

---

# Joint Velocity-Growth Flow Matching for Single-Cell Dynamics Modeling

---

Dongyi Wang<sup>1,\*</sup>, Yuanwei Jiang<sup>1,\*</sup>, Zhenyi Zhang<sup>2,\*</sup>, Xiang Gu<sup>1,†</sup>, Peijie Zhou<sup>3,†</sup>, Jian Sun<sup>1,†</sup>

<sup>1</sup>School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, China

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

<sup>3</sup>Center for Machine Learning Research, Peking University, Beijing, China.

{dongyiwang, jyw1578857771}@stu.xjtu.edu.cn; {xianggu, jiansun}@xjtu.edu.cn;  
zhenyizhang@stu.pku.edu.cn; pjzhou@pku.edu.cn

## Abstract

Learning the underlying dynamics of single cells from snapshot data has gained increasing attention in scientific and machine learning research. The destructive measurement technique and cell proliferation/death result in unpaired and unbalanced data between snapshots, making the learning of the underlying dynamics challenging. In this paper, we propose joint Velocity-Growth Flow Matching (VGFM), a novel paradigm that jointly learns state transition and mass growth of single-cell populations via flow matching. VGFM builds an ideal single-cell dynamics containing velocity of state and growth of mass, driven by a presented two-period dynamic understanding of the static semi-relaxed optimal transport, a mathematical tool that seeks the coupling between unpaired and unbalanced data. To enable practical usage, we approximate the ideal dynamics using neural networks, forming our joint velocity and growth matching framework. A distribution fitting loss is also employed in VGFM to further improve the fitting performance for snapshot data. Extensive experimental results on both synthetic and real datasets demonstrate that VGFM can capture the underlying biological dynamics accounting for mass and state variations over time, outperforming existing approaches for single-cell dynamics modeling. Our code is available at <https://github.com/DongyiWang-66/VGFM>.

## 1 Introduction

Inferring the latent dynamics of complex systems from sparse and noisy data is a fundamental challenge in science and engineering. In many domains, *e.g.*, stock markets [1], climate systems [2], and biological processes [3–11], continuous trajectories are rarely fully observed by sensors. Instead, cross-sectional snapshot data collected at discrete time points are commonly provided. This challenge is especially important in single-cell RNA sequencing [12–16], where destructive sampling yields unpaired population-level snapshots across time without having the tracked individual cell fates. Furthermore, due to the mass changes during cellular development or response processes, observed data often exhibits mass unbalancedness across time, violating mass conservation. Consequently, reconstructing the time-evolving, unnormalized density function from limited samples has become an important research problem and attracted increasing attention [14, 15, 17].

Deep learning-based models for dynamics inference have demonstrated great potential [3, 7, 18–29]. These models typically employ ordinary or stochastic differential equations (ODEs or SDEs) parameterized by neural networks to approximate the velocity field governing density evolution.

---

\*These authors contributed equally. †Corresponding authors.

One class of methods, based on simulation [3, 18, 19, 23, 24, 26, 30–32], generates synthetic trajectories by feeding initial data through neural networks and solving ODEs or SDEs numerically, and compares simulated results with observations for loss computation. However, these simulation-based methods rely heavily on numerical solvers during training, which significantly increases computational cost. In high-dimensional settings, the enlarged search space further exacerbates instability, hindering scalability and convergence. Another line of research focuses on simulation-free approaches [22, 29, 33–43], where the velocity field is trained efficiently by constructing conditional probability paths without trajectory simulation, yielding better efficiency and stability in training compared with simulation approaches.

Nevertheless, the simulation-free approaches mentioned above rely solely on the velocity field  $v$ , ignoring the unbalancedness of observed data, which can result in incorrect reconstruction of underlying dynamics or unsatisfying generation performance [26, 37, 42]. Distributional imbalance is a common phenomenon in single-cell dynamics, where cellular proliferation and death occur, thus necessitating the introduction of a growth term  $g$  to allow for mass increasing or decreasing. To address this, [44, 45] proposed to jointly learn velocity  $v$  and growth  $g$  by minimizing the Wasserstein-Fisher-Rao (WFR) metric [46, 47]. However, it mathematically enforces  $v = \nabla g$ , which lacks clear mechanistic justification in biological systems. Other WFR-inspired approaches [23, 26] design separate neural networks for  $v$  and  $g$  and prioritize fitting the distributions. However, these methods require heavy simulation during training, limiting their scalability to high dimensions.

In this paper, we propose joint Velocity-Growth Flow Matching (VGFM), a novel approach for single-cell dynamics modeling. VGFM aims to learn the joint state transition<sup>2</sup> and mass growth<sup>3</sup> of single-cell evolution by flow matching, as illustrated in Fig. 1. VGFM is based on the static semi-relaxed optimal transport that allows mass variation in building coupling between snapshots, for which we present a dynamic understanding that enables mass growth and state transition accomplished in two time periods, respectively. Based on this dynamic understanding, we build a joint transition and growth dynamics, inheriting the optimal properties of semi-relaxed optimal transport. We then approximate the velocity and growth in the built dynamics using neural networks with finite samples, yielding a joint velocity and growth flow matching framework. To further improve fitting performance for the single-cell snapshots, we employ a distribution fitting loss based on the Wasserstein distance. Thanks to the joint velocity and growth flow matching, our approach enjoys more stable training and better scalability to high dimensions, compared with simulation-based approaches. Extensive experiments on both synthetic and real single-cell datasets are conducted to evaluate our proposed approach. Experimental results demonstrate accurate dynamics reconstruction and superior performance achieved by VGFM over recent baselines.

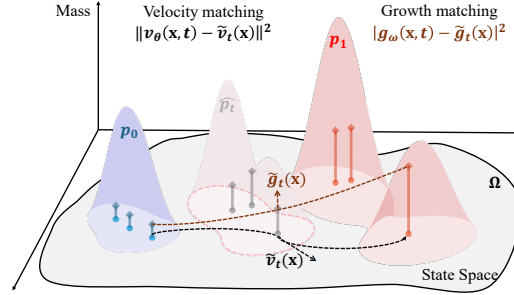


Figure 1: The goal of this paper is to learn a joint state transition (controlled by  $v_\theta(\mathbf{x}, t)$ ) and mass growth (controlled by  $g_\omega(\mathbf{x}, t)$ ) dynamics for single-cell evolution.

## 2 Background and Related Works

This section revisits optimal transport, and reviews the flow-matching-based methods for single-cell and unbalancedness-aware distribution learning methods, which are mostly related to this paper.

**Optimal transport.** Optimal transport, a tool for transforming one distribution into another at minimal cost [48], has gained remarkable prominence in recent years across fields such as image processing [49–54] and bioinformatics [18, 21, 22, 26, 41]. Given two distributions  $p_0$  and  $p_1$  on domain  $\Omega$  satisfying  $\int_\Omega p_0(\mathbf{x}) d\mathbf{x} = \int_\Omega p_1(\mathbf{x}) d\mathbf{x}$ , the optimal transport [48] aims to find a joint distribution of  $p_0$  and  $p_1$ , named coupling, such that the transport cost is minimized, formulated as

<sup>2</sup>State transition describes changes in gene expression in a cell.

<sup>3</sup>Mass growth means mass increasing or decreasing.

the following optimization problem:

$$\min_{\pi \in U(p_0, p_1)} \int_{\Omega^2} c(\mathbf{x}_0, \mathbf{x}_1) d\pi(\mathbf{x}_0, \mathbf{x}_1), \text{ s.t. } U(p_0, p_1) = \{\pi \geq 0 : P_{\#}^0 \pi = p_0, P_{\#}^1 \pi = p_1\}, \quad (1)$$

where  $c$  is the cost function. For a map  $T$ , we define  $T_{\#}p_0(\mathbf{x}) = \int_{\mathbf{x}': T(\mathbf{x}')=\mathbf{x}} p_0(\mathbf{x}') d\mathbf{x}'$ . Let  $P^i(\mathbf{x}_0, \mathbf{x}_1) = \mathbf{x}_i$  for  $i = 0, 1$ , then we have  $P_{\#}^i \pi(\mathbf{x}_i) = \int_{\Omega} \pi(\mathbf{x}_0, \mathbf{x}_1) d\mathbf{x}_i$  for  $i = 0, 1$ . When  $c(\mathbf{x}_0, \mathbf{x}_1) = \|\mathbf{x}_0 - \mathbf{x}_1\|^2$ , by Brenier's theorem [55], the optimal coupling can be expressed as  $\pi^* = (\text{Id}, T^*)_{\#}p_0$ , where  $T^*$  is called the Monge map.

**Flow matching for learning single cell dynamics.** Flow matching [33–35] is a simulation-free approach where the velocity field is trained efficiently by constructing conditional probability paths without trajectory simulation. Building on this foundation, further improvements have been achieved by incorporating optimal transport guidance [22, 36], considering unbalancedness [37], accounting for manifold structures [38], and approximating Schrödinger bridge via flow and score matching [41], all of which has been applied to model single-cell dynamics [22, 37, 38, 41]. Different from these methods, our method explicitly models the simultaneous matching of both the velocity field and the growth function, driven from our developed ideal joint state transition and mass growth dynamics, which does not require the construction of conditional probability paths like the above approaches.

**Unbalancedness-aware distribution learning methods.** Data imbalance is prevalent across a variety of domains, such as image synthesis and protein generation, motivating the development of flow matching models that account for unbalanced distributions [37, 42, 56]. However, these methods often do not model growth functions, which are essential in single-cell contexts. In the single-cell domain, several approaches have been proposed to model growth dynamics [5, 23, 44, 45, 57], yet they either fail to leverage the informative variations in cell abundance observable from snapshot data [44, 45], or rely heavily on computationally expensive simulations [5, 23, 57], limiting their scalability and efficiency. Differently, our method explicitly learns the growth function from observed snapshot data, yielding better performance for single cells than these methods as shown in experiments.

### 3 Method

Given the snapshot population of single cells, this paper aims to build the dynamic trajectory of single cells. Considering the unbalancedness between snapshot populations due to the undergo cell proliferation or death, we aim to build a model that can transform between unbalanced distributions by jointly learning a velocity field for controlling state transition and a growth function for controlling mass variation. Towards this goal, we propose joint Velocity-Growth Flow Matching (VGFM) based on semi-relaxed optimal transport to learn the dynamics of single cells. As illustrated in Fig. 2, we first present a decoupled understanding of state transition and mass growth for unbalanced dynamics based on semi-relaxed optimal transport (Figs. 2 (a), (b)), upon which we build a dynamic process between unbalanced distributions (Fig. 2 (c)). Finally, we propose the velocity and growth flow matching using the built dynamic process to learn the velocity and growth with samples (Fig. 2 (d)). Next, we first describe the unbalanced dynamics of single-cell data, and then discuss the details of each component of our method.

#### 3.1 Unbalanced Dynamics of Single Cell

Given two adjacent snapshots of populations/distributions denoted as  $p_0$  and  $p_1$ , if the system is balanced, the dynamics  $\bar{p}_t (t \in [0, 1])$  between  $p_0$  and  $p_1$  can be generated by ODE  $\frac{d\mathbf{x}_t}{dt} = v_t(\mathbf{x}_t)$  such that  $\mathbf{x}_0 \sim p_0$  and  $\mathbf{x}_1 \sim p_1$ . Corresponding to this ODE,  $\bar{p}_t$  is governed by continuity equation  $\partial_t \bar{p}_t = -\nabla \cdot (\bar{p}_t v_t)$ . However, in biological systems, cells can proliferate and die. Such behaviors can be captured by a time-dependent weight  $w_t(\mathbf{x}_t)$  associated with the state transition, whose evolution is controlled by a growth function  $g$ , simulating cell proliferation or death processes:

$$\begin{cases} \frac{d\mathbf{x}_t}{dt} = v_t(\mathbf{x}_t), \\ \frac{d \log w_t(\mathbf{x}_t)}{dt} = g_t(\mathbf{x}_t). \end{cases} \quad (2)$$

Under this formulation, the unbalanced distribution dynamics  $p_t$  is generated jointly by the position  $\mathbf{x}_t$  and the weight  $w_t(\mathbf{x}_t)$ . Specifically,  $w_t(\mathbf{x}_t)$  models the variation of  $p_t$  against  $\bar{p}_t$ , *i.e.*,  $w_t(\mathbf{x}_t) =$

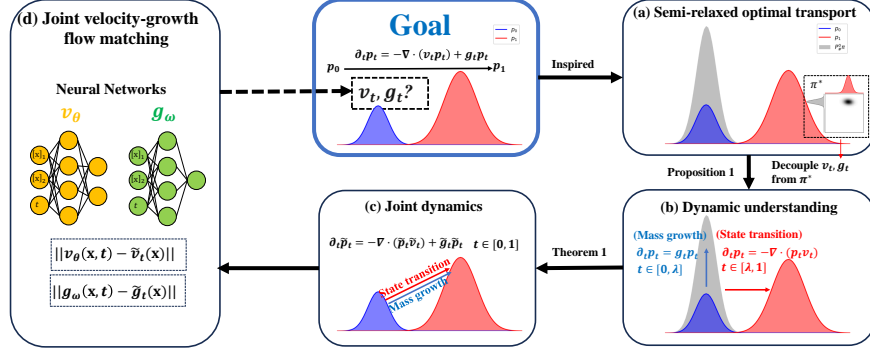


Figure 2: Illustration of our proposed VGFM, consisting of the velocity and growth flow matching deduced by the dynamic reformulation of the semi-relaxed optimal transport.

$p_t(\mathbf{x}_t)/\bar{p}_t(\mathbf{x}_t)$  for  $\bar{p}_t(\mathbf{x}_t) \neq 0$ . Corresponding to Eq. (2),  $p_t$  is governed by

$$\partial_t p_t = -\nabla \cdot (p_t v_t) + g_t p_t. \quad (3)$$

### 3.2 Building Unbalanced Dynamics Based on Semi-Relaxed Optimal Transport

Though Eqs. (2) or (3) provide a formulation of unbalanced dynamics between  $p_0$  and  $p_1$ , the equations admit a wide range of admissible dynamics. We next seek the dynamics based on semi-relaxed optimal transport. Specifically, given the semi-relaxed optimal transport

$$\min_{\pi \geq 0} \mathcal{J}_{\text{sot}}(\pi) \triangleq \int_{\Omega^2} c(\mathbf{x}_0, \mathbf{x}_1) d\pi(\mathbf{x}_0, \mathbf{x}_1) + \text{KL}(P_{\#}^0 \pi \| p_0) \quad \text{subject to} \quad P_{\#}^1 \pi = p_1, \quad (4)$$

between two unbalanced distributions  $p_0$  and  $p_1$ , we will first present a dynamic understanding of Eq. (4) that decouples the state transition and mass growth into different periods. Based on this dynamic understanding, we will build a reasonable unbalanced dynamics between  $p_0$  and  $p_1$ , which naturally drives our velocity and growth matching to learn the dynamics with samples (see Sect. 3.3).

**Dynamic understanding of semi-relaxed optimal transport.** As mentioned above, the solution of the semi-relaxed optimal transport allows both state transition and mass variation/growth. From a dynamic perspective, we understand the transport as a two-period dynamic process, in which we perform mass growth in the first period and state transition in the second period. Note that such an understanding is not practical in applications, but it provides a decoupled modeling of state transition and mass growth, which promotes a reasonable and friendly-to-matching unbalanced dynamics modeling, driving our flow matching algorithm for learning velocity and growth, as we will make clear later. More concretely, given  $\lambda \in (0, 1)$ , we perform mass growth controlled by  $g_t$  for  $t \in [0, \lambda]$  through  $\partial_t p_t = g_t p_t$  with initial and ending distributions of  $p_0$  and  $p_\lambda$  respectively. For  $t \in (\lambda, 1]$ , the state transition controlled by  $v_t$  through  $\partial_t p_t = -\nabla \cdot (p_t v_t)$  is performed with initial and ending distributions of  $p_\lambda$  and  $p_1$ , respectively. We then define the following two-period transport model:

$$\min_{(v_t, g_t) \in \mathcal{C}_\lambda(p_0, p_1)} \mathcal{J}_{\text{tp}}^\lambda(v_t, g_t) \triangleq (1 - \lambda) \int_{\Omega} \int_{\lambda}^1 p_t(\mathbf{x}) \|v_t(\mathbf{x})\|^2 dt d\mathbf{x} + \mathcal{H}(v_t, g_t, p_t), \quad (5)$$

where  $\mathcal{H}(v_t, g_t, p_t) = \int_{\Omega} p_0(\mathbf{x}) (e^{\int_0^\lambda g_t(\mathbf{x}) dt} (\int_0^\lambda g_t(\mathbf{x}) dt - 1) + 1) dx$ ,  $\mathcal{C}_\lambda(p_0, p_1) = \{(v_t, g_t) : \partial_t p_t = g_t p_t, t \in [0, \lambda]; \partial_t p_t = -\nabla \cdot (p_t v_t), t \in (\lambda, 1]\}$ , and  $p_0$  and  $p_1$  are the given distributions. The following proposition shows the relation between the semi-relaxed optimal transport model in Eq. (4) and the two-period transport model in Eq. (5).

