# scChrono: Learning a Transcriptomic Virtual Embryo through Multi-Rate Developmental Dynamics

**Anonymous authors**
Paper under double-blind review

## Abstract

Virtual Embryo aims to construct a data-backed digital twin of embryogenesis that computationally describes continuous development. At the transcriptomic level, this requires recovering continuous developmental dynamics from discrete single-cell observations. However, transcriptional changes occur at different temporal rates, with sustained developmental progression coexisting with more localized state changes, making the complete developmental process difficult to capture with a single dynamical process. We introduce scChrono, a neural stochastic differential equation (SDE) for learning multi-rate developmental dynamics and constructing a transcriptomic virtual embryo. scChrono decomposes deterministic dynamics into Persistent Dynamics for sustained progression and Fast-Transition Dynamics for temporally gated corrections, while stochastic diffusion models population-level heterogeneity. scChrono reconstructs intermediate states, completes unobserved developmental intervals, forecasts unseen future transcriptomic distributions, and transfers learned developmental representations to downstream Cell Fate Prediction. Across the two primary forecasting settings on the Mouse Developmental Cell and Lineage Atlas (mdCLA), scChrono reduces normalized mean absolute error (NMAE) by 14.1–16.0% and Co-expression Structure Score (CSS) by 4.5% in highly variable gene (HVG) space, and Sinkhorn divergence by 16.1–21.9% in principal component analysis (PCA) space, relative to the strongest baseline for each metric, while achieving the best performance on all six metrics in both temporal completion settings. On lineage-resolved LARRY, transfer improves dominant top-1 fate accuracy by 6.2% over training from scratch. These results support multi-rate developmental dynamics as a framework for building continuous transcriptomic virtual embryos from cross-sectional single-cell atlases. Code is available at `https://anonymous.4open.science/r/scChrono-54CC/`.

## 1 Introduction

Embryogenesis is one of the most fundamental and complex biological processes underlying the formation of life. Cells continuously divide and differentiate from a fertilized egg, progressively giving rise to diverse tissues and organs (Pijuan-Sala et al., 2019; Qiu et al., 2022a; 2024). Deviations from normal developmental trajectories can lead to severe birth defects, affecting approximately 8 million infants worldwide each year (World Health Organization, 2026). Understanding continuous embryonic development is therefore important for studying normal organ formation and developmental abnormalities, yet its complete dynamics are difficult to observe directly and continuously in experiments. In this context, the Virtual Embryo has been proposed as a computational model of embryonic development and defined as a data-backed digital twin of embryogenesis, aiming to computationally reconstruct continuous developmental processes (Cao et al., 2026a).

At the transcriptomic level, a fundamental requirement for constructing a transcriptomic virtual embryo is to recover continuous population-level developmental dynamics from discrete cross-sectional single-cell atlases. Single-cell RNA sequencing resolves embryonic states at cellular resolution but destroys assayed cells, preventing the same cell from being continuously tracked through development (Chen et al., 2022). Existing developmental atlases therefore consist primarily



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

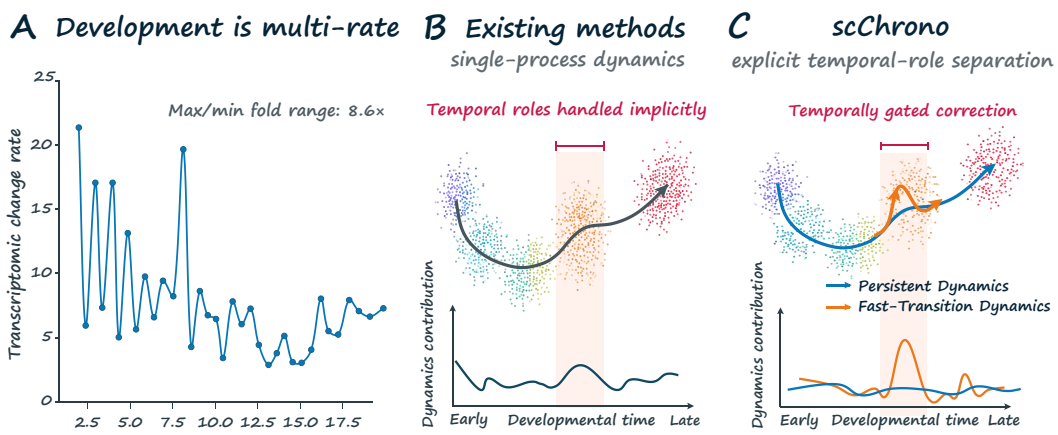


Figure 1: Motivation for multi-rate developmental dynamics. (A) Heterogeneous transcriptomic change rates across development. (B) Existing methods use a unified dynamics process. (C) scChrono explicitly separates Persistent Dynamics and Fast-Transition Dynamics.

of cross-sectional snapshots collected at discrete developmental times, leaving the state evolution between them unobserved (Pijuan-Sala et al., 2019). Optimal Transport, ordinary differential equation (ODE)/continuous normalizing flow (CNF), stochastic differential equation (SDE), and diffusion-based models recover continuous cell-state dynamics from these discrete snapshots (Klein et al., 2025; Tong et al., 2020; Yeo et al., 2021; Zhang et al., 2025), providing a technical basis for transcriptomic virtual embryos.

Constructing a transcriptomic virtual embryo across the complete developmental course poses two challenges. First, transcriptomic change rates vary substantially across developmental intervals. In mdCLA, lineage-aware comparisons of adjacent stages still span an 8.6-fold range after reducing the influence of changing cell composition, as shown in Figure 1A. This is especially consequential for long-horizon forecasting, where local dynamics errors can accumulate through repeated rollout (Sanchez-Gonzalez et al., 2020). Second, a virtual embryo must connect the complete whole-embryo developmental timeline within one dynamical model, recovering missing intermediate stages and forecasting future states unseen during training.

