PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Modeling single cell trajectory using forward-backward stochastic differential equations

Kevin Zhang[1], **Junhao Zhu**[1], **Dehan Kong**[1]*, **Zhaolei Zhang**[2,3,4]*

1 Department of Statistical Sciences, University of Toronto, Toronto, Ontario, Canada, **2** Department of Molecular Genetics, University of Toronto, Toronto, Ontario, Canada, **3** Department of Computer Science, University of Toronto, Toronto, Ontario, Canada, **4** Donnelly Centre for Cellular and Biomolecular Research, University of Toronto, Toronto, Ontario, Canada

* dehan.kong@utoronto.ca (DK); zhaolei.zhang@utoronto.ca (ZZ)

Citation: Zhang K, Zhu J, Kong D, Zhang Z (2024) Modeling single cell trajectory using forward-backward stochastic differential equations. PLoS Comput Biol 20(4): e1012015. https://doi.org/10.1371/journal.pcbi.1012015

Editor: Jianhua Xing, University of Pittsburgh School of Medicine, UNITED STATES

Received: October 25, 2023

Accepted: March 22, 2024

Published: April 15, 2024

Data Availability Statement: Data availability: – The Human Embryonic Stem Cells dataset provided by Moon et al is available from the Mendeley Data repository at https://doi.org/10.17632/v6n743h5ng.1. – The Mouse Embryonic Fibroblasts dataset provided by Schiebinger et al is available from NCBI Gene Expression Omnibus at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE122662. – The Arabidopsis thaliana Stem Cells dataset provided by Shahan et al is available from NCBI Gene Expression Omnibus at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE152766 • Code availability: All code used in this

Abstract

Recent advances in single-cell sequencing technology have provided opportunities for mathematical modeling of dynamic developmental processes at the single-cell level, such as inferring developmental trajectories. Optimal transport has emerged as a promising theoretical framework for this task by computing pairings between cells from different time points. However, optimal transport methods have limitations in capturing nonlinear trajectories, as they are static and can only infer linear paths between endpoints. In contrast, stochastic differential equations (SDEs) offer a dynamic and flexible approach that can model non-linear trajectories, including the shape of the path. Nevertheless, existing SDE methods often rely on numerical approximations that can lead to inaccurate inferences, deviating from true trajectories. To address this challenge, we propose a novel approach combining forward-backward stochastic differential equations (FBSDE) with a refined approximation procedure. Our FBSDE model integrates the forward and backward movements of two SDEs in time, aiming to capture the underlying dynamics of single-cell developmental trajectories. Through comprehensive benchmarking on multiple scRNA-seq datasets, we demonstrate the superior performance of FBSDE compared to other methods, highlighting its efficacy in accurately inferring developmental trajectories.

Author summary

In this study, we address the challenge of modeling the trajectories of single-cell gene expression over time. A “trajectory” in this context refers to the dynamic evolution of gene expression within a single cell, and this challenge is exacerbated by the fact that the gene expression of sequenced cells cannot be observed in subsequent time points. To overcome this limitation, we employ innovative stochastic differential equations (SDEs), a technique recently utilized in trajectory modeling studies, to more accurately capture the dynamic evolution of single-cell gene expression. Rigorously tested on multiple single-cell RNA sequencing datasets, our enhanced SDE system demonstrates promising performance in capturing these intricate gene expression dynamics, thereby advancing our capacity to glean critical insights across various biological contexts.

study are available from the GitHub respository at https://github.com/Diebrate/population_model.

Funding: DK and ZZ acknowledge financial support from a Catalyst Grant from Data Science Institute and Medicine by Design, University of Toronto. KZ and JZ are partially supported by CANSSI (Canadian Statistical Sciences Institute). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Introduction

The technology of single-cell RNA sequencing (scRNA-seq) is a revolutionary breakthrough in the study of cellular developmental dynamics [1, 2]. With the availability of time series scRNA-seq data, lineage tracing and cell differentiation processes have been studied in multiple cell types such as salivary glands, liver cells, lung cells, kidney cells, neuronal cells, and tumor cells [3–8]. Despite their success, the sequencing process is destructive as the sampled cells are destroyed and their future gene expressions cannot be measured. Several mathematical approaches have emerged to take these time series data as input and recover the true cellular differentiation processes. Earlier modelling approaches include continuous-state hidden Markov models (CSHMM) [9], and a graph-based model Tempora [10]. These methods typically cluster cells into a small number of distinct cell types and model the trajectory of these cell types instead of modelling individual cells. These approaches are simpler to model and are easier to interpret but lose cell-level information after reducing a population of cells into discrete cell types. Recently, optimal transport (OT) becomes a state-of-art approach in modelling cell developmental trajectories from single cell RNA-seq data. In particular, Schiebinger [11] described a new framework (Waddington-OT) to identify the pairings between cells from different time points where gene expressions of single cells are assumed to follow a developmental trajectory with the lowest transport energy cost. Waddington-OT uses a joint probability mass function to infer how likely a cell sampled at a previous time point will become another cell sampled at a later time point. The key of the OT model is that it assumes the cell population evolves in an energy-efficient fashion analogous to the optimal transport plan in probability theory. In this context, the energy term is defined as the differences in genome-wide gene expression profile of a population of cells between consecutive time points [11]. In contrast to graph-based models, optimal transport does not require clustering cells into discrete cell types. Instead, it models the developmental processes at the single cell level, making it a potentially more powerful approach.

Despite its success and later improvement [12], Waddington-OT still has two limitations. First, it can only infer the endpoints of a path from sampled cells, making generative modelling infeasible because paths starting from an unobserved point cannot be simulated. Second, path or trajectory inference in Waddington-OT is static as it only provides linear interpolation between endpoints. In this context, linear trajectory means the shape of a path is simply a straight line connecting the endpoints. Instead of matching path endpoints with OT, a potential solution is to model path movements with stochastic differential equations (SDEs), which are more adept in dynamic trajectory inferences [13, 14]. In general, an SDE model considers X_t as the gene expression of a cell in a population existing between initial time $t = 0$ and terminal time $t = T$; the change in X_t at time t is denoted by dX_t and equals $vdt + \sigma dW_t$ where v and σ represent drift and volatility. The drift term v takes a cell's spatial and temporal information as input and produces the direction in which the cell changes its gene expression as output. The volatility term σ mimics the pure randomness in cell movements by scaling Brownian motions denoted by W_t. As a result, SDE is able to parametrize non-linear trajectories and provide feasible generative simulations on a continuous timeline. The volatility term σ in SDE is also equivalent to the entropy regulation in optimal transport [15]. In practice, SDE models can be parameterized by neural networks [16, 17], which have shown good performance in recovering scRNA trajectories [18].

Single cell sequencing has allowed us to study cell-to-cell interactions, which has not been included in earlier trajectory inference methods [19, 20]. In biology, it is well-established that cells can influence the development of other cells by modulating their gene expression or other cellular programs. This modulation can occur through the secretion of small molecules or via direct interactions between receptors and ligand molecules on cell surfaces. Interestingly, these cell-to-cell interactions bear resemblance to concepts found in mean field control theory, which aims to optimize the collective behavior of a population while accounting for interactions among individual members [16, 17]. Drawing inspiration from this, a recent model called TrajectoryNet [18] incorporated mean field control theory to capture interactions among single cells.

In the following, we explore the rationale and limitations of TrajectoryNet, which is the most closely related method to our proposed approach. It serves as a valuable baseline for our method. The model assumes that gene expressions in cells at each time point $t = 0, 1, 2, \ldots, T$ follow a distribution q_t. TrajectoryNet begins with q_0 and generates a sequence of distributions ρ_t for $t = 1, 2, \ldots, T$ based on the SDE model (refer to Eq 1 in Methods). This process can be achieved through simulations, starting with m cells randomly subsampled from time point $t = 0$ and then simulating subsequent cells at time points $t = 1, 2, \ldots, T$ using the SDE model. The gene expressions of these simulated cells are denoted by $x^*_{t,1}, x^*_{t,2}, \ldots, x^*_{t,m}$, and their empirical distribution is denoted as $\hat{\rho}_t$. The main optimization problem in TrajectoryNet is similar to a road-building problem, where the objective is to construct a road connecting two points A and B, while passing through multiple intermediate checkpoints. TrajectoryNet aims to find an optimal SDE model that captures the collective displacement of simulated cells, akin to the construction cost of a road. Additionally, TrajectoryNet incorporates interactions among cells, resembling the movement of cars passing and merging on the road. Similar to the road-building problem, one would expect $\hat{\rho}_t = q_t$ for each time point $t = 0, 1, 2, \ldots, T$, which can be interpreted as the initial and terminal points of the road and the intermediate checkpoints it must traverse. However, TrajectoryNet only satisfies the initial constraint $\hat{\rho}_0 = q_0$ but not the subsequent constraints at $t = 1, 2, \ldots, T$. Instead, TrajectoryNet relaxes these constraints by penalizing the divergence between q_t and $\hat{\rho}_t$. In other words, TrajectoryNet approximates the constraints as $\hat{\rho}_t \approx q_t$ for $t = 1, 2, \ldots, T$. Consequently, TrajectoryNet is less effective due to the lack of exact matching between the simulated distributions $\hat{\rho}_t$ and the true distributions q_t for $t = 1, 2, \ldots, T$.

Motivated to overcome these limitations, we propose two key enhancements to improve trajectory inference. Firstly, we enhance the approximation of intermediate constraints by introducing a new penalty term in the optimization problem. Secondly, we enforce the terminal constraint $\hat{\rho}_T = q_t$ by implementing the FBSDE method (refer to Fig 1 for a visual illustration). The proposed enhancements aim to enhance the accuracy of trajectory inference by achieving a closer alignment between the simulated distributions $\hat{\rho}_t$ and the true distributions q_t for each time point $t = 1, 2, \ldots, T$. This improved alignment ensures that the inferred trajectories more faithfully capture the underlying dynamics of the system. In the rest of this section, we provide an overview of our innovative approaches and their key components.

To enhance the approximation of intermediate constraints, we introduce a new penalty term as compared to the TrajectoryNet method discussed above [18]. To satisfy the conditions $\hat{\rho}_t \approx q_t$ for all time points t, TrajectoryNet implements two penalty terms. First, it directly penalizes the divergence between p_t and $\hat{\rho}_t$ using the maximum likelihood function. Second, it also introduces a penalty term which is proportional to the sum of distances among k nearest neighbours. To illustrate this, suppose we have n observations and n simulations at each time point t and denote $x_{t,i}$ and $x^*_{t,j}$ the gene expressions of ith observed cell and jth simulated cell respectively at time t. The conditions $\hat{\rho}_t \approx q_t$ imply that there is a large overlap between the locations occupied by the observations and those by the simulations in the gene expression space (see Fig 2a and 2b).