**Proposition 1.** Assume  $c(\mathbf{x}_0, \mathbf{x}_1) = \|\mathbf{x}_0 - \mathbf{x}_1\|^2$  and if we enforce  $P_{\#}^0 \pi$  and  $p_0$  to share the same support for admissible solution  $\pi$  to problem (4), then we have  $\min_{\pi} \mathcal{J}_{\text{sot}}(\pi) = \min_{v_t, g_t} \mathcal{J}_{\text{tp}}^\lambda(v_t, g_t)$ ,  $\forall \lambda \in (0, 1)$ . Moreover, for any  $\lambda \in (0, 1)$ , given the optimal transport plan  $\pi^*$  to problem (4), let  $p_\lambda^* \triangleq P_{\#}^0 \pi^*$ , then  $\pi^*$  can be expressed as  $\pi^* = (\text{Id}, T^*)_{\#} p_\lambda^*$  where  $T^*$  is the Monge map between  $p_\lambda^*$  and  $p_1$ . Meanwhile, there exist a  $g_t^*$  such that  $p_\lambda^* = p_0(\mathbf{x}) e^{\int_0^\lambda g_t^*(\mathbf{x}) dt}$ , and a  $v_t^*$  given by  $v_t^*(\mathbf{x} + \frac{t-\lambda}{1-\lambda}(T^*(\mathbf{x}) - \mathbf{x})) = \frac{T^*(\mathbf{x}) - \mathbf{x}}{1-\lambda}$ , satisfying  $(v_t^*, g_t^*) \in \arg \min_{v_t, g_t} \mathcal{J}_{\text{tp}}^\lambda(v_t, g_t)$ .



The proof is provided in the Appendix A.2. Proposition 1 indicates that problems (4) and (5) share the same optimal objective function value. Meanwhile, the optimal objective function value of the two-period transport problem (5) is independent of  $\lambda$ . Moreover,  $v_t^*, g_t^*$  can be constructed using  $\pi^*$ .

**Building unbalanced dynamics.** Although we have identified a scheme in which  $p_0$  evolves into  $p_1$  through a two-period process governed separately by  $v_t$  and  $g_t$  respectively, this decoupling does not align with the behavior observed in biological systems, where transport and growth typically occur simultaneously. To address this, we next build an unbalanced dynamics based on the dynamic understanding discussed above that allows velocity and growth to jointly drive the state transition and mass growth while ensuring consistency with the target distribution. Specifically, given  $v_t$  ( $t \in (\lambda, 1]$ ),  $g_t$  ( $t \in [0, \lambda]$ ) of the two-period process, we define the joint velocity  $\tilde{v}_t$  and growth  $\tilde{g}_t$  by

$$\begin{aligned}\tilde{v}_t(\mathbf{x}) &= (1 - \lambda) \cdot v_{(1-\lambda)t+\lambda}(\mathbf{x}), \\ \tilde{g}_t(\mathbf{x}) &= \lambda \cdot g_{\lambda t}(\psi_{\tilde{v},t}^{-1}(\mathbf{x})),\end{aligned}\tag{6}$$

where  $t \in [0, 1]$ , and  $\psi_{\tilde{v},t}$  is the flow generated by  $\tilde{v}$ , i.e.,  $\psi_{\tilde{v},t}(\mathbf{x}_0) = \mathbf{x}_0 + \int_0^t \tilde{v}_s(\mathbf{x}) ds$ .

**Theorem 1.** *Given the initial distribution  $p_0$ , denote the ending distribution of the two-period dynamics*

$$\partial_t p_t = g_t p_t, t \in [0, \lambda]; \quad \partial_t p_t = -\nabla \cdot (p_t v_t), t \in (\lambda, 1],\tag{7}$$

as  $p_1$ , and denote the ending distribution of the joint dynamics starting from  $p_0$

$$\partial_t \tilde{p}_t = -\nabla \cdot (\tilde{p}_t \tilde{v}_t) + \tilde{g}_t \tilde{p}_t, \quad t \in [0, 1], \quad \tilde{p}_0 = p_0,\tag{8}$$

as  $\tilde{p}_1$ , then it holds that  $\tilde{p}_1 = p_1$ .

The proof is detailed in Appendix A.3. Theorem 1 indicates that given the same initial distribution at  $t = 0$ , the two-period and the joint dynamics yield the same distribution at  $t = 1$ . Since  $\tilde{v}_t, \tilde{g}_t$  are defined on  $[0, 1]$ , the joint dynamics is more practical in applications, e.g., biological systems. Meanwhile, if  $v_t, g_t$  is defined as in Proposition 1,  $\tilde{v}_t, \tilde{g}_t$  will inherit the optimality properties of the semi-relaxed optimal transport, benefitting our flow matching algorithm, as shown in Sect. 3.3.

### 3.3 Velocity and Growth Flow Matching

Since the optimal growth function  $g_t^*$  in Proposition 1 is required to evolve  $p_0$  to  $P_{\#}^0 \pi^*$ , satisfying the marginal constraint  $p_0 \exp\left(\int_0^\lambda g_s^* ds\right) = P_{\#}^0 \pi^*$ , possible choice for  $g_t^*$  is not unique. For simplicity and ease of implementation, we choose a time-independent form:  $g_t^*(\mathbf{x}) = \frac{\log P_{\#}^0 \pi^*(\mathbf{x}) - \log p_0(\mathbf{x})}{\lambda}$ , which minimizes the  $L^2$ -norm energy functional of  $g$  and can be explained by the Malthusian growth model [58], i.e., the exponential growth model (see Appendix A.4). By plugging the expression of  $v_t^*$  and consists with from Proposition 1 and  $g_t^*$  into Eq. (6), we obtain

$$\tilde{v}_t(\psi_{\tilde{v},t}(\mathbf{x}_0)) = T^*(\mathbf{x}_0) - \mathbf{x}_0, \quad \tilde{g}_t(\psi_{\tilde{v},t}(\mathbf{x}_0)) = \log P_{\#}^0 \pi^*(\mathbf{x}_0) - \log p_0(\mathbf{x}_0), \text{ where } \mathbf{x}_0 \sim p_0.\tag{9}$$

In practice, only finite samples of  $p_0$  and  $p_1$  can be accessed. We then aim to learn neural networks  $v_\theta(\mathbf{x}, t), g_\omega(\mathbf{x}, t)$  to approximate  $\tilde{v}_t(\mathbf{x}), \tilde{g}_t(\mathbf{x})$  by velocity and growth flow matching as

$$\min_{\theta, \omega} \mathbb{E}_{\mathbf{x}_0} \mathbb{E}_t \left[ \|v_\theta(\psi_{\tilde{v},t}(\mathbf{x}_0), t) - \tilde{v}_t(\psi_{\tilde{v},t}(\mathbf{x}_0))\|^2 + |g_\omega(\psi_{\tilde{v},t}(\mathbf{x}_0), t) - \tilde{g}_t(\psi_{\tilde{v},t}(\mathbf{x}_0))|^2 \right],\tag{10}$$

that can generalize to new samples. We first estimate  $\pi^*$  and  $T^*$  using samples. Given samples  $\{\mathbf{x}_0^i\}_{i=1}^n$  of  $p_0$  and  $\{\mathbf{x}_1^j\}_{j=1}^m$  of  $p_1$ , We seek the optimal transport plan  $\pi^{0 \rightarrow 1} \approx \pi^*(\mathbf{x}_0, \mathbf{x}_1)$  by solving the entropy-regularized semi-relaxed transport problem using Sinkhorn-based algorithm [59, 60] as

$$\pi^{0 \rightarrow 1} = \arg \min_{\pi \geq 0} \sum_{i,j} c_{ij} \pi_{ij} + \epsilon H(\pi) + \tau \text{KL}(\pi \mathbf{1}_m || \mathbf{1}_n), \quad \text{subject to } \pi^\top \mathbf{1}_n = \mathbf{1}_m,\tag{11}$$

where  $c_{ij} = \|\mathbf{x}_0^i - \mathbf{x}_1^j\|^2$ ,  $\mathbf{1}_m$  is the all-one vector,  $H(\pi)$  denotes the negative entropy of  $\pi$ , and  $\epsilon$  and  $\tau$  are hyperparameters. With  $\pi^{0 \rightarrow 1}$ ,  $T^*$  can be estimated using barycentric mapping  $T^*(\mathbf{x}_0^i) \approx \frac{1}{N^i} \sum_j \pi_{ij}^{0 \rightarrow 1} \mathbf{x}_1^j$  where  $N^i = \sum_j \pi_{ij}^{0 \rightarrow 1}$ . In implementation, for each  $\mathbf{x}_0^i$ , we sample  $j$

from  $(1, 2, \dots, m)$  with probability  $\frac{1}{N^i} \pi_{ij}^{0 \rightarrow 1}$ , and approximate  $T^*(\mathbf{x}_0^i) \approx \mathbf{x}_1^j$ . This approximation is accurate as  $\epsilon \rightarrow 0$ . Therefore,  $\psi_{\tilde{v},t}(\mathbf{x}_0^i) \approx \mathbf{x}_0^i + t(\mathbf{x}_1^j - \mathbf{x}_0^i) \triangleq \mathbf{x}_t$  for  $t \in [0, 1]$ , and Eq. (10) becomes

$$\mathcal{L}_{\text{VGFM}}(\theta, \omega) = \sum_{i=1}^n \sum_{j=1}^m \pi_{ij}^{0 \rightarrow 1} \mathbb{E}_t \left[ \left\| v_\theta(\mathbf{x}_t, t) - (\mathbf{x}_1^j - \mathbf{x}_0^i) \right\|^2 + \left| g_\omega(\mathbf{x}_t, t) - \log([\pi^{0 \rightarrow 1} \mathbf{1}_m]_i) \right|^2 \right], \quad (12)$$

where  $[\cdot]_i$  is the  $i$ -th element. The last term in Eq. (12) is because  $p_0(\mathbf{x}_0^i) = 1$ , for  $i = 1, \dots, n$ . In experiments,  $\mathcal{L}_{\text{VGFM}}$  can be implemented using mini-batch samples. Specifically, we sample  $(i, j)$  from  $\pi^{0 \rightarrow 1}$  and  $t$  from  $\mathcal{U}(0, 1)$ , then calculate the loss in square brackets. To improve robustness, we add a Gaussian noise to  $\mathbf{x}_t$  in experiments.

### 3.4 Training Process

Our ultimate goal is to generate data close to the real data distribution. The current learned velocity field of flow matching is the expectation of the conditional velocity field, *i.e.*,  $v_\theta(x, t) = \mathbb{E}_{z|(x,t)} v_\theta(x, t|z)$ . In practice, since we use limited samples to learn the velocity field  $v$ , each numerical integration step introduces an approximation error, which can be accumulated during the numerical ODE solving process, resulting in a deviation from the true trajectory [61]. To address this bias, we employ distribution fitting loss besides the flow matching loss in Eq. (12), to improve performance further. Next, we introduce the distribution fitting loss and our training algorithm.

To achieve better performance, we incorporate the Wasserstein distance between generated samples and observed samples as part of the loss function. Specifically, let  $\mathcal{X}_t = \{\mathbf{x}_t^i\}_{i=1}^{N_t}$  for  $t = 0, 1, \dots, T-1$  denote the observed samples at different time points, and denote  $p(\mathcal{X}_t) = \sum_{i=1}^{N_t} \delta_{\mathbf{x}_t^i}$ . We define  $\phi_{v_\theta} : \mathbb{R}^d \times \mathcal{T} \rightarrow \mathbb{R}^{d|\mathcal{T}|}$  as the trajectory mapping function parameterized by the neural network  $v_\theta$ , which takes a given starting point as the initial condition and outputs particle coordinates at time indices  $\mathcal{T}$  according to ODE dynamics, where  $\mathcal{T}$  denotes a set of time steps. Given an initial set  $\mathcal{X}_0$ , the model  $\phi_{v_\theta}$  predicts particle positions at future time points in  $\mathcal{T}$ . Specifically, the predicted samples are computed by applying the neural ODE to  $\mathcal{X}_0$  over the time indices  $\{1, \dots, T-1\}$ , *i.e.*,  $\hat{\mathcal{X}}_1, \dots, \hat{\mathcal{X}}_{T-1} = \phi_{v_\theta}(\mathcal{X}_0, \{1, \dots, T-1\})$ . Similarly, we define  $\phi_{g_\omega}$  as the particle weight mapping function parameterized by the neural network  $g_\omega$ , which takes the initial weight of particle  $i$  as input and outputs the corresponding weight values at  $\mathcal{T}$  under ODE dynamics. For simplicity of notation, we use  $\hat{w}(\hat{\mathbf{x}})$  for  $\hat{\mathbf{x}} \in \hat{\mathcal{X}}_t$  to denote the weight generated by  $\phi_{g_\omega}$  corresponding to the particles in set  $\hat{\mathcal{X}}$  which is generated by  $\phi_{v_\theta}$ . The distribution fitting loss is defined as the Wasserstein distance as

$$\mathcal{L}_{\text{OT}}(\theta, \omega) = \sum_{t=1}^{T-1} \mathcal{W}_1 \left( \frac{1}{N_t} p(\mathcal{X}_t), \frac{1}{\sum_{\hat{\mathbf{x}} \in \hat{\mathcal{X}}_t} \hat{w}(\hat{\mathbf{x}})} p_{\hat{w}}(\hat{\mathcal{X}}_t) \right), \quad (13)$$

where  $p_{\hat{w}}(\hat{\mathcal{X}}_t) = \sum_{\hat{\mathbf{x}} \in \hat{\mathcal{X}}_t} \hat{w}(\hat{\mathbf{x}}) \delta_{\hat{\mathbf{x}}}$ ,  $\mathcal{W}_1$  is the 1-Wasserstein distance with Euclidean norm.

**Total loss and algorithm.** Note that the matching loss can be easily generalized to multi-time snapshots by employing Eq. (12) among two consecutive snapshots  $p_t$  and  $p_{t+1}$ . Combined with distribution fitting loss in Eq. (13), we obtain the final loss

$$\mathcal{L}(\theta, \omega) = \mathcal{L}_{\text{VGFM}}(\theta, \omega) + \mathcal{L}_{\text{OT}}(\theta, \omega) \quad (14)$$

During training, we employ a parameter scheduling scheme in which we initially use  $\mathcal{L}_{\text{VGFM}}$  as a warm-up stage, before enabling both loss terms for joint training. This design provides a notable advantage: after the warm-up stage,  $v_\theta$  and  $g_\omega$  are well-initialized through conditional matching, which facilitates convergence when switching to  $\mathcal{W}_1$ -based training. The detailed algorithm is presented in Algorithm 1. For further details on the warm-up procedure and parameter scheduling, please refer to Appendix B.1.

## 4 Experiments

We evaluate VGFM on synthetic and real datasets, and compare it with state-of-the-art approaches, including methods that account for distributional unbalancedness [23, 26, 57] as well as those that assume mass conservation [22, 38, 41]. Our code will be released online.

