TracingFlow: A Simulation-Free Trajectory Inference Framework Based on Second-Order Dynamics

Yuhao Sun[1]*, Zekun Wu[2]*, Zixun Huang[2] , Peijie Zhou[1,3,4,5] †

[1]Center for Machine Learning Research, Peking University
[2]School of Mathematical Sciences, Peking University
[3]Center for Quantitative Biology, Peking University
[4]National Engineering Laboratory for Big Data Analysis and Applications, Beijing
[5]AI for Science Institute, Beijing

Abstract

Inferring continuous system evolution from sparse temporal snapshots is a key challenge in generative modeling and single-cell omics. While Optimal Transport (OT) is popular, existing frameworks are largely restricted to first-order dynamics, assuming memoryless velocity fields. This limits expressiveness, as first-order systems fail to account for regulatory momentum and time-delayed responses inherent in processes like cell differentiation. Here, we introduce TracingFlow, a simulation-free Flow Matching framework generalizing to second-order dynamics. By using neural networks to regress the acceleration field, TracingFlow provides an exact, efficient solution to the Dynamical Optimal Acceleration Transport (DOAT) problem. Unlike first-order methods yielding over-smoothed trajectories, our second-order formulation captures high-curvature transitions and nonlinear evolutions by learning the underlying force fields. Evaluated on complex synthetic and large-scale scRNA-seq datasets, TracingFlow achieves superior accuracy in distributional reconstruction and trajectory faithfulness. Moreover, by integrating lineage tracing priors, it recovers dynamical structures that are both mathematically optimal and biologically plausible.

1 Introduction

Recovering underlying dynamics from discrete observations is a pivotal task in single-cell omics, known as Trajectory Inference (TI) [9, 74]. To identify the least costly evolution between distributions, Optimal Transport (OT) frameworks have emerged [7, 73]. These generally categorize into Static OT methods [51, 28, 22], which learn direct mappings, and Dynamical OT methods [58, 24, 69, 44], which capture continuous evolution via flow maps.

Early Dynamical OT frameworks relied on Neural ODEs [9], incurring high computational costs due to numerical integration. To address this, Flow Matching [35, 36, 48] was proposed as a simulation-free alternative, regressing velocity fields to push source distributions to targets. By decomposing transport costs into single-particle trajectories, Flow Matching efficiently solves Dynamical OT problems [60, 27, 50], including its various variants [61, 15, 8, 45] .

However, most TI frameworks assume first-order dynamics ($\dot{\boldsymbol{x}} = \boldsymbol{v}(\boldsymbol{x}, t)$), inherently restricting $\boldsymbol{v}$ to be single-valued w.r.t $\boldsymbol{x}$. This limits expressiveness when modeling complex biological priors, such as lineage tracing [30, 38], potentially yielding dynamics that contradict with additional biological

*Equal contribution
†Corresponding author: pjzhou@pku.edu.cn

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

prior information. Furthermore, [19, 34, 23] pointed out that if more modalities such as chromatin accessibility or proteomics are considered, the biological kinetics may follow higher-order dynamics or could be modelled in augmented space. While 3MSBM [57] recently introduced a second-order Momentum Schrödinger Bridge for smoother trajectories, it relies on iterative retraining the acceleration field similar to Rectified Flow [36] to determine optimal couplings. Furthermore, it employs a heuristic method to estimate the initial velocity, rather than incorporating this estimation as part of the optimal control problem.

To overcome these limitations, we introduce **TracingFlow**, a simulation-free framework utilizing second-order dynamical systems. We propose the Dynamical Optimal Acceleration Transport (DOAT) problem, which determines evolution by minimizing acceleration costs. TracingFlow directly regresses the acceleration field and initial velocity, avoiding ODE simulation. Our experiments show that TracingFlow outperforms existing simulation-free frameworks [60, 61, 71, 42] in reconstruction accuracy and effectively incorporates biological priors. Our contributions include:

- We propose TracingFlow, a simulation-free TI framework based on second-order dynamics. It solves the proposed DOAT problem by regressing acceleration fields, reducing computational costs compared to ODE-based methods.
- We provide theoretical guarantees by decoupling the design of the optimal transport plan and single-particle control. We further introduce an iterative strategy to transform the real-world Velocity Missed-DOAT (VM-DOAT) problem into a standard DOAT problem, enabling approximate solutions.
- We demonstrate TracingFlow's effectiveness on multi-time-point real world datasets. Results indicate superior distribution reconstruction accuracy and enhanced capability in preserving biological priors compared to state-of-the-art methods.

2 Related Works

Solving Optimal Transport via Flow Matching. Flow Matching [35, 36, 48, 1] transports probability distributions via learned flow maps, offering scalable simulation-free training. As an optimal control problem, Optimal Transport (OT) [47] decomposes into Optimal Coupling and single-particle geodesics, facilitating solutions via Flow Matching [60, 27, 50, 15]. This approach extends to various variants and generalizations [61, 15, 8, 13, 63, 45, 26, 70, 4, 46]. However, these typically employ first-order dynamics. While second-order approaches have emerged [57], they require iterative training of the acceleration field to achieve optimal coupling. Moreover, they rely on heuristic methods for initial velocity estimation, rather than integrating it as a component of the optimal control problem. TracingFlow addresses this by decomposing the Dynamical Acceleration Optimal Transport (DOAT) problem into optimal coupling and single particle optimal trajectory, achieving a truly simulation-free process for DOAT.

Overcoming Single-Valued Velocity Constraint in Flow Matching. In standard Flow Matching, the velocity field is modeled as a function solely dependent on $\boldsymbol{x}$ and t. This single-valued dependency prevents the model from representing intersecting trajectories with distinct velocities, often resulting in curved inference paths and increased computational costs. To decouple the velocity from the current position, the velocity network can incorporate additional inputs like class labels [75, 55], initial positions [14], or hidden states [20]. Additionally, [71] employs hierarchical generation to mitigate this issue. However, these methods do not extend to second-order dynamics. While [42] utilized a second-order system, it is limited to constant-acceleration paths. In contrast, TracingFlow offers a flexible second-order setting. By designing minimal-cost paths, it effectively resolves the limitation of the single-valued velocity field.

Single-cell Trajectory Inference and Lineage Integration. Optimal Transport (OT) is a robust framework for inferring cellular dynamics from scRNA-seq data [51, 28, 58, 24, 40, 41, 2, 76, 37, 68, 12, 62, 33, 54, 29, 6, 10, 25, 72, 56, 44, 67, 52, 11]. However, transcriptomic similarity alone often fails to resolve complex trajectories. To mitigate this, studies incorporate lineage tracing priors such as clonal barcode information. Existing approaches integrate such info via regularization [17, 49], structural alignment [32], velocity mapping [64], or sparse transition modeling [65, 21, 18]. While improving accuracy, these generally focus on discrete couplings or static matrices. TracingFlow

advances this by embedding priors into a continuous OT-based second-order flow matching framework, capturing complex dynamics consistent with lineage information.

3 Mathematical Background

In this section, we provide the mathematical formulation of the problem addressed by TracingFlow.

Dynamical Optimal Acceleration Transport (DOAT) Problem Let $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{V} \subset \mathbb{R}^d$ denote the *position* and *velocity* spaces, respectively. We define the *augmented space* as $\mathcal{S} \coloneqq \mathcal{X} \times \mathcal{V}$, so that the full system state is $\boldsymbol{s} = (\boldsymbol{x}, \boldsymbol{v}) \in \mathcal{S}$. Consider the second-order dynamic system:

$$\dot{\boldsymbol{x}} = \boldsymbol{v} \quad \dot{\boldsymbol{v}} = \boldsymbol{a}(\boldsymbol{x}, \boldsymbol{v}, t), \quad t \in [t_0, t_K]. \tag{1}$$

Modeling a large ensemble of such particles via a probability density ρ_t in $\mathcal{S}$, we follow [5, 11] to formulate the following optimal control problem:

$$\min_{\boldsymbol{a},\rho} \mathcal{J}_{\text{DOAT}}(\boldsymbol{a}, \rho) = \int_{t_0}^{t_K} \int_{\mathcal{S}} \frac{1}{2} \|\boldsymbol{a}(\boldsymbol{x}, \boldsymbol{v}, t)\|^2 \rho_t(\boldsymbol{x}, \boldsymbol{v}) \, \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v}\mathrm{d}t \tag{2}$$

$$\text{s.t.} \quad \partial_t \rho_t + \nabla_{\boldsymbol{x}} \cdot (\boldsymbol{v}\rho_t) + \nabla_{\boldsymbol{v}} \cdot (\boldsymbol{a}\rho_t) = 0, \tag{3}$$

$$\rho_{t_i}(\boldsymbol{x}, \boldsymbol{v}) = \mu_i(\boldsymbol{x}, \boldsymbol{v}), \quad i = 0, \ldots, K. \tag{4}$$

Here, constraints are imposed at $K+1$ timestamps t_i with given densities μ_i. Equation (3) represents the continuity equation in the augmented space. Analogous to standard Dynamical Optimal Transport [47], we term this the *Dynamical Optimal Acceleration Transport (DOAT)* problem. To facilitate the subsequent numerical solution via particle methods, we assume the existence of an optimal flow map.

Velocity-Missed Dynamical Optimal Acceleration Transport (VM-DOAT) Problem However, in practical applications such as single-cell datasets, only particle positions are observable, rendering velocities as unknown quantities. This presents a discrepancy with the DOAT formulation.

Consequently, TracingFlow addresses a relaxation of the problem:

$$\min_{\boldsymbol{a},\rho} \mathcal{J}_{\text{VM-DOAT}}(\boldsymbol{a}, \rho) = \int_{t_0}^{t_K} \int_{\mathcal{S}} \frac{1}{2} \|\boldsymbol{a}(\boldsymbol{x}, \boldsymbol{v}, t)\|^2 \rho_t(\boldsymbol{x}, \boldsymbol{v}) \, \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v}\mathrm{d}t \tag{5}$$

$$\text{s.t.} \quad \partial_t \rho_t(\boldsymbol{x}, \boldsymbol{v}) + \nabla_{\boldsymbol{x}} \cdot (\boldsymbol{v}\rho_t) + \nabla_{\boldsymbol{v}} \cdot (\boldsymbol{a}\rho_t) = 0, \tag{6}$$

$$\int \rho_{t_i}(\boldsymbol{x}, \boldsymbol{v}) \, \mathrm{d}\boldsymbol{v} = \mu_i^{(\text{pos})}(\boldsymbol{x}), \; i = 0, 1, \ldots, K. \tag{7}$$

Here, the constraints are imposed only on the marginal distribution of positions, $\int \rho_{t_j}(\boldsymbol{x}, \boldsymbol{v}) \, \mathrm{d}\boldsymbol{v}$, at each time point, with no constraints on the velocity distribution. We refer to this problem as *Velocity Missed-Dynamical Optimal Acceleration Transport (VM-DOAT)*.

4 Methodology of TracingFlow

In this section, we first introduce the key properties of the DOAT problem that are essential for our algorithm design. Subsequently, drawing inspiration from Flow Matching approaches for standard Dynamical OT, we devise an algorithm to exactly solve the DOAT problem by combining Optimal Coupling with single-particle optimal trajectories. Finally, we introduce a global path-based velocity completion scheme that effectively bridges the gap between VM-DOAT and the standard DOAT framework. By inferring missing velocities through the lens of global trajectory consistency, our approach enables the application of optimal transport theory to solve the VM-DOAT problem in a principled manner.

4.1 Key Properties of DOAT Problem

Optimal Single Particle Trajectory The DOAT problem is an optimal control problem over distributions. We begin by considering the corresponding single-particle optimal control problem. Specifically, the optimal trajectory is a *cubic interpolation* determined by the boundary states.

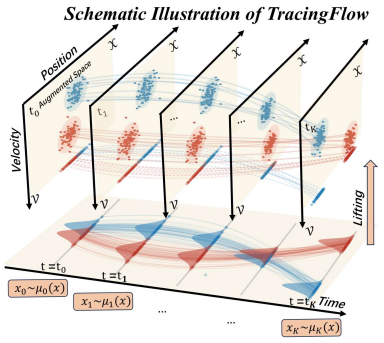


Figure 1: Visual overview of TracingFlow

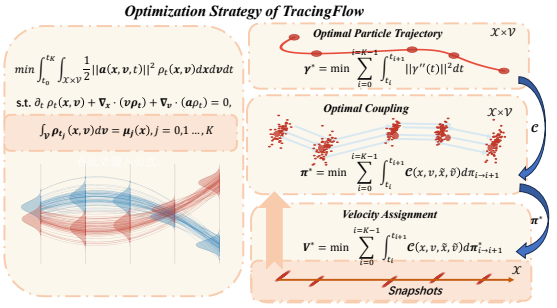


Figure 2: Illustration figure of VM-DOAT problem

Proposition 4.1. *Let $\gamma : [0,T] \to \mathbb{R}^d$ be a twice continuously differentiable curve satisfying the boundary conditions $\gamma(0) = \boldsymbol{x}_0$, $\gamma'(0) = \boldsymbol{v}_0$, $\gamma(T) = \boldsymbol{x}_T$, and $\gamma'(T) = \boldsymbol{v}_T$. The variational problem*

$$\min_{\gamma} \int_0^T \frac{1}{2} \|\gamma''(t)\|_2^2 \, \mathrm{d}t \tag{8}$$

admits a unique minimizer, which is a cubic polynomial of the form $\gamma(t) = c_3 t^3 + c_2 t^2 + c_1 t + c_0$, where the coefficients and the detailed proof are detailed in Section A.1.

We can immediately obtain the minimum cost for transporting a particle from $(\boldsymbol{x}_0, \boldsymbol{v}_0)$ to $(\boldsymbol{x}_T, \boldsymbol{v}_T)$:

Corollary 4.2. *The minimal cost of Equation* (8) *is*

$$\mathcal{C}_{[0,T]} = \frac{2}{T^3} \left\{ (\|\boldsymbol{v}_0\|^2 + \langle \boldsymbol{v}_0, \boldsymbol{v}_T \rangle + \|\boldsymbol{v}_T\|^2) T^2 + 3\langle \boldsymbol{v}_0 + \boldsymbol{v}_T, \boldsymbol{x}_0 - \boldsymbol{x}_T \rangle T + 3\|\boldsymbol{x}_0 - \boldsymbol{x}_T\|^2 \right\} \tag{9}$$

Optimal Coupling Using the minimal cost above, we can define an optimal transport problem between $t = 0$ and $t = T$:

$$\pi^*_{[0,T]} = \min_{\pi_{[0,T]}} \int \mathcal{C}_{[0,T]}[(\boldsymbol{x}_0, \boldsymbol{v}_0), (\boldsymbol{x}_T, \boldsymbol{v}_T)] \mathrm{d}\pi_{[0,T]} \tag{10}$$

where $\pi_{[0,T]}[(\boldsymbol{x}_0, \boldsymbol{v}_0), (\boldsymbol{x}_T, \boldsymbol{v}_T)]$ is any coupling satisfying the constraints $\int \pi_{[0,T]} \mathrm{d}\boldsymbol{x}_0 \mathrm{d}\boldsymbol{v}_0 = \mu_T(\boldsymbol{x}, \boldsymbol{v})$ and $\int \pi_{[0,T]} \mathrm{d}\boldsymbol{x}_T \mathrm{d}\boldsymbol{v}_T = \mu_0(\boldsymbol{x}, \boldsymbol{v})$, $\mu_0(\boldsymbol{x}, \boldsymbol{v})$ and $\mu_T(\boldsymbol{x}, \boldsymbol{v})$ are two distributions in augmented space. We term this formulation the **Static Optimal Acceleration Transport (SOAT)** problem.

While DOAT permits crossings in the projected position space, it prohibits trajectory crossing in the augmented space, which guarantees the neatness of trajectories. The following remark provides a mathematical justification, demonstrating that any such intersection implies a strictly higher transport cost, with a more detailed illustration in Section A.2.

Remark 4.3. Consider two pairs of points in $\mathbb{R}^{2d}$, $(\boldsymbol{x}_0, \boldsymbol{v}_0), (\boldsymbol{x}_T, \boldsymbol{v}_T)$ and $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{v}}_0), (\tilde{\boldsymbol{x}}_T, \tilde{\boldsymbol{v}}_T)$ with no points overlapping. Denote the cubic interpolation curves between each pair by $\gamma(t), \tilde{\gamma}(t) (t \in [0,T])$, respectively. If the curves cross with each other at time point $t_0 \in (0,T)$, then we have

$$\mathcal{C}_{[0,T]}(\boldsymbol{x}_0, \boldsymbol{v}_0, \boldsymbol{x}_T, \boldsymbol{v}_T) + \mathcal{C}_{[0,T]}(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{v}}_0, \tilde{\boldsymbol{x}}_T, \tilde{\boldsymbol{v}}_T) > \mathcal{C}_{[0,T]}(\boldsymbol{x}_0, \boldsymbol{v}_0, \tilde{\boldsymbol{x}}_T, \tilde{\boldsymbol{v}}_T) + \mathcal{C}_{[0,T]}(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{v}}_0, \boldsymbol{x}_T, \boldsymbol{v}_T).$$

Equation (2) represents a multi-marginal optimal transport problem over the interval $[t_0, t_K]$. To address this, we formulate sequential SOAT problems between adjacent time steps t_i and t_{i+1} by applying the substitutions $\boldsymbol{x}_0 \to \boldsymbol{x}_i$, $\boldsymbol{x}_T \to \boldsymbol{x}_{i+1}$, $\boldsymbol{v}_0 \to \boldsymbol{v}_i$, $\boldsymbol{v}_T \to \boldsymbol{v}_{i+1}$, and $T \to t_{i+1} - t_i$ to Equation (10). For simplicity, let $\mathcal{C}_{i \to i+1}$ denote the resulting pairwise cost and $\pi^*_{i \to i+1}$ the optimal plan. Due to the Markovian nature of the dynamics, the full multi-marginal plan decomposes into the sequence of these local plans $\{\pi^*_{i \to i+1}\}$ for $i = 0, \ldots, K-1$.

4.2 Conditional and Marginal Probability Path

Directly constructing the marginal probability path for DOAT is difficult. We can introduce a conditional variable $\boldsymbol{z}$ and mix over conditional probability paths, which is the standard trick in flow matching[35, 60]. Here in our setting, the condition variable $\boldsymbol{z} = [(\boldsymbol{x}_0, \boldsymbol{v}_0), \cdots, (\boldsymbol{x}_K, \boldsymbol{v}_K)]$ consists of $K+1$ points selected respectively from the data snapshots at $t = 0, 1, \cdots, K$. The marginal probability density is

$$\rho_t(\boldsymbol{x}, \boldsymbol{v}) = \int \rho_t(\boldsymbol{x}, \boldsymbol{v}|\boldsymbol{z}) q(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} \tag{11}$$

Where $\rho_t(\boldsymbol{x}, \boldsymbol{v}|\boldsymbol{z})$ is guided by a conditional acceleration field $\boldsymbol{a}(\boldsymbol{x}, \boldsymbol{v}, t|\boldsymbol{z})$ starting from the initial conditional distribution $\rho_0(\boldsymbol{x}, \boldsymbol{v}|\boldsymbol{z})$.

Theorem 4.4. *The marginal acceleration field is*

$$\boldsymbol{a}(\boldsymbol{x}, \boldsymbol{v}, t) := \mathbb{E}_{q(\boldsymbol{z})} \left\{ \frac{\boldsymbol{a}(\boldsymbol{x}, \boldsymbol{v}, t|\boldsymbol{z}) \rho_t(\boldsymbol{x}, \boldsymbol{v}|\boldsymbol{z})}{\rho_t(\boldsymbol{x}, \boldsymbol{v})} \right\} \tag{12}$$

See the proof in Section A.3.

Similar to first order flow matching, we parameterize the field by a neural network $\boldsymbol{a}_{\boldsymbol{\theta}}(\cdot, \cdot, \cdot) : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^1 \to \mathbb{R}^d$, aiming to minimize the global L_2 regression loss(acceleration flow matching loss):

$$\mathcal{L}_{\text{AFM}} = \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{v}) \sim \rho_t(\boldsymbol{x}, \boldsymbol{v}), t \sim \mathcal{U}[t_0, t_K]} \left\{ \|\boldsymbol{a}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{v}, t) - \boldsymbol{a}(\boldsymbol{x}, \boldsymbol{v}, t)\|^2 \right\}$$

However, $\boldsymbol{a}(\boldsymbol{x}, \boldsymbol{v}, t)$ is generally intractable because it is defined via an expectation (integral), and the inside denominator $\rho_t(\boldsymbol{x}, \boldsymbol{v})$ also requires integration to compute. In contrast, the conditional acceleration field $\boldsymbol{a}(\boldsymbol{x}, \boldsymbol{v}, t|z)$ and the conditional probability path $\rho_t(\boldsymbol{x}, \boldsymbol{v}|z)$ have simple forms. Therefore, in actual training, we use the following conditional acceleration flow matching (CAFM) objective:

$$\mathcal{L}_{\text{CAFM}} = \mathbb{E}_{\substack{(\boldsymbol{x}, \boldsymbol{v}) \sim \rho_t(\boldsymbol{x}, \boldsymbol{v}|z), \\ t \sim \mathcal{U}[t_0, t_K], z \sim q(z)}} \left[\|\boldsymbol{a}_\theta(\boldsymbol{x}, \boldsymbol{v}, t) - \boldsymbol{a}(\boldsymbol{x}, \boldsymbol{v}, t|z)\|^2 \right]$$

These two objectives are equivalent in training, as described by the following theorem:

Theorem 4.5. *If $\rho_t(\boldsymbol{x}, \boldsymbol{v}) > 0$ for any $\boldsymbol{x} \in \mathbb{R}^d, \boldsymbol{v} \in \mathbb{R}^d, t \in [t_0, t_K]$, then $\mathcal{L}_{AFM}$ and $\mathcal{L}_{CAFM}$ differ only by a constant independent of $\boldsymbol{\theta}$. In other words, $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{AFM} = \nabla_{\boldsymbol{\theta}} \mathcal{L}_{CAFM}$. The proof is left to Section A.4.*

The design of conditional probability path and the distribution of mixture variable $\boldsymbol{z}$ is based on the mathematical discussions of DOAT problem in (4.1).

