Variational Mixtures of ODEs for Inferring Cellular Gene Expression Dynamics

Yichen Gu [1] **David Blaauw** [*1] **Joshua Welch** [*1 2]

Abstract

A key problem in computational biology is discovering the gene expression changes that regulate cell fate transitions, in which one cell type turns into another. However, each individual cell cannot be tracked longitudinally, and cells at the same point in real time may be at different stages of the transition process. This can be viewed as a problem of learning the behavior of a dynamical system from observations whose times are unknown. Additionally, a single progenitor cell type often bifurcates into multiple child cell types, further complicating the problem of modeling the dynamics. To address this problem, we developed an approach called variational mixtures of ordinary differential equations. By using a simple family of ODEs informed by the biochemistry of gene expression to constrain the likelihood of a deep generative model, we can simultaneously infer the latent time and latent state of each cell and predict its future gene expression state. The model can be interpreted as a mixture of ODEs whose parameters vary continuously across a latent space of cell states. Our approach dramatically improves data fit, latent time inference, and future cell state estimation of single-cell gene expression data compared to previous approaches.

1. Introduction

The human body contains many cell types with distinct forms and functions, which arise from progenitor cells in a stepwise developmental process. A key question in molecular biology is what regulates this process of cellular development. In general, the diversity of cell types arises not from cell-to-cell differences in the DNA sequence itself, but in which portions of the DNA sequence (genes) are used (expressed) in each cell. The central dogma of molecular biology states that genes are first transcribed into messenger RNAs (mRNAs) and these mRNAs are then translated into proteins, which carry out biochemical functions. The expression level of a gene in a cell can thus be quantified by the number of mRNA molecules present in the cell. Therefore, understanding cellular development requires modeling how mRNA expression changes over time. Such models are crucial for numerous areas of biology and medicine, such as neuroscience, cancer research, and regenerative stem-cell therapies.

We are interested in the following problem that arises in the context of modeling cellular gene expression changes. Each sample (cell), indexed by i, is represented by a vector $X_i(t) \in \mathbb{R}^d$ parametrized by time t. The trajectory $X_i(t)$ is governed by some differential equation plus random noise. However, for each i, only the vector $x_i := X_i(t_i)$ is observed at some unknown time t_i. Our goal is two-fold: recover the latent time t_i for each sample and predict future states, i.e., $X_i(t)$ for $t > t_i$.

This unusual observation model stems from the limitations of single-cell RNA sequencing (scRNA-seq), the predominant experimental technology for measuring gene expression. The scRNA-seq technology (Tang et al., 2009) counts the number of mRNA sequences expressed within a set of individual cells, ultimately yielding a matrix of expression levels for $20,000$ genes across $10^4 - 10^6$ cells. However, measurement destroys the cell, so scRNA-seq gives only one single static snapshot of each cell at some moment in time. Second, the process of cell development is asynchronous– each cell takes a different amount of time to develop, so at a given moment, cells in a population will be at different developmental stages. An additional challenge is that a single starting cell type often bifurcates into multiple distinct cell types, so that cell-type-specific dynamics emerge over time.

Our key insight is that knowledge about the biochemical steps required for gene expression can serve as a regularization or constraint for this otherwise ill-posed problem. By partially specifying the form of a differential equation describing the data generation process, we can simultaneously recover the unknown times and predict future states

[*]Equal contribution [1]Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, United States [2]Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, United States. Correspondence to: David Blaauw <blaauw@umich.edu>, Joshua Welch <welchjd@umich.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

of the system. Our model can be interpreted as a variational autoencoder that reconstructs the data with an ODE whose parameters vary continuously across a latent space of cell states. Thus, we refer to our approach as a variational mixture of ODEs.

To our knowledge, this problem of learning a dynamical system from observations with unknown times has not been well studied. Previous papers have used both deep generative models and dynamical systems. But no previous work has demonstrated that these two approaches can be combined to solve the two-fold problem described earlier. Thus, this is a great example of an interesting problem arising from a computational biology application. Our work also adds another item to the growing list of neural network models that achieve a new state of the art on a problem of high scientific interest.

The novel aspects of this work include:

1. We simultaneously estimate the times and dynamics of observations with unknown time labels.
2. By incorporating mechanistic insights about the biochemical process of gene expression, our model learns latent variables with clear biological meanings.
3. Our approach dramatically improves the accuracy of time estimation and future state prediction compared to state-of-the-art approaches used by computational biologists, and thus has significant implications for biomedical research.

2. Related Work

Pseudotime Inference Methods. Various methods have been applied to scRNA-seq data to uncover cellular development paths. Pseudotime inference methods use distance from a manually-specified starting cell to rank cells according to degree of development. Diffusion pseudotime (Haghverdi et al., 2016) models cell development as a Markov process with a transition matrix. Other works (Qiu et al., 2017; Schiebinger et al., 2019) directly aim at determining the trajectory, i.e. putting the cells on one or multiple developmental paths.

RNA Velocity. La Manno et al. (2018) developed the concept of RNA velocity based on the observation that both unspliced and spliced mRNA molecules appear in sequencing outputs. The relative ratio of spliced and unspliced counts can indicate whether the gene was being turned on or turned off at the time the cell was sequenced. They introduced an ODE model to describe the gene expression process and used a steady state assumption to estimate parameters. Later work (Bergen et al., 2020) relaxed the steady-state assumption, allowing all cells to be used in parameter estimation. These RNA velocity methods have been widely used by biologists to help understand cellular development processes (Plass et al., 2018; Wilk et al., 2020; Litviňuková et al., 2020) and are currently the state of the art in this area.

Deep Generative Models for scRNA-seq Data. Previous papers have applied deep generative models to study scRNA-seq data. Many works (Wang & Gu, 2018; Lopez et al., 2018; Grønbech et al., 2020) have shown that variational autoencoders can learn useful latent representations for identifying cell types. In addition, Lotfollahi et al. (2019) showed that arithmetic operations of latent representation learned from scRNA-seq data can generate meaningful data corresponding to gene perturbation. BasisDeVAE (Danks & Yau, 2021) used a VAE to simultaneously infer similarity-based pseudotime and cluster genes by their pseudotime trends. VeloAE (Qiao & Huang, 2021) embedded RNA velocity estimates from the steady-state model and spliced gene expression in the same latent space.

Learning a Dynamical System. The problem of learning dynamical systems from high-dimensional datasets has been studied in many science and engineering domains. Early works (Calderhead et al., 2009; Dondelinger et al., 2013) applied gradient matching to estimate differential equation parameters. These methods involve MCMC sampling during the inference. Later works (Gorbach et al., 2017; Ghosh et al., 2021) improved the scalability and computational cost using variational inference. Another type of method called Neural ODEs (Chen et al., 2018; Yildiz et al., 2019; Huang et al., 2021) was proposed to model time series. It assumes a dynamical system described by an ODE in the latent space.

Key Limitations of Previous Work. Each of these four classes of approaches has key limitations. Cell trajectory inference is based purely on pairwise similarity and cannot infer the directions or rates of cell development. RNA velocity enables mechanistic modeling of cell development, allowing quantitative analysis of gene expression and cell fate prediction. However, current methods have many limiting assumptions and fail to yield accurate results in many cases, such as when transcription rates vary over time or multiple lineages arise from the same progenitor cell type (Bergen et al., 2021). Deep generative models for single-cell data can learn cellular representations, but they have not incorporated the mechanistic insights from the RNA velocity approaches. General methods for learning dynamical systems require time information, so they are not directly applicable to datasets without time labels. To address these limitations, we propose VeloVAE, a variational mixture of ODEs that jointly recovers cell times and gene expression dynamics.

3. Methods

Section 3.1 and 3.2 introduce the problem statement and background information about previous computational methods. Next, we describe a basic model that assumes fixed

cellular dynamics to infer latent time in 3.3. Finally, we describe our proposed method, a variational mixture of ODEs.

3.1. Problem Setup

The key biochemical insight underlying our approach is that to express a gene, two types of RNA, nascent unspliced and mature spliced RNA, are produced sequentially. First, unspliced RNAs are directly transcribed from DNA sequences and contain non-protein-encoding sequences (introns). Next, the introns are removed so that nascent molecules are converted into mature ones (Fig. 1). To put it another way, increases in the unspliced count (u) for a gene must precede increases in the spliced count (s). This simple insight makes it possible to recover the ordering of cells lacking time labels.