---

**Algorithm 1:** Training algorithm of joint velocity-growth flow matching

---

**Input:** Observed data  $\mathcal{X}_0, \dots, \mathcal{X}_{T-1}$ ; warm-up epochs  $M_1$ ; training epochs  $M_2$

**Output:** Trained velocity field  $v_\theta$  and growth function  $g_\omega$

Compute transport plans  $\pi^{t \rightarrow t+1}$  for  $t = 0, \dots, T-2$ ;

**for**  $i = 1$  **to**  $M_2$  **do**

**for**  $t_0 = 0$  **to**  $T-2$  **do**

        Sample a batch  $(i, j) \sim \pi^{t_0 \rightarrow t_0+1}$ ;

        Sample  $t \sim \mathcal{U}(0, 1) + t_0$ , and  $\mathbf{x}_t \sim \mathcal{N}(t\mathbf{x}_1^j + (1-t)\mathbf{x}_0^i, \sigma^2 I)$ ;

$\mathcal{L} \leftarrow \mathcal{L} + \left\| v_\theta(\mathbf{x}_t, t) - (\mathbf{x}_1^j - \mathbf{x}_0^i) \right\|^2 + \left| g_\omega(\mathbf{x}_t, t) - \log([\pi^{0 \rightarrow 1} \mathbf{1}_m]_i) \right|^2$ ;

**if**  $i > M_1$  **then**

$\hat{\mathcal{X}}_{t_0+1} \leftarrow \phi_{v_\theta}(\hat{\mathcal{X}}_{t_0}, t_0 + 1)$ ,  $\hat{w}_{t_0+1}(\hat{\mathcal{X}}_{t_0+1}) \leftarrow \phi_{g_\omega}(\hat{\mathcal{X}}_{t_0}, t_0 + 1)$ ;

$\mathcal{L} \leftarrow \mathcal{L} + \mathcal{W}_1 \left( \frac{1}{N_{t_0+1}} p(\mathcal{X}_{t_0+1}), \frac{1}{\sum_{\hat{\mathbf{x}} \in \hat{\mathcal{X}}_t} \hat{w}(\hat{\mathbf{x}})} p_{\hat{w}}(\hat{\mathcal{X}}_{t_0+1}) \right)$ ;

    Update  $\theta, \omega$  using  $\mathcal{L}$

---

**Synthetic datasets.** Inspired by [23, 26], we adopt the Simulation Gene dataset that applies a three-gene regulatory network to produce a quiescent region and an area exhibiting both transition and growth that can be observed. We also use Dynngen [62] to simulate a scRNA-seq dataset from a dynamic cellular process, which exhibits pronounced branching unbalancedness, with significantly different cell abundances across divergent lineages. Following the experiment setup of [31], we use PHATE [63] to reduce its dimensions to 5. Additionally, inspired by [26] that employs a high-dimensional Gaussian mixture model [64] to evaluate the scalability of models, we adopt a more challenging setting: 1000-dimensional Gaussian mixtures.

**Real-world dataset.** We conduct experiments on three real-world datasets, Embryoid Body (EB) [65], CITE-seq (CITE) [66] and Pancreas [67], preprocessed following the procedures in [18, 22, 37]. EB and CITE dataset are evaluated under both 5- and 50-dimensional PCA projections, while Pancreas dataset is evaluated under 2000 highly variable gene space to assess VGFM’s scalability on real-world data. Specifically, the EB (5D), CITE (5D), and CITE (50D) configurations are assessed using a hold-out strategy, same as [38], in which an intermediate time point is excluded during training. The model is then used to predict the distribution at the hold-out time, and we compute the  $\mathcal{W}_1$  distance between the predicted and true distributions at that time point. The EB (50D) setting is further used for comparison against [26] and ablation study, evaluated using both  $\mathcal{W}_1$  and the Relative Mass Error (RME) (see below). In the Pancreas dataset, in addition to  $\mathcal{W}_1$  and RME, we also present the mean–variance trends between generated and real gene expressions and analyze the interpretability of the learned growth function  $g$ .

**Evaluation metrics.** We follow [26] to use  $\mathcal{W}_1$  to measure the distance between the predicted cell distribution and the true cell distribution. Additionally, to assess the accuracy of the growth function  $g_\omega$ , we normalize the total mass at time 0 to 1, and denote the relative mass at time  $t$  w.r.t time 0 as  $m_t$ , i.e.,  $m_t = \frac{N_t}{N_0}$ . The predicted relative mass at time  $t$  based on the evolution induced by  $g_\omega$  is denoted as  $\hat{m}_t = \sum_{\hat{\mathbf{x}} \in \hat{\mathcal{X}}_t} \hat{w}(\hat{\mathbf{x}})/N_0$ , where  $\sum_{\hat{\mathbf{x}} \in \hat{\mathcal{X}}_t} \hat{w}(\hat{\mathbf{x}})$  is the predicted total mass at time  $t$ . We define the relative mass error, denoted as RME, i.e.,  $\text{RME} = \frac{|m_t - \hat{m}_t|}{m_t}$ .

**Implementation details.** We employ a 3-layer (5-layer for dimensions greater than 50) MLP with 256 hidden units and LeakyReLU activation to parameterize both the velocity field  $v_\theta$  and the growth function  $g_\omega$ . Optimization is performed using the Adam optimizer with a learning rate of  $10^{-3}$  at warm-up stage and  $10^{-4}$  after applying distribution fitting loss. A warm-up stage of 500 iterations for synthetic datasets and 5000 iterations for real-world dataset is applied (only matching loss in Eq. (12)), after which the distribution fitting loss  $\mathcal{L}_{\text{OT}}$  defined in Eq. (13) is applied for an additional 30 training epochs. We will discuss the strategy to set  $\epsilon$  and  $\tau$  in the Appendix B.1.

## 4.1 Results and Analysis

**Ability to reconstruct cellular dynamics.** As shown in Tab. 1, our model achieves lowest mean  $\mathcal{W}_1$  and RME on synthetic datasets, outperforming both unbalancedness-aware methods [23, 26, 57] and

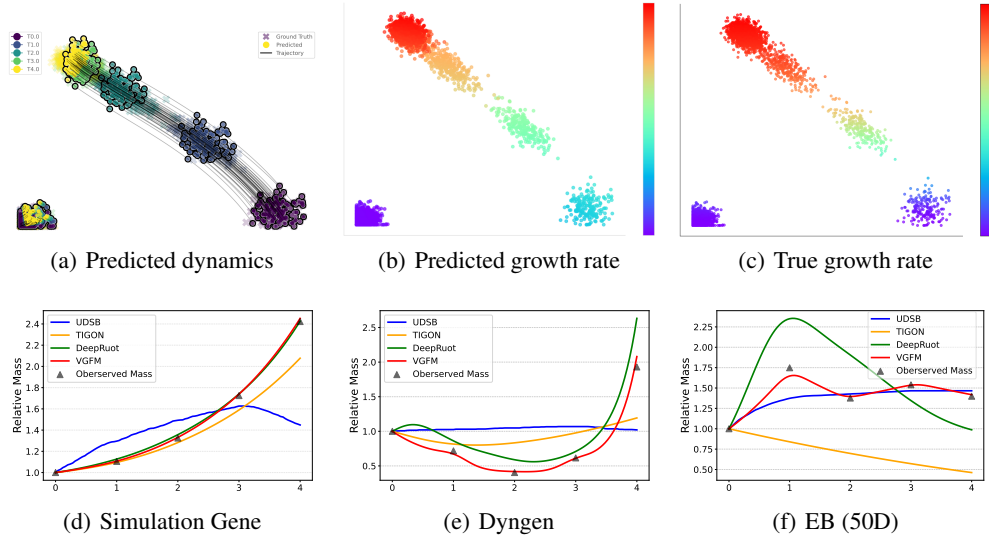


Figure 3: (a) Predicted dynamics with trajectories of VGFM on Simulation Gene. (b) Predicted and (c) true growth rates on Simulation Gene. Note that for Simulation Gene, the true growth rates could be accessed. (d), (e) and (f) respectively compare predicted relative mass by UDSB, TIGON, DeepRUOT, and VGFM, and observed relative mass from data, on three datasets. Note that in (f), on 50D dataset, the mass predicted by TIGON deviates significantly from the observed trend, because of its difficulty in handling high-dimensional data as discussed in their paper [23].

Table 1: Mean  $\mathcal{W}_1$  and RME over all time on synthetic datasets. \*OT-CFM and OT-MFM do not model the growth function, thus, RME is not computed. “N/C” means “not converge”.

Method	Simulation Gene (2D)		Dyngen (5D)		Gaussian (1000D)	
	$\mathcal{W}_1$	RME	$\mathcal{W}_1$	RME	$\mathcal{W}_1$	RME
OT-CFM* [22]	0.302	—	3.926	—	10.126	—
OT-MFM* [38]	0.311	—	3.976	—	11.008	—
UDSB [57]	0.665	0.192	1.914	0.658	N/C	N/C
TIGON [23]	0.099	0.065	1.029	0.542	N/C	N/C
DeepRUOT [26]	0.068	0.016	0.474	0.199	N/C	N/C
<b>VGFM</b>	<b>0.046</b>	<b>0.006</b>	<b>0.420</b>	<b>0.053</b>	<b>3.010</b>	<b>0.037</b>

unbalancedness-ignoring methods [22, 38], which demonstrate the ability of VGFM to model cellular dynamics for data and mass fitting. We also visualize the learned trajectory and growth function on the Simulation Gene dataset in Figs. 3 (a-c). Figs. 3 (a-c) show that VGFM not only reconstructs accurate trajectories but also produces faithful predictions of spatially varying growth rates.

**Scalability and mass-matching performance compared with existing unbalanced models.** While existing SOTA approaches of UDSB [57], TIGON [23] and DeepRUOT [26] incorporate unbalanced modeling, these models either introduce biologically implausible transitions [57] (as discussed in [26]) or face difficulties in scaling to high-dimensions [23, 26, 57]. As a result, these methods do not perform well or converge on the 1000-dimensional Gaussian dataset, as in Tab. 1. In contrast, VGFM integrates flow matching based on unbalanced transport, allowing it to scale more effectively to high-dimensional datasets. Our method more accurately recovers the ground-truth dynamics defined in [26] and achieves superior mass-matching accuracy, as shown in Tab. 1 and Figs. 3 (d-f).

**Hold-one-out results on real-world datasets.** Table 2 compares hold-one-out results of different methods on EB (5D) and CITE (5D and 50D) datasets. It can be observed that VGFM outperforms most existing approaches in terms of  $\mathcal{W}_1$  distance. We attribute this to the integration of complementary strengths from both flow matching and simulation-based (Neural ODE) frameworks. On the one hand, flow matching provides a robust initialization for learning the velocity field and growth rate,

which helps maintain training stability after incorporating the distribution fitting loss  $\mathcal{L}_{OT}$ , leading to better performance than purely simulation-based models, as also shown in Fig. 4. On the other hand, since the flow matching objective serves only as an upper bound on the true Wasserstein distance [68], the introduction of  $\mathcal{L}_{OT}$  allows direct optimization of the distribution fitting loss, further enhancing model performance. Moreover, by explicitly modeling cell growth through a time-varying weight function, our framework generalizes beyond the conventional setting where Wasserstein distance is computed between unweighted discrete measures, thereby offering greater flexibility in evaluation and enabling more possible transport solutions.

**Pancreas dataset under 2000-dimensional gene space.** To further explore the scalability of VGFM, we applied our model and compared methods explicitly modeling  $g_t(x)$  [23, 26] to the Pancreas dataset with 2000 highly variable genes. We first observed that VGFM is the only method showing a steadily decreasing training loss, both for  $\mathcal{L}_{VGFM}$  and  $\mathcal{L}_{OT}$ , as shown in Fig. A-11. Next, following [69], we calculated the means and variances of the real and generated gene at day 15.5 and plotted the corresponding mean-variance trend and histograms. The results show that the generated samples closely follow the real data, as shown in Fig. A-15. Finally, we analyzed the learned  $g_w(x, t)$  and found that our model successfully reconstructs the unbalanced pattern and identifies key genes without being given any prior knowledge of the cell types. The results are reported in Fig. A-16, A-17 and A-18. For more details, please refer to Appendix B.8.

**Ablation study on loss terms.** To assess the contribution of each loss component in our framework, we perform an ablation study on the EB dataset with unwhitened 50-dimensional PCA features. Table 3 reports the performance across four time points. VGFM in general achieves the lowest  $\mathcal{W}_1$  and RME values (excluding time 1) compared with VGFM (w/o  $\mathcal{L}_{OT}$ ) and VGFM (w/o  $\mathcal{L}_{VGFM}$ ), demonstrating the effectiveness of both our joint velocity-growth matching loss and distribution fitting loss. Notably, by removing  $\mathcal{L}_{VGFM}$ , both  $\mathcal{W}_1$  and RME increase by a significant margin, implying the importance of our velocity and growth matching to our framework. Consistent with our motivation in Sect. 3.4, adding in distribution fitting loss  $\mathcal{L}_{OT}$  results in lower  $\mathcal{W}_1$  and RME, improved fitting ability to observed distributions.

Table 2: Mean hold-one-out results on EB and CITE datasets over hold-out times.

Method	EB	CITE	
	5D	5D	50D
OT-CFM [22]	0.790	0.882	38.756
SF <sup>2</sup> M [41]	0.793	0.920	38.524
UDSB [57]	1.206	2.023	44.168
OT-MFM [38]	0.713	<b>0.724</b>	<b>36.394</b>
DeepRUOT [26]	0.774	0.845	38.681
<b>VGFM</b>	<b>0.676</b>	0.745	37.386

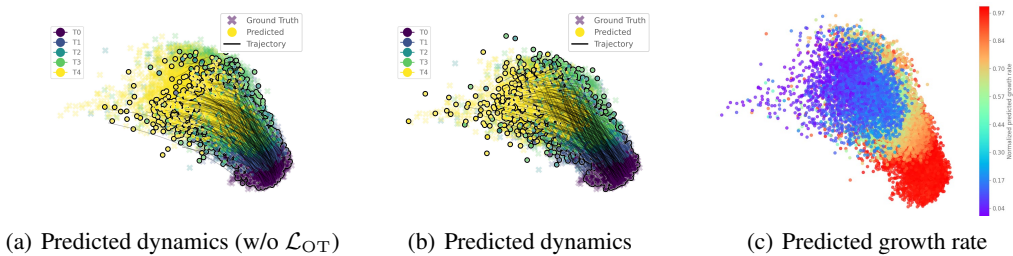


Figure 4: Visualization of predicted dynamics by (a) VFGM (w/o  $\mathcal{L}_{OT}$ ) and (b) VGFM, and growth rate by (c) VFGM, on EB (5D) dataset, where the hold-out time is the first intermediate timepoint.

**Comparison on computational efficiency.** We compare the computational cost of our method and the SOTA method DeepRUOT [26] in Tab. 3. For fair comparison, all methods in Tab. 3 adopt the same networks 5-layer MLP with 256 hidden units, and are trained on the same GPU until convergence. As shown in Table 3, our full model requires slightly more training time than the variant using only  $\mathcal{L}_{VGFM}$ , due to the training with additional distribution fitting loss. However, it still achieves faster convergence compared with simulation-based methods of DeepRUOT [26] and the ablation variant without  $\mathcal{L}_{VGFM}$ . This demonstrates the critical role of the loss  $\mathcal{L}_{VGFM}$  for facilitating training convergence.



To make  $\mathcal{L}_{\text{OT}}$  a fully differentiable component in our training pipeline, we also employed the Sinkhorn divergence [70] to replace  $\mathcal{W}_1$  as the distribution fitting loss  $\mathcal{L}_{\text{OT}}$ , forming a variant of VGFM denoted as VGFM(\*) in Tab. 3. It can be observed that using the Sinkhorn divergence leads to a further improvement in the accuracy of dynamics reconstruction, while reducing the computational cost for distribution fitting. For more details, please refer to Appendix B.6.

Table 3: Ablation study on EB (50D) dataset. For comparison, we also report the results of the SOTA approach DeepRUOT [26] in this table.

Method	$t_1$		$t_2$		$t_3$		$t_4$		Training time
	$\mathcal{W}_1$	RME	$\mathcal{W}_1$	RME	$\mathcal{W}_1$	RME	$\mathcal{W}_1$	RME	
DeepRUOT [26]	8.169	0.416	9.041	0.415	9.348	0.119	9.808	0.296	90 (mins)
VGFM (w/o $\mathcal{L}_{\text{OT}}$ )	8.915	0.020	10.590	0.098	10.915	0.067	11.635	0.088	6 (mins)
VGFM (w/o $\mathcal{L}_{\text{VGFM}}$ )	8.644	0.650	10.167	0.710	11.052	0.823	11.530	0.862	62 (mins)
VGFM	7.951	0.089	<b>8.747</b>	0.042	9.244	<b>0.019</b>	9.620	<b>0.044</b>	13 (mins)
VGFM (*)	<b>7.902</b>	<b>0.018</b>	8.767	<b>0.013</b>	<b>9.063</b>	0.083	<b>9.507</b>	0.096	9 (mins)

## 5 Conclusion, Limitation, and Future Work

The paper proposes the joint Velocity-Growth Flow Matching (VGFM) method that jointly learns state transition and mass growth of single-cell populations via flow matching. VGFM designs an ideal dynamics containing a velocity field and a growth function, driven by our presented two-period dynamic understanding of the static semi-relaxed optimal transport model. Approximating the ideal dynamics using neural networks yields the velocity-growth joint flow matching framework.