Design of Conditional Probability Path and Mixture Variable Consider the joint distributions $\mu_0(\boldsymbol{x}, \boldsymbol{v})$, $\mu_1(\boldsymbol{x}, \boldsymbol{v})$, ..., $\mu_K(\boldsymbol{x}, \boldsymbol{v})$ at $K+1$ time points. The optimal transport plans calculated via solving SOAT problem are $\pi^*_{0\to1}, \pi^*_{1\to2}, \cdots, \pi^*_{K-1\to K}$. Taking the condition variable $\boldsymbol{z} = [(\boldsymbol{x}_0, \boldsymbol{v}_0), \cdots, (\boldsymbol{x}_K, \boldsymbol{v}_K)]$, and similar to standard flow matching, we choose a path with time-varying mean and invariant variance as the conditional probability path, i.e.:

$$\rho_t(\boldsymbol{x}, \boldsymbol{v}|z) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{m}_x(t), \sigma_x^2 I) \cdot \mathcal{N}(\boldsymbol{v}|\boldsymbol{m}_v(t), \sigma_v^2 I) \tag{13}$$

Here, $\boldsymbol{m}_x, \boldsymbol{m}_v$ are given by the minimum cost trajectory described in Proposition 4.1, where for $t \in [t_i, t_{i+1}]$:

$$\boldsymbol{m}_x(t) = c_3(t-t_i)^3 + c_2(t-t_i)^2 + c_1(t-t_i) + c_0 \quad \boldsymbol{m}_v(t) = 3c_3(t-t_i)^2 + 2c_2(t-t_i) + c_1$$

The coefficients are identical to those given in Section A.1, with $\boldsymbol{x}_0, \boldsymbol{v}_0$ replaced by $\boldsymbol{x}_i, \boldsymbol{v}_i$, $\boldsymbol{x}_T, \boldsymbol{v}_T$ replaced by $\boldsymbol{x}_{i+1}, \boldsymbol{v}_{i+1}$, and T replaced by $t_{i+1} - t_i$. Based on the optimal transport plans, we define the optimal propagator (transition kernel) from time t_i to t_{i+1} as the probability of a particle at $(\boldsymbol{x}_i, \boldsymbol{v}_i)$ at time t_i being transported to $(\boldsymbol{x}_{i+1}, \boldsymbol{v}_{i+1})$ at time t_{i+1}:

$$\mathcal{K}^*_{i\to i+1}[(\boldsymbol{x}_i, \boldsymbol{v}_i), (\boldsymbol{x}_{i+1}, \boldsymbol{v}_{i+1})] = \frac{\pi^*_{i\to i+1}[(\boldsymbol{x}_i, \boldsymbol{v}_i), (\boldsymbol{x}_{i+1}, \boldsymbol{v}_{i+1})]}{\mu_i(\boldsymbol{x}_i, \boldsymbol{v}_i)} \tag{14}$$

Then,

$$\boldsymbol{z} \sim \mu_0(\boldsymbol{x}_0, \boldsymbol{v}_0) \cdot \prod_{i=0}^{K-1} \mathcal{K}^*_{i\to i+1}[(\boldsymbol{x}_i, \boldsymbol{v}_i), (\boldsymbol{x}_{i+1}, \boldsymbol{v}_{i+1})]$$

In the following, we demonstrate that the constructed marginal probability path recovers the true distribution at every given time point.

Theorem 4.6. *When $\sigma_x \to 0, \sigma_v \to 0$, the marginal probability density constructed by Equation* (11) *satisfies $\rho_{t_i}(\boldsymbol{x}, \boldsymbol{v}) \to \mu_i(\boldsymbol{x}, \boldsymbol{v})$ at the $K+1$ given time points. Therefore, the marginal probability path constructed by mixing conditional probability paths can correctly reconstruct the marginal distributions. See the proof in Section A.5.*

Furthermore, we demonstrate that the design yields an exact solution to the DOAT problem.

Proposition 4.7. *As $\sigma_x \to 0$ and $\sigma_v \to 0$, the marginal probability path and the marginal acceleration field will converge to the solution to the DOAT problem. See the proof in Section A.6.*

4.3 Transform VM-DOAT to DOAT

Relations between DOAT and VM-DOAT The optimal transport cost of the DOAT problem is a functional of the sequence of joint distributions $\mu_i(\boldsymbol{x}, \boldsymbol{v})$ for $i = 0, \ldots, K$. Formally, we denote this dependency as $\mathcal{J}_{\text{DOAT}}[\{\mu_i(\boldsymbol{x}, \boldsymbol{v})\}|_{i=0}^K]$. In the VM-DOAT setting, we only have access to the marginal spatial distributions $\mu_i^{(\text{pos})}(\boldsymbol{x}) = \int \mu_i(\boldsymbol{x}, \boldsymbol{v}) \mathrm{d}\boldsymbol{v}$. Consequently, the cost of VM-DOAT, denoted as $\mathcal{J}_{\text{VM-DOAT}}[\{\mu_i^{(\text{pos})}(\boldsymbol{x})\}|_{i=0}^K]$, relates to $\mathcal{J}_{\text{DOAT}}$ through the following minimization:

$$\mathcal{J}_{\text{VM-DOAT}}[\{\mu_i^{(\text{pos})}(\boldsymbol{x})\}] = \min_{\omega_i(\boldsymbol{v}|\boldsymbol{x})} \mathcal{J}_{\text{DOAT}}[\{\mu_i^{(\text{pos})}(\boldsymbol{x}) \cdot \omega_i(\boldsymbol{v}|\boldsymbol{x})\}], \tag{15}$$

where $\omega_i(\boldsymbol{v}|\boldsymbol{x})$ represents the conditional distribution of velocity, such that the joint distribution factorizes as $\mu_i(\boldsymbol{x}, \boldsymbol{v}) = \mu_i^{(\text{pos})}(\boldsymbol{x}) \cdot \omega_i(\boldsymbol{v}|\boldsymbol{x})$. Thus, our objective is to determine the optimal $\omega_i(\boldsymbol{v}|\boldsymbol{x})$ that minimizes the expression above, thereby transforming the VM-DOAT problem into a solvable DOAT instance.

This transformation rests on two key properties: 1) The SOAT problem can be solved efficiently using the Sinkhorn algorithm. Once the Transport Plan is computed, we can sample a trajectory $[(\boldsymbol{x}_0, \boldsymbol{v}_0), \cdots, (\boldsymbol{x}_K, \boldsymbol{v}_K)]$; 2) Given the positions $\boldsymbol{x}_{0:K}$ of a trajectory, consider the following multi-marginal optimal control problem:

$$\min \int_{t_0}^{t_K} \frac{1}{2} \|\boldsymbol{\gamma}''(t)\|^2 \mathrm{d}t, \quad \text{s.t. } \boldsymbol{\gamma}(t_i) = \boldsymbol{x}_i,\ i = 0, \ldots, K. \tag{16}$$

The velocities of the optimal trajectory, denoted as $V = [\boldsymbol{\gamma}'(t_0), \cdots, \boldsymbol{\gamma}'(t_K)]$, can be obtained by solving a sparse linear system, as illustrated in the following proposition:

Proposition 4.8. *The velocities V defined above are the solution to the sparse linear system $VA = B$, where $A \in \mathbb{R}^{(K+1)\times(K+1)}$ is a tridiagonal matrix and $B \in \mathbb{R}^{d\times(K+1)}$. A detailed derivation and the explicit expressions for A and B are provided in Section A.7.*

It is worth noting that solving this linear system requires $\mathcal{O}(K)$ time, ensuring high computational efficiency.

Iterative Scheme to Perform the Transformation Based on this decomposition, we employ an iterative scheme to estimate the conditional velocity distributions across $t_0, \ldots, t_K$. We initialize $\omega_i^{(0)}(\boldsymbol{v}|\boldsymbol{x}) = \delta(\boldsymbol{v} - \boldsymbol{0})$ and subsequently alternate between the following two steps:

- *Solve SOAT Plans*: Compute the K optimal SOAT plans between the distributions $\mu_i^{(\text{pos})}(\boldsymbol{x}) \cdot \omega_i^{(m)}(\boldsymbol{v}|\boldsymbol{x})$ for adjacent time points. This yields the plans $\pi^{*(m)}_{i\to i+1}[(\boldsymbol{x}_i, \boldsymbol{v}_i), (\boldsymbol{x}_{i+1}, \boldsymbol{v}_{i+1})]$ for $i = 0, \ldots, K-1$, where m denotes the iteration index. This step effectively provides the multi-marginal optimal transport plan given the current $\omega_i^{(m)}(\boldsymbol{v}|\boldsymbol{x})$.

- *Update Velocity Distributions*: Update the conditional velocity distributions based on the computed SOAT plans:

$$\omega_i^{(m+1)}(\boldsymbol{v}|\boldsymbol{x}_i) = \frac{\int q^{(m)}(\boldsymbol{z})\delta(\hat{\boldsymbol{v}}_i[\boldsymbol{x}_0 \cdots \boldsymbol{x}_K] - \boldsymbol{v})\,\mathrm{d}\boldsymbol{z}_{\setminus i}}{\int q^{(m)}(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z}_{\setminus i}}, \tag{17}$$

where $\mathrm{d}\boldsymbol{z}_{\setminus i}$ denotes the volume element $\mathrm{d}\boldsymbol{x}_0\mathrm{d}\boldsymbol{v}_0 \ldots \mathrm{d}\boldsymbol{x}_K\mathrm{d}\boldsymbol{v}_K$ excluding $\mathrm{d}\boldsymbol{x}_i$ (i.e., integrating over all variables except $\boldsymbol{x}_i$). The term $\hat{\boldsymbol{v}}_i[\boldsymbol{x}_0, \ldots, \boldsymbol{x}_K]$ denotes the optimal velocity at time t_i derived from the sequence $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_K$ via Proposition 4.8. Intuitively, $\omega_i^{(m+1)}(\boldsymbol{v}|\boldsymbol{x}_i)$ is set to the distribution induced by the velocities of all optimal control trajectories passing through $\boldsymbol{x}_i$ at time t_i. Since the total transport cost is the sum of costs along individual trajectories, aligning $\boldsymbol{v}$ with the optimal velocity of each trajectory minimizes the global objective.

In the update step above, the trajectory variable $\boldsymbol{z} = [(\boldsymbol{x}_0, \boldsymbol{v}_0), \ldots, (\boldsymbol{x}_K, \boldsymbol{v}_K)]$ follows the joint distribution $q^{(m)}(\boldsymbol{z})$ defined as:

$$q^{(m)}(\boldsymbol{z}) = \mu_0^{(\mathrm{pos})}(\boldsymbol{x}_0) \cdot \omega_0^{(m)}(\boldsymbol{v}_0|\boldsymbol{x}_0) \times \prod_{i=0}^{K-1} \mathcal{K}_{i\to i+1}^{*(m)}[(\boldsymbol{x}_i, \boldsymbol{v}_i), (\boldsymbol{x}_{i+1}, \boldsymbol{v}_{i+1})],$$

where the propagator $\mathcal{K}_{i\to i+1}^{*(m)}$ is given by:

$$\mathcal{K}_{i\to i+1}^{*(m)}[(\boldsymbol{x}_i, \boldsymbol{v}_i), (\boldsymbol{x}_{i+1}, \boldsymbol{v}_{i+1})] = \frac{\pi_{i\to i+1}^{*(m)}[(\boldsymbol{x}_i, \boldsymbol{v}_i), (\boldsymbol{x}_{i+1}, \boldsymbol{v}_{i+1})]}{\mu_i^{(\mathrm{pos})}(\boldsymbol{x}_i) \cdot \omega_i^{(m)}(\boldsymbol{v}_i|\boldsymbol{x}_i)}.$$

During the iteration process, the objective $\mathcal{L}_{\mathrm{DOAT}}[\mu_i^{(\mathrm{pos})}(\boldsymbol{x}) \cdot \omega_i^{(m)}(\boldsymbol{v}|\boldsymbol{x})]$ decreases monotonically. Since the cost is bounded below by 0, the iterative algorithm is guaranteed to converge.

5 TracingFlow Algorithm for Trajectory Inference

In the practical implementation of TracingFlow, we first perform an approximation of the conditional velocity distribution to reduce the VM-DOAT problem to a DOAT problem. Subsequently, we employ flow matching to solve for the acceleration field of the DOAT problem. Notably, for lineage tracing data, we incorporate biological priors into this framework.

5.1 Approximation of Conditional Velocity Distribution

Determining the full conditional velocity distribution $\omega_i(\boldsymbol{v}|\boldsymbol{x})$ to transform VM-DOAT to DOAT (Section 4.3) is computationally prohibitive, potentially requiring an auxiliary generative model. We therefore adopt a deterministic approximation, assigning a single velocity to each data point. Consequently, Equation (17) is modified to:

$$\omega_i^{(m+1)}(\boldsymbol{v}|\boldsymbol{x}_i) = \delta(\boldsymbol{v} - \boldsymbol{v}_{\mathrm{exp}}^{(m+1)}(\boldsymbol{x}_i)), \quad \boldsymbol{v}_{\mathrm{exp}}^{(m+1)}(\boldsymbol{x}_i) = \frac{\int q^{(m)}(\boldsymbol{z})\hat{\boldsymbol{v}}_i[\boldsymbol{x}_0 \cdots \boldsymbol{x}_K]\,\mathrm{d}\boldsymbol{z}_{\setminus i}}{\int q^{(m)}(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z}_{\setminus i}}. \tag{18}$$

Intuitively, this assigns $\boldsymbol{v}_{\mathrm{exp}}^{(m+1)}(\boldsymbol{x}_i)$ as the mean velocity of all optimal control trajectories passing through $\boldsymbol{x}_i$ at time t_i.

5.2 MiniBatch-OT

Solving SOAT repeatedly is computationally intensive for large-scale datasets. However, since the conditional velocity distribution $\omega_i(\boldsymbol{v}|\boldsymbol{x})$ is approximated as a Dirac delta, the joint empirical distribution $\mu_i(\boldsymbol{x}, \boldsymbol{v})$ effectively becomes a superposition of Dirac deltas. This allows us to employ a minibatch strategy: we partition μ_i and μ_{i+1} into B corresponding minibatches $\{\mu_i^{(n)}\}_{n=1}^B$ and $\{\mu_{i+1}^{(n)}\}_{n=1}^B$. By solving the local plans $\pi_{i\to i+1}^{*(n)}$ independently and aggregating them as $\pi_{i\to i+1}^* = \oplus_{n=1}^B \pi_{i\to i+1}^{*(n)}$, we significantly enhance computational efficiency.

Table 1: Average $\mathcal{W}_1$ and $\mathcal{W}_2$ distances between the generated and ground truth distributions at various time points on the 2D Simulation , Cite 5D, and Cite 100D datasets for different algorithms.

Method	Dynamics	2D Simulation		Cite 5D		Cite 100D	
		$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$
OT-CFM	1st Order	0.5270 $\pm$ 0.0612	0.5922 $\pm$ 0.0813	0.8298 $\pm$ 0.0162	0.9155 $\pm$ 0.0160	10.5498 $\pm$ 0.0291	10.6220 $\pm$ 0.0310
SF^{2}M		0.9926 $\pm$ 0.1857	1.1318 $\pm$ 0.1734	0.7650 $\pm$ 0.0375	0.8956 $\pm$ 0.0408	12.3426 $\pm$ 0.0735	12.4757 $\pm$ 0.0783
3MSBM	2nd Order	1.1855 $\pm$ 0.2636	1.2767 $\pm$ 0.2215	3.6369 $\pm$ 0.3508	3.7703 $\pm$ 0.3178	15.7708 $\pm$ 0.1473	15.8833 $\pm$ 0.1832
MMFM		2.3689 $\pm$ 0.1830	2.6466 $\pm$ 0.1769	2.5346 $\pm$ 0.1520	2.7576 $\pm$ 0.1877	12.5353 $\pm$ 0.6268	12.7868 $\pm$ 0.7061
HRF		1.5390 $\pm$ 0.1130	1.6631 $\pm$ 0.1214	1.4016 $\pm$ 0.0437	1.5313 $\pm$ 0.0430	10.1532 $\pm$ 0.0643	10.2336 $\pm$ 0.0707
CAF		1.6385 $\pm$ 0.2777	1.6431 $\pm$ 0.2766	4.5589 $\pm$ 0.1073	4.7411 $\pm$ 0.1620	16.7429 $\pm$ 0.4371	16.9837 $\pm$ 0.4324
TF (Ours)		**0.3748** $\pm$ 0.0505	**0.5180** $\pm$ 0.0360	**0.5861** $\pm$ 0.0688	**0.6386** $\pm$ 0.0775	**8.0700** $\pm$ 0.1537	**8.2530** $\pm$ 0.1889

5.3 Incorporation of Biological Priors

In lineage tracing applications, let indices l_i and l_{i+1} denote individual samples at time points t_i and t_{i+1}, respectively. Each data point $(\boldsymbol{x}_{i,l_i}, \boldsymbol{v}_{i,l_i})$ is associated with a barcode $b_{i,l_i} \in \mathbb{N}^+$. We incorporate this prior by modifying the cost matrix $\mathcal{C}_{i\to i+1}$. Specifically, when computing the cost between the l-th sample at t_i and the k-th sample at t_{i+1}, we compare their barcodes: if $b_{i,l_i} \neq b_{i+1,l_{i+1}}$, the standard cost is multiplied by a penalty factor $p_0 (p_0 > 1)$. This soft constraint discourages transitions between distinct lineages, effectively embedding biological knowledge into the dynamics learning process.

5.4 Algorithm of TracingFlow

The TracingFlow algorithm comprises three main steps: First, we apply the iterative approximation from Section 5.1 to determine initial velocities, effectively transforming the VM-DOAT problem into a DOAT problem. Second, we compute the Optimal Transport Plan for SOAT as defined in Equation (10). Third, we sample conditional probability paths to train the acceleration network $\boldsymbol{a}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{v}, t)$ via regression. Additionally, to provide initial conditions for inference, we parameterize the unique initial velocities derived in the first step using a neural network $\boldsymbol{v}_{0,\boldsymbol{\xi}}(\boldsymbol{x})$, trained similarly via regression. The algorithm's pseudocode is provided in Section D.

Theoretically, TracingFlow recovers the exact distribution if the acceleration and initial velocity are perfectly fitted. In practice, where losses are non-zero, we prove that the positional $\mathcal{W}_2$ distance between the generated and true distributions is bounded by the sum of the velocity and AFM losses, and the relevant theorem is detailed in Section A.8.

6 Experiment Results

To evaluate the effectiveness of our algorithm, TracingFlow, we conducted three categories of experiments: 1) verifying whether the acceleration field of TracingFlow can transport the initial data distribution $\mu_0^{(\text{pos})}(\boldsymbol{x}) = \int \rho_{t_0}(\boldsymbol{x}, \boldsymbol{v}) \mathrm{d}\boldsymbol{v}$ to the data distribution $\mu_i^{(\text{pos})}(\boldsymbol{x})$ at other given times t_i; 2) verifying whether TracingFlow can effectively perform distribution interpolation and extrapolation; and 3) verifying whether TracingFlow can more effectively infer the dynamical laws in lineage tracing data given biological prior knowledge. The baselines include first order methods(OT-CFM[59], SF2M[61]) and second order methods(3MSBM[57], MMFM[50], HRF[71], CAF[42]).

Distribution Transport We evaluated TracingFlow on a 2-dimensional synthetic dataset and real single-cell omics datasets of varying dimensions[31]. The evaluation metrics were the 1-Wasserstein ($\mathcal{W}_1$) and 2-Wasserstein ($\mathcal{W}_2$) distances, measuring the discrepancy between the fitted and ground truth distributions. The second-order dynamics model of TracingFlow provides stronger expressive power than first-order algorithms, yielding lower average $\mathcal{W}_1$ and $\mathcal{W}_2$ values across all datasets (Table 1). We visualized the evolutionary trajectories learned by OT-CFM and TracingFlow on the synthetic dataset in Figure 3. TracingFlow's second-order model allows it to learn trajectories that cross in the position space $\boldsymbol{x}_1, \boldsymbol{x}_2$, whereas OT-CFM is unable to do so. In Section C.1, we present the distribution reconstruction accuracy at each time point, as well as experimental results on additional datasets (such as the Mexico Gulf dataset[57, 53]).

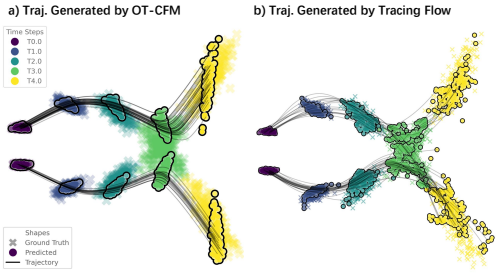


Figure 3: On the Simulation 2D dataset: a) non-crossing paths learned by OT-CFM, and b) crossing paths learned by TracingFlow.

Table 2: Average $\mathcal{W}_1$ and $\mathcal{W}_2$ distances between the predicted and ground truth distributions at held-out time points on the EB 5D dataset for different algorithms.