We assume that a dynamical system $F(t;\boldsymbol{\theta})$ generates scRNA count data. Here, $\boldsymbol{\theta}$ is a set of parameters describing the system, such as the transcription, splicing and degradation rates (introduced later; see Fig. 1). Our goal is to use observed scRNA data to simultaneously estimate the parameters $\boldsymbol{\theta}$ of F and infer the unknown cell times t.

Definition 3.1. Let u_g and s_g denote the unspliced and spliced mRNA count of the g-th gene. Let $\mathcal{G} = \{1, 2, \ldots, G\}$ be a set of genes measured in an scRNA-seq experiment. The **feature vector** of a cell is defined as $\mathbf{x} = [u_1, u_2, \ldots, u_G, s_1, s_2 \ldots, s_G]^T$.

Definition 3.2. The **kinetic equation** of gene g is defined as a system of ordinary differential equations relating changes in u and s over time. If there exists a solution $F(t;\boldsymbol{\theta})$ to the initial value problem with $u(0) = u_0, s(0) = s_0$, we call this solution the **kinetic function** for g.

Definition 3.3. Given a kinetic function $u(t)$ and $s(t)$ of a gene, the **RNA velocity** of the gene is defined as $\frac{ds}{dt}$.

3.2. Modeling Gene Expression Kinetics

In previous work (La Manno et al., 2018), the kinetic equation is modeled by a system of two linear ODEs:

$$\frac{du}{dt} = \alpha I_{\{t<t_{off}\}} - \beta u, \quad \frac{ds}{dt} = \beta u - \gamma s, \tag{1}$$

where $I_{\{\cdot\}}$ is an indicator function for the condition in brackets. The model parameters α, β and γ correspond to the RNA transcription, splicing and degradation rates, respectively. The model assumes that two discrete phases can occur in the gene expression process: (1) induction, when new unspliced RNA molecules are being transcribed and (2) repression, when the transcription process stops and no new unspliced molecules are made. The induction phase is assumed to start at $t_{on} = 0$ and the transition from induction to repression occurs at time t_{off}.

Parameter Estimation by Steady-State Assumption. If the induction phase lasts for a long time, u and s will asymptotically converge to a stable value, called the steady state. We denote the steady-state values u_{ss} and s_{ss}. The initial approach to estimating the parameters of the kinetic equation in the absence of cell times was to assume that the cells have reached steady state (La Manno et al., 2018). A simple calculation shows that the steady-state condition of the kinetic equation (1) is $u_{ss} = \frac{\alpha}{\beta}$ and $s_{ss} = \frac{\alpha}{\gamma}$. Suppose we have a set of measurements of u and s. We pick the top quantile, u^* and s^*, as the approximate steady-state values. If we further assume that $\beta = 1$, then the estimated parameters are $\hat{\alpha} = u^*$, $\hat{\beta} = 1$ (by assumption), and $\hat{\gamma} = \frac{u^*}{s^*}$.

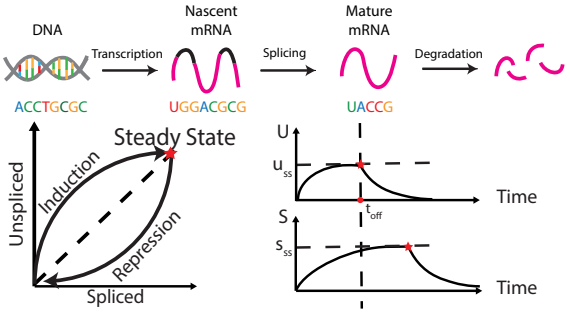

Figure 1: **Gene Expression Kinetics.** *Top*: A gene is transcribed into nascent RNA before being spliced into mature RNA and subsequently degraded. *Bottom*: temporal relationships between u and s implied by the model above.

Dynamical Model and EM Algorithm. In practice, real-world datasets contain many cells that are not at the steady state; in fact, for some genes, only transient states are observed. The steady-state estimation method does not utilize these transient cells. Thus, Bergen et al. (2020) developed a dynamical model called scVelo for estimating the parameters of the kinetic equation without the steady-state assumption. To deal with the absence of time, scVelo uses an expectation-maximization (EM) algorithm to jointly infer the latent times and model parameters. In this approach, they first solve the kinetic equations (1) analytically to obtain the kinetic function:

$$u(t) = u_0 \exp(-\beta\tau) + \frac{\widetilde{\alpha}}{\beta}(1 - \exp(-\beta\tau)) \tag{2}$$

$$s(t) = s_0 \exp(-\gamma\tau) + \frac{\widetilde{\alpha}}{\gamma}(1 - \exp(-\gamma\tau)) + \frac{\widetilde{\alpha} - \beta u_0}{\gamma - \beta}(\exp(-\gamma\tau) - \exp(-\beta\tau)) \tag{3}$$

$$\widetilde{\alpha} := \alpha I_{\{t<t_{off}\}}, \ \tau := tI_{\{t<t_{off}\}} + (t - t_{off})I_{\{t\geq t_{off}\}}$$

Note that the solution depends on the initial conditions $u(0) = u_0, s(0) = s_0$. ScVelo assumes that, given cell time t, u and s are Gaussian random variables whose means are given by the kinetic function (2),(3). Because the dynamical model makes use of the full ODE solution, it does

not require the steady-state assumption and produces better RNA velocity estimates.

Limitations of scVelo. However, scVelo has several significant limitations. First, scVelo infers time separately for each gene, which neglects crucial information about the covariance of related genes and often leads to times that are inconsistent across genes. This gene-specific notion of time also makes it hard to compare the switch-off time (time when a cell stops producing new RNA) across genes. The lack of a common time scale, combined with the assumption that induction starts at $t = 0$, also leads to frequent errors in estimating the overall direction of a gene (increasing or decreasing). Genes with a short or missing induction phase are particularly prone to being fit incorrectly by scVelo. Second, scVelo assumes a constant transcription rate α within the induction phase for each gene. In a recent review paper, the scVelo developers note that this assumption is often violated in real-world datasets, which leads to a variety of pathological behaviors (Bergen et al., 2021). Finally, scVelo's model does not account for cell type bifurcations, which frequently occur in cellular development (Bergen et al., 2021).

3.3. VeloVAE: Basic Model (Fixed Transcription Rate)

We first describe a deep generative model that recovers gene expression dynamics and cell time jointly assuming a single constant transcription rate for each gene. The model described in this section is thus a basic form of the variational mixture of ODE approach in section 3.4.

Generative Process. We assume that cell time t is first randomly sampled from a normal prior $\mathcal{N}(t_0, \sigma_0^2)$. If cell capture times are available (e.g., if cells were isolated separately on days 7 and 14), we can use them as an informative prior; otherwise, we can simply use a standard normal prior. We model the u and s counts for each gene using the kinetic function given by the analytical ODE solution, assuming that the genes are conditionally independent given cell times. Then, given N i.i.d. time samples $t_1, t_2, \ldots, t_N$, gene expression data $\mathbf{x}_i$, $i = 1, 2, \ldots, N$, are generated by $\mathbf{x_i} = F(t_i; \boldsymbol{\theta}) + \mathbf{r_i}, \mathbf{r_i} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma_r})$. Here, $F(t; \boldsymbol{\theta}) = [u_1(t), u_2(t), \ldots, u_G(t), s_1(t), s_2(t), \ldots, s_G(t)]^T$ is a vector containing all kinetic functions evaluated at t and $\mathbf{r}$ is Gaussian random noise. Equivalently, this means that the distribution of $\mathbf{x}$ conditioned on time is $p(\mathbf{x}|t, \boldsymbol{\theta}) \sim \mathcal{N}(F(t; \boldsymbol{\theta}), \boldsymbol{\Sigma_r})$. We further assume that the noise variables $\mathbf{r}$ are mutually independent, i.e. $\boldsymbol{\Sigma_r}$ is diagonal with nonzero entries $\sigma_{u,1}^2, \sigma_{u,2}^2, \ldots, \sigma_{u,G}^2, \sigma_{s,1}^2, \sigma_{s,2}^2, \ldots, \sigma_{s,G}^2$.

