

NEURALGENE: INFERRING GENE REGULATION AND CELL-FATE DYNAMICS FROM NEURAL ODES

Yutong Sha,^{1,†} Yuchi Qiu,^{2,†} & Qing Nie^{1,3,4,*}

¹Department of Mathematics, University of California, Irvine, California 92697, USA

²Department of Mathematics, Michigan State University, East Lansing, Michigan 48824, USA

³Department of Developmental and Cell Biology, University of California, Irvine, California 92697, USA

⁴The NSF-Simons Center for Multiscale Cell Fate Research, University of California, Irvine, California 92697, USA

*Address all correspondence to: Qing Nie, Department of Mathematics, University of California, Irvine, CA 92697, USA; Department of Developmental and Cell Biology, University of California, Irvine, CA 92697, USA; The NSF-Simons Center for Multiscale Cell Fate Research, University of California, Irvine, CA 92697, USA, E-mail: qnie@math.uci.edu

[†]These authors contributed equally to this work.

Original Manuscript Submitted: 12/29/2022; Final Draft Received: 7/6/2023

In biology, cell-fate decisions are controlled by complex gene regulation. Although gene expression data may be collected at multiple time points, it remains difficult to construct the continuous dynamics from the data. In this work, we developed a data-driven approach, NeuralGene, a model based on neural ordinary differential equations (ODEs), to reconstruct continuous dynamical systems governing gene regulation from temporal gene expression data. In addition, NeuralGene has the flexibility of incorporating partial prior biological information in the model to further improve its accuracy. For a given cell at a static time point, the NeuralGene model can impute its continuous gene expression dynamics and predict its cell fate. We applied NeuralGene to a simulation toggle-switch model to verify its utility in modeling and reconstructing temporal dynamics. In addition, NeuralGene was applied to experimental single-cell qPCR data to show its ability for gene expression imputation and cell-fate prediction.

KEY WORDS: data-driven modeling, cell-fate decisions, deep learning

1. INTRODUCTION

In biology, cells exhibit distinct phenotypic states and fates in multicellular organisms. The cell fate is associated with the temporal dynamics of gene regulation, where thousands of genes interact with each other in a complex network, such as gene regulatory networks (GRNs). Mathematical modeling has been widely applied to study the functions of networks, such as adaptivity to fluctuations and noise attenuation (Shen-Orr et al., 2002; Qiao et al., 2019; Nie et al., 2020).

While genes interact with each other within an intracellular environment, extracellular signaling, morphogen, provides stimulation to the GRN to induce distinct fates over space (Lander et al., 2005, 2007). By integrating well-studied biological processes, the forward-engineering modeling successfully reveals the spatiotemporal dynamics of the biological tissues and dissects the individual functions and coordination of those complex processes (Qiu et al., 2019, 2021; Zhang et al., 2012; Zhu et al., 2020).

With the recent advance of machine learning and deep learning techniques along with the availability of genomic data, data-driven modeling becomes possible in inferring equations governing the temporal dynamics. Physical-informed neural networks (PINNs) can embed physical laws into the partial differential equations (PDEs), recovering the dynamics of the systems even with limited data (Raissi et al., 2019). For example, PINNs have been applied to infer parameters and dynamics of biological systems (Yazdani et al., 2020). Sparse identification of nonlinear dynamics (SINDy) can build differential equations from a library of physical laws using the sparsity-promoting techniques and machine learning (Brunton et al., 2016; Lusch et al., 2018). SINDy has been applied to infer gene regulatory network from the dynamical data (Mangan et al., 2016). Both PINNs and SINDy require partial physical laws as inputs. Alternatively, the residual network (ResNet)-based models are used to derive the temporal dynamics from the data without physical laws (Qin et al., 2019, 2020; Chen et al., 2022). Neural ordinary differential equations (neural ODEs) provide more a general approach that learns physical laws and continuous temporal dynamics from the data (Chen et al., 2018; Zhuang et al., 2020). Neural ODEs may infer complex interaction (i.e., right-hand side of ODEs) from neural networks and have the flexibility of partial prior knowledge to improve the model accuracy. Neural ODEs have been successfully applied to infer mechanistic models from biological data (Roesch et al., 2021). For example, TrajectoryNet describes cell trajectories by combining a known convection term with unknown neural network-based velocity fields (Tong et al., 2020), and a recurrent neural network recovers the spatiotemporal dynamics and GRNs in a spatial gene regulation model (Shen et al., 2021).

In this work, we present a neural ODEs-based model, NeuralGene, in learning dynamics of gene regulation from temporal gene expression data. In Section 2, we introduce the algorithms of NeuralGene and its utility in inferring dynamics of gene regulation and integrating known biological information along with black-box neural networks. In Section 3, we test NeuralGene on a simulated toggle-switch model with bifurcations. We show that NeuralGene can accurately capture the temporal dynamics of gene expression and classify cell-fate decisions governed by the toggle-switch system. With additional inputs of biological information, accuracy of NeuralGene can be largely improved. In Section 4, we apply NeuralGene to a single-cell qPCR dataset with bifurcation governing by two fate-marker genes. NeuralGene successfully reconstructs continuous dynamical systems in controlling the cell-fate mapping with different initial conditions. Moreover, given the expression data for one cell at a static time point, NeuralGene can impute the gene expression dynamics and accurately predict the cell fate at the final time. Lastly, in Section 5 we conclude and discuss.

2. METHODS AND BACKGROUND

Cells can select multiple cell fates through temporal dynamics of gene regulation. The cell-fate decision of an individual cell can be modeled by a multi steady-state gene regulation model using an autonomous system of ODEs:

$$\frac{dx}{dt} = \mathbf{f}(\mathbf{x}), \quad \mathbf{x}(t = 0) = \mathbf{x}_0, \quad (1)$$

where $\mathbf{x}(t) = (x_1(t), x_2(t), \dots, x_n(t))$ represents temporal dynamics of the expression of n genes. $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ describes the regulations between these n genes, which is fully or partially unknown. The steady states of the system (i.e., cell fates) depend on the initial state $\mathbf{x}_0$. To decipher the regulation and cell fate, NeuralGene learns from K series of temporal gene expression data, $\{\mathbf{X}^k\}_{k=1}^K$, obtained from different initial stages:

$$\mathbf{X}^k = \{x^k(t = t_0), x^k(t = t_1), \dots, x^k(t = t_j), \dots, x^k(t = t_J)\}, \quad (2)$$

where $t_0 = 0$ and $\mathbf{x}^k(t = t_0) = \mathbf{x}_0^k$ are the initial conditions. $\{t_0, t_1, \dots, t_J\}$ is a set of sample time points for collecting time-series data, and the final time t_J is large enough to allow all genes to converge to steady states. $\mathbf{x}_j^k$ is the expression state at t_j with initial condition $\mathbf{x}_0^k$.