Method	Dynamics	EB 5D	
		$\mathcal{W}_1$	$\mathcal{W}_2$
OT-CFM	1st Order	5.1205 $\pm$ 0.2400	5.6910 $\pm$ 0.2561
SF^{2}M		4.5602 $\pm$ 0.1041	4.9905 $\pm$ 0.1262
3MSBM	2nd Order	7.1901 $\pm$ 0.8667	7.4697 $\pm$ 0.8588
MMFM		9.7777 $\pm$ 0.9006	10.5980 $\pm$ 0.9407
TF (Ours)		**4.4974** $\pm$ 0.0950	**4.7859** $\pm$ 0.1857

Table 3: Average lineage-weighted $\mathcal{W}_1$ and $\mathcal{W}_2$ distances between the generated and ground truth distributions at various time points on the 3D Simulation Lineage and Hematopoiesis datasets for different algorithms.

Method	Dynamics	3D Sim-Lineage		Hematopoiesis	
		$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$
OT-CFM	1st Order	2.2383 $\pm$ 0.0205	2.3162 $\pm$ 0.0216	14.9945 $\pm$ 0.3442	15.3971 $\pm$ 0.3180
SF^{2}M		1.5255 $\pm$ 0.0513	1.5435 $\pm$ 0.0515	18.7170 $\pm$ 0.1124	18.9756 $\pm$ 0.1168
3MSBM	2nd Order	2.3126 $\pm$ 0.4756	2.4905 $\pm$ 0.4582	18.3313 $\pm$ 0.9834	18.6427 $\pm$ 0.9413
MMFM		2.6098 $\pm$ 0.3149	3.0309 $\pm$ 0.3265	27.0830 $\pm$ 2.9299	29.5421 $\pm$ 3.1420
HRF		1.7934 $\pm$ 0.4043	1.8085 $\pm$ 0.4022	15.4571 $\pm$ 0.1266	15.7138 $\pm$ 0.1282
CAF		2.6806 $\pm$ 0.2777	2.7547 $\pm$ 0.3521	15.2165 $\pm$ 0.0685	15.4974 $\pm$ 0.0576
TF w/o Bio-Prior		1.5292 $\pm$ 0.1714	1.5532 $\pm$ 0.2303	16.6166 $\pm$ 0.1229	17.0110 $\pm$ 0.1456
TF (Ours)		**0.4538** $\pm$ 0.2039	**0.5328** $\pm$ 0.2066	**14.0470** $\pm$ 0.1530	**14.4589** $\pm$ 0.1815

Interpolation and Extrapolation To verify TracingFlow's interpolation and extrapolation capabilities, we conducted a Hold-One-Out experiment on the 5-dimensional EB dataset (time points $t = 0, \ldots, 4$)[39]. In each iteration, we withheld one time point from the latter four for training and calculated the $\mathcal{W}_1$ and $\mathcal{W}_2$ distances between the interpolated and true distributions. As shown in Table 2, TracingFlow achieved the highest accuracy, demonstrating that models based on second-order dynamics are superior in capturing high-curvature and non-linear trajectories.

Lineage Tracing Data To verify whether TracingFlow better handles lineage tracing data given biological priors, we experimented on a synthetic 3D simulation lineage dataset and the Hematopoiesis dataset[66]. Following Section 5.3, we introduced biological priors into TracingFlow by modifying the transport cost matrix, other baselines also incorporates priors via conditional velocity fields. We used lineage-weighted $\mathcal{W}_1$ and $\mathcal{W}_2$ distances to evaluate consistency with biological priors while learning dynamics. Results in Table 3 show that incorporating priors enables TracingFlow to outperform other algorithms. In Figure 4, we visualized dynamics on the simulation dataset to demonstrate how TracingFlow learns trajectories conforming to priors. In Figure 5, we plotted the PCA-reduced positions of two barcodes from the Hematopoiesis Dataset at $t = 1$; the positions inferred by TracingFlow are significantly closer to the ground truth than those by OT-CFM.

7 Conclusion and Limitation

In this work, we introduce TracingFlow, a Flow Matching framework for the Dynamical Acceleration Optimal Transport (DOAT) problem. By pre-solving SOAT to learn acceleration and initial velocity, it enables efficient, simulation-free solutions for large-scale DOAT problem. Experiments on simulated and real world data demonstrate that second-order dynamics enhance expressivity, yielding precise temporal distribution recovery. Additionally, incorporating biological priors better preserves intercellular lineages.

A limitation is the computationally intensive SOAT pre-calculation, though this is mitigable via minibatch-OT. Furthermore, explicitly modeling the theoretical conditional velocity distribution per data point is prohibitive, so we approximate it using its expectation. The objective $\int_0^T \frac{1}{2}\|\boldsymbol{a}\|^2 \mathrm{d}t$ also lacks a clear physical interpretation, as standard classical actions exclude second time derivatives (see Section E.2). Despite current sequencing lacking velocity data, physics and biological priors

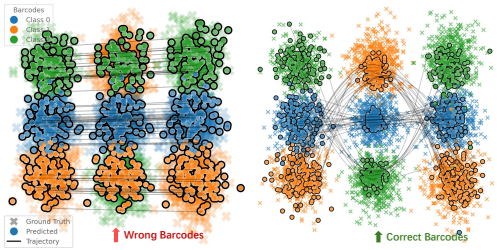


Figure 4: On the Sim-Lineage dataset: a) paths learned by OT-CFM without considering biological priors, and b) paths learned by TracingFlow that preserve biological priors.

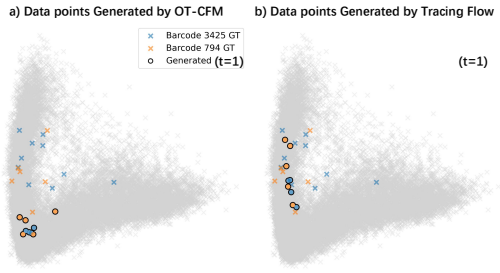


Figure 5: Comparison of generated (circles) and ground truth ('x') positions at $t = 1$ for two barcodes on the hematopoiesis dataset [66]. (a) OT-CFM. (b) TracingFlow. Visualized via 2D PCA.

suggest biological systems follow second-order dynamics. Future work will extend TracingFlow to further model these dynamics.

Acknowledgments and Disclosure of Funding

Peijie Zhou acknowledges support from the National Natural Science Foundation of China under grants 12288101, 8206100646, and T2321001, the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China under grant JYB2025XDXM502, and the Fundamental Research Funds for the Central Universities. We also acknowledge the support from the High-performance Computing Platform of Peking University for computation. Furthermore, we thank Jingyuan Lu for helpful discussions on lineage-traced scRNA-seq datasets.

References

[1] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

[2] 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.

[3] Vladimir Igorevich Arnol'd. *Mathematical methods of classical mechanics*, volume 60. Springer Science & Business Media, 2013.

[4] 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.

[5] Jean-David Benamou, Thomas O Gallouët, and François-Xavier Vialard. Second-order models for optimal transport and cubic splines on the wasserstein space. *Foundations of Computational Mathematics*, 19(5):1113–1143, 2019.

[6] 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*, pages 5802–5833. PMLR, 2023.

[7] Charlotte Bunne, Geoffrey Schiebinger, Andreas Krause, Aviv Regev, and Marco Cuturi. Optimal transport for single-cell and spatial omics. *Nature Reviews Methods Primers*, 4(1):58, 2024.

[8] Zihan Cao, Yu Zhong, and Liang-Jian Deng. Taming flow matching with unbalanced optimal transport into fast pansharpening. *arXiv preprint arXiv:2503.14975*, 2025.

[9] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[10] Tianrong Chen, Guan-Horng Liu, and Evangelos Theodorou. Likelihood training of schrödinger bridge using forward-backward SDEs theory. In *International Conference on Learning Representations*, 2022.

[11] Yongxin Chen, Giovanni Conforti, Tryphon T Georgiou, and Luigia Ripani. Multi-marginal schrödinger bridges. In *International Conference on Geometric Science of Information*, pages 725–732. Springer, 2019.

[12] 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.

[13] Gabriele Corso, Vignesh Ram Somnath, Noah Getz, Regina Barzilay, Tommi Jaakkola, and Andreas Krause. Composing unbalanced flows for flexible docking and relaxation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[14] Valentin De Bortoli, Guan-Horng Liu, Tianrong Chen, Evangelos A Theodorou, and Weilie Nie. Augmented bridge matching. *arXiv preprint arXiv:2311.06978*, 2023.

[15] 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.

[16] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, et al. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

[17] Aden Forrow and Geoffrey Schiebinger. Lineageot is a unified framework for lineage tracing and trajectory inference. *Nature Communications*, 12(1), 2021.

[18] Mingze Gao, Melania Barile, Shirom Chabra, Myriam Haltalli, Emily F. Calderbank, Yiming Chao, Weizhong Zheng, Nicola K. Wilson, Elisa Laurenti, Berthold Göttgens, and Yuanhua Huang. CLADES: a hybrid neuralode-gillespie approach for unveiling clonal cell fate and differentiation dynamics. *Nature Communications*, 16(1), 2025.

[19] Gennady Gorin, Valentine Svensson, and Lior Pachter. Protein velocity and acceleration from single-cell multiomics experiments. *Genome biology*, 21(1):39, 2020.

[20] Pengsheng Guo and Alexander G Schwing. Variational rectified flow matching. *arXiv preprint arXiv:2502.09616*, 2025.

[21] Wenbo Guo, Zeyu Chen, Xinqi Li, Jingmin Huang, Qifan Hu, and Jin Gu. sctrace+: Enhancing cell fate inference by integrating the lineage-tracing and multi-faceted transcriptomic similarity information. *Cell Systems*, 16(9):101398, 2025.

[22] Peter Halmos, Xinhao Liu, Julian Gold, Feng Chen, Li Ding, and Benjamin J Raphael. Dest-ot: Alignment of spatiotemporal transcriptomics data. *Cell Systems*, 16(2), 2025.

[23] Ari Hong, Sangseon Lee, and Kwangsoo Kim. Multi-omic relay velocity modeling uncovers dynamic chromatin-transcription regulation across cell states. *Nature Communications*, 2025.

[24] 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.

[25] 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.

[26] Kacper Kapusniak, Peter Potaptchik, Teodora Reu, Leo Zhang, Alexander Tong, Michael Bronstein, Joey Bose, and Francesco Di Giovanni. Metric flow matching for smooth interpolations on the data manifold. *Advances in Neural Information Processing Systems*, 37:135011–135042, 2024.

[27] Dominik Klein, Théo Uscidda, Fabian Theis, and Marco Cuturi. Genot: Entropic (gromov) wasserstein flow matching with applications to single-cell genomics. *Advances in Neural Information Processing Systems*, 37:103897–103944, 2024.

[28] Dominik Klein, Giovanni Palla, Marius Lange, Michal Klein, Zoe Piran, Manuel Gander, Laetitia Meng-Papaxanthos, Michael Sterr, Lama Saber, Changying Jing, et al. Mapping cells through time and space with moscot. *Nature*, pages 1–11, 2025.

[29] Takeshi Koshizuka and Issei Sato. Neural lagrangian schrödinger bridge: Diffusion modeling for population dynamics. In *The Eleventh International Conference on Learning Representations*, 2023.

[30] Kai Kretzschmar and Fiona M. Watt. Lineage tracing. *Cell*, 148(1–2):33–45, 2012.

[31] Christopher Lance, Malte D Luecken, Daniel B Burkhardt, Robrecht Cannoodt, Pia Rautenstrauch, Anna Laddach, Aidyn Ubingazhibov, Zhi-Jie Cao, Kaiwen Deng, Sumeer Khan, et al. Multimodal single cell data integration challenge: results and lessons learned. *BioRxiv*, pages 2022–04, 2022.

[32] Marius Lange, Zoe Piran, Michal Klein, Bastiaan Spanjaard, Dominik Klein, Jan Philipp Junker, Fabian J. Theis, and Mor Nitzan. Mapping lineage-traced cells across time points with moslin. *Genome Biology*, 25(1), 2024.

[33] 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.

[34] Chen Li, Maria C Virgilio, Kathleen L Collins, and Joshua D Welch. Multi-omic single-cell velocity models epigenome–transcriptome interactions and improves cell fate prediction. *Nature biotechnology*, 41(3):387–398, 2023.

[35] 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.

[36] 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.

[37] Suryanarayana Maddu, Victor Chardès, Michael Shelley, et al. Inferring biological processes with intrinsic noise from cross-sectional data. *arXiv preprint arXiv:2410.07501*, 2024.

[38] Shanjun Mao, Chenyang Zhang, Runjiu Chen, Shan Tang, Xiaodan Fan, and Jie Hu. Cell lineage tracing: Methods, applications, and challenges. *Quantitative Biology*, 13(4), 2025.

[39] Kevin R Moon, David Van Dijk, Zheng Wang, Scott Gigante, Daniel B Burkhardt, William S Chen, Kristina Yim, Antonia van den Elzen, Matthew J Hirn, Ronald R Coifman, et al. Visualizing structure and transitions in high-dimensional biological data. *Nature biotechnology*, 37 (12):1482–1492, 2019.

[40] Kirill Neklyudov, Rob Brekelmans, Daniel Severo, and Alireza Makhzani. Action matching: Learning stochastic dynamics from samples. In *International conference on machine learning*, pages 25858–25889. PMLR, 2023.

[41] 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.

[42] Dogyun Park, Sojin Lee, Sihyeon Kim, Taehoon Lee, Youngjoon Hong, and Hyunwoo J Kim. Constant acceleration flow. *Advances in Neural Information Processing Systems*, 37: 90030–90060, 2024.

[43] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[44] Qiangwei Peng, Peijie Zhou, and Tiejun Li. stVCR: Spatiotemporal dynamics of single cells. *bioRxiv*, pages 2024–06, 2024.

[45] Qiangwei Peng, Zihan Wang, Junda Ying, Yuhao Sun, Qing Nie, Lei Zhang, Tiejun Li, and Peijie Zhou. WFR-FM: Simulation-free dynamic unbalanced optimal transport. *arXiv preprint arXiv:2601.06810*, 2026.

[46] 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.

[47] Gabriel Peyré and Marco Cuturi. *Computational optimal transport: With applications to data science*. Now Foundations and Trends, 2019.

[48] Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky TQ Chen. Multisample flow matching: Straightening flows with minibatch couplings. *arXiv preprint arXiv:2304.14772*, 2023.

[49] Neha Prasad, Karren Yang, and Caroline Uhler. Optimal transport using gans for lineage tracing. *arXiv preprint arXiv:2007.12098*, 2020.

[50] Martin Rohbeck, Edward De Brouwer, Charlotte Bunne, 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.

[51] 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, 2019.

[52] 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.

[53] Yunyi Shen, Renato Berlinghieri, and Tamara Broderick. Multi-marginal schr\" odinger bridges with iterative reference refinement. *arXiv preprint arXiv:2408.06277*, 2024.

[54] Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. *Advances in Neural Information Processing Systems*, 36, 2024.

[55] Sagar Shrestha and Xiao Fu. Diversified flow matching with translation identifiability. *arXiv preprint arXiv:2511.05558*, 2025.

[56] Yuhao Sun, Zhenyi Zhang, Zihan Wang, Tiejun Li, and Peijie Zhou. Variational regularized unbalanced optimal transport: Single network, least action. *arXiv preprint arXiv:2505.11823*, 2025.

[57] Panagiotis Theodoropoulos, Augustinos D Saravanos, Evangelos A Theodorou, and Guan-Horng Liu. Momentum multi-marginal schr\" odinger bridge matching. *arXiv preprint arXiv:2506.10168*, 2025.

[58] 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*, pages 9526–9536. PMLR, 2020.

[59] Alexander Tong, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Kilian Fatras, Guy Wolf, and Yoshua Bengio. Conditional flow matching: Simulation-free dynamic optimal transport. *arXiv preprint arXiv:2302.00482*, 2(3), 2023.

[60] 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*, 2024. ISSN 2835-8856. Expert Certification.

[61] 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*, pages 1279–1287. PMLR, 2024.

[62] 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.

[63] Dongyi Wang, Yuanwei Jiang, Zhenyi Zhang, Xiang Gu, Peijie Zhou, and Jian Sun. Joint velocity-growth flow matching for single-cell dynamics modeling. *arXiv preprint arXiv:2505.13413*, 2025.

[64] Kun Wang, Liangzhen Hou, Xin Wang, Xiangwei Zhai, Zhaolian Lu, Zhike Zi, Weiwei Zhai, Xionglei He, Christina Curtis, Da Zhou, and Zheng Hu. Phylovelo enhances transcriptomic velocity field mapping using monotonically expressed genes. *Nature Biotechnology*, 42(5): 778–789, 2023.

[65] Shou-Wen Wang, Michael J. Herriges, Kilian Hurley, Darrell N. Kotton, and Allon M. Klein. Cospar identifies early cell fate biases from single-cell transcriptomic and lineage information. *Nature Biotechnology*, 40(7):1066–1074, 2022.

[66] 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.

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

[68] 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, 2021.

[69] 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 2024. ISSN 1367-4811.

[70] Xi Nicole Zhang, Yuan Pu, Yuki Kawamura, Andrew Loza, Yoshua Bengio, Dennis Shung, and Alexander Tong. Trajectory flow matching with applications to clinical time series modelling. *Advances in Neural Information Processing Systems*, 37:107198–107224, 2024.

[71] Yichi Zhang, Yici Yan, Alex Schwing, and Zhizhen Zhao. Towards hierarchical rectified flow. *arXiv preprint arXiv:2502.17436*, 2025.

[72] 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*, 2025.

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

[74] Grace XY Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler, Geoff P McDermott, Junjie Zhu, et al. Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 8(1):14049, 2017.

[75] Qunxi Zhu and Wei Lin. Switched flow matching: Eliminating singularities via switching odes. *arXiv preprint arXiv:2405.11605*, 2024.

[76] 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.

Contents of Appendix

A Proofs of Main Theorems 16

A.1 Proof of Proposition 4.1 : Optimal Trajectory of Single Particle 16

A.3 Proof of Remark 4.4 : No-Cross Property 17

A.4 Proof of Theorem 4.5 : Marginal Acceleration Field 18

A.5 Proof of Theorem 4.6 : The Relation Between AFM Loss and CAFM Loss 18

A.6 Proof of Theorem 4.7 : Marginal Probability Path 19

A.7 Proof of Proposition 4.8 : TracingFlow Exactly Solves DOAT Problem 20

A.8 Proof of Proposition 4.9 : Solve Linear System to Get Optimal Velocity 21

A.9 Proof of Theorem A.1 : $\mathcal{W}_2$ Distance can be bounded by $\mathcal{L}_{\text{AFM}}$ and $\mathcal{L}_{v_0}$ 22

B Datasets and Evaluation Metric 25

B.1 Experiment Setup 25

B.2 Evaluation Metrics 25

B.3 Datasets 26

C Experiment Details 26

C.1 Experiment Details across Various Datasets 26

C.2 Training Time and Scalability of Tracing Flow 30

C.3 Sensitivity Analysis on Minibatch-OT 31

D Pseudocode for Algorithm 33

E Discussion 34

E.1 Relation with other Algorithms 34

E.2 Is $\mathcal{S} = \mathcal{X} \times \mathcal{V}$ a phase space? 34

A Proofs of Main Theorems

A.1 Proof of Proposition 4.1

Proof. First, we show that the unique minimizer is a cubic polynomial. Consider the energy functional defined on the interval $[0, T]$:

$$\mathcal{J}[\gamma] = \frac{1}{2} \int_0^T \|\gamma''(t)\|_2^2 \, \mathrm{d}t. \tag{19}$$

The integrand $L(\gamma, \gamma', \gamma'') = \frac{1}{2}\|\gamma''\|_2^2$ depends on derivatives up to the second order. The necessary condition for optimality is given by the Euler–Lagrange equation:

$$\frac{\partial L}{\partial \gamma} - \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \gamma'}\right) + \frac{\mathrm{d}^2}{\mathrm{d}t^2}\left(\frac{\partial L}{\partial \gamma''}\right) = 0. \tag{20}$$

Since $\partial L/\partial \gamma = 0$, $\partial L/\partial \gamma' = 0$, and $\partial L/\partial \gamma'' = \gamma''$, the equation simplifies to:

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}\gamma''(t) = \gamma^{(4)}(t) = 0. \tag{21}$$

This implies that the optimal trajectory $\gamma(t)$ is a cubic polynomial:

$$\gamma(t) = c_3t^3 + c_2t^2 + c_1t + c_0. \tag{22}$$

Imposing the boundary conditions $\gamma(0) = \boldsymbol{x}_0, \gamma'(0) = \boldsymbol{v}_0$ and $\gamma(T) = \boldsymbol{x}_T, \gamma'(T) = \boldsymbol{v}_T$ leads to the linear system:

$$\begin{cases} c_0 = \boldsymbol{x}_0, \\ c_1 = \boldsymbol{v}_0, \\ c_3T^3 + c_2T^2 + c_1T + c_0 = \boldsymbol{x}_T, \\ 3c_3T^2 + 2c_2T + c_1 = \boldsymbol{v}_T. \end{cases} \tag{23}$$

Solving for the coefficients yields the unique solution:

$$\begin{cases} c_0 = \boldsymbol{x}_0, \\ c_1 = \boldsymbol{v}_0, \\ c_2 = \dfrac{3(\boldsymbol{x}_T - \boldsymbol{x}_0) - (2\boldsymbol{v}_0 + \boldsymbol{v}_T)T}{T^2}, \\ c_3 = \dfrac{(\boldsymbol{v}_0 + \boldsymbol{v}_T)T - 2(\boldsymbol{x}_T - \boldsymbol{x}_0)}{T^3}. \end{cases} \tag{24}$$

□

A.2 Proof of Remark 4.3

Proof. First we propose a lemma:

Lemma. Let $\mathcal{C}_{[0,t]}(\boldsymbol{x}_0, \boldsymbol{v}_0, \boldsymbol{x}_t, \boldsymbol{v}_t)$ be the least acceleration cost through $[0, t]$,$(\boldsymbol{x_0}, \boldsymbol{v}_0), (\boldsymbol{x}_T, \boldsymbol{v}_T) \in \mathcal{X} \times \mathcal{V}$ are two different points. Then for arbitrary midpoint $\tau \in (0, 1)$ and state $(\boldsymbol{x}, \boldsymbol{v}) \in \mathcal{X} \times \mathcal{V}$, the following inequality holds:

$$\mathcal{C}_{[0,\tau]}(\boldsymbol{x_0}, \boldsymbol{v_0}, \boldsymbol{x}, \boldsymbol{v}) + \mathcal{C}_{[\tau,T]}(\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{x}_T, \boldsymbol{v}_T) \geq \mathcal{C}_{[0,T]}(\boldsymbol{x}_0, \boldsymbol{v}_0, \boldsymbol{x}_T, \boldsymbol{v}_T) \tag{25}$$

The "=" holds if and only if $(\boldsymbol{x}, \boldsymbol{v}) = \gamma^*(\tau)$, where $\gamma^*(t)(t \in [0, T])$ is the cubic interpolation trajectory connecting $(\boldsymbol{x_0}, \boldsymbol{v_0})$ and $(\boldsymbol{x}_T, \boldsymbol{v_T})$.

