

scMix: learning temporal dynamics of gene expression under irregular time intervals

Shangjin Han and Dongsup Kim*, 

Department of Bio and Brain Engineering, KAIST, Daejeon 34141, South Korea

*Corresponding author. Department of Bio and Brain Engineering, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, South Korea. E-mail: kds@kaist.ac.kr
 Associate Editor: Anthony Mathelier

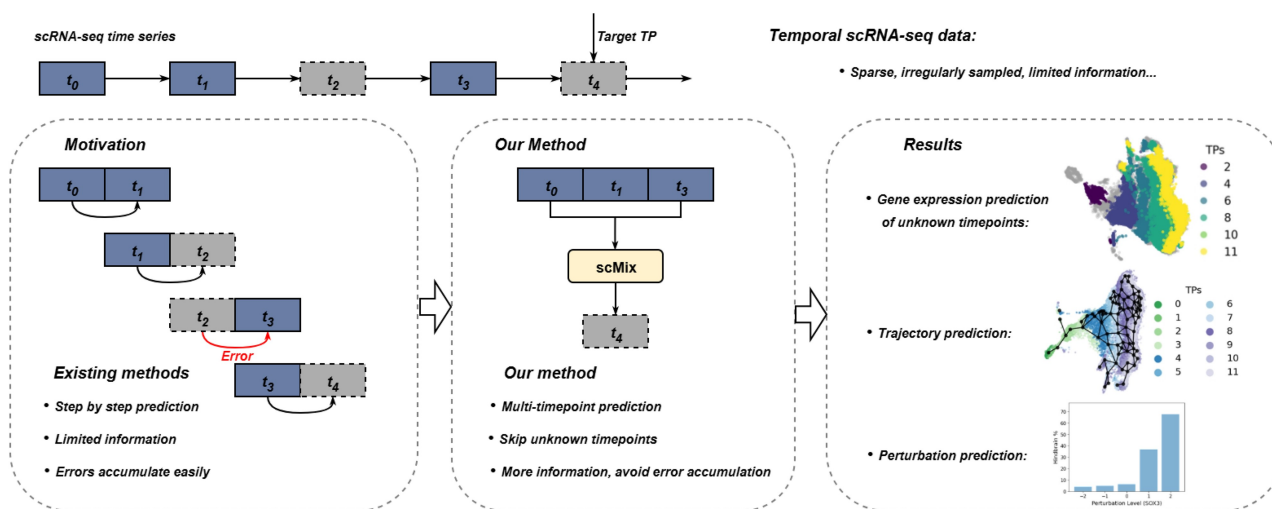
Abstract

Motivation: Understanding temporal gene expression is fundamental in the study of cellular development and differentiation. In practice, temporal single-cell datasets tend to contain only a limited number of measured time points, which are often unevenly spaced, resulting in irregular intervals between observations due to experimental constraints. Existing methods typically address these intervals by sequentially predicting one time point after another, yet lack mechanisms to explicitly model time intervals, leading to error accumulation.

Results: In this work, we introduce scMix, a language-model-based framework for predicting single-cell gene expression, which enables prediction from multiple historical time points. We build scMix on the Receptance Weighted Key Value architecture and use its time decay mechanism to model temporal dependencies over time. Moreover, scMix proposes a delta-time mechanism that allows the model to bypass unmeasured time points, reducing error accumulation and improving robustness. In addition, we incorporate a trend regularization strategy to enhance the temporal coherence of predicted gene expression trajectories. scMix demonstrates state-of-the-art performance in predicting gene expression at unmeasured time points, surpassing existing methods, and also achieves outstanding results on downstream tasks.

Availability and implementation: The code used for this study is available at <https://doi.org/10.5281/zenodo.18287184>.

Graphical abstract



Received: 18 October 2025. Revised: 21 January 2026. Accepted: 10 February 2026

© The Author(s) 2026. Published by Oxford University Press.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

Embryonic development, cellular differentiation, and disease progression are dynamic processes driven by gene expression programs that change over time (Raj and Van Oudenaarden 2008, Trapnell *et al.* 2014, La Manno *et al.* 2018). Elucidating these processes requires methods capable of characterizing cellular states and capturing their temporal evolution along developmental trajectories. The advent of single-cell RNA sequencing (scRNA-seq) has made this possible by enabling transcriptome-wide measurements at cellular resolution across multiple time points (Trapnell *et al.* 2014, Klein *et al.* 2015, Qiu *et al.* 2017). This technology enables the reconstruction of developmental pathways and lineage relationships, providing valuable insights into the temporal dimension of biology (Raj and Van Oudenaarden 2008, Treutlein *et al.* 2016).

Unlike conventional time series data, temporal scRNA-seq has distinct characteristics that create specific challenges for analysis. First, these datasets are often sparse, containing only a limited number of time points (La Manno *et al.* 2018, Saelens *et al.* 2019, Bergen *et al.* 2020). Second, some time points may be missing due to experimental constraints, creating gaps in the timeline (La Manno *et al.* 2018, Lähnemann *et al.* 2020, Wagner and Klein 2020). In addition, each cell is profiled only once, which prevents longitudinal tracking and eliminates the possibility of one-to-one correspondence between cells at different time points (Raj and Van Oudenaarden 2008, Schiebinger *et al.* 2019, Weinreb *et al.* 2020). These characteristics make modeling temporal single-cell data particularly challenging.

Despite the challenges, several methods have been proposed to characterize dynamic gene expression. Diffusion pseudotime (Haghverdi *et al.* 2016) reconstructs lineage branching by ordering cells along diffusion trajectories, while Slingshot (Street *et al.* 2018) reconstructs developmental lineages by fitting smooth curves through clustered cells. RNA velocity (La Manno *et al.* 2018) infers short-term transcriptional changes from spliced and unspliced mRNAs, while scVelo (Bergen *et al.* 2020) and Dynamo (Qiu *et al.* 2022) extend this framework by modeling full splicing dynamics and vector fields, enabling transient-state analysis and fate prediction. Although these methods provide valuable dynamic insights, they do not directly predict temporal gene expression.