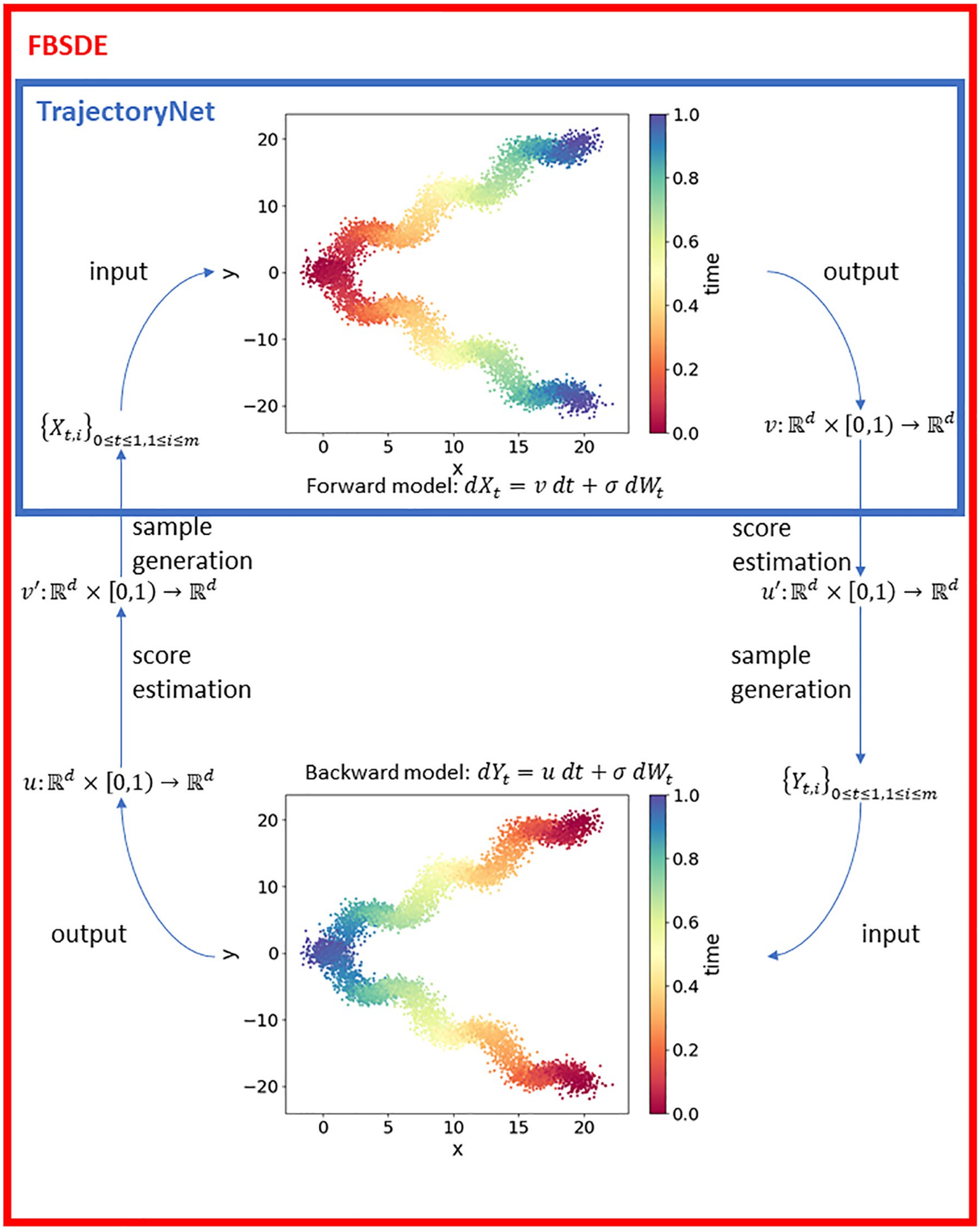
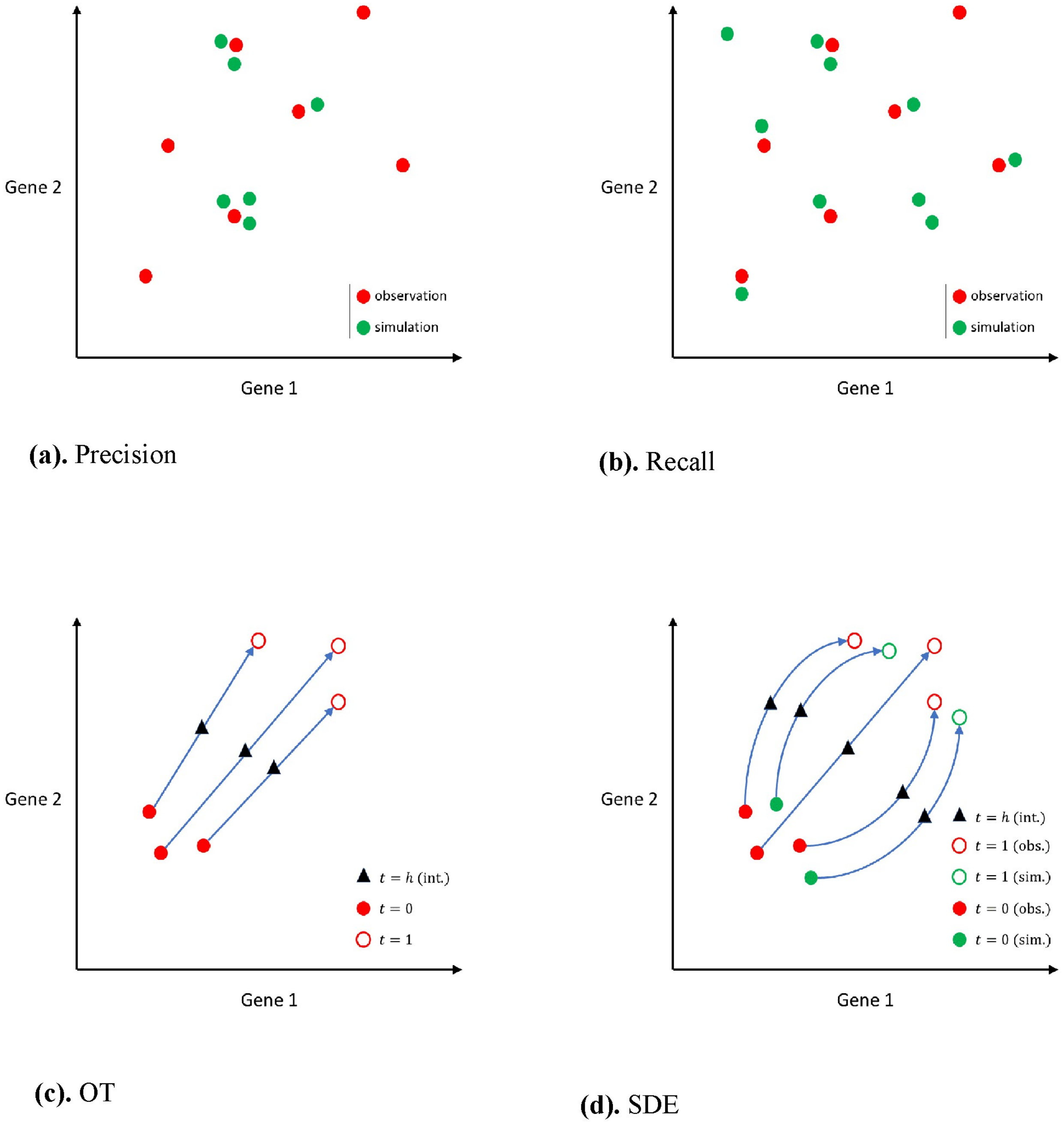

FBSDE
TrajectoryNet
input
$\{X_{t,i}\}_{0\leq t\leq 1, 1\leq i\leq m}$
20
10
0
−10
−20
y
0 5 10 15 20
x
1.0
0.8
0.6
0.4
0.2
0.0
time
Forward model: $dX_t = v\,dt + \sigma\,dW_t$
output
$v: \mathbb{R}^d \times [0,1) \rightarrow \mathbb{R}^d$
sample generation
$v': \mathbb{R}^d \times [0,1) \rightarrow \mathbb{R}^d$
score estimation
$u: \mathbb{R}^d \times [0,1) \rightarrow \mathbb{R}^d$
score estimation
$u': \mathbb{R}^d \times [0,1) \rightarrow \mathbb{R}^d$
sample generation
$\{Y_{t,i}\}_{0\leq t\leq 1, 1\leq i\leq m}$
Backward model: $dY_t = u\,dt + \sigma\,dW_t$
output
input

Fig 1. Overview of FBSDE model. Colours in the scatter plots represent time where red is the earliest and blue is the latest time point. In the forward model, the population is modelled from left to right. In the backward model, the population evolves from right to left. The blue rectangle on top represents the TrajectoryNet framework by Tong et al [18] which is equivalent to half of one iteration of our FBSDE model. The FBSDE model iterates between the Forward and Backward models; traversing through the Forward model generates new simulated data points which are subsequently used as training set by the backward model and vice versa.

https://doi.org/10.1371/journal.pcbi.1012015.g001



(a). Precision

(b). Recall

(c). OT

(d). SDE

Fig 2. **Fig** 2a and 2b) Comparison between precision and recall. Red circles represent observations and green circles represent simulations. In **Fig** 2a, a high precision model means that simulations are close to observations; however, there can be some observations without any simulations nearby. In **Fig** 2b, a high recall model means that most observations have some simulations nearby; however, there can be some simulations that are far away from any observations. **Fig** 2c and 2d) Comparison between OT and SDE. Filled circles represent cells from $t = 0$. Unfilled circles represent cells from $t = 1$. The red colour represents observations. The green colour represents simulations. Triangles indicated interpolated cells at time $t = h$. **Fig** 2c) OT infers the endpoints of a path and the trajectory is simply a straight line connecting the endpoints. OT can only infer paths whose endpoints come from observations. **Fig** 2d) SDE infers both linear and non-linear trajectories using differential equations. In addition, SDE also infers paths originating from points that are not observations (i.e., given an arbitrary starting point).

https://doi.org/10.1371/journal.pcbi.1012015.g002

At each time point t, for each simulated data point $x^*_{t,i}$ (i.e., gene expression of single cells), TrajectoryNet computes its distances to k closest observations among $\{x_{t,i}\}$. These soft marginal constraints force the simulated points to be in a close neighbourhood of observations. In other words, it is less likely to have simulations far away from observations. Therefore, this penalty improves the precision of the model. We modified the k-nearest-neighbour distances in TrajectoryNet [18] by additionally computing the distances to k closest simulations among $\{x^*_{t,j}\}$ for each observed point $x_{t,i}$. This additional term allows more observations to be included in a close neighbourhood of simulated points (see Methods for details). As a result, our model also improves recall as illustrated in Fig 2b.

To enforce the exact terminal equality constraint $\hat{\rho}_T = q_T$, we utilize the forward-backward stochastic differential equations (FBSDE) proposed by Vargas et al [21]. The FBSDE model consists of a forward model which goes from initial time to terminal time ($0 \rightarrow T$) and a backward model which goes from terminal time to initial time ($T \rightarrow 0$) (see Methods for details). The forward model generates training data for the backward model and vice versa. The core concept of FBSDE is that training the two models where the time goes in opposite directions will eventually lead to convergence such that $\hat{\rho}_T = q_T$ is met by the forward model and $\hat{\rho}_0 = q_0$ is met by the backward model. A recent study by Bunne et al models cell trajectory with FBSDEs and demonstrates that the model indeed preserves terminal equality constraints $\hat{\rho}_T = q_T$ [22]. However, this implementation of the FBSDE model considers only the initial and terminal constraints, and does not model the intermediate time points. To further utilize the information from these observed intermediate time points, we extend the original FBSDE by enforcing soft constraints along all the intermediate time points as discussed above.

We benchmark FBSDE with two other methods Waddington-OT (WOT) and TrajectoryNet on three scRNA data sets: Human Embryonic Stem Cells Dataset [23], Mouse Embryonic Fibroblasts Dataset [11] and *Arabidopsis thaliana* Stem Cells Dataset [12]. We adopt a standard leave-one-out benchmark framework (i.e., removing one time point and trying to recover gene expression of hidden time point from other time points). Our results show that the FBSDE model consistently outperforms TrajectoryNet and Waddington-OT.

Methods

This section is structured as follows. In **Mathematical details**, we begin by providing a mathematical formulation of the developmental trajectory of a single cell and discussing the optimization problem typically employed in trajectory inference methods based on stochastic differential equations (SDEs). We also emphasize the importance of considering constraints within this optimization problem. Additionally, we present an overview of the TrajectoryNet framework [18], which serves as baseline for comparison.

Next, in **Estimation of cost functional**, we address the approximation of the cost functional, which represents the optimal value of our optimization problem. The cost functional is defined as the expectation of a stochastic integral. To approximate this integral, we employ time discretization, also known as the Euler-Maruyama method. We further discuss the approximation of probability density functions (PDFs), which play a vital role in SDE-based trajectory inference. By implementing these numerical techniques, we illustrate how the cost functional can be effectively approximated given a finite sample.

In **Approximation of intermediate constraints**, we propose modifications to the TrajectoryNet framework to improve the approximation of the intermediate constraints. We provide detailed explanations of how these modifications can be implemented using a finite set of cells, leading to enhanced trajectory inference accuracy.

Finally, in **Satisfying terminal constraint**, we introduce the Forward-Backward Stochastic Differential Equation (FBSDE) system and highlight its advantages in satisfying the terminal constraint. We implement a numerical algorithm proposed by Vargas et al. [21] for solving the FBSDE system.