Proof. Fix $\tau \in (0, T)$. Define objective function $F_\tau(\boldsymbol{x}, \boldsymbol{v}) = \mathcal{C}_{[0,\tau]}(\boldsymbol{x_0}, \boldsymbol{v_0}, \boldsymbol{x}, \boldsymbol{v}) + \mathcal{C}_{[\tau,T]}(\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{x}_T, \boldsymbol{v}_T)$. By Corollary 4.2 , we can rewrite

$$\begin{aligned} F_\tau(\boldsymbol{x}, \boldsymbol{v}) = \frac{2}{\tau^3}\left\{(\|\boldsymbol{v}_0\|^2 + \langle \boldsymbol{v}_0, \boldsymbol{v}\rangle + \|\boldsymbol{v}\|^2)\tau^2 + 3\langle \boldsymbol{v}_0 + \boldsymbol{v}, \boldsymbol{x}_0 - \boldsymbol{x}\rangle\tau + 3\|\boldsymbol{x}_0 - \boldsymbol{x}\|^2\right\} + \\ \frac{2}{(T-\tau)^3}\left\{(\|\boldsymbol{v}_T\|^2 + \langle \boldsymbol{v}_T, \boldsymbol{v}\rangle + \|\boldsymbol{v}\|^2)(T-\tau)^2 + 3\langle \boldsymbol{v}_T + \boldsymbol{v}, \boldsymbol{x} - \boldsymbol{x}_T\rangle T + 3\|\boldsymbol{x}_T - \boldsymbol{x}\|^2\right\} \end{aligned} \tag{26}$$

Which is a quadratic form of $(\boldsymbol{x}, \boldsymbol{v})$ up to a constant. $F_\tau(\boldsymbol{x}, \boldsymbol{v}) \geq 0$ always holds, namely, the quadratic form has a finite lower bound, so it is semi-quadratic. To minimize $F_\tau(\boldsymbol{x}, \boldsymbol{v})$, we can take differentiations:

$$\nabla_{\boldsymbol{x}} F_\tau(\boldsymbol{x}, \boldsymbol{v}) = 3\left(\frac{-(\boldsymbol{v}_0 + \boldsymbol{v})\tau + 2(\boldsymbol{x} - \boldsymbol{x}_0)}{\tau^2} + \frac{(\boldsymbol{v} + \boldsymbol{v}_T)(T-\tau) + 2(\boldsymbol{x} - \boldsymbol{x}_T)}{(T-\tau)^2}\right) = 0. \tag{27}$$

$$\nabla_{\boldsymbol{v}} F_\tau(\boldsymbol{x}, \boldsymbol{v}) = \frac{(\boldsymbol{v}_0 + 2\boldsymbol{v})\tau + 3(\boldsymbol{x}_0 - \boldsymbol{x})}{\tau^2} + \frac{(\boldsymbol{v}_T + 2\boldsymbol{v})(T-\tau) + 3(\boldsymbol{x} - \boldsymbol{x}_T)}{(T-\tau)^2} = 0, \tag{28}$$

Denote the cubic interpolation from $(\boldsymbol{x_0}, \boldsymbol{v_0})$ to $(\boldsymbol{x}, \boldsymbol{v})$ on $[0, \tau]$ and $(\boldsymbol{x}, \boldsymbol{v})$ to $(\boldsymbol{x}_T, \boldsymbol{v}_T)$ on $[\tau, T]$ by $\gamma_{[0,\tau]}(t)$ and $\gamma_{[\tau,T]}(t)$, respectively.The coefficients are known according to Section A.1. Then we have:

$$\gamma''_{[0,\tau]}(\tau^-) = 6\cdot\frac{(\boldsymbol{v}_0 + \boldsymbol{v})\tau - 2(\boldsymbol{x} - \boldsymbol{x}_0)}{\tau^2} + 2\cdot\frac{3(\boldsymbol{x} - \boldsymbol{x}_0) - (2\boldsymbol{v}_0 + \boldsymbol{v})\tau}{\tau^2} = 2\cdot\frac{3(\boldsymbol{x_0} - \boldsymbol{x}) + (\boldsymbol{v_0} + 2\boldsymbol{v})\tau}{\tau^2}, \tag{29}$$

$$\begin{aligned} \gamma''_{[\tau,T]}(\tau^+) &= 6\cdot\frac{(\boldsymbol{v} + \boldsymbol{v}_T)\tau - 2(\boldsymbol{x}_T - \boldsymbol{x})}{\tau^2} + 2\cdot\frac{3(\boldsymbol{x}_T - \boldsymbol{x}) - (2\boldsymbol{v} + \boldsymbol{v}_T)(T-\tau)}{(T-\tau)^2} \\ &= 2\cdot\frac{3(\boldsymbol{x} - \boldsymbol{x}_T) + (\boldsymbol{v} + 2\boldsymbol{v}_T)(T-\tau)}{(T-\tau)^2} \end{aligned} \tag{30}$$

$$\gamma^{'''}_{[0,\tau]}(\tau^-) = 6 \cdot \frac{(\boldsymbol{v}_0 + \boldsymbol{v})\tau - 2(\boldsymbol{x} - \boldsymbol{x}_0)}{\tau^2} \tag{31}$$

$$\gamma^{'''}_{[\tau,T]}(\tau^+) = 6 \cdot \frac{(\boldsymbol{v} + \boldsymbol{v}_T)(T - \tau) - 2(\boldsymbol{x}_T - \boldsymbol{x})}{(T - \tau)^2} \tag{32}$$

Then we can find that 27 and 28 appear to be $\gamma^{'''}_{[0,\tau]}(\tau^-) = \gamma^{'''}_{[\tau,T]}(\tau^+), \gamma^{''}_{[0,\tau]}(\tau^-) = \gamma^{''}_{[\tau,T]}(\tau^+)$. Because the two curves are cubic, and $\gamma^{(i)}_{[0,\tau]}(\tau^-) = \gamma^{(i)}_{[\tau,T]}(\tau^+), i = 0, 1, 2, 3$, so we know that in the optimal case $\gamma_{[0,\tau]} = \gamma^*\big|_{[0,\tau]}, \gamma_{[\tau,T]} = \gamma^*\big|_{[\tau,T]}$.. That is, the unique solution of 27 and 28 is $(\gamma^*(\tau), \gamma^{*'}(\tau))$, which completes the proof. □

Back to our target, assume γ and $\tilde{\gamma}$ cross at $(\boldsymbol{x}_\tau, \boldsymbol{v}_\tau)$, then we have

$$\begin{aligned}
&\mathcal{C}_{[0,T]}(\boldsymbol{x}_0, \boldsymbol{v}_0, \boldsymbol{x}_T, \boldsymbol{v}_T) + \mathcal{C}_{[0,T]}(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{v}}_0, \tilde{\boldsymbol{x}}_T, \tilde{\boldsymbol{v}}_T) \\
=&\big\{\mathcal{C}_{[0,\tau]}(\boldsymbol{x}_0, \boldsymbol{v}_0, \boldsymbol{x}_\tau, \boldsymbol{v}_\tau) + \mathcal{C}_{[\tau,T]}(\boldsymbol{x}_\tau, \boldsymbol{v}_\tau, \boldsymbol{x}_T, \boldsymbol{v}_T)\big\} + \big\{\mathcal{C}_{[0,\tau]}(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{v}}_0, \boldsymbol{x}_\tau, \boldsymbol{v}_\tau) + \mathcal{C}_{[\tau,T]}(\boldsymbol{x}_\tau, \boldsymbol{v}_\tau, \tilde{\boldsymbol{x}}_T, \tilde{\boldsymbol{v}}_T)\big\} \\
=&\big\{\mathcal{C}_{[0,\tau]}(\boldsymbol{x}_0, \boldsymbol{v}_0, \boldsymbol{x}_\tau, \boldsymbol{v}_\tau) + \mathcal{C}_{[\tau,T]}(\boldsymbol{x}_\tau, \boldsymbol{v}_\tau, \tilde{\boldsymbol{x}}_T, \tilde{\boldsymbol{v}}_T)\big\} + \big\{\mathcal{C}_{[0,\tau]}(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{v}}_0, \boldsymbol{x}_\tau, \boldsymbol{v}_\tau) + \mathcal{C}_{[\tau,T]}(\boldsymbol{x}_\tau, \boldsymbol{v}_\tau, \boldsymbol{x}_T, \boldsymbol{v}_T)\big\} \\
\geq&\mathcal{C}_{[0,T]}(\boldsymbol{x}_0, \boldsymbol{v}_0, \tilde{\boldsymbol{x}}_T, \tilde{\boldsymbol{v}}_T) + \mathcal{C}_{[0,T]}(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{v}}_0, \boldsymbol{x}_T, \boldsymbol{v}_T).
\end{aligned} \tag{33}$$

The equality holds if and only if γ and $\tilde{\gamma}$ coincide. Consequently, the strict inequality holds for distinct trajectories, which implies the non-crossing property. This completes the proof. □

A.3 Proof of Theorem 4.4

Proof. Consider the time derivative of the marginal distribution:

$$\begin{aligned}
\partial_t \rho_t(\boldsymbol{x}, \boldsymbol{v}) &= \partial_t \left\{ \int \rho_t(\boldsymbol{x}, \boldsymbol{v}|z) q(z) \mathrm{d}z \right\} \\
&= \int \partial_t \rho_t(\boldsymbol{x}, \boldsymbol{v}|z) q(z) \mathrm{d}z \\
&= -\int \nabla_{\boldsymbol{x}} \cdot (\rho_t(\boldsymbol{x}, \boldsymbol{v}|z)\boldsymbol{v}) q(z) \mathrm{d}z - \int \nabla_{\boldsymbol{v}} \cdot (\rho_t(\boldsymbol{x}, \boldsymbol{v}|z)\boldsymbol{a}(\boldsymbol{x}, \boldsymbol{v}, t|z)) q(z) \mathrm{d}z \\
&= -\nabla_{\boldsymbol{x}} \cdot \left(\int \rho_t(\boldsymbol{x}, \boldsymbol{v}|z)\boldsymbol{v} q(z) \mathrm{d}z \right) - \nabla_{\boldsymbol{v}} \cdot \left(\int \rho_t(\boldsymbol{x}, \boldsymbol{v}|z)\boldsymbol{a}(\boldsymbol{x}, \boldsymbol{v}, t|z) q(z) \mathrm{d}z \right) \\
&= -\nabla_{\boldsymbol{x}} \cdot (\rho_t(\boldsymbol{x}, \boldsymbol{v})\boldsymbol{v}) - \nabla_{\boldsymbol{v}} \cdot (\boldsymbol{a}_t(\boldsymbol{x}, \boldsymbol{v})\rho_t(\boldsymbol{x}, \boldsymbol{v}))
\end{aligned} \tag{34}$$

Therefore, $\rho_t(\boldsymbol{x}, \boldsymbol{v})$ satisfies the continuity equation:

$$\partial_t \rho_t(\boldsymbol{x}, \boldsymbol{v}) = -\nabla_{\boldsymbol{x}} \cdot (\rho_t(\boldsymbol{x}, \boldsymbol{v})\boldsymbol{v}) - \nabla_{\boldsymbol{v}} \cdot (\boldsymbol{a}_t(\boldsymbol{x}, \boldsymbol{v})\rho_t(\boldsymbol{x}, \boldsymbol{v})) \tag{35}$$

This completes the proof. □

A.4 Proof of Theorem 4.5

Proof. Consider the gradient of the AFM Loss and CAFM Loss:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{AFM}} = \nabla_\theta \mathbb{E}_{\rho_t(\boldsymbol{x}, \boldsymbol{v})} (\|\boldsymbol{a}_\theta(\boldsymbol{x}, \boldsymbol{v}, t)\|^2 - 2\boldsymbol{a}_\theta^T(\boldsymbol{x}, \boldsymbol{v}, t)\boldsymbol{a}(\boldsymbol{x}, \boldsymbol{v}, t)) \tag{36}$$

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{CAFM}} = \nabla_\theta \mathbb{E}_{\rho_t(\boldsymbol{x}, \boldsymbol{v}|\boldsymbol{z}) q(\boldsymbol{z})} (\|\boldsymbol{a}_\theta(\boldsymbol{x}, \boldsymbol{v}, t)\|^2 - 2\boldsymbol{a}_\theta^T(\boldsymbol{x}, \boldsymbol{v}, t)\boldsymbol{a}(\boldsymbol{x}, \boldsymbol{v}, t|\boldsymbol{z})) \tag{37}$$

For the first term of $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{AFM}}$:

$$\begin{aligned}
\mathbb{E}_{\rho_t(\boldsymbol{x}, \boldsymbol{v})} \|\boldsymbol{a}_\theta(\boldsymbol{x}, \boldsymbol{v}, t)\|^2 &= \int \rho_t(\boldsymbol{x}, \boldsymbol{v}) \|\boldsymbol{a}_\theta(\boldsymbol{x}, \boldsymbol{v}, t)\|^2 \mathrm{d}\boldsymbol{x} \mathrm{d}\boldsymbol{v} \\
&= \int \rho_t(\boldsymbol{x}, \boldsymbol{v}|z) q(\boldsymbol{z}) \|\boldsymbol{a}_\theta(\boldsymbol{x}, \boldsymbol{v}, t)\|^2 \mathrm{d}\boldsymbol{x} \mathrm{d}\boldsymbol{v} \mathrm{d}z \\
&= \mathbb{E}_{\rho_t(\boldsymbol{x}, \boldsymbol{v}|\boldsymbol{z}) q(\boldsymbol{z})} \|\boldsymbol{a}_\theta(\boldsymbol{x}, \boldsymbol{v}, t)\|^2
\end{aligned} \tag{38}$$

For the second term:

$$\begin{aligned}\mathbb{E}_{\rho_t(\boldsymbol{x},\boldsymbol{v})}[\boldsymbol{a}_\theta^T(\boldsymbol{x},\boldsymbol{v},t)\boldsymbol{a}(\boldsymbol{x},\boldsymbol{v},t)] &= \int \rho_t(\boldsymbol{x},\boldsymbol{v})\boldsymbol{a}_\theta^T(\boldsymbol{x},\boldsymbol{v},t)\boldsymbol{a}(\boldsymbol{x},\boldsymbol{v},t)\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} \\ &= \int \rho_t(\boldsymbol{x},\boldsymbol{v})\boldsymbol{a}_\theta^T(\boldsymbol{x},\boldsymbol{v},t)\left(\int q(\boldsymbol{z})\frac{\boldsymbol{a}(\boldsymbol{x},\boldsymbol{v},t|\boldsymbol{z})\rho_t(\boldsymbol{x},\boldsymbol{v}|\boldsymbol{z})}{\rho_t(\boldsymbol{x},\boldsymbol{v})}\mathrm{d}z\right)\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} \\ &= \int\int \boldsymbol{a}_\theta^T(\boldsymbol{x},\boldsymbol{v},t)\boldsymbol{a}(\boldsymbol{x},\boldsymbol{v},t|\boldsymbol{z})\rho_t(\boldsymbol{x},\boldsymbol{v}|\boldsymbol{z})q(z)\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v}\mathrm{d}\boldsymbol{z} \\ &= \mathbb{E}_{\rho_t(\boldsymbol{x},\boldsymbol{v}|\boldsymbol{z})q(\boldsymbol{z})}(\boldsymbol{a}_\theta^T(\boldsymbol{x},\boldsymbol{v},t)\boldsymbol{a}(\boldsymbol{x},\boldsymbol{v},t|\boldsymbol{z})) \end{aligned} \tag{39}$$

□

A.5 Proof of Theorem 4.6

Proof. Consider a given time t_i. Based on the expressions for $\boldsymbol{m}_{\boldsymbol{x}}, \boldsymbol{m}_{\boldsymbol{v}}$, at this moment we exactly have:

$$\rho_{t_i}(\boldsymbol{x},\boldsymbol{v}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{x}_i,\sigma_x^2\boldsymbol{I})\cdot\mathcal{N}(\boldsymbol{v}|\boldsymbol{v}_i,\sigma_v^2\boldsymbol{I}) \tag{40}$$

For simplicity, let $\mathrm{d}\boldsymbol{s}_j = \mathrm{d}\boldsymbol{x}_j\mathrm{d}\boldsymbol{v}_j$Therefore, the Marginal Probability Path is:

$$\begin{aligned}\rho_{t_i}(\boldsymbol{x},\boldsymbol{v}) &= \int \rho_{t_i}(\boldsymbol{x},\boldsymbol{v}|\boldsymbol{z})q(\boldsymbol{z})\mathrm{d}\boldsymbol{z} \\ &= \int \rho_{t_i}(\boldsymbol{x},\boldsymbol{v}|(\boldsymbol{x}_i,\boldsymbol{v}_i))\mu_0(\boldsymbol{x}_0,\boldsymbol{v}_0)\cdot\prod_{j=0}^{K-1}\mathcal{K}^*_{j\to j+1}[(\boldsymbol{x}_j,\boldsymbol{v}_j),(\boldsymbol{x}_{j+1},\boldsymbol{v}_{j+1})]\mathrm{d}\boldsymbol{s}_0\cdots\mathrm{d}\boldsymbol{s}_K\end{aligned} \tag{41}$$

Using the definition of the propagator, we know that:

$$\int \mathcal{K}^*_{j\to j+1}[(\boldsymbol{x}_j,\boldsymbol{v}_j),(\boldsymbol{x}_{j+1},\boldsymbol{v}_{j+1})]\mathrm{d}\boldsymbol{x}_{j+1}\mathrm{d}\boldsymbol{v}_{j+1} = 1 \tag{42}$$

Thus, we can integrate out the future terms ($j > i$) first , and then iteratively collapse the past terms ($j < i$):

$$\begin{aligned}\rho_{t_i}(\boldsymbol{x},\boldsymbol{v}) &= \int \rho_{t_i}(\boldsymbol{x},\boldsymbol{v}|(\boldsymbol{x}_i,\boldsymbol{v}_i))\mu_0(\boldsymbol{x}_0,\boldsymbol{v}_0)\cdot\prod_{j=0}^{K-1}\mathcal{K}^*_{j\to j+1}[(\boldsymbol{x}_j,\boldsymbol{v}_j),(\boldsymbol{x}_{j+1},\boldsymbol{v}_{j+1})]\mathrm{d}\boldsymbol{s}_0\cdots\mathrm{d}\boldsymbol{s}_K \\ &= \int \rho_{t_i}(\boldsymbol{x},\boldsymbol{v}|(\boldsymbol{x}_i,\boldsymbol{v}_i))\mu_0(\boldsymbol{x}_0,\boldsymbol{v}_0)\cdot\prod_{j=0}^{i-1}\mathcal{K}^*_{j\to j+1}[(\boldsymbol{x}_j,\boldsymbol{v}_j),(\boldsymbol{x}_{j+1},\boldsymbol{v}_{j+1})]\mathrm{d}\boldsymbol{s}_0\cdots\mathrm{d}\boldsymbol{s}_i \\ &= \int \rho_{t_i}(\boldsymbol{x},\boldsymbol{v}|(\boldsymbol{x}_i,\boldsymbol{v}_i))\pi^*_{0\to 1}[(\boldsymbol{x}_0,\boldsymbol{v}_0),(\boldsymbol{x}_1,\boldsymbol{v}_1)]\prod_{j=1}^{i-1}\mathcal{K}^*_{j\to j+1}[(\boldsymbol{x}_j,\boldsymbol{v}_j),(\boldsymbol{x}_{j+1},\boldsymbol{v}_{j+1})]\mathrm{d}\boldsymbol{s}_0\cdots\mathrm{d}\boldsymbol{s}_i \\ &= \int \rho_{t_i}(\boldsymbol{x},\boldsymbol{v}|(\boldsymbol{x}_i,\boldsymbol{v}_i))\mu_1(\boldsymbol{x}_1,\boldsymbol{v}_1)\prod_{j=1}^{i-1}\mathcal{K}^*_{j\to j+1}[(\boldsymbol{x}_j,\boldsymbol{v}_j),(\boldsymbol{x}_{j+1},\boldsymbol{v}_{j+1})]\mathrm{d}\boldsymbol{s}_1\cdots\mathrm{d}\boldsymbol{s}_i \\ &\ \ \vdots \\ &= \int \rho_{t_i}(\boldsymbol{x},\boldsymbol{v}|(\boldsymbol{x}_i,\boldsymbol{v}_i))\mu_i(\boldsymbol{x}_i,\boldsymbol{v}_i)\mathrm{d}\boldsymbol{s}_i \\ &= \int (\mathcal{N}(\boldsymbol{x}|\boldsymbol{x}_i,\sigma_x^2\boldsymbol{I})\cdot\mathcal{N}(\boldsymbol{v}|\boldsymbol{v}_i,\sigma_v^2\boldsymbol{I}))\mu_i(\boldsymbol{x}_i,\boldsymbol{v}_i)\mathrm{d}\boldsymbol{s}_i \\ &= (\mu_i * g)(\boldsymbol{x},\boldsymbol{v})\end{aligned} \tag{43}$$

where $g(\boldsymbol{x},\boldsymbol{v}) = (\mathcal{N}(\boldsymbol{x}|\boldsymbol{0},\sigma_x^2\boldsymbol{I})\cdot\mathcal{N}(\boldsymbol{v}|\boldsymbol{0},\sigma_v^2\boldsymbol{I}))$. When $(\sigma_{\boldsymbol{x}},\sigma_{\boldsymbol{v}}) \to \boldsymbol{0}$, we have $\rho_{t_i}(\boldsymbol{x},\boldsymbol{v}) \to \mu_i(\boldsymbol{x},\boldsymbol{v})$ in distribution.