ODE Formulation. We use a similar ODE model for $F(t; \boldsymbol{\theta})$ as in previous work (La Manno et al., 2018; Bergen et al., 2020), with a slight modification. Instead of assuming all genes start generating mRNA at $t = 0$, we allow asynchronous generation by adding a gene-specific parameter, t_{on}. This small change, in addition to inferring latent time jointly across genes, should alleviate scVelo's difficulty in fitting genes with an absent or short induction phase. The kinetic function thus has the same form as equations (2) and (3), except that the definition of τ changes:

$$\tau := (t - t_{on}) I_{\{t_{on} \le t < t_{off}\}} + (t - t_{off}) I_{\{t \ge t_{off}\}}$$

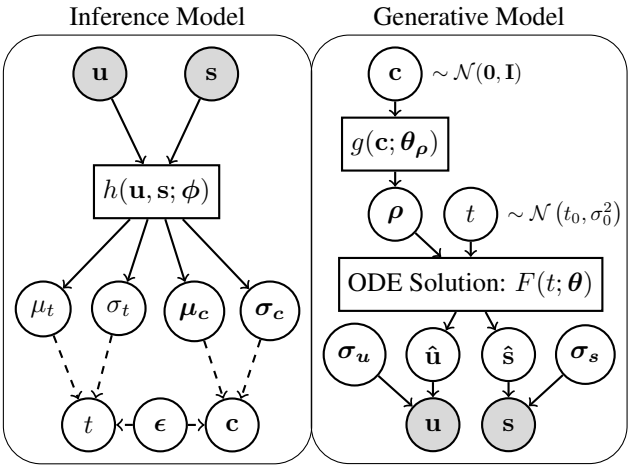

Figure 2: **Graphical Model.** Observed variables are colored gray. Dashed arrows indicate sampling.

Parameter Inference. Having formulated a generative model, our goal is to estimate both the ODE parameters $\boldsymbol{\theta}$ and the unknown cell times t_i. However, the posterior distribution of t_i is intractable. Furthermore, unlike the scVelo model in which each gene has its own separate estimate of time, EM becomes much more difficult once cell time is shared across genes (see Appendix A). Instead, we use variational inference to find t and $\boldsymbol{\theta}$. For our variational approximation, we use a Gaussian distribution whose parameters are output by a neural network. That is, $q(t|\mathbf{x}) \sim \mathcal{N}(h(\mathbf{u}, \mathbf{s}; \boldsymbol{\phi}))$ where $h(\cdot)$ is a neural network that outputs mean and variance. Following the argument by Kingma & Welling (2014), we apply the reparameterization trick to approximate the evidence lower bound (ELBO) via sampling:

$$\begin{aligned} ELBO &= \sum_{i=1}^{N} \mathbb{E}_{q(t|\mathbf{x_i})} \left[\log p(\mathbf{x_i}|t)\right] - KL(q(t|\mathbf{x_i})||p(t)) \\ &\approx \frac{1}{2} \sum_{i=1}^{N} \left[-2G \log(2\pi) - \log|\boldsymbol{\Sigma_r}| - d(\mathbf{x_i}, F(t_i; \boldsymbol{\theta}); \boldsymbol{\Sigma_r})^2\right] \\ &\quad + \frac{1}{2}\left[\log \frac{\sigma_p^2}{\sigma_q^2} + \frac{\sigma_q^2}{\sigma_p^2} + \frac{(\mu_p - \mu_q)^2}{\sigma_p^2} - 1\right] \end{aligned} \tag{4}$$

where $F(t_i; \boldsymbol{\theta})$ is the kinetic function and $d(\cdot, \cdot; \boldsymbol{\Sigma})$ denotes the Mahalanobis distance with $\boldsymbol{\Sigma}$ as the covariance matrix. We can then jointly estimate the neural network weights $\boldsymbol{\phi}$, the ODE parameters $\boldsymbol{\theta}$, and the cell times t_i by minimiz-

ing the negative ELBO using minibatch stochastic gradient descent.

Neural Network Architecture. The encoder is a multilayer perceptron (MLP) containing two hidden layers (500 and 250 neurons, respectively) with batch normalization (Ioffe & Szegedy, 2015) and dropout (Srivastava et al., 2014). The bottleneck layer outputs the mean and standard deviation parameters of the variational distribution.

3.4. VeloVAE: Variational Mixture of ODE Model

Although the model in the previous section can jointly infer cell times and ODE parameters, it still fails to capture important aspects of cellular development. In particular, constant transcription rates cannot account for bifurcations, which occur when a single type of stem cell develops into multiple descendant cell types. In fact, the possibility of bifurcations means that $u(t)$ and $s(t)$ may no longer be functions–multiple distinct cell states may be present at a given point in time. To capture these complex dynamics, we introduce a latent cell state variable $\mathbf{c}$ in addition to latent time and allow the transcription rate to vary smoothly over cell state space.

ODE Formulation. We adopt an ODE formulation similar to (1), except that the transcription rate for each gene is not a single constant α anymore. Instead, we assume that the kinetic equation is a continuous mixture of ODEs with transcription rate parameters $\widetilde{\alpha} = \rho\alpha$. The relative transcription rate $\rho \in [0, 1]$ is a function of latent cell state $\mathbf{c}$, and thus may be slightly different in each cell. The new kinetic equation is:

$$\frac{du}{dt} = \rho\alpha - \beta u, \quad \frac{ds}{dt} = \beta u - \gamma s \tag{5}$$

Note that there are no longer discrete induction and repression phases. This can be viewed as a generalization of (1), since $\rho = 1$ and $\rho = 0$ correspond to the discrete induction and repression phases, respectively, used in the simpler formulation. Because ρ is constant with respect to time, we can still solve the kinetic equation analytically to obtain a closed form for the kinetic function $F(t; \boldsymbol{\theta})$ in terms of α, β, γ, and ρ. The solution is the same as (2) and (3) except that $\widetilde{\alpha} = \rho\alpha$. Note also that for each gene, α, β, and γ are still shared across cells. This model can now capture continuous transcription changes such as those in a bifurcating developmental process.

Generative Process. The generative process for the variational mixture of ODE model is as follows:

$$t \sim \mathcal{N}(t_0, \sigma_0^2),\ \mathbf{c} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$
$$\tilde{\boldsymbol{\alpha}} = \boldsymbol{\rho} \odot \boldsymbol{\alpha},\ \boldsymbol{\rho} = g(\mathbf{c}; \boldsymbol{\theta}_{\boldsymbol{\rho}})$$
$$\mathbf{x} \sim \mathcal{N}(F(t; \boldsymbol{\theta}), \boldsymbol{\Sigma}_{\mathbf{r}})$$

Here, $g(\cdot)$ is a neural network with parameters $\boldsymbol{\theta}_{\boldsymbol{\rho}}$, $\odot$ is the elementwise product, F is the kinetic function of all genes, and $\boldsymbol{\Sigma}_{\mathbf{r}}$ is a diagonal covariance matrix. This generative process relies on a function g mapping latent cell states $\mathbf{c}$ to relative transcription rates $\boldsymbol{\rho}$. Intuitively, the cell states can model continuous and bifurcating developmental paths, allowing the entire set of cells to be described as a family of ODEs whose parameters vary smoothly over the cell state manifold. We assume that $\boldsymbol{\rho}$ varies smoothly across the cell state space and that each point in cell state space maps to a unique $\boldsymbol{\rho}$. Although g is deterministic for given $\mathbf{c}$, the inferred cell state for each cell $\mathbf{x}$ is probabilistic. Thus, the distribution of $\boldsymbol{\rho}$ can encompass multiple states near a bifurcation. Our generative model is summarized in figure 2.

Parameter Inference and Neural Network Architecture. The objective function is the ELBO shown in equation (4), with modified kinetic functions $F(t; \boldsymbol{\theta})$ and an updated KL divergence term incorporating the prior for $\mathbf{c}$. For h, we use the same MLP structure as the simple model with two additional outputs to produce the posterior mean and standard deviation of $\mathbf{c}$. We use an MLP that is the mirror image of h (two layers with 250 and 500 neurons, respectively) to learn the mapping g from $\mathbf{c}$ to $\boldsymbol{\rho}$. Source code is available online [1].

Initial Conditions. Because each cell now potentially has different ODE parameters, determining the initial conditions is more complex. Thus, instead of making the initial conditions trainable parameters, we simply train the model with $u_0 = s_0 = 0$ in all of our experiments. This still yields excellent data reconstruction and latent time inference (Table 1). However, the initial conditions are important for accurately predicting the future state of each cell. To improve the accuracy of future state prediction, we first train the VeloVAE to convergence using $u_0 = s_0 = 0$ so that latent times and cell states are accurate, then determine the initial conditions for a cell at time t by simply averaging the (u, s) values observed in an immediately preceding time interval $[t - \delta_1, t - \delta_2]$. We then fine-tune the ODE parameters using these updated initial conditions, keeping latent time and cell state fixed.

4. Experiments

4.1. Datasets

We evaluated our method on 6 different scRNA-seq datasets: pancreatic endocrinogenesis (PE) (Bastidas-Ponce et al., 2019), dentate gyrus (DG1,DG2) (Hochgerner et al., 2018; La Manno et al., 2018), embryonic E18 mouse brain cortex from 10X Genomics (MB1)[2], the erythroid lineage from mouse gastrulation (ET) (Pijuan-Sala et al., 2019),

[1] https://github.com/welch-lab/VeloVAE

[2] https://www.10xgenomics.com/resources/datasets/fresh-embryonic-e-18-mouse-brain-5-k-1-standard-1-0-0

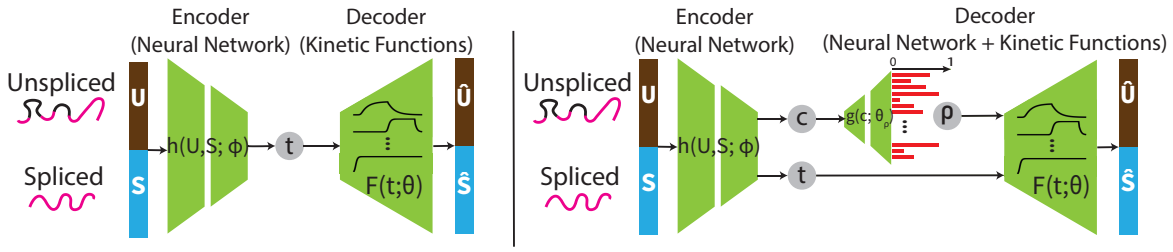

Figure 3: **VeloVAE Architecture. (a) Basic Model.** An encoder network infers a latent time from all u and s values for each cell. The data are reconstructed from inferred time using the kinetic function, whose analytical form is known. **(b) Full Model.** An encoder network infers both latent time and latent cell state $\mathbf{c}$. A decoder network generates the transcription rates $\boldsymbol{\rho}$, which are unique for each cell and each gene. The data are then reconstructed from t and $\boldsymbol{\rho}$ using the kinetic function.

and part of a whole mouse brain development dataset (MB2) (La Manno et al., 2021). See Appendix B for details. Each dataset contains two cell-by-gene count matrices–one for unspliced counts and one for spliced counts. The matrices are preprocessed as described in the scVelo paper.

4.2. Training

For all experiments, we performed minibatch stochastic gradient descent using the ADAM optimizer with learning rate 2×10^{-4} and batch size of 128. For each dataset, we trained on 70% of the data until the ELBO converged on the training set (number of epochs varied due to differences in dataset size), then evaluated the reconstruction error and likelihood on the held-out test set. We used 5 latent dimensions for cell state $\mathbf{c}$ in all experiments. For datasets with more than one capture time, we used the capture times to initialize the ODE parameters; otherwise, we used the steady-state approximation for initialization.

4.3. Results

We evaluated our method and compared it with scVelo, the state-of-the-art method for RNA velocity computation. To assess the importance of the mixture of ODEs, we also evaluated the basic model with fixed transcription rate. We used several metrics to compare the performance of the methods. First, we assessed how well the models fit the observed data. The limitations of the single-cell data itself preclude ground truth for the cell times. However, the inferred times should at least be correlated with the cell capture times when available (usually on the order of days). We also evaluated the results qualitatively using biological knowledge of the overall properties of cellular development in the systems we studied. Our results show that VeloVAE fits the data and estimates cell times far more accurately than scVelo, while recovering qualitative properties of cellular development that scVelo cannot model.

Table 1: Performance on scRNA-seq Datasets. We compare scVelo (SOTA), Basic Model (VAE with fixed rates), and VeloVAE (our proposed method). The metrics we use are (1) MSE = Mean Squared Error; (2) k_t = Time correlation; and (3) k_t(Info.) = Time correlation under informative prior

Dataset	Method	MSE	k_t	k_t(Info.)
PE	scVelo	2.107	N/A	N/A
	Basic Model	6.815	N/A	N/A
	VeloVAE	**0.823**	N/A	N/A
DG1	scVelo	0.670	N/A	N/A
	Basic Model	0.574	N/A	N/A
	VeloVAE	**0.243**	N/A	N/A
MB1	scVelo	10.160	N/A	N/A
	Basic Model	10.431	N/A	N/A
	VeloVAE	**1.886**	N/A	N/A
ET	scVelo	0.873	-0.707	N/A
	Basic Model	0.246	**0.802**	0.802
	VeloVAE	**0.151**	0.622	**0.855**
DG2	scVelo	1.385	-0.158	N/A
	Basic Model	0.968	0.304	0.306
	VeloVAE	**0.159**	**0.529**	**0.707**
MB2	scVelo	18.19	-0.777	N/A
	Basic Model	2.295	0.621	0.629
	VeloVAE	**0.152**	**0.870**	**0.897**

4.3.1. Data Reconstruction

We used three metrics–mean squared error (MSE), mean absolute error (MAE), and log likelihood (LL)–to assess how well each method fits the data. For our two models, we calculated these metrics on both a training dataset (70%) and held-out test dataset (30%). Note that we are not able to calculate these metrics on a test set using scVelo, because it does not have a way to perform out-of-sample prediction. Training MSE results are shown in Table 1; other metrics can be found in Appendix C. The basic model generally achieves better MSE than scVelo, although the results are

worse on the PE and MB1 datasets. This may be because scVelo fits each gene separately, estimating $N \times G$ latent time parameters (one for each cell and each gene) rather than N latent time values estimated by the basic model. This allows scVelo to essentially overfit the data by separately adjusting the latent time values for each gene, but leads to severe inconsistency in cell time across genes and poor recovery of the overall cell times, as shown in Section 4.3.2. In contrast, the VeloVAE model consistently achieves the best reconstruction by a wide margin despite estimating only N latent times. This suggests that the variational mixture of ODEs is crucial for accurately fitting the data. Furthermore, the test set is reconstructed nearly as accurately as the training set, indicating that the VeloVAE generalizes well and is not simply overfitting the training data.

4.3.2. Time Inference

Evaluating the latent time inference is challenging, because ground truth times are not available due to experimental limitations. However, three of the datasets (ET, DG2, MB2) contain data collected in multiple experiments across several days. The time stamps of these experiments (capture times) have very coarse granularity, and cells captured at the same time will span a wide range of developmental stages. Nevertheless, the inferred cell times should at least be correlated with the capture times. Thus, we computed the Spearman correlation between the cell times inferred by each method and the capture times. Because VeloVAE can use the capture times as an informative prior for the cell times, we reported the correlation when using either a capture time prior or an uninformative prior in Table 1. Although scVelo infers latent time separately for each gene, the tool provides a post-hoc procedure for estimating a single global time for each cell. Using this global time for comparison with our methods casts scVelo in the best possible light because the global time is more robust than the gene-specific latent times. Table 1 indicates that VeloVAE and the basic model both significantly outperform scVelo at inferring latent time. In fact, the scVelo global time is anticorrelated with capture time in all three datasets. In contrast, VeloVAE achieves the best performance, inferring latent times that are strongly correlated with capture time even with an uninformative time prior. The informative prior further increases the correlation. Figure 4 (a)-(c) visualize the true capture time and inferred cell time on the UMAP coordinates.

Table 2: Correlation between scVelo's gene-specific and global time, averaged across all genes

Dataset	Correlation
PE	0.262
DG1	0.097
MB1	0.226
ET	0.103
DG2	-0.008
MB2	-0.272

The low time correlation from scVelo may be partly explained by inconsistency among the different notions of time fitted for each gene. To investigate this further, we computed the average time correlation between scVelo's gene-specific and global latent time. As Table 2 shows, the correlation is indeed quite low. Furthermore, it has been reported (Bergen et al., 2021) that genes whose kinetics violate some of the assumptions of scVelo's ODE model can lead to inferred time that proceeds in the wrong direction– consistent with what we observed here.

4.3.3. Qualitative Advantages of VeloVAE

VeloVAE Fits Early Repression and Late Induction Genes. The restrictive assumptions of scVelo's ODE model, in concert with the separate inference of time for each gene, lead to very poor fits for many genes. In particular, scVelo suffers from systematic errors in genes that are turned off at the beginning of the process (early repression) or do not turn on until late in the process (late induction). For example, Fig. 5 shows the predicted values for *Smoc1* (early repression gene) and *Gng12* (late induction gene) in the PE dataset. In this dataset, the endocrine progenitor cells (Ngn3 low EP and Ductal) develop into four terminal cell types, alpha, beta, delta and epsilon. To fit *Smoc1*, scVelo rearranges the cell times to force an induction phase, creating a biologically incorrect ordering where progenitor cells appear in the middle of time and incorrectly predicting an increase in gene expression at the beginning of time. Similarly, when fitting *Gng12*, scVelo rearranges cell times to force all of the cells into the induction phase, leading to the incorrect prediction that *Gng12* expression is constantly increasing. In contrast, VeloVAE fits the correct trends.

VeloVAE Detects Transcriptional Boosts. A recent review (Bergen et al., 2021) showed that current RNA velocity approaches cannot account for "transcriptional boosts". These occur when the transcription rate rapidly increases over time, making $u(t)$ and $s(t)$ concave upward. This confounds the assumptions of the simple ODE model, leading to a time estimate that is backward. However, as shown in Fig. 4d, VeloVAE is able to accurately model such genes because the ρ parameter varies by cell.

VeloVAE Models Cell Type Bifurcations. In most scRNA datasets (including 5 we analyzed here), a single progenitor type produces multiple cell types. A single ODE with a constant transcription rate cannot model time-varying kinetics, including bifurcation. Thus, neither scVelo nor our basic model can accurately model cell type bifurcations. However, VeloVAE flexibly models the emergence of cell-type-specific kinetics. For example, VeloVAE models the three-way branching expression pattern of *Gng12*, which diverges as alpha, beta, and delta cells are formed (Fig. 5).

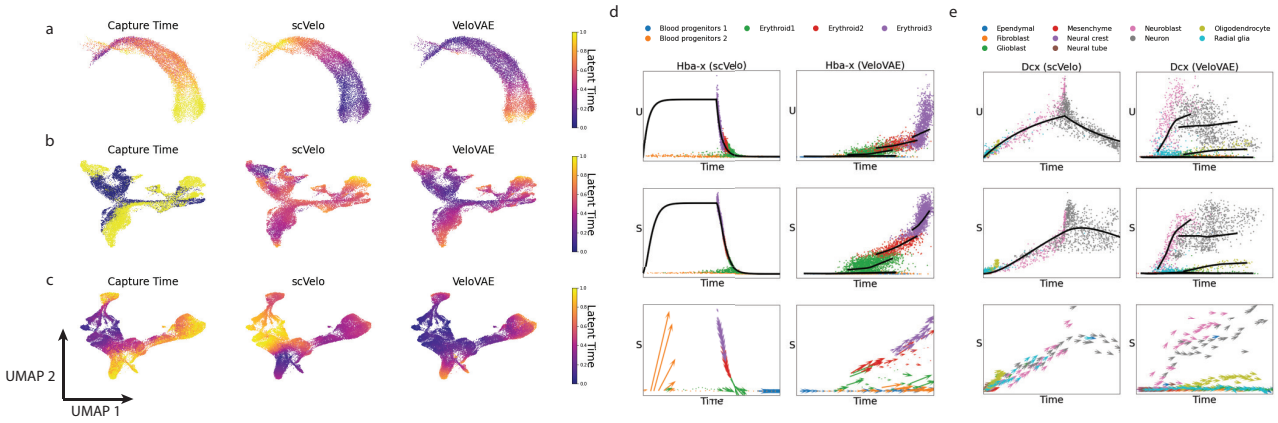

Figure 4: **Comparison of Inferred Time and Fit (a)-(c)** UMAP plots of scRNA data colored by capture time (left column), scVelo global time (middle), and VeloVAE time (right) for ET **(a)**, DG2 **(b)**, and MB2 **(c)** datasets. **(d)-(e)** Fitted (lines) and real (points) values from scVelo and VeloVAE for *Hba-x* gene in ET dataset and *Dcx* gene in MB2 dataset. Colors indicate cell types. Note that the VeloVAE fits are actually a point cloud, not a line (see inset plots in Fig. 5); the fit is so accurate that it would hide the real points, so we summarize the fit with a separate LOESS smooth per cell type to avoid overplotting. VeloVAE correctly models transcriptional boosts **(d)** and bifurcating gene expression trends **(e)**. Arrows in the bottom row of plots indicate predicted future cell states from RNA velocity estimates. Cells are randomly subsampled for clarity.

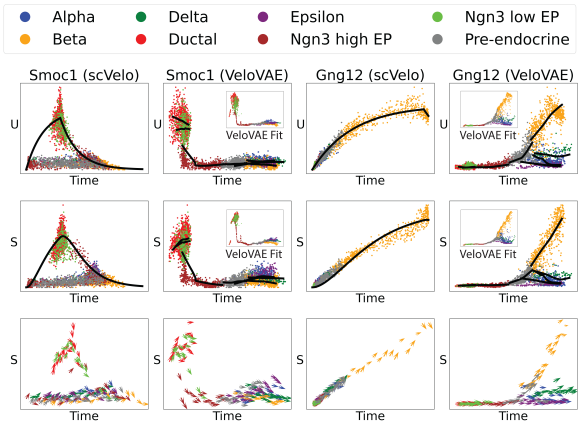

Figure 5: **VeloVAE Correctly Models Early Repression and Late Induction.** Fitted (lines) and real (points) values from scVelo and VeloVAE for the *Smoc1* (early repression) and *Gng12* (late induction, branching dynamics) genes in the PE dataset. Colors indicate cell types. Note that the VeloVAE fits are actually a point cloud, not a line; the fit is so accurate that it would hide the real points, so we summarize the fit with a separate LOESS smooth per cell type to avoid overplotting. The inset plots show the complete point clouds predicted by VeloVAE.

Similarly, VeloVAE models branching *Dcx* expression in neurons and oligodendrocytes (Fig. 4e). In contrast, scVelo rearranges gene-specific latent time to force the cells onto a single trajectory, erasing the cell-type-specific kinetics. A 2D visualization of RNA velocity for all genes in the PE dataset (Fig. 6) confirms that VeloVAE better predicts branches leading to alpha, beta, delta, and epsilon cells.

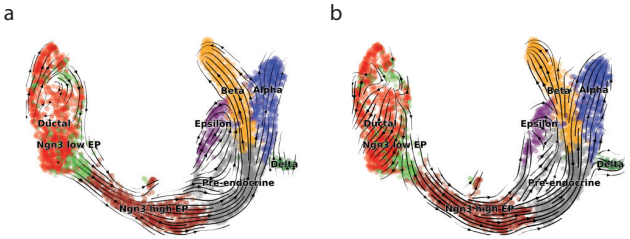

Figure 6: **Visualization of RNA Velocity from PE Dataset.** 2D projection of RNA velocity vectors predicted by scVelo **(a)** and VeloVAE **(b)**. VeloVAE more accurately predicts the branching dynamics to terminal cell types (alpha, beta, delta, and epsilon).

Cell states are meaningful representations of cell differentiation. In section 3, we described the cell state as a continuous representation of cell types. We validate this claim by showing a 3D scatter plot of cell state versus time (Fig. 7). For all six datasets, the cell state changes continuously over time and extends to multiple branches at cell type bifurcation points.

In addition, we note that the cell state models multiple states at bifurcation. We do not infer a single decision point in which a discrete cell fate decision occurs. Instead, we model the emergence of cell types as a smooth transition in which the cell state assignment has low uncertainty in undifferentiated progenitors, high uncertainty when cell fate decision is occurring, and low uncertainty again after the

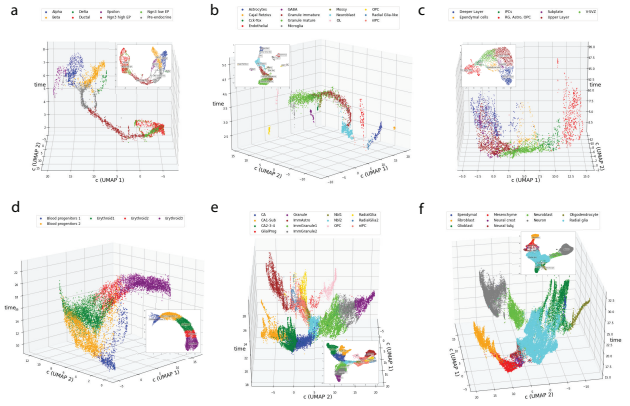

Figure 7: **Cell State Evolution over Time.** The vertical axis is the latent time and the horizontal plane contains 2D UMAP coordinates of **c**. **(a)** Pancreas **(b)** Dentate Gyrus **(c)** 10x Mouse Brain **(d)** Erythroid **(e)** Dentate Gyrus 2 **(f)** Whole Mouse Brain. Insets show UMAPs from original expression data.

fate decision. To measure the uncertainty, we picked uni- and multi-variate coeffcient of variation (CV) (Van Valen, 1974) as our metric for uncertainty. For example, the CV of **c** is the highest for ductal cells deciding between cell cycle progression and exit to the Ngn3 progenitor state, as well as for Ngn3 progenitors deciding among the alpha, beta, delta, and epsilon fates (Figure 8).

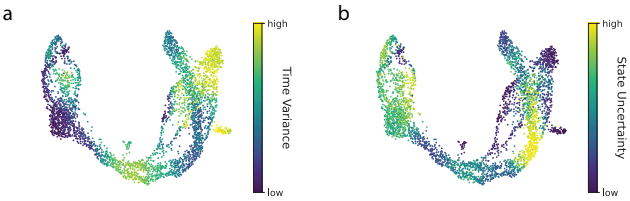

Figure 8: **Cell Time and State Uncertainty of the Pancreas Dataset.** We used CV as a measure of cell time and state uncertainty. The values are log-transformed for better visualization.

Scalability. We think scalability is a key benefit of our approach. Minibatch optimization enables memory usage independent of cell number, whereas scVelo needs the entire dataset in memory. Number of iterations required by VeloVAE should also increase sublinearly with number of cells. As a rough benchmark, we trained our model for 600 epochs with an NVIDIA Tesla V100 GPU and ran scVelo on a single core of a 2.4 GHz Intel Xeon Gold 6148 CPU. We have not yet optimized our implementation for runtime or memory efficiency, and 600 epochs is likely overkill for large datasets. But we are already at least as fast as scVelo and 600 epochs on the whole mouse brain dataset took about 5 hours (Figure 9).

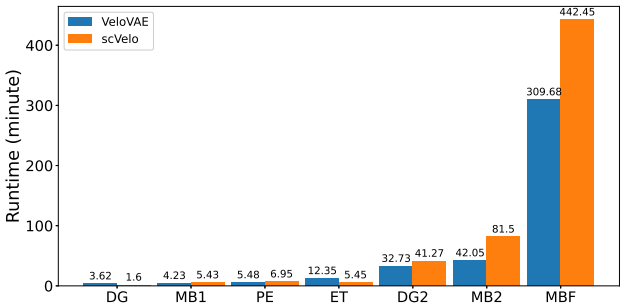

Figure 9: **Run-Time Comparison.**

5. Discussion

In this work, we developed VeloVAE, a deep generative model for inferring cellular gene expression dynamics. We demonstrated that VeloVAE can infer meaningful cell times while also fitting a model of gene expression dynamics. VeloVAE achieved much better performance than the state-of-the-art method, scVelo, on multiple scRNA-seq datasets.

Our principled probabilistic framework provides a strong foundation for future extensions. The current model assumes that genes are conditionally independent given both time and cell state. Relaxing this conditional independence assumption to infer groups of co-regulated genes is an exciting future direction. Another possible direction is modeling u and s as integer counts rather than normalized continuous variables.

Our approach can be interpreted in several intuitive ways. From one perspective, we constrain the joint distribution of $u(t)$ and $s(t)$ to reflect our prior knowledge of the data generating process. From another point of view, our approach is a variational autoencoder modified so that the latent variables learned by the encoder have clear biological meanings (cell time and cell state) by construction. Another interpretation is that knowing any two of three quantities–time, observations, and underlying dynamics–enables inference of the third. Many previous papers have shown how to infer dynamics when time and observations are known; we show that having observations and general knowledge about how they are generated allows recovery of unknown times.

6. Acknowledgements

This work was funded by NIH grant R01HG010883 to JDW. We would like to thank Reetuparna Das for help with GPU resources, and Chen Li for helpful discussions.

References

Bastidas-Ponce, A., Tritschler, S., Dony, L., Scheibner, K., Tarquis-Medina, M., Salinno, C., Schirge, S., Burtscher, I., Böttcher, A., Theis, F. J., Lickert, H., Bakhti, M., Klein, A., and Treutlein, B. Comprehensive single cell mRNA profiling reveals a detailed roadmap for pancreatic endocrinogenesis. *Development*, 146(12), 06 2019. ISSN 0950-1991. doi: 10.1242/dev.173849. URL https://doi.org/10.1242/dev.173849. dev173849.

Bergen, V., Lange, M., Peidli, S., Wolf, F. A., and Theis, F. J. Generalizing rna velocity to transient cell states through dynamical modeling. *Nature biotechnology*, 38 (12):1408–1414, 2020. ISSN 1546-1696. doi: 10.1038/s41587-020-0591-3. URL https://doi.org/10.1038/s41587-020-0591-3.

Bergen, V., Soldatov, R. A., Kharchenko, P. V., and Theis, F. J. Rna velocity—current challenges and future perspectives. *Molecular Systems Biology*, 17(8):e10282, 2021. doi: https://doi.org/10.15252/msb.202110282. URL https://www.embopress.org/doi/abs/10.15252/msb.202110282.

Calderhead, B., Girolami, M., and Lawrence, N. Accelerating bayesian inference over nonlinear differential equations with gaussian processes. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper/2008/file/07563a3fe3bbe7e3ba84431ad9d055af-Paper.pdf.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf.

Danks, D. and Yau, C. Basisdevae: Interpretable simultaneous dimensionality reduction and feature-level clustering with derivative-based variational autoencoders. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2410–2420. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/danks21a.html.

Dondelinger, F., Husmeier, D., Rogers, S., and Filippone, M. Ode parameter inference using adaptive gradient matching with gaussian processes. In Carvalho, C. M. and Ravikumar, P. (eds.), *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, volume 31 of *Proceedings of Machine Learning Research*, pp. 216–228, Scottsdale, Arizona, USA, 29 Apr–01 May 2013. PMLR. URL https://proceedings.mlr.press/v31/dondelinger13a.html.

Ghosh, S., Birrell, P., and De Angelis, D. Variational inference for nonlinear ordinary differential equations. In Banerjee, A. and Fukumizu, K. (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2719–2727. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/ghosh21b.html.

Gorbach, N. S., Bauer, S., and Buhmann, J. M. Scalable variational inference for dynamical systems. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/e71e5cd119bbc5797164fb0cd7fd94a4-Paper.pdf.

Grønbech, C. H., Vording, M. F., Timshel, P. N., Sønderby, C. K., Pers, T. H., and Winther, O. scVAE: variational auto-encoders for single-cell gene expression data. *Bioinformatics*, 36(16):4415–4422, 05 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa293. URL https://doi.org/10.1093/bioinformatics/btaa293.

Haghverdi, L., Büttner, M., Wolf, F. A., Buettner, F., and Theis, F. J. Diffusion pseudotime robustly reconstructs lineage branching. *Nature methods*, 13(10):845–848, 2016. ISSN 1548-7105. doi: 10.1038/nmeth.3971. URL https://doi.org/10.1038/nmeth.3971.

Hochgerner, H., Zeisel, A., Lönnerberg, P., and Linnarsson, S. Conserved properties of dentate gyrus neurogenesis across postnatal development revealed by single-cell rna sequencing. *Nature Neuroscience*, 21 (2):290–299, 2018. ISSN 1546-1726. doi: 10.1038/s41593-017-0056-2. URL https://doi.org/10.1038/s41593-017-0056-2.

Huang, H., Liu, H., Wang, H., Xiao, C., and Wang, Y. Strode: Stochastic boundary ordinary differential equation. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4435–4445. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/huang21d.html.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/ioffe15.html.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y. (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

La Manno, G., Soldatov, R., Zeisel, A., Braun, E., Hochgerner, H., Petukhov, V., Lidschreiber, K., Kastriti, M. E., Lönnerberg, P., Furlan, A., et al. Rna velocity of single cells. *Nature*, 560(7719):494–498, 2018. ISSN 1476-4687. doi: 10.1038/s41586-018-0414-6. URL https://doi.org/10.1038/s41586-018-0414-6.

La Manno, G., Siletti, K., Furlan, A., Gyllborg, D., Vinsland, E., Mossi Albiach, A., Mattsson Langseth, C., Khven, I., Lederer, A. R., Dratva, L. M., Johnsson, A., Nilsson, M., Lönnerberg, P., and Linnarsson, S. Molecular architecture of the developing mouse brain. *Nature*, 596(7870):92–96, 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03775-x. URL https://doi.org/10.1038/s41593-017-0056-2.

Li, T., Shi, J., Wu, Y., and Zhou, P. On the mathematics of rna velocity i: Theoretical analysis. *bioRxiv*, 2020. doi: 10.1101/2020.09.19.304584. URL https://www.biorxiv.org/content/early/2020/09/20/2020.09.19.304584.

Litviňuková, M., Talavera-López, C., Maatz, H., Reichart, D., Worth, C. L., Lindberg, E. L., Kanda, M., Polanski, K., Heinig, M., Lee, M., et al. Cells of the adult human heart. *Nature*, 588(7838):466–472, 2020. ISSN 1476-4687. doi: 10.1038/s41586-020-2797-4. URL https://doi.org/10.1038/s41586-020-2797-4.

Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., and Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12):1053–1058, 2018. ISSN 1548-7105. doi: 10.1038/s41592-018-0229-2. URL https://doi.org/10.1038/s41592-018-0229-2.

Lotfollahi, M., Wolf, F. A., and Theis, F. J. scgen predicts single-cell perturbation responses. *Nature methods*, 16 (8):715–721, 2019. ISSN 1548-7105. doi: 10.1038/s41592-019-0494-8. URL https://doi.org/10.1038/s41592-019-0494-8.

Pijuan-Sala, B., Griffiths, J. A., Guibentif, C., Hiscock, T. W., Jawaid, W., Calero-Nieto, F. J., Mulas, C., Ibarra-Soria, X., Tyser, R. C. V., Ho, D. L. L., Reik, W., Srinivas, S., Simons, B. D., Nichols, J., Marioni, J. C., and Göttgens, B. A single-cell molecular map of mouse gastrulation and early organogenesis. *Nature*, 566(7745):490–495, 2019. ISSN 1476-4687. doi: 10.1038/s41586-019-0933-9. URL https://doi.org/10.1038/s41586-019-0933-9.

Plass, M., Solana, J., Wolf, F. A., Ayoub, S., Misios, A., Glažar, P., Obermayer, B., Theis, F. J., Kocks, C., and Rajewsky, N. Cell type atlas and lineage tree of a whole complex animal by single-cell transcriptomics. *Science*, 360(6391):eaaq1723, 2018. doi: 10.1126/science.aaq1723. URL https://www.science.org/doi/abs/10.1126/science.aaq1723.

Qiao, C. and Huang, Y. Representation learning of rna velocity reveals robust cell transitions. *Proceedings of the National Academy of Sciences*, 118 (49), 2021. ISSN 0027-8424. doi: 10.1073/pnas.2105859118. URL https://www.pnas.org/content/118/49/e2105859118.

Qiu, X., Mao, Q., Tang, Y., Wang, L., Chawla, R., Pliner, H. A., and Trapnell, C. Reversed graph embedding resolves complex single-cell trajectories. *Nature methods*, 14(10):979–982, 2017. ISSN 1548-7105. doi: 10.1038/nmeth.4402. URL https://doi.org/10.1038/nmeth.4402.

Schiebinger, G., Shu, J., Tabaka, M., Cleary, B., Subramanian, V., Solomon, A., Gould, J., Liu, S., Lin, S., Berube, P., Lee, L., Chen, J., Brumbaugh, J., Rigollet, P., Hochedlinger, K., Jaenisch, R., Regev, A., and Lander, E. S. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943.e22, 2019. ISSN 0092-8674. doi: https://doi.org/10.1016/j.cell.2019.01.006. URL https://www.sciencedirect.com/science/article/pii/S009286741930039X.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Tang, F., Barbacioru, C., Wang, Y., Nordman, E., Lee, C., Xu, N., Wang, X., Bodeau, J., Tuch, B. B., Siddiqui, A., et al. mrna-seq whole-transcriptome analysis of a single cell. *Nature methods*, 6(5):377–382, 2009. ISSN

1548-7105. doi: 10.1038/nmeth.1315. URL https://doi.org/10.1038/nmeth.1315.

Van Valen, L. Multivariate structural statistics in natural history. *Journal of Theoretical Biology*, 45(1):235–247, 1974. ISSN 0022-5193. doi: https://doi.org/10.1016/0022-5193(74)90053-8. URL https://www.sciencedirect.com/science/article/pii/0022519374900538.

Wang, D. and Gu, J. Vasc: Dimension reduction and visualization of single-cell rna-seq data by deep variational autoencoder. *Genomics, Proteomics Bioinformatics*, 16(5):320–331, 2018. ISSN 1672-0229. doi: https://doi.org/10.1016/j.gpb.2018.08.003. URL https://www.sciencedirect.com/science/article/pii/S167202291830439X. Bioinformatics Commons (II).

Wilk, A. J., Rustagi, A., Zhao, N. Q., Roque, J., Martínez-Colón, G. J., McKechnie, J. L., Ivison, G. T., Ranganath, T., Vergara, R., Hollis, T., et al. A single-cell atlas of the peripheral immune response in patients with severe covid-19. *Nature medicine*, 26(7):1070–1076, 2020. ISSN 1546-170X. doi: 10.1038/s41591-020-0944-y. URL https://doi.org/10.1038/s41591-020-0944-y.

Yildiz, C., Heinonen, M., and Lahdesmaki, H. Ode2vae: Deep generative second order odes with bayesian neural networks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/99a401435dcb65c4008d3ad22c8cdad0-Paper.pdf.

A. Expectation-Maximization is Intractable When Cell Time is Shared Across Genes

Gene-Shared Latent Times. It is shown in scVelo (Bergen et al., 2020) that given the u and s values of a single gene, cell time can be inferred using an EM algorithm. However, their approach results in different cell times for different genes. Instead, we would like to develop an EM algorithm to infer the unique cell time, which is shared across all genes. We denote time as $\mathbf{t}$ and the ODE parameters as $\boldsymbol{\theta}$. Following the standard EM algorithm, we obtain the E-step at the $(j+1)$-th iteration:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta};\boldsymbol{\theta}^{(j)}) &= \mathbb{E}_{p(\mathbf{t}|\mathbf{X};\boldsymbol{\theta}^{(j)})}\left[\ln p(\mathbf{X}|\mathbf{t};\boldsymbol{\theta})\right] \\ &= \sum_{i=1}^{N} \mathbb{E}_{p\left(t^{(i)}|\mathbf{x}^{(i)};\boldsymbol{\theta}^{(j)}\right)}\left[\ln p(\mathbf{x}^{(i)}|t^{(i)};\boldsymbol{\theta})\right]\end{aligned}$$

Here, we make the assumption that t_i and $\mathbf{x_i}$ $(i = 1, 2, \ldots, N)$ are mutually independent. First, without computing the exact form, we can show that the posterior is intractable.

$$p(t|\mathbf{x};\boldsymbol{\theta}) = \frac{p(\mathbf{x}|t;\boldsymbol{\theta})p(t)}{\int_{t'=-\infty}^{+\infty} p(\mathbf{x}|t';\boldsymbol{\theta})p(t')dt'}$$

It's natural to assume that the time prior $p(t)$ is uniform in $[0, T]$. However, VeloVAE assumes a Gaussian prior $\mathcal{N}(\mu_0, \sigma_0^2)$ because the support of a uniform distribution is not $\mathbb{R}$ and the KL divergence might be undefined in some cases. For the purpose of analysis, we choose the uniform prior here. Later in the case of unshared latent time, we will see that with certain approximations, using a uniform prior results in the same algorithm as scVelo.

In addition, we assume the covariance matrix of u and s of all genes is diagonal, i.e. $\boldsymbol{\Sigma} = \text{diag}(\sigma_{u,1}, \ldots, \sigma_{u,G}, \sigma_{s,1}, \ldots, \sigma_{s,G})$. Since $p(\mathbf{x}|t';\boldsymbol{\theta})$ is Gaussian, we have

$$p(t|\mathbf{x};\boldsymbol{\theta}) = \frac{\frac{1}{(2\pi)^G|\boldsymbol{\Sigma}|^{\frac{1}{2}}}e^{-d(t)^2}\cdot\frac{1}{T}I_{\{t\in[0,T]\}}}{\int_{t'=-\infty}^{+\infty}\frac{1}{(2\pi)^G|\boldsymbol{\Sigma}|^{\frac{1}{2}}}e^{-d(t')^2}\cdot\frac{1}{T}I_{\{t'\in[0,T]\}}dt'} = \frac{e^{-d(t)^2}I_{\{t\in[0,T]\}}}{\int_{t'=0}^{T}e^{-d(t')^2}dt'} \tag{6}$$

$$\text{where } d(t)^2 := \sum_{g=1}^{G}\frac{1}{2\sigma_{u,g}^2}(u_g - \hat{u}_g(t))^2 + \frac{1}{2\sigma_{s,g}^2}(s_g - \hat{s}_g(t))^2 \tag{7}$$

Now consider the integral in the denominator. We know that both $\hat{u}(t)$ and $\hat{s}(t)$ have at least one exponential term involving t. Without exact calculation, we know that the denominator will involve the integral of $\exp(\exp(-ct))$ with some constant c and this cannot be expressed by any elementary function. Let $C(\boldsymbol{\theta})$ be the constant equal to the integral in the denominator. Therefore, the total likelihood function is

$$\mathcal{L}(\boldsymbol{\theta};\boldsymbol{\theta}^{(j)}) = \sum_{i=1}^{N}\int_t \frac{e^{-d\left(t^{(i)};\boldsymbol{\theta}^{(j)}\right)^2}}{C^{(i)}(\boldsymbol{\theta}^{(j)})}\left[-G\ln(2\pi) - \frac{1}{2}\ln(|\boldsymbol{\Sigma}|) - d(t^{(i)};\boldsymbol{\theta})^2\right]$$

Because $C^{(i)}(\boldsymbol{\theta}^{(j)})$ is intractable, it's hard to directly optimize the total likelihood function. Hence, the EM algorithm cannot be easily applied.

In addition, if we pick a Gaussian prior, we would end up with the same result with slight difference in the form of $d(t)$:

$$d(t)^2 := \frac{(t-t_0)^2}{2\sigma_0^2} + \sum_{g=1}^{G}\frac{1}{2\sigma_{u,g}^2}(u_g - \hat{u}_g(t))^2 + \frac{1}{2\sigma_{s,g}^2}(s_g - \hat{s}_g(t))^2$$

Unshared Latent Times. Now let's consider the special case of $G = 1$, which is just the local gene fitting in scVelo. The M-step in scVelo (Bergen et al., 2020) is simply to minimize the sample mean square error:

$$\boldsymbol{\theta}^{(j+1)} = \arg\min_{\boldsymbol{\theta}}\frac{1}{N}\sum_{i=1}^{N}\left[\left(u^{(i)} - \hat{u}^{(i)}(t)\right)^2 + \left(s^{(i)} - \hat{s}^{(i)}(t)\right)^2\right] \tag{8}$$

where $\hat{u}, \hat{s}$ are predictions by the learned kinetic function of the gene. First, we need to assume $C^{(i)}(\boldsymbol{\theta}^{(j)})$ is a constant for

all i. With this approximation, the M-step becomes

$$\begin{aligned}
&\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}; \boldsymbol{\theta}^{(j)}) \\
=&\max_{\boldsymbol{\theta}} \sum_{i=1}^{N} \int_{t^{(i)}=0}^{T} \frac{e^{-d\left(t^{(i)};\boldsymbol{\theta}^{(j)}\right)^2}}{C^{(i)}(\boldsymbol{\theta}^{(j)})} \left[-c_\sigma - d(t^{(i)};\boldsymbol{\theta})^2\right] dt^{(i)} \\
=&\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \int_{t^{(i)}=0}^{T} \frac{e^{-d(t^{(i)};\boldsymbol{\theta}^{(j)})^2}}{C^{(i)}(\boldsymbol{\theta}^{(j)})} d(t^{(i)};\boldsymbol{\theta})^2 dt^{(i)} \qquad (9) \\
\approx&\min_{\boldsymbol{\theta}} \frac{1}{C(\boldsymbol{\theta}^{(j)})} \sum_{i=1}^{N} \int_{t^{(i)}=0}^{T} d(t^{(i)};\boldsymbol{\theta})^2 e^{-d\left(t^{(i)};\boldsymbol{\theta}^{(j)}\right)^2} dt^{(i)} \qquad (10)
\end{aligned}$$

We further assume $\sigma_u = \sigma_s = \sigma$, so $d(t)^2 = \frac{1}{2\sigma^2}\left[(u-\hat{u}(t))^2 + (s-\hat{s}(t))^2\right]$. We assume that $\sigma \approx 0$ and $d(t;\boldsymbol{\theta})$ has a global minimum $t_0 \in [0,T]$. As σ approaches 0, $d(t;\boldsymbol{\theta}^{(j)})$ approaches infinity. Following the analysis by Li et al. (2020), we apply the Laplace's method to approximate the integral:

$$\begin{aligned}
\int_{t^{(i)}=0}^{T} d(t^{(i)};\boldsymbol{\theta})^2 e^{-d\left(t^{(i)};\boldsymbol{\theta}^{(j)}\right)^2} dt^{(i)} &\approx \sqrt{\frac{2\pi}{2d''(t_0;\boldsymbol{\theta}^{(j)})d(t_0;\boldsymbol{\theta}^{(j)}) + 2d'(t_0;\boldsymbol{\theta}^{(j)})^2}} e^{-d\left(t_0;\boldsymbol{\theta}^{(j)}\right)^2} d(t;\boldsymbol{\theta})^2 \qquad (11) \\
&\propto d(t;\boldsymbol{\theta})^2 \qquad (12)
\end{aligned}$$

Using (10) and (12), we obtain the final result:

$$\arg\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta};\boldsymbol{\theta}^{(j)}) = \arg\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{i=1}^{N} \left[\left(u^{(i)} - \hat{u}^{(i)}(t)\right)^2 + \left(s^{(i)} - \hat{s}^{(i)}(t)\right)^2\right]$$

Therefore, minimizing the mean square error is equivalent to maximizing the total likelihood function under all the assumptions and approximations we made above.

B. Test Datasets

Table 3: Dataset Description

Dataset Name	Cells	Genes	Cell Types	No. Time Points
Pancreatic Endocrinogenesis (PE)	3696	2000	8	1
Dentate Gyrus (DG1)	2930	800	14	1
10x Mouse Brain (MB1)	3365	1000	7	1
Erythroid (ET)	9815	1000	5	7
Dentate Gyrus (DG2)	18213	2000	14	2
Mouse Brain Development (MB2)*	29994	1000	10	20

*Subsampled to 30,000 cells

C. Other Test Results

Table 4: Performance on scRNA-seq Datasets. The metrics we compared, from left to right, are (1) Training and Testing Mean Squared Error; (2) Training and Testing Mean Absolute Error; and (3) Log Likelihood. Test metrics are not reported for scVelo because the method does not allow out-of-sample prediction.

Dataset	Method	MSE (Train, Test)	MAE (Train, Test)	LL (Train, Test)
PE	scVelo	2.107, N/A	0.423, N/A	-1702, N/A
	Basic Model	6.815, 5.163	0.356, 0.351	271.71, 274.68
	VeloVAE	**0.823**, **0.616**	**0.191**,**0.192**	**727.42**, **717.20**
DG1	scVelo	0.670, N/A	0.316, N/A	-2287, N/A
	Basic Model	0.574, 0.560	0.302, 0.304	41.43, 41.96
	VeloVAE	**0.243**, **0.253**	**0.190**, **0.194**	**237.63**, **234.57**
MB1	scVelo	10.160, N/A	0.947, N/A	-1779, N/A
	Basic Model	10.431, 10.254	0.916, 0.921	-456.07, -498.74
	VeloVAE	**1.886**, **1.942**	**0.392**, **0.398**	**440.61**, **440.65**
ET	scVelo	0.873,N/A	0.456, N/A	-809.3, N/A
	Basic Model	0.246,0.246	0.251, 0.151	42.83, 44.25
	VeloVAE	**0.151**,**0.161**	**0.194**, **0.196**	**67.36**, **66.88**
DG2	scVelo	1.385, N/A	0.366, N/A	-3513, N/A
	Basic Model	0.968, 0.970	0.294, 0.295	954.74, 950.64
	VeloVAE	**0.159**, **0.163**	**0.120**, **0.121**	**1797.30**, **1791.32**
MB2	scVelo	18.19, N/A	0.47, N/A	-7258, N/A
	Basic Model	2.295, 2.440	0.359, 0.364	-328.16, -338.89
	VeloVAE	**0.152**, **0.147**	**0.089**, **0.091**	**926.98**, **924.48**