Mathematical details

Suppose all the cells live in the space of $\mathbb{R}^d$ in the finite time frame $[0, T]$ for some constant $T > 0$. The gene expression profile of a cell at time t is denoted by $X_t \in \mathbb{R}^d$, which is a continuous random vector with probability density function q_t for $t \in [0, T]$.

Remark 1. *In practical applications, gene expression data is often preprocessed and undergoes dimensional reduction to reduce its dimensionality. Typically, the reduced dimension d is much smaller, commonly set to 2 or 3. In our study, all the datasets were obtained in the form of two-dimensional data published by authors cited in our manuscript, who employed dimension reduction techniques such as PHATE and UMAP on their raw data. These techniques were chosen for their effectiveness in preserving the global structure of high-dimensional data. For the selection of appropriate dimensions, we advise users to consider the specific protocols of the chosen dimension reduction technique, integrated with domain-specific knowledge.*

We are interested in the single-cell developmental trajectory $\{X_t, t \in [0, T]\}$. Let X_0 be an initial gene expression of a cell at time 0, then the trajectory of this cell at time t can be modeled by the stochastic differential equation as described by Zhang et al [12]:

$$X_t = X_0 + \int_0^t v(X_s, s)ds + \int_0^t \sigma(X_s, s)dW_s, \tag{1}$$

where $v : \mathbb{R}^d \times [0, T]$ is a drift function and $v(X_s, s)$ characterizes the shift change caused by X_s at time s. The integral $\int_0^t v(X_s, s)ds$ represents the cumulative mean shift induced by the trajectories of X_s up to time t, which is known as the drift term. The $\sigma : \mathbb{R}^d \times [0, T]$ is a volatility function and $\{W_t\}_{0\leq t\leq T}$ is a standard Brownian motion. At time s, dW_s refers to the change in the Brownian motion W_s. The term $\int_0^t \sigma(X_s, s)dW_s$ represents the random variation induced by the trajectories of X_s up to time t, which is known as the volatility term.

To infer the cell trajectory, we aim to learn the unknown drift function $v(X_t, t)$, also called action. In our analysis, we assume the volatility function is constant and set it equal to 1 as the default value.

Remark 2. *We advise users to normalize their input data. This normalization step is critical as it minimizes the impact of scaling on the diffusion term, ensuring a more consistent and reliable application of the method across different datasets. The selection of σ will be discussed in* ***Parameter tuning***.

Eq 1 characterizes the temporal dynamics of gene expression in an individual cell. In a population consisting of millions of cells, each cell undergoes simultaneous changes in its gene expression. Consequently, an action v governs the evolution of the population distribution over time. Remarkably, given an initial distribution ρ_0, the sequence of distributions $\rho(\cdot, t)_{0<t\leq T}$ can be mathematically represented as a function of the action v and volatility σ by leveraging the well-known Fokker-Plank Equation [24]. In this context, $\rho(\cdot, t)$ denotes the probability density function generated by the action v and satisfies $\rho_t = \rho(\cdot, t) : \mathbb{R}^d \times [0, T] \to \mathbb{R}$. We assume that ρ is a continuous function with continuous partial derivatives in the space $\mathbb{R}^d \times [0, T]$.

As cells undergo changes in their gene expression over time, there is often an associated energy cost. The optimal action $\tilde{v}$ can be determined by seeking a function that minimizes this cost functional. This prompts the question of whether an optimal strategy exists for all cells in the population to collectively modify their gene expressions in order to minimize the cost functional. In general, most SDE-based inference methods consider minimizing the following cost functional:

$$\begin{aligned} \tilde{v} &= \arg\min_{v} \tilde{J}(v, \rho_0) \\ \text{subject to}: \quad & dX_t = v(X_t, t)dt + \sigma dW_t, \end{aligned} \tag{2}$$

where

$$\tilde{J}(v, q_0) = \mathbb{E}_{q_0}\left[\int_0^T \left(\frac{1}{2}||v(X_t, t)||^2 + \alpha_F \log(\rho(X_t, t))\right)dt\right], \tag{3}$$

and $\alpha_F > 0$ is a regularization constant tuned by cross-validation.

Since the initial gene expression X_0 of a cell is a random variable with distribution q_0, the cost functional is defined as the expected cost of a trajectory originating from all possible values of X_0. The first term in the integral represents the non-interactive cost, where a cell's trajectory is independent of the actions of other cells. The integral of this term is typically referred to as the kinetic energy of the trajectory, representing the amount of energy required to change a cell's gene expression along this trajectory. In practice, the trajectory of one cell is likely to be influenced by other cells, therefore we need the second term in the integral, which is the running cost or interactive cost, representing how a cell is affected by other cells while performing action v. As a result, this term increases as the probability density at X_t increases. This concept is similar to the idea of crowdedness or entropy in terms of gene expression. For example, cells with similar gene expressions may compete for resources for growth, leading to trajectories with similar gene expressions having higher costs (i.e., higher $\rho(X_t, t)$).

In trajectory inference settings [11, 25], one often observes q_t (or more precisely the empirical version of it) at a grid of time points $0 = t_0, t_1, t_2, \ldots, t_K = T$. The trajectory inference problem aims to determine the optimal action $\tilde{v}$ such that the population evolves from q_0 to q_1, q_2, and ultimately to q_T, while minimizing the cost functional specified in Eq 2.

Let $\{\rho_t: t \in (0, T]\}$, denote a sequence of distributions generated from the initial distribution q_0 through the application of the SDE model governed by the action v. In this context, the trajectory inference problem can be formulated as a constrained optimization problem,

$$\begin{aligned} J(q_0) &= \inf_{v} \tilde{J}(v, q_0) \\ &= \inf_{v} \mathbb{E}_{q_0}\left[\int_0^T \left(\frac{1}{2}||v(X_t, t)||^2 + \alpha_F \log(\rho(X_t, t))\right)dt\right]. \\ & \text{subject to}: \begin{cases} dX_t = v(X_t, t)dt + \sigma dW_t \\ \rho_{t_k} = q_{t_k} \ \forall\ 0 \leq k \leq K \end{cases} \end{aligned} \tag{4}$$

Remark 3. *Consider the scenario where one aims to construct a road connecting two points A and B, while also passing through multiple intermediate checkpoints. In Eq 4, the SDE constraint $dX_t = v(X_t, t)dt + \sigma dW_t$ models the shape of the road. The initial, terminal, and intermediate constraints can be seen as analogous to the endpoints A and B and the intermediate checkpoints, respectively. Furthermore, the cost functional $\tilde{J}$ in the optimization problem corresponds to the cost associated with building the road. Thus, this optimization problem can be viewed as the task of designing a road that efficiently connects all the required checkpoints, while minimizing the overall cost $\tilde{J}$.*

However, previous studies [18, 21, 26] have argued that imposing equality constraints $\rho_{t_k} = q_{t_k}$ can be numerically challenging. These methods satisfy the initial constraint $\rho_0 = q_0$ by design. For all subsequent constraints at $t = t_1, t_2, \ldots, T$, they approximate these equality constraints, denoted by $\rho_{t_k} \approx q_{t_k}$. One example is TrajectoryNet which solves the following optimization problem:

$$\begin{aligned} J(q_0) &= \inf_{v} \tilde{J}(v, q_0) \\ &= \inf_{v} \mathbb{E}_{q_0}\left[\int_0^T \left(\frac{1}{2}||v(X_t, t)||^2 + \alpha_F \log(\rho(X_t, t))\right) dt\right]. \\ &\text{subject to}: \begin{cases} dX_t = v(X_t, t)dt + \sigma dW_t \\ \rho_0 = q_0 \\ \rho_{t_k} \approx q_{t_k} \ \forall\, 1 \leq k \leq K \end{cases} \end{aligned} \tag{5}$$

Ideally, the optimal solution to the relaxed problem in Eq 5 should be close to the optimal solution to the constrained problem in Eq 4. Nonetheless, the relaxed problem is prone to aggregation of errors. Moreover, it is important to note that the accuracy of approximating the constraint $\rho_{t_{k+1}} \approx q_{t_{k+1}}$ is dependent on the accuracy of approximating the previous constraint $\rho_{t_k} \approx q_{t_k}$. As a result, errors in approximation accumulate as t_k increases, potentially causing the terminal distribution ρ_T to deviate significantly from the true distribution q_T. This deviation can lead to erroneous information about the gene expressions of fully differentiated cells within the population. To mitigate this issue, we choose to enforce an exact terminal equality constraint $\rho_T = q_T$. By satisfying this constraint, we expect to improve the accuracy of approximations for the intermediate constraints. To fulfill the terminal constraint, we employ a Forward-Backward Stochastic Differential Equations (FBSDE) model proposed by Vargas et al [21]. In the following sections, we will describe how we integrate the TrajectoryNet framework and FBSDE to satisfy the terminal equality constraint and approximate the intermediate equality constraints. Specifically, our aim is to solve the following optimization problem:

$$\begin{aligned} J(q_0) &= \inf_{v} \tilde{J}(v, q_0) \\ &= \inf_{v} \mathbb{E}_{q_0}\left[\int_0^T \left(\frac{1}{2}||v(X_t, t)||^2 + \alpha_F \log(\rho(X_t, t))\right) dt\right] \\ &\text{subject to}: \begin{cases} dX_t = v(X_t, t)dt + \sigma dW_t \\ \rho(\cdot, 0) = q_0 \\ \rho(\cdot, t_k) \approx q_{t_k} \ \forall\, 1 \leq k \leq K-1 \\ \rho(\cdot, T) = q_T \end{cases} . \end{aligned} \tag{6}$$

Remark 4. *To better understand our model, imagine a string on a flat surface. The goal is to shape the string in a way that it connects two fixed points (initial and terminal constraints) while passing through specific points on the surface (intermediate constraints). TrajectoryNet holds one end of the string fixed while allowing the other end to roam freely, hoping to be close to the intermediate checkpoints and the terminal end. In contrast, our model fixes both ends of the string,* *increasing the likelihood of passing through the intermediate checkpoints and providing a better approximation of the intermediate constraints.*

Implementation

In order to solve the optimization problem in Eq 6, we need to address the following numerical problems.

Firstly, we need to compute the cost functional $\tilde{J}(\nu, X_0)$, which involves calculating the expectation of a stochastic integral over continuous time with respect to the initial distribution q_0. To approximate this integral, we adopt time discretization. By discretizing the SDE model on a dense set of time points between 0 and T, we can generate a finite number of simulated data to estimate the expectation. In particular, since the cost functional represents the expected cost of trajectories originating from initial points distributed according to q_0, we sample m cells from the observations at $t = 0$ and employ a discrete-time approximation of the SDE model to simulate m trajectories.

Secondly, we need to estimate the true distribution q_t and the generated distribution ρ_t at all time points. This requires computing the probability density functions (PDFs) at the selected time points used in the stochastic integral approximation. To accomplish this, we employ kernel density estimation of the empirical distributions.

Thirdly, we need to approximate the intermediate constraints $\rho_{t_k} \approx q_{t_k}$. In order to ensure that the simulated distributions closely match the empirical distributions at intermediate time points, we introduce a new penalty term along with the ones proposed by TrajectoryNet.

Lastly, it is crucial to satisfy the terminal equality constraint that ensures the agreement between the true distribution ρ_T and the simulated distribution q_T at the final time point T. To enforce this constraint, we incorporate the Forward-Backward Stochastic Differential Equations (FBSDE) model.

Additionally, in our notation, we assume the observations X_t are collected at $K + 1$ observed time points: $0 = t_0 < t_1 < t_2 < \ldots < t_{K-1} < t_K = T$. At each time point t_k, we have n_{t_k} observations, denoted by $x_{t_k,1}, x_{t_k,2}, \ldots, x_{t_k,n_{t_k}}$.

Estimation of cost functional. To estimate the cost functional, which represents the expectation of a stochastic integral, a discretization approach is employed. The integral of a function over continuous time is approximated by summing the function values multiplied by the interval lengths over numerous small intervals. Therefore, we need to generate a set of discrete time points for this purpose.

To be more specific, to account for the constraints imposed by the optimization problem in Eq 6 at the observed time points $\{t_k\}_{0\leq k\leq K}$, the set of discrete time points U is constructed by take a union of the observed time points $\{t_k\}_{0\leq k\leq K}$ and a denser set of equally spaced time points $\{t'_l\}_{0\leq l\leq L'}$ with $t'_0 = 0$ and $t'_{L'} = T$. With a bit abuse of notation, we denote $U = \{t'_0, t'_1, \ldots, t'_L\}$, where $0 = t'_0 < t'_1 \ldots t'_l < t'_L = T$. This incorporation of observed time points ensures that the discretization captures the necessary information at these specific instances.

Remark 5. *Consider, for example, a scenario where* $T = 1$ *and the observed time points are* 0.33 *and* 0.66. *In this case, the set U can be constructed by first merging the sets* {0, 0.05, 0.1, . . ., 0.95, 1} *and* {0.33, 0.66}. *The resulting set is then arranged in ascending order. Our empirical analysis of three datasets indicates that dividing the interval into approximately 200 subintervals tends to be adequate. Nonetheless, we advise users of our model to tailor the number of intervals to the specific characteristics of their data. Factors to consider include the quantity of observed time points and the overall data complexity. Moreover, it is beneficial to normalize the interval* $[0, T]$ *to* $[0, 1]$ *by applying a scaling factor of* $1/T$.

Additionally, the expectation is approximated by averaging m simulated paths generated through the approximation of the stochastic differential equation (SDE) $dX_t = v(X_t, t)dt + \sigma dW_t$. The simulation process begins with the selection of an initial set of m cells with gene expressions $x^*_{0,j}$ for $1 \leq j \leq m$, from the set of initial observations $\{x_{0,i}\}_{1 \leq i \leq n_0}$.

Based on the definition of U as the set of time points, the interval length $\Delta_l = t'_l - t'_{l-1}$ is assigned for $1 \leq l \leq L$. The simulations are generated using the Euler-Maruyama scheme as follows [21]:

$$x^*_{t'_l,j} = x^*_{t'_{l-1},j} + v(x^*_{t'_{l-1},j}, t'_{l-1})\Delta_l + \sigma\epsilon_{t'_{l-1},j}\sqrt{\Delta_l}. \tag{7}$$

Here, $x^*_{t'_l,j}$ represents the gene expression of the j-th simulated cell at time t'_l for $1 \leq j \leq m$ and $0 \leq l \leq L$. The noise terms $\epsilon_{t'_{l-1},j}$ are independent standard normal random variables. Eq 7 approximates the constraint $dX_t = v(X_t, t)dt + \sigma dW_t$. It is important to note that the accuracy of the approximation in Eq 7 improves as the number of simulated time points L increases.

The outlined approximation scheme produces m simulated paths. In order to assess the stochastic integral, it becomes necessary to estimate the probability density function (PDF) ρ_t for each $t \in U$. Furthermore, it is essential to estimate q_t to ensure that the constraints $\rho_t \approx q_t$ are satisfied at the observed time points t.

In particular, we can not obtain the true density functions ρ_{t_k} and q_{t_k}, and we need to replace them by their estimates $\hat{\rho}_{t_k}$ and $\hat{q}_{t_k}$ in solving the optimization problem in Eq 5. To estimate the true density function q_{t_k} and ρ_{t_k}, we use the Nadaraya-Watson density estimator. Based on the observations $x_{t_k,1}, x_{t_k,2}, \ldots, x_{t_k,n_{t_k}}$, we estimate q_{t_k} by

$$\hat{q}_{t_k}(x) = \frac{1}{n_{t_k}}\sum_{i=1}^{n_{t_k}} K_H(x - x_{t_k,i}),$$

where H is a diagonal matrix with diagonal entries h. The bandwidth h is tuned by cross validation.

Similarly, based on the simulated points $\{x^*_{t'_l,j} : 0 \leq l \leq L, 1 \leq j \leq m\}$, we can estimate the distribution of the simulated points at time t'_l by

$$\hat{\rho}_{t_k}(x) = \frac{1}{m}\sum_{j=1}^{m} K_H(x - x^*_{t'_l,j}),$$

Again, the same kernel function K_H is used to compute the empirical distributions.

To summarize, we employ the discretized SDEs and perform kernel density estimations to simulate multiple trajectories and estimate the expected value of the stochastic integral. This allows us to approximate the cost functional. Specifically, for a given action v, the cost functional is computed as the average of the following terms over the simulated trajectories:

$$\frac{1}{m}\sum_{j=1}^{m}\sum_{l=0}^{L}\frac{1}{2}\|v(\hat{x}_{t'_l,j}, t'_l)\|_2^2 + \alpha_F \log(\hat{\rho}_{t'_l}(\hat{x}_{t'_l,j})),$$

Approximation of intermediate constraints. In order to approximate the intermediate constraints $\hat{\rho}_{t_k} = \hat{q}_{t_k}$, we introduce an additional penalty term along with the penalties proposed by TrajectoryNet [18]. The purpose of this new penalty term is to enhance the recall of the model, as demonstrated in Fig 2b.

We start by introducing some notations. Let

$$D^p_{t_k,j} = \{\|x^*_{t_k,j} - x\| \mid x \in \{x_{t_k,i}\}_{1\le i\le n_{t_k}}\},$$
$$D^r_{t_k,i} = \{\|x_{t_k,i} - x\| \mid x \in \{x^*_{t_k,j}\}_{1\le j\le m}\}.$$

The set $D^p_{t_k,j}$ is the set of distances between the jth simulation at t_k and all observations at t_k. The set $D^r_{t_k,i}$ is the set of distances between the ith observation at t_k and all simulations at t_k. Furthermore, we order both sets in ascending order such that $d^{p(z)}_{t_k,j}$ denotes the zth smallest element in $D^p_{t_k,j}$ and $d^{r(z)}_{t_k,i}$ denotes the zth smallest element in $D^r_{t_K,i}$.

Subsequently, we implement the following penalty function g_k at observed time point t_k,

$$\begin{aligned} g_k = \quad & -\frac{\alpha_1}{mn_{t_k}} \sum_{j=1}^{m} \sum_{i=1}^{n_{t_k}} \log \hat{q}_{t_k}(x^*_{t_k,j}) \\ & + \frac{\alpha_2}{m} \sum_{j=1}^{m} \sum_{z=1}^{Z} d^{p(z)}_{t_k,j} \qquad\qquad . \\ & + \frac{\alpha_3}{n_{t_k}} \sum_{i=1}^{n_{t_k}} \sum_{z=1}^{Z} d^{r(z)}_{t_k,i} \end{aligned} \tag{8}$$

Here, α_1, α_2, and α_3 are positive constants used for regularization.

The first two penalty terms in the penalty function are proposed by TrajectoryNet. The former term applies the maximum likelihood function to the simulations, computed from empirical distributions. It aims to encourage the simulations to closely match the empirical distribution of the observations at time t_k. This term ensures that the simulations capture the statistical characteristics of the observed data.

The second term aims to improve the precision of the model by penalizing large distances between the simulated data points and observations. It pushes the simulations to be closer to the observed data points, promoting a better alignment between the simulated trajectories and the actual observations. By minimizing this penalty term, the model becomes more precise in capturing the observed trajectory patterns.

We introduce the third penalty term to enhance the recall of the model. It ensures that a larger number of observations are included in a close neighbourhood around the simulations. This term is necessary when the simulated data points are only close to a few observations, resulting in the first penalty term being close to zero, while the distribution of simulated data points, $\hat{\rho}_{t_k}$, is not close to the empirical distribution $\hat{q}_{t_k}$ (see Fig 2a and 2b).

With the combination of these three penalty terms, we aim to improve both precision and recall, leading to more accurate trajectory inference.

Satisfying terminal constraint. This section presents a detailed implementation of Forward-Backward Stochastic Differential Equations (FBSDE) in order to achieve the terminal constraint $\hat{\rho}_T = \hat{q}_T$ mentioned earlier.

In the continuous setting, an FBSDE model is a system of two stochastic differential equations where X_t is the forward process and Y_s is the backward process such that,

$$dX_t = v(X_t, t)dt + \sigma dW^f_t,$$
$$dY_s = u(Y_s, s)ds + \sigma dW^b_s,$$
$$X_t \overset{d}{=} Y_{T-t} \sim \rho_t,$$

where $\{W_t^f\}_{0\leq t\leq T}$ and $\{W_s^b\}_{0\leq s\leq T}$ are two independent standard Brownian motions. The stochastic processes $\{X_t\}_{0\leq t\leq T}$ and $\{Y_s\}_{0\leq s\leq T}$ in $\mathbb{R}^d$ with constant volatility σ are involved in the FBSDE model. The process $\{X_t\}_{0\leq t\leq T}$ is equivalent to the trajectory expressed in Eq 1, describing the gene expression changes of a cell over time. The forward drift v specifies the current gene expression of a cell and current time to the direction in which the cell changes its gene expression at that time. The backward process $\{Y_s\}_{0\leq s\leq T}$ reverses the sequence $\{X_0, X_1, \ldots, X_T\}$. In other words, if the distribution of gene expressions of fully differentiated cells is ρ_T at terminal time T, the initial distribution ρ_0 at time 0 can be obtained by applying the backward process to the random variable X_T. Finally, the mathematical connection between the forward drift v and the backward drift u is stated in the following theorem.

Theorem 1 (Nelson's Duality Theorem [27]).

$$v(x,t) - u(x, T-t) = \sigma\nabla \log_x \rho_t(x) \tag{9}$$

Theorem 1 establishes a relationship between the forward drift v for a cell with gene expression x at time t and the backward drift u for a cell with the same gene expression x at time $T - t$. Based on this relationship, Vargas et al. [21] proposed an algorithm to solve a simplified optimization problem described as follows:

$$\begin{aligned} J(q_0) &= \inf_v \tilde{J}(v, q_0) \\ &= \inf_v \mathbb{E}_{\rho_0}\left[\int_0^T \left(\frac{1}{2}||v(X_t,t)||^2 + \alpha_F \log(\rho(X_t,t))\right)dt\right] \\ &\text{subject to}: \begin{cases} dX_t = v(X_t,t)dt + \sigma dW_t \\ \rho(\cdot,0) = q_0 \\ \rho(\cdot,T) = q_T \end{cases}. \end{aligned} \tag{10}$$

However, since this problem only considers the initial and terminal constraints, we need to incorporate penalty terms, as discussed in **Approximation of intermediate constraints**, to approximate the intermediate constraints presented in our main optimization problem described as follows:

$$\begin{aligned} J(\hat{q}_0) &= \min_v \left(\frac{1}{m}\sum_{j=1}^{m}\sum_{l=0}^{L}\frac{1}{2}\|v(x^*_{t'_l,j}, t'_l)\|_2^2 + \alpha_F \log(\hat{\rho}_{t'_l}(x^*_{t'_l,j}))\right) + \left(\sum_{k=1}^{K-1} g_k\right) \\ &\text{subject to}: \begin{cases} x^*_{t'_l,j} = x^*_{t'_{l-1},j} + v(x^*_{t'_{l-1},j}, t'_{l-1})\Delta_l + \sigma\epsilon_{t'_{l-1},j}\sqrt{\Delta_l} \\ \epsilon_{t'_{l-1},j} \sim \mathcal{N}(0,1) \\ \hat{q}_0 = \hat{\rho}_0 \\ \hat{q}_T = \hat{\rho}_T \end{cases}. \end{aligned} \tag{11}$$

The algorithm proposed by Vargas et al. [21], as shown in Fig 1, consists of an iterative procedure involving two optimization problems: a forward optimization problem and a backward optimization problem. The forward optimization problem focuses on enforcing the constraint $\rho_0 = q_0$ and approximating the remaining constraints $\rho_{t_k} \approx q_{t_k}$ for all $k = 1, 2, \ldots, K$. Conversely, the backward optimization problem focuses on enforcing the constraint $\rho_T = q_T$ and approximating the constraints $\rho_{t_k} = q_{t_k}$ for $k = 0, 1, 2, \ldots, K - 1$. Both optimization problems incorporate a new penalty term. Specifically, the forward optimization problem penalizes the discrepancy between simulations generated by the forward drift and references generated by the solution to the backward optimization problem. Similarly, the backward optimization problem penalizes the discrepancy between simulations generated by the backward drift and references generated by the solution to the forward optimization problem.