A variety of deep learning models have been proposed for temporal single-cell gene expression prediction. PRESCIENT (Yeo *et al.* 2021) models cell differentiation by learning a potential energy function whose gradient field represents cellular dynamics, providing an energy landscape interpretation of trajectories within the Waddington-optimal transport (OT) framework (Schiebinger *et al.* 2019). This approach is formulated within a stochastic differential equation (SDE) framework. Building on PRESCIENT, PI-SDE (Jiang and Wan 2024) incorporates the Hamilton–Jacobi equation (Bressloff 2021), yielding a physics-informed SDE. MIOFlow (Huguet *et al.* 2022) combines OT with neural ordinary differential equations (ODEs) (Chen *et al.* 2018) in a manifold embedding space and employs a Geodesic Autoencoder (Huguet *et al.* 2022) to preserve geometry. scNODE (Zhang *et al.* 2024) integrates a variational autoencoder (Kingma and Welling 2013) with neural ODEs and introduces dynamic regularization to maintain temporal

consistency in the latent space. Although these methods differ in strategy, they all rely on SDEs or neural ODEs, which have inherent limitations. First, predictions must be computed sequentially, requiring step-by-step progression between time points. Each step depends only on the previous one, preventing the integration of multiple time points and limiting predictive power. Second, the sequential structure prohibits skipping time points: unobserved intermediates must be estimated before later states, causing errors to accumulate as the prediction range increases.

To address the limitations, we propose scMix, which overcomes the constraints of SDE- and ODE-based models and leverages language models for temporal single-cell gene expression prediction. Specifically, we treat gene expression observations at different time points as temporal tokens, and employ the Time-Mixing module of the Receptance Weighted Key Value (RWKV) (Peng *et al.* 2023) model to integrate information from multiple previous known time points for prediction. The Time-Mixing module is based on a time decay mechanism, which aligns closely with the dynamic process of cell development by modeling the phenomenon in which the influence of early information gradually diminishes over time. In addition, we introduce a novel Δt mechanism as an extension to the original RWKV, enabling the model to perceive irregular time intervals. This mechanism enables time point skipping, allowing the model to directly predict future time points while bypassing intermediate unknown ones, thereby reducing error accumulation. Moreover, our model proposes trend regularization to enhance temporal coherence, ensuring smooth transitions between consecutive time points. Experiments demonstrate that scMix significantly outperforms baseline methods in predicting gene expression at unmeasured time points, as well as on downstream tasks.

2 Materials and methods

2.1 Datasets

We evaluate our model on four benchmark temporal scRNA-seq datasets. Each dataset is derived from a distinct species, providing a diverse evaluation setting: *Zebrafish* (*Danio rerio*, 12 time points; Farrell *et al.* 2018), *Drosophila* (*Drosophila melanogaster*, 11 time points; Calderon *et al.* 2022), *Schiebinger2019* (*Mus musculus*, 19 time points; Schiebinger *et al.* 2019) and *Veres* (*Homo sapiens*, 8 time points; Veres *et al.* 2019).

For datasets *Zebrafish*, *Drosophila*, and *Schiebinger2019*, we followed the preprocessing strategy described in scNODE (Zhang *et al.* 2024). For each dataset, we selected the top 2000 highly variable genes (HVGs) to retain the most informative expression patterns, and log transformation is applied to the expression. For dataset *Veres*, the preprocessing method was adapted from the procedure described in PI-SDE (Jiang and Wan 2024). Gene expression was z-score normalized, and features were filtered based on correlations with the marker gene TOP2A, removing genes with Pearson correlation coefficients above 0.15. In our implementation, we further applied HVG selection, retaining the top 100 genes. Detailed data preprocessing

procedures are provided in [Supplementary S1](#), available as [supplementary data](#) at *Bioinformatics* online..

2.2 Problem formulation

Let $\{t_1, t_2, \dots, t_K\}$ denote the observed time points, where each $t_k \in \mathbb{R}$. Since the time points are not evenly spaced, we define the intervals as $\Delta t = \{0, t_2 - t_1, t_3 - t_2, \dots, t_K - t_{K-1}\}$, which has length K . At each time point t_k , we observe single-cell gene expression profiles $\mathcal{X}_{t_k} = \{x_{t_k}^{(i)} \in \mathbb{R}^G\}_{i=1}^{N_k}$, where G is the fixed number of genes and N_k is the number of cells, which varies across time points. Given a target time point $t_{\text{target}} \in \mathbb{R}$, the goal is to characterize the gene expression distribution of the cell population at t_{target} . This target distribution is denoted as $\mathcal{X}_{t_{\text{target}}} = \{x_{t_{\text{target}}}^{(i)}\}_{i=1}^{N_{\text{target}}}$, where N_{target} denotes the number of cells observed at the target time point, and is modeled as a function of the observed time points and the corresponding time information:

$$\hat{\mathcal{X}}_{t_{\text{target}}} = f_{\theta}(\{x_{t_k}, t_k, t_{\text{target}}, \Delta t_k\}_{k=1}^T).$$

2.3 Delta-RWKV block

2.3.1 Delta-Time Mixing

To account for unevenly spaced time intervals in time series data, we propose *Delta-Time Mixing* (Fig. 1b), adapted from the

RWKV (Peng et al. 2023) framework. This block extends the original Time-Mixing mechanism by explicitly incorporating the time intervals between observations.

Let $z_t \in \mathbb{R}^C$ denote the feature representation at time step t , and let z_{t-1} denote the representation at the previous step. Following the original RWKV (Peng et al. 2023), the receptance (r_t), key (k_t), and value (v_t) vectors are obtained by linearly mixing the current and previous inputs with learnable parameters $\mu_r, \mu_k, \mu_v \in \mathbb{R}^C$ and projecting them as

$$r_t = W_r(\mu_r \odot z_t + (1 - \mu_r) \odot z_{t-1}), \quad (1)$$

$$k_t = W_k(\mu_k \odot z_t + (1 - \mu_k) \odot z_{t-1}), \quad (2)$$

$$v_t = W_v(\mu_v \odot z_t + (1 - \mu_v) \odot z_{t-1}), \quad (3)$$

where $W_r, W_k, W_v \in \mathbb{R}^{C \times C}$ are learnable projection matrices, $\odot$ denotes element-wise multiplication.