If $\mu_i(\boldsymbol{x}, \boldsymbol{v})$ is absolutely continuous with respect to the Lebesgue measure and admits a density in $L^1(\mathbb{R}^{2d})$, we can guarantee strong convergence in the L^1 norm: $\lim_{\sigma \to 0} \|\rho_{t_i}(\boldsymbol{x}, \boldsymbol{v}) - \mu_i(\boldsymbol{x}, \boldsymbol{v})\|_{L^1} = 0$ for all $i = 0, 1, ..., K$.

□

A.6 Proof of Proposition 4.7

First, we prove that the Marginal Probability Path constructed by TracingFlow converges to the optimal solution of the DOAT problem when $\sigma_{\boldsymbol{x}} \to 0, \sigma_{\boldsymbol{v}} \to 0$.

We investigate directly in the augmented space. Let $\boldsymbol{s} = [\boldsymbol{x}, \boldsymbol{v}]^T \in \mathcal{X} \times \mathcal{V}$, and consider the cost of the DOAT problem. The subsequent position of a particle starting from the initial point $\boldsymbol{s}_0$ can be represented by the flow map $\boldsymbol{\phi}_t(\boldsymbol{s}_0)$. Therefore, the cost can be written as:

$$\begin{aligned}
\mathcal{L}_{\text{DOAT}}(\boldsymbol{a}, \rho) &= \int_{t_0}^{t_k} \int_{\mathcal{X}\times\mathcal{V}} \frac{1}{2} \|\boldsymbol{a}(\boldsymbol{s}, t)\|^2 \rho_t(\boldsymbol{s}) \mathrm{d}\boldsymbol{x} \mathrm{d}\boldsymbol{v} \mathrm{d}t \\
&= \int_{t_0}^{t_k} \int_{\mathcal{X}\times\mathcal{V}} \frac{1}{2} \|\boldsymbol{a}(\boldsymbol{\phi}_t(\boldsymbol{s}_0), t)\|^2 \det \left(\frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{s}_0} \right)^{-1} \rho_0(\boldsymbol{s}_0) \det \left(\frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{s}_0} \right) \mathrm{d}\boldsymbol{s}_0 \mathrm{d}t \\
&= \int_{\mathcal{X}\times\mathcal{V}} \rho_0(\boldsymbol{s}_0) \mathrm{d}\boldsymbol{s}_0 \int_{t_0}^{t_k} \frac{1}{2} \|\boldsymbol{a}(\boldsymbol{\phi}_t(\boldsymbol{s}_0), t)\|^2 \mathrm{d}t \\
&= \int_{\Omega} \left(\int_{t_0}^{t_k} \frac{1}{2} \|\boldsymbol{\gamma}''(t)\|^2 \mathrm{d}t \right) \mathrm{d}\mathbb{P}[\boldsymbol{\gamma}]
\end{aligned} \tag{44}$$

where $\mathbb{P}$ is a probability measure in the path space, satisfying the constraint $(e_{t_i})_{\#}\mathbb{P} = \mu_i(\boldsymbol{x}, \boldsymbol{v})$. Here, e_t is the evaluation map $e_t[\boldsymbol{\gamma}] = \boldsymbol{\gamma}(t)$, and $(e_{t_i})_{\#}\mathbb{P}$ denotes the push-forward of the probability measure by the evaluation map. According to **??**, the minimum cost path connecting $\boldsymbol{s}_i$ and $\boldsymbol{s}_{i+1}$ is a cubic spline, and the minimum cost is $\mathcal{C}_{i\to i+1}[\boldsymbol{s}_i, \boldsymbol{s}_{i+1}]$. Therefore, for any path $\boldsymbol{\gamma}$:

$$\int_{t_i}^{t_{i+1}} \frac{1}{2} \|\boldsymbol{\gamma}''(t)\|^2 \mathrm{d}t \geq \mathcal{C}_{i\to i+1}[\boldsymbol{s}_i, \boldsymbol{s}_{i+1}] \tag{45}$$

Thus, the DOAT problem cost has a lower bound:

$$\begin{aligned}
\mathcal{L}_{\text{DOAT}}(\rho, \boldsymbol{a}) &= \int_{\Omega} \sum_{i=0}^{K-1} \left(\int_{t_i}^{t_{i+1}} \frac{1}{2} \|\boldsymbol{\gamma}''(t)\|^2 \mathrm{d}t \right) \mathrm{d}\mathbb{P}[\boldsymbol{\gamma}] \\
&\geq \int_{\Omega} \sum_{i=0}^{K-1} \mathcal{C}_{i\to i+1}[\boldsymbol{s}_i, \boldsymbol{s}_{i+1}] \mathrm{d}\mathbb{P}[\boldsymbol{\gamma}] \\
&\geq \int \sum_{i=0}^{K-1} \mathcal{C}_{i\to i+1}[\boldsymbol{s}_i, \boldsymbol{s}_{i+1}] \mathrm{d}\pi^*(\boldsymbol{s}_i, \boldsymbol{s}_{i+1})
\end{aligned} \tag{46}$$

where $\pi^*(\boldsymbol{s}_i, \boldsymbol{s}_{i+1})$ is the Optimal Transport Plan for the SOAT problem between $\mu(\boldsymbol{s}_i)$ and $\mu(\boldsymbol{s}_{i+1})$.

Consider the path constructed by TracingFlow. In the case where $\sigma_{\boldsymbol{x}}, \sigma_{\boldsymbol{v}} \to 0$, we have $\rho_t(\boldsymbol{x}, \boldsymbol{v}|\boldsymbol{z}) = \rho_t(\boldsymbol{s}|\boldsymbol{z}) \to \delta(\boldsymbol{s} - \boldsymbol{\gamma}^*_{\boldsymbol{z}}(t))$, where $\boldsymbol{\gamma}^*_{\boldsymbol{z}}$ is the optimal single-particle trajectory connecting $\boldsymbol{z} = [\boldsymbol{s}_0, \boldsymbol{s}_1, \cdots, \boldsymbol{s}_K]$. We calculate the transport cost of the constructed path:

$$\begin{aligned}
\mathcal{L}_{\text{TF}} &= \int q(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} \sum_{i=0}^{K-1} \int_{t_i}^{t_{i+1}} \|\boldsymbol{\gamma}''^*_{\boldsymbol{z}}(t)\|^2 \mathrm{d}t \\
&= \int q(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} \sum_{i=0}^{K-1} \mathcal{C}_{i\to i+1}[\boldsymbol{s}_i, \boldsymbol{s}_{i+1}] \\
&= \int \mu_0(\boldsymbol{s}_0) \prod_{i=0}^{K-1} \mathcal{K}^*_{i\to i+1}[\boldsymbol{s}_i, \boldsymbol{s}_{i+1}] \left(\sum_{j=0}^{K-1} \mathcal{C}_{j\to j+1}[\boldsymbol{s}_j, \boldsymbol{s}_{j+1}] \right) \mathrm{d}\boldsymbol{z}
\end{aligned} \tag{47}$$

Here, using the definition of the propagator:

$$\begin{aligned}\int \mu_0(\boldsymbol{s}_0) \prod_{i=0}^{K-1} \mathcal{K}^*_{i\to i+1}[\boldsymbol{s}_i, \boldsymbol{s}_{i+1}] \left(\mathcal{C}_{j\to j+1}[\boldsymbol{s}_j, \boldsymbol{s}_{j+1}]\right) \mathrm{d}\boldsymbol{z} \\ = \int \pi^*_{j\to j+1}[\boldsymbol{s}_j, \boldsymbol{s}_{j+1}] \mathcal{C}_{j\to j+1}[\boldsymbol{s}_j, \boldsymbol{s}_{j+1}] \mathrm{d}\boldsymbol{s}_j \mathrm{d}\boldsymbol{s}_{j+1}\end{aligned} \tag{48}$$

Therefore:

$$\mathcal{L}_{\text{TF}} = \sum_{i=0}^{K-1} \int \pi^*_{i\to i+1}[\boldsymbol{s}_i, \boldsymbol{s}_{i+1}] \mathcal{C}_{i\to i+1}[\boldsymbol{s}_i, \boldsymbol{s}_{i+1}] \mathrm{d}\boldsymbol{s}_i \mathrm{d}\boldsymbol{s}_{i+1} \tag{49}$$

This coincides exactly with the lower bound of the DOAT problem cost derived above. Thus, the path constructed by TracingFlow is exactly the optimal solution to the DOAT problem in the case where $\sigma_{\boldsymbol{x}}, \sigma_{\boldsymbol{v}} \to 0$.

Next, we prove that under the construction of TracingFlow, the acceleration field $\boldsymbol{a}(\boldsymbol{s}, t)$ is single-valued with respect to $\boldsymbol{s}$ and t. By Remark 4.3, under the Optimal Transport Plan, the trajectories of flow maps starting from different points must not intersect. This implies that for a given time t, any point $\boldsymbol{s}$ in the augmented space is traversed by at most one flow map trajectory. Therefore, the acceleration field $\boldsymbol{a}(\boldsymbol{s}, t)$ possesses single-valuedness. In other words, the solution constructed by TracingFlow can indeed be expressed by a single optimal acceleration field $\boldsymbol{a}(\boldsymbol{s}, t)$.

A.7 Proof of Proposition 4.8

Proof. Let $\gamma : [0, T] \to \mathbb{R}^d$ be a path that is C^1 on $[0, T]$ and piecewise C^2 on each interval $[t_{j-1}, t_j]$. The optimal solution is given by a piecewise cubic polynomial:

$$\begin{aligned}\gamma(t_{j-1} + \tau) = &\frac{(\boldsymbol{v}_{j-1} + \boldsymbol{v}_j)\Delta t_j - 2(\boldsymbol{x}_j - \boldsymbol{x}_{j-1})}{(\Delta t_j)^3}\tau^3 \\ &+ \frac{3(\boldsymbol{x}_j - \boldsymbol{x}_{j-1}) - (2\boldsymbol{v}_{j-1} + \boldsymbol{v}_j)\Delta t_j}{(\Delta t_j)^2}\tau^2 + \boldsymbol{v}_{j-1}\tau + \boldsymbol{x}_{j-1},\end{aligned} \tag{50}$$

for $\tau \in [0, \Delta t_j]$, where $\Delta t_j = t_j - t_{j-1}$,

According to Corollary 4.2, it suffices to minimize

$$\begin{aligned}\mathcal{J}(\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_K) = &2\sum_{j=1}^{K} \frac{1}{(\Delta t_j)^3} \Big\{ (\|\boldsymbol{v}_{j-1}\|^2 + \langle \boldsymbol{v}_{j-1}, \boldsymbol{v}_j \rangle + \|\boldsymbol{v}_j\|^2)(\Delta t_j)^2 \\ &+ 3\langle \boldsymbol{v}_{j-1} + \boldsymbol{v}_j, x_{j-1} - x_j \rangle \Delta t_j + 3\|x_{j-1} - x_j\|^2 \Big\}\end{aligned} \tag{51}$$

$\mathcal{J} \geq 0$ always holds, so the the formula above is a semi-positive quadratic form.

Take differentiation, we have

$$\nabla_{\boldsymbol{v}_j} \mathcal{J} = 2\left(\frac{\boldsymbol{v}_{j-1} + 2\boldsymbol{v}_j}{\Delta t_j} + \frac{2\boldsymbol{v}_j + \boldsymbol{v}_{j+1}}{\Delta t_{j+1}}\right) + 6\left(\frac{\boldsymbol{x}_{j-1} - \boldsymbol{x}_j}{(\Delta t_j)^2} + \frac{\boldsymbol{x}_j - \boldsymbol{x}_{j+1}}{(\Delta t_{j+1})^2}\right), \quad \text{for } j = 1, \ldots, K-1, \tag{52}$$

$$\nabla_{\boldsymbol{v}_0} \mathcal{J} = \frac{2(2\boldsymbol{v}_0 + \boldsymbol{v}_1)}{\Delta t_1} + \frac{6(\boldsymbol{x}_0 - \boldsymbol{x}_1)}{(\Delta t_1)^2}, \nabla_{\boldsymbol{v}_K} \mathcal{J} = \frac{2(\boldsymbol{v}_{K-1} + 2\boldsymbol{v}_K)}{\Delta t_K} + \frac{6(\boldsymbol{x}_{K-1} - \boldsymbol{x}_K)}{(\Delta t_K)^2}. \tag{53}$$

Set the gradients to zero and rearranging the terms to separate the velocities $\boldsymbol{v}$ and positions $\boldsymbol{x}$, we obtain:

For $j = 0$:

$$\frac{2}{\Delta t_1}\boldsymbol{v}_0 + \frac{1}{\Delta t_1}\boldsymbol{v}_1 = \frac{3(\boldsymbol{x}_1 - \boldsymbol{x}_0)}{(\Delta t_1)^2}. \tag{54}$$

For $j = 1, \ldots, K-1$:

$$\frac{1}{\Delta t_j}\boldsymbol{v}_{j-1} + 2\left(\frac{1}{\Delta t_j} + \frac{1}{\Delta t_{j+1}}\right)\boldsymbol{v}_j + \frac{1}{\Delta t_{j+1}}\boldsymbol{v}_{j+1} = 3\left(\frac{\boldsymbol{x}_j - \boldsymbol{x}_{j-1}}{(\Delta t_j)^2} + \frac{\boldsymbol{x}_{j+1} - \boldsymbol{x}_j}{(\Delta t_{j+1})^2}\right). \tag{55}$$

For $j = K$:

$$\frac{1}{\Delta t_K}\boldsymbol{v}_{K-1} + \frac{2}{\Delta t_K}\boldsymbol{v}_K = \frac{3(\boldsymbol{x}_K - \boldsymbol{x}_{K-1})}{(\Delta t_K)^2}. \tag{56}$$

Denote $V = [\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_K] \in \mathbb{R}^{d\times(K+1)}$ and writing this system in matrix form $VA = B$, where

$$A = \begin{bmatrix} \frac{2}{\Delta t_1} & \frac{1}{\Delta t_1} & 0 & \cdots & 0 & 0 \\ \frac{1}{\Delta t_1} & 2\left(\frac{1}{\Delta t_1} + \frac{1}{\Delta t_2}\right) & \frac{1}{\Delta t_2} & \cdots & 0 & 0 \\ 0 & \frac{1}{\Delta t_2} & 2\left(\frac{1}{\Delta t_2} + \frac{1}{\Delta t_3}\right) & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \frac{1}{\Delta t_{K-1}} & 0 \\ 0 & 0 & \cdots & \frac{1}{\Delta t_{K-1}} & 2\left(\frac{1}{\Delta t_{K-1}} + \frac{1}{\Delta t_K}\right) & \frac{1}{\Delta t_K} \\ 0 & 0 & \cdots & 0 & \frac{1}{\Delta t_K} & \frac{2}{\Delta t_K} \end{bmatrix} \in \mathbb{R}^{(K+1)\times(K+1)} \tag{57}$$

$$\begin{aligned} B = \Bigg[&\frac{3(\boldsymbol{x}_1 - \boldsymbol{x}_0)}{(\Delta t_1)^2}, 3\left(\frac{\boldsymbol{x}_1 - \boldsymbol{x}_0}{(\Delta t_1)^2} + \frac{\boldsymbol{x}_2 - \boldsymbol{x}_1}{(\Delta t_2)^2}\right), \ldots, \\ &3\left(\frac{\boldsymbol{x}_{K-1} - \boldsymbol{x}_{K-2}}{(\Delta t_{K-1})^2} + \frac{\boldsymbol{x}_K - \boldsymbol{x}_{K-1}}{(\Delta t_K)^2}\right), \frac{3(\boldsymbol{x}_K - \boldsymbol{x}_{K-1})}{(\Delta t_K)^2}\Bigg] \in \mathbb{R}^{d\times(K+1)} \end{aligned} \tag{58}$$

The solution satisfies $\gamma'(t_j) = \boldsymbol{v}_j$ for all $j = 0, 1, ...K$.

Since A is strictly diagonally dominant, the solution V is unique. □

A.8 Proof of Theorem A.1

Theorem A.1. *Let $\boldsymbol{a}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{v}, t)$ be L_θ-Lipschitz continuous of $(\boldsymbol{x}, \boldsymbol{v})$. The Wasserstein-2 distance between the generated distribution $\hat{\rho}_{t_i}^{(pos)}(\boldsymbol{x}) = \int \hat{\rho}_{t_i}(\boldsymbol{x}, \boldsymbol{v})\mathrm{d}\boldsymbol{v}$ and the true distribution $\mu_i^{(pos)}(\boldsymbol{x})$ in the position space is bounded by*

$$\mathcal{W}_2^2(\hat{\rho}_{t_i}^{(pos)}, \mu_i^{(pos)}) \leq 2e^{\kappa(t_i - t_0)}\left(\mathcal{L}_{v_0} + (t_i - t_0)\mathcal{L}_{AFM}|_{[t_0,t_i]}\right) \tag{59}$$

where the cumulative AFM loss and the initial velocity loss are defined as:

$$\mathcal{L}_{AFM}|_{[t_0,t_i]} = \mathbb{E}_{t,(\boldsymbol{x},\boldsymbol{v})\sim\rho_t}\|\boldsymbol{a}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{v}, t) - \boldsymbol{a}(\boldsymbol{x}, \boldsymbol{v}, t)\|^2, \mathcal{L}_{v_0} = \mathbb{E}_{(\boldsymbol{x}_0,\boldsymbol{v}_0)\sim\rho_{t_0}}\|\boldsymbol{v}_{0,\boldsymbol{\xi}}(\boldsymbol{x}_0) - \boldsymbol{v}_0\|^2.$$

and κ is a constant only depending on L_θ.

Proof. We address the problem directly within the augmented space $\mathcal{X} \times \mathcal{V}$. Let the system state be denoted by $\boldsymbol{s} = [\boldsymbol{x}, \boldsymbol{v}]^T$, and let $\mu_i(\boldsymbol{s})$ represent the data distribution in this space. We define a flow map $\boldsymbol{\phi}_t : [t_0, t_K] \times \mathcal{X} \times \mathcal{V} \to \mathcal{X} \times \mathcal{V}$, which is governed by the exact marginal acceleration field $\boldsymbol{a}(\boldsymbol{s}, t)$ defined in Equation (12).

Using $(\boldsymbol{\phi}_t)_\#$ to denote the push-forward operator, the time-dependent density evolves as $\rho_t = (\boldsymbol{\phi}_t)_\#\rho_{t_0}$, where $\rho_{t_0} = \mu_0$. Similarly, let $\boldsymbol{\phi}_t^{\boldsymbol{\theta}}$ be the flow map induced by the learned acceleration field $\boldsymbol{a}_{\boldsymbol{\theta}}(\boldsymbol{s}, t)$. The generated distribution $\hat{\rho}_t$ then evolves according to $\hat{\rho}_t = (\boldsymbol{\phi}_t^{\boldsymbol{\theta}})_\#\hat{\rho}_{t_0}$. It is important to note that $\hat{\rho}_{t_0} \neq \rho_{t_0}$, as the conditional distribution of the initial velocity is approximated via a neural network.

We introduce a bridging measure $\nu_t = (\boldsymbol{\phi}_t^\theta)_\#\rho_{t_0}$, then the Wasserstein-2 distance between joint distributions can be decomposed as

$$\mathcal{W}_2^2(\rho_t, \hat{\rho}_t) \leq \left(\mathcal{W}_2(\rho_t, \nu_t) + \mathcal{W}_2(\nu_t, \hat{\rho}_t)\right)^2 \leq 2\left(\mathcal{W}_2^2(\rho_t, \nu_t) + \mathcal{W}_2^2(\nu_t, \hat{\rho}_t)\right) \tag{60}$$

Estimation of Term 1: Define $Q_t = \int \mu_0(\boldsymbol{s}_0)\|\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0) - \boldsymbol{\phi}_t(\boldsymbol{s}_0)\|^2 \mathrm{d}\boldsymbol{s}_0$,here $\mu_0 = \rho_{t_0}$ according to definition. Noting that $Q_{t_0} = 0$ since the starting points are identical. Define $A = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}$.