Existing methods model single-cell dynamics from complementary perspectives, including cross-time coupling, continuous trajectories, stochastic heterogeneity, and complex distribution generation. However, developmental changes at different temporal scales are typically handled implicitly by a unified dynamical or generative process, without explicit separation of temporal roles, as illustrated in Figure 1B. Moreover, most methods are validated over local developmental intervals, specific tissues, or lineage systems, rather than using one unified model across the complete whole-embryo developmental timeline. This raises a central question: how can multi-rate developmental dynamics be learned within a unified continuous-time model while supporting stable whole-embryo long-horizon forecasting?

To address these challenges, we propose scChrono, a neural SDE for learning multi-rate developmental dynamics and constructing a transcriptomic virtual embryo. scChrono decomposes deterministic dynamics into Persistent Dynamics for sustained developmental progression and Fast-Transition Dynamics for temporally gated corrections, while stochastic diffusion models population-level heterogeneity. Figure 1C illustrates this decomposition. The resulting dynamics span the complete mouse embryonic timeline and support intermediate-state reconstruction, temporal completion, and forecasting of unseen future transcriptomic states within a unified model.

On the Mouse Developmental Cell and Lineage Atlas (mdCLA) (Cao et al., 2026b), scChrono achieves the best result in 11 of 12 metric comparisons across the two primary forecasting settings and in all 12 comparisons across the two temporal completion settings. Ablation results show that Fast-Transition Dynamics contributes more clearly under Strict Long-Horizon Extrapolation; transferring the learned developmental representation to lineage-resolved LARRY (Weinreb et al., 2020) further improves dominant top-1 fate accuracy by 6.2% over training from scratch.

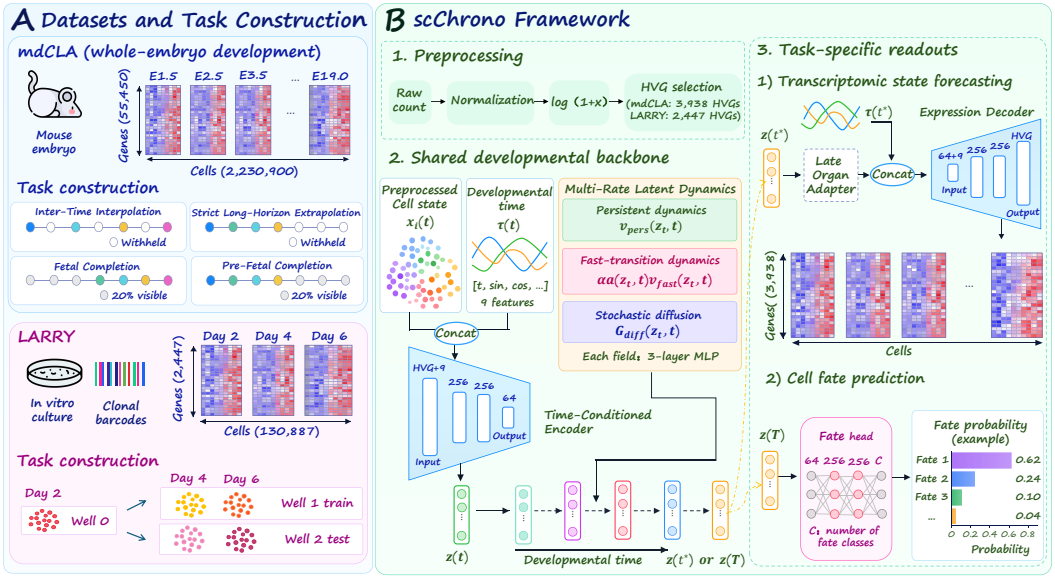


Figure 2: Overview of scChrono. (A) Datasets and task construction. (B) Shared developmental backbone and task-specific readouts.

Our main contributions are as follows:

**Multi-rate developmental dynamics.** We formulate temporal-scale differences across developmental intervals as a modeling problem for whole-embryo developmental dynamics and explicitly separate Persistent Dynamics from Fast-Transition Dynamics within a unified continuous-time model to capture sustained developmental progression and temporally gated corrections.

**Transcriptomic virtual embryo.** scChrono learns unified continuous developmental dynamics spanning the complete mouse embryonic timeline, supporting intermediate-state reconstruction, temporal completion, and unseen-future forecasting, thereby transforming a discrete whole-embryo single-cell atlas into a continuous transcriptomic virtual embryo.

**Strong forecasting performance and transferability.** scChrono reduces $\mathrm{NMAE}_{\mathrm{HVG}}$ by 14.1–16.0%, $\mathrm{CSS}_{\mathrm{HVG}}$ by 4.5%, and $\mathrm{Sinkhorn}_{\mathrm{PCA}}$ by 16.1–21.9% across the two primary forecasting settings. Transfer to LARRY further improves dominant top-1 fate accuracy by 6.2%.

## 2 Related Work

### 2.1 Transcriptomic State Forecasting from Cross-Sectional Snapshots

Methods for forecasting transcriptomic dynamics from cross-sectional snapshots span four main paradigms. Optimal Transport (OT) methods, including Waddington-OT (Schiebinger et al., 2019) and moscot (Klein et al., 2025), couple cell populations across discrete observation times. ODE models such as TrajectoryNet (Tong et al., 2020) and scNODE (Zhang et al., 2024) represent continuous deterministic state evolution, while SDE models such as PRESCIENT (Yeo et al., 2021) and scDiffEq (Vinyard et al., 2025) additionally model stochastic heterogeneity. Diffusion models such as scDiffusion (Luo et al., 2024) and cfDiffusion (Zhang et al., 2025) learn conditional cell-state distributions through diffusion-based generative processes. scChrono is closest to the SDE family, but further decomposes deterministic drift into persistent and fast-transition dynamics and fits a unified model across the whole-embryo timeline.

### 2.2 Cell Fate Prediction and Lineage-Resolved Modeling

Cell Fate Prediction estimates the probabilities that early cells reach different terminal fates. RNA velocity, scVelo (Bergen et al., 2020), and dynamo use transcriptional kinetics to estimate local

state-change directions (La Manno et al., 2018; Qiu et al., 2022b), and CellRank combines such directionality with Markov transitions for terminal-state inference (Lange et al., 2022). Lineage-resolved methods instead use clonal relationships: CoSpar reconstructs state transitions from transcriptomic and lineage information (Wang et al., 2022), while LARRY directly links early cells to terminal descendants through expressed clonal barcodes (Weinreb et al., 2020; Wagner & Klein, 2020). Stochastic landscape and neural-SDE models can also connect simulated state evolution with fate outcomes (Yeo et al., 2021; Vinyard et al., 2025). In scChrono, fate prediction is used as a transfer evaluation: the whole-embryo developmental backbone is learned from cross-sectional data and then adapted to an independent lineage-resolved setting with clone-derived supervision.