Taking $\{\mathbf{X}^k\}_{k=1}^K$ as inputs, NeuralGene utilizes neural ODEs (Chen et al., 2018; Zhuang et al., 2020) to infer the unknown regulation $\mathbf{f}$.

In NeuralGene, the right-hand side of Eq. (1) is approximated by a deep-learning-based black-box function, specifically deep neural network (DNN). With no prior knowledge available, the entire right-hand side $\mathbf{f}$ can be approximated by a DNN $\mathbf{N} : \mathbf{f}(\mathbf{x}) \approx \mathbf{R}(\mathbf{x}; \theta) = \mathbf{N}(\mathbf{x}; \theta)$, where θ is the set of hyperparameters for DNN. If there is available biological knowledge for the systems, such as degradation rates of genes, then DNN can be used to model partial unknown right-hand side: $\mathbf{f}(\mathbf{x}) \approx \mathbf{R}(\mathbf{x}; \theta) = \mathbf{f}_0(\mathbf{x}) + \mathbf{N}(\mathbf{x}; \theta)$, where $\mathbf{f}_0(\mathbf{x})$ is the well-known term. The architecture of DNN is a sequence of fully connected feedforward neural network with L hidden layers $\mathbf{N}(\mathbf{x}; \theta) : \mathbb{R}^n \rightarrow \mathbb{R}^n$:

$$\mathbf{N}(\mathbf{x}; \theta) = W_L \circ (\sigma_{L-1} \circ W_{L-1}) \circ (\sigma_{L-2} \circ W_{L-2}) \circ \dots \circ (\sigma_0 \circ W_0)(\mathbf{x}), \quad (3)$$

where $\circ$ is the composition operator. W_i is the linear layer, and $W_i(\mathbf{z}) = c_i \mathbf{z} + b_i$, which maps the i th layer to the $(i + 1)$ th layer. In the linear layer, c_i is the linear weight and b_i is the bias term. 0th and $(L + 1)$ th layers are input and output layers, respectively. σ_i is the nonlinear activation function, and we designed the network without activation function for the output layer. We used rectified linear unit (ReLU) function, which is a component-wise operation, $\sigma_i(x) = \max(0, x)$.

For a given initial stage $\mathbf{x}_0^k$, we can integrate the neural network to obtain approximate solution of Eq. (1), which is inspired by the neural ODEs (Chen et al., 2018):

$$\mathbf{y}^k(t) = \mathbf{x}_0^k + \int_0^t \mathbf{R}(\mathbf{y}^k(s); \theta) ds. \quad (4)$$

$\mathbf{y}^k(t)$ at time points $t_1, \dots, t_J$ can be numerically solved by using an arbitrary ODE solver:

$$\mathbf{y}_1^k, \mathbf{y}_2^k, \dots, \mathbf{y}_J^k. \quad (5)$$

Then, the mean-square error (MSE) was taken as the loss function to describe the discrepancy between the gene expression data and the estimated temporal data:

$$L(\theta) = \frac{1}{KJ} \sum_{k=1}^K \sum_{j=1}^J \|\mathbf{y}_j^k - \mathbf{x}_j^k\|^2, \quad (6)$$

where $\|\cdot\|$ is the L_2 norm. The loss function was minimized to update hyperparameters θ . The numerical ODE solver can also be regarded as a deep neural network without hyperparameters

being optimized during training. NeuralGene can be regarded as a deep neural network by combining the DNN and the numerical ODE solver. Indeed, hyperparameters θ are optimized via backpropagation for both DNN and ODE solver. The gradient of θ was obtained from auto-differentiation of the loss function, and Adam optimizer (Kingma et al., 2014) was employed to optimize θ . Although the adjoint method proposed in neural ODEs can reduce the memory used to store the gradient, it requires longer computational time (Rubanova et al., 2019). Due to the relatively small size of neural network in this work, one single GPU is always sufficient to store the gradient. Indeed, we used the traditional auto-differentiation to accelerate computations. The DOPRI5 method (Dormand and Prince, 1980), one type of Runge-Kutta model with adaptive step size, is used for the ODE solver in neural ODEs.

3. APPLICATIONS TO A SIMULATED BISTABLE TWO-GENE REGULATORY NETWORK MODEL

3.1 NeuralGene Identifies Cell-Fate Bifurcation

Genetic toggle switch is a canonical gene regulatory motif in biology to study the bistability of cell-fate decisions (i.e., bifurcation) (Gardner et al., 2000; Kramer et al., 2004). Here, we study a classic toggle switch widely found in biology (Qiu et al., 2021; Zhang et al., 2012) [Fig. 1(a)]: two interactive genes A and B along with an extracellular signal activating both genes. For example, morphogen is a prevalent signal that diffuses over space to form a gradient, and it induces heterogeneity of cell-fate decisions via a concentration-dependent manner in spatial pattern formation (Qiu et al., 2021; Zhang et al., 2012; Zhu et al., 2020; Lander, 2011). Meanwhile, these two genes mutually inhibit each other with an auto-activation for their own expression. The gene regulatory network is a bistable system depending on initial gene expression levels. The dynamics of two-gene interactions are modeled by a system of ODEs:

$$\begin{aligned}\frac{d[A]}{dt} &= m_A \frac{s + a_A[A]^2}{s + a_A[A]^2 + b_B[B]^2} - d_A[A], \\ \frac{d[B]}{dt} &= m_B \frac{s + a_B[B]^2}{s + a_B[B]^2 + b_A[A]^2} - d_B[B],\end{aligned}\tag{7}$$

where $[A](t)$ and $[B](t)$ are concentration of two genes at time t . s is the signal strength, a_A and a_B are strengths of auto-activation for the two genes, and b_A and b_B are strengths of mutual inhibition. The signal, mutual inhibition, and auto-activation are modeled by Hill functions. In addition, $d_A[A]$ and $d_B[B]$ are degradations for gene A and B , respectively. In this work, we used one set of parameters allowing two genes to have the same interactive strength: $m_A = m_B = 1.0$, $a_A = a_B = 1.0$, $b_A = b_B = 1.0$, $d_A = d_B = 0.4$, and $s = 0.8$.

Due to the mutual inhibition, only one gene can maintain a high expression level whereas the other gene is suppressed at a low level [Figs. 1(c) and 1(d)]. Indeed, the bistability can be observed with different initial expression levels, where two distinct cell states, fate A and fate B , are named by the highly expressed gene. By taking the identical parameters in the equations of two genes, the boundary of the phase diagram is equally split by the initial expression level of two genes [Fig. 1(b)].

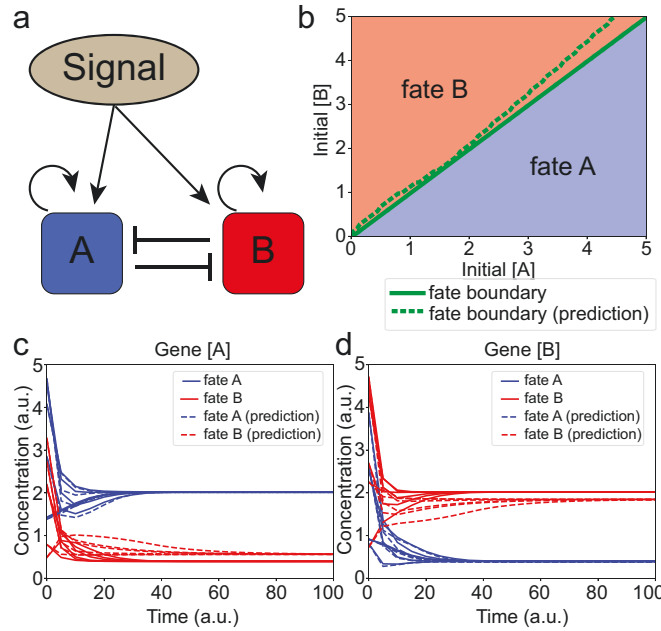


FIG. 1: NeuralGene identifying cell-fate bifurcation in a toggle switch model. (a) Illustration of gene regulatory network for the toggle switch. Gene *A* and gene *B* mutually inhibit each other, and they have auto-activation for their own expression. An extracellular signal activates both genes. (b) Phase diagram of cell-fate bifurcation under different initial expression levels. Cell fate is determined by expressions of *A* and *B* at the final time. The fate named by a gene indicates this gene has higher expression than the other at the final time. The solid green line is the ground-truth boundary for distinct cell fates. The predicted bifurcation boundary from NeuralGene is given by a dashed green line. (c) and (d) Temporal dynamics of gene expression for gene *A* (c) and gene *B* (d). Solid lines represent ground truth and dashed lines represent predictions from NeuralGene. Fate *A* and fate *B* are represented by blue and red, respectively.

3.2 NeuralGene Learns and Predicts Temporal Dynamics of Gene Expression and Cell-Fate Bifurcation

Here, we applied NeuralGene to the genetic toggle-switch model in learning and predicting its temporal dynamics of gene expression and cell fates. The simulations assume the well-established degradation while the neural network approximates the Hill functions in Eq. (7). The range of initial value of each gene is restricted in $[0, 5]$. For training set, its initial gene expression is generated using Latin hypercube sampling (Tang, 1993) from the rectangular space $[0, 5]^2$ for gene *A* and *B*. For testing set, we uniformly divided $[0, 5]$ with mesh size 0.05 to obtain 101 sample points for each gene, and the testing set contains $101^2 = 10,201$ sets of initial gene expression values. The temporal data of training and testing sets were generated by DOPRI5 ODEs solver at 20 time points uniformly distributed in $[0, 100]$.

The model accurately captures the temporal dynamics of gene expression and the cell-fate decisions [Figs. 1(c) and 1(d)]. The gene expression only evolves quickly for a short time and then slows down to hit the steady states, and the predictions successfully capture such dynamics. With only 400 sets of training data, NeuralGene accurately classifies the bifurcation of the system [Fig. 1(b)]. Specifically, in the $[0, 5]^2$ initial gene expression region, NeuralGene achieves a high accuracy 0.95, and a high F1 score 0.95 in classifying cell fates.

3.3 Inclusion of Partial Biological Information Improves NeuralGene's Accuracy

An essential feature of NeuralGene is that it can couple with well-known biological terms in the model. To explore the inclusion of known biological knowledge, we compared three models with different amounts of inputting biological information. One uses DNN to approximate the gene regulation terms for both genes, and the other two models include more information about gene regulation, where DNN only approximates a regulation term for one gene by assuming the other is well-known.

All three models accurately identify cell-fate decisions for different initial gene expression levels [Fig. 2(a)]. The inclusion of one gene regulation term significantly improves the classification accuracy, where the predicted boundary separating two fates is closer to the ground-truth boundary [Fig. 2(a)]. For the case masking both regulation terms, more simulations show poor

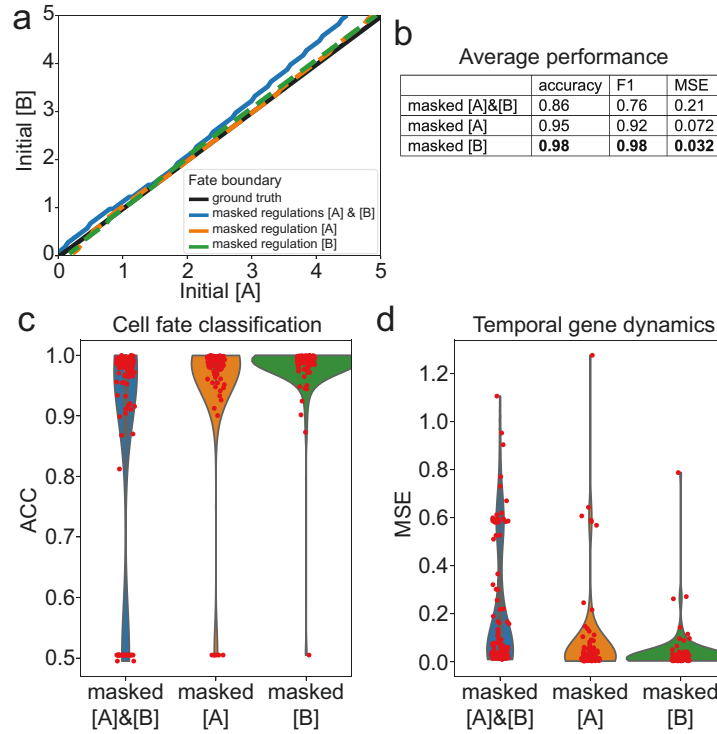


FIG. 2: Impacts of additional prior knowledge for NeuralGene. NeuralGene models are performed on simulated toggle-switch model. Three models are tested with different amount of prior knowledge, where the degradation terms for both genes are explicitly given in the right-hand sides of the equations for all cases. Masked [A]&[B] indicates regulation terms for both genes are inferred by NeuralGene. Masked [A] indicates only regulation term for gene A is inferred by NeuralGene. Masked [B] indicates only regulation term for gene B is inferred by NeuralGene. (a) Phase diagram of cell-fate bifurcation with cell-fate boundaries obtained from ground truth and predictions from three types of NeuralGene models. (b) Average accuracy of three NeuralGene models' performance in classifying cell fate and predicting temporal gene dynamics. $n = 100$ independent repeats are used to obtain the statistics. (c) and (d) Violin plots and scatter plots show distributions of metrics obtained by three NeuralGene models from $n = 100$ repeats for (c) accuracy (ACC) of cell-fate classifications and (d) mean-squared errors (MSEs) of accuracy in predicting temporal gene dynamics.

accuracy around 0.5 (i.e., equivalent accuracy using random sampling), and a relatively low average accuracy, 0.86, is achieved [Figs. 2(b) and 2(c)]. By explicitly including one regulation term, fewer simulations show the poor accuracy, and the average accuracy has at least 10.4% improvement over the case masking both regulation terms [Fig. 2(b)]. Other than the cell-fate classification, we also investigated the accuracy in predicting temporal dynamics of gene expression, which is measured by MSE. The case masking two interaction terms shows relatively large MSE with 0.21 while significant reduction of MSE is observed, at least 65.7% reduction, for the case masking interaction of A or B [Figs. 2(b) and 2(d)].

Overall, with the increasingly available biological information, NeuralGene can improve the model accuracy for both fate classification and gene expression predictions.

4. APPLICATION TO SINGLE-CELL QPCR DATA

Next, we applied NeuralGene to an experimental single-cell qPCR dataset (Bargaje et al., 2017). By profiling 96 developmental genes at single-cell resolution, the experiment observed a cell-fate bifurcation event where induced pluripotent stem cells (iPSCs) differentiated to two distinct lineages, mesodermal and endodermal lineages, in cardiomyocyte [Fig. 3(a)]. From 2.5 days, HAND1 and SOX17 expression start and they interact as a toggle switch to determine cell-fate bifurcation. In this study we selected data collected later than 2.5 days to study the bifurcation event for the interactions between HAND1 and SOX17. HAND1 and SOX17 provide markers

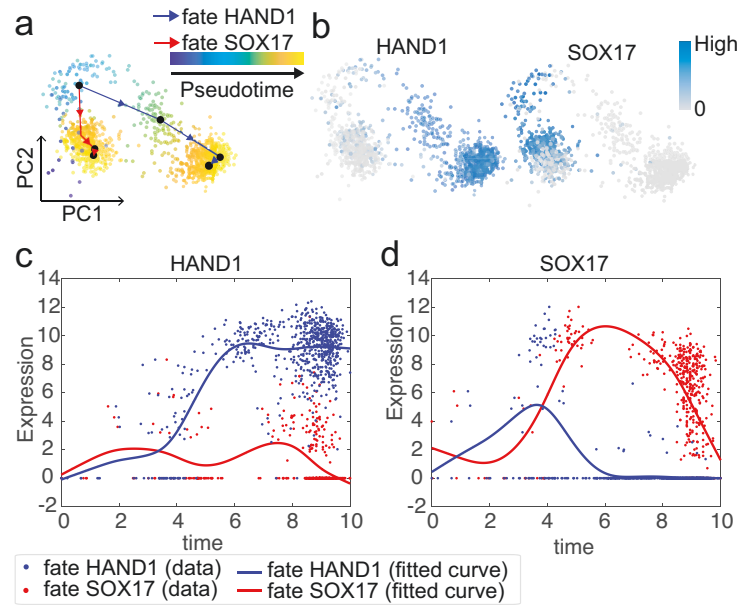


FIG. 3: Data preprocessing and training data generation for single-cell qPCR dataset. (a) The dataset is visualized at top two principal components (PC1 and PC2). (b) HAND1 and SOX17 provide fate marker for mesodermal (i.e., fate HAND1) and endodermal (i.e., fate SOX17) fates, respectively. (c) and (d) Cells are ordered in pseudotime (i.e., abscissa) plotted against expression of HAND1 and SOX17. Each cell is classified into one cell fate from pseudotime calculation denoted by blue (fate HAND1) and red (fate SOX17), respectively. Solid lines show the fitted curves from smoothing spline interpolation for gene expression data against pseudotime axis for both fates.

for mesodermal and endodermal lineages, respectively, where each cell fate only has one highly expressed gene [Fig. 3(b)] (Bargaje et al., 2017).

Although the dataset consists of data from multiple physical time points, the physical time may not present the real differentiation due to the stochasticity in gene expression and experimental errors. We reconstructed the time for the data, utilizing Slingshot (Street et al., 2018), a pseudotime reconstruction method, by assigning and ordering cells along two distinct lineages [Fig. 3(a)]. We selected day 2.5, day 3, day 4, and day 5 cells and processed them together. Cells are assigned to one of two lineages with an index, pseudotime, indicating its relative developmental time in the entire dataset. Specifically, we first preprocessed the dataset using full quantile normalization to scale each cell to the same distribution of expression values. Then, principal components analysis (PCA) was used to project the data to a lower dimensional space for preparation of Slingshot, where top three PCs were used. Last, Slingshot integrating with the default clustering method, Gaussian mixture modeling (Scrucca et al., 2016), reconstructed pseudotime. The iPSCs cluster was used as the root of lineages, and the lineages with two bifurcation fates were recovered correspondingly. After obtaining the pseudotime, the smoothing spline interpolation was used to fit the expression curve for HAND1 and SOX17 for two distinct trajectories [Figs. 3(c) and 3(d)]. The two smoothed trajectories were taken as the training data for NeuralGene, where we picked data at 21 uniformly distributed pseudotime points.

To learn the dynamics of gene expression, the autonomous system can be used as we did before in Eq. (1). DNN was used to approximate the entire right-hand side due to no available prior knowledge: $\mathbf{f}(\mathbf{x}) \approx \mathbf{R}(\mathbf{x}; \theta) = \mathbf{N}(\mathbf{x}; \theta)$. However, the interactions between HAND1 and SOX17 may depend on other genes. Alternatively, their interactions can also be written as a non-autonomous system where the right-hand side depends on the variable t implicitly indicating the temporal dynamics of other genes:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t), \quad \mathbf{x}(t = 0) = \mathbf{x}_0. \quad (8)$$

DNN for the non-autonomous system takes gene-state variable $\mathbf{x}$ and an additional time variable t as input: $\mathbf{f}(\mathbf{x}, t) \approx \mathbf{R}((\mathbf{x}, t); \theta) = \mathbf{N}((\mathbf{x}, t); \theta)$. We performed both autonomous (Fig. 4) and non-autonomous (Fig. 5) systems to compare their performance in predicting dynamics of gene expression and cell fate.

In the autonomous system, NeuralGene successfully learns the temporal dynamics of gene expression with training error $\text{MSE} = 1.40$ [Figs. 4(a) and 4(b)], which fails to capture the two waves in HAND1 expression for SOX17 fate. The model was then used to reconstruct gene expression maps at final time (i.e., $t = 10$) with respect to initial expression levels [Figs. 4(c) and 4(d)]. The bifurcation regions can be distinguished from gene expression maps by the location with sharp expression changes. However, the boundary between two regions (i.e., blue and red) has a complicated non-linear geometry. Since the training data only scatter at two points in the map, the complicated non-linear geometry may indicate an overfitting issue.

In the non-autonomous system, we performed identical calculations as the autonomous system (Fig. 5). NeuralGene has better training accuracy with $\text{MSE} = 0.75$ [Figs. 5(a) and 5(b)]. The two waves in HAND1 expression for SOX17 fate can also be captured from the model. The gene expression maps also identify the bifurcation regions with distinct expression levels [Figs. 5(c) and 5(d)]. Especially, a sharp boundary with a simple linear geometry can be observed for HAND1 expression [Fig. 5(c)]. Since HAND1 and SOX17 interact with each other in a toggle-switch manner, the boundary for distinguishing bifurcation regions is supposed to be sharp with respect to gene expression (Zhang et al., 2012) [also see Figs. 1(c) and 1(d)]. Our

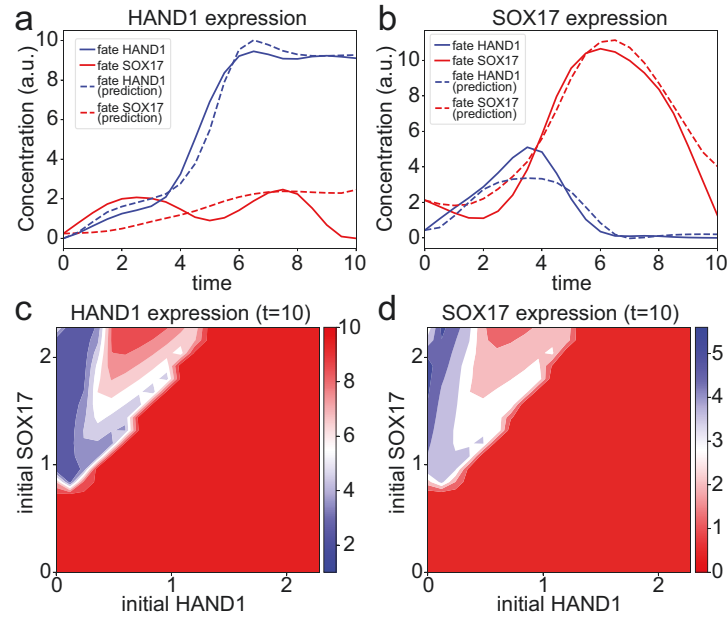


FIG. 4: HAND1-SOX17 fate predictions from autonomous system for single-cell qPCR data. (a) and (b) Temporal dynamics of expression for (a) HAND1 and (b) SOX17 where solid lines show training data and dashed lines show predictions for training data given the identical initial conditions. (c) and (d) Expression map for (c) HAND1 and (d) SOX17 given by different initial expression levels.

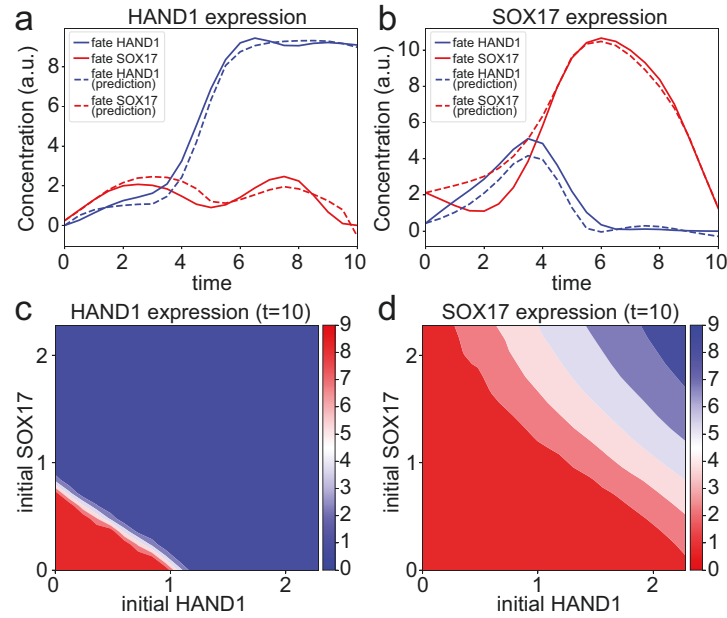


FIG. 5: HAND1-SOX17 fate predictions from non-autonomous system for single-cell qPCR data. (a) and (b) Temporal dynamics of expression for (a) HAND1 and (b) SOX17 where solid lines show training data and dashed lines show predictions for training data given the identical initial conditions. (c) and (d) Expression map for (c) HAND1 and (d) SOX17 given by different initial expression levels.

predicted gene expression maps from the model using non-autonomous system comply with this prior knowledge, which indicates the model may have better generalizability in predicting gene expression dynamics and cell-fate decisions.

Furthermore, we imputed gene expression dynamics and predicted cell fate at the final time for cells in the single-cell qPCR dataset. For cells with pseudotime $t > 9.9$, we assume cells have committed their fates and they are excluded from the imputation experiment. Indeed, 1209 cells are selected where 710 and 499 of them belong to fate HAND1 and SOX17, respectively. Gene expression dynamics for all cells can be imputed along the time axis up to the final time (i.e., $t = 10$). Then, cell-fate decisions can be determined by expression levels of HAND1 and SOX17 at the final time, where a cell has the fate of a marker gene if this gene has higher level than the other. Indeed, a boundary for distinguishing cell fates can be drawn on the expression graph with a linear function (i.e., $y = x$) [Figs. 6(a) and 6(b)]. For both autonomous and non-autonomous systems, most cells locate sufficiently away from the cell-fate boundary, indicating their clear commitment to one fate. Particularly, the model using non-autonomous system shows more clear separation to the cell-fate boundary. Using the lineage inferred from pseudotime as the ground truth, we can quantify the fate classification accuracy achieved by each model. The

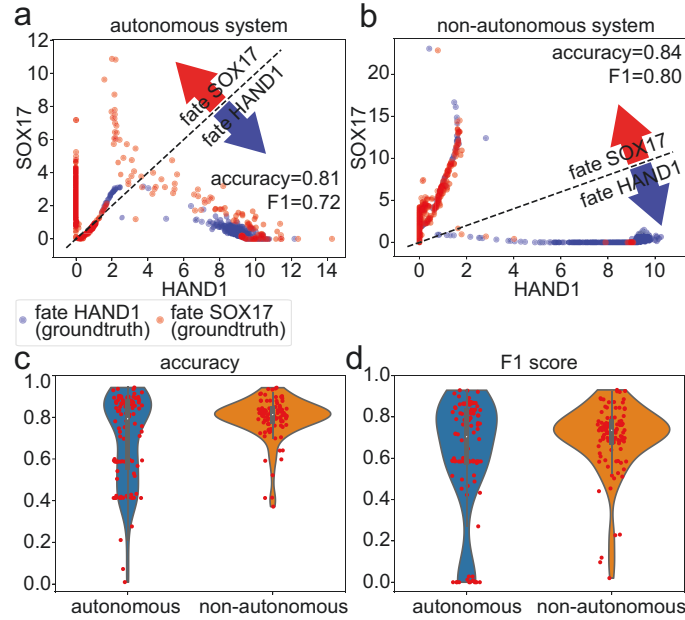


FIG. 6: Comparisons for NeuralGene using autonomous and non-autonomous systems for single-cell qPCR data. (a) and (b) NeuralGene imputes the dynamics of gene expression for cells from the single-cell qPCR dataset, and predictions of gene expression at final time are plotted for HAND1 and SOX17 genes. The ground-truth cell fate from pseudotime calculation is denoted by blue (fate HAND1) and red (fate SOX17) in the scatter plot. The plots are generated for (a) autonomous system and (b) non-autonomous system. (c) and (d) Violin plots and scatter plots show cell-fate classification performance from $n = 100$ independent repeats for both autonomous system and non-autonomous systems using metrics (c) ACC and (d) F1 score. The inner box in the violin plot shows five-number summary of the $n = 100$ data, where center of box shows median; upper and lower limits of box show upper and lower quartiles; and upper and lower whiskers show the maximum and the minimum by excluding “outliers” outside the interquartile range.

non-autonomous system has clear better accuracy and F1 score than the autonomous system with 3.7% and 11.1% improvement, respectively. We further trained each model with $n = 100$ independent repeats with different random seeds. The statistics of accuracy and F1 score show consistent results from the comparison in single repeat, where the five-number summary (i.e., median, lower and upper quartile, and maximum and minimum) is shown in the inner boxes in the violin plots [Figs. 6(c) and 6(d)]. The non-autonomous system achieves 0.80 and 0.70 for average accuracy and average F1 score, respectively. The autonomous system achieves 0.70 and 0.61 for average accuracy and average F1 score, respectively. Indeed, the non-autonomous system shows significant improvement over autonomous system with 14.3% and 14.8% average improvement on accuracy and F1 score, respectively. Interestingly, the autonomous system is more sensitive than the non-autonomous system, where many repeats have blow-up solutions along with the extremely low accuracy and F1 score.

In building complicated gene interaction model, it may be reasonable to consider only a few critical genes for simplification. However, many other genes may be involved in the interactions. The non-autonomous systems permit a solution to include these genes implicitly via the time-dependent right-hand side. As a result, the non-autonomous systems provide more accurate predictions in gene expression dynamics and cell-fate predictions than the autonomous systems.

5. CONCLUSIONS AND DISCUSSION

We present a DNN-based dynamical model in learning temporal dynamics of gene interactions from expression data. NeuralGene successfully recapitulates the cell-fate decision from a simulated toggle-switch model. It has a strong flexibility to integrate with arbitrary prior knowledge to improve its accuracy. We further applied it to a single-cell qPCR dataset. Despite the fact that each cell only provides the gene expression at a static time point, NeuralGene can accurately impute the temporal dynamics of gene expression and consequently predict its cell fate. Furthermore, we explored different formation of ODEs in modeling the dynamics, concretely, autonomous and non-autonomous systems. Since we only investigated the interactions among a small number of genes, additional unknown genes may be critical to the interactions. The non-autonomous systems with time-dependent right-hand sides provide better description regarding these unknown genes in the complex interactions. As a result, a non-autonomous system may be a better candidate in studying complex gene regulation.

In single-cell data, cells collected at the same physical time point may have different developmental stages. Our NeuralGene provides an approach using pseudotime to infer the intrinsic developmental stages of cells. However, this may lead to limited training data; for examples, two sets of training data are obtained for HAND1 and SOX17 fates in the qPCR data. To handle this problem, it may be possible to utilize the physical time for dynamics inference by using single-cell data collected at various time points. This is challenging for a direct application of NeuralGene or other similar approaches using neural ODEs, since no paired data are provided. For unpaired data, optimal transport is an alternative to link cells from different snapshots for inferring cell transition, such as Waddington OT (Schiebinger et al., 2019). Moreover, it is possible to formulate a density-based model to deal with the unpaired data (Tong et al., 2020), but this may be more computationally expensive with partial differential equations involved.

NeuralGene provides a proof of principle by approximating the right-hand side of ODEs using DNNs. In principle, this framework could be extended to the general case by using more sophisticated architectures in DNNs or different types of neural networks due to its property for universal approximation (Hornik, 1991). For example, one may consider different activation

functions such as Tanh that is widely used in PINN. One may add dropout and batch normalization layers in DNNs to have better generalizability for noisy data, limited training set, etc. Furthermore, more advanced layers such as convolutional layers, attention layers, etc., can be used to approximate the right-hand side with possible better accuracy. When prior knowledge for gene regulation is available, it is also possible to truncate the connections between different genes in the neural network; for example, one can remove the edges in neural networks if two genes are not interacting. In addition, it is possible to infer dynamics of gene regulation by considering a system with more genes involved. The computational cost of NeuralGene is linearly proportional to the number of genes being considered. The utilization of GPUs could potentially offer superior scalability, possible resulting in computational costs that are sublinear to the number of genes being evaluated.

ACKNOWLEDGMENTS

The work was supported by a National Science Foundation Grant No. DMS1763272, a Grant from the Simons Foundation (No. 594598 to Q.N.), and a National Institutes of Health Grant No. R01AR079150. The authors declare no competing interests.

REFERENCES

- Bargaje, R., Trachana, K., Shelton, M.N., McGinnis, C.S., Zhou, J.X., Chadick, C., Cook, S., Cavanaugh, C., Huang, S., and Hood, L., Cell Population Structure Prior to Bifurcation Predicts Efficiency of Directed Differentiation in Human Induced Pluripotent Cells, *Proc. Natl. Acad. Sci.*, vol. **114**, no. 9, pp. 2271–2276, 2017.
- Brunton, S.L., Proctor, J.L., and Kutz, J.N., Discovering Governing Equations from Data by Sparse Identification of Nonlinear Dynamical Systems, *Proc. Natl. Acad. Sci.*, vol. **113**, no. 15, pp. 3932–3937, 2016.
- Chen, R.T., Rubanova, Y., Bettencourt, J., and Duvenaud, D.K., Neural Ordinary Differential Equations, *Proc. of the 31st Conf. on Advances in Neural Information Processing Systems*, Montreal, Canada, 2018.
- Chen, Z., Churchill, V., Wu, K., and Xiu, D., Deep Neural Network Modeling of Unknown Partial Differential Equations in Nodal Space, *J. Comput. Phys.*, vol. **449**, p. 110782, 2022.
- Dormand, J.R. and Prince, P.J., A Family of Embedded Runge-Kutta Formulae, *J. Comput. Appl. Math.*, vol. **6**, no. 1, pp. 19–26, 1980.
- Gardner, T.S., Cantor, C.R., and Collins, J.J., Construction of a Genetic Toggle Switch in *Escherichia coli*, *Nature*, vol. **403**, no. 6767, pp. 339–342, 2000.
- Hornik, K., Approximation Capabilities of Multilayer Feedforward Networks, *Neural Networks*, vol. **4**, no. 2, pp. 251–257, 1991.
- Kingma, D.P. and Ba, J., Adam: A Method for Stochastic Optimization, *3rd Int. Conf. for Learning Representations*, San Diego, CA, 2014.
- Kramer, B.P., Viretta, A.U., Baba, M.D.-E., Aubel, D., Weber, W., and Fussenegger, M., An Engineered Epigenetic Transgene Switch in Mammalian Cells, *Nat. Biotechnol.*, vol. **22**, no. 7, pp. 867–870, 2004.
- Lander, A., Pattern, Growth, and Control, *Cell*, vol. **144**, no. 6, pp. 955–969, 2011.
- Lander, A., Nie, Q., and Wan, F., Membrane-Associated Non-Receptors and Morphogen Gradients, *Bull. Math. Biol.*, vol. **69**, no. 1, pp. 33–54, 2007.
- Lander, A.D., Nie, Q., and Wan, F.Y., Spatially Distributed Morphogen Production and Morphogen Gradient Formation, *Math. Biosci. Eng.*, vol. **2**, no. 2, p. 239, 2005.

- Lusch, B., Kutz, J.N., and Brunton, S.L., Deep Learning for Universal Linear Embeddings of Nonlinear Dynamics, *Nat. Commun.*, vol. **9**, no. 1, pp. 1–10, 2018.
- Mangan, N.M., Brunton, S.L., Proctor, J.L., and Kutz, J.N., Inferring Biological Networks by Sparse Identification of Nonlinear Dynamics, *IEEE Trans. Mol. Biol. Multi-Scale Commun.*, vol. **2**, no. 1, pp. 52–63, 2016.
- Nie, Q., Qiao, L., Qiu, Y., Zhang, L., and Zhao, W., Noise Control and Utility: From Regulatory Network to Spatial Patterning, *Sci. China Math.*, vol. **63**, no. 3, pp. 425–440, 2020.
- Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., Devito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A., Automatic Differentiation in Pytorch, in *31st Conf. on Neural Information Processing Systems*, Long Beach, CA, 2017.
- Qiao, L., Zhao, W., Tang, C., Nie, Q., and Zhang, L., Network Topologies That Can Achieve Dual Function of Adaptation and Noise Attenuation, *Cell Syst.*, vol. **9**, no. 3, pp. 271–285, 2019.
- Qin, T., Wu, K., and Xiu, D., Data Driven Governing Equations Approximation Using Deep Neural Networks, *J. Comput. Phys.*, vol. **395**, pp. 620–635, 2019.
- Qiu, Y., Chen, W., and Nie, Q., Stochastic Dynamics of Cell Lineage in Tissue Homeostasis, *Discrete Continuous Dyn. Syst. Ser. B*, vol. **24**, no. 8, p. 3971, 2019.
- Qiu, Y., Fung, L., Schilling, T.F., and Nie, Q., Multiple Morphogens and Rapid Elongation Promote Segmental Patterning during Development, *PLoS Comput. Biol.*, vol. **17**, no. 6, p. e1009077, 2021.
- Raissi, M., Perdikaris, P., and Karniadakis, G.E., Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations, *J. Comput. Phys.*, vol. **378**, pp. 686–707, 2019.
- Roesch, E., Rackauckas, C., and Stumpf, M.P., Collocation Based Training of Neural Ordinary Differential Equations, *Stat. Appl. Gen. Mol. Biol.*, vol. **20**, no. 2, pp. 37–49, 2021.
- Rubanova, Y., Chen, R.T., and Duvenaud, D.K., Latent Ordinary Differential Equations for Irregularly-Sampled Time Series, *Proc. of the 32st Conf. on Advances in Neural Information Processing Systems*, New York, NY: Curran Associates, 2019.
- Schiebinger, G., Shu, J., Tabaka, M., Cleary, B., Subramanian, V., Solomon, A., Gould, J., Liu, S., Lin, S., and Berube, P., Optimal-Transport Analysis of Single-Cell Gene Expression Identifies Developmental Trajectories in Reprogramming, *Cell*, vol. **176**, no. 4, pp. 928–943, 2019.
- Scrucca, L., Fop, M., Murphy, T.B., and Raftery, A.E., mclust 5: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models, *R J.*, vol. **8**, no. 1, p. 289, 2016.
- Shen, J., Liu, F., Tu, Y., and Tang, C., Finding Gene Network Topologies for Given Biological Function with Recurrent Neural Network, *Nat. Commun.*, vol. **12**, no. 1, pp. 1–10, 2021.
- Shen-Orr, S. S., Milo, R., Mangan, S., and Alon, U., Network Motifs in the Transcriptional Regulation Network of *Escherichia coli*, *Nat. Gen.*, vol. **31**, no. 1, pp. 64–68, 2002.
- Street, K., Risso, D., Fletcher, R.B., Das, D., Ngai, J., Yosef, N., Purdom, E., and Dudoit, S., Slingshot: Cell Lineage and Pseudotime Inference for Single-Cell Transcriptomics, *BMC Gen.*, vol. **19**, no. 1, pp. 1–16, 2018.
- Tang, B., Orthogonal Array-Based Latin Hypercubes, *J. Am. Stat. Assoc.*, vol. **88**, no. 424, pp. 1392–1397, 1993.
- Tong, A., Huang, J., Wolf, G., Van Dijk, D., and Krishnaswamy, S., Trajectorynet: A Dynamic Optimal Transport Network for Modeling Cellular Dynamics, in *Int. Conf. on Machine Learning*, Virtual, pp. 9526–9536, 2020.
- Yazdani, A., Lu, L., Raissi, M., and Karniadakis, G.E., Systems Biology Informed Deep Learning for Inferring Parameters and Hidden Dynamics, *PLoS Comput. Biol.*, vol. **16**, no. 11, p. e1007575, 2020.

- Zhang, L., Radtke, K., Zheng, L., Cai, A.Q., Schilling, T.F., and Nie, Q., Noise Drives Sharpening of Gene Expression Boundaries in the Zebrafish Hindbrain, *Mol. Syst. Biol.*, vol. **8**, no. 1, p. 613, 2012.
- Zhu, Y., Qiu, Y., Chen, W., Nie, Q., and Lander, A.D., Scaling a Dpp Morphogen Gradient through Feedback Control of Receptors and Co-Receptors, *Dev. Cell*, vol. **53**, no. 6, pp. 724–739, 2020.
- Zhuang, J., Dvornek, N., Li, X., Tatikonda, S., Papademetris, X., and Duncan, J., Adaptive Checkpoint Adjoint Method for Gradient Estimation in Neural ODE, in *Int. Conf. on Machine Learning*, Virtual, pp. 11639–11649, 2020.

APPENDIX A. EVALUATING METRICS

In this study, mean-squared error (MSE) was used to quantify the accuracy from the predicted temporal dynamics of gene expression comparing to the ground-truth data. The MSE is defined as

$$\text{MSE} = \frac{1}{J} \sum_{j=1}^J \|y_j - x_j\|^2, \quad (\text{A.1})$$

where x_j and y_j are, respectively, ground truth and predicted gene expression vector including all genes at time t_j .

Another task is to classify cell-fate decisions, particularly, it is a binary classification in our work with two cell fates determined by two toggle-switch genes. In this work, we determine cell fate by comparing expression levels of two genes at the final time point. The cell commits to the fate marked by one gene if its level is higher than the other gene. We denote one cell fate as positive state and the other as negative state. Specifically, we take state *A* in simulation data as the positive state and fate HAND1 in real data as the positive state. A confusion matrix can be constructed by comparing testing results from NeuralGene and the ground-truth cell fates over all cells in the testing set. True positive (TP) and false negative (FN) are occurrences where testing results correctly indicate the ground-truth positive and negative states, respectively. False positive (FP) and false negative (FN) are occurrences where testing results wrongly indicate the ground-truth positive and negative states, respectively. We used accuracy and F1 score (F1) to quantify the performance for the binary classification. Specifically, accuracy quantifies the frequency that predictions correctly predict the cell fates:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \quad (\text{A.2})$$

Accuracy = 0 indicates none of predictions is correct, Accuracy = 1 indicates all predictions are correct, and random classification can lead to Accuracy = 0.5. Indeed, a classifier is effective when Accuracy > 0.5.

Furthermore, precision determines the proportion of positive identifications was correct, and recall determines the proportion of actual positive was identified correctly:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (\text{A.3})$$

Then F1, score is taken as the harmonic mean of precision and recall quantifying the classification accuracy:

$$\text{F1} = \frac{2}{\text{Precision}^{-1} + \text{Recall}^{-1}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \quad (\text{A.4})$$

The range of F1 is between 0 and 1, where 1 indicates the best performance and 0 indicates the worst performance.

APPENDIX B. NEURAL ODES IMPLEMENTATION AND HYPERPARAMETER SELECTION

We used the adaptive checkpoint adjoint method along with their package to implement neural ODEs (Zhuang et al., 2020). The neural network in neural ODEs was implemented by PyTorch (Paszke et al., 2017). DOPRI5 (Dormand and Prince, 1980), a Runge-Kutta method, was used as the ODEs solver to perform forward propagation for neural ODEs. The adjoint method proposed by neural ODEs is a memory-efficient method to update gradients, while it requires longer computational time. Due to the small size of neural network architecture which requires small memory, we used the naive auto-differentiation in PyTorch for backward propagation to update gradients during training. The neural network is fully connected, and its architecture is given in Eq. (6).

To find the appropriate hyperparameters, we performed a grid search to find the optimal hyperparameters. For simulating toggle-switch model, we searched for hyperparameters for the case inferring both regulation of gene *A* and gene *B*, and the identical hyperparameters were used to the other two models in inferring regulation of either gene *A* or gene *B* (as shown in Fig. 2). For the single-cell qPCR dataset, we searched hyperparameters independently for both autonomous and non-autonomous systems. In each round of search, a set of hyperparameters was explored in ranges: number hidden layers ranged from 1 to 7, dimension of hidden layers ranged from 2^2 to 2^6 , learning rate ranged from 10^{-3} to 3×10^{-2} , and batch size ranged from 2^2 to 2^7 . Model with each set of hyperparameters can generate a plot for the training results [like Figs. 4(a) and 4(b)]. We manually select the best-fit curves for the optimal hyperparameters. Then, $n = 100$ independent repeats with different random seeds were performed using the optimal hyperparameters to investigate the statistics of performance.

Particularly, for the single-cell qPCR, there are only two time-series training datasets and batch size is then taken as 2. Other hyperparameters used in the model are listed in Table B1.

TABLE B1: Hyperparameters of NeuralGene

	Simulation	Single-cell qPCR data (autonomous system)	Single-cell qPCR data (non-autonomous system)
Number of hidden layers	1	3	7
Dimension of hidden layers	16	32	64
Learning rate	0.001	0.007	0.001
Batch size	32	2	2