Taking the time derivative, we have:

$$\begin{aligned}
\frac{\mathrm{d}Q_t}{\mathrm{d}t} &= 2\int \mu_0(\boldsymbol{s}_0)(\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0) - \boldsymbol{\phi}_t(\boldsymbol{s}_0))^T \left(\frac{\mathrm{d}\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0)}{\mathrm{d}t} - \frac{\mathrm{d}\boldsymbol{\phi}_t(\boldsymbol{s}_0)}{\mathrm{d}t}\right) \mathrm{d}\boldsymbol{s}_0 \\
&= 2\int \mu_0(\boldsymbol{s}_0)(\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0) - \boldsymbol{\phi}_t(\boldsymbol{s}_0))^T \left(A\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0) - A\boldsymbol{\phi}_t(\boldsymbol{s}_0) + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{a}_\theta(\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0), t) \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \boldsymbol{a}(\boldsymbol{\phi}_t(\boldsymbol{s}_0), t) \end{bmatrix}\right) \mathrm{d}\boldsymbol{s}_0 \\
&= 2\int \mu_0(\boldsymbol{s}_0) \Bigg[(\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0) - \boldsymbol{\phi}_t(\boldsymbol{s}_0))^T A(\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0) - \boldsymbol{\phi}_t(\boldsymbol{s}_0)) \\
&\quad + (\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0) - \boldsymbol{\phi}_t(\boldsymbol{s}_0))^T \begin{bmatrix} \mathbf{0} \\ \boldsymbol{a}_\theta(\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0), t) - \boldsymbol{a}_\theta(\boldsymbol{\phi}_t(\boldsymbol{s}_0), t) \end{bmatrix} \\
&\quad + (\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0) - \boldsymbol{\phi}_t(\boldsymbol{s}_0))^T \begin{bmatrix} \mathbf{0} \\ \boldsymbol{a}_\theta(\boldsymbol{\phi}_t(\boldsymbol{s}_0), t) - \boldsymbol{a}(\boldsymbol{\phi}_t(\boldsymbol{s}_0), t) \end{bmatrix} \Bigg] \mathrm{d}\boldsymbol{s}_0 \qquad (61)
\end{aligned}$$

We bound the three terms inside the integral separately. Let $\Delta\boldsymbol{\phi}_t(\boldsymbol{s}_0) = \boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0) - \boldsymbol{\phi}_t(\boldsymbol{s}_0)$.

1. Since $\|A\| \leq 1$, we have $\Delta\boldsymbol{\phi}_t(\boldsymbol{s}_0)^T A \Delta\boldsymbol{\phi}_t(\boldsymbol{s}_0) \leq \|\Delta\boldsymbol{\phi}_t(\boldsymbol{s}_0)\|^2$.
2. Using the Lipschitz continuity of the network $\boldsymbol{a}_\theta$ with constant L_θ:
$$\begin{aligned}
\Delta\boldsymbol{\phi}_t(\boldsymbol{s}_0)^T \begin{bmatrix} \mathbf{0} \\ \boldsymbol{a}_\theta(\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0), t) - \boldsymbol{a}_\theta(\boldsymbol{\phi}_t(\boldsymbol{s}_0), t) \end{bmatrix} &\leq \|\Delta\boldsymbol{\phi}_t(\boldsymbol{s}_0)\| \cdot \|\boldsymbol{a}_\theta(\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0), t) - \boldsymbol{a}_\theta(\boldsymbol{\phi}_t(\boldsymbol{s}_0), t)\| \\
&\leq L_\theta \|\Delta\boldsymbol{\phi}_t(\boldsymbol{s}_0)\|^2. \qquad (62)
\end{aligned}$$
3. Using Cauchy-Schwarz and the inequality $2xy \leq x^2 + y^2$:
$$\begin{aligned}
\Delta\boldsymbol{\phi}_t(\boldsymbol{s}_0)^T \begin{bmatrix} \mathbf{0} \\ \boldsymbol{a}_\theta(\boldsymbol{\phi}_t(\boldsymbol{s}_0), t) - \boldsymbol{a}(\boldsymbol{\phi}_t(\boldsymbol{s}_0), t) \end{bmatrix} &\leq \|\Delta\boldsymbol{\phi}_t(\boldsymbol{s}_0)\| \cdot \|\boldsymbol{a}_\theta(\boldsymbol{\phi}_t(\boldsymbol{s}_0), t) - \boldsymbol{a}(\boldsymbol{\phi}_t(\boldsymbol{s}_0), t)\| \\
&\leq \frac{1}{2}\|\Delta\boldsymbol{\phi}_t(\boldsymbol{s}_0)\|^2 + \frac{1}{2}\|\boldsymbol{a}_\theta(\boldsymbol{\phi}_t(\boldsymbol{s}_0), t) - \boldsymbol{a}(\boldsymbol{\phi}_t(\boldsymbol{s}_0), t)\|^2. \qquad (63)
\end{aligned}$$

Substituting these bounds back into the derivative:

$$\begin{aligned}
\frac{\mathrm{d}Q_t}{\mathrm{d}t} &\leq 2\int \mu_0(\boldsymbol{s}_0) \left[(1 + L_\theta + \frac{1}{2})\|\Delta\boldsymbol{\phi}_t(\boldsymbol{s}_0)\|^2 + \frac{1}{2}\|\boldsymbol{a}_\theta(\boldsymbol{\phi}_t(\boldsymbol{s}_0), t) - \boldsymbol{a}(\boldsymbol{\phi}_t(\boldsymbol{s}_0), t)\|^2 \right] \mathrm{d}\boldsymbol{s}_0 \\
&= (2L_\theta + 3)Q_t + \int \mu_0(\boldsymbol{s}_0)\|\boldsymbol{a}_\theta(\boldsymbol{\phi}_t(\boldsymbol{s}_0), t) - \boldsymbol{a}(\boldsymbol{\phi}_t(\boldsymbol{s}_0), t)\|^2 \mathrm{d}\boldsymbol{s}_0 \qquad (64)
\end{aligned}$$

By Gronwall's inequality, since $Q_{t_0} = 0$:

$$Q_t \leq e^{(2L_\theta+3)(t-t_0)} \int_{t_0}^t \left(\int \mu_0(\boldsymbol{s}_0)\|\boldsymbol{a}_\theta(\boldsymbol{\phi}_\tau(\boldsymbol{s}_0), \tau) - \boldsymbol{a}(\boldsymbol{\phi}_\tau(\boldsymbol{s}_0), \tau)\|^2 \mathrm{d}\boldsymbol{s}_0 \right) \mathrm{d}\tau \qquad (65)$$

By the definition of Wasserstein-2 distance, we have:

$$\begin{aligned}
t_0 \mathcal{W}_2^2(\rho_t, \nu_t) &= \min_\pi \int \|\boldsymbol{s}_1 - \boldsymbol{s}_2\|^2 \mathrm{d}\pi \quad \text{s.t.} \int \pi \mathrm{d}\boldsymbol{s}_1 = \rho_t, \int \pi \mathrm{d}\boldsymbol{s}_2 = \nu_t \\
&\leq \int \mu_0(\boldsymbol{s}_0)\|\boldsymbol{\phi}_t^{\boldsymbol{\theta}}(\boldsymbol{s}_0) - \boldsymbol{\phi}_t(\boldsymbol{s}_0)\|^2 \mathrm{d}\boldsymbol{s}_0 \\
&= Q_t \leq e^{(2L_\theta+3)(t-t_0)} \int_{t_0}^t \mathbb{E}_{\boldsymbol{s}\sim\mu_\tau}\|\boldsymbol{a}_\theta(\boldsymbol{s}, \tau) - \boldsymbol{a}(\boldsymbol{s}, \tau)\|^2 \mathrm{d}\tau \qquad (66) \\
&= e^{(2L_\theta+3)(t-t_0)}(t - t_0)\mathcal{L}_{\text{AFM}}|_{[t_0,t]}. \qquad (67)
\end{aligned}$$

Estimation of Term 2:Take $\pi_0(\boldsymbol{s}_0, \boldsymbol{s}_0^\xi)$ as the optimal coupling that attains $\mathcal{W}_2^2(\nu_{t_0}, \hat{\rho}_{t_0}) = \int \|\boldsymbol{s}_0 - \boldsymbol{s}_0^\xi\|^2 \mathrm{d}\pi_0(\boldsymbol{s_0}, \boldsymbol{s}_0^\xi)$. Note that $\nu_{t_0} = \mu_0$ (the true data distribution).

From a pair of starting points $(\boldsymbol{s}_0, \boldsymbol{s}_0^\xi) \sim \pi_0$, define the square distance between trajectories under the same approximate flow as $\Delta(t) = \|\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0) - \boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0^\xi)\|^2$. Taking the time derivative, we have:

$$\begin{aligned}
\frac{\mathrm{d}\Delta(t)}{\mathrm{d}t} &= 2(\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0) - \boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0^\xi))^T \left(\frac{\mathrm{d}\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0)}{\mathrm{d}t} - \frac{\mathrm{d}\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0^\xi)}{\mathrm{d}t} \right) \\
&= 2(\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0) - \boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0^\xi))^T \left(A(\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0) - \boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0^\xi)) + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{a}_\theta(\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0), t) - \boldsymbol{a}_\theta(\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0^\xi), t) \end{bmatrix} \right) \\
&\leq 2 \left(\|A\| \cdot \|\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0) - \boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0^\xi)\|^2 + \|\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0) - \boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0^\xi)\| \cdot \|\boldsymbol{a}_\theta(\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0), t) - \boldsymbol{a}_\theta(\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0^\xi), t)\| \right) \\
&\leq 2(1 + L_\theta)\Delta(t). \qquad (68)
\end{aligned}$$

Again by Gronwall's inequality, we have:

$$\Delta(t) \leq e^{2(1+L_\theta)(t-t_0)} \Delta(t_0) = e^{2(1+L_\theta)(t-t_0)} \|\boldsymbol{s}_0 - \boldsymbol{s}_0^\xi\|^2. \qquad (69)$$

Since $\nu_t = (\boldsymbol{\phi}_t^\theta)_\# \nu_{t_0}$ and $\hat{\rho}_{t_0} = (\boldsymbol{\phi}_t^\theta)_\# \hat{\rho}_{t_0}$, the push-forward of the optimal coupling $\pi_t = (\boldsymbol{\phi}_t^\theta, \boldsymbol{\phi}_t^\theta)_\# \pi_0$ is a valid coupling for $(\nu_t, \hat{\rho}_t)$. Integrate the squared distance over all starting pairs with respect to π_0, we obtain:

$$\begin{aligned}
\mathcal{W}_2^2(\nu_t, \hat{\rho}_t) &\leq \int \|\boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0) - \boldsymbol{\phi}_t^\theta(\boldsymbol{s}_0^\xi)\|^2 \mathrm{d}\pi_0(\boldsymbol{s}_0, \boldsymbol{s}_0^\xi) \\
&\leq \int e^{2(1+L_\theta)(t-t_0)} \|\boldsymbol{s}_0 - \boldsymbol{s}_0^\xi\|^2 \mathrm{d}\pi_0(\boldsymbol{s}_0, \boldsymbol{s}_0^\xi) \\
&= e^{2(1+L_\theta)(t-t_0)} \mathcal{W}_2^2(\nu_{t_0}, \hat{\rho}_{t_0}) \qquad (70)
\end{aligned}$$

Let $\tilde{\pi}(\boldsymbol{s_0}, \boldsymbol{s}_0^\xi)$ be the coupling between ν_0 and $\hat{\rho}_0$ induced by the identity mapping on the position space, i.e., pairing $\boldsymbol{s}_0 = (\boldsymbol{x}_0, \boldsymbol{v}_0)$ with $\boldsymbol{s}_0^\xi = (\boldsymbol{x}_0, \boldsymbol{v}_0^\xi)$. We can bound Wasserstein distance by L_2 loss:

$$\begin{aligned}
\mathcal{W}_2^2(\nu_{t_0}, \hat{\rho}_{t_0}) &\leq \int \|\boldsymbol{s}_0 - \boldsymbol{s}_0^\xi\|^2 \mathrm{d}\tilde{\pi}(\boldsymbol{s}_0, \boldsymbol{s}_0^\xi) \\
&= \int \left(\|\boldsymbol{x}_0 - \boldsymbol{x}_0^\xi\|^2 + \|\boldsymbol{v}_0 - \boldsymbol{v}_0^\xi\|^2 \right) \mathrm{d}\tilde{\pi}(\boldsymbol{x}_0, \boldsymbol{v}_0, \boldsymbol{x}_0^\xi, \boldsymbol{v}_0^\xi) \\
&= \int \|\boldsymbol{v}_0 - \boldsymbol{v}_0^\xi\|^2 \mathrm{d}\tilde{\pi}(\boldsymbol{x}_0, \boldsymbol{v}_0, \boldsymbol{x}_0^\xi, \boldsymbol{v}_0^\xi) = \mathcal{L}_{v_0} \qquad (71)
\end{aligned}$$

Thus $\mathcal{W}_2^2(\nu_t, \hat{\rho}_t) \leq e^{2(1+L_\theta)(t-t_0)} \mathcal{L}_{v_0}$.

Combining the bounds above, we have

$$\begin{aligned}
\mathcal{W}_2^2(\hat{\rho}_{t_i}, \rho_{t_i}) &\leq 2 \left[\mathcal{W}_2^2(\hat{\rho}_{t_i}, \nu_{t_i}) + \mathcal{W}_2^2(\nu_{t_i}, \rho_{t_i}) \right] \\
&\leq 2 \left[e^{(2L_\theta+3)(t_i-t_0)} \cdot (t_i - t_0) \mathcal{L}_{\text{AFM}}|_{[t_0,t_i]} + e^{(2L_\theta+2)(t_i-t_0)} \mathcal{L}_{v_0} \right]. \qquad (72)
\end{aligned}$$

Define $\kappa = (2L_\theta + 3)$,then the final bound is obtained by:

$$\begin{aligned}
\mathcal{W}_2^2(\hat{\rho}_{t_i}^{(\text{pos})}, \rho_{t_i}^{(\text{pos})} = \mu_i^{(\text{pos})}) &= \min_{\pi_{\boldsymbol{x}}} \int \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2 \mathrm{d}\pi_{\boldsymbol{x}}(\boldsymbol{x}_1, \boldsymbol{x}_2) \quad \text{s.t.} \int \pi_{\boldsymbol{x}} \mathrm{d}\boldsymbol{x}_2 = \rho_{t_i}^{(\text{pos})}, \quad \int \pi_{\boldsymbol{x}} \mathrm{d}\boldsymbol{x}_1 = \hat{\rho}_{t_i}^{(\text{pos})} \\
&\leq \min_{\pi} \int \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2 \mathrm{d}\pi(\boldsymbol{s}_1, \boldsymbol{s}_2) \quad \text{s.t.} \int \pi \mathrm{d}\boldsymbol{s}_2 = \rho_{t_i}, \quad \int \pi \mathrm{d}\boldsymbol{s}_1 = \hat{\rho}_{t_i} \\
&\leq \min_{\pi} \int \|\boldsymbol{s}_1 - \boldsymbol{s}_2\|^2 \mathrm{d}\pi(\boldsymbol{s}_1, \boldsymbol{s}_2) \qquad (73) \\
&= \mathcal{W}_2^2(\hat{\rho}_{t_i}, \rho_{t_i}) \leq 2e^{\kappa(t_i-t_0)} \left((t_i - t_0) \mathcal{L}_{\text{AFM}}|_{[t_0,t_i]} + \mathcal{L}_{v_0} \right). \qquad (74)
\end{aligned}$$

This completes the proof. Note that our derivation holds even for multi-valued or stochastic cases, since the upper bound on the Wasserstein distance is established via a valid coupling, which remains well-defined regardless of whether the mapping is deterministic.

□

B Datasets and Evaluation Metric

B.1 Experiment Setup

The experiments were conducted on a shared high-performance computing (HPC) cluster. All experimental runs utilized GPUs equipped with 80GB of memory. Both the network $\boldsymbol{a}_\theta(\boldsymbol{x}, \boldsymbol{v}, t)$, employed to approximate acceleration, and the network $\boldsymbol{v}_\phi(\boldsymbol{x}, \boldsymbol{v}, t)$, employed to approximate velocity, utilize residual architectures. Specifically, $\boldsymbol{a}_\theta(\boldsymbol{x}, \boldsymbol{v}, t)$ consists of 4 hidden layers, while $\boldsymbol{v}_\phi(\boldsymbol{x}, \boldsymbol{v}, t)$ comprises 2 hidden layers. The Optimal Transport (OT) calculations were implemented using the Python Optimal Transport (POT) library [16], and the training process of neural networks were implemented using Pytorch [43].

In all experiments, TracingFlow employs a completely consistent set of hyperparameters. When solving for OT using the Sinkhorn algorithm, the regularization coefficient is set to $\epsilon = 1 \times 10^{-3}$. When incorporating biological priors, the penalty coefficient for barcode mismatches is set to $p_0 = 25$. Since the number of cells at each time point in the datasets processed in this paper is on the order of 10^3, we do not utilize mini-batch OT; instead, we compute the Transport Plan on the full dataset.

B.2 Evaluation Metrics

We employ the 1-Wasserstein Distance ($\mathcal{W}_1$) and the 2-Wasserstein Distance ($\mathcal{W}_2$) to quantify the discrepancy between the predicted distribution and the ground truth distribution. These metrics are defined as follows:

$$\mathcal{W}_1(\mu, \nu) = \min_{\pi \in \Pi(\mu,\nu)} \int \|\boldsymbol{x} - \boldsymbol{y}\| \mathrm{d}\pi(\boldsymbol{x}, \boldsymbol{y}) \tag{75}$$

$$\mathcal{W}_2(\mu, \nu) = \left(\min_{\pi \in \Pi(\mu,\nu)} \int \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 \mathrm{d}\pi(\boldsymbol{x}, \boldsymbol{y}) \right)^{1/2} \tag{76}$$

For the evaluation of datasets containing lineage information, we utilize the lineage-weighted $\mathcal{W}_1$ and $\mathcal{W}_2$ distances. These are defined based on the decomposition of the distributions. Consider the distribution generated by the model, denoted as $\mu(\boldsymbol{x})$, and the reference distribution provided by the dataset, denoted as $\nu(\boldsymbol{x})$. Both consist of L lineages and can be decomposed as:

$$\mu(\boldsymbol{x}) = \sum_{i=1}^{L} w_i \mu^{(i)}(\boldsymbol{x}), \quad \nu(\boldsymbol{x}) = \sum_{i=1}^{L} v_i \nu^{(i)}(\boldsymbol{x}) \tag{77}$$

where $\mu^{(i)}(\boldsymbol{x})$ and $\nu^{(i)}(\boldsymbol{x})$ represent the distributions of the i-th lineage within $\mu(\boldsymbol{x})$ and $\nu(\boldsymbol{x})$, respectively, satisfying $\int \mu^{(i)}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = 1$ and $\int \nu^{(i)}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = 1$. The terms w_i and v_i denote the weights of the i-th lineage in $\mu(\boldsymbol{x})$ and $\nu(\boldsymbol{x})$, respectively, subject to the constraint $\sum_{i=1}^{L} w_i = \sum_{i=1}^{L} v_i = 1$. The lineage-weighted $\mathcal{W}_1$ and $\mathcal{W}_2$ distances are defined as:

$$\mathcal{W}_1^{(L)}(\mu \| \nu) = \sum_{i=1}^{L} v_i \mathcal{W}_1(\mu^{(i)}, \nu^{(i)}), \quad \mathcal{W}_2^{(L)}(\mu \| \nu) = \sum_{i=1}^{L} v_i \mathcal{W}_2(\mu^{(i)}, \nu^{(i)}) \tag{78}$$

In essence, we first calculate the $\mathcal{W}_1$ and $\mathcal{W}_2$ distances between the distributions of corresponding lineages in the two datasets. Subsequently, we compute a weighted average of these distances using the proportions of each lineage given by the reference distribution $\nu(\boldsymbol{x})$. It is important to note that the lineage-weighted $\mathcal{W}_1$ and $\mathcal{W}_2$ metrics serve solely as indicators of the model's ability to preserve lineage prior information while learning the dynamics. They are not true distance metrics in the mathematical sense, as they are evidently asymmetric with respect to μ and ν.

B.3 Datasets

3D Simulation Lineage Data. Consider the following dynamics with three lineage barcodes and a twist structure:

$$\begin{cases} dX_t^{(0)} = v\,dt + \sigma_x dB_t^X \\ dY_t^{(0)} = \sigma_y dB_t^Y \\ dZ_t^{(0)} = \sigma_z dB_t^Z \end{cases} \quad \begin{cases} dX_t^{(1)} = v\,dt + \sigma_x dB_t^X \\ dY_t^{(1)} = -\mu Z_t^{(1)}\,dt + \sigma_y dB_t^Y \\ dZ_t^{(1)} = \mu Y_t^{(1)}\,dt + \sigma_z dB_t^Z \end{cases} \quad \begin{cases} dX_t^{(2)} = v\,dt + \sigma_x dB_t^X \\ dY_t^{(2)} = \mu Z_t^{(2)}\,dt + \sigma_y dB_t^Y \\ dZ_t^{(2)} = -\mu Y_t^{(2)}\,dt + \sigma_z dB_t^Z \end{cases} \tag{79}$$

We take $v = 3.0, \mu = 3.14, \sigma_x = \sigma_y = \sigma_z = 0.1$ and time points $t = 0, 1, 2$. Initialize $(X_0^{(0)}, Y_0^{(0)}, Z_0^{(0)}) \sim \mathcal{N}([0,0,0], 0.5I), (X_0^{(1)}, Y_0^{(1)}, Z_0^{(1)}) \sim \mathcal{N}([0,-2,0], 0.5I), (X_0^{(2)}, Y_0^{(2)}, Z_0^{(2)}) \sim \mathcal{N}([0,2,0], 0.5I)$ and sample 500 particles for each.

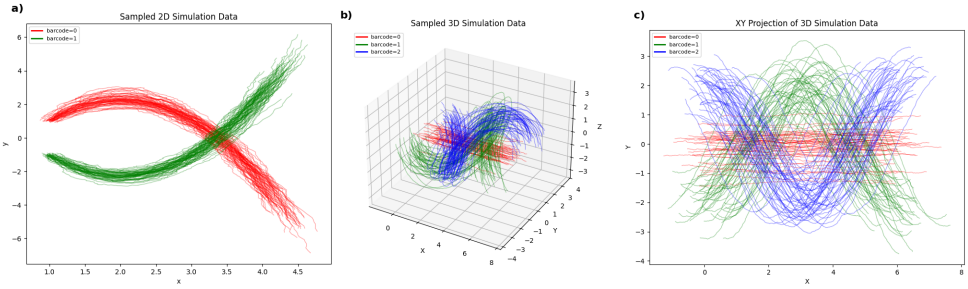


Figure 6: Dynamics of simulation datasets. (a)2D simulation data, (b)3D simulation data and (c)3D simulation data projected to XY plane.

Cite Data. We utilized the Cite-seq dataset introduced in [31], which comprises 31,240 cells collected across 4 distinct time points. To evaluate the performance of TracingFlow on real-world biological data, we applied Principal Component Analysis (PCA) to reduce the dimensionality of the gene expression profiles to 100 and 5 dimensions, respectively.

Embryoid Bodies Data. We employed the Embryoid Bodies (EB) dataset from [39], consisting of 16,819 cells sampled at 5 time points. We utilized PCA to reduce the dimensionality of the gene expression data to 5 dimensions.

Hematopoiesis Data. We employed the Hematopoiesis dataset from [66], consisting of 130,887 cells sampled at 3 time points. Part of the cells are recorded with barcodes. We selected 22,329 cells with barcodes from the dataset and utilized PCA to reduce the dimensionality of gene expression data to 50 dimensions. When processing this dataset, we filtered for data where barcodes were present at all three time points. The barcodes for the remaining data were set to undefined, incurring no additional transport cost during the incorporation of biological priors (Section 5.3).

Mexico Gulf Data To assess the capability and robustness of Tracing Flow in capturing long-term system dynamics, we utilize the Mexico Gulf dataset, following [57]. This two-dimensional dataset comprises 9 time points.

C Experiment Details

C.1 Experiment Details across various datasets

Simulation 2D Data Table 4 presents the performance of TracingFlow and other algorithms on the 2D Simulation dataset. On average, TracingFlow reconstructs the data distribution at each time point with higher accuracy.

Cite 5D & Cite 100D Data We present the distribution reconstruction performance of each algorithm on the Cite 5D and Cite 100D datasets in Table 5 and Table 6, respectively. On these datasets

Table 4: $\mathcal{W}_1$ and $\mathcal{W}_2$ distances between the generated and ground-truth distributions at each time point on the 2D Simulation dataset.

Method	Dynamics	$t=1$		$t=2$		$t=3$		$t=4$	
		$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$
OT-CFM	1st Order	**0.1215** ± 0.0164	**0.1441** ± 0.0164	0.3644 ± 0.1177	0.4143 ± 0.1168	0.5863 ± 0.2615	0.6238 ± 0.2657	1.0356 ± 0.2325	1.1867 ± 0.2890
SF^2M	1st Order	0.2389 ± 0.1217	0.2668 ± 0.1305	0.5700 ± 0.2574	0.6108 ± 0.2654	1.0630 ± 0.2769	1.3557 ± 0.2770	2.0986 ± 1.3198	2.2938 ± 1.3193
3MSBM	2nd Order	1.4029 ± 0.6892	1.4459 ± 0.6596	0.9318 ± 0.2553	1.0181 ± 0.2050	1.8079 ± 0.7761	1.9162 ± 0.7529	**0.5993** ± 0.6176	**0.7265** ± 0.6935
MMFM	2nd Order	1.3226 ± 0.1774	1.9287 ± 0.3002	1.8387 ± 0.5624	1.1779 ± 0.5114	1.8158 ± 0.1794	1.8740 ± 0.1701	4.4984 ± 0.2937	4.6059 ± 0.2977
HRF	2nd Order	1.2036 ± 0.0425	1.2096 ± 0.0244	0.9955 ± 0.0921	1.0009 ± 0.0873	1.9829 ± 0.1311	2.0040 ± 0.1582	1.9740 ± 0.1873	2.4379 ± 0.2157
CAF	2nd Order	0.4051 ± 0.1712	0.5117 ± 0.1738	0.8857 ± 0.3630	0.8996 ± 0.3689	1.3274 ± 0.1194	1.3731 ± 0.1107	3.9358 ± 0.4572	3.7880 ± 0.4530
TF(Ours)	2nd Order	0.1609 ± 0.0232	0.1787 ± 0.0258	**0.1753** ± 0.0299	**0.2045** ± 0.0277	**0.4091** ± 0.0523	**0.4821** ± 0.0698	0.6404 ± 0.2661	1.1436 ± 0.1577

TracingFlow achieves superior reconstruction accuracy at all time steps except for $t = 3$. The trajectories learned by TracingFlow on these two datasets are illustrated in Figure 7.

Table 5: $\mathcal{W}_1$ and $\mathcal{W}_2$ distances between the generated and ground-truth distributions at each time point on the Cite 5D dataset

Method	Dynamics	$t=1$		$t=2$		$t=3$	
		$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$
OT-CFM	1st Order	0.5615 ± 0.0156	0.6484 ± 0.0268	0.8472 ± 0.0463	0.9416 ± 0.0443	1.0536 ± 0.0534	1.1566 ± 0.0573
SF^2M	1st Order	0.5932 ± 0.0249	0.6797 ± 0.0424	0.7768 ± 0.0451	0.8670 ± 0.0480	0.9251 ± 0.0279	1.0221 ± 0.0352
3MSBM	2nd Order	5.3735 ± 0.3961	5.4244 ± 0.3582	4.3118 ± 0.4589	4.5458 ± 0.4141	1.2254 ± 0.2998	1.3427 ± 0.3307
MMFM	2nd Order	0.5335 ± 0.0134	0.5911 ± 0.0057	2.2290 ± 0.2151	2.4557 ± 0.1613	4.8413 ± 0.2538	5.2261 ± 0.4061
HRF	2nd Order	0.4225 ± 0.0152	0.4622 ± 0.0185	0.6227 ± 0.0421	0.6814 ± 0.0392	**0.7638** ± 0.0738	**0.8394** ± 0.0713
CAF	2nd Order	5.1724± 0.0452	5.6242± 0.0763	5.4333 ± 0.0914	5.4637 ± 0.1425	3.0710 ± 0.1853	3.1354 ± 0.2672
TF (Ours)	2nd Order	**0.2682** ± 0.0241	**0.2916**± 0.0312	**0.6094** ± 0.0615	**0.6516** ± 0.0684	0.8808 ± 0.1208	0.9725 ± 0.1329

Table 6: $\mathcal{W}_1$ and $\mathcal{W}_2$ distances between the generated and ground-truth distributions at each time point on the Cite 100D dataset

Method	Dynamics	$t=1$		$t=2$		$t=3$	
		$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$
OT-CFM	1st Order	10.1136 ± 0.0104	10.1873 ± 0.0135	10.2504 ± 0.0209	10.3145 ± 0.0201	11.2682 ± 0.0938	11.3642 ± 0.0997
SF^2M	1st Order	10.3598 ± 0.0051	10.4448 ± 0.0053	11.4647 ± 0.0384	11.5456 ± 0.0402	15.2032 ± 0.1813	15.4367 ± 0.1949
3MSBM	2nd Order	19.9677± 0.1373	20.1015 ± 0.1105	15.3218 ± 0.1232	15.4089 ± 0.1343	12.0229 ± 0.2895	12.1395 ± 0.3621
MMFM	2nd Order	9.8726 ± 0.0383	9.9305 ± 0.0361	12.1374 ± 0.2422	12.3076 ± 0.2495	15.5960 ± 1.4972	16.1222 ± 2.0195
HRF	2nd Order	9.9660 ± 0.0215	10.0580 ± 0.0341	9.9451 ± 0.0632	9.9986 ± 0.1526	**10.5485** ± 0.1082	**10.6442** ± 0.1254
CAF	2nd Order	18.1182 ± 0.3214	18.1971 ± 0.2845	17.4947 ± 0.4267	17.5767 ± 0.4912	14.6158 ± 0.5632	15.1773 ± 0.5215
TF (Ours)	2nd Order	**2.8955** ± 0.1429	**3.2763** ± 0.1776	**9.8235** ± 0.1489	**9.8773**± 0.1827	11.4910 ± 0.1693	11.6054± 0.2064

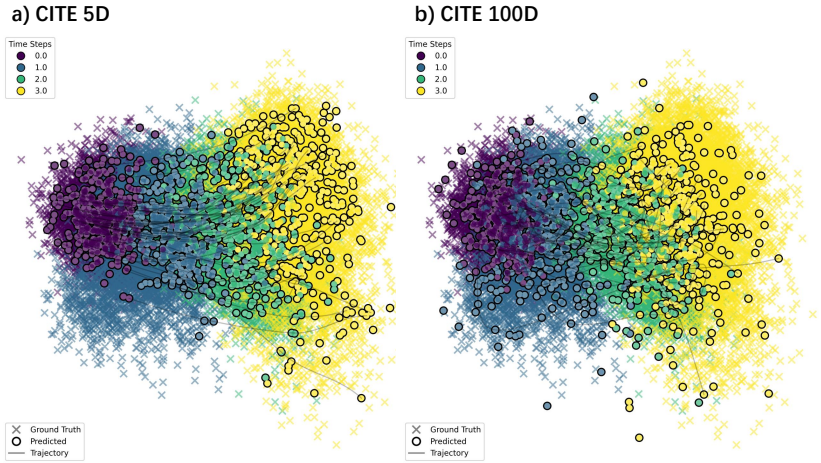


Figure 7: Trajectories learned by TracingFlow on the (a) Cite 5D and (b) Cite 100D datasets, projected to 2D using PCA.

EB 5D Data Table 7 displays the interpolation accuracy on the EB 5D dataset, evaluated by holding out one time point during training. TracingFlow outperformed other methods in reconstruction accuracy for all time points except $t = 1$. Figure 8 illustrates the interpolated distributions generated by TracingFlow across the four time points.

Table 7: $\mathcal{W}_1$ and $\mathcal{W}_2$ distances between predicted and ground-truth distributions at held-out time points on the EB 5D dataset.

Method	Dynamics	$t=1$		$t=2$		$t=3$		$t=4$	
		$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$
OT-CFM	1st Order	3.4335 ± 0.1465	3.8462 ± 0.1449	4.8404 ± 0.1496	5.5792 ± 0.1358	5.0713 ± 0.2839	5.6448 ± 0.2861	7.1368 ± 0.3799	7.6938 ± 0.4574
SF^{2}M		**3.4204** ± 0.0637	**3.6713** ± 0.0589	**3.5511** ± 0.0394	3.9441 ± 0.0418	4.3175 ± 0.2517	4.8738 ± 0.3247	**6.9516** ± 0.0616	7.4727 ± 0.0795
3MSBM	2nd Order	6.1086 ± 1.1215	6.5434 ± 1.2227	7.3626 ± 0.8043	8.0596 ± 1.0252	7.5862 ± 1.1170	8.1734 ± 0.7900	7.7029 ± 0.4241	**7.1022** ± 0.3971
MMFM		10.7803 ± 0.6718	11.8376 ± 0.7823	5.7932 ± 0.2162	6.2242 ± 0.1828	9.3326 ± 0.5617	9.9837 ± 0.4994	13.2046 ± 2.1528	14.3464 ± 2.2982
TF (Ours)		3.5611 ± 0.0362	3.8930 ± 0.0243	3.5002 ± 0.0983	**3.8035** ± 0.1086	**3.5643** ± 0.1140	**3.7947** ± 0.4123	7.3639 ± 0.1316	7.6523 ± 0.1976

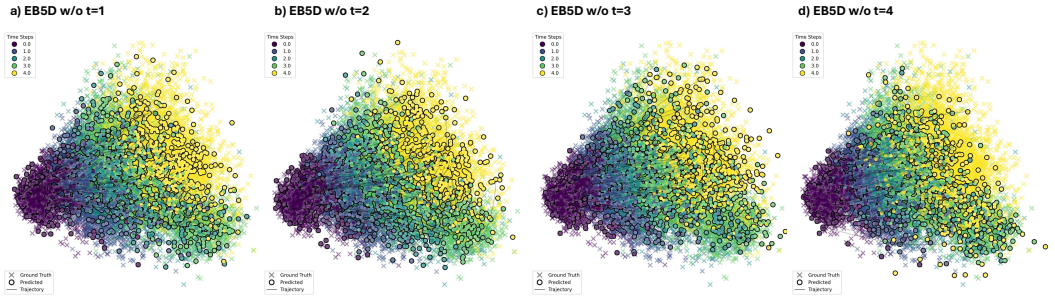


Figure 8: Predicted distributions by TracingFlow at four held-out time points, projected to 2D using PCA.

3D Simulation Lineage Dataset Table 8 demonstrates the ability of various algorithms to preserve biological priors on the 3D Simulation Lineage dataset. Since the trajectories learned by TracingFlow align with biological priors (see Figure 4), it achieves lower lineage-weighted $\mathcal{W}1$ and $\mathcal{W}2$ scores on average.

Table 8: lineage-weighted $\mathcal{W}_1$ and $\mathcal{W}_2$ distances between the generated and ground-truth distributions at each time point on the 3D Simulation Lineage dataset

Method	Dynamics	$t=1$		$t=2$	
		$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$
OT-CFM	1st Order	2.6080 ± 0.0029	2.6869 ± 0.0030	1.8686 ± 0.0382	1.9456 ± 0.0403
SF^{2}M		2.8262 ± 0.0409	2.8373 ± 0.0462	0.2247 ± 0.0617	0.2533 ± 0.0568
3MSBM	2nd Order	4.0141 ± 0.8788	4.2262 ± 0.8596	0.7548 ± 0.0726	0.6751 ± 0.0573
MMFM		1.8585 ± 0.0350	2.1618 ± 0.0496	3.5695 ± 0.5692	4.1028 ± 0.6037
HRF		2.9150 ± 0.2595	2.9265 ± 0.3901	0.6717 ± 0.5491	0.6905 ± 0.4143
CAF		1.6284 ± 0.2249	1.6623 ± 0.3726	3.7328 ± 0.3305	3.8471 ± 0.3316
TF w/o Bio. Prior		2.8267 ± 0.2896	2.8376 ± 0.3714	0.2317 ± 0.0512	0.2687 ± 0.0892
TF(Ours)		**0.4282** ± 0.1606	**0.4989** ± 0.1585	0.4795 ± 0.2680	0.5667 ± 0.2654

Hematopoiesis Dataset Table 9 presents the performance of different algorithms in retaining biological priors on real-world datasets, measured by the lineage-weighted $\mathcal{W}1, \mathcal{W}2$ distance. TracingFlow surpassed other algorithms at all time points. We also evaluated the algorithms without introducing biological priors by Vanilla $\mathcal{W}_1$, $\mathcal{W}_2$ distance, and as shown in Table 10, TracingFlow maintained superior performance. Figure 9 visualizes the data points generated by TracingFlow on the real-world dataset under both conditions (with and without biological priors).

Table 9: lineage-weighted $\mathcal{W}_1$ and $\mathcal{W}_2$ distances between the generated and ground-truth distributions at each time point on the Hematopoiesis dataset

Method	**Dynamics**	$t = 1$		$t = 2$	
		$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$
OT-CFM	1st Order	13.2972 $\pm$ 0.0804	13.7134 $\pm$ 0.0831	16.6918 $\pm$ 0.1973	17.0807 $\pm$ 0.2025
SF^{2}M		14.1427 $\pm$ 0.0341	14.4199 $\pm$ 0.0303	23.2913 $\pm$ 0.2590	23.5314 $\pm$ 0.2640
3MSBM	2nd Order	17.6487 $\pm$ 0.5250	17.8710 $\pm$ 0.5204	19.0139 $\pm$ 1.4418	19.4144 $\pm$ 1.3622
MMFM		18.4040 $\pm$ 2.8129	19.8538 $\pm$ 2.8906	36.0412 $\pm$ 3.0429	40.2282 $\pm$ 3.3934
HRF		14.1007 $\pm$ 0.0804	14.3122 $\pm$ 0.0831	16.8135 $\pm$ 0.1973	17.1154 $\pm$ 0.2025
CAF		13.1098 $\pm$ 0.0628	13.3263 $\pm$ 0.0742	17.3232 $\pm$ 0.0580	17.6685 $\pm$ 0.0572
TF w/o Bio. Prior		14.1260 $\pm$ 0.0526	14.5958 $\pm$ 0.0537	19.1073 $\pm$ 0.1932	19.4262 $\pm$ 0.2375
TF		**11.5203** $\pm$ 0.0695	**12.0151** $\pm$ 0.0809	**16.5736** $\pm$ 0.2603	**16.9027** $\pm$ 0.2999

Table 10: $\mathcal{W}1$ and $\mathcal{W}2$ distances between the generated and ground-truth distributions at each time point on the Hematopoiesis dataset

Method	**Dynamics**	$t = 1$		$t = 2$		**Average**	
		$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$
OT-CFM	First-order	9.8780	10.6132	10.7621	11.4466	10.3201	11.0299
SF^{2}M		10.2352	10.8017	13.2507	13.6879	11.7430	12.2428
3MSBM	Second-order	12.8214	13.2501	20.0262	20.6374	16.4238	16.9438
MMFM		10.4305	10.8695	22.3269	23.5322	16.3787	17.2008
HRF		10.6233	11.4510	10.9570	11.6201	10.7902	11.5356
CAF		21.8716	22.3309	17.1251	18.3914	19.4984	20.3612
TF w/o Bio. Prior		**7.0139**	**7.4140**	**10.1642**	**10.5925**	**8.5891**	**9.0032**

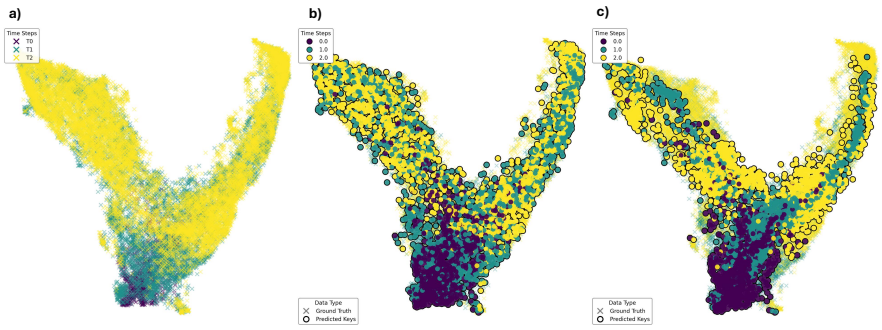


Figure 9: Original data (a), and data points generated by TracingFlow on the Hematopoiesis dataset without biological priors (b) and with biological priors (c), projected to 2D using UMAP.

Mexico Gulf Dataset Table 11 presents the performance of various algorithms on the Mexico Gulf dataset. Tracing Flow achieves the best distribution reconstruction accuracy at all time points, yielding smooth trajectories.

Table 11: $\mathcal{W}_1$ and $\mathcal{W}_2$ distances between the generated and ground-truth distributions at each time point on the Mexico Gulf dataset

Method	**Dynamics**	$t=1$		$t=2$		$t=3$		$t=4$	
		$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$
OT-CFM	First-order	0.0336	0.0375	0.0791	0.0887	0.1367	0.1415	0.1409	0.1474
SF^{2}M		0.0768	0.0917	0.0802	0.0964	0.0819	0.0968	0.0809	0.0987
3MSBM	Second-order	0.4407	0.6305	0.5565	0.7617	0.5887	0.9113	0.5231	1.0714
MMFM		0.0346	0.0364	0.2961	0.3008	1.0538	1.0571	1.8755	1.8766
HRF		0.6205	0.6217	1.2004	1.2011	1.5334	1.5337	1.4986	1.5013
CAF		1.7659	1.7663	2.4713	2.4714	2.0742	2.0755	1.3040	1.3071
TF		**0.0128**	**0.0145**	**0.0301**	**0.0339**	**0.0407**	**0.0458**	**0.0348**	**0.0456**

Method	**Dynamics**	$t=5$		$t=6$		$t=7$		$t=8$		**Average**	
		$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$
OT-CFM	First-order	0.1318	0.1370	0.1364	0.1425	0.1610	0.1799	0.1871	0.2236	0.1258	0.1373
SF^{2}M		0.1030	0.1191	0.0851	0.0956	0.1401	0.1494	0.1965	0.2068	0.1056	0.1193
3MSBM	Second-order	0.2185	1.2783	0.5362	2.0436	0.8474	4.5953	1.3606	13.2434	0.6340	3.0669
MMFM		2.3926	2.3928	2.7214	2.7215	2.9191	2.9199	2.9930	2.9945	1.7857	1.7875
HRF		1.3022	1.3075	0.9476	0.9518	0.6480	0.6535	0.3296	0.3341	1.0101	1.0131
CAF		0.7086	0.7124	0.4669	0.4751	0.6672	0.6914	0.9641	0.9914	1.3028	1.3113
TF		**0.0545**	**0.0657**	**0.0778**	**0.0824**	**0.1039**	**0.1133**	**0.1206**	**0.1309**	**0.0594**	**0.0665**

C.2 Training time and scalability of TracingFlow

To evaluate the scalability of TracingFlow, we report the training time required by each method across various datasets, as presented in Table 12. Experimental results indicate that Flow Matching algorithms based on second-order dynamics (e.g., 3 MSBM and our TracingFlow) generally require more training time compared to first-order dynamics algorithms (e.g., OT-CFM and SF^{2}M). Notably, the training time of TracingFlow is slightly lower than that of 3 MSBM, demonstrating that our method does not introduce additional computational overhead.

Table 12: Training time comparison (in seconds) of different methods across various datasets.

Method	**Simulation**	**Cite 5D**	**Cite 100D**	**SimLineage-3D**	**RealLineage**
OT-CFM	12 s	175 s	218 s	13 s	196 s
SF^{2}M	26 s	79 s	84 s	34 s	61 s
MMFM	42 s	96 s	179 s	68 s	201 s
3 MSBM	220 s	495 s	910 s	244 s	855 s
CAF	94 s	274 s	382 s	256 s	264 s
HRF	247 s	271 s	359 s	289 s	347 s
TF	225 s	319 s	806 s	269 s	742 s

To further empirically evaluate the scalability of TracingFlow, we construct a simulated dataset where 500 data points are sampled from a 500-dimensional Gaussian distribution at each time step. For a fair comparison, all methods use the same 4-layer MLP to regress the velocity or acceleration fields, with a batch size of 2000. The training time and peak GPU memory usage of each method are reported in Table 13. TracingFlow requires less training time than CAF and 3 MSBM, which are also second-order Flow Matching methods. This is because TracingFlow avoids the repeated fitting of the acceleration field required by 3 MSBM, despite iteratively solving for the velocities and the optimal coupling.

Table 13: Training time (in seconds) and peak GPU memory usage (in MB) of different methods on the simulated 500D dataset.

Method	**OT-CFM**	**SF^{2}M**	**HRF**	**CAF**	**3 MSBM**	**TF**
Training Time (s)	224.9	236.2	478.8	826.5	2072.0	584.0
Peak Memory (MB)	1583	1611	1422	3382	8086	7972

The peak GPU memory usage of TracingFlow is comparable to that of 3 MSBM. The main memory consumption occurs during the iterative optimization of velocities and the optimal coupling described in Section 4.3. Specifically, we store a large number of trajectories sampled according to the computed coupling and construct a large linear system to solve for the velocities along these trajectories before averaging them. To reduce memory usage, one could instead process fewer trajectories at a time and incrementally update the mean velocity at each point. However, this may slightly increase training time due to reduced GPU parallelism. We plan to provide this memory-efficient mode as an option in our code.

C.3 Sensitivity Analysis on Minibatch-OT

In all experiments presented in this paper, the full dataset was utilized to compute the OT plan. However, as the scale of the data increases, the computational cost of the OT plan grows rapidly, making the adoption of the minibatch computation method described in Section 5.2 inevitable. We conducted a sensitivity analysis on the Cite-5D dataset to evaluate the impact of batch size in minibatch-OT computation on the accuracy of distribution reconstruction. The results are presented in Table 14. Experimental results demonstrate that TracingFlow consistently and accurately reconstructs the distributions at each time point when the batch size is adjusted within a certain range (batch size $\sim 10^3$).

Table 14: Sensitivity analysis of batch size for Minibatch-OT on the Cite-5D dataset.

Batch Size	$t=1$		$t=2$		$t=3$		**Average**	
	$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$
1000	0.4344	0.4798	0.6522	0.7183	0.7658	0.8479	0.6175	0.6820
2000	0.4196	0.4624	0.5640	0.6251	0.7972	0.8820	0.5936	0.6565
Full Dataset	0.2868	0.3185	0.5487	0.5997	0.7689	0.8499	0.5348	0.5894

C.4 Further Evaluation of the Effectiveness of Biological Priors

Since the lineage-weighted $\mathcal{W}_1$ and $\mathcal{W}_2$ are asymmetric and not standard metrics, we introduce an additional evaluation metric, Branching Probability Accuracy (BPA), to evaluate lineage consistency.

Specifically, for the i-th cell at the initial time point with barcode b_i, its predicted position at time t_j is denoted as $x(x_i(0), t_j)$. We count the number of cells sharing the same barcode b_i among the M nearest neighbors ($M = 10$) of this predicted position. The proportion of these cells provides an empirical estimate of the probability that cell i differentiates into the lineage with barcode b_i, denoted as $\mathbb{P}(b_i, t_j|i)$. We average this estimate over all initial cells and evaluated time points to obtain

$$\mathrm{BPA} = \frac{1}{NK} \sum_{i=1}^{N} \sum_{j=1}^{K} \mathbb{P}(b_i, t_j|i), \tag{80}$$

where N is the number of initial cells and K is the number of evaluated time points.

In addition, we compute a random baseline by randomly selecting a cell from the observed dataset at time t_j as the descendant of the i-th initial cell. The reported results are the ratios of BPA to its random baseline.

We evaluate this metric on the 3D Simulation Lineage and real Hematopoiesis datasets. As described in the main text, the baseline methods also incorporate lineage priors by using barcodes as conditions in the neural networks that regress the velocity fields. The experimental results are presented in Table 15. The random baseline values are 0.3332 and 2.203×10^{-3} for the two datasets, respectively.

Table 15: Branching Probability Accuracy (BPA) normalized by the random baseline on the 3D Simulation Lineage and real Hematopoiesis datasets. Higher values indicate better performance.

Dataset	**OT-CFM**	**SF^{2}M**	**HRF**	**CAF**	**MMFM**	**3 MSBM**	**TF w/o prior**	**TF with prior**
3D Simulation Lineage	1.758	1.899	1.922	1.186	1.101	1.467	1.912	**2.688**
Hematopoiesis	3.074	1.912	3.094	1.855	0.8339	1.369	1.970	**3.191**

On the synthetic dataset, all methods achieve BPA values above the random baseline, while on the real Hematopoiesis dataset, all methods except MMFM outperform random guessing. TracingFlow with biological priors achieves the highest BPA on both datasets. On the synthetic dataset, TracingFlow outperforms all other methods. On the real Hematopoiesis dataset, TracingFlow performs slightly better than OT-CFM and HRF, and better than the remaining methods. Furthermore, incorporating biological priors consistently improves the BPA of TracingFlow on both datasets.

D Pseudocode for Algorithm

The pseudocode for the training process of Tracing Flow is provided in Algorithm 1. Additionally, our project is available in an anonymous GitHub repository at `https://anonymous.4open.science/r/TracingFlow_Review-5D75`.

Algorithm 1 Tracing-Flow Matching Training

Require: Snapshot datasets $\mathcal{D}_k = \{\boldsymbol{x}_k^{(i)}\}_{i=1}^{N_k}$ at times $t_0 < t_1 < \cdots < t_K$;
Acceleration Network $\boldsymbol{a}_{\boldsymbol{\theta}}$; Velocity Network $\boldsymbol{v}_{\boldsymbol{\xi}}$;
Hyperparameters: batch size B_{traj}, learning rates $\eta_{\boldsymbol{a}}, \eta_{\boldsymbol{v}}$.

Initialize: Estimated velocity sets $\mathcal{V}_k = \{\boldsymbol{v}_k^{(i)}\}_{i=1}^{N_k} \leftarrow \{\mathbf{0}\}$ for all k.
while Velocity Estimates Not Converged **do**
  Stage 1: Velocity Assignment (Iterative SOAT)
  Initialize velocity accumulators $\hat{\mathcal{V}}_k^{(i)} \leftarrow \emptyset$ for all k, i.
  // Step 1.1: Solve Optimal Transport
  for $k = 0$ to $K-1$ **do**
    Compute cost matrix $\mathcal{C}_{i \to i+1}$ with / w.o. biological prior.
    Compute optimal coupling $\pi^*_{k \to k+1}$ between $(\boldsymbol{x}_k, \boldsymbol{v}_k)$ and $(\boldsymbol{x}_{k+1}, \boldsymbol{v}_{k+1})$ by solving the SOAT Problem.
    Compute propagator (transition) matrix $\mathcal{K}^*_{k \to k+1}$ by normalizing $\pi^*_{k \to k+1}$:
      $\mathcal{K}^*_{k \to k+1}(i,j) = \frac{\pi^*_{k \to k+1}(i,j)}{\sum_{j'} \pi^*_{k \to k+1}(i,j')}$
  end for
  // Step 1.2: Trajectory Sampling & Velocity Update
  Sample B_{traj} discrete trajectories $T_m = \{(\boldsymbol{x}_k^{(s_{m,k})}, \boldsymbol{v}_k^{(s_{m,k})})\}_{k=0}^K$ for $m = 1 \dots B_{\text{traj}}$.
    · Initial indices $s_{m,0}$ are sampled uniformly.
    · Subsequent indices $s_{m,k+1} \sim \mathcal{K}^*_{k \to k+1}(s_{m,k}, \cdot)$.
  for $m = 1$ to B_{traj} **do**
    Compute optimal path velocities $\{\hat{\boldsymbol{v}}_k^{(s_{m,k})}\}_{k=0}^K$ for trajectory T_m using Proposition 4.8.
    Accumulate: $\hat{\mathcal{V}}_k^{(s_{m,k})} \leftarrow \hat{\mathcal{V}}_k^{(s_{m,k})} \cup \{\hat{\boldsymbol{v}}_k^{(s_{m,k})}\}$ for $k = 0 \dots K$.
  end for
  // Step 1.3: Average Accumulated Velocities
  for $k = 0$ to K, $i = 1$ to N_k **do**
    if $\hat{\mathcal{V}}_k^{(i)} \neq \emptyset$ **then**
      $\boldsymbol{v}_k^{(i)} \leftarrow \frac{1}{|\hat{\mathcal{V}}_k^{(i)}|} \sum_{\boldsymbol{v} \in \hat{\mathcal{V}}_k^{(i)}} \boldsymbol{v}$
    end if
  end for
end while
Stage 2: Final Transport Plan Computation
Compute optimal couplings π^* and propagators $\mathcal{K}^*$ using the converged velocities $\mathcal{V}_k$.
Stage 3: Neural Network Training
while Not Converged **do**
  Sample random time $t \sim \mathcal{U}[t_0, t_K]$ and coupling $\boldsymbol{z} \sim q(\boldsymbol{z})$.
  Compute Acceleration Matching Loss:
    $\mathcal{L}_{\text{CAFM}} = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{v}) \sim \rho_t(\cdot|z), \boldsymbol{z} \sim q(\boldsymbol{z}), t \sim \mathcal{U}[t_0, t_K]} \left[\| \boldsymbol{a}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{v}, t) - \boldsymbol{a}(\boldsymbol{x}, \boldsymbol{v}, t|z) \|^2 \right]$
  Compute Initial Velocity Loss:
    $\mathcal{L}_{\boldsymbol{v}_0} = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{v}) \sim \rho_0} \left[\| \boldsymbol{v}_{\boldsymbol{\xi}}(\boldsymbol{x}) - \boldsymbol{v} \|^2 \right]$
  Update parameters (Gradient Descent):
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta_{\boldsymbol{a}} \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{CAFM}}$
    $\boldsymbol{\xi} \leftarrow \boldsymbol{\xi} - \eta_{\boldsymbol{v}} \nabla_{\boldsymbol{\xi}} \mathcal{L}_{\boldsymbol{v}_0}$
end while

E Discussion

E.1 Relation to other Algorithms

Relation to 3 MSBM 3 MSBM considers the following stochastic optimal control problem:

$$\min \mathbb{E}_{\rho_t} \int [\|\boldsymbol{a}_t\|^2] \mathrm{d}t \tag{81}$$

$$\mathrm{d}\boldsymbol{m}_t = A\boldsymbol{m}_t \mathrm{d}t + \boldsymbol{u}_t \mathrm{d}t + g\mathrm{d}\boldsymbol{W}_t, \quad \boldsymbol{x}_n \sim q_n = \int \pi_n(\boldsymbol{x}, \boldsymbol{v}) \mathrm{d}\boldsymbol{v}_n \tag{82}$$

where $\boldsymbol{m}_t = [\boldsymbol{x}_t, \boldsymbol{v}_t]^T$, $A = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$, and $g = \begin{bmatrix} 0 & 0 \\ 0 & \sigma \end{bmatrix}$. This problem can be viewed as a "stochastic" version of the VM-DOAT problem proposed in this paper. 3 MSBM also provides a simulation-free training method to regress the acceleration field. However, 3 MSBM does not explicitly solve this stochastic optimal control problem using the form of optimal coupling plus optimal single-particle trajectories. Its optimal coupling requires iterative solutions similar to Rectified Flow [36] : a process of repeatedly selecting the coupling and training a_θ. Furthermore, it relies on a heuristic method to estimate the initial velocity (via normal initialization and iteratively solving forward and backward SDEs), rather than incorporating this estimation as part of the optimal control problem. Experiments show that Tracing Flow achieves better distribution reconstruction accuracy than 3MSBM.

Relation to MMFM In MMFM, the single-particle path is similarly specified by:

$$\int \|\gamma''(t)\|^2 \mathrm{d}t, \quad \boldsymbol{x}_k = \gamma(t_k) \tag{83}$$

which corresponds to a natural spline. However, MMFM does not solve an optimal control problem over distributions. Furthermore, the velocity field in MMFM remains single-valued with respect to $\boldsymbol{x}$, which prevents it from learning trajectories that cross in the position space.

Relation to OAT-FM OAT-FM introduces a DOAT problem formulation consistent with this paper; however, the objective of OAT-FM is not to genuinely solve the DOAT problem within the augmented space $\mathcal{X} \times \mathcal{V}$, but rather to fine-tune a pre-trained FM model. It employs the loss:

$$\begin{aligned} \mathcal{L}_{\text{CAFM}} = \mathbb{E}_{\pi^*[(\boldsymbol{x}_0, \boldsymbol{v}_0), (\boldsymbol{x}_1, \boldsymbol{v}_1)]} \big[& \alpha \| \frac{\boldsymbol{x}_t - \boldsymbol{x}_0}{t} - \frac{\boldsymbol{v}_0 + \boldsymbol{v}_{\boldsymbol{\theta}}}{2} \|^2 + (1-\alpha) \|\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{v}_0\|^2 \\ & + \alpha \| \frac{\boldsymbol{x}_1 - \boldsymbol{x}_t}{1-t} - \frac{\boldsymbol{v}_{\boldsymbol{\theta}} + \boldsymbol{v}_1}{2} \|^2 + (1-\alpha) \|\boldsymbol{v}_1 - \boldsymbol{v}_\theta(\boldsymbol{x}_t, t)\|^2 \big] \end{aligned} \tag{84}$$

to enforce the velocity field along the line segment connecting any two samples to be as parallel to that line as possible. As a fine-tuning approach, it effectively still addresses a distribution transport problem involving only two time points within the position space $\mathcal{X}$, rather than tackling a multi-marginal optimal control problem. Furthermore, OAT-FM does not establish a connection between its loss function and the cost along each trajectory $\frac{1}{2}\int_0^1 \|\ddot{\gamma}\|^2 \mathrm{d}t$, nor does it explicitly learn the acceleration field $\boldsymbol{a}(\boldsymbol{x}, \boldsymbol{v}, t)$.

E.2 Is $\mathcal{S} = \mathcal{X} \times \mathcal{V}$ a Phase Space?

In the Conclusion and Limitation section, we noted that if $\boldsymbol{x}$ is regarded as the generalized coordinate and $\boldsymbol{v}$ as its time derivative, the term $\int \frac{1}{2}\|\boldsymbol{a}\|^2 \mathrm{d}t$ cannot be interpreted as a physical action, as the action in classical mechanics typically does not involve second-order time derivatives. Here, we offer an alternative physical interpretation of this control objective.

Proposition E.1. *Consider a d-dimensional second-order dynamical system governed by the equations:*

$$\dot{\boldsymbol{x}} = \boldsymbol{v} \tag{85}$$

$$\dot{\boldsymbol{v}} = \boldsymbol{a} \tag{86}$$

where $\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{a} \in \mathbb{R}^d$. The optimal control objective is given by:

$$\min \frac{1}{2} \int_0^T \|\boldsymbol{a}\|^2 \mathrm{d}t \tag{87}$$

The evolution of this system under the optimal control law is equivalent to a Hamiltonian system in $4d$-dimensional phase space with the following Hamiltonian:

$$H = \boldsymbol{p}_1^T \boldsymbol{q}_2 + \frac{1}{2}\|\boldsymbol{p}_2\|^2 \tag{88}$$

where $\boldsymbol{q}_1, \boldsymbol{q}_2 \in \mathbb{R}^d$ are the canonical coordinates, and $\boldsymbol{p}_1, \boldsymbol{p}_2 \in \mathbb{R}^d$ are the canonical momenta (note that $\boldsymbol{q}_1$ does not appear in the Hamiltonian). Here, $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ correspond to $\boldsymbol{x}$ and $\boldsymbol{v}$ in the optimal control problem, respectively, while $\boldsymbol{p}_2$ corresponds to $\boldsymbol{a}$.

Proof. We derive the equations of motion directly using Hamilton's canonical equations:

$$\dot{\boldsymbol{q}}_1 = \frac{\partial H}{\partial \boldsymbol{p}_1} = \boldsymbol{q}_2, \quad \dot{\boldsymbol{q}}_2 = \frac{\partial H}{\partial \boldsymbol{p}_2} = \boldsymbol{p}_2, \quad \dot{\boldsymbol{p}}_1 = -\frac{\partial H}{\partial \boldsymbol{q}_1} = \boldsymbol{0}, \quad \dot{\boldsymbol{p}}_2 = -\frac{\partial H}{\partial \boldsymbol{q}_2} = -\boldsymbol{p}_1 \tag{89}$$

This implies that $\boldsymbol{p}_1$ is constant. Solving these equations sequentially yields:

$$\boldsymbol{p}_2(t) = \boldsymbol{c}_2' - \boldsymbol{p}_1 t \tag{90}$$

$$\boldsymbol{q}_2(t) = \boldsymbol{c}_1' + \boldsymbol{c}_2' t - \frac{1}{2}\boldsymbol{p}_1 t^2 \tag{91}$$

$$\boldsymbol{q}_1(t) = \boldsymbol{c}_0' + \boldsymbol{c}_1' + \frac{1}{2}\boldsymbol{c}_2' t^2 - \frac{1}{6}\boldsymbol{p}_1 t^3 \tag{92}$$

Comparing this with the results in Proposition 4.1, we observe that $\boldsymbol{q}_1(t)$ and $\boldsymbol{q}_2(t)$ satisfy the same equations as $\boldsymbol{x}(t)$ and $\boldsymbol{v}(t)$, with the undetermined constants determined by the boundary conditions. This completes the proof. □

Consequently, we observe that although the augmented space $\mathcal{S} = \mathcal{X} \times \mathcal{V}$ concatenates position $\boldsymbol{x}$ and velocity (momentum) $\boldsymbol{v}$, it **cannot** be interpreted as a **phase space**. Instead, it should be viewed as the **configuration space** of the classical mechanical system described by the Hamiltonian above. We can attempt to recover the Lagrangian from this Hamiltonian. The canonical equations established that $\dot{\boldsymbol{q}}_1 = \boldsymbol{q}_2$ and $\dot{\boldsymbol{q}}_2 = \boldsymbol{p}_2$. Applying the Legendre transformation:

$$\begin{aligned} L &= \boldsymbol{p}_1^T \dot{\boldsymbol{q}}_1 + \boldsymbol{p}_2^T \dot{\boldsymbol{q}}_2 - H \\ &= \boldsymbol{p}_1^T (\dot{\boldsymbol{q}}_1 - \boldsymbol{q}_2) + \frac{1}{2}\|\dot{\boldsymbol{q}}_2\|^2 \end{aligned} \tag{93}$$

We find that the term $\boldsymbol{p}_1^T$ cannot be eliminated; the Lagrangian cannot be expressed solely as a function of the canonical coordinates and their first derivatives. This is a characteristic of constrained systems (in Dirac's theory of constraints, a constrained system is typically defined as one where the Hessian matrix of the Lagrangian is singular. To see this intuitively, if we regard $\boldsymbol{p}_1$ in the Lagrangian as d new canonical coordinates, their corresponding canonical momenta are zero, implying the motion is confined to a submanifold of the phase space).

In Equation (9), we derived the minimum cost for the system to transition from $(\boldsymbol{x}_1, \boldsymbol{v}_1)$ to $(\boldsymbol{x}_2, \boldsymbol{v}_2)$. While this cost is a symmetric positive-definite quadratic form with respect to $\boldsymbol{x}$ and $\boldsymbol{v}$, it does not constitute a metric on the augmented space $\mathcal{S} = \mathcal{X} \times \mathcal{V}$. (If it were a metric, the augmented space would be flat, and the optimal control trajectories—geodesics—would be linear functions of t). Maupertuis' principle suggests a close relationship between the metric in the configuration space and the system's Lagrangian [3]: if the Lagrangian takes the form:

$$L(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \frac{1}{2} m_{ij}(\boldsymbol{q}) \dot{q}^i \dot{q}^j - V(\boldsymbol{q}) \tag{94}$$

then the configuration space possesses the metric:

$$g_{ij}(\boldsymbol{q}) = 2(E - V(\boldsymbol{q})) m_{ij}(\boldsymbol{q}) \tag{95}$$

where E is the total energy of the particle. For the optimal control problem discussed herein, we have seen that the Lagrangian cannot be written in such a form; therefore, it is fundamentally impossible to equip the configuration space $\mathcal{S} = \mathcal{X} \times \mathcal{V}$ with such a metric.

E.3 Broader Impacts

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here. However, as our algorithm is applied to single-cell data analysis, the fidelity of the generated trajectories is heavily dependent on the quality of the input data. Consequently, when utilizing our method for biological or medical research purposes, it is critical to employ high-quality datasets, incorporate established and accurate biological priors, and ensure that the algorithmic outputs are rigorously validated by domain experts.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract clearly articulates the contributions of this paper. Our theoretical and algorithmic contributions are primarily detailed in Sections 3, 4, and 5, while Section 6 provides experimental validation.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: The limitations of this work are discussed in Section 7.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The proofs of all theorems are provided in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All code for this paper is available in the anonymous GitHub repository `https://anonymous.4open.science/r/TracingFlow_Review-5D75` for reproducing the experiments. The detailed pseudocode for the training algorithm is provided in Appendix D.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All code for this paper is available in the anonymous GitHub repository `https://anonymous.4open.science/r/TracingFlow_Review-5D75` for reproducing the experiments.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental settings are provided in Appendix B.1, detailing the neural network architectures and the computational resources required for the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For all experiments in the main text, we report the mean and standard deviation for each metric.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the computational resources required for the experiments in Appendix B.1, and the training times in Appendix C.2.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have all reviewed and confirmed this research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discussed broader impacts in Section E.3.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper poses no such risks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All the datasets used for analysis in this study are publicly available from the original source. They are properly credited, respected, mentioned and used. All datasets are listed in Appendix B.3.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not release new datasets.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We used LLMs solely for grammar checking and language polishing. No LLMs were involved in the core research process.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.