# 3 METHOD

## 3.1 PROBLEM SETUP

At developmental time $t$, we denote the cross-sectional transcriptomic snapshot as

$$X_t = \{x_i^{(t)}\}_{i=1}^{n_t} \in \mathbb{R}^{n_t \times G},$$

where $x_i^{(t)}$ is the transcriptomic state of cell $i$. scChrono uses one shared developmental backbone for two tasks. Transcriptomic State Forecasting rolls cells from a source time $t_s$ to a target time $t^\star$ and predicts the target snapshot $\widehat{X}_{t^\star}$ without assuming cell-wise correspondence. Cell Fate Prediction rolls an early cell toward terminal time $T$ and predicts its terminal fate distribution from clone-derived soft fate supervision. Figure 2A summarizes the corresponding mdCLA forecasting settings and LARRY lineage-resolved supervision used by these two tasks.

## 3.2 scCHRONO FRAMEWORK

As shown in Figure 2B, scChrono consists of a shared developmental backbone and task-specific readouts. The backbone combines a time-conditioned encoder with multi-rate latent dynamics; forecasting decodes the rollout endpoint into transcriptomic space, whereas fate prediction maps the terminal latent state to a fate distribution.

## 3.3 SHARED DEVELOPMENTAL BACKBONE

### 3.3.1 TIME-CONDITIONED ENCODING

To represent nonlinear dependence on continuous developmental time, we use Fourier features (Tancik et al., 2020)

$$\tau(t) = \left[t, \{\sin(2^b t)\}_{b=0}^{B-1}, \{\cos(2^b t)\}_{b=0}^{B-1}\right]. \tag{1}$$

With $B = 4$, $\tau(t)$ is a 9-dimensional time encoding. At source time $t_s$, the encoder maps $[x_i^{(t_s)}, \tau(t_s)]$ to a 64-dimensional latent state,

$$z_i(t_s) = E_{\theta_E}([x_i^{(t_s)}, \tau(t_s)]) \in \mathbb{R}^{64}.$$

During rollout, $\tau(t_k)$ conditions the latent dynamics at each integration time. The endpoint is concatenated with $\tau(t^\star)$ and decoded to the $G$-dimensional gene-expression space; the time encoding is auxiliary conditioning and is not part of the 64-dimensional latent state.

### 3.3.2 MULTI-RATE LATENT DYNAMICS

scChrono separates sustained developmental progression, temporally gated directional correction, and stochastic heterogeneity through Persistent Dynamics, Fast-Transition Dynamics, and diffusion:

$$dz_t = v_{\text{pers}}(z_t, t)\, dt + \alpha\, a(z_t, t)\, v_{\text{fast}}(z_t, t)\, dt + G_{\text{diff}}(z_t, t)\, dW_t, \tag{2}$$

where

$$\begin{aligned} v_{\text{pers}}(z, t) &= f_{\text{pers}}(z, \tau(t)), \\ v_{\text{fast}}(z, t) &= f_{\text{fast}}(z, \tau(t)), \\ G_{\text{diff}}(z, t) &= \text{softplus}(g_{\text{raw}}(z, \tau(t))) + 10^{-4}. \end{aligned}$$

Here, $v_{\text{pers}}$ and $v_{\text{fast}}$ are the learned persistent and fast-transition directional fields, respectively. The effective Fast-Transition Dynamics contribution is $\alpha\, a(z,t) v_{\text{fast}}(z,t)$. The term $g_{\text{raw}}$ is the diffusion MLP before the element-wise softplus transform; the additive $10^{-4}$ floor keeps the latent-dimension-wise diffusion scales positive. The scalar $\alpha$ is a fixed fast-transition scale (default 1.0) and is not learned. The fast-transition contribution is modulated by a learned state–time gate and a deterministic temporal prior:

$$a(z,t) = \sigma(h_{\text{gate}}(z, \tau(t)))\, p(t), \tag{3}$$

where $h_{\text{gate}}$ is a two-layer MLP with a scalar output and $p(t)$ is a fixed sigmoid-window prior defined directly over the raw scalar time $t$. The persistent field captures sustained progression, while the fast-transition field provides a temporally gated correction during rollout.

For forward forecasting from $t_s$ to $t^\star$, we integrate Eq. (2) with an Euler–Maruyama update (Higham, 2001):

$$\begin{aligned} z_k^{\text{p}} &= z_k + \Delta t\, v_{\text{pers}}(z_k, t_k), \\ z_k^{\text{f}} &= z_k^{\text{p}} + \Delta t\, \alpha\, a(z_k^{\text{p}}, t_k)\, v_{\text{fast}}(z_k^{\text{p}}, t_k), \\ z_{k+1} &= z_k^{\text{f}} + \sqrt{\Delta t + \varepsilon}\, G_{\text{diff}}(z_k^{\text{f}}, t_k) \odot \xi_k, \qquad \xi_k \sim \mathcal{N}(0, I). \end{aligned} \tag{4}$$

Here $z_k^{\text{p}}$ and $z_k^{\text{f}}$ are temporary within-step states used to apply the persistent, fast-transition, and diffusion updates sequentially.

### 3.4 Task-Specific Readouts

**Transcriptomic State Forecasting.** A source cell is encoded and rolled from $t_s$ to $t^\star$, optionally refined by a lightweight Late Organ Adapter at late developmental stages, and decoded into the predicted transcriptomic state. The Late Organ Adapter (LOA) is a residual, organ-routed refinement of the rollout endpoint that is smoothly gated around E11; adapter routing at inference is determined by model-predicted routing probabilities (Appendix B.3).

**Cell Fate Prediction.** The same encoder and multi-rate latent dynamics roll an early cell from $t_s$ to terminal time $T$. The fate head then operates directly on the 64-dimensional terminal latent state $z_i(T)$, without concatenating the Fourier time encoding, and outputs $\widehat{\pi}_i$. In transfer experiments, the pretrained encoder and dynamics initialize the developmental backbone, while the fate head is initialized independently.

### 3.5 Learning Objectives

**Forecasting Objective.** For Transcriptomic State Forecasting, the objective combines reconstruction, cross-time consistency, maximum mean discrepancy (MMD) distribution matching, and dynamics regularization:

$$\begin{aligned} \mathcal{L}_{\text{TSF}} = \lambda_{\text{rec}}(\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{endpoint}}) + \lambda_{\text{anchor}}\mathcal{L}_{\text{anchor}} + \lambda_{\text{dist}}\mathcal{L}_{\text{MMD}} \\ + \lambda_{\text{reg}}\mathcal{L}_{\text{Jac}} + \lambda_{\text{ft}}\mathcal{L}_{\text{ft}}. \end{aligned}$$

Here, $\mathcal{L}_{\text{endpoint}}$ and $\mathcal{L}_{\text{anchor}}$ use Weak Lineage Anchors (WLA), centroid pairs of the same coarse lineage at adjacent training-visible times, as coarse cross-time supervision (Appendix B.4). $\mathcal{L}_{\text{MMD}}$ is the distribution-matching term, and $\mathcal{L}_{\text{ft}}$ regularizes the magnitude of the effective fast-transition contribution and encourages localized corrections.

**Fate Prediction Objective.** Clone-resolved supervision uses the soft terminal fate composition as the primary target and the dominant fate as auxiliary classification supervision.

## 4 Experiments

### 4.1 Experimental Settings

**Datasets and Preprocessing.** We use mdCLA, containing approximately 2.23 million cells across 37 developmental time points from E1.5 to E19.0, for whole-embryo forecasting (Cao et al., 2026b),

and LARRY for lineage-resolved transfer evaluation using expressed clonal barcodes (Weinreb et al., 2020). Both datasets are library-size normalized to $10^4$ and log-transformed; mdCLA uses a fixed highly variable gene (HVG) input space across temporal splits, while LARRY genes are mapped by name onto the input layer of the transferred encoder. Data accessions and full preprocessing details are provided in Appendix C.1.

**Task Settings.** Transcriptomic State Forecasting uses two primary settings: Inter-Time Interpolation, where an intermediate target stage is fully excluded from training, and Strict Long-Horizon Extrapolation, where later target stages are unseen during training. We additionally evaluate Fetal Completion and Pre-Fetal Completion, which mask $80\%$ of target observations and evaluate recovery of transcriptomic states within partially observed developmental periods. Cell Fate Prediction follows the LARRY protocol used by scDiffEq (Vinyard et al., 2025) and uses clone-derived terminal fate supervision.

**Baselines.** For forecasting, we compare representative Optimal Transport, ODE, SDE, and Diffusion methods under the same frozen splits and evaluator, retaining canonical preprocessing when required. Because cfDiffusion conditions on discrete target stages (Zhang et al., 2025), we evaluate it only in masked completion, where $20\%$ of target-stage observations remain visible. moscot interpolates between OT couplings of training-visible time points and therefore cannot extrapolate to the unseen stages of Strict Long-Horizon Extrapolation. scNODE and cfDiffusion generate unpaired population samples, for which DES is undefined. The released PRESCIENT and scNODE implementations only integrate forward in time and do not support the backward rollout required by Pre-Fetal Completion. PlainODE and PlainSDE model the dynamics directly in the 50-dimensional PCA space with a single drift, without and with diffusion respectively, and no multi-rate decomposition. Unavailable evaluations are reported as N/A. Fate prediction follows the LARRY benchmark design used by scDiffEq (Vinyard et al., 2025).

**Metrics.** We evaluate Transcriptomic State Forecasting from four complementary perspectives. Normalized Mean Absolute Error (NMAE) measures population-mean transcriptomic-state reconstruction error (lower is better); Sinkhorn divergence measures mismatch between predicted and observed cell-population distributions (lower is better) (Feydy et al., 2019); Differential Expression Score (DES) evaluates recovery of true up- and down-regulated genes (higher is better) (Virtual Embryo Challenge, 2026); and Co-expression Structure Score (CSS) measures discrepancy in gene–gene co-expression structure (lower is better) (Virtual Embryo Challenge, 2026). NMAE and Sinkhorn are evaluated in both fixed 50-dimensional PCA and shared HVG spaces, whereas DES and CSS are evaluated only in HVG space. The PCA basis is fitted on training-visible cells and frozen for held-out evaluation. Cell Fate Prediction uses dominant top-1 fate accuracy. Full metric definitions are provided in Appendix C.4.

To assess robustness to design choices, we additionally vary the latent dimension, hidden dimension, rollout steps, anchor weight, and temporal-prior window one factor at a time. Full sensitivity results and task-specific trade-offs are reported in Appendix C.6. We further report practical wall-clock and peak-memory costs together with asymptotic training-time and memory complexity in Appendix C.7.

## 4.2 Transcriptomic State Forecasting

### 4.2.1 Whole-Embryo Forecasting Results.

Figure 3 summarizes the two primary forecasting settings. The largest gains are observed in NMAE and Sinkhorn, which capture transcriptomic-state reconstruction and population-distribution matching, whereas differential-expression recovery remains challenging under strict extrapolation.

In Inter-Time Interpolation, scChrono achieves the best result on all six metrics, indicating recovery of multiple complementary aspects of fully withheld intermediate developmental states. Relative to the strongest competing value for each metric, $\mathrm{Sinkhorn}_{\mathrm{PCA}}$, $\mathrm{NMAE}_{\mathrm{HVG}}$, and $\mathrm{Sinkhorn}_{\mathrm{HVG}}$ improve by **21.9%**, **16.0%**, and **11.8%**, respectively; $\mathrm{NMAE}_{\mathrm{PCA}}$ and $\mathrm{CSS}_{\mathrm{HVG}}$ improve by **3.5%** and **4.5%**, while $\mathrm{DES}_{\mathrm{HVG}}$ increases from $0.251$ to $0.258$. The gains span transcriptomic-state reconstruction, population-distribution matching, and gene-level structure recovery, while the smaller DES gain indicates that stage-specific differential-expression recovery remains more challenging.

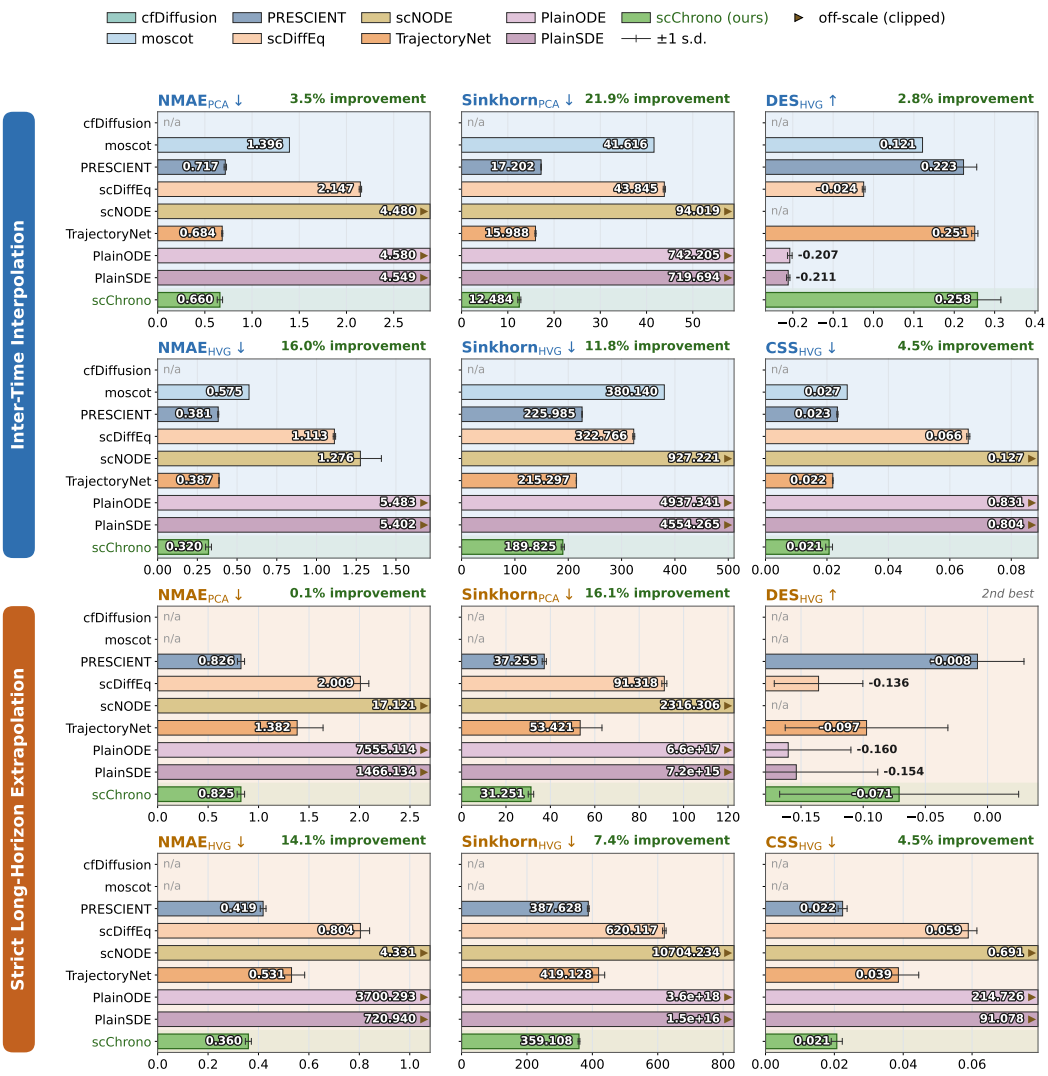


Figure 3: Whole-embryo Transcriptomic State Forecasting on the two primary settings. Bars show mean performance and error bars denote $\pm 1$ SD; triangles indicate off-scale values and N/A denotes unavailable evaluations.

The advantage largely persists under the more challenging Strict Long-Horizon Extrapolation, where future target stages are completely unseen during training: scChrono achieves the best result on five of six metrics. Relative to the strongest competing values, $\mathrm{Sinkhorn}_{\mathrm{PCA}}$, $\mathrm{NMAE}_{\mathrm{HVG}}$, $\mathrm{Sinkhorn}_{\mathrm{HVG}}$, and $\mathrm{CSS}_{\mathrm{HVG}}$ improve by **16.1%**, **14.1%**, **7.4%**, and **4.5%**, respectively; $\mathrm{NMAE}_{\mathrm{PCA}}$ also achieves the best value. The only exception is $\mathrm{DES}_{\mathrm{HVG}}$: PRESCIENT obtains $-0.008 \pm 0.037$, compared with $-0.071 \pm 0.096$ for scChrono. Because DES is low across all evaluated methods in this setting, these results show strong recovery of overall transcriptomic states and cell-population distributions while precise differential-expression forecasting remains challenging under strict extrapolation.

The two masked temporal completion settings provide complementary evidence: scChrono achieves the best result on all six metrics in both Fetal Completion and Pre-Fetal Completion. Full completion results are reported in Appendix Table 7.

Table 1: Component ablation on the two primary forecasting settings. Results are mean ± SD over five seeds. Abbreviations: Fast-Transition Dynamics (FTD), Persistent Dynamics (PD), Late Organ Adapter (LOA), and Weak Lineage Anchors (WLA).

| Variant | $\mathrm{NMAE}_{\mathrm{PCA}} \downarrow$ | $\mathrm{Sinkhorn}_{\mathrm{PCA}} \downarrow$ | $\mathrm{NMAE}_{\mathrm{HVG}} \downarrow$ | $\mathrm{Sinkhorn}_{\mathrm{HVG}} \downarrow$ | $\mathrm{DES}_{\mathrm{HVG}} \uparrow$ | $\mathrm{CSS}_{\mathrm{HVG}} \downarrow$ |
|---|---|---|---|---|---|---|
| *Inter-Time* | | | | | | |
| **Full** | $0.660 \pm 0.028$ | $12.484 \pm 0.332$ | $0.320 \pm 0.018$ | $189.825 \pm 2.951$ | $0.258 \pm 0.058$ | $0.021 \pm 0.001$ |
| −FTD | $0.648 \pm 0.014$ | $12.307 \pm 0.150$ | $0.304 \pm 0.009$ | $188.085 \pm 1.149$ | $0.284 \pm 0.026$ | $0.019 \pm 0.001$ |
| PD only | $1.711 \pm 0.152$ | $110.017 \pm 30.473$ | $1.373 \pm 0.152$ | $580.007 \pm 56.006$ | $0.023 \pm 0.032$ | $0.083 \pm 0.008$ |
| −LOA | $0.658 \pm 0.019$ | $12.468 \pm 0.279$ | $0.314 \pm 0.011$ | $189.966 \pm 1.716$ | $0.264 \pm 0.043$ | $0.020 \pm 0.001$ |
| −WLA | $0.656 \pm 0.009$ | $12.682 \pm 0.132$ | $0.279 \pm 0.005$ | $184.762 \pm 0.828$ | $0.352 \pm 0.032$ | $0.018 \pm 0.000$ |
| *Strict Extrapolation* | | | | | | |
| **Full** | $0.825 \pm 0.037$ | $31.251 \pm 1.211$ | $0.360 \pm 0.012$ | $359.108 \pm 2.585$ | $-0.071 \pm 0.096$ | $0.021 \pm 0.002$ |
| −FTD | $0.908 \pm 0.037$ | $33.451 \pm 1.704$ | $0.444 \pm 0.032$ | $369.541 \pm 8.072$ | $-0.168 \pm 0.079$ | $0.027 \pm 0.002$ |
| PD only | $0.956 \pm 0.052$ | $33.933 \pm 0.850$ | $0.460 \pm 0.062$ | $371.795 \pm 3.915$ | $-0.140 \pm 0.139$ | $0.027 \pm 0.004$ |
| −LOA | $0.861 \pm 0.042$ | $31.824 \pm 1.279$ | $0.366 \pm 0.018$ | $358.691 \pm 4.290$ | $-0.031 \pm 0.128$ | $0.021 \pm 0.002$ |
| −WLA | $0.902 \pm 0.033$ | $39.385 \pm 2.000$ | $0.367 \pm 0.008$ | $383.938 \pm 12.882$ | $0.030 \pm 0.093$ | $0.018 \pm 0.000$ |

#### 4.2.2 Ablation and Model Components

Table 1 summarizes the component ablations, which reveal clear task dependence. The PD-only variant is substantially weaker than Full scChrono, while the effects of FTD, LOA, and WLA vary across forecasting settings.

Notably, the Inter-Time PD-only configuration exhibits numerical instability even under interpolation: $\mathrm{NMAE}_{\mathrm{PCA}} = 1.711$ and $\mathrm{Sinkhorn}_{\mathrm{PCA}} = 110.017$, while its Strict values remain $0.956$ and $33.933$, respectively. These values are not a reporting error: without diffusion or Fast-Transition Dynamics, the drift-only model diverges numerically in this setting, so we do not express its gap as a percentage.

Under Inter-Time Interpolation, removing FTD slightly improves all six metrics, with five of the six differences within one SD of Full scChrono. The pattern reverses under Strict Long-Horizon Extrapolation: without FTD, $\mathrm{NMAE}_{\mathrm{PCA}}$, $\mathrm{Sinkhorn}_{\mathrm{PCA}}$, $\mathrm{NMAE}_{\mathrm{HVG}}$, $\mathrm{Sinkhorn}_{\mathrm{HVG}}$, and $\mathrm{CSS}_{\mathrm{HVG}}$ are worse by approximately **10.1%, 7.0%, 23.3%, 2.9%, and 28.6%**, respectively, and $\mathrm{DES}_{\mathrm{HVG}}$ drops from $-0.071$ to $-0.168$. FTD thus matters little for local interpolation and becomes important once the learned dynamics must be propagated into distant, completely unseen future stages.

LOA and WLA show smaller, more task-dependent effects; detailed trade-offs are reported in Appendix D.3.

Additional model capacity does not explain the strict-forward advantage. A matched-single-drift control with hidden width 406 has 221,334 parameters, close to the 221,569 parameters of Persistent Dynamics, Fast-Transition Dynamics, and the gate together, yet Full scChrono remains better on all six Strict Long-Horizon Extrapolation metrics. The gain therefore comes from the structured decomposition itself, since a single drift with the same parameter budget does not reproduce it; detailed results are reported in Appendix Table 8.

Gate-mechanism controls show that the learned state–time gate and fixed temporal prior play complementary roles: Full scChrono combines both and is best on five Strict Long-Horizon Extrapolation metrics and tied on $\mathrm{CSS}_{\mathrm{HVG}}$, whereas removing either factor or both reduces performance. Detailed comparisons are reported in Appendix Table 9.

A horizon-wise analysis further shows that FTD contributes increasingly with forecast distance. Using independently retrained Full and −FTD models over five seeds, the $\mathrm{NMAE}_{\mathrm{HVG}}$ gap widens from approximately $0.01$ at $\Delta t = 0.5$ to $0.21$ at $\Delta t = 5.0$ and is fully monotonic with forecast horizon, as shown in Figure 4 (Spearman $\rho = 1.000$, exact permutation $p = 5.5 \times 10^{-7}$, the minimum attainable with ten horizons). Other NMAE, Sinkhorn, and CSS metrics show the same positive association, whereas DES is weaker. In contrast, suppressing FTD only at inference in a fixed Full model yields smaller and mixed changes, and the horizon trend reverses for $\mathrm{NMAE}_{\mathrm{HVG}}$, $\mathrm{CSS}_{\mathrm{HVG}}$, and $\mathrm{DES}_{\mathrm{HVG}}$. FTD therefore acts mainly by shaping the dynamics learned during training, not as an instantaneous correction at rollout time. Detailed statistics are reported in Appendix Table 10.

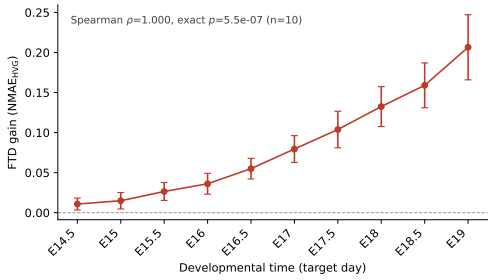


Figure 4: Horizon-dependent effect of Fast-Transition Dynamics under Strict Long-Horizon Extrapolation. Positive values indicate degradation in $\mathrm{NMAE}_{\mathrm{HVG}}$ after retraining without FTD; error bars denote bootstrap uncertainty.

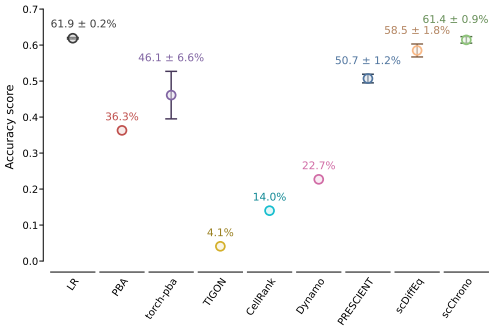


Figure 5: Dominant top-1 fate accuracy on the LARRY dataset. The comparison includes scChrono, competing methods, and the directly supervised linear-regression reference reported by scDiffEq; error bars denote $\pm 1$ SD.

### 4.3 Transfer to Lineage-Resolved Fate Prediction

The whole-embryo developmental representation learned by scChrono transfers to an independent lineage-resolved Cell Fate Prediction task. Following the LARRY task setting, data split, and dominant top-1 fate accuracy used by scDiffEq (Vinyard et al., 2025), Figure 5 compares scChrono with the results reported in the original study, including PBA and TIGON (Weinreb et al., 2018; Sha et al., 2024). scChrono reaches $61.4 \pm 0.9\%$, compared with $58.5 \pm 1.8\%$ for scDiffEq and $50.7 \pm 1.2\%$ for PRESCIENT (Yeo et al., 2021), and comes within 0.5 percentage points of the directly supervised linear-regression reference reported by scDiffEq ($61.9 \pm 0.2\%$).

To isolate the contribution of developmental pretraining, Appendix Table 12 compares the same downstream architecture initialized either from the mdCLA-pretrained shared developmental backbone (Transfer) or from random initialization (Scratch). Transfer improves dominant top-1 fate accuracy from $0.578 \pm 0.013$ to $0.614 \pm 0.009$, a 6.2% relative improvement. Because the downstream architecture and training setting are unchanged, this result isolates the contribution of the transferred representation. Together, the benchmark and Transfer–Scratch comparison show that developmental dynamics learned from whole-embryo cross-sectional data can support an independent fate-prediction setting with lineage-derived supervision.

## 5 Conclusion

We introduced scChrono, a neural SDE for learning multi-rate developmental dynamics and constructing a transcriptomic virtual embryo from whole-embryo cross-sectional transcriptomic snapshots. By combining Persistent Dynamics, Fast-Transition Dynamics, and stochastic diffusion, scChrono supports interpolation, temporal completion, and unseen-future forecasting across the developmental timeline. Experiments on mdCLA show consistent improvements over representative baselines. Under Strict Long-Horizon Extrapolation, capacity-matched controls show that the gain is not explained by parameter count, gate-mechanism controls support complementary roles for the learned state–time gate and temporal prior, and the gap between Full and $-$FTD widens with forecast horizon across the principal reconstruction and distributional metrics. The learned developmental backbone further transfers to lineage-resolved Cell Fate Prediction on LARRY. Together, these results support structured multi-rate dynamics as a unified continuous model for transforming discrete whole-embryo single-cell atlases into a predictive transcriptomic virtual embryo. Future work will integrate spatial, perturbational, and multimodal measurements to extend this framework toward richer models of normal and abnormal development.

## AI USE STATEMENT

We used generative AI tools (GPT and Claude) in the implementation and analysis of this work. Specifically, they assisted in writing and modifying experimental code, including model implementation, baseline adaptation, and evaluation scripts, and in checking results, including tracing reported numbers to their data sources and running statistical tests. All AI-assisted code and analyses were reviewed and tested by the authors. In addition, generative AI tools were used to polish the language and improve the readability of the manuscript. We take full responsibility for the final content of this work.

## REFERENCES

Volker Bergen, Marius Lange, Stefan Peidli, F Alexander Wolf, and Fabian J Theis. Generalizing rna velocity to transient cell states through dynamical modeling. *Nature biotechnology*, 38(12): 1408–1414, 2020.

Natalie Cao, Yifan Lu, and Xiaojie Qiu. Towards predictive virtual embryos with genomics and ai. *Nature Methods*, 23(9):1666–1670, 2026a.

Shangtao Cao, Lihui Lin, Huijian Feng, Pan Chen, Guangming Wu, Baomei Cai, Mengjie Pan, Qingmei Shen, Huan Chen, Xuzhao Zhai, et al. Developmental chronology of mouse embryo from 2-cell stage through birth. *Nature Cell Biology*, 28(7):1545–1555, 2026b.

Wanze Chen, Orane Guillaume-Gentil, Pernille Yde Rainer, Christoph G Gäbelein, Wouter Saelens, Vincent Gardeux, Amanda Klaeger, Riccardo Dainese, Magda Zachara, Tomaso Zambelli, et al. Live-seq enables temporal transcriptomic recording of single cells. *Nature*, 608(7924):733–740, 2022.

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd international conference on artificial intelligence and statistics*, pp. 2681–2690. PMLR, 2019.

Chris Finlay, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam Oberman. How to train your neural ode: the world of jacobian and kinetic regularization. In *International conference on machine learning*, pp. 3154–3164. PMLR, 2020.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The journal of machine learning research*, 13:723–773, 2012.

Desmond J Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM review*, 43(3):525–546, 2001.

Dominik Klein, Giovanni Palla, Marius Lange, Michal Klein, Zoe Piran, Manuel Gander, Laetitia Meng-Papaxanthos, Michael Sterr, Lama Saber, Changying Jing, et al. Mapping cells through time and space with moscot. *Nature*, 638(8052):1065–1075, 2025.

Gioele La Manno, Ruslan Soldatov, Amit Zeisel, Emelie Braun, Hannah Hochgerner, Viktor Petukhov, Katja Lidschreiber, Maria E Kastriti, Peter Lönnerberg, Alessandro Furlan, et al. Rna velocity of single cells. *Nature*, 560(7719):494–498, 2018.

Marius Lange, Volker Bergen, Michal Klein, Manu Setty, Bernhard Reuter, Mostafa Bakhti, Heiko Lickert, Meshal Ansari, Janine Schniering, Herbert B Schiller, et al. Cellrank for directed single-cell fate mapping. *Nature methods*, 19(2):159–170, 2022.

Erpai Luo, Minsheng Hao, Lei Wei, and Xuegong Zhang. scdiffusion: conditional generation of high-quality single-cell data using diffusion model. *Bioinformatics*, 40(9):btae518, 2024.

Blanca Pijuan-Sala, Jonathan A Griffiths, Carolina Guibentif, Tom W Hiscock, Wajid Jawaid, Fernando J Calero-Nieto, Carla Mulas, Ximena Ibarra-Soria, Richard CV Tyser, Debbie Lee Lian Ho, et al. A single-cell molecular map of mouse gastrulation and early organogenesis. *Nature*, 566(7745):490–495, 2019.

Chengxiang Qiu, Junyue Cao, Beth K Martin, Tony Li, Ian C Welsh, Sanjay Srivatsan, Xingfan Huang, Diego Calderon, William Stafford Noble, Christine M Disteche, et al. Systematic reconstruction of cellular trajectories across mouse embryogenesis. *Nature genetics*, 54(3):328–341, 2022a.

Chengxiang Qiu, Beth K Martin, Ian C Welsh, Riza M Daza, Truc-Mai Le, Xingfan Huang, Eva K Nichols, Megan L Taylor, Olivia Fulton, Diana R O'Day, et al. A single-cell time-lapse of mouse prenatal development from gastrula to birth. *Nature*, 626(8001):1084–1093, 2024.

Xiaojie Qiu, Yan Zhang, Jorge D Martin-Rufino, Chen Weng, Shayan Hosseinzadeh, Dian Yang, Angela N Pogson, Marco Y Hein, Kyung Hoi Joseph Min, Li Wang, et al. Mapping transcriptomic vector fields of single cells. *Cell*, 185(4):690–711, 2022b.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International conference on machine learning*, pp. 8459–8468. Pmlr, 2020.

Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, et al. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943, 2019.

Yutong Sha, Yuchi Qiu, Peijie Zhou, and Qing Nie. Reconstructing growth and dynamic trajectories from single-cell transcriptomics data. *Nature Machine Intelligence*, 6(1):25–39, 2024.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

Alexander Tong, Jessie Huang, Guy Wolf, David Van Dijk, and Smita Krishnaswamy. Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics. In *International conference on machine learning*, pp. 9526–9536. PMLR, 2020.

Michael E Vinyard, Anders W Rasmussen, Ruitong Li, Allon M Klein, Gad Getz, and Luca Pinello. Learning cell dynamics with neural differential equations. *Nature Machine Intelligence*, 7(12): 1969–1984, 2025.

Virtual Embryo Challenge. Evaluation. `https://virtualembryo.ai/challenge/evaluation`, 2026. Accessed: 2026-09-22.

Daniel E Wagner and Allon M Klein. Lineage tracing meets single-cell omics: opportunities and challenges. *Nature Reviews Genetics*, 21(7):410–427, 2020.

Shou-Wen Wang, Michael J Herriges, Kilian Hurley, Darrell N Kotton, and Allon M Klein. Cospar identifies early cell fate biases from single-cell transcriptomic and lineage information. *Nature Biotechnology*, 40(7):1066–1074, 2022.

Caleb Weinreb, Samuel Wolock, Betsabeh K Tusi, Merav Socolovsky, and Allon M Klein. Fundamental limits on dynamic inference from single-cell snapshots. *Proceedings of the National Academy of Sciences*, 115(10):E2467–E2476, 2018.

Caleb Weinreb, Alejo Rodriguez-Fraticelli, Fernando D Camargo, and Allon M Klein. Lineage tracing on transcriptional landscapes links state to fate during differentiation. *Science*, 367(6479): eaaw3381, 2020.

World Health Organization. WHO urges scale up of newborn screening to improve early detection and care of birth defects. `https://www.who.int/news/item/23-06-2026-who-urges-scale-up-of-newborn-screening-to-improve-early-detection-and-care-of-birth-defects`, June 2026. News release, accessed 16 September 2026.

Grace Hui Ting Yeo, Sachit D Saksena, and David K Gifford. Generative modeling of single-cell time series with prescient enables prediction of cell trajectories with interventions. *Nature communications*, 12(1):3222, 2021.

Jiaqi Zhang, Erica Larschan, Jeremy Bigness, and Ritambhara Singh. scnode: generative model for temporal single cell transcriptomic data prediction. *Bioinformatics*, 40(Supplement_2):ii146–ii154, 2024.

Tianjiao Zhang, Zhongqian Zhao, Jixiang Ren, Ziheng Zhang, Hongfei Zhang, and Guohua Wang. cfdiffusion: diffusion-based efficient generation of high quality scrna-seq data with classifier-free guidance. *Briefings in Bioinformatics*, 26(1):bbaf071, 2025.

# A Supplementary Motivation Analysis

## A.1 Empirical Transcriptomic Change-Rate Analysis

To quantify temporal heterogeneity while reducing the influence of changing whole-embryo cell composition, we perform a lineage-aware analysis using the six broad `level1_group` annotations in mdCLA: Mesenchymal, Neural, Hematