 samples were drawn from a Gaussian distribution at location B; for the terminal distribution, 200 samples were drawn from Gaussian distributions at locations C and D, and 1000 samples were drawn from a Gaussian distribution at location A.

**Other High Dimensional Datasets** In addition, we employed two real-world datasets. One is the Mouse Blood Hematopoiesis dataset from (Weinreb et al., 2020), which comprises data collected at three time points with a total of 49,302 cells, and we reduced its dimensionality to 50 using PCA, where the dataset used our main text is its subset. The other is the Pancreatic  $\beta$ -cell Differentiation dataset from (Veres et al., 2019), which consists of 51,274 cells sampled across eight time points, and we reduced it to 30 dimensions via PCA.

**Table 5.** Simulation parameters for the gene regulatory network.

| Parameter   | Value              | Description   |
|-------------|--------------------|---|
| $\alpha_1$  | 0.5                | Self-activation strength for $X_1$ .                    |
| $\gamma_1$  | 0.5                | Inhibition strength exerted by $X_3$ on $X_1$ .         |
| $\alpha_2$  | 1                  | Self-activation strength for $X_2$ .                    |
| $\gamma_2$  | 1                  | Inhibition strength exerted by $X_3$ on $X_2$ .         |
| $\alpha_3$  | 1                  | Self-activation strength for $X_3$ .                    |
| $\gamma_3$  | 10                 | Half-saturation constant in the inhibition term.        |
| $\delta_1$  | 0.4                | Degradation rate for $X_1$ .                            |
| $\delta_2$  | 0.4                | Degradation rate for $X_2$ .                            |
| $\delta_3$  | 0.4                | Degradation rate for $X_3$ .                            |
| $\eta_1$    | 0.05               | Noise intensity for $X_1$ .                             |
| $\eta_2$    | 0.05               | Noise intensity for $X_2$ .                             |
| $\eta_3$    | 0.01               | Noise intensity for $X_3$ .                             |
| $\eta_d$    | 0.014              | Noise intensity for perturbations during cell division. |
| $\beta$     | 1                  | External signal activating $X_1$ and $X_2$ .            |
| $dt$        | 1                  | Time step size.   |
| Time Points | [0, 8, 16, 24, 32] | Discrete time points when data is recorded.             |

## B.2 Evaluation Metrics

To assess the fitting accuracy of the learned dynamics to the data distribution, we compute the  $\mathcal{W}_1$  and  $\mathcal{W}_2$  distances between the data points generated by the model and the real data points. They are defined as

$$\mathcal{W}_1(p, q) = \min_{\pi \in \Pi(p, q)} \int \|\mathbf{x} - \mathbf{y}\|_2 d\pi(\mathbf{x}, \mathbf{y}),$$

and

$$\mathcal{W}_2(p, q) = \left( \min_{\pi \in \Pi(p, q)} \int \|\mathbf{x} - \mathbf{y}\|_2^2 d\pi(\mathbf{x}, \mathbf{y}) \right)^{1/2}.$$

We compute these two metrics using the `emd` function from the `pot` library.

To evaluate the action of the dynamics learned by the model, we directly compute the action loss. [Appendix A.1.5](#) guarantees that the expectation of the loss is equal to the action defined in the RUOT problem. The action loss is:

$$\mathcal{L}_{\text{Action}}^N = \frac{1}{N} \sum_{i=1}^N \left( \int_0^1 \frac{1}{2} \|\mathbf{u}(\mathbf{X}_i^t, t)\|^2 w_i(t) dt + \int_0^1 \alpha \psi(g(\mathbf{X}_i^t, t)) w_i(t) dt \right)$$

where

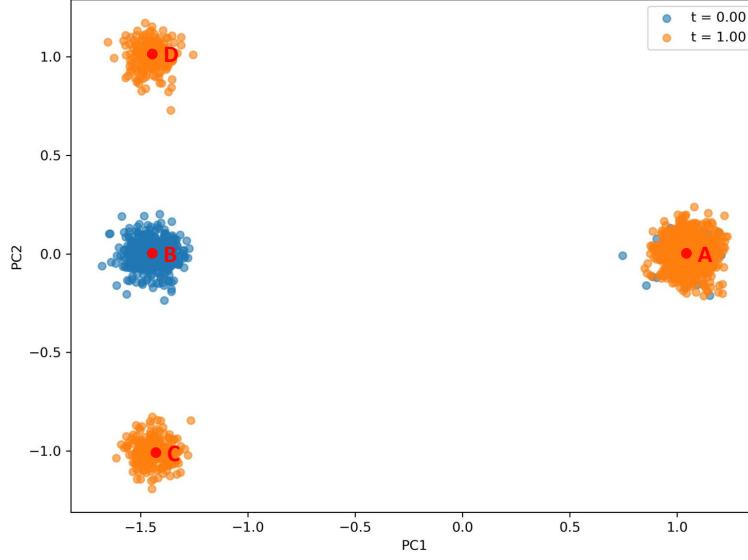
$$d\mathbf{X}_i^t = \mathbf{u}(\mathbf{X}_i^t, t) dt + \sigma(t) d\mathbf{W}_t,$$

$$dw_i = g(\mathbf{X}_i^t, t) w_i dt,$$

with initial conditions  $\mathbf{X}_i^0 \sim \rho(\mathbf{x}, 0)$  and  $w_i(0) = 1$ . We run our model 5 times on each dataset, to calculate the mean and standard deviation of  $\mathcal{W}_1, \mathcal{W}_2$  and action.

To evaluate the training speed of the model, we use the `SamplesLoss` class from the `geomloss` library to compute the OT loss at each epoch during the training process for each method, with the blur parameter set to 0.10. We sum the OT losses at all time points to obtain the total OT loss. For each model, we perform 5 training runs, recording the number of epochs and the time required for the OT loss to drop below a specified threshold. We then compute the mean and standard deviation of these values, with the mean reflecting the training/convergence speed and the standard deviation reflecting the training stability.

For models whose dynamics are governed by stochastic differential equations, the choice of  $\sigma$  directly affects the results (both the OT loss and the path action). Therefore, when running the RUOT and our Var-RUOT models on each dataset,  $\sigma$  is set to 0.10.



**Figure 5.** Diagram of the 50-dimensional and 100-dimensional Gaussian distribution data. PCA is used to reduce the data to 2 dimensions.

## C Additional Experiment Results

### C.1 Additional Results on Training Speed and Stability

We plotted the average loss per epoch across five training runs in Fig. 6. Experimental results show that on the Simulation Gene dataset, our algorithm converges approximately 10 times faster than the fastest among the other algorithms (RUOT with 3-epoch pretraining), and on the EMT dataset, our algorithm converges roughly 20 times faster than the fastest alternative (TIGON).

### C.2 Hyperparameter Selection and Ablation Study

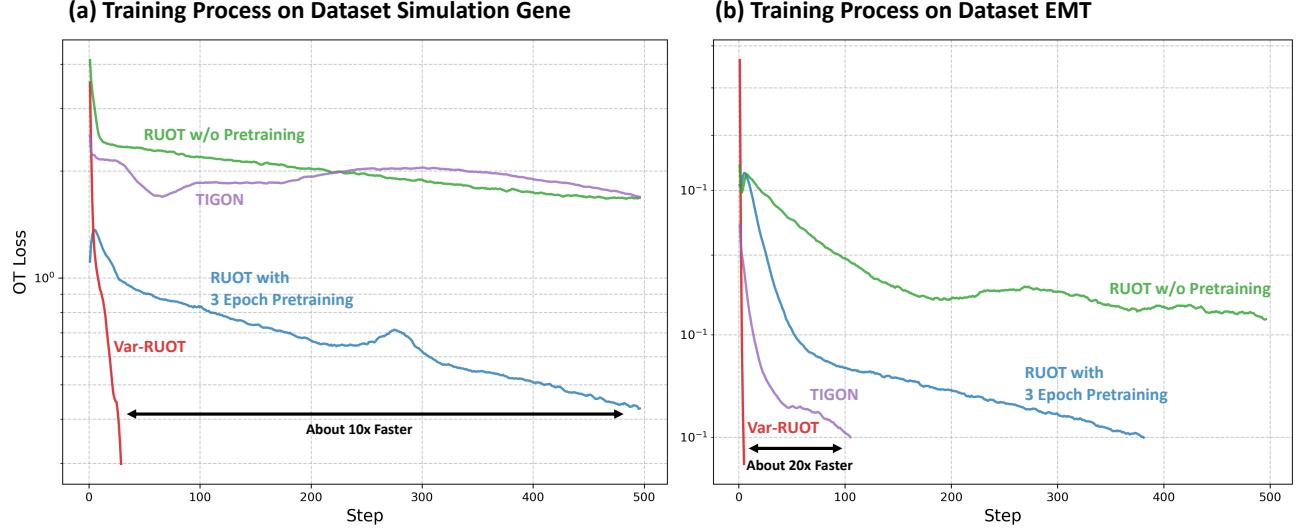
**Hyperparameter Selection** We used NVIDIA A100 GPUs (with 40G memory) and 128-core CPUs to conduct the experiments described in this paper. The neural network used to fit  $\lambda(\mathbf{x}, t)$  is a fully connected network augmented with layer normalization and residual connections. It consists of 2 hidden layers, each with 512 dimensions. In our algorithm, the main hyperparameters that need tuning include the penalty coefficient  $\alpha$  for growth in the action, and the weights  $\gamma_{\text{HJB}}$  and  $\gamma_{\text{Action}}$  for the two regularization losses,  $\mathcal{L}_{\text{HJB}}$  and  $\mathcal{L}_{\text{Action}}$ , respectively. Here,  $\alpha$  represents our prior regarding the strength of cell birth and death in the data; a larger  $\alpha$  imposes a greater penalty on cell birth and death, thereby making it easier for the model to learn solutions with lower birth and death intensities. Meanwhile, the HJB loss and the action loss, serving as regularizers, are both designed to ensure that the solution obtained by the algorithm has as low an action as possible—the HJB equation being a necessary condition for the action to reach its minimum, and the inclusion of the action loss ensuring that the model learns a solution with an even smaller action under those necessary conditions.

To ensure that our algorithm generalizes well across a wide range of real-world datasets, we only used two sets of parameters: one for the standard WFR Metric ( $\psi(g) = \frac{1}{2}g^2$ ) and one for the Modified Metric ( $\psi(g) = g^{2/15}$ ). The parameters used in each case are listed in Table 6. The primary reason for using two sets is that different metrics yield different scales for the HJB loss.

**Table 6.** Parameter settings for standard WFR metric and modified metric.

| Parameter  | $\gamma_{\text{HJB}}$ | $\gamma_{\text{Action}}$ | $\alpha$ | Learning Rate      | Optimizer |
|--|-----------------------|--------------------------|----------|--------------------|-----------|
| Standard WFR Metric ( $\psi_1(g) = \frac{1}{2}g^2$ ) | $6.25 \times 10^{-2}$ | $6.25 \times 10^{-2}$    | 2.00     | $1 \times 10^{-4}$ | AdamW     |
| Modified Metric ( $\psi_2(g) = g^{2/15}$ )           | $6.25 \times 10^{-3}$ | $6.25 \times 10^{-2}$    | 7.00     | $2 \times 10^{-5}$ | AdamW     |

**Sensitivity Analysis of  $\alpha$**  To demonstrate the robustness of our algorithm with respect to hyperparameter selection, we



**Figure 6.** Loss curves of each algorithm over five training runs. The curves are obtained by averaging the losses from the five runs.

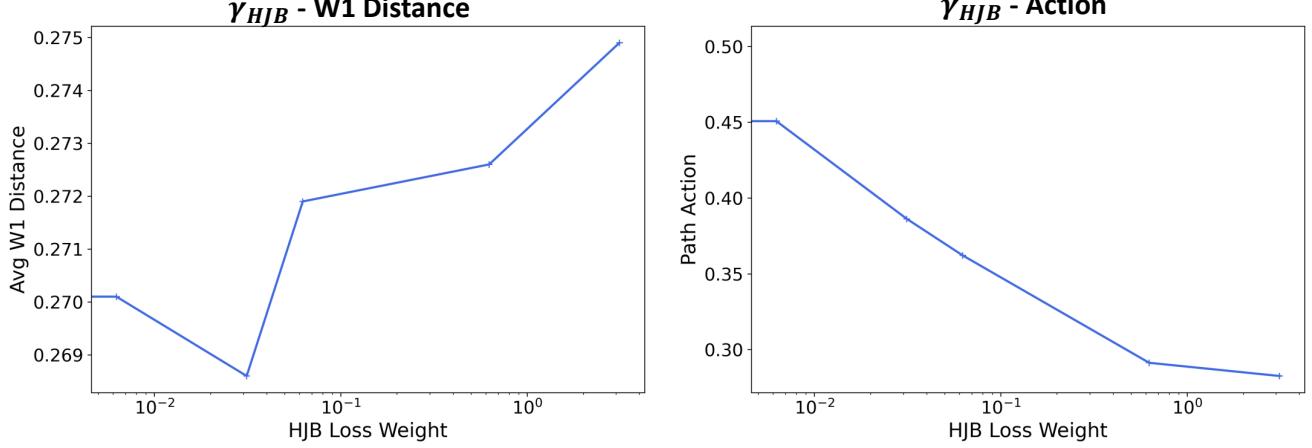
first varied the penalty coefficient  $\alpha$  for growth and examined the resulting changes in model performance. This sensitivity analysis was conducted on the 2D Mouse Blood Hematopoiesis dataset, and we performed the analysis for both the Standard WFR metric and the modified metric. The performance of the model under different values of  $\alpha$  is shown in Table 7. The experimental results indicate that our algorithm is not sensitive to  $\alpha$ , as similar performance can be achieved across multiple different values of  $\alpha$ . In comparison with the Standard WFR metric, however, the algorithm appears to be somewhat more sensitive to  $\alpha$  when the modified metric is used.

**Table 7.** On the Mouse Blood Hematopoiesis dataset, the Wasserstein distances ( $\mathcal{W}_1$  and  $\mathcal{W}_2$ ) between the predicted distributions of Var-RUOT with different  $\alpha$  and the true distribution at each time points. Each experiment was run five times to compute the mean and standard deviation.

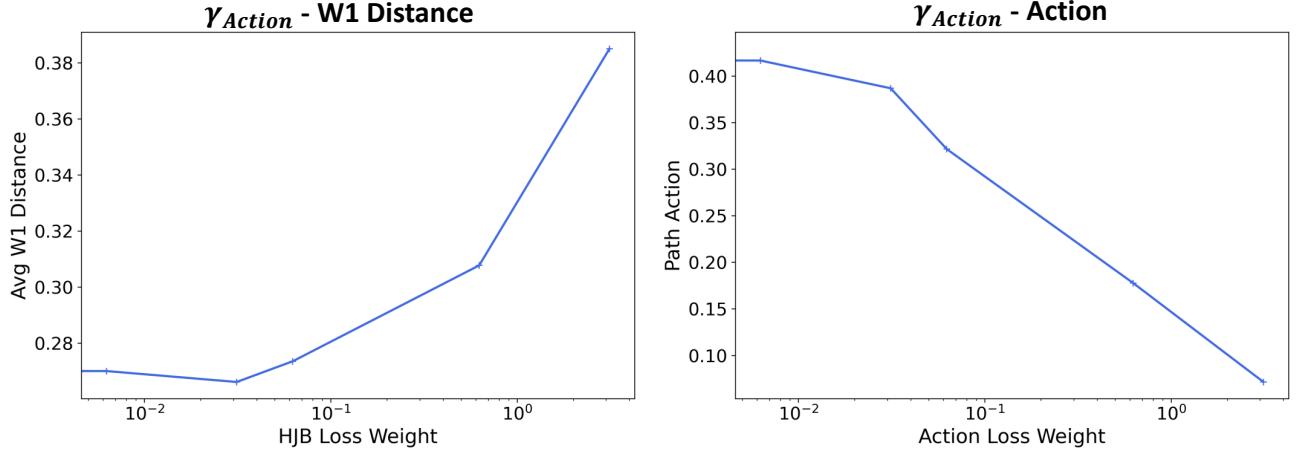
| Model                                     | $t = 1$             |                     | $t = 2$             |                     |
|---|---------------------|---------------------|---------------------|---------------------|
|   | $\mathcal{W}_1$     | $\mathcal{W}_2$     | $\mathcal{W}_1$     | $\mathcal{W}_2$     |
| Var-RUOT (Standard WFR, $\alpha = 1$ )    | $0.1622 \pm 0.0072$ | $0.2027 \pm 0.0097$ | $0.1280 \pm 0.0123$ | $0.1522 \pm 0.0178$ |
| Var-RUOT (Standard WFR, $\alpha = 2$ )    | $0.1203 \pm 0.0060$ | $0.1498 \pm 0.0043$ | $0.1389 \pm 0.0068$ | $0.1701 \pm 0.0096$ |
| Var-RUOT (Standard WFR, $\alpha = 3$ )    | $0.1402 \pm 0.0054$ | $0.1704 \pm 0.0077$ | $0.1350 \pm 0.0100$ | $0.1655 \pm 0.0132$ |
| Var-RUOT (Modified Metric, $\alpha = 5$ ) | $0.3783 \pm 0.0194$ | $0.3326 \pm 0.0128$ | $0.2110 \pm 0.0164$ | $0.2226 \pm 0.0219$ |
| Var-RUOT (Modified Metric, $\alpha = 7$ ) | $0.2953 \pm 0.0357$ | $0.3117 \pm 0.0323$ | $0.1917 \pm 0.0140$ | $0.2226 \pm 0.0170$ |
| Var-RUOT (Modified Metric, $\alpha = 9$ ) | $0.2737 \pm 0.0095$ | $0.3116 \pm 0.0072$ | $0.1970 \pm 0.0072$ | $0.2224 \pm 0.0075$ |

**Ablation Study of  $\gamma_{\text{HJB}}$  and  $\gamma_{\text{Action}}$**  In order to verify whether  $\mathcal{L}_{\text{HJB}}$  and  $\mathcal{L}_{\text{Action}}$  help the algorithm find solutions with lower action, we conducted ablation studies. These experiments were carried out on the EMT data, since in this dataset the transition from the initial distribution to the terminal distribution can be achieved through relatively simple dynamics (each particle moving in a straight line). Therefore, if the HJB loss and the action loss are effective, the model will learn this simple dynamics rather than a more complex one. We varied the HJB loss weight  $\gamma_{\text{HJB}}$  to the following values:  $[0, 6.25 \times 10^{-3}, 3.125 \times 10^{-2}, 6.25 \times 10^{-2}, 6.25 \times 10^{-1}, 3.125]$ , while keeping the action loss weight  $\gamma_{\text{Action}}$  fixed at 1. We then plotted both the mean  $\mathcal{W}_1$  distances between the predicted and true distributions at four different time points and the trajectory action (as shown in Fig. 7). Similarly, we fixed  $\gamma_{\text{HJB}} = 1$  and varied  $\lambda_{\text{Action}}$  over the same set of values:  $[0, 6.25 \times 10^{-3}, 3.125 \times 10^{-2}, 6.25 \times 10^{-2}, 6.25 \times 10^{-1}, 3.125]$ , with the corresponding results illustrated in Fig. 8. The figures indicate

that as  $\gamma_{HJB}$  and  $\gamma_{Action}$  increase, the action of the learned trajectories decreases monotonically, demonstrating that both loss terms are effective. However, as these weights increase, the model's ability to fit the distribution deteriorates. Therefore, we recommend that in practical applications, both  $\gamma_{HJB}$  and  $\gamma_{Action}$  should be set to values below 0.1, as configured in this paper.



**Figure 7.** Ablation study on  $\gamma_{HJB}$ . The changes in the average  $W_1$  distance and the trajectory action for different values of  $\gamma_{HJB}$  are shown.



**Figure 8.** Ablation study on  $\gamma_{Action}$ . The changes in the average  $W_1$  distance and the trajectory action for different values of  $\gamma_{Action}$  are shown.

### C.3 Hold one out Experiment

In order to validate whether our algorithm can learn the correct dynamical equations from a limited set of snapshot data, we conducted hold-one-out experiments on the three-gene simulated data, the EMT data, and the 2D Mouse Blood Hematopoiesis data. This experiment is designed to test the interpolation and extrapolation capabilities of the algorithm. For a dataset with  $n$  time points, we perform  $n$  experiments. In each experiment, one time point is removed from the  $n$  time points, and the model is trained using the remaining time points. Afterwards, we compute the  $W_1$  and  $W_2$  distances between the predicted distribution and the true distribution at the missing time point. When a time point from  $\{1, 2, \dots, n-1\}$  is removed, the model is performing interpolation; when the time point  $n$  is removed, the model is performing extrapolation. The results of these experiments are shown in Table 8, Table 9, and Table 10. The experimental results indicate that our model's interpolation performance is superior to that of TIGON and comparable to that of DeepRUOT; in the EMT data and the Mouse Blood Hematopoiesis data, our model's extrapolation performance is significantly better than that of the other algorithms.

From a physical viewpoint, the dynamical equations governing the biological processes of cells can be formulated in the form of a minimum action principle (in this work, ITI RUOT Problem is a surrogate model whose action is not the true action derived from studying the biological process, but rather a simple and numerically convenient form of action). Compared to other algorithms, our method can find trajectories with lower action, i.e., It is more capable of learning dynamics that conform to the prior prescribed by the action functional. These dynamics yield better extrapolation performance, which indicates that the design of the action in the RUOT problem is at least partially reasonable. From a machine learning perspective, forcing the model to learn trajectories corresponding to the minimum action serves as a form of regularization that enhances the model's generalization capability.

In addition, we separately illustrate the learned trajectories and growth profiles on the three-gene simulated dataset after removing four different time points, as shown in Fig. 9 and Fig. 10, respectively. The consistency in the learned results indirectly demonstrates that the model is still able to learn the correct dynamics and perform effective interpolation and extrapolation, even when snapshots at certain time points are missing. We further illustrate the interpolated and extrapolated trajectories of both the DeepRUOT and Var-RUOT algorithms on the Mouse Blood Hematopoiesis dataset, as shown in Fig. 11 and Fig. 12, respectively. This dataset comprises only three time points,  $t = 0, 1, 2$ . When one time point is removed, Var-RUOT tends to favor a straight-line trajectory connecting the remaining two time points (since such a trajectory represents the minimum-action path), which serves as an effective prior and leads to a reasonably accurate interpolation. In contrast, because DeepRUOT does not explicitly incorporate the minimum-action objective into its model, the trajectories it learns tend to be more intricate and curved. These more complex trajectories might present challenges for generalization, making accurate interpolation or extrapolation more difficult.

**Table 8.** On the three-gene simulated dataset, after removing the data of each time point in turn and training on the remaining data, the Wasserstein distances (i.e.,  $\mathcal{W}_1$  and  $\mathcal{W}_2$  distances) between the model-predicted data for the missing time points and the ground truth are computed.

| Model          | $t = 1$                               |                                       | $t = 2$                               |                                       | $t = 3$                               |                                       | $t = 4$                               |                                       |
|----------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|
|                | $\mathcal{W}_1$                       | $\mathcal{W}_2$                       | $\mathcal{W}_1$                       | $\mathcal{W}_2$                       | $\mathcal{W}_1$                       | $\mathcal{W}_2$                       | $\mathcal{W}_1$                       | $\mathcal{W}_2$                       |
| TIGON          | $0.1205 \pm 0.0000$                   | $0.1679 \pm 0.0000$                   | $0.0931 \pm 0.0000$                   | $0.1919 \pm 0.0000$                   | $0.2390 \pm 0.0000$                   | $0.3369 \pm 0.0000$                   | $0.2403 \pm 0.0000$                   | $0.3616 \pm 0.0000$                   |
| RUOT           | $0.0960 \pm 0.0027$                   | $0.1505 \pm 0.0018$                   | <b><math>0.0887 \pm 0.0069</math></b> | <b><math>0.1501 \pm 0.0062</math></b> | $0.1184 \pm 0.0058$                   | <b><math>0.1704 \pm 0.0079</math></b> | $0.1428 \pm 0.0062$                   | <b><math>0.2179 \pm 0.0135</math></b> |
| Var-RUOT(Ours) | <b><math>0.0880 \pm 0.0036</math></b> | <b><math>0.1210 \pm 0.0066</math></b> | $0.1043 \pm 0.0035$                   | $0.2293 \pm 0.0045$                   | <b><math>0.0943 \pm 0.0029</math></b> | $0.1769 \pm 0.0092$                   | <b><math>0.1401 \pm 0.0047</math></b> | $0.3382 \pm 0.0045$                   |

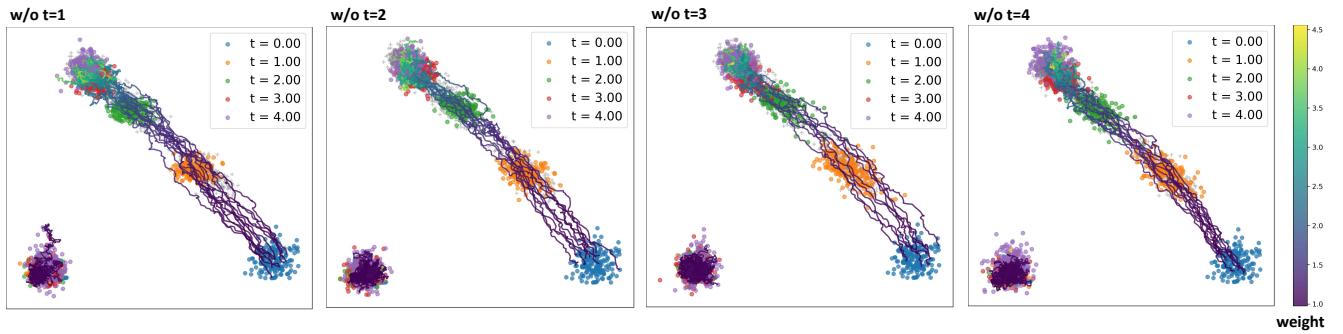
**Table 9.** On the EMT dataset, after removing the data of each time point in turn and training on the remaining data, the Wasserstein distances (i.e.,  $\mathcal{W}_1$  and  $\mathcal{W}_2$  distances) between the model-predicted data for the missing time points and the ground truth are computed.

| Model          | $t = 1$                               |                                       | $t = 2$                               |                                       | $t = 3$                               |                                       |
|----------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|
|                | $\mathcal{W}_1$                       | $\mathcal{W}_2$                       | $\mathcal{W}_1$                       | $\mathcal{W}_2$                       | $\mathcal{W}_1$                       | $\mathcal{W}_2$                       |
| TIGON          | $0.3457 \pm 0.0000$                   | $0.3560 \pm 0.0000$                   | $0.3733 \pm 0.0000$                   | $0.3849 \pm 0.0000$                   | $0.5260 \pm 0.0000$                   | $0.5424 \pm 0.0000$                   |
| RUOT           | $0.3107 \pm 0.0017$                   | $0.3201 \pm 0.0016$                   | <b><math>0.3344 \pm 0.0024</math></b> | <b><math>0.3445 \pm 0.0021</math></b> | $0.4947 \pm 0.0019$                   | $0.5074 \pm 0.0019$                   |
| Var-RUOT(Ours) | <b><math>0.3018 \pm 0.0030</math></b> | <b><math>0.3104 \pm 0.0031</math></b> | $0.3375 \pm 0.0027$                   | $0.3460 \pm 0.0028$                   | <b><math>0.4082 \pm 0.0027</math></b> | <b><math>0.4189 \pm 0.0027</math></b> |

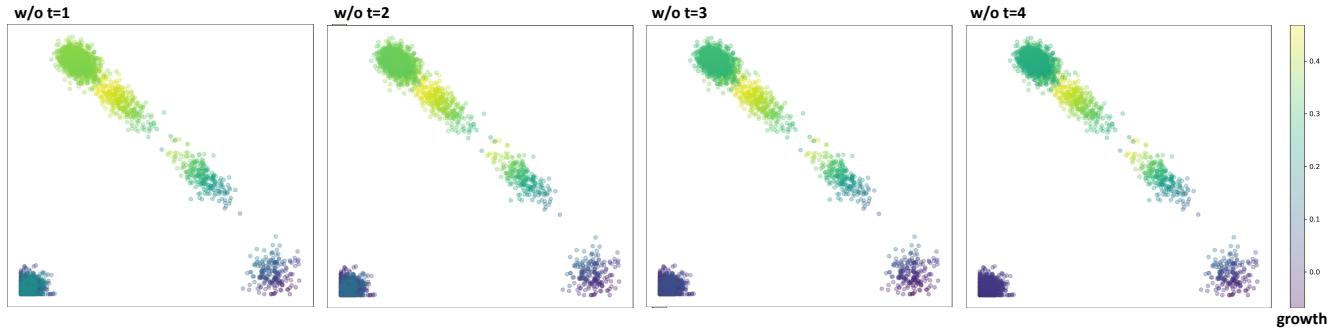
#### C.4 Experiments on High Dimensional Dataset

**High Dimensional Gaussian Dataset** To evaluate the effectiveness of our method on high-dimensional datasets, we first tested it on Gaussian datasets of 50 dimensions and 100 dimensions. We learned the dynamics of the data using the standard WFR metric ( $\psi(g) = \frac{1}{2}g^2$ ) as well as a modified growth penalty function,  $\psi(g) = g^{2/15}$ , which ensures  $\psi''(g) < 0$ . The learned trajectories and growth rates are illustrated in Fig. 13. Under both choices of  $\psi(g)$ , our method captures reasonable dynamics: the Gaussian distribution centered on the left shifts upward and downward, while the Gaussian distribution on the right exhibits growth without displacement.

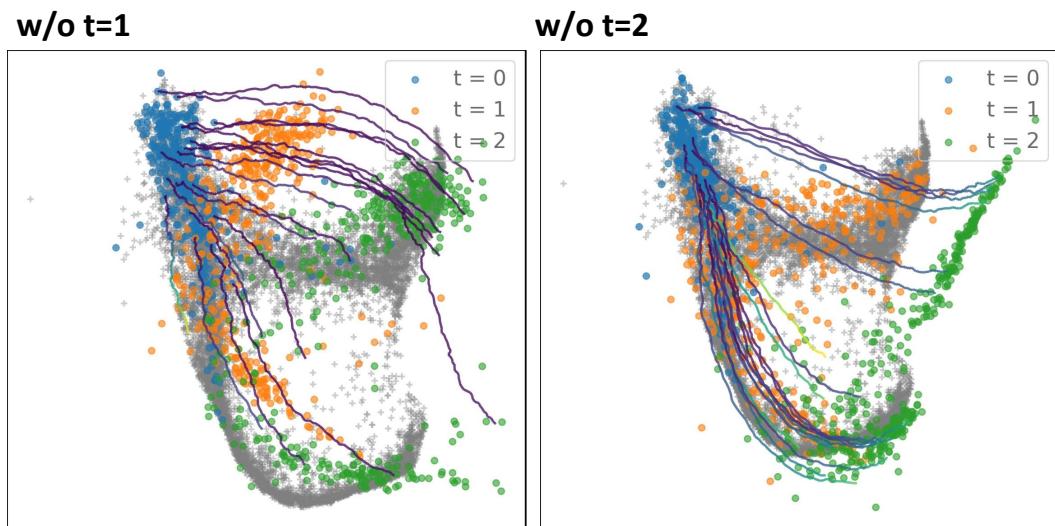
**50D Mouse Blood Hematopoiesis and Pancreatic  $\beta$ -cell Differentiation Dataset** We tested our method on two high-dimensional real scRNA-seq datasets including 50D Mouse Blood Hematopoiesis dataset and Pancreatic  $\beta$ -cell Differentiation dataset. We used UMAP to reduce the dimensionality of the datasets to 2 (only for visualization), plotted the growth of each data point, and visualized the vector fields  $\mathbf{u}(\mathbf{x}, t)$  on the reduced dimensions using the scvelo library. The results for the two



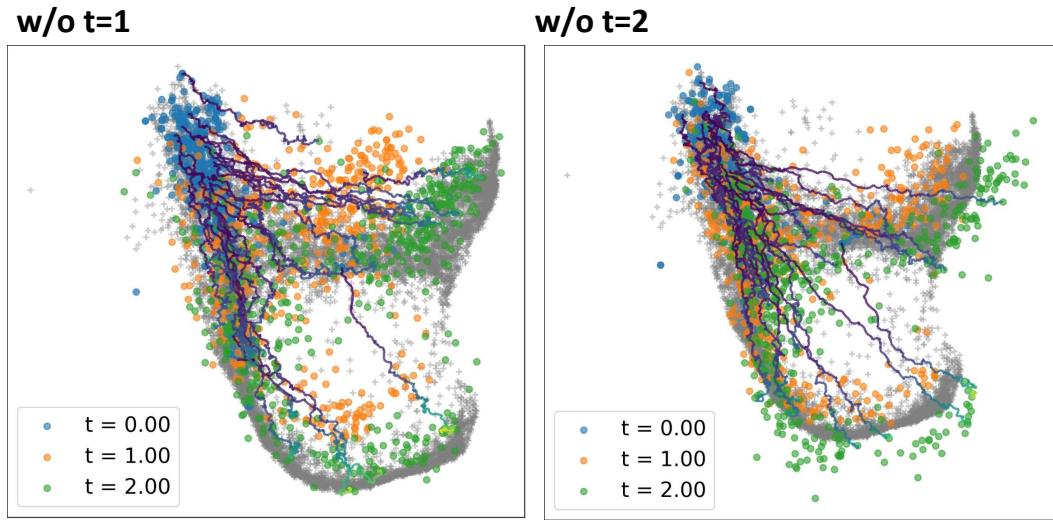
**Figure 9.** Trajectories learned on the three-gene simulated dataset after individually removing  $t = 1, 2, 3, 4$ .



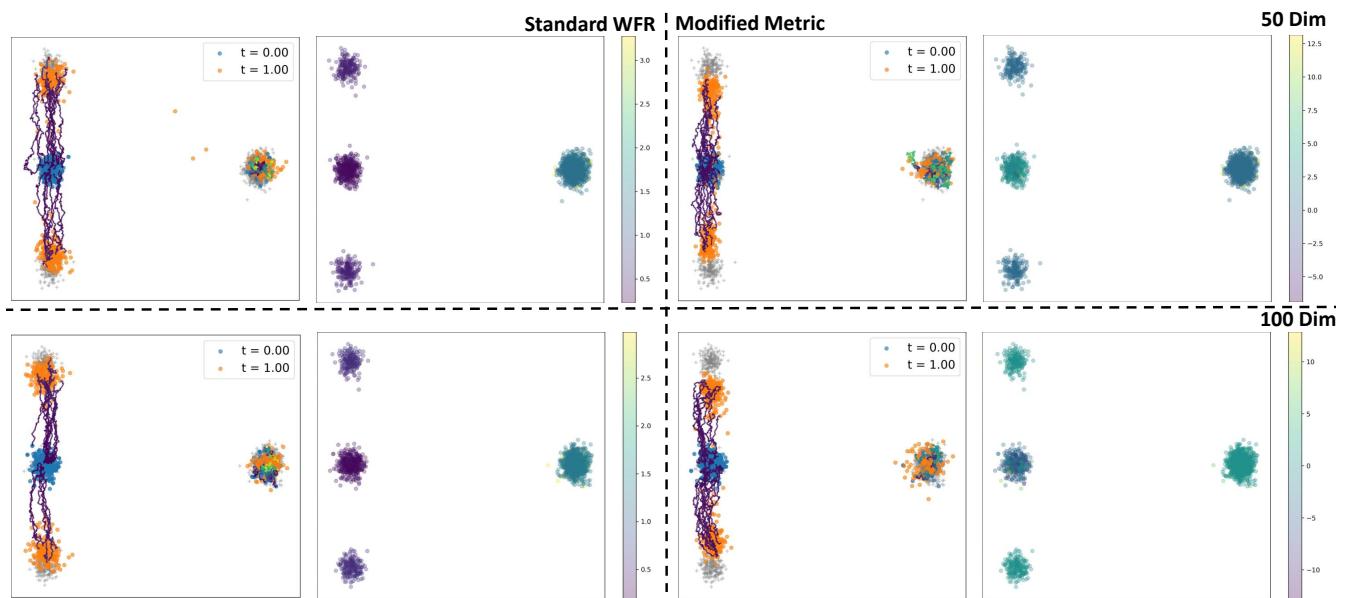
**Figure 10.** Growth learned on the three-gene simulated dataset after individually removing  $t = 1, 2, 3, 4$ .



**Figure 11.** The results of the DeepRUOT algorithm on the 2D Mouse Blood Hematopoiesis dataset for interpolation (with  $t=1$  removed) and extrapolation (with  $t=2$  removed)



**Figure 12.** The results of the Var-RUOT algorithm on the 2D Mouse Blood Hematopoiesis dataset for interpolation (with  $t=1$  removed) and extrapolation (with  $t=2$  removed)

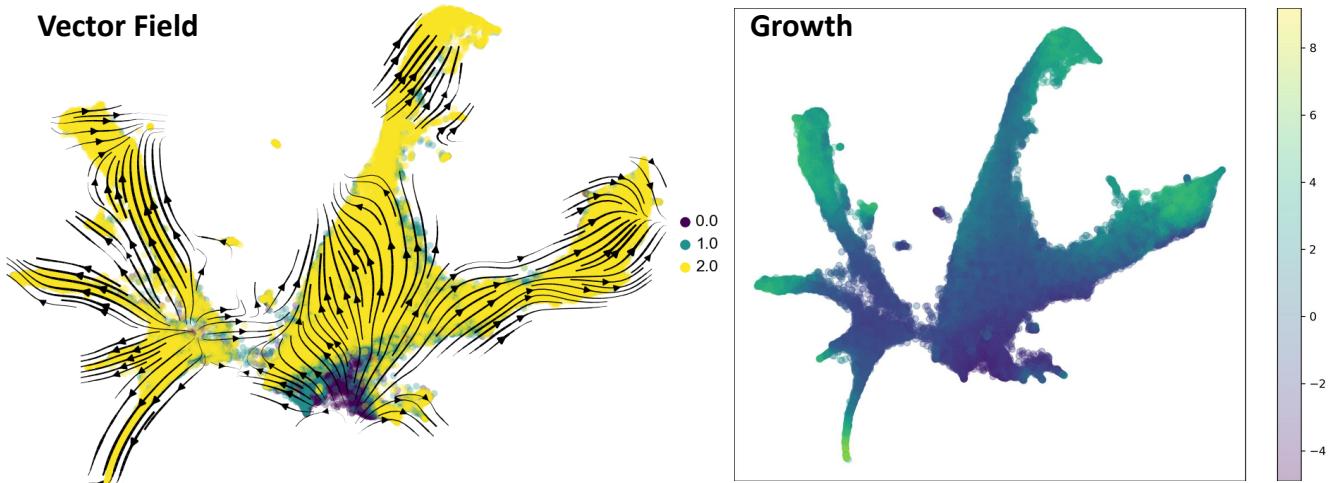


**Figure 13.** Trajectories and growth learned on the 50-dimensional and 100-dimensional Gaussian datasets using the Standard WFR Metric and Modified Metric.

**Table 10.** On the Mouse Blood Hematopoiesis dataset, after removing the data of each time point in turn and training on the remaining data, the Wasserstein distances (i.e.,  $\mathcal{W}_1$  and  $\mathcal{W}_2$  distances) between the model-predicted data for the missing time points and the ground truth are computed.

| Model          | $t = 1$                                 |   | $t = 2$                                 |   |
|----------------|---|---|---|---|
|                | $\mathcal{W}_1$                         | $\mathcal{W}_2$                         | $\mathcal{W}_1$                         | $\mathcal{W}_2$                         |
| TIGON          | $0.5838_{\pm 0.0000}$                   | $0.6726_{\pm 0.0000}$                   | $1.3264_{\pm 0.0000}$                   | $1.3928_{\pm 0.0000}$                   |
| RUOT           | $0.6235_{\pm 0.0014}$                   | $0.6971_{\pm 0.0012}$                   | $1.0723_{\pm 0.0096}$                   | $1.1397_{\pm 0.0120}$                   |
| Var-RUOT(Ours) | <b><math>0.2696_{\pm 0.0054}</math></b> | <b><math>0.3279_{\pm 0.0044}</math></b> | <b><math>0.2594_{\pm 0.0069}</math></b> | <b><math>0.3016_{\pm 0.0095}</math></b> |

datasets are shown in Fig. 14 and Fig. 15, respectively. As can be seen from the figures, the reduced velocity field points from points with smaller  $t$  to those with larger  $t$ , which indicates that our model can correctly learn a vector field that transfers the distribution even in high-dimensional data.



**Figure 14.** Learned vector field  $\mathbf{u}(\mathbf{x}, t)$  and growth on the 50D Mouse Blood Hematopoiesis dataset (data reduced using UMAP and vector field stream plots generated with the scvelo library).

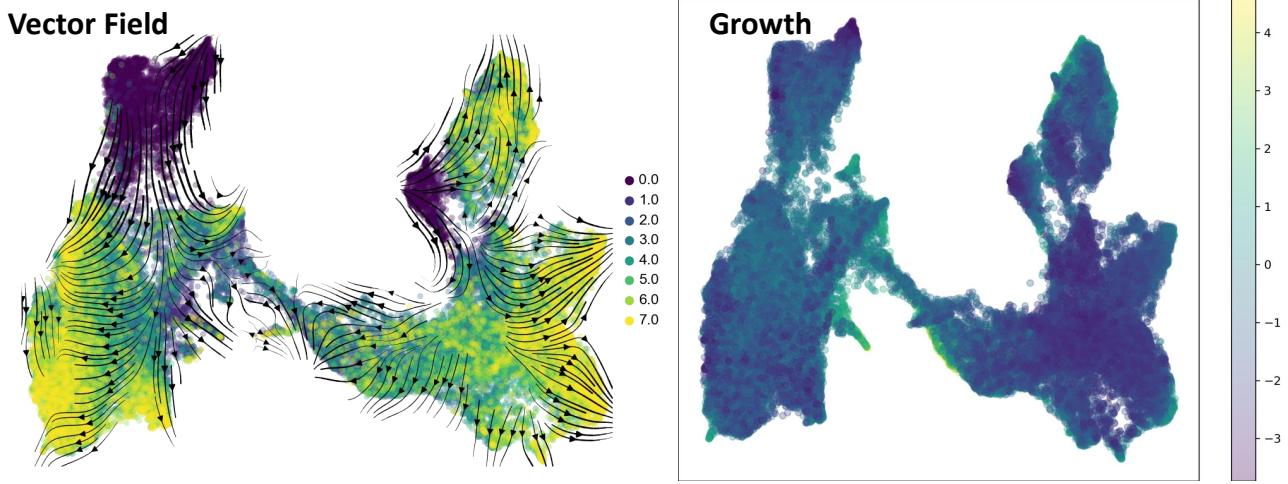
## D Limitations and Broader Impacts

### D.1 Limitations

The algorithm presented in this paper offers new insights for solving the RUOT problem; however, it still has several limitations. Firstly, although Var-RUOT parameterizes  $\mathbf{u}$  and  $g$  with a single neural network, and designs the loss function based on the necessary conditions for the minimal action solution, since neural network optimization only finds local minima, there is still no guarantee that the solution found is indeed the one with minimal action. This could be addressed by conducting a more detailed analysis on simpler versions of the RUOT problem (for instance, transferring Gaussian distributions to Gaussian distributions).

Furthermore, when using the modified metric, the goodness-of-fit in the distribution deteriorates, which may suggest that the  $\mathbf{u}$  and  $g$  satisfying the optimal necessary conditions derived via the variational method are limited in transporting the initial distribution to the terminal distribution. This might reflect a controllability issue in control theory that warrants further investigation.

Finally, the choice of  $\psi(g)$  in the action is dependent on biological priors. To automate it, one could approximate  $\psi$  with a neural network or derive it from microscopic or mesoscopic dynamics, such as the branching Wiener process to model cell division for a more physically grounded action.



**Figure 15.** Learned vector field  $\mathbf{u}(\mathbf{x}, t)$  and growth on the Pancreatic  $\beta$ -cell Differentiation dataset (data reduced using UMAP and vector field stream plots generated with the scvelo library).

## D.2 Broader Impacts

Var-RUOT explicitly incorporates the first-order optimality conditions of the RUOT problem into both the parameterization process and the loss function. This approach enables our algorithm to find solutions with a smaller action while maintaining excellent distribution fitting accuracy. Compared to previous methods, Var-RUOT employs only a single network to approximate a scalar field, which results in a faster and more stable training process. Additionally, we observe that the selection of the growth penalty function  $\psi(g)$  within the WFR metric is highly correlated with the underlying biological priors. Consequently, our new algorithm provides a novel perspective on the RUOT problem.

Our approach can be extended to other analogous systems. For example, in the case of simple mesoscopic particle systems—where the action can be explicitly formulated, such as in diffusion or chemical reaction processes—our framework can effectively infer the evolution of particle trajectories and distributions. This capability makes it applicable to tasks such as experimental data processing and interpolation. In the biological or medical field, our method can be employed to predict cellular developmental fate and to provide quantitative diagnostic results or treatment plans for certain diseases.

It should be noted that the performance of Var-RUOT largely depends on the quality of the data. Datasets containing significant noise may lead the model to produce results with a slight bias. Moreover, the particular form of the action can have a substantial impact on the model’s outcomes, potentially affecting important biological priors. These factors could present challenges for subsequent biological analyses or clinical decision-making, and care must be taken in the use and dissemination of the model-generated interpolation results to avoid data contamination.

When applying our method in biological or medical contexts, it is crucial to train the model using high-quality experimental data, select an action formulation that is well-aligned with the relevant domain-specific priors, and ensure that the results are validated by domain experts. Furthermore, there is a need to enhance the interpretability of the model and to further improve training speed through methods such as simulation-free techniques. These directions represent important avenues for our future work.