Although our approach achieves better scalability and training efficiency compared to simulation-based methods, it is not entirely simulation-free due to the incorporation of the distribution fitting loss  $\mathcal{L}_{\text{OT}}$ . Moreover, since the learned growth rate relies on the observed cell counts at each time point, it may inadvertently capture effects that are not solely attributable to biological growth. Future work could address them by integrating biological priors into the learning process of the growth function  $g$ , and by exploring fully simulation-free alternatives that can achieve comparable performance.

## Acknowledgment

This work was supported by the National Key R&D Program 2021YFA1003002, NSFC (12125104, 623B2084, 12501709, 12426313), China National Postdoctoral Program for Innovative Talents (BX20240276), China Fundamental Research Funds for the Central Universities (xzy022025047) and China Postdoctoral Science Foundation (2025M773058).

## References

- [1] V Gontis, J Ruseckas, and A Kononovičius. A long-range memory stochastic model of the return in financial markets. *Physica A: Statistical Mechanics and its Applications*, 389(1):100–106, 2010.
- [2] Christian LE Franzke, Terence J O’Kane, Judith Berner, Paul D Williams, and Valerio Lucarini. Stochastic climate theory and modeling. *Wiley Interdisciplinary Reviews: Climate Change*, 6(1):63–78, 2015.
- [3] 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.
- [4] 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.

- [5] Stephen Zhang, Anton Afanassiev, Laura Greenstreet, Tetsuya Matsumoto, and Geoffrey Schiebinger. Optimal transport analysis reveals trajectories in steady-state systems. *PLoS Computational Biology*, 17(12):e1009466, 2021.
- [6] Conrad H Waddington. Canalization of development and the inheritance of acquired characters. *Nature*, 150(3811):563–565, 1942.
- [7] Charlotte Bunne, Stefan G Stark, Gabriele Gut, Jacobo Sarabia Del Castillo, Mitch Levesque, Kjong-Van Lehmann, Lucas Pelkmans, Andreas Krause, and Gunnar Rätsch. Learning single-cell perturbation responses using neural optimal transport. *Nature Methods*, 20(11):1759–1768, 2023.
- [8] James A Hay, Lee Kennedy-Shaffer, Sanjat Kanjilal, Niall J Lennon, Stacey B Gabriel, Marc Lipsitch, and Michael J Mina. Estimating epidemiologic dynamics from cross-sectional viral load distributions. *Science*, 373(6552):eabh0635, 2021.
- [9] Jim Oeppen and James W Vaupel. Broken limits to life expectancy. *Science*, 296(5570):1029–1031, 2002.
- [10] 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.
- [11] Alessandro Palma, Sergei Rybakov, Leon Hetzel, Stephan Günnemann, and Fabian J Theis. Enforcing latent euclidean geometry in single-cell VAEs for manifold interpolation. In *ICML*, 2025.
- [12] Evan Z Macosko, Anindita Basu, Rahul Satija, James Nemesh, Karthik Shekhar, Melissa Goldman, Itay Tirosh, Allison R Bialas, Nolan Kamitaki, Emily M Martersteck, et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5):1202–1214, 2015.
- [13] Allon M Klein, Linas Mazutis, Ilke Akartuna, Naren Tallapragada, Adrian Veres, Victor Li, Leonid Peshkin, David A Weitz, and Marc W Kirschner. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 161(5):1187–1201, 2015.
- [14] Dragomirka Jovic, Xue Liang, Hua Zeng, Lin Lin, Fengping Xu, and Yonglun Luo. Single-cell rna sequencing technologies and applications: A brief overview. *Clinical and Translational Medicine*, 12(3):e694, 2022.
- [15] Antoine-Emmanuel Saliba, Alexander J Westermann, Stanislaw A Gorski, and Jörg Vogel. Single-cell rna-seq: advances and future challenges. *Nucleic Acids Research*, 42(14):8845–8860, 2014.
- [16] Ashraful Haque, Jessica Engel, Sarah A Teichmann, and Tapio Lönnberg. A practical guide to single-cell rna-sequencing for biomedical research and clinical applications. *Genome Medicine*, 9:1–12, 2017.
- [17] David Lähnemann, Johannes Köster, Ewa Szczurek, Davis J McCarthy, Stephanie C Hicks, Mark D Robinson, Catalina A Vallejos, Kieran R Campbell, Niko Beerenwinkel, Ahmed Mahfouz, et al. Eleven grand challenges in single-cell data science. *Genome Biology*, 21:1–35, 2020.
- [18] Alexander Tong, Jessie Huang, Guy Wolf, David Van Dijk, and Smita Krishnaswamy. Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics. In *ICML*, 2020.
- [19] Charlotte Bunne, Laetitia Papaxanthos, Andreas Krause, and Marco Cuturi. Proximal optimal transport modeling of population dynamics. In *AISTATS*, 2022.
- [20] Zhenyi Zhang, Zihan Wang, Yuhao Sun, Tiejun Li, and Peijie Zhou. Modeling cell dynamics and interactions with unbalanced mean field schrödinger bridge. *arXiv preprint arXiv:2505.11197*, 2025.

- [21] 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.
- [22] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguette, 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*, pages 1–34, 2024.
- [23] 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.
- [24] 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, 2024.
- [25] Antonio Terpin, Nicolas Lanzetti, Martín Gadea, and Florian Dorfler. Learning diffusion at lightspeed. In *NeurIPS*, 2024.
- [26] Zhenyi Zhang, Tiejun Li, and Peijie Zhou. Learning stochastic dynamics from snapshots through regularized unbalanced optimal transport. In *ICLR*, 2025.
- [27] Xi Zhang, Yuan Pu, Yuki Kawamura, Andrew Loza, Yoshua Bengio, Dennis Shung, and Alexander Tong. Trajectory flow matching with applications to clinical time series modelling. In *NeurIPS*, 2024.
- [28] Anming Gu, Edward Chien, and Kristjan Greenewald. Partially observed trajectory inference using optimal transport and a dynamics prior. In *ICLR*, 2025.
- [29] Martin Rohbeck, Charlotte Bunne, Edward De Brouwer, Jan-Christian Huetter, Anne Biton, Kelvin Y. Chen, Aviv Regev, and Romain Lopez. Modeling complex system dynamics with flow matching across time and conditions. In *ICLR*, 2025.
- [30] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *NeurIPS*, 2018.
- [31] Guillaume Huguette, Daniel Sumner Magruder, Alexander Tong, Oluwadamilola Fasina, Manik Kuchroo, Guy Wolf, and Smita Krishnaswamy. Manifold interpolating optimal-transport flows for trajectory inference. In *NeurIPS*, 2022.
- [32] Takeshi Koshizuka and Issei Sato. Neural lagrangian schrödinger bridge: Diffusion modeling for population dynamics. In *ICLR*, 2023.
- [33] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023.
- [34] Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.
- [35] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *ICLR*, 2023.
- [36] 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. In *ICLR*, 2023.
- [37] 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 *ICLR*, 2024.
- [38] Kacper Kapusniak, Peter Potapchik, Teodora Reu, Leo Zhang, Alexander Tong, Michael Bronstein, Joey Bose, and Francesco Di Giovanni. Metric flow matching for smooth interpolations on the data manifold. In *NeurIPS*, 2024.

- [39] 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 *ICML 2024 Workshop on Geometry-grounded Representation Learning and Generative Modeling*, 2024.
- [40] Dominik Klein, Théo Uscidda, Fabian J Theis, and marco cuturi. GENOT: Entropic (gromov) wasserstein flow matching with applications to single-cell genomics. In *NeurIPS*, 2024.
- [41] Alexander Y Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguët, Guy Wolf, and Yoshua Bengio. Simulation-free schrödinger bridges via score and flow matching. In *AISTATS*, 2024.
- [42] Gabriele Corso, Vignesh Ram Somnath, Noah Getz, Regina Barzilay, Tommi Jaakkola, and Andreas Krause. Composing unbalanced flows for flexible docking and relaxation. In *ICLR*, 2025.
- [43] Alessandro Palma, Till Richter, Hanyi Zhang, Manuel Lubetzki, Alexander Tong, Andrea Dittadi, and Fabian J Theis. Multi-modal and multi-attribute generation of single cells with cfgen. In *ICLR*, 2025.
- [44] Kirill Neklyudov, Rob Brekelmans, Daniel Severo, and Alireza Makhzani. Action matching: Learning stochastic dynamics from samples. In *ICML*, 2023.
- [45] Kirill Neklyudov, Rob Brekelmans, Alexander Tong, Lazar Atanackovic, qiang liu, and Alireza Makhzani. A computational framework for solving wasserstein lagrangian flows. In *ICML*, 2024.
- [46] Lenaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Unbalanced optimal transport: Dynamic and kantorovich formulations. *Journal of Functional Analysis*, 274(11):3090–3123, 2018.
- [47] Matthias Liero, Alexander Mielke, and Giuseppe Savaré. Optimal entropy-transport problems and a new hellinger–kantorovich distance between positive measures. *Inventiones Mathematicae*, 211(3):969–1117, 2018.
- [48] Leonid V Kantorovich. On the translocation of masses. In *Dokl. Akad. Nauk. USSR (NS)*, 1942.
- [49] Xiaole Tang, Xin Hu, Xiang Gu, and Jian Sun. Residual-conditioned optimal transport: towards structure-preserving unpaired and paired image restoration. In *ICML*, 2024.
- [50] Xiaole Tang, Xiang Gu, Xiaoyi He, Xin Hu, and Jian Sun. Degradation-aware residual-conditioned optimal transport for unified image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.
- [51] Xiang Gu, Liwei Yang, Jian Sun, and Zongben Xu. Optimal transport-guided conditional score-based diffusion model. In *NeurIPS*, 2023.
- [52] Nikita Gushchin, Alexander Kolesov, Alexander Korotin, Dmitry P. Vetrov, and Evgeny Burnaev. Entropic neural optimal transport via diffusion processes. In *NeurIPS*, 2023.
- [53] Alexander Korotin, Daniil Selikhanovych, and Evgeny Burnaev. Neural optimal transport. In *The Eleventh International Conference on Learning Representations*, 2023.
- [54] Taoran Zheng, Xing Li, Yan Yang, Xiang Gu, Zongben Xu, and Jian Sun. Towards prospective medical image reconstruction via knowledge-informed dynamic optimal transport. *arXiv preprint arXiv:2505.17644*, 2025.
- [55] Yann Brenier. Décomposition polaire et réarrangement monotone des champs de vecteurs. *CR Acad. Sci. Paris Sér. I Math.*, 305:805–808, 1987.
- [56] Jaemoo Choi and Jaewoong Choi. Scalable simulation-free entropic unbalanced optimal transport. *arXiv preprint arXiv:2410.02656*, 2024.
- [57] Matteo Pariset, Ya-Ping Hsieh, Charlotte Bunne, Andreas Krause, and Valentin De Bortoli. Unbalanced diffusion schrödinger bridge. *arXiv preprint arXiv:2306.09099*, 2023.

- [58] Thomas Robert Malthus. *An Essay on the Principle of Population*. 1798.
- [59] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, 2013.
- [60] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.
- [61] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *ICLR*, 2025.
- [62] Robrecht Cannoodt, Wouter Saelens, Louise Deconinck, and Yvan Saeys. Spearheading future omics analyses using dyngen, a multi-modal simulator of single cells. *Nature Communications*, 12(1):3942, 2021.
- [63] 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.
- [64] Lars Ruthotto, Stanley J Osher, Wuchen Li, Levon Nurbekyan, and Samy Wu Fung. A machine learning framework for solving high-dimensional mean field game and mean field control problems. *Proceedings of the National Academy of Sciences*, 117(17):9183–9193, 2020.
- [65] Kevin R Moon, Jay S Stanley III, Daniel Burkhardt, David van Dijk, Guy Wolf, and Smita Krishnaswamy. Manifold learning-based methods for analyzing single-cell rna-sequencing data. *Current Opinion in Systems Biology*, 7:36–46, 2018.
- [66] 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.
- [67] Aimée Bastidas-Ponce, Sophie Tritschler, Leander Dony, Katharina Scheibner, Marta Tarquis-Medina, Ciro Salinno, Silvia Schirge, Ingo Burtcher, Anika Böttcher, Fabian J Theis, et al. Comprehensive single cell mrna profiling reveals a detailed roadmap for pancreatic endocrinogenesis. *Development*, 146(12):dev173849, 2019.
- [68] Joe Benton, George Deligiannidis, and Arnaud Doucet. Error bounds for flow matching methods. *Transactions on Machine Learning Research*, 2024.
- [69] Alessandro Palma, Till Richter, Hanyi Zhang, Manuel Lubetzki, Alexander Tong, Andrea Dittadi, and Fabian J Theis. Multi-modal and multi-attribute generation of single cells with cfgen. In *ICLR*, 2025.
- [70] Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trounev, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *AISTATS*, 2019.
- [71] Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.
- [72] Alessio Figalli and Federico Glaudo. *An invitation to optimal transport, Wasserstein distances, and gradient flows*. EMS Press, 2021.
- [73] 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.



## Outline of Appendix

The appendix is organized into three main parts.

Appendix A presents the theoretical foundations. We begin by reviewing the Brenier–Benamou theorem [71], followed by detailed proofs of Proposition 1 and Theorem 1 that are stated in the main text, as well as the justification for our choice of the reparameterized growth function  $\tilde{g}$  in Eq. (9).

Appendix B focuses on experiments. We first describe our general implementation setup, including the strategy for selecting the hyperparameters in Eq. (11). Next, we provide detailed settings, visualizations and analysis for each experiment.

Appendix C discusses the broader impacts of our work.

## A Proofs

### A.1 Background: Brenier-Benamou Formulation

We first recall the dynamic formulation of balanced optimal transport, which is known as Brenier-Benamou formula [71]. In this paper, we assume  $p_0$  and  $p_1$  are absolutely continuous w.r.t. Lebesgue measure, which is omitted for convenience of description.

**Theorem A-2** (Brenier-Benamou formula). *Given two probability measures  $p_0, p_1 \in \mathcal{P}_2(\Omega)$ , it holds that*

$$\mathcal{W}_2^2(p_0, p_1) = \inf_{p_t, v_t} \left\{ \int_0^1 \int_{\Omega} \|v_t(\mathbf{x})\|^2 p_t(\mathbf{x}) \, d\mathbf{x} dt \mid \partial_t p_t = -\nabla \cdot (p_t v_t), p_{t=0} = p_0, p_{t=1} = p_1 \right\} \quad (\text{A-15})$$

and the optimal  $v_t^*(\mathbf{x})$  can be expressed by Monge map between  $p_0$  and  $p_1$ . i.e.,

$$v_t(\mathbf{x} + t(T^*(\mathbf{x}) - \mathbf{x})) = T^*(\mathbf{x}) - \mathbf{x}, \mathbf{x} \sim p_0 \quad (\text{A-16})$$

### A.2 Proof of Proposition 1

For the convenience of reading, here we restate Proposition 1 in the main text as Proposition A-2.

**Proposition A-2.** *Assume  $c(\mathbf{x}_0, \mathbf{x}_1) = \|\mathbf{x}_0 - \mathbf{x}_1\|^2$  and if we enforce  $P_{\#}^0 \pi$  and  $p_0$  to share the same support for admissible solution  $\pi$  to problem (4), then we have  $\min_{\pi} \mathcal{J}_{\text{sot}}(\pi) = \min_{v_t, g_t} \mathcal{J}_{\text{tp}}^{\lambda}(v_t, g_t), \forall \lambda \in (0, 1)$ . Moreover, for any  $\lambda \in (0, 1)$ , given the optimal transport plan  $\pi^*$  to problem (4), let  $p_{\lambda}^* \triangleq P_{\#}^0 \pi^*$ , then  $\pi^*$  can be expressed as  $\pi^* = (\text{Id}, T^*)_{\#} p_{\lambda}^*$  where  $T^*$  is the Monge map between  $p_{\lambda}^*$  and  $p_1$ . Meanwhile, there exist a  $g_t^*$  such that  $p_{\lambda}^* = p_0(\mathbf{x}) e^{\int_0^{\lambda} g_t^*(\mathbf{x}) dt}$ , and a  $v_t^*$  given by*

$$v_t^* \left( \mathbf{x} + \frac{t - \lambda}{1 - \lambda} (T^*(\mathbf{x}) - \mathbf{x}) \right) = \frac{T^*(\mathbf{x}) - \mathbf{x}}{1 - \lambda}, \quad (\text{A-17})$$

satisfying  $(v_t^*, g_t^*) \in \arg \min_{v_t, g_t} \mathcal{J}_{\text{tp}}^{\lambda}(v_t, g_t)$ .

*Proof.* Recall that when  $c(\mathbf{x}_0, \mathbf{x}_1) = \|\mathbf{x}_0 - \mathbf{x}_1\|^2$ , problems (4) and (5) are respectively

$$\min_{\pi \geq 0} \mathcal{J}_{\text{sot}}(\pi) \triangleq \int_{\Omega^2} \|\mathbf{x}_0 - \mathbf{x}_1\|^2 \, d\pi(\mathbf{x}_0, \mathbf{x}_1) + \text{KL}(P_{\#}^0 \pi \| p_0) \quad \text{subject to} \quad P_{\#}^1 \pi = p_1,$$

and

$$\min_{(v_t, g_t) \in \mathcal{C}_{\lambda}(p_0, p_1)} \mathcal{J}_{\text{tp}}^{\lambda}(v_t, g_t) \triangleq (1 - \lambda) \int_{\Omega} \int_{\lambda}^1 p_t(\mathbf{x}) \|v_t(\mathbf{x})\|^2 dt d\mathbf{x} + \mathcal{H}(v_t, g_t, p_t),$$

where  $\mathcal{H}(v_t, g_t, p_t) = \int_{\Omega} p_0(\mathbf{x}) (e^{\int_0^{\lambda} g_t(\mathbf{x}) dt} (\int_0^{\lambda} g_t(\mathbf{x}) dt - 1) + 1) d\mathbf{x}$ ,  $\mathcal{C}_{\lambda}(p_0, p_1) = \{(v_t, g_t) : \partial_t p_t = g_t p_t, t \in [0, \lambda]; \partial_t p_t = -\nabla \cdot (p_t v_t), t \in (\lambda, 1]\}$

(1) We first prove  $\min_{\pi} \mathcal{J}_{\text{sot}}(\pi) \geq \min_{v_t, g_t} \mathcal{J}_{\text{tp}}^{\lambda}(v_t, g_t)$ .

Let  $\pi^*$  be the optimal solution to problem (4). For any  $\lambda \in (0, 1)$ , define  $p_\lambda = P_{\#}^0 \pi^*$ , and let  $w_\lambda(\mathbf{x}) := \frac{p_\lambda(\mathbf{x})}{p_0(\mathbf{x})}$ . Define the growth as

$$\frac{d}{dt} \log w_t(\mathbf{x}) = g_t(\mathbf{x}), t \in [0, \lambda].$$

This implies the continuity equation  $\partial_t p_t = g_t p_t$  for  $t \in [0, \lambda]$ , and hence we have  $p_0(\mathbf{x}) \exp\left(\int_0^\lambda g_t(\mathbf{x}) dt\right) = P_{\#}^0 \pi^*$ .

By Brenier's theorem [60], we have  $\pi^* = (\text{Id}, T^*)_{\#} p_\lambda$ , where  $T^*(\mathbf{x})$  is the Monge map from  $p_\lambda$  to  $p_1$ . Define

$$g_t(\mathbf{x}) = \frac{\log p_\lambda(\mathbf{x}) - \log p_0(\mathbf{x})}{\lambda}, \quad v_t(\mathbf{x} + \frac{t - \lambda}{1 - \lambda}(T^*(\mathbf{x}) - \mathbf{x})) = \frac{T^*(\mathbf{x}) - \mathbf{x}}{1 - \lambda}. \quad (\text{A-18})$$

Plugging into the dynamic objective and using the definition of KL-divergence:

$$\text{KL}(p_\lambda \| p_0) = \int_{\Omega} p_0(\mathbf{x}) \left( \frac{p_\lambda(\mathbf{x})}{p_0(\mathbf{x})} \left( \log \frac{p_\lambda(\mathbf{x})}{p_0(\mathbf{x})} - 1 \right) + 1 \right) d\mathbf{x},$$

we obtain

$$\begin{aligned} \mathcal{J}_{\text{tp}}^\lambda(v_t, g_t) &= \int_{\Omega} p_0(\mathbf{x}) \left( e^{\int_0^\lambda g_t(\mathbf{x}) dt} \left( \int_0^\lambda g_t(\mathbf{x}) dt - 1 \right) + 1 \right) d\mathbf{x} + (1 - \lambda) \int_{\Omega} \|T^*(\mathbf{x}) - \mathbf{x}\|^2 p_\lambda(\mathbf{x}) d\mathbf{x} \\ &= \text{KL}(p_\lambda \| p_0) + \int_{\Omega^2} \|\mathbf{x}_0 - \mathbf{x}_1\|^2 d\pi^*(\mathbf{x}_0, \mathbf{x}_1) \\ &= \mathcal{J}_{\text{sot}}(\pi^*) \end{aligned}$$

Thus,  $\mathcal{J}_{\text{sot}}(\pi^*) = \mathcal{J}_{\text{tp}}^\lambda(v_t, g_t) \geq \min_{v_t, g_t} \mathcal{J}_{\text{tp}}^\lambda(v_t, g_t)$ .

(2) To show the reverse inequality  $\min_{\pi} \mathcal{J}_{\text{sot}}(\pi) \leq \min_{v_t, g_t} \mathcal{J}_{\text{tp}}^\lambda(v_t, g_t)$ , we assume the contrary that  $\min_{\pi} \mathcal{J}_{\text{sot}}(\pi) > \min_{v_t, g_t} \mathcal{J}_{\text{tp}}^\lambda(v_t, g_t)$ . Let  $(v_t^*, g_t^*)$  be an optimal solution to problem (5), and define

$$\tilde{p}_\lambda(\mathbf{x}) := p_0(\mathbf{x}) \exp\left(\int_0^\lambda g_t^*(\mathbf{x}) dt\right),$$

Since the second-stage evolution  $p_t$ , for  $t \in (\lambda, 1]$ , is governed solely by the velocity field  $v_t$  and does not involve mass creation or destruction, both  $\tilde{p}_\lambda$  and  $p_1$  have the same total mass. Thus, the Monge map  $\tilde{T}^*$  from  $\tilde{p}_\lambda$  to  $p_1$  under quadratic cost is well-defined. Then, by the Benamou–Brenier formulation [71, 72], we have

$$v_t^*(\mathbf{x} + \frac{t - \lambda}{1 - \lambda}(\tilde{T}^*(\mathbf{x}) - \mathbf{x})) = \frac{\tilde{T}^*(\mathbf{x}) - \mathbf{x}}{1 - \lambda},$$

and the corresponding coupling  $\tilde{\pi}^* := (\text{Id}, \tilde{T}^*)_{\#} \tilde{p}_\lambda$  satisfies

$$\begin{aligned} \mathcal{J}_{\text{sot}}(\tilde{\pi}^*) &= \text{KL}(\tilde{p}_\lambda \| p_0) + \int_{\Omega^2} \|\mathbf{x}_0 - \mathbf{x}_1\|^2 d\tilde{\pi}^*(\mathbf{x}_0, \mathbf{x}_1) \\ &= \int_{\Omega} p_0(\mathbf{x}) \left( e^{\int_0^\lambda g_t^*(\mathbf{x}) dt} \left( \int_0^\lambda g_t^*(\mathbf{x}) dt - 1 \right) + 1 \right) d\mathbf{x} + (1 - \lambda) \int_{\Omega} \|v_t^*(\mathbf{x})\|^2 \tilde{p}_\lambda(\mathbf{x}) d\mathbf{x} \\ &= \mathcal{J}_{\text{tp}}^\lambda(v_t^*, g_t^*) = \min_{v_t, g_t} \mathcal{J}_{\text{tp}}^\lambda(v_t, g_t) < \min_{\pi} \mathcal{J}_{\text{sot}}(\pi) \end{aligned}$$

which leads to the contradiction.

Combining (1) and (2), we have  $\mathcal{J}_{\text{sot}}(\pi^*) = \mathcal{J}_{\text{tp}}^\lambda(v_t^*, g_t^*)$ .

Given  $\pi^*$ , we construct  $g_t^*, v_t^*$  as in Eq. (A-18). According to the proof of (1), we have  $\mathcal{J}_{\text{tp}}^\lambda(v_t^*, g_t^*) = \mathcal{J}_{\text{sot}}(\pi^*) = \min_{v_t, g_t} \mathcal{J}_{\text{tp}}^\lambda(v_t, g_t)$ .  $\square$

### A.3 Proof and Empirical Evidence of Theorem 1

For the convenience of reading, here we restate Theorem 1 in the main text as Theorem A-3.

**Theorem A-3.** *Given the initial distribution  $p_0$ , denote the ending distribution of the two-period dynamics*

$$\partial_t p_t = g_t p_t, t \in [0, \lambda]; \quad \partial_t p_t = -\nabla \cdot (p_t v_t), t \in (\lambda, 1], \quad (\text{A-19})$$

*as  $p_1$ , and denote the ending distribution of the joint dynamics starting from  $p_0$*

$$\partial_t \tilde{p}_t = -\nabla \cdot (\tilde{p}_t \tilde{v}_t) + \tilde{g}_t \tilde{p}_t, \quad t \in [0, 1], \quad \tilde{p}_0 = p_0, \quad (\text{A-20})$$

*as  $\tilde{p}_1$ , then it holds that  $\tilde{p}_1 = p_1$ .*

*Proof.* Given  $\mathbf{x}_0 \sim p_0$ , consider the two systems below.

*System I (original two-period transport dynamics):*

$$\begin{cases} \frac{d\mathbf{x}_t}{dt} = v_t(\mathbf{x}_t) \cdot \mathbb{I}_{(\lambda, 1]}(t), \\ \frac{d}{dt} \log w_t(\mathbf{x}_t) = g_t(\mathbf{x}_t) \cdot \mathbb{I}_{[0, \lambda]}(t), \end{cases} \quad (\text{A-21})$$

where  $w_t$  is the time-dependent weight function and  $\mathbb{I}_\Omega$  is the indicator function, for any function

$$f, f \cdot \mathbb{I}_\Omega(t) = \begin{cases} f, & \text{If } t \in \Omega, \\ 0, & \text{Otherwise.} \end{cases}$$

*System II (joint dynamics defined via reparameterization (6)):*

$$\begin{cases} \frac{d\mathbf{x}_t}{dt} = \tilde{v}_t(\mathbf{x}_t), \\ \frac{d}{dt} \log w_t(\mathbf{x}_t) = \tilde{g}_t(\mathbf{x}_t), \end{cases} \quad (\text{A-22})$$

with  $\tilde{w}_t$  being the time-dependent weight under joint dynamics. We first recall the definition  $\tilde{v}_t(\mathbf{x}) = (1 - \lambda) \cdot v_{(1-\lambda)t+\lambda}(\mathbf{x})$ ,  $\tilde{g}_t(\mathbf{x}) = \lambda \cdot g_{\lambda t}(\psi_{\tilde{v},t}^{-1}(\mathbf{x}))$ . To prove Theorem 1, it suffices to show that given the same initialization  $\mathbf{x}_0, w_0(\mathbf{x}_0)$ , the final state of system I ( $\mathbf{x}_1, w_1(\mathbf{x}_1)$ ) and II ( $\tilde{\mathbf{x}}_1, \tilde{w}_1(\tilde{\mathbf{x}}_1)$ ) are identical, *i.e.*,  $\tilde{\mathbf{x}}_1 = \mathbf{x}_1$  and  $\tilde{w}_1(\tilde{\mathbf{x}}_1) = w_1(\mathbf{x}_1)$ .

$$\begin{aligned} \tilde{\mathbf{x}}_1 &= \mathbf{x}_0 + \int_0^1 \tilde{v}_t(\mathbf{x}) dt \\ &= \mathbf{x}_0 + \int_0^1 v_{(1-\lambda)t+\lambda}(\mathbf{x}) \cdot (1 - \lambda) dt \\ &= \mathbf{x}_0 + \int_\lambda^1 v_s(\mathbf{x}) ds \quad (\text{let } s = (1 - \lambda)t + \lambda) \\ &= \mathbf{x}_1. \end{aligned}$$

Meanwhile,

$$\begin{aligned} \log w_1(\mathbf{x}_1) &= \log w_\lambda(\mathbf{x}_\lambda) = \log w_\lambda(\mathbf{x}_0) \\ &= \log w_0(\mathbf{x}_0) + \int_0^\lambda g_s(\mathbf{x}_0) ds \\ &= \log w_0(\mathbf{x}_0) + \int_0^1 \lambda \cdot g_{\lambda t}(\mathbf{x}_0) dt \quad (\text{let } t = \frac{1}{\lambda}s) \\ &= \log w_0(\mathbf{x}_0) + \int_0^1 \lambda \cdot g_{\lambda t}(\psi_{\tilde{v},t}^{-1}(\mathbf{x})) dt \quad (\mathbf{x} = \mathbf{x}_0 + \int_0^t \tilde{v}_s(\mathbf{x}) ds) \\ &= \log w_0(\mathbf{x}_0) + \int_0^1 \tilde{g}_t(\mathbf{x}) dt \\ &= \log \tilde{w}_1(\tilde{\mathbf{x}}_1). \end{aligned}$$

Hence, the final state and mass at  $t = 1$  under both systems coincide. Applying the above analysis to all  $\mathbf{x}_0 \sim p_0$  completes the proof.  $\square$

**Empirical evidence of Theorem 1.** We provide the following example as an evidence to Theorem 1. Set  $p_0 = \mathcal{N}(2, 0.5)$ ,  $v_t(\mathbf{x}) = 2t$ ,  $g_t(\mathbf{x}) = -\log(\mathbf{x} + 1) + t^3$ , using Eq.(6), we have  $\tilde{v}_t(\mathbf{x}) = (1 - \lambda)v_{(1-\lambda)t+\lambda}(\mathbf{x})$ ,  $\tilde{g}_t(\mathbf{x}) = \lambda g_{\lambda t}(\psi_{\tilde{v}_t}^{-1}(\mathbf{x})) = \lambda g_{\lambda t}(\mathbf{x} - ((1 - \lambda)t + \lambda)^2 + \lambda^2)$ . By setting  $\lambda = 0.4$  and using numerical solvers to solve the corresponding continuity equation, we validate our correctness of our theorem. As shown in Fig. A-5, the two dynamics ended in the same distribution.

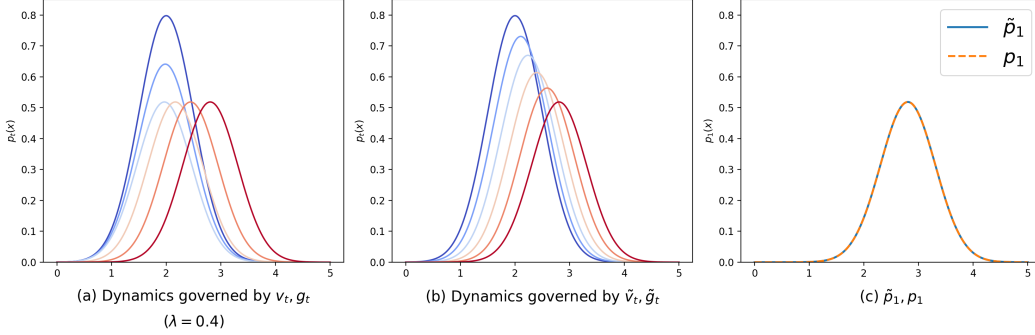


Figure A-5: Empirical evidence of Theorem 1. (a) Original two-period transport dynamics. (b) Joint dynamics defined via reparameterization 6. (c) The two dynamics ended in the same distribution.

#### A.4 Motivation for time-independent growth function $\tilde{g}_t$ according to Eq. (9)

**Minimization of  $L^2$  energy potential.** We aim to prove the following statement: Given a function  $w_t(\mathbf{x})$  whose logarithmic derivative satisfies  $\frac{d}{dt} \log w_t(\mathbf{x}) = g_t(\mathbf{x})$  and the boundary condition

$$\log w_1(\mathbf{x}) - \log w_0(\mathbf{x}) = \int_0^1 g_t(\mathbf{x}) dt,$$

then among all functions  $g_t(\mathbf{x})$  satisfying this constraint, the one minimizing the energy functional

$$\mathcal{E}(g) := \int_{\Omega} \int_0^1 \|g_t(\mathbf{x})\|_2^2 dt d\mathbf{x}$$

is the constant function  $\int_0^1 g_t(\mathbf{x}) dt = \log w_1(\mathbf{x}) - \log w_0(\mathbf{x})$ . To show this, we apply the method of Lagrange multipliers. Introduce a multiplier  $\lambda(\mathbf{x})$  and consider the Lagrangian

$$\mathcal{L}(g_t) = \int_{\Omega} \left( \int_0^1 (g_t(\mathbf{x})^2 + \lambda(\mathbf{x}) g_t(\mathbf{x})) dt - \lambda(\mathbf{x}) (\log w_1(\mathbf{x}) - \log w_0(\mathbf{x})) \right) d\mathbf{x}.$$

Taking the first variation of  $\mathcal{L}$  with respect to  $g_t(\mathbf{x})$  yields the optimality condition

$$\frac{\delta \mathcal{L}}{\delta g_t}(\mathbf{x}) = 2g_t(\mathbf{x}) + \lambda(\mathbf{x}) = 0 \Rightarrow g_t(\mathbf{x}) = -\frac{\lambda(\mathbf{x})}{2}.$$

This shows that the optimal  $g_t(\mathbf{x})$  is constant with respect to time. Plugging this into the constraint gives  $g_t(\mathbf{x}) = \log w_1(\mathbf{x}) - \log w_0(\mathbf{x})$ . Hence, the constant function  $g_t(\mathbf{x})$  is the unique minimizer of the energy functional under the given constraint.

**Explanation from the Malthusian growth model [58], i.e., the exponential growth model.** The Malthusian growth model is  $\frac{dp_t}{dt} = gp_t$ , where  $p_t$  is the population size and  $g$  is a constant growth rate. Our model on the growth  $\frac{\partial p_t}{\partial t} = g_t p_t$  is consistent with the above Malthusian growth model. In these models,  $g$  is treated as a constant based on the assumption that the resources are abundant and the environment is stable, causing growth rates to remain relatively stable over time. In the context of scRNA-seq experiments, these conditions are typically satisfied because cells are often cultured or sampled under controlled laboratory conditions, where nutrient supply, temperature, and other environmental factors are maintained at constant levels. Thus, choosing this form is biologically reasonable, especially when there is no prior knowledge about the growth rate.

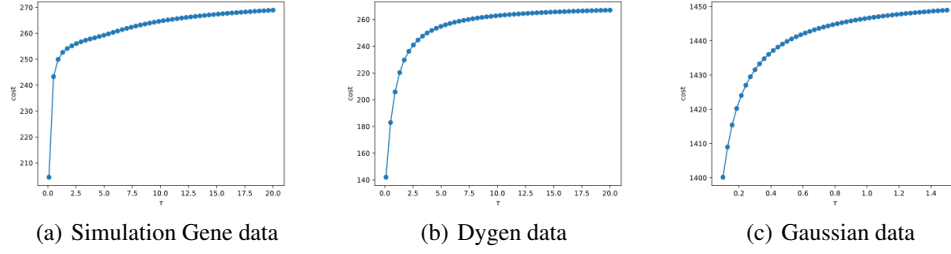


Figure A-6: Fixing  $\epsilon$ , we plot the transport cost as a function of  $\tau$ . (a) In Simulation Gene, the curve flattens around  $\tau = 5$ ; (b) In Dyngen, the transport cost also stabilizes near  $\tau = 5$ .

## B Experimental Details

The model architecture has been described in Sect. 4. We next detail the strategy for selecting the hyperparameters  $\epsilon$  and  $\tau$  in Eq. (11), as well as provide a comprehensive description of each experiment. All experiments are performed on a single-core CPU without GPU acceleration, and all visualizations are based on projections onto the first two dimensions of the high-dimensional data.

### B.1 Determine the Entropy Regularization Parameter $\epsilon$ and Relaxation Coefficient $\tau$

Our algorithm involves solving a static semi-relaxed optimal transport problem (11), which includes two critical hyperparameters  $\epsilon$  and  $\tau$ . These parameters play a pivotal role in the behavior and stability of the model. On one hand, if  $\epsilon$  is too small, it may lead to numerical instability; if it is too large, it can cause incorrect cross-branch matching in unbalanced data scenarios. To select  $\epsilon$ , we first set  $\tau$  to a moderately large value, such as  $\tau = 50$ , and then perform a grid search over increasing values of  $\epsilon$  starting from 0 until numerical stability is achieved. The choice of  $\tau$  is even more crucial. When  $\tau$  is too small, the KL divergence term between  $P_{\#}^0 \pi$  and  $p_0$  receives a weak penalty. As a result, the transport mass becomes overly concentrated around points close to  $p_1$ , leading to highly uneven marginals  $\pi \mathbf{1}_m$  in the discrete setting. This causes instability in the learning of the growth function  $g$  and may result in erroneous modeling. Conversely, as  $\tau \rightarrow +\infty$ , Eq. (11) reduces to a balanced OT problem, which contradicts the unbalanced nature of our formulation. To determine a suitable value for  $\tau$ , we fix the previously selected small  $\epsilon$  and gradually increase  $\tau$  while observing the variation in  $\sum \pi_{ij} c_{ij}$ . This curve typically exhibits an increasing-then-flattening trend. Similar to the “elbow metho” used in clustering to select the optimal number of clusters, we identify the region where the curve becomes stable and choose  $\tau$  accordingly, as illustrated in Fig. A-6.

On the *simulation gene* dataset, we fix  $\epsilon = 0.001$  and evaluate the model with different values of  $\tau \in \{5, 10, 15, 20\}$ , all of which lie within the identified stable region. The corresponding model performance metrics are summarized in Tab. A-4. We observe that the models trained with  $\tau$  in this range are stable, validating the effectiveness of our hyperparameter selection strategy.

Table A-4: Fix  $\epsilon = 0.001$ , change different  $\tau$  on simulation gene data, we present the Wasserstein-1 distance of each timepoint.

Models	$t_1$		$t_2$		$t_3$		$t_4$	
	$\mathcal{W}_1$	RME	$\mathcal{W}_1$	RME	$\mathcal{W}_1$	RME	$\mathcal{W}_1$	RME
VFGM ( $\tau = 5$ )	0.046	0.007	0.062	0.001	0.053	0.003	0.063	0.003
VFGM ( $\tau = 10$ )	0.041	0.005	0.053	0.005	0.038	0.007	0.040	0.008
VFGM ( $\tau = 15$ )	0.045	0.003	0.056	0.003	0.046	0.005	0.052	0.004
VFGM ( $\tau = 20$ )	0.043	0.007	0.057	0.004	0.045	0.007	0.058	0.011



## B.2 Simulation of Gene Data

Following the setup of [26], the dynamics of the simulated gene regulatory network are governed by the following system of differential equations:

$$\begin{aligned}\frac{dX_1}{dt} &= \frac{\alpha_1 X_1^2 + \beta}{1 + \alpha_1 X_1^2 + \gamma_2 X_2^2 + \gamma_3 X_3^2 + \beta} - \delta_1 X_1 + \eta_1 \xi_t, \\ \frac{dX_2}{dt} &= \frac{\alpha_2 X_2^2 + \beta}{1 + \gamma_1 X_1^2 + \alpha_2 X_2^2 + \gamma_3 X_3^2 + \beta} - \delta_2 X_2 + \eta_2 \xi_t, \\ \frac{dX_3}{dt} &= \frac{\alpha_3 X_3^2}{1 + \alpha_3 X_3^2} - \delta_3 X_3 + \eta_3 \xi_t,\end{aligned}\tag{A-23}$$

where  $X_i(t)$  denotes the expression level of gene  $i$  at time  $t$ . Genes  $X_1$  and  $X_2$  mutually inhibit each other while self-activating, forming a toggle-switch regulatory motif. An external signal  $\beta$  independently activates both  $X_1$  and  $X_2$ , whereas  $X_3$  inhibits the expression of both  $X_1$  and  $X_2$ .

The parameters  $\alpha_i$  and  $\gamma_i$  determine the strengths of self-activation and inhibition, respectively. The  $\delta_i$  terms denote degradation rates, and  $\eta_i \xi_t$  represents additive Gaussian noise with intensity  $\eta_i$ . The cell division rate is positively correlated with  $X_2$  expression and is calculated as

$$g = \frac{\alpha_2 X_2^2}{1 + X_2^2} \%.$$

At each cell division event, daughter cells inherit gene expression values  $(X_1, X_2, X_3)$  from the parent, subject to small perturbations  $\eta_d \mathcal{N}(0, 1)$  per gene. Post-division, cells evolve independently according to the same stochastic dynamics.

**Initial Conditions and Simulation Setup.** Initial expression states are sampled from two normal distributions:  $\mathcal{N}([2, 0.2, 0], 0.01)$  and  $\mathcal{N}([0, 0, 2], 0.01)$ . At every step, negative values are clipped to zero. Gene expression data is recorded at time points  $t \in \{0, 8, 16, 24, 32\}$ .

By setting  $\tau = 10$  and  $\epsilon = 0.003$  according to our selection scheme described in Appendix B.1 and illustrated in Fig. A-6 (a). Our model achieve the best performance in terms of (weighted)  $\mathcal{W}_1$  and relative mass error as shown in Tab. 1.

Table A-5: Simulation parameters on gene regulatory network [26].

Parameter	Value	Description
$\alpha_1$	0.5	Strength of self-activation for $X_1$
$\gamma_1$	0.5	Inhibition of $X_2$ by $X_1$
$\alpha_2$	1	Strength of self-activation for $X_2$
$\gamma_2$	1	Inhibition of $X_1$ by $X_2$
$\alpha_3$	1	Strength of self-activation for $X_3$
$\gamma_3$	10	Half-saturation constant for inhibition
$\delta_1$	0.4	Degradation rate for $X_1$
$\delta_2$	0.4	Degradation rate for $X_2$
$\delta_3$	0.4	Degradation rate for $X_3$
$\eta_1$	0.05	Noise intensity for $X_1$
$\eta_2$	0.05	Noise intensity for $X_2$
$\eta_3$	0.05	Noise intensity for $X_3$
$\eta_d$	0.014	Noise for perturbation during cell division
$\beta$	1	External activation signal
$dt$	1	Simulation time step
Time Points	$\{0, 8, 16, 24, 32\}$	Observation time points

### Growth rate correlation analysis.

To validate the accuracy of our growth rate modeling and learning, we perform correlation analysis between the predicted growth rates and the ground truth. We utilize this dataset, which is generated

from the prescribed dynamical systems (Eq. (A-23)). Since the ground truth growth rates can be computed analytically, we conduct **correlation analysis on out-of-distribution (OOD) time points**—excluded from the training data—to assess the generalization capability of VGFM. The results are summarized in Table A-6.

Table A-6: Correlation analysis on four OOD time points for Simulation Gene Data

OOD Time Point	Pearson Correlation Coefficient
$t_{0.5}$	0.980
$t_{1.5}$	0.992
$t_{2.5}$	0.995
$t_{3.5}$	0.996

The growth rates predicted by our VGFM exhibit strong correlation with the ground truth values.

**Uncertainty quantification.** Although our current model is deterministic, we incorporate dropout into the velocity and growth networks to quantify predictive uncertainty. During inference, we compute the average variance for each dimension of velocity and growth across all data points, considering different branches (0 denotes the quiescent region *i.e.*, the lower left corner of Fig. 3 while 1 denotes the region where exhibits mass variation and state transition) and time points. We conduct the experiment with a dropout rate of 0.1, performing 5 stochastic forward passes through the velocity and growth networks for all data points. The results are presented in Table A-7.

Table A-7: Uncertainty estimation of predicted velocity (2-dimensional) and growth on Simulation Gene Data.  $\text{var}_{x_1}$  and  $\text{var}_{x_2}$  represent the average variance of the first and second velocity dimensions, respectively, while  $\text{var}_g$  denotes the variance of growth. All variance values are scaled by  $10^{-3}$ .

Branch	Time	$\text{var}_{x_1} (\times 10^{-3})$	$\text{var}_{x_2} (\times 10^{-3})$	$\text{var}_g (\times 10^{-3})$
0	0	0.1080	0.1000	0.0550
0	1	0.0770	0.0600	0.0660
0	2	0.0660	0.0670	0.1100
0	3	0.0850	0.0820	0.1910
0	4	0.1220	0.1370	0.3500
1	0	2.9320	6.5010	0.2510
1	1	3.0230	4.2870	0.3060
1	2	2.0410	1.9780	0.5590
1	3	0.9530	0.9760	0.7660
1	4	0.2800	0.3100	1.1680

We analyze the uncertainty patterns from both spatial and temporal perspectives:

For spatial analysis, cells in branch 0 exhibit minimal movement with little variation in state and counts. Correspondingly, the variances of  $v_1$ ,  $v_2$ , and  $g$  in this branch are consistently small, indicating low predictive uncertainty. In contrast, branch 1 demonstrates substantially larger variances, reflecting higher uncertainty in predictions.

For temporal analysis, The Simulation Gene Data features significant state transitions and quantity changes during initial phases, with diminishing magnitudes over time. The variance patterns of  $v_1$ ,  $v_2$ , and  $g$  accurately capture this temporal evolution, validating our uncertainty estimation approach. Specifically, for branch 1, the of  $v$  decrease monotonically over time, aligning with the reduced dynamics in later stages.

**Effect of sampling bias.** In cellular dynamics reconstruction, it is conventionally assumed that observed datasets accurately capture the temporal evolution of the true cell distribution. Consequently, the observed cell counts should align with the biological processes of cell proliferation and death. Evaluating the robustness of VGFM when trained on biased observational data holds significant practical relevance.

To generate non-uniform or subsampled data, we resample the original dataset with perturbed branch ratios. Specifically, for each time point  $t$ , we randomly sample a perturbation factor  $a_t \sim \mathcal{N}(0, \sigma^2)$ . Given an original branch ratio of  $s_t : (1 - s_t)$ , the resampled ratio becomes  $(s_t - |a_t|) : (1 - s_t + |a_t|)$ ,

where  $\sigma$  quantifies the bias level. We train our model on the resampled data and evaluate its performance on the original dataset to assess VGFM’s robustness.

Table A-8: Performance evaluation ( $\mathcal{W}_1$ /RME metrics) on Simulation Gene Data with varying sampling bias levels, averaged over three random seeds.

Bias Factor $\sigma$	$t_1$	$t_2$	$t_3$	$t_4$
0 (unbiased)	0.041/0.007	0.053/0.005	0.038/0.007	0.040/0.008
0.1	0.074/0.035	0.094/0.051	0.075/0.063	0.077/0.032
0.2	0.085/0.059	0.113/0.073	0.134/0.062	0.123/0.125

### B.3 Dyngen Data

Dyngen is a multi-modal simulation engine for studying dynamic cellular processes at single-cell resolution [62]. We follow the setup of [31], which simulates gene expression time-series data that mimics cell proliferation processes, including branching and temporal progression. In the analyzed instance of the dyngen dataset, the samples span five discrete time points, labeled from 0 to 4, capturing the temporal evolution of the cell population.

We use PHATE [63] to reduce its dimensions to 5, the same as [31]. Starting from time point 0, cells can be divided into two branches based on the sign of the second PHATE coordinate ( $x_2$ ). Quantitative analysis reveals a pronounced branch imbalance: for instance, at time point 4, 88 cells belong to one branch while 213 belong to the other, which poses a great challenge to the models that do not consider unbalancedness, resulting in cross-branch inference. Throughout this study, we assume that the temporal change in cell counts within each branch is entirely governed by a growth function, without contributions from migration or observational noise.

By setting  $\tau = 5$  and  $\epsilon = 0.03$  according to our selection scheme described in Appendix B.1 and illustrated in Fig. A-6 (b). Our model successfully avoids cross-branch reconstructions in trajectory modeling, as illustrated in Fig. A-7. This leads to generated samples that remain well-aligned with the underlying manifold. In terms of growth rate estimation, our model also accurately captures the rapid expansion observed in the branch below between time point 3 and time point 4, where the number of observed cells increases dramatically from 25 to 213, indicating the highest growth rate in the system.

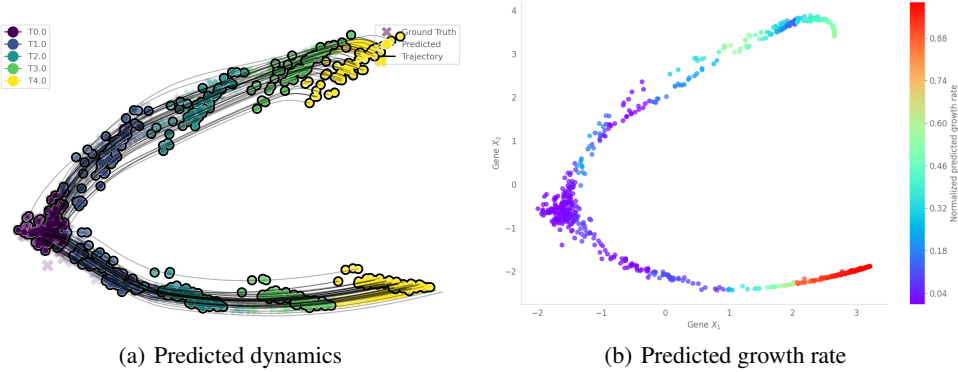


Figure A-7: Predicted dynamics and growth rates by VGFM on Dyngen data.

### B.4 Gausssian 1000D Data

DeepRUOT [26] employs a high-dimensional Gaussian mixture distribution (100D) [64] to evaluate the scalability of models. In this work, we adopt an even more challenging setting by testing scalability at 1000 dimensions. Following the setup of [26], for the initial distribution, we generated 400 samples from the Gaussian located lower in the  $(x_1, x_2)$  plane, and 100 samples from the Gaussian positioned higher. For the final distribution, we generated 1,000 samples from the upper Gaussian, and 200

samples each from the two lower Gaussians, and assume cells in the upper region exhibit proliferation without transport [26]. We observe that simulation-based methods [26] encounter training instability in the 1000-dimensional setting and flow matching-based methods fail to learn a reliable velocity field due to the unbalancedness of data, as shown in Tab. 1. (Some works incorporating unbalancedness [37, 42] may as well reconstruct correct velocity field but fail to construct growth rate). In contrast, By setting  $\tau = 5$  and  $\epsilon = 0.03$  according to our selection scheme described in Appendix B.1 and illustrated in Fig. A-6 (c). Our method remains effective even in high-dimensional regimes for both velocity and growth field learning.

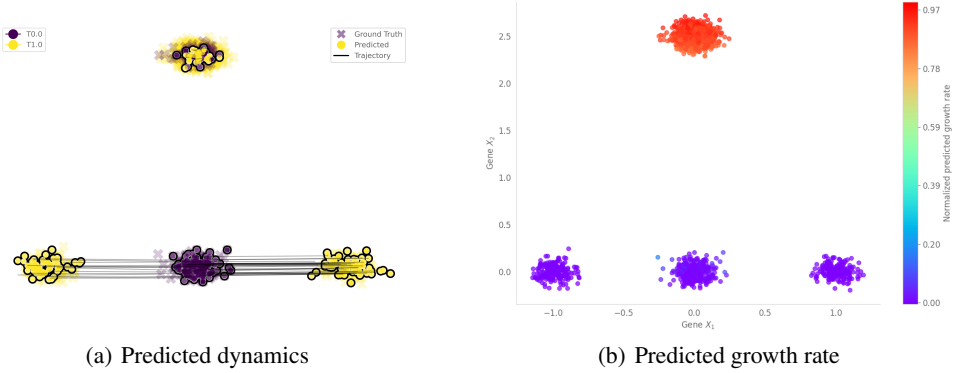


Figure A-8: Predicted dynamics and growth rates by VGFM on Gaussian 1000D data.

### B.5 Additional Experiment: Mouse Hematopoiesis Data

We also validate our algorithm on the mouse hematopoiesis data previously analyzed in [23, 26, 73]. This dataset leverages lineage tracing to track differentiation trajectories. After applying batch correction to integrate data across multiple experiments, the cells were embedded into a two-dimensional force-directed layout (SPRING plot). The resulting visualization reveals a pronounced bifurcation, where early progenitor cells diverge into two distinct differentiation lineages.

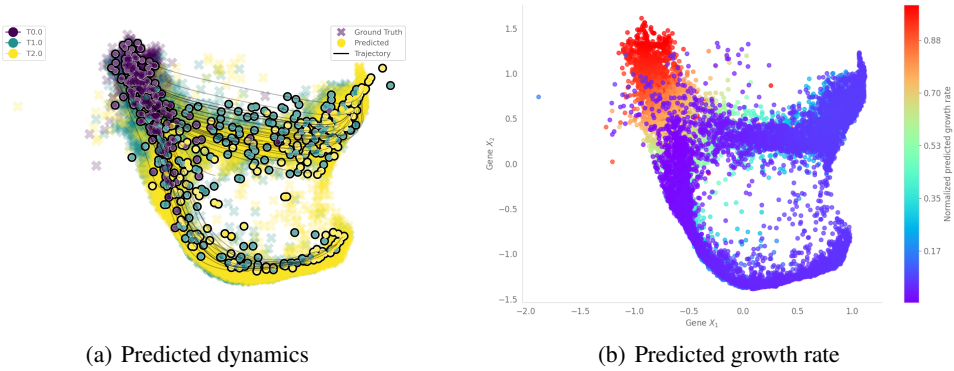


Figure A-9: Predicted dynamics and growth rates by VGFM on mouse hematopoiesis data.

Our model effectively captures the underlying branching structure in the data, and the predicted growth rates align well with those reported in [26], as well as with established biological priors and are consistent with known biological lineage patterns. Following the evaluation protocol of [26], we also compute the Wasserstein-1 distance as a quantitative metric. Under this evaluation, by setting  $\tau = 20$  and  $\epsilon = 0.005$  according to our selection scheme described in Appendix B.1, we train the model with a warm-up stage of 200 iterations. After that, the distribution fitting loss  $\mathcal{L}_{OT}$  is introduced for an additional 100 training epochs. Our method achieves superior performance, demonstrating improved trajectory inference. In addition, we evaluate the RME (Relative Mass Error)

metric introduced in this paper. The results indicate that our model significantly outperforms the other baselines in terms of mass-matching reconstruction accuracy, as shown in Tab. A-9 and Fig. A-10.

Table A-9:  $\mathcal{W}_1$  and RME over all time on mouse hematopoiesis data. Part of the results were adopted from [26].

Models	$t_1$		$t_2$	
	$\mathcal{W}_1$	RME	$\mathcal{W}_1$	RME
SF <sup>2</sup> M [41]	0.167	—	0.190	—
uAM[44]	0.745	—	0.777	—
UDSB [57]	0.388	0.159	0.128	0.249
TIGON [23]	0.314	0.124	0.342	0.177
DeepRUOT [26]	0.145	0.140	0.132	0.202
VGFM	<b>0.115</b>	<b>0.043</b>	<b>0.094</b>	<b>0.019</b>

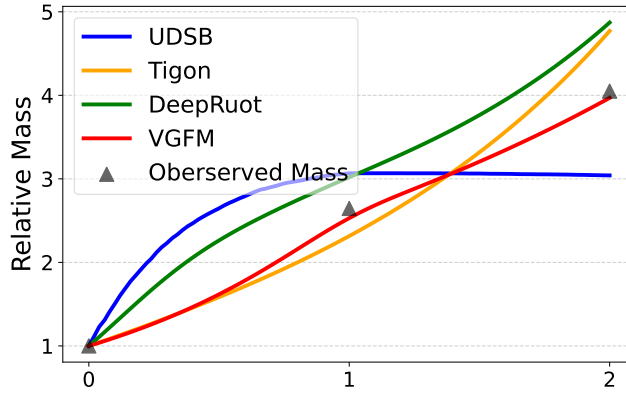


Figure A-10: Comparison of predicted relative mass (UDSB, TIGON, DeepRUOT, VFGM) with observed values from mouse hematopoiesis data.

## B.6 EB Data

Extensive experiments on the Embryoid Body (EB) [65] dataset with varying PCA dimensions are conducted in Sect. 4. We preprocess this dataset first by normalizing EB (5D) and leaving EB (50D) unnormalized. The proposed VFGM demonstrates strong performance in both cellular dynamics reconstruction and growth prediction, as illustrated in Fig. A-11. Specifically, we set the parameters as  $\epsilon = 0.01$ ,  $\tau = 5$  and normalize the cost matrix to ensure numerical stability when computing the semi-relaxed optimal transport plans (Eq. (11)).

**Big batch strategy for large-scale datasets.** When the number of observed data points is large, computing the transport plan between two consecutive time points using Eq. (11) becomes computationally inefficient. To address this, we adopt a Big-Batch Strategy. Specifically, we partition the data at each time point into  $n$  subsets, referred to as Big Batches. For each of the  $n$  Big Batches, we precompute and store the transport plans between adjacent time points. During training, we randomly select one Big Batch and sample smaller mini-batches from it based on the precomputed transport plan for model optimization. In real-world datasets, we set  $n = 5$ , while in synthetic datasets above we do not apply this strategy, *i.e.*,  $n = 1$ , as the sample size is relatively smaller. The mini-batch size is set to 256 for all experiments, except for Dyngen, where we use a smaller batch size of 60 due to its limited number of samples.

**Comparison with sinkhorn divergence as distribution fitting loss  $\mathcal{L}_{OT}$ .** We compute the distribution fitting loss  $\mathcal{W}_1$  in Sect. 3.4 using EMD distance by `pot` library, following MIOFlow [31] and DeepRUOT [26]. Particularly, we compute the OT using the `pot` library, using the function



`pot.emd()`. This function solves for  $\pi$  using a network flow algorithm, which is not compatible with PyTorch’s automatic differentiation. However, the gradients of the cost function  $c$  can still be backpropagated.

We replace the EMD distance with Sinkhorn divergence [70], which is fully differentiable and enjoys many favorable properties, using CUDA/C++ based `geomLoss` library. Tab. A-10 verified the effectiveness of Sinkhorn divergence for the distribution fitting loss.

Sinkhorn divergence between distributions  $p_0$  and  $p_1$  is defined as  $S_\epsilon(p_0, p_1) = \mathcal{W}_\epsilon(p_0, p_1) - \frac{1}{2}\mathcal{W}_\epsilon(p_0, p_1) - \frac{1}{2}\mathcal{W}_\epsilon(p_1, p_1)$ , where  $\mathcal{W}_\epsilon(p_0, p_1)$  is the entropy-regularized optimal transport distance (Sinkhorn distance) and  $\epsilon > 0$  is the entropy regularization parameter.

Table A-10: Comparison of EMD and sinkhorn divergence (with different entropic regularization parameters) on EB 50D dataset with metrics ( $\mathcal{W}_\infty$ /RME at each timepoint).

$\mathcal{L}_{OT}$	$t_1$	$t_2$	$t_3$	$t_4$	Training time(min)
EMD	7.951/0.039	8.747/0.042	9.244/0.019	9.620/0.044	13
$S_\epsilon(\epsilon = 0.001)$	7.902/0.018	8.767/0.013	9.063/0.083	9.507/0.096	9
$S_\epsilon(\epsilon = 0.002)$	7.904/0.032	8.791/0.033	9.111/0.104	9.523/0.120	9
$S_\epsilon(\epsilon = 0.005)$	7.917/0.048	8.768/0.062	9.086/0.137	9.518/0.151	9
$S_\epsilon(\epsilon = 0.01)$	7.904/0.035	8.801/0.038	9.102/0.113	9.511/0.127	9
$S_\epsilon(\epsilon = 0.05)$	7.908/0.036	8.787/0.045	9.080/0.120	9.512/0.129	9
$S_\epsilon(\epsilon = 0.1)$	7.916/0.030	8.773/0.022	9.111/0.091	9.591/0.108	9

It can be found that the sinkhorn divergence achieves further improvement on the accuracy of dynamics reconstruction, and costs less time for computing the distribution fitting loss.

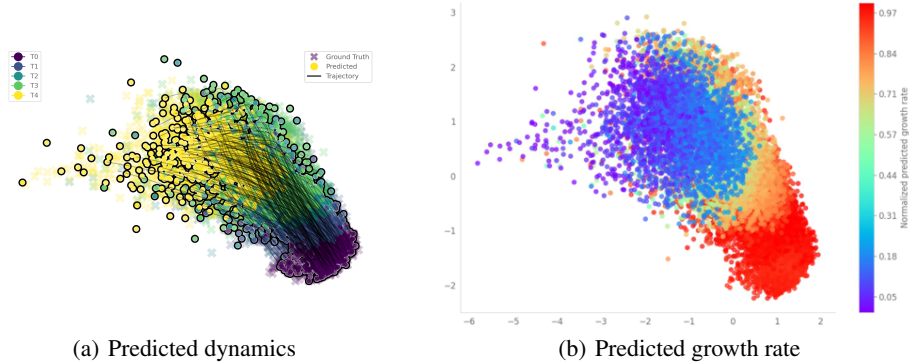


Figure A-11: Predicted dynamics and predicted growth rates by VGFM on 5D EB data with timepoint 1 held out.

We also examine whether the distribution fitting loss might lead to overfitting on the given population snapshots. In the hold-one-out experiments (Tab. 2), we relabel the timestamps of the four timepoints as 0, 1, 2, 3, and 4. Following the protocol in [38], we evaluate the model by holding out one intermediate timepoint at a time (*i.e.*, 1, 2, and 3) and report the average Wasserstein-1 distance. While hold-one-out experiments have already validated the superiority of VFGM over other methods, we further analyze the loss curves of both the distribution fitting loss and the  $\mathcal{W}_1$  distance on a hold-out distribution (*e.g.*, the distribution at time 1) that is not seen during training (Fig. A-12). The results show that the  $\mathcal{W}_1$  distance on the hold-out distribution decreases during the optimization of  $\mathcal{L}_{OT}$ , indicating the absence of overfitting and demonstrating the generalization ability of the distribution fitting loss in modeling unseen distributions.

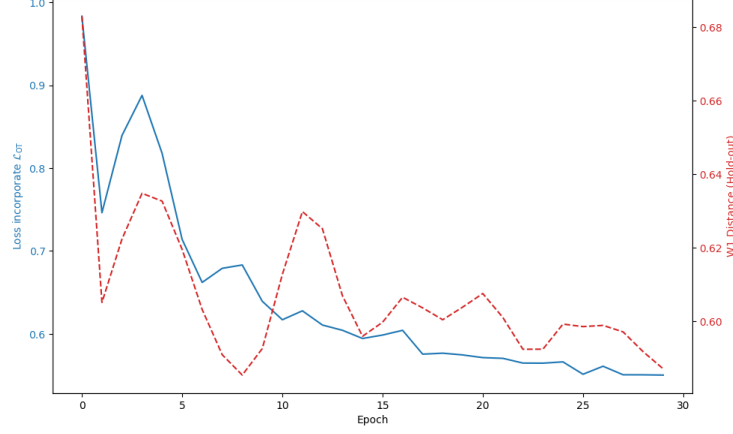


Figure A-12: Loss curves of the training loss after incorporating  $\mathcal{L}_{OT}$  (blue) and the  $\mathcal{W}_1$  distance on a hold-out distribution at timepoint 1 (red) on 5D EB data.

## B.7 CITE Data

CITE-seq (Cellular Indexing of Transcriptomes and Epitopes by Sequencing) [66] is an advanced technique that enables the simultaneous profiling of transcriptomes and surface protein expression at the single-cell level through the use of antibody-derived tags. In this study, we utilize only the gene expression matrix from the CITE-seq dataset, and preprocess the data by normalizing CITE (5D) and leaving CITE (50D) unnormalized. The experimental settings for the CITE dataset are consistent with those used in the EB data experiments, employing PCA with 5 and 50 dimensions, and hyperparameters set as  $\epsilon = 0.01$ ,  $\tau = 5$ . We relabel the timestamps of the four timepoints as 0, 1, 2, and 3. Following the protocol in [38], we evaluate the model by holding out one intermediate timepoint at a time (*i.e.*, 1 and 2) and report the average Wasserstein-1 distance.

To assess the model’s capability to capture state trajectories (Fig. A-13 (a)) and predict mass growth (Fig. A-13 (b)), the distribution at time point 1 is held out. Additionally, the distribution at time point 2 is held out to evaluate the performance of the  $\mathcal{L}_{OT}$  (Fig. A-14). The strong performance observed in the visualizations substantiates the potential of the proposed VGFM framework.

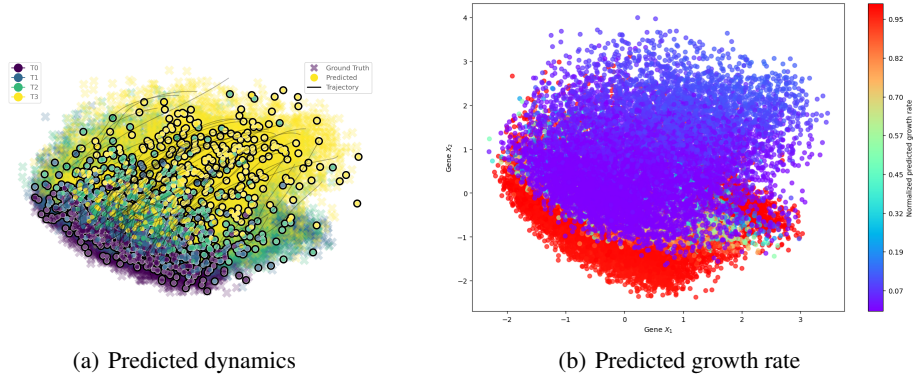


Figure A-13: Predicted dynamics and predicted growth rates by VGFM on 5D CITE Data with timepoint 1 held out.

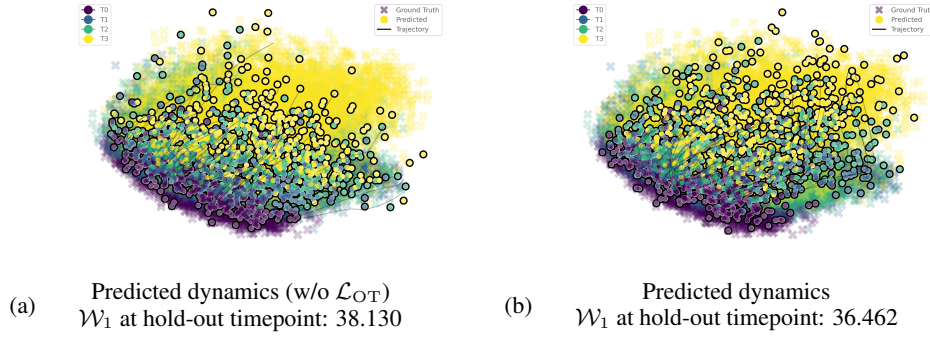


Figure A-14: Visualization of predicted dynamics by (a) VFGM (w/o  $\mathcal{L}_{OT}$ ) and (b) VGFM on CITE (50D) dataset, where the hold-out time is the second intermediate timepoint.

## B.8 Pancreas Data Analysis

To further explore the scalability of VGFM, we applied our model and compared methods explicitly modeling  $g_t(x)$  [23, 26] to the Pancreas dataset [67] comprising measurements of the developing mouse pancreas. We select cells at day 14.5 and 15.5 and relabel them as 0, 1 and select 2000 highly variable genes. As shown in [37], we assume the following cell type transitions are exclusively correct (denoted by  $\rightarrow$ ), *i.e.*, there is no descending cell type (or set of cell types) other than the given one. We partition all considered cell type transitions into three branches.

### Endocrine branch (ED) transitions.

- Fev+ Alpha (**FA**)  $\rightarrow$  Alpha (**A**)
- Fev+ Beta (**FB**)  $\rightarrow$  Beta (**B**)
- Fev+ Delta (**FD**)  $\rightarrow$  Delta (**D**)
- Fev+ Epsilon (**FE**)  $\rightarrow$  Epsilon (**E**)
- A  $\rightarrow$  A
- B  $\rightarrow$  B
- D  $\rightarrow$  D
- E  $\rightarrow$  E

### Ngn3 EP transitions.

- Ngn3 high early (**NE**)  $\rightarrow$  ED
- Ngn3 high late (**NL**)  $\rightarrow$  ED

### Non-endocrine branch (NEB) transitions.

- Ductal (**DU**)  $\rightarrow$  DU
- Tip (**T**)  $\rightarrow$  Acinar (**AC**)
- AC  $\rightarrow$  AC

We observed that VGFM is the only method showing a steadily decreasing training loss, both for  $\mathcal{L}_{VGFM}$  and  $\mathcal{L}_{OT}$ . We report  $\mathcal{W}_1$  and RME as shown in Tab. A-11.

This indicates that both loss terms are still effective in high-dimensional real-world data.

**Analysis on mean and variance.** We calculated the means and variances of the real and generated gene at day 15.5 and plotted the corresponding mean-variance trend (Fig. A-15 (a), (b)) and histograms (Fig. A-15 (c), (d)). The results show that the generated samples closely follow the mean-variance trend of the real data, especially by incorporating  $\mathcal{L}_{OT}$ .

Table A-11:  $\mathcal{W}_1$  and RME at day 15.5 between real data and generated data from VGFM and its variant.

Method	$\mathcal{W}_1$	RME
VGFM (w/o $\mathcal{L}_{OT}$ )	27.002	0.025
VGFM	24.416	0.017

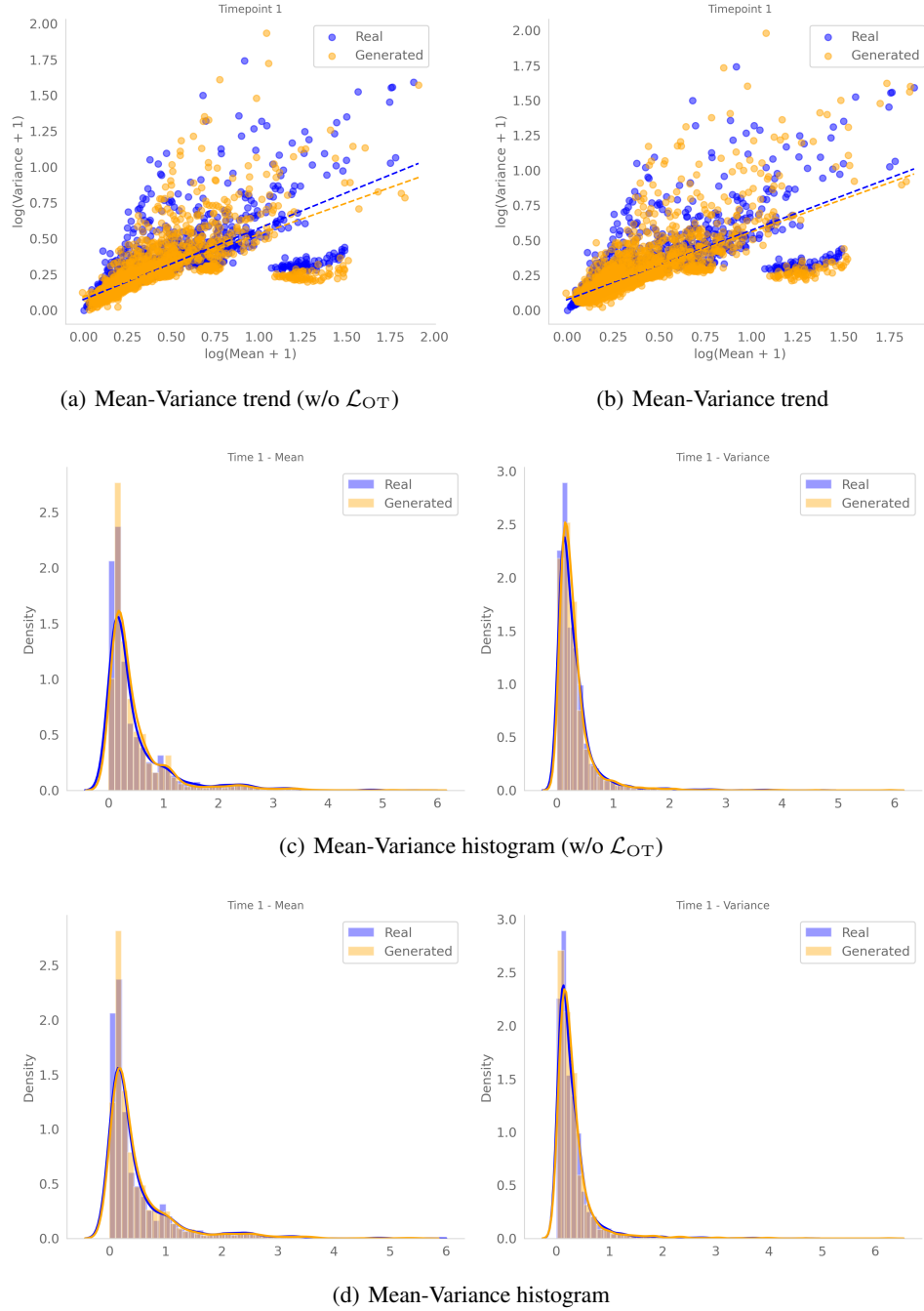


Figure A-15: mean-variance trend ((a), (b)) and histograms ((c), (d)) of real and generated gene data.

**Interpretable learned growth function.** We highlight that our main contribution lies in modeling and training the growth function  $g_t(x)$ , with VGFM in the flow matching framework, to leverage snapshot mass change for learning  $g_w(x, t)$ . During training, the growth loss rapidly converges to a negligible value, much faster than the velocity loss. To further investigate  $g_w(x, t)$ , We first calculate growth rate of each cell at the initial time point (day 14.5) and visualize them in Fig. A-16. At this time, the proliferation observed during this developmental stage mainly originates from Acinar and Ductal cells. Remarkably, our model successfully recapitulates this pattern without being given any prior knowledge of the cell types, demonstrating its ability to infer biologically meaningful dynamics directly from the data.

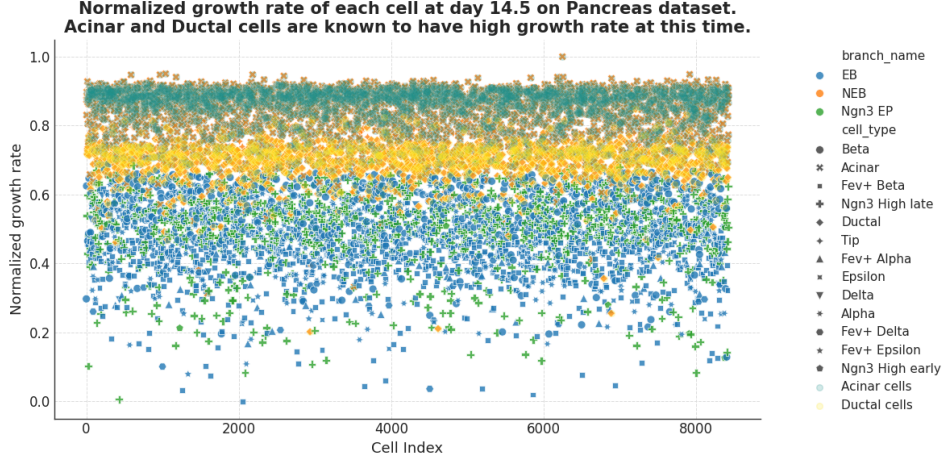


Figure A-16: Normalized growth rate of each cell at time point 0 (day 14.5).

We also demonstrate that the learned  $g_w(x, t)$  not only successfully recapitulates the relative growth rates of different cell types without any prior information about cell identities, but also that the branch-wise mass obtained from numerical integration of the ODE closely matches the total branch mass at time point 1.

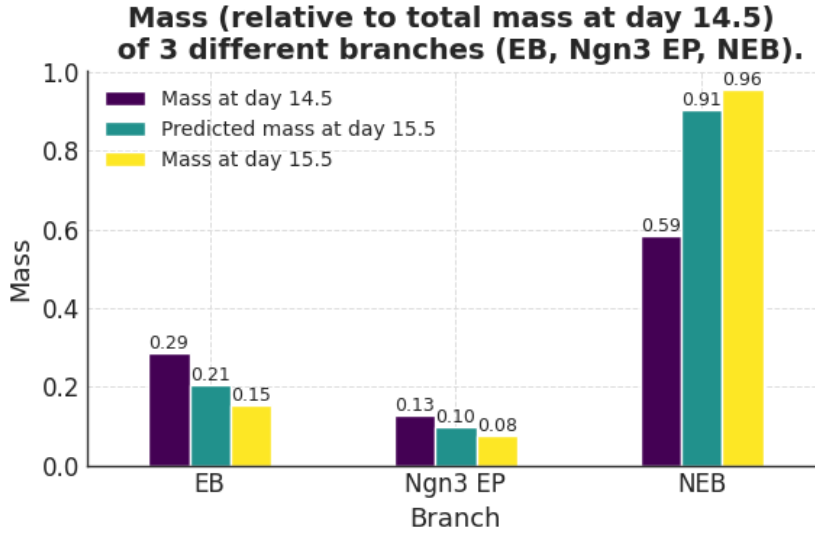


Figure A-17: Relative mass of different branches

we then follow [23], for each time  $t$ , compute the element-wise absolute value  $\left| \frac{\partial g(x, t)}{\partial x_i} \right|$  in the gene space for every cell, and average across cells to quantify each gene's contribution to growth

dynamics. From our analysis of  $g_w(x, t)$ , We have identified Pnliprp1, Clps, and Ctrb1, as shown in Fig. A-18. These genes are well-known markers of Acinar cells and are highly expressed in the exocrine pancreas. Notably, as shown above, cells from the non-endocrine branch, particularly Acinar and Ductal cells, are much more abundant at later stages due to their high proliferation rates. This alignment between the learned growth-driving genes and known biological processes indicates that VGFM captures growth dynamics in a biologically interpretable manner, successfully linking the latent growth function to meaningful cell-type-specific proliferation patterns.

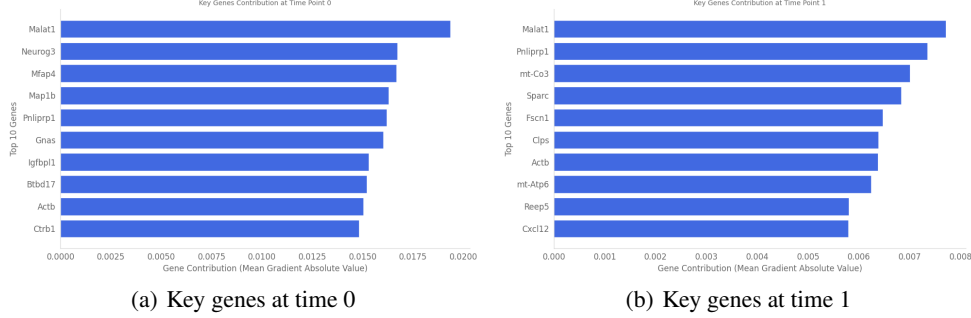


Figure A-18: Key genes identified by  $g_w(x, t)$  at time 0 (a) and 1 (b).

## C Broader Impacts

Our method provides a scalable and efficient framework for modeling cellular dynamics, enabling trajectory reconstruction and growth rate inference in high-dimensional single-cell datasets. In biomedical and clinical settings, such capabilities can facilitate a deeper understanding of developmental processes, disease progression, and response to treatment at a single-cell resolution. For example, modeling differentiation trajectories and proliferation patterns of stem or immune cells could inform therapeutic strategies in cancer, regenerative medicine, and immunotherapy.

However, the proposed method is inherently data-driven and relies on statistical patterns learned from observational data. As such, it may produce biologically implausible trajectories or growth behaviors that conflict with known biological priors, especially when the training data is noisy, biased, or incomplete. This could potentially lead to misleading interpretations or incorrect clinical hypotheses if not carefully validated by domain experts. Therefore, we emphasize that any downstream medical conclusions drawn from the model's output should be interpreted with caution and in conjunction with biological prior knowledge and experimental validation.

## NeurIPS Paper Checklist

### 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 main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

Guidelines:

- The answer NA 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 NA 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: Please see Sect. 5

Guidelines:

- The answer NA 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: We provide the full set of assumptions and a complete (and correct) proof for each theoretical result.

Guidelines:

- The answer NA 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: The paper fully discloses all the information needed to reproduce the main experimental results of the paper.

Guidelines:

- The answer NA 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: Our code is available at <https://github.com/DongyiWang-66/VGFM>.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.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://nips.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, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper specifies all the training and test details necessary to understand the results.

Guidelines:

- The answer NA 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: [No]

Justification: The current version of the paper does not report error bars or statistical significance metrics.

Guidelines:

- The answer NA 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: The paper provide sufficient information on the computer resources needed to reproduce the experiments.

Guidelines:

- The answer NA 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: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA 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: Please see Appendix C.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA 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: The creators or original owners of assets used in the paper are properly credited and the license and terms of use are explicitly mentioned and properly respected.

Guidelines:

- The answer NA 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](https://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: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA 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: [NA]

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

Guidelines:

- The answer NA 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: [NA]

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

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA 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 (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.