Next, k_t and v_t are used to compute wkv_t using an attention mechanism that aggregates past information:

$$wkv_t = \frac{\sum_{i=1}^{t-1} e^{-(t-1-i)w \cdot \Delta t_i + k_i} \odot v_i + e^{u + k_t} \odot v_t}{\sum_{i=1}^{t-1} e^{-(t-1-i)w \cdot \Delta t_i + k_i} + e^{u + k_t}}. \quad (4)$$

Here, $w \in \mathbb{R}^C$ is a learnable decay parameter applied to past information, and u is a learnable bias term for current

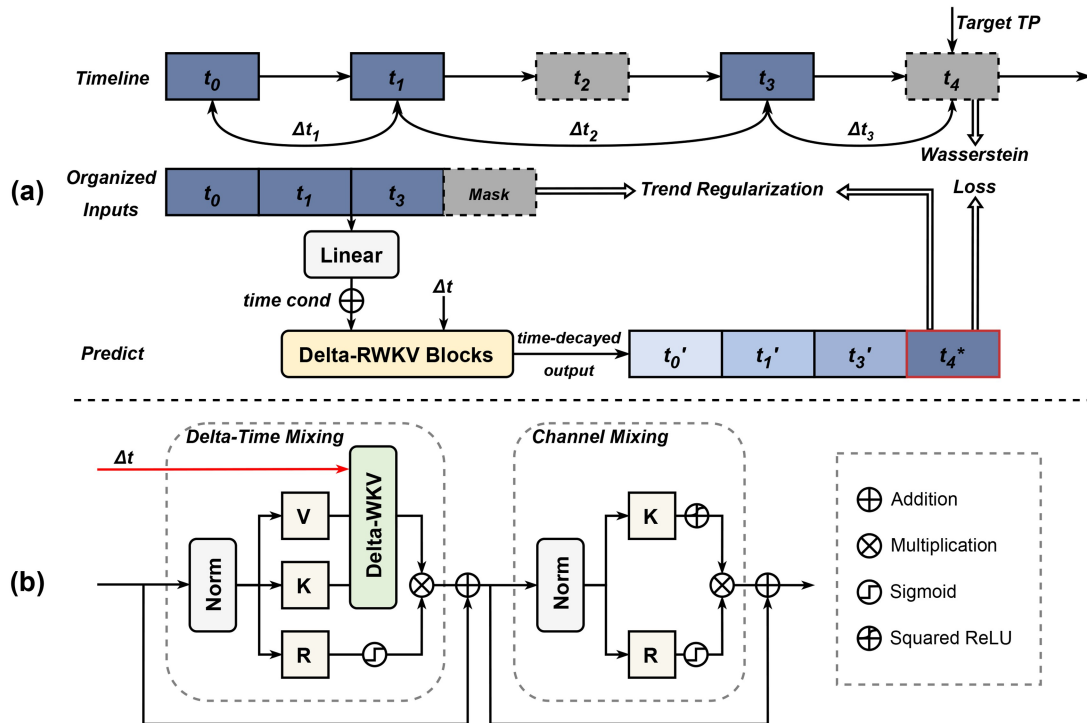


Figure 1 scMix architecture. (a) scMix overall. For each target time point, several preceding visible time points are taken as input, followed by an appended zero mask token serves as output position. The input sequence is first projected by a linear layer and then fused with time information embeddings. The fused representation, together with Δt , is fed into Delta-RWKV blocks, which outputs a decayed information sequence, where the final mask token is taken as the prediction for the target time point. The Wasserstein distance between predicted and target expression is computed, and the predicted expression is also compared with visible time points to calculate trend regularization loss. (b) Delta-RWKV block. It consists of a Delta-Time Mixing block and a Channel Mixing block. Delta-Time Mixing extends the RWKV Time-Mixing mechanism by incorporating Δt into the decay term, enabling the model to account for uneven time intervals. Channel Mixing captures nonlinear interactions across feature channels.

information. We apply the time interval Δt to the past term w , making larger intervals induce stronger decay and smaller intervals weaker decay.

The weighted aggregation is then scaled by the receptance and linearly projected to produce the final output:

$$o_t = W_o(\sigma(r_t) \odot wkv_t), \quad (5)$$

where $\sigma(r_t)$ acts as a gating factor applied element-wise to wkv_t , and W_o projects the gated result to the final output.

2.3.2 Channel Mixing

Channel Mixing (Fig. 1b) complements Delta-Time Mixing to form the Delta-RWKV block, capturing nonlinear feature interactions at each time step. Given input $z_t \in \mathbb{R}^C$, channel-level representations are obtained by mixing the current and preceding states through learnable projections:

$$r_t = W_r(\mu_r \odot z_t + (1 - \mu_r) \odot z_{t-1}), \quad (6)$$

$$k_t = W_k(\mu_k \odot z_t + (1 - \mu_k) \odot z_{t-1}), \quad (7)$$

where $W_r, W_k \in \mathbb{R}^{C \times C}$ are learnable weights and μ_r, μ_k are learnable mixing factors.

The output is then obtained through a gated transformation:

$$o_t = \sigma(r_t) \odot (W_v \cdot (\max(k_t, 0))^2), \quad (8)$$

with $W_v \in \mathbb{R}^{C \times C}$. Here, the gating term $\sigma(r_t)$ controls the contribution of each channel, while the squared ReLU term $(\max(k_t, 0))^2$ introduces sparsity and nonlinearity. Through this mechanism, Channel Mixing enhances the representational capacity of channel features.

2.4 Overall scMix architecture

Given a target time point t_{target} , the model takes as input the four most recent observed time points, denoted as $t_{i-3}, t_{i-2}, t_{i-1}, t_i$, where t_i is the last observation preceding t_{target} . The number of input steps can be adjusted, but experiments on datasets with 8–18 time points show that using four preceding steps provides the best balance between model performance and the ability to capture gene–cell developmental relationships, with further details provided in the [Supplementary S3](#), available as [supplementary data](#) at [Bioinformatics](#) online. If fewer than four observations are available, the earliest observed time point is used for padding. The input sequence consists of T observed steps, each represented by a gene expression vector $x_t \in \mathbb{R}^G$, where G is the number of genes. For a batch of size B , these sequences form a tensor of shape $B \times T \times G$. To predict the expression at t_{target} , we append an all-zero vector after the observed time points, so that the input has shape $B \times (T+1) \times G$. The model output at the appended position is then used as the prediction for t_{target} .

In addition to gene expression, we incorporate time information as a condition. The time information consists of three components: the observed time t_{obs} , the target time t_{target} , and their distance Δt . Each component is mapped by an embedding

function $\phi(\cdot)$. The resulting embeddings are fused by applying a linear transformation W_e to the aggregated time embeddings and adding it to the linearly projected gene representation $\text{Linear}(x_t)$:

$$z_t = \text{Linear}(x_t) + W_e(\phi(t_{\text{obs}}) + \phi(t_{\text{target}}) + \phi(\Delta t)). \quad (9)$$

The fused embeddings $\{z_t\}_{t=1}^{T+1}$ are processed in temporal order through a stack of Delta-RWKV blocks. At each time t , the hidden representation is updated as:

$$h_t = \text{Delta-RWKV}(z_t, \Delta t_t). \quad (10)$$

The Delta-RWKV blocks produce hidden states $\{h_t\}_{t=1}^{T+1}$, and the final state h_{T+1} , corresponding to the appended zero vector, is used for prediction:

$$\hat{x}_{t_{\text{target}}} = h_{T+1}. \quad (11)$$

2.5 Loss function

2.5.1 Wasserstein distance loss

Since cells across time points are not one-to-one matched, the evaluation focuses on population-level distribution differences. To this end, we employ the Wasserstein distance ([Peyré and Cuturi 2019](#)), which quantifies the discrepancy between two distributions. Formally, the p -Wasserstein distance is defined as:

$$W_p(\hat{X}_t, X_t) = \left(\inf_{\gamma \in \Pi(\hat{X}_t, X_t)} \int \|u - v\|^p d\gamma(u, v) \right)^{1/p}, \quad (12)$$

where X_t and $\hat{X}_t$ denote the observed and predicted distributions, respectively. Here, $\Pi(\hat{X}_t, X_t)$ denotes the set of joint distributions between $\hat{X}_t$ and X_t . We use the 2-Wasserstein distance W_2 , approximated with debiased Sinkhorn divergence.

Based on the Wasserstein distance, the loss is decomposed into two parts: *prediction loss* and *reconstruction loss*. Prediction loss aligns the predicted and true distributions at the target time point:

$$\mathcal{L}_{\text{pred}} = W_2(\hat{X}_{t_{\text{target}}}, X_{t_{\text{target}}}). \quad (13)$$

However, optimizing only the prediction loss may cause the model to overlook the dynamical consistency at intermediate time points. To address this, we introduce the reconstruction loss that aligns the model outputs with the observations at each observed time point, averaged over time ($1/T$):

$$\mathcal{L}_{\text{recon}} = \frac{1}{T} \sum_{t=1}^T W_2(\hat{X}_t, X_t). \quad (14)$$

The overall objective of Wasserstein distance loss is:

$$\mathcal{L}_{\text{wd}} = \mathcal{L}_{\text{pred}} + \mathcal{L}_{\text{recon}}. \quad (15)$$

2.5.2 Trend regularization loss

In addition to the *prediction* and *reconstruction* losses, we introduce a *trend regularization loss* to regularize the temporal

smoothness of the predicted trajectory. Specifically, we define the discrete temporal change rate between the prediction at the target time and the observation at the input time t as:

$$r_t = \frac{\hat{x}_{\text{target}} - x_t}{t_{\text{target}} - t_{\text{obs}}^{(t)}}, \quad (16)$$

where x_t denotes the observed state, $\hat{x}_{\text{target}}$ is the prediction at the target time point, $t_{\text{obs}}^{(t)}$ is the observed time stamp, t_{target} is the target time stamp. Based on r_t , the trend regularization loss is defined as the sum of squared temporal changes over genes and cells across the observed input time points:

$$\mathcal{L}_{\text{trend}} = \frac{1}{G} \sum_{t=1}^T \sum_{i=1}^N \sum_{g=1}^G \left(\frac{\hat{x}_{\text{target},i,g} - x_{t,i,g}}{t_{\text{target}} - t_{\text{obs}}^{(t)}} \right)^2. \quad (17)$$

Here, G , N , and T denote the numbers of genes, cells, and observed input time points, respectively. This formulation encourages smooth temporal evolution by penalizing large changes in the temporal change rate.

Combining the above components, the overall training objective is defined as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{wd}} + \lambda_{\text{trend}} \mathcal{L}_{\text{trend}}, \quad (18)$$

where λ_{trend} is a balancing coefficient. The coefficient λ_{trend} takes different values across datasets or tasks, and can be adjusted (details in Table S3, available as [supplementary data](#) at *Bioinformatics* online).

3 Results

3.1 scMix accurately predicts gene expressions at unseen time points

To evaluate the performance of scMix in predicting gene expression at unseen time points, we design two tasks corresponding

to time point splitting strategies: *Random Recovery* and *Forecasting*. For *Random Recovery*, we randomly mask approximately half of the available time points from the dataset and use the remaining time points as input to recover the missing ones. This task evaluates the model's ability to recover unobserved data from irregular time points. For *Forecasting*, we select the last 2 or 3 time points of each dataset as the test set. This task evaluates the model's extrapolation ability. The specific timepoint splits are summarized in Table S2, available as [supplementary data](#) at *Bioinformatics* online.

We first evaluated the Wasserstein distance between the predicted and true gene expression. Each model was trained and evaluated over five independent runs with different random seeds. As shown in Fig. 2, scMix achieved the best results across almost all time points on the four datasets. At only a few time points (e.g. time point 5 of *Schiebinger2019* and time point 2 of *Veres*), the baselines outperformed scMix, but the differences were minimal. Notably, in the *Forecasting* tasks, our model significantly outperforms all baselines, showing strong extrapolation ability. In addition, the baselines exhibit a lack of generalizability. For example, PRESCIENT performs well on *Veres* but poorly on *Zebrafish*. PI-SDE also performs well on *Veres* but performs poorly on *Schiebinger2019*. This indicates that the baselines are sensitive to specific data distributions. Furthermore, the baselines exhibit less stable performance. For example, MIOFlow shows large variance on both *Schiebinger2019* and *Veres* datasets, while scNODE also demonstrates high variance on *Veres*. By comparison, scMix consistently achieves the best performance across all datasets with low variance, demonstrating strong robustness.

We further visualized the results using Uniform Manifold Approximation and Projection (UMAP) (McInnes et al. 2018) to provide a more intuitive comparison between the predicted and target gene expression. Figure 3 shows the performance of different models on the *Random Recovery* task of the *Zebrafish* and *Veres* datasets (The remaining results are in Fig. S1, available as [supplementary data](#) at *Bioinformatics* online). Across all results, the predictions of scMix are consistently closer to the target distributions.

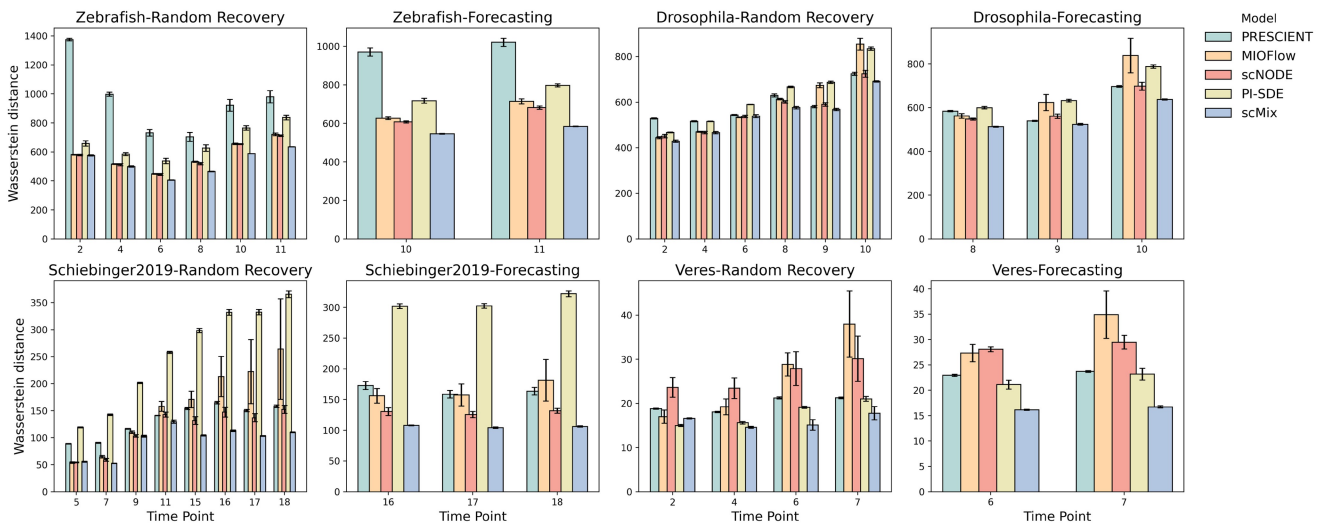


Figure 2 Comparison of Wasserstein distances between scMix and baselines across four datasets (*Zebrafish*, *Drosophila*, *Schiebinger 2019*, and *Veres*). Each dataset includes two tasks: *Random Recovery* task evaluates the ability of models to predict gene expression at randomly masked time points, while *Forecasting* task assesses extrapolation ability of the models. Across all datasets and tasks, scMix achieves lower Wasserstein distances.

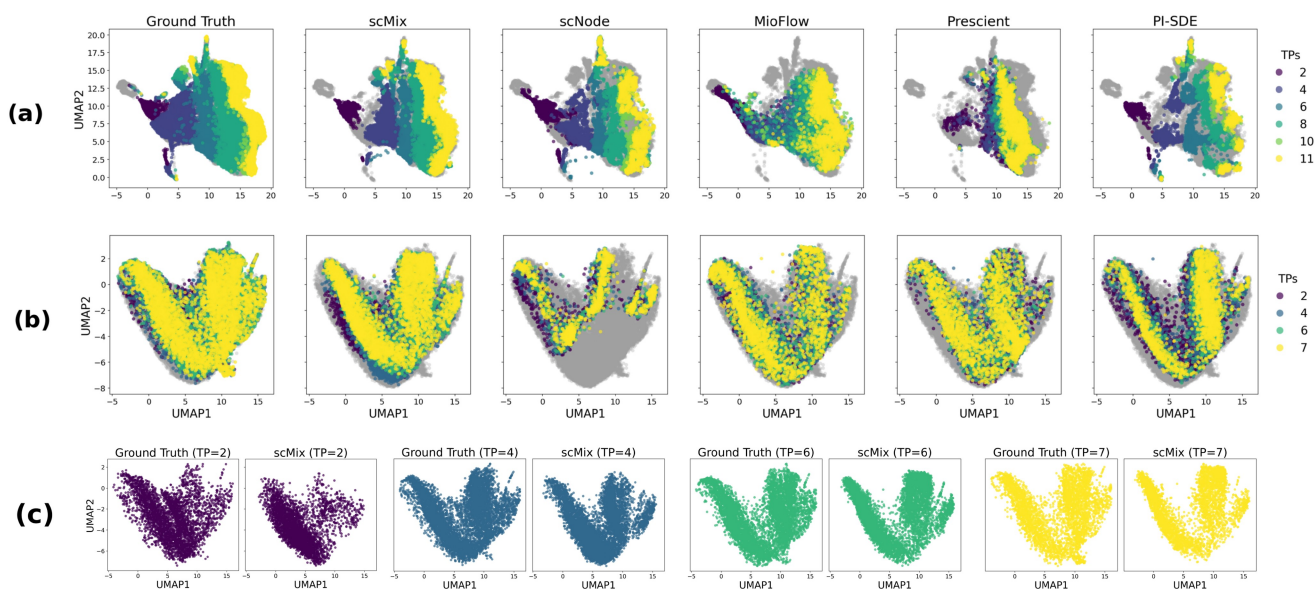


Figure 3 UMAP was used to project high-dimensional expression to two dimensions for visualization. (a) *Zebrafish* dataset on *Random Recovery* task. (b) *Veres* dataset on *Random Recovery* task. (c) Individual visualizations of each time point shown in subfigure (b).

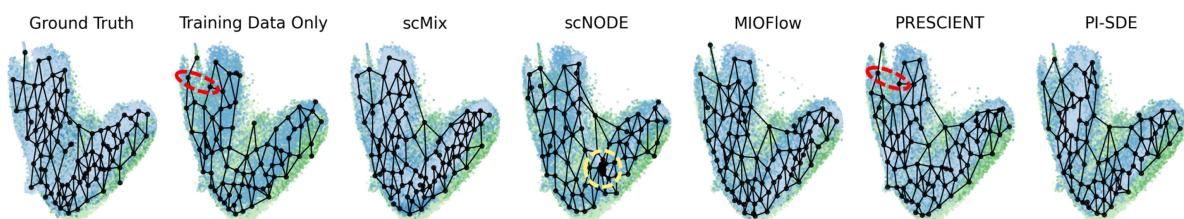


Figure 4 PAGA visualization was used to simulate cell trajectories and assess gene expression coherence. The *Training Data Only* panel shows the PAGA topology from training time points. For evaluation, predicted test time points were concatenated with training data to assess overall trajectory continuity. This result is from the *Veres* dataset on *Random Recovery* task. Trajectory discontinuities are highlighted by red circles, and over-aggregated regions are highlighted by yellow circles.

3.2 scMix exhibits greater coherence in trajectory prediction

To evaluate the temporal continuity of the predicted trajectories, we followed the experimental design of scNODE (Zhang et al. 2024). Specifically, we focus on the *Random Recovery* task, where training and predicted time points are combined, and we examine their joint distribution to assess overall temporal continuity.

For the visualization, we applied partition-based graph abstraction (PAGA) (Wolf et al. 2019), a graph-based method that reconstructs the topological structure of cellular populations. PAGA provides a coarse-grained representation of the connectivity between clusters by constructing a graph of nodes and edges, which enables evaluation of whether predicted cells form smooth connections across neighboring time points. In Fig. 4, the training time points (Training Data Only) are shown, with test time points removed; trajectories exhibit clear discontinuities in the PAGA graph (red circles). We then reconstruct cells at the missing test time points using different models. A reasonable prediction is expected to reconnect these broken trajectories and produce a PAGA structure consistent with the target topology. The results show that our model effectively repairs

the broken trajectories and produces a reasonable distribution of predicted cells. In contrast, the MIOFlow, PRESCIENT, and PI-SDE still exhibit trajectory breaks after prediction on the *Veres* and *Zebrafish* datasets (highlighted by red circles in Fig. 4 and Fig. S2a, available as supplementary data at Bioinformatics online), while some baseline predictions do not show explicit discontinuities but suffer from severe point aggregation, which is also considered unreasonable (highlighted by yellow circles in Fig. 4, Fig. S2b and c, available as supplementary data at Bioinformatics online). PAGA results of *Drosophila* and *Schiebinger2019* datasets are in Fig. S2, available as supplementary data at Bioinformatics online.

To further validate the PAGA results, we adopted the IM distance (Saelens et al. 2019) as a quantitative criterion. The IM distance measures the extent to which the topological structure of the predicted data aligns with that of the target distribution, based on their respective neighborhood graphs. A smaller IM distance indicates closer alignment between predicted and reference manifolds, whereas larger values reflect greater structural discrepancies. The metric ranges from 0, indicating identical structures, to 1, representing maximal dissimilarity. As shown in Fig. 5, we performed 5 trials, and our model achieved the lowest average IM distance. (The remaining results are

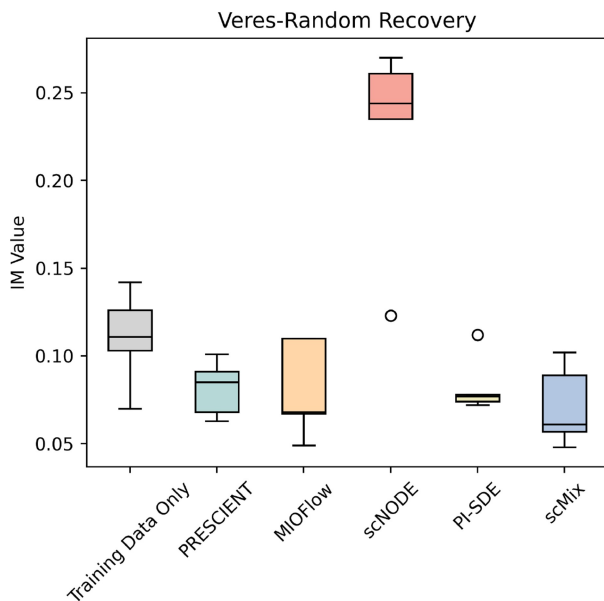


Figure 5 IM scores comparing PAGA trajectory graphs constructed from real and predicted data. The metric measures how well predicted trajectories align with those from real data, with lower values indicating higher consistency.

provided in [Fig. S3](#), available as [supplementary data](#) at [Bioinformatics online](#).)

3.3 scMix can learn gene–cell interactions

To assess whether our model can capture the interaction between genes and cells, we followed the perturbation experiment design introduced in scNODE (Zhang *et al.* 2024). This experiment is designed to assess whether the model can capture interactions between genes and cells by analyzing changes in cell proportions after gene perturbation. In this experiment, the analysis of the *Zebrafish* dataset in scNODE identified TBX16 as a regulator of presomitic mesoderm (PSM) cells (Warga *et al.* 2013) and SOX3 as a factor associated with Hindbrain cells (Dee *et al.* 2008).

Building on this design, we trained our model on all time points of the *Zebrafish* dataset and perturbed the expression of TBX16 and SOX3 at the initial time point by scaling their expression levels with multiplicative factors ranging from 10^{-2} to 10^2 , thereby simulating gene knock-out and overexpression. We then evaluated changes in cell proportions at the final time point of the dataset (time point 11 of *Zebrafish* dataset). As shown in [Fig. 6](#), TBX16 overexpression led to a pronounced increase in the proportion of PSM cells, reaching nearly 60% at the highest perturbation level, whereas knock-out reduced the proportion compared with the control condition. SOX3 overexpression strongly promoted Hindbrain cell differentiation, with the proportion rising to almost 70%, and knock-out decreased the proportion accordingly. However, perturbing SOX3 showed little influence on PSM cells, and TBX16 had negligible effects on Hindbrain. These results indicate that scMix can capture gene–cell relationships, enabling the identification of additional, unknown gene–cell pairs and supporting downstream biological analyses.

3.4 Ablation study

3.4.1 Comparison of multi- and single-time-point input

A key feature of our model is its use of multiple past time points as input. To evaluate its benefit, we conducted an ablation experiment comparing two input settings: (i) using the most recent four observed time points before the target and (ii) using only the most recent single time point.

We report the results as barplots of Wasserstein distance across five runs on *Zebrafish* dataset, covering *Random Recovery* and *Forecasting* tasks. As shown in [Fig. 7a](#), the multi-time point setting outperforms the single time point setting on the two tasks.

3.4.2 Comparison of with and without trend regularization

To evaluate the contribution of *trend regularization*, we conducted an ablation study on *Zebrafish* dataset for *Random Recovery* task, comparing performance with and without *trend regularization*. As illustrated in [Fig. 7b](#), the UMAP visualizations highlight that predictions without *trend regularization* exhibit noticeable discontinuities, whereas the inclusion of *trend regularization* yields predictions that are more continuous and better aligned with the target distribution. [Figure 7b](#) also shows that in several local regions, the predictions with *trend regularization* better capture fine-grained structures.

4 Discussion

In this work, we proposed scMix, a language-model-based framework for multi-time point temporal gene expression prediction. Unlike existing approaches that rely on SDEs or neural ODEs, our model innovatively leverages language models, thus avoiding their inherent limitations. scMix captures the information decay observed during cell development and incorporates a Δt mechanism that enables it to perceive temporal intervals and skip unknown time points. In addition, we introduced trend regularization to enforce smooth transitions across time points and enhance the temporal coherence of predictions.

The results indicate that across all four datasets, scMix achieves lower Wasserstein distances and exhibits greater stability compared with baselines. Moreover, the PAGA results show that the distributions generated by our model in the trajectory recovery task are more consistent with the target distributions, which is further quantitatively validated by the IM values. The perturbation experiments confirm that our model effectively captures the gene–cell relationships. In addition, the ablation experiments demonstrate that incorporating multiple time points substantially reduces the Wasserstein distance of predictions, while the trend regularization contributes to producing more continuous results.

Despite these encouraging results, scMix exhibits a specific limitation. The model is explicitly designed to emphasize temporal continuity in gene expression dynamics, and the trend regularization further enforces smooth transitions across time points. As a result, scMix is particularly well suited for scenarios where cellular states evolve gradually under natural conditions. However, when cells are exposed to strong stimuli, gene

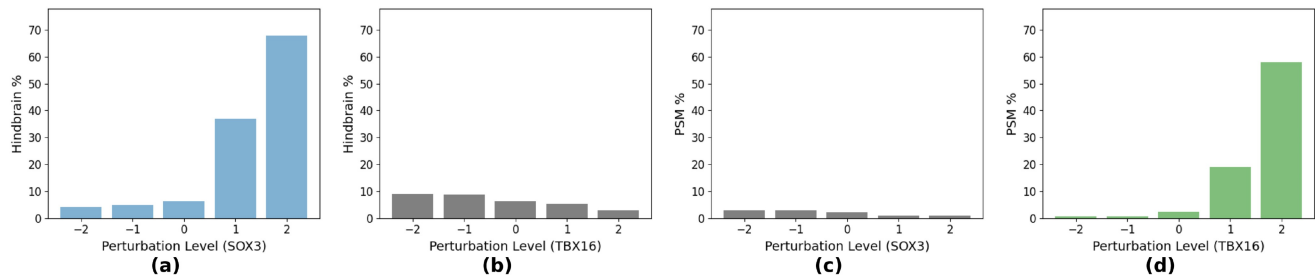


Figure 6 Perturbation experiments illustrating changes in cell proportions after gene perturbations, to examine whether the model captures gene-cell interactions. (a) *Hindbrain* percentage after *SOX3* perturbation. (b) *Hindbrain* percentage after *TBX16* perturbation. (c) *PSM* percentage after *SOX3* perturbation. (d) *PSM* percentage after *TBX16* perturbation. The horizontal axis indicates perturbation strength (10^{-2} , 10^{-1} , 10^0 , 10^1 , 10^2), with 10^0 representing no perturbation, and the vertical axis represents the proportion of the corresponding cell type.

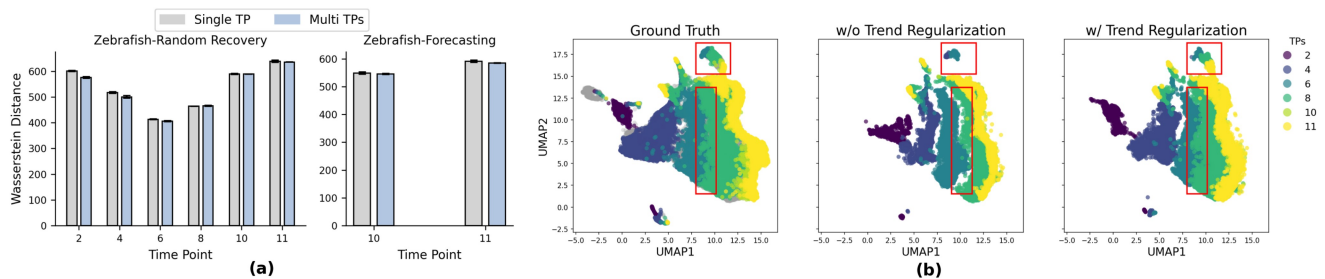


Figure 7 Ablation studies. (a) Ablation study comparing multi-time point and single-time point settings, illustrated by Wasserstein distances. Results on *Zebrafish* dataset with *Random Recovery* task (left) and *Forecasting* task (right). (b) Ablation study of trend regularization. UMAP visualizations are shown for the ground truth (left), the result without trend regularization (middle), and the result with trend regularization (right).

expression may exhibit abrupt changes that deviate from smooth temporal evolution, and in such cases, the model may not achieve reliable predictions.

In future work, cell-level state information such as cell cycle or proliferation status could be considered to provide additional context for temporal gene expression dynamics. These factors are known to influence the pace and pattern of expression changes over time, particularly across longer time spans. This provides promising directions for future research on single-cell temporal modeling.

Author contributions

Shangjin Han (Conceptualization [lead], Data curation [lead], Formal analysis [lead], Investigation [lead], Methodology [lead], Software [lead], Validation [lead], Visualization [lead], Writing—original draft [lead], Writing—review & editing [supporting]) and Dongsup Kim (Funding acquisition [lead], Project administration [lead], Supervision [lead], Writing—review & editing [lead])

Supplementary material

[Supplementary material](#) is available at *Bioinformatics* online.

Conflict of interests

No competing interest is declared.

Funding

This work was supported by the National Research Foundation of Korea grants funded by Korean Government [grant numbers RS-2024-00344154, RS-2024-00399520, RS-2025-00523107].

Data availability

All single-cell RNA-seq datasets used in this study are publicly available. *Zebrafish* dataset can be accessed through the Single Cell Portal (SCP162). *Drosophila* dataset is available from the GEO database (GSE190147). *Schiebinger2019* dataset is available from the Broad Institute WOT tutorial (WOT). *Veres* dataset can be accessed through GEO (GSE114412).

References

- Bergen V, Lange M, Peidli S *et al.* Generalizing RNA velocity to transient cell states through dynamical modeling. *Nat Biotechnol* 2020;**38**:1408–14.
- Bressloff PC. *Stochastic Processes in Cell Biology*. Volume II. Cham, Switzerland: Springer, 2021.
- Calderon D, Blecher-Gonen R, Huang X *et al.* The continuum of drosophila embryonic development at single-cell resolution. *Science* 2022;**377**:eabn5800.
- Chen RT, Rubanova Y, Bettencourt J *et al.* Neural ordinary differential equations. *Adv Neural Inf Process Syst* 2018;**31**:6571–83.

- Dee CT, Hirst CS, Shih Y-H *et al.* Sox3 regulates both neural fate and differentiation in the zebrafish ectoderm. *Dev Biol* 2008; **320**:289–301.
- Farrell JA, Wang Y, Riesenfeld SJ *et al.* Single-cell reconstruction of developmental trajectories during zebrafish embryogenesis. *Science* 2018; **360**:eaar3131.
- Haghverdi L, Büttner M, Wolf FA *et al.* Diffusion pseudotime robustly reconstructs lineage branching. *Nat Methods* 2016; **13**:845–8.
- Huguet G, Magruder DS, Tong A *et al.* Manifold interpolating optimal-transport flows for trajectory inference. *Adv Neural Inf Process Syst* 2022; **35**:29705–18.
- Jiang Q, Wan L. A physics-informed neural SDE network for learning cellular dynamics from time-series scRNA-seq data. *Bioinformatics* 2024; **40**:ii120–ii127.
- Kingma DP, Welling M. Auto-encoding variational bayes. In: *International Conference on Learning Representations (ICLR)*, 2014.
- Klein AM, Mazutis L, Akartuna I *et al.* Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell* 2015; **161**:1187–201.
- La Manno G, Soldatov R, Zeisel A *et al.* RNA velocity of single cells. *Nature* 2018; **560**:494–8.
- Lähnemann D, Köster J, Szczurek E *et al.* Eleven grand challenges in single-cell data science. *Genome Biol* 2020; **21**:31.
- McInnes L, Healy J, Melville J. Umap: Uniform manifold approximation and projection for dimension reduction. *J Open Source Softw* 2018; **3**:861.
- Peng B, Alcaide E, Anthony Q *et al.* Rwkv: Reinventing rnns for the transformer era. arXiv, arXiv: 2305.13048, 2023, preprint: not peer reviewed.
- Peyré G, Cuturi M. Computational optimal transport: with applications to data science. *Found Trends Mach Learn* 2019; **11**:355–607.
- Qiu X, Mao Q, Tang Y *et al.* Reversed graph embedding resolves complex single-cell trajectories. *Nat Methods* 2017; **14**:979–82.
- Qiu X, Zhang Y, Martin-Rufino JD *et al.* Mapping transcriptomic vector fields of single cells. *Cell* 2022; **185**:690–711.e45.
- Raj A, Van Oudenaarden A. Nature, nurture, or chance: stochastic gene expression and its consequences. *Cell* 2008; **135**:216–26.
- Saelens W, Cannoodt R, Todorov H *et al.* A comparison of single-cell trajectory inference methods. *Nat Biotechnol* 2019; **37**:547–54.
- Schiebinger G, Shu J, Tabaka M *et al.* Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell* 2019; **176**:928–43.e22.
- Street K, Risso D, Fletcher RB *et al.* Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics* 2018; **19**:477.
- Trapnell C, Cacchiarelli D, Grimsby J *et al.* The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat Biotechnol* 2014; **32**:381–6.
- Treutlein B, Lee QY, Camp JG *et al.* Dissecting direct reprogramming from fibroblast to neuron using single-cell rna-seq. *Nature* 2016; **534**:391–5.
- Veres A, Faust AL, Bushnell HL *et al.* Charting cellular identity during human in vitro β -cell differentiation. *Nature* 2019; **569**:368–73.
- Wagner DE, Klein AM. Lineage tracing meets single-cell omics: opportunities and challenges. *Nat Rev Genet* 2020; **21**:410–27.
- Warga RM, Mueller RL, Ho RK *et al.* Zebrafish tbx16 regulates intermediate mesoderm cell fate by attenuating FGF activity. *Dev Biol* 2013; **383**:75–89.
- Weinreb C, Rodriguez-Fraticelli A, Camargo FD *et al.* Lineage tracing on transcriptional landscapes links state to fate during differentiation. *Science* 2020; **367**:eaaw3381.
- Wolf FA, Hamey FK, Plass M *et al.* Paga: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biol* 2019; **20**:59.
- Yeo GHT, Saksena SD, Gifford DK. Generative modeling of single-cell time series with prescient enables prediction of cell trajectories with interventions. *Nat Commun* 2021; **12**:3222.
- Zhang J, Larschan E, Bigness J *et al.* Scnode: generative model for temporal single cell transcriptomic data prediction. *Bioinformatics* 2024; **40**:ii146–ii154.