The forward optimization problem is formulated as follows. Let u' be the backward drift, and let $x'_{0,j}$ be the initial points sampled from observations $x_{0,i}$. We define the m **reference** paths as:

$$x'_{t'_l,j} = x'_{t'_{l-1},j} + (u'(x'_{t'_{l-1},j}, t'_{l-1}) + \sigma\nabla \log_x \rho_{t'_{l-1}}(x'_{t'_{l-1},j}))\Delta_l + \sigma\epsilon_{t'_{l-1},j}\sqrt{\Delta_l}, \tag{12}$$

where $x'_{t'_l,j}$ represents the j-th reference at time t'_l.

The corresponding forward optimization problem becomes:

$$\begin{aligned} J(\hat{q}_0) \quad &= \min_{v}\left(\frac{1}{m}\sum_{j=1}^{m}\sum_{l=0}^{L}\frac{1}{2}\|v(x^*_{t'_l,j}, t'_l)\|_2^2 + \alpha_F \log(\hat{\rho}(x^*_{t'_l,j}))\right) + \left(\sum_{k=1}^{K} g_k\right) \\ &\quad + \left(\frac{\beta_f}{m}\sum_{j=1}^{m}\sum_{l=1}^{L}\|x^*_{t'_l,j} - x'_{t'_l,j}\|_2^2\right) \\ &\text{subject to}: \begin{cases} x^*_{t'_l,j} = x^*_{t'_{l-1},j} + v(x^*_{t'_{l-1},j}, t'_{l-1})\Delta_l + \sigma\epsilon_{t'_{l-1},j}\sqrt{\Delta_l} \\ x^*_{0,j} = x'_{0,j} \\ \epsilon_{t'_{l-1},j} \sim \mathcal{N}(0,1) \\ \hat{\rho}_0 = \hat{q}_0 \end{cases} . \end{aligned} \tag{13}$$

The corresponding forward optimization problem, denoted by Eq 13, aims to minimize the cost functional $J(\hat{q}_0)$ with respect to the action v.

Note that the objective function of the forward optimization problem includes an additional penalty term (with regularization constant β_f tuned by cross validation) to account for the divergence between the simulations $x^*_{t'l,j}$ and the reference paths $x'_{t'_l,j}$. This term penalizes the differences between the trajectories generated by the optimal forward drift and the reference paths. Additionally, the terminal equality constraint $\hat{\rho}_T = \hat{q}_T$ is converted into the penalty term g_K in the objective function.

In summary, the first three terms in the objective function are similar to the mean field problem described in Eq 11, with an additional term that quantifies the differences between the simulations generated by the optimal forward drift and the reference paths generated by the backward drift.

The backward optimization problem is defined as follows. Let v' be the forward drift, and let $x'_{T,j}$ be the terminal points sampled from observations $x_{T,i}$. we define the m **reference** paths as:

$$x'_{t'_{l-1},j} = x'_{t'_l,j} + (v'(x'_{t'_l,j}, t'_l) - \sigma\nabla \log_x \rho_{t'_l}(x'_{t'_l,j}))\Delta_l + \sigma\epsilon_{t'_l,j}\sqrt{\Delta_l}, \tag{14}$$

where $x'_{t'l-1,j}$ represents the j-th reference path at time t'_{l-1}.

The corresponding backward optimization problem becomes:

$$\begin{aligned} J(\hat{q}_T) \quad &= \min_u \left(\frac{1}{m} \sum_{j=1}^{m} \sum_{l=0}^{L} \frac{1}{2} \| u(x^*_{t'_{L-l},j}, T - t'_l) \|_2^2 + \alpha_F \log(\hat{\rho}_{t'_l}(x^*_{t'_l,j})) \right) \\ &\qquad + \left(\sum_{k=0}^{K-1} g_k \right) + \left(\frac{\beta_b}{m} \sum_{j=1}^{m} \sum_{l=1}^{L} \| x^*_{t'_l,j} - x'_{t'_l,j} \|_2^2 \right) \\ &\text{subject to :} \begin{cases} x^*_{t'_l,j} = x^*_{t'_{l-1},j} + u(x^*_{t'_{l-1},j}, t'_{l-1})\Delta_l + \sigma \epsilon_{t'_{l-1},j} \sqrt{\Delta_l} \\ x^*_{T,j} = x'_{T,j} \\ \epsilon_{t'_{l-1},j} \sim \mathcal{N}(0,1) \\ \hat{\rho}_T = \hat{q}_T \end{cases} . \end{aligned} \tag{15}$$

Similar to the forward optimization problem, the equality constraint $\hat{\rho}_0 = \hat{q}_0$ is converted into the penalty term g_0. The objective function of the backward optimization problem includes the first three terms, which are the same as the mean field problem described in Eq 11. Additionally, there is an extra term that quantifies the differences between the simulations generated by the optimal backward drift and the references generated by the forward drift. This term captures the discrepancies between the backward simulations and the forward references.

To summarize, the proposed algorithm by Vargas et al. [21] iterates between the forward and backward optimization problems, where the optimal forward drift generates references for the computation of the optimal backward drift, and vice versa. In each iteration, penalty terms are used to penalize the differences between simulations generated by the optimal drift in one direction and references generated by the drift in the opposite direction. The algorithm has been shown to converge [21, 22, 26, 28], with the optimal forward drift satisfying the terminal constraint $\hat{\rho}_T = \hat{q}_T$ and the optimal backward drift satisfying the initial constraint $\hat{\rho}_0 = \hat{q}_0$. This implies that the optimal forward drift obtained from this iterative process is indeed the solution to the original optimization problem in Eq 11, satisfying both the initial and terminal constraints.

To prepare for the computational algorithm, we will parameterize the drift functions v and u using neural networks that will be denoted as φ and ψ, respectively. Specifically, we will initialize the parameters in these neural networks so that they can be trained to approximate the desired drift functions. Both neural networks will be fully connected, with default setting of three hidden layers each containing 128 units, and they will use ReLU activation functions throughout.

Algorithm 1 Solving FBSDE Model in Eq 11

```
Initialize: $\phi_0$, $\psi_0$
N ⇐ 1
while $N < N_{\max}$ do
  Sample $\{x'_{t'_l,j}\}_{1\le j\le m, 0\le l\le L}$ with $u(y, s; \psi_{N-1})$ and Eq 12
  $\phi \Leftarrow \phi_{N-1}$
  while not converged do
    Simulate $\{x^*_{t'_l,j}\}_{1\le j\le m, 0\le l\le L}$ with $v(x, t; \phi)$
    Compute $J(v, \hat{q}_0)$ in Eq 13
    $\phi \Leftarrow$ StochasticGradientDescent($\nabla_\phi J(v, \hat{q}_0)$)
  end while
  $\phi_N \Leftarrow \phi$
  Sample $\{x'_{t'_l,j}\}_{1\le j\le m, 0\le l\le L}$ with $v(y, s; \phi_N)$ and Eq 14
  $\psi \Leftarrow \psi_{N-1}$
  while not converged do
    Simulate $\{x^*_{t'_l,j}\}_{1\le j\le m, 0\le l\le L}$ with $u(x, t; \psi)$
    Compute $J(u, \hat{q}_0)$ in Eq 15
    $\phi \Leftarrow$ StochasticGradientDescent($\nabla_\phi J(u, \hat{q}_0)$)
  end while
  $\psi_N \Leftarrow \psi$
  $N \Leftarrow N + 1$
end while
return $\phi$, $\psi$
```

Parameter tuning. In alignment with TrajectoryNet methodology, we conducted a grid search for optimal parameter values, testing $\alpha_F, \alpha_1, \alpha_2, \alpha_3 \in \{1, 5, 10\}$ and $\sigma \in \{0.01, 0.1, 1\}$, with performance evaluated on a validation set. We found that for the volatility term σ, values between 0.01 and 0.1 are effective for normalized data, as higher values lead to "blurry" trajectories due to excessive noise, while values approaching zero render the model near-deterministic, similar to a Partial Differential Equation (PDE) model, rather than an SDE model.

Remark 6. *Similar to TrajectoryNet, our FBSDE model framework has the capacity to incorporate a velocity penalty term. In our extensive parameter validation studies, we observed that including this velocity penalty did not markedly influence the performance of the FBSDE model across the datasets we examined. This minimal impact may be specific to the unique attributes of our selected datasets. It is conceivable that other datasets with distinct characteristics could benefit more substantially from the incorporation of a velocity penalty. Recognizing the potential variability in utility, the developers of TrajectoryNet have provided the velocity penalty as an optional feature in their code repository. This choice reflects the term's adaptable relevance depending on the dataset being analyzed. In alignment with this adaptable methodology, we have similarly implemented the velocity penalty as an optional element in our FBSDE model codebase. This allows users the flexibility to activate it when analyzing datasets where RNA velocity considerations are deemed important.*

Results

Overview of the FBSDE model

Fig 1 provides an overview of the FBSDE. The key innovation of our approach over previously published methods such as TrajectoryNet [18] is that we model the population growth in both forward and backward directions. As such, the FBSDE framework consists of two parts, the forward model and the backward model; the forward model is essentially the same as TrajectoryNet. It has been widely observed that using Forward model alone may not always satisfy the terminal equality constraints if the true trajectory has many non-linear segments [21]. This can be attributed to the lack of closed form solutions in modelling stochastic differential equations [22]. In other words, the stochastic differential equations cannot be solved analytically. As an alternative, numerical approximations sometimes can alleviate these difficulties, but their performances are not always satisfactory. By including a Backward model in the FBSDE framework, one can convert the terminal constraint $\hat{\rho}_T = q_T$ to an initial constraint for the Backward model which can be numerically satisfied [21]. Furthermore, the iteration between the Forward model and the Backward model has been proven to converge to an optimal solution where both the initial constraint $\hat{\rho}_0 = \rho_0$ and the terminal constrain $\hat{\rho}_T = \rho_T$ are both satisfied [28].

As alluded to in the Introduction, optimal transport (OT) has been recently adopted in modelling cell growth trajectory and achieved promising performance [11, 12]. Typically, OT is more effective in modelling linear trajectories, whereas SDE can model non-linear trajectories. Fig 2c and 2d illustrate the differences between OT and SDE. An OT model connects the endpoints with a straight line. In contrast, an SDE model infers non-linear trajectories connecting the endpoints. In addition, an OT model (Fig 2c) can only infer trajectories whose endpoints are observations whereas an SDE model (Fig 2d) can generate trajectories originating from any arbitrary starting point. Fig 2c and 2d display the trajectories of the cells from time $t = 0$ to $t = 1$ and demonstrate how to interpolate their gene expression at some intermediate time point $t = h$, where $0 < h < 1$. The generative modelling differences between OT and SDE are further illustrated in Fig 3. When given a set of initial and terminal distributions, OT can only predict a linear trajectory, as depicted in Fig 3e. On the other hand, SDE can infer trajectories with various shapes depending on the parameter choices, as shown in **Fig** fig:circlea-d. Specifically, the shapes of the trajectories are affected by the extent to which the population accommodates high entropy (i.e. more crowded), as indicated in this example. As shown in **Fig** fig:circlea-d, the paths display greater angular rotation as the model is trained to accommodate higher levels of entropy. This is reflected in the curly shape of the paths, which allows the population to remain denser for longer periods. In contrast, optimal transport employs linear path interpolations as shown in Fig 3e.

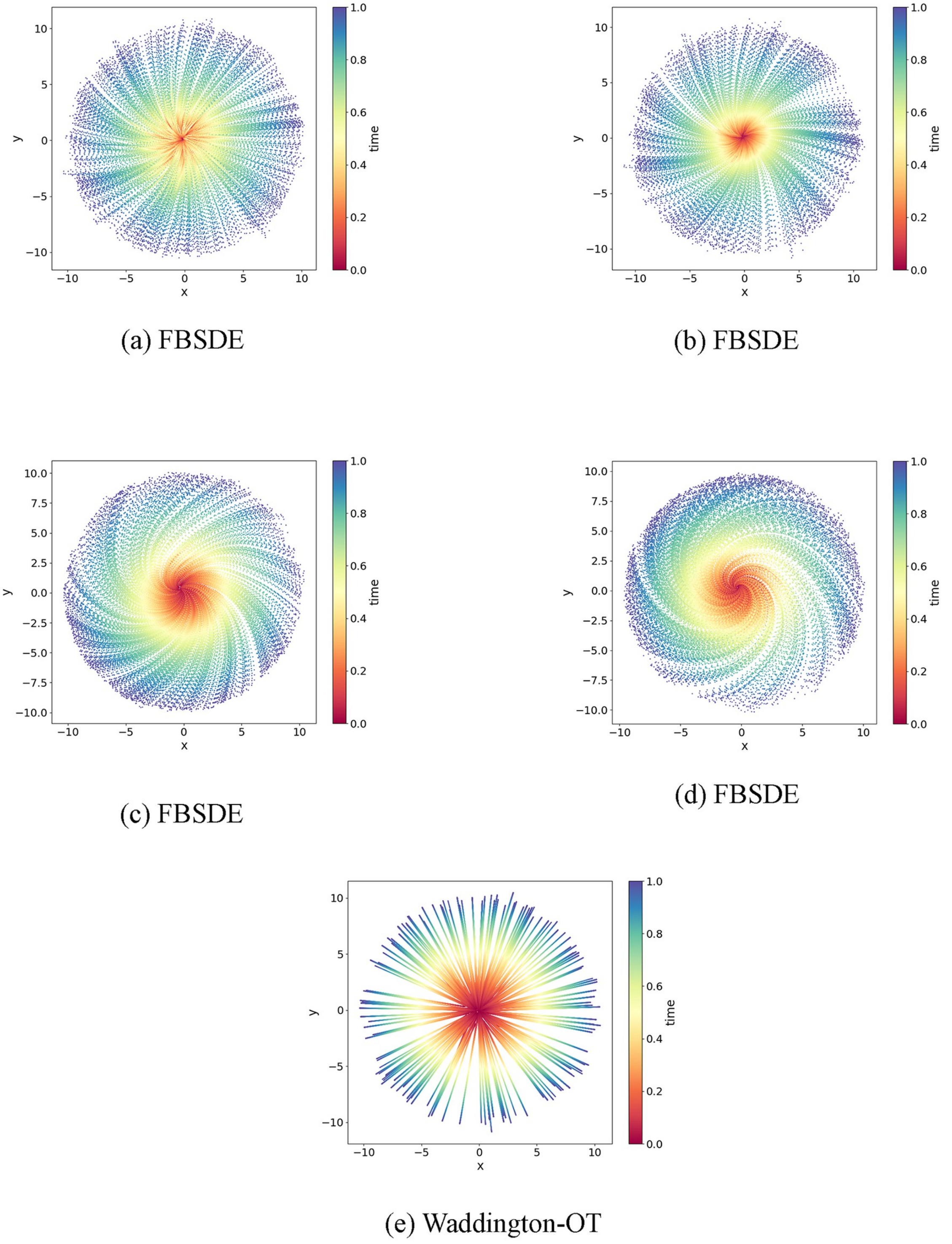
Performance on a human embryonic stem cells dataset

We compared the FBSDE model with WOT and TrajectoryNet by visually examining the generative power using two-dimensional human embryonic stem cells data obtained from Moon et al [23]. The dataset contained 16, 825 cells grown as embryoid bodies, which were sampled from 10 time points over a period of 27 days at 3-day intervals. All three models were provided with the same starting population of cells at the initial time point ($t = 0$) and generated samples at 200 equally spaced time points in addition to the already observed time points. Fig 4 illustrates the comparison among the ground truth distribution and samples generated by all three models, demonstrating that our FBSDE model generated samples that more closely resemble the observations from human embryonic stem cells data.

Furthermore, Fig 5b highlights why linear interpolations can fail in trajectory inference. The WOT model interpolated some points mainly in the upper portion of the figure that were not seen in real observations. Fig 5c and 5d show that both TrajectoryNet and FBSDE generated simulations that seemed to match the observations at the terminal time point $t = T$. In addition, both TrajectoryNet and FBSDE generate smooth trajectories as seen in the observations (Fig 5a) in contrast to the segmented linear trajectories generated by the WOT model. However, at intermediate time points, the FBSDE generated simulations represented the observations more accurately than TrajectoryNet. Finally, the FBSDE model visually resembled the observations more closely than the other two models, albeit by a small margin.

Performance on a mouse embryonic fibroblasts dataset

We further evaluated the generative power of FBSDE, WOT, and TrajectoryNet using a mouse embryonic fibroblasts dataset obtained from Schiebinger et al [11]. This dataset contains 259, 155 cells sampled at 36 time points over an 18-day period. Fig 5 shows that all three models performed relatively well on this dataset. Notably, the observations in Fig 5a exhibit two distinct sharp turns (indicated by arrows in Fig 5) in the shape, which were effectively captured by both WOT (Fig 5b) and FBSDE (Fig 5d) models but not recognized by TrajectoryNet (Fig 5c). However, unlike WOT, which relies on simulations using observed points (as depicted in Fig 3c), FBSDE utilizes a neural network to fully parameterize the population movement. FBSDE is a mesh-free method, which means that once a model has been fitted, simulations can be generated without relying on observations. This demonstrates that the FBSDE model



(a) FBSDE

(b) FBSDE

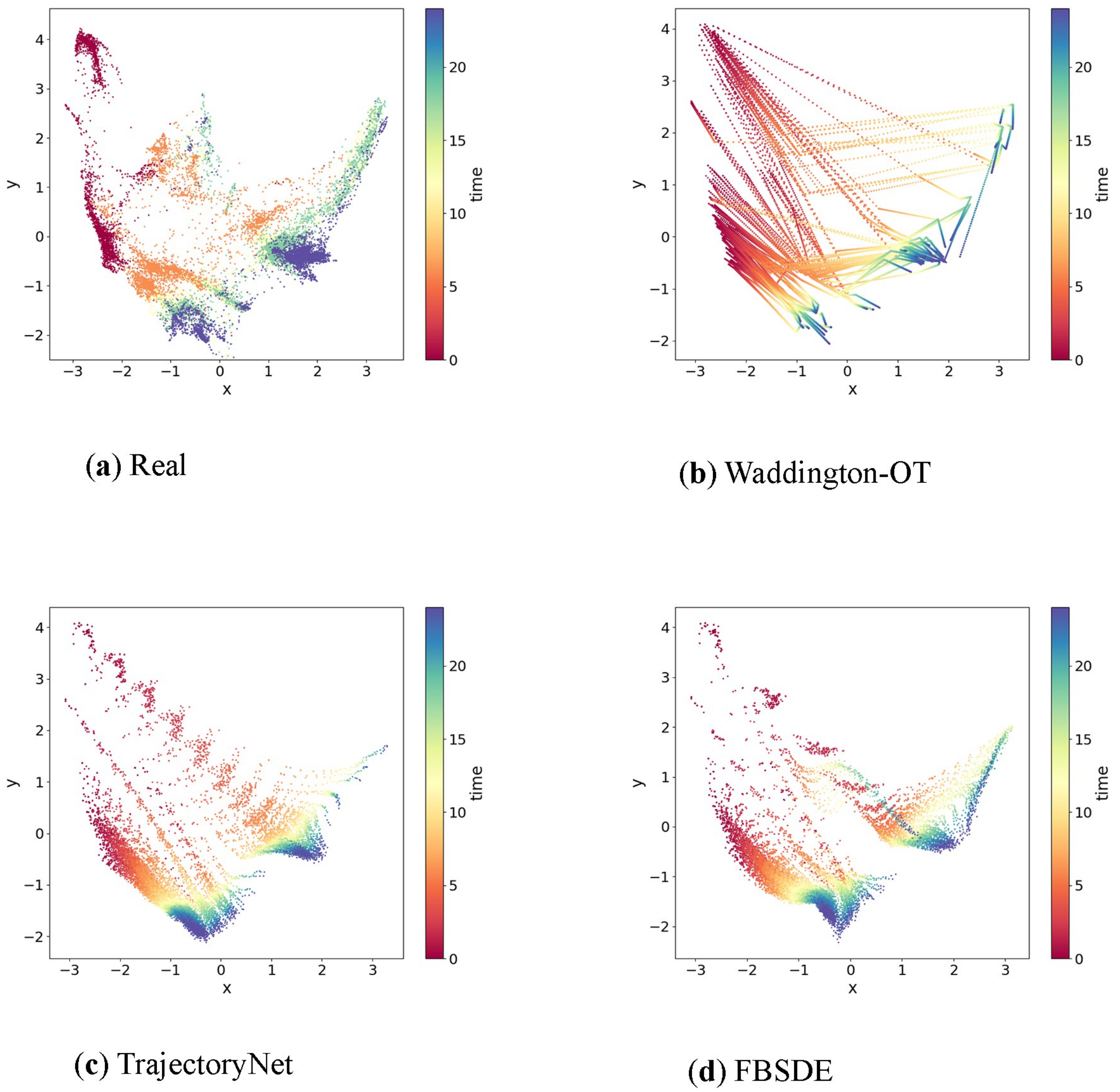
(c) FBSDE

(d) FBSDE

(e) Waddington-OT

Fig 3. Difference between FBSDE and Waddington-OT in modelling a simulated dataset. The time is coded in the vertical bars next to each panel; the time points are normalized with 0 and 1 representing starting and end point, respectively. At $t = 0$, the population follows the standard normal distribution. At $t = 1$, the population is uniformly distributed on the ring centred at (0, 0) with a radius of 10. At $0 < t < 1$, the distribution is interpolated by either the FBSDE or Waddington-OT model. In **Fig** 3a, **Fig** 3b, **Fig** 3c and **Fig** 3d, the population evolves based on a stochastic differential equation where a neural network parametrizes the drift term. From **Fig** 3a to **Fig** 3d, the population exhibits more angular rotation as it favours a higher level of entropy (i.e. density). In **Fig** 3e, the endpoints for each point at $t = 0$ are sampled based on the optimal transport map and the interpolation is performed linearly by connecting each pair of points.

https://doi.org/10.1371/journal.pcbi.1012015.g003



(a) Real

(b) Waddington-OT

(c) TrajectoryNet

(d) FBSDE

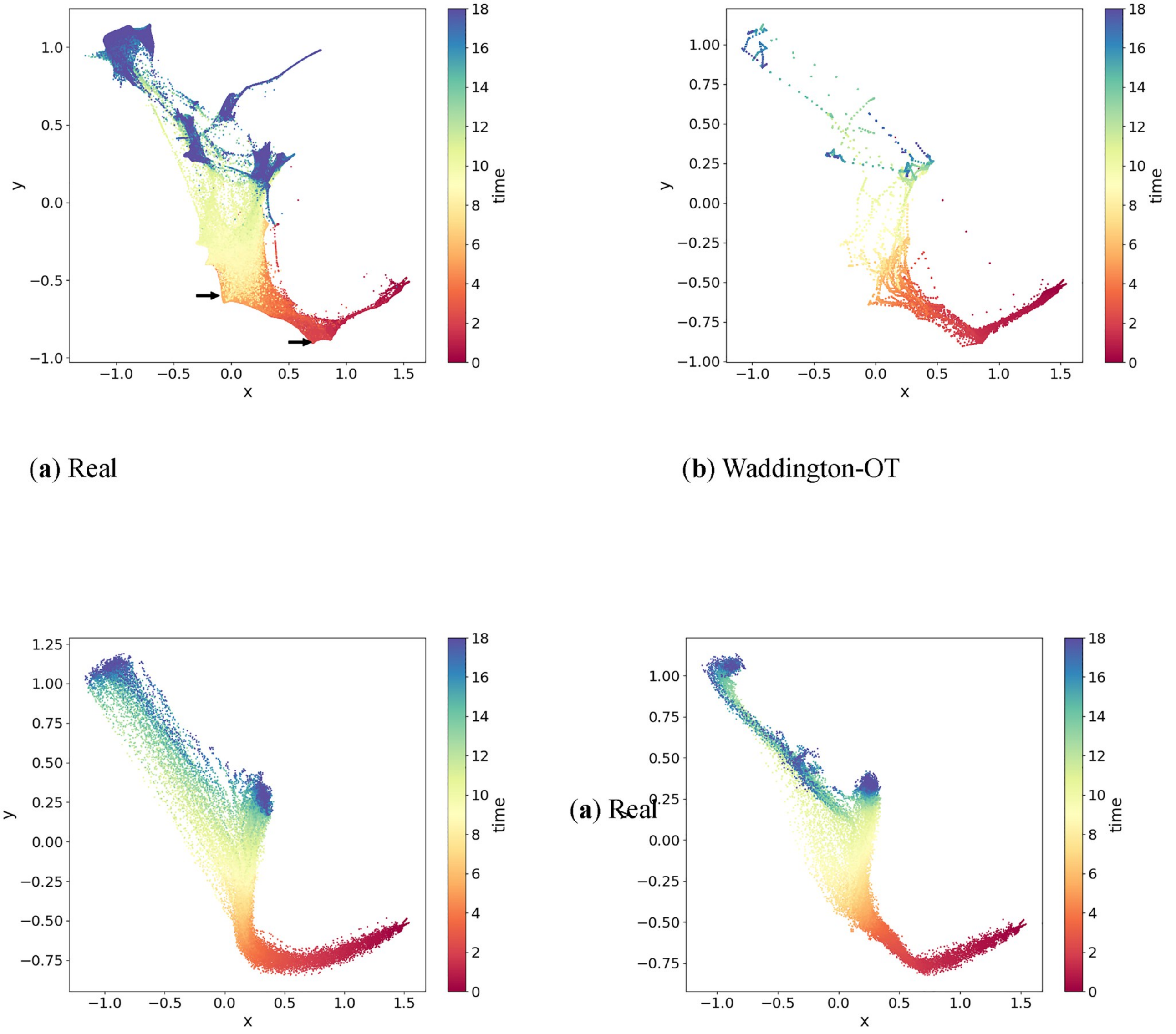
Fig 4. Performances on a human embryonic stem cells dataset. Fig 4a) Gene expression profiles of single cells are reduced to two dimensions using the PHATE method. The data correspond to a total of 10 time points over 27 days at 3 days intervals. The time points are color coded on a spectrum from red to deep blue. **Fig** 4b-d) Trajectory inference by three different methods (Waddington-OT, TrajectoryNet and FBSDE). In each experiment, a total of 200 equally spaced time points are added between the starting and end time points. As can be seen, FBSDE provides a trajectory most resembling the ground truth in **Fig** 4a.

https://doi.org/10.1371/journal.pcbi.1012015.g004

has the ability to capture not only smooth but also sharp shape features of developmental trajectories. The successful performance of FBSDE on this dataset provides further evidence of its effectiveness in accurately modelling complex biological processes. Furthermore, a thorough evaluation of the numerical performances of these three methods will be conducted using cross-validation, which will be discussed in detail in **Cross-validation study**.

Performance on a *Arabidopsis thaliana* stem cells dataset

Finally, we compared the performance of FBSDE, WOT, and TrajectoryNet on the Arabidopsis thaliana stem cells dataset generated by Shahan and colleagues [30]. This dataset contains



(**a**) Real

(**b**) Waddington-OT

(**a**) Real

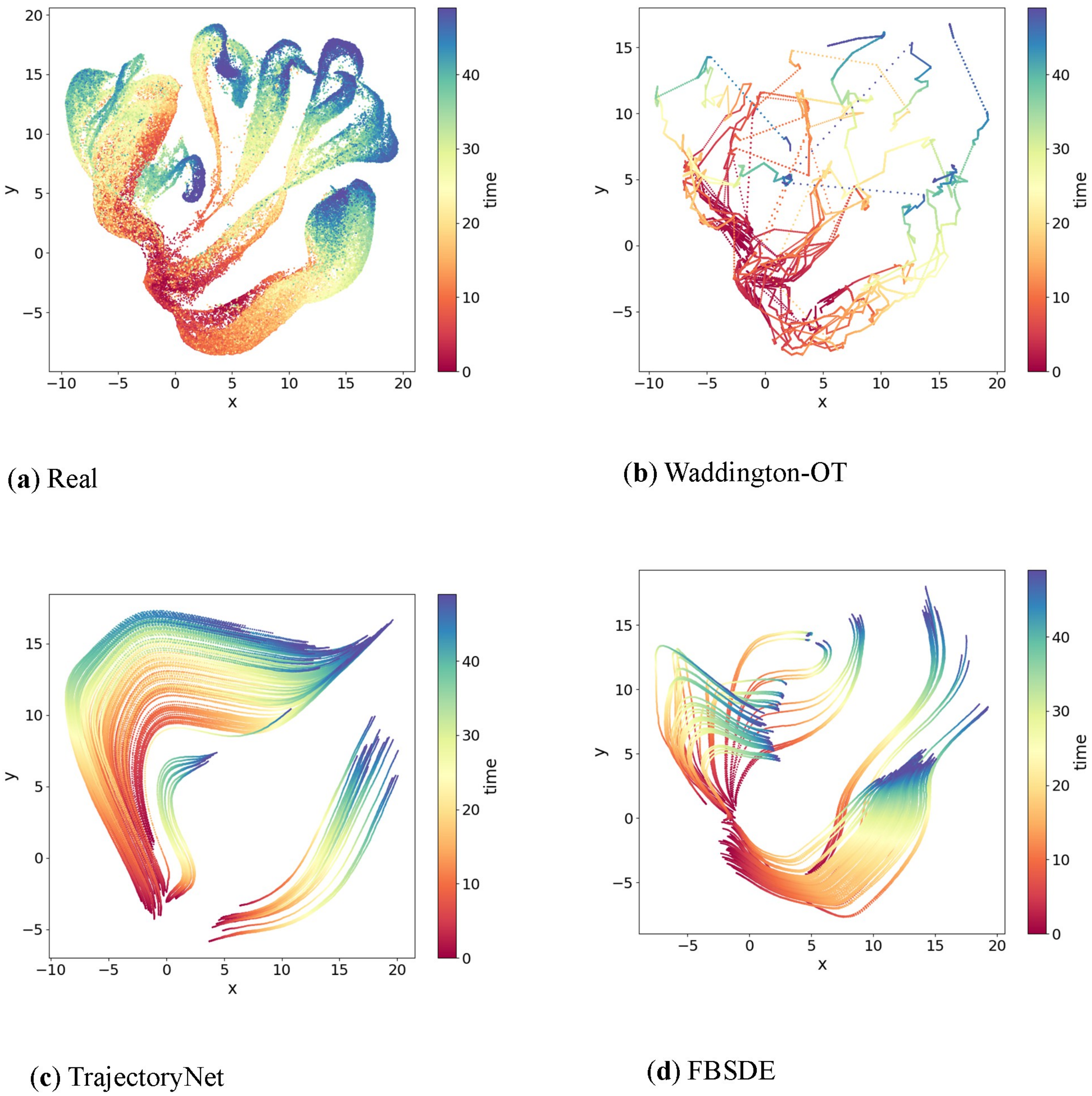
(**c**) TrajectoryNet

(**e**) FBSDE

Fig 5. Performance on a mouse embryonic fibroblasts dataset. Fig 5a) Gene expression of single cells are reduced to two dimensions using the force-directed layout embedding proposed in the study by Weinreb and colleagues [29]. The data correspond to a total of 37 time points over 18 days at 12 hours intervals. The time points are color coded on a spectrum from red to deep blue. Two visible sharp turns are indicated by arrows. **Fig** 5b-d) Trajectory inference by three different methods (Waddington-OT, TrajectortNet and FBSDE). In each experiment, a total of 200 equally spaced time points are added between the starting and end time points. The same color code is used. As can be seen, all three methods provide trajectories similar to the ground truth in **Fig** 5a although FBSDE outperforms the others modestly.

https://doi.org/10.1371/journal.pcbi.1012015.g005

110, 427 cells with pseudotime labels ranging from 0 to 50 and exhibits several distinct developmental branches. The use of pseudotime, as established by Shahan et al. [30], provides a meaningful way to encode temporal information, particularly in cases where absolute time points are not available or are less relevant. Fig 6a illustrates the complexity of this dataset, making it the most challenging to model compared to the previous two datasets. Fig 6c shows



(a) Real

(b) Waddington-OT

(c) TrajectoryNet

(d) FBSDE

Fig 6. Performance on a *Arabidopsis thaliana* stem cells dataset. Fig 6a) Gene expression of single cells are reduced to two dimensions using the UMAP embedding method. The data correspond to a total of 51 time points on a pseudotime scale from 0 to 50. **Fig** 6b-d) Trajectory inference by three different methods (Waddington-OT, TrajectoryNet and FBSDE). in each experiment, a total of 200 equally spaced time points are added between the starting and end time points. The same color code is used. As can be seen. Waddington-OT produces some unrealistic paths among different branches in **Fig** 6a. TrajectoryNet fails to capture the geometric features of the true trajectories. FBSDE produces some trajectories that resemble the ground truth to some degree.

https://doi.org/10.1371/journal.pcbi.1012015.g006

that TrajectoryNet simulations exhibit a significant visual discrepancy from the observations in Fig 6a. The simulated distributions at observed time points do not match the observed data points, reflecting the challenging nature of the dataset. The WOT model simulations in Fig 6b resemble the observations at observed time points, but there appear to be many crossings among branches that are not likely to be biologically relevant. In contrast, the FBSDE simulations in Fig 6d resemble the observations more closely, with a few recognizable branches and almost no unrealistic long-range crossings seen. In summary, none of the models perform well on the Arabidopsis thaliana stem cells dataset. However, the WOT model retains shape at observed time points, while the FBSDE model is capable of distinguishing some developmental branches.

Cross-validation study

In this study, we performed a comprehensive benchmarking of the FBSDE method against two other prominent approaches, Waddington-OT (WOT) and TrajectoryNet. To evaluate their performances, we utilized cross-validation techniques on several well-established single-cell datasets.

For each validation set, we considered a total of $n + 1$ observed points labeled from t_0 to t_n. In the validation process, we randomly selected 20% of the time points as the test data, while the remaining time points were used for training. The model then interpolated the data from the selected 20% time points, as depicted in Fig 2, and compared the interpolated data with the ground truth (i.e., the test data). This process was repeated for a total of $M = 50$ times to ensure robustness, and the average performance metric across all repetitions was reported.

To assess the accuracy of interpolation, we chose the Wasserstein Distance W_1 as the standard metric to measure the divergence between the interpolated data and the test data.

Table 1 offers a comprehensive comparison of the numerical performance of three models across various datasets. Additionally, we assess a variant of our model, named FBSDE_minus. This variant maintains the exact terminal equality constraint but does not include our novel penalty term, which is designed to improve intermediate constraint approximation. We also compare our results with the Stationary-OT method, as adapted by Shahan et al. [12, 30], which modifies the regularization term for better numerical stability.

For the human embryonic stem cells dataset, both the WOT and our FBSDE model demonstrate comparable accuracy in aligning with the empirical distributions. In the *Arabidopsis thaliana* stem cells dataset, the WOT model shows higher precision (defined in Fig 3a), primarily because its simulations are based on observational data. However, the generative strengths of SDE models like TrajectoryNet and FBSDE become more prominent when simulating a larger number of time points, as seen in the mouse embryonic fibroblasts dataset. Here, our FBSDE model notably outperforms both WOT and TrajectoryNet.

The comparison between FBSDE_minus and the full FBSDE model reveals a substantial gap in performance. The FBSDE_minus variant, lacking the new penalty term, exhibits

Table 1. Performance comparison.

	MEF	ATS	HES
FBSDE	0.270 +- 0.018	0.497 +- 0.006	0.611 +- 0.011
FBSDE_minus	0.627 +- 0.001	1.154 +- 0.001	0.722 +- 0.002
TrajectoryNet	0.566 +- 0.002	0.956 +- 0.004	0.647 +- 0.005
TrajectoryNet_plus	0.559 +- 0.002	0.861 +- 0.005	0.741 +- 0.005
Waddington-OT	0.644 +- 0.008	0.485 +- 0.008	0.622 +- 0.046
Stationary-OT	0.720 +- 0.009	0.446 +- 0.009	0.672 +- 0.081

The performance metric is the Wasserstein Distance W_1 with mean +- standard deviation reported. HES: Human Embryonic Stem Cells Dataset. MEF: Mouse Embryonic Fibroblasts Dataset. ATS: *Arabidopsis thaliana* Stem Cells Dataset.

https://doi.org/10.1371/journal.pcbi.1012015.t001 significantly reduced accuracy in trajectory inference, highlighting the importance of this feature in our complete model. Additionally, the introduction of the TrajectoryNet_plus variant, which incorporates an additional recall penalty term while omitting the terminal equality constraint, results in an observable enhancement in accuracy relative to the original TrajectoryNet model. This improvement highlights the significance of the recall penalty term. Importantly, the integration of the additional penalty term with the satisfaction of the terminal constraint in the FBSDE model significantly enhances model accuracy. Finally, the Stationary-OT method displays a performance level similar to that of Waddington-OT.

In conclusion, our cross-validation study indicates that, for the three datasets tested, the FBSDE model stands out among current methods in single cell trajectory inference.

Discussion

In this paper, we present a new method called Forward-Backward Stochastic Differential Equations (FBSDE) to model the developmental trajectories of single cells and compare it to two other recent methods–Waddington-OT and TrajectoryNet. The FBSDE method has several advantages. First, the use of stochastic differential equations allows for the generative modelling of developmental trajectories on a continuous space-time manifold. Second, iterations between the forward SDE and the backward SDE converge to a solution that satisfies both the initial and terminal probability distribution constraints leading to higher accuracy in trajectory modelling. Third, the FBSDE method integrates concepts from mean field theory to mimic the effects of cell-to-cell interactions allowing for more realistic biological modelling.

We apply all three methods on several scRNA-seq data sets. In summary, the FBSDE method outperforms the other two methods significantly. In situations where accurate modelling is difficult to achieve (e.g., *Arabidopsis thaliana* stem cells dataset), the FBSDE method still retains a fairly high level of similarity to the ground truth.

Although FBSDE demonstrates great promise and success in generative modelling of single cell developmental trajectories, there exist a few limitations. Firstly, our model is applied to a two-dimensional dataset, which has been processed and dimensionally reduced from an initial dimensionality of up to 20,000 genes. While the model is promising in low dimensions, the curse of high dimensionality should not be overlooked in cases where a large number of genes cannot be reduced in a dataset. Recent works by Chen et al [31] and Liu et al [32] aim to address such a problem where each gene of each cell is modelled by a separate stochastic differential equation. In addition, Qiu et al [33] proposed to incorporate domain knowledge while modelling these differential equations. Secondly, the interactions currently implemented in the FBSDE model do not cover a wide range of biological interactions. Despite their effectiveness, more complex interactions that model typical biological interactions such as ligand-signalling should be further investigated. Thirdly, it should be noted that the FBSDE model relies on the structure of the implemented neural network, and further exploration of different network architectures is warranted. Alternative forms of neural networks, such as the neural ODE proposed by Chen et al. [34] and recurrent neural networks, may have the potential to improve the accuracy of generative modelling and should be considered for future investigation.

Our FBSDE method introduces a novel aspect of modelling single cell developmental trajectories. There are a few improvements that can be made in future directions. First, instead of a single SDE, the FBSDE model can incorporate a mixture of stochastic differential equations analogous to Gaussian mixture models in probability density estimation. The introduction of the mixture model has the potential to increase model accuracy in situations such as the *Arabidopsis thaliana* stem cells dataset. Second, the FBSDE method can be generalized to systemic biology by incorporating spatial data into the genetic data. The introduction of spatial information allows for the modelling of cell-to-cell interactions that require physical proximity.

Author Contributions

Conceptualization: Kevin Zhang, Dehan Kong, Zhaolei Zhang.

Data curation: Kevin Zhang.

Formal analysis: Kevin Zhang, Junhao Zhu.

Funding acquisition: Dehan Kong, Zhaolei Zhang.

Investigation: Kevin Zhang, Junhao Zhu.

Methodology: Junhao Zhu.

Supervision: Dehan Kong.

Visualization: Kevin Zhang.

Writing – original draft: Kevin Zhang.

Writing – review & editing: Kevin Zhang, Dehan Kong, Zhaolei Zhang.

References

1. Wu AR, Neff NF, Kalisky T, Dalerba P, Treutlein B, Rothenberg ME, et al. Quantitative assessment of single-cell RNA-sequencing methods. *Nat Methods*. 2014; 11(1):41–46. https://doi.org/10.1038/nmeth.2694 PMID: 24141493
2. Kolodziejczyk AA, Kim JK, Svensson V, Marioni JC, Teichmann SA. The technology and biology of single-cell RNA sequencing. *Mol Cell*. 2015; 58(4):610–620. https://doi.org/10.1016/j.molcel.2015.04.005 PMID: 26000846
3. Song EAC, Min S, Oyelakin A, Smalley K, Bard JE, Liao L, et al. Genetic and scRNA-seq analysis reveals distinct cell populations that contribute to salivary gland development and maintenance. *Sci Rep*. 2018; 8(1):14043. https://doi.org/10.1038/s41598-018-32343-z PMID: 30232460
4. Spanjaard B, Hu B, Mitic N, Olivares-Chauvet P, Janjuha S, Ninov N, et al. Simultaneous lineage tracing and cell-type identification using CRISPR–Cas9-induced genetic scars. *Nat Biotechnol*. 2018; 36 (5):469–473. https://doi.org/10.1038/nbt.4124 PMID: 29644996
5. Treutlein B, Brownfield DG, Wu AR, Neff NF, Mantalas GL, Espinoza FH, et al. Reconstructing lineage hierarchies of the distal lung epithelium using single-cell RNA-seq. *Nature*. 2014; 509(7500):371–375. https://doi.org/10.1038/nature13173 PMID: 24739965
6. Muto Y, Humphreys BD. Recent advances in lineage tracing for the kidney. *Kidney Int*. 2021; 100 (6):1179–1184. https://doi.org/10.1016/j.kint.2021.05.040 PMID: 34217781
7. Treutlein B, Lee QY, Camp JG, Mall M, Koh W, Shariati SAM, et al. Dissecting direct reprogramming from fibroblast to neuron using single-cell RNA-seq. *Nature*. 2016; 534(7607):391–395. https://doi.org/10.1038/nature18323 PMID: 27281220
8. Ocasio JK, Babcock B, Malawsky D, Weir SJ, Loo L, Simon JM, et al. scRNA-seq in medulloblastoma shows cellular heterogeneity and lineage expansion support resistance to SHH inhibitor therapy. *Nat Commun*. 2019; 10(1):5829. https://doi.org/10.1038/s41467-019-13657-6 PMID: 31863004
9. Lin C, Bar-Joseph Z. Continuous-state HMMs for modeling time-series single-cell RNA-Seq data. *Bioinform*. 2019; 35(22):4707–4715. https://doi.org/10.1093/bioinformatics/btz296 PMID: 31038684
10. Tran TN, Bader GD. Tempora: cell trajectory inference using time-series single-cell RNA sequencing data. *PLoS Comput Biol*. 2020; 16(9):e1008205. https://doi.org/10.1371/journal.pcbi.1008205 PMID: 32903255
11. Schiebinger G, Shu J, Tabaka M, Cleary B, Subramanian V, Solomon A, et al. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*. 2019; 176(4):928–943. https://doi.org/10.1016/j.cell.2019.01.006 PMID: 30712874
12. Zhang S, Afanassiev A, Greenstreet L, Matsumoto T, Schiebinger G. Optimal transport analysis reveals trajectories in steady-state systems. *PLoS Comput Biol*. 2021; 17(12):e1009466. https://doi.org/10.1371/journal.pcbi.1009466 PMID: 34860824

13. Huguet G, Magruder DS, Tong A, Fasina O, Kuchroo M, Wolf G, et al. Manifold interpolating optimal-transport flows for trajectory inference. *NIPS*. 2022; 35:29705–29718. PMID: 37397786

14. Aubin-Frankowski PC, Vert JP. Gene regulation inference from single-cell RNA-seq data with linear differential equations and velocity inference. *Bioinform*. 2020; 36(18):4774–4780. https://doi.org/10.1093/bioinformatics/btaa576 PMID: 33026066

15. Chen Y, Georgiou TT, Pavon M. Optimal transport over a linear dynamical system. *IEEE Trans Autom Control*. 2016; 62(5):2137–2152. https://doi.org/10.1109/TAC.2016.2602103

16. Ruthotto L, Osher SJ, Li W, Nurbekyan L, Fung SW. A machine learning framework for solving high-dimensional mean field game and mean field control problems. *PNAS*. 2020; 117(17):9183–9193. https://doi.org/10.1073/pnas.1922204117 PMID: 32273389

17. Carmona R, Laurière M. Convergence analysis of machine learning algorithms for the numerical solution of mean field control and games: II—the finite horizon case. *Ann Appl Probab*. 2022; 32(6):4065–4105. https://doi.org/10.1214/21-AAP1715

18. Tong A, Huang J, Wolf G, Van Dijk D, Krishnaswamy S. Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics. In: *ICML*. PMLR; 2020. p. 9526–9536.

19. Weinreb C, Wolock S, Tusi BK, Socolovsky M, Klein AM. Fundamental limits on dynamic inference from single-cell snapshots. *PNAS*. 2018; 115(10):E2467–E2476. https://doi.org/10.1073/pnas.1714723115 PMID: 29463712

20. Lavenant H, Zhang S, Kim YH, Schiebinger G. Towards a mathematical theory of trajectory inference. *arXiv preprint arXiv:210209204*. 2021;.

21. Vargas F, Thodoroff P, Lamacraft A, Lawrence N. Solving Schrödinger bridges via maximum likelihood. *Entropy*. 2021; 23(9):1134. https://doi.org/10.3390/e23091134 PMID: 34573759

22. Bunne C, Hsieh YP, Cuturi M, Krause A. Recovering stochastic dynamics via gaussian schr\"odinger bridges. *arXiv preprint arXiv:220205722*. 2022;.

23. Moon KR, van Dijk D, Wang Z, Gigante S, Burkhardt DB, Chen WS, et al. Visualizing structure and transitions in high-dimensional biological data. *Nat Biotechnol*. 2019; 37(12):1482–1492. https://doi.org/10.1038/s41587-019-0336-3 PMID: 31796933

24. Jordan R, Kinderlehrer D, Otto F. The variational formulation of the Fokker–Planck equation. *SIAM J Math Anal*. 1998; 29(1):1–17. https://doi.org/10.1137/S0036141096303359

25. Hashimoto T, Gifford D, Jaakkola T. Learning population-level diffusions with generative RNNs. In: *ICML*. PMLR; 2016. p. 2417–2426.

26. De Bortoli V, Thornton J, Heng J, Doucet A. Diffusion Schrödinger bridge with applications to score-based generative modeling. *NIPS*. 2021; 34:17695–17709.

27. Nelson E. Dynamical theories of Brownian motion. vol. 101. New Jersey: Princeton university press; 2020.

28. Song Y, Sohl-Dickstein J, Kingma DP, Kumar A, Ermon S, Poole B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:201113456*. 2020;.

29. Weinreb C, Wolock S, Klein AM. SPRING: a kinetic interface for visualizing high dimensional single-cell expression data. *Bioinform*. 2018; 34(7):1246–1248. https://doi.org/10.1093/bioinformatics/btx792

30. Shahan R, Hsu CW, Nolan TM, Cole BJ, Taylor IW, Greenstreet L, et al. A single-cell Arabidopsis root atlas reveals developmental trajectories in wild-type and cell identity mutants. *Dev Cell*. 2022; 57(4):543–560. https://doi.org/10.1016/j.devcel.2022.01.008 PMID: 35134336

31. Chen Z, King WC, Hwang A, Gerstein M, Zhang J. DeepVelo: Single-cell transcriptomic deep velocity field learning with neural ordinary differential equations. *Sci Adv*. 2022; 8(48):eabq3745. https://doi.org/10.1126/sciadv.abq3745 PMID: 36449617

32. Liu R, Pisco AO, Braun E, Linnarsson S, Zou J. Dynamical systems model of RNA velocity improves inference of single-cell trajectory, pseudo-time and gene regulation. *JMB*. 2022; 434(15):167606. https://doi.org/10.1016/j.jmb.2022.167606 PMID: 35489382

33. Qiu X, Zhang Y, Martin-Rufino JD, Weng C, Hosseinzadeh S, Yang D, et al. Mapping transcriptomic vector fields of single cells. Cell. 2022; 185(4):690–711. https://doi.org/10.1016/j.cell.2021.12.045 PMID: 35108499

34. Chen RT, Rubanova Y, Bettencourt J, Duvenaud DK. Neural ordinary differential equations. *NIPS*. 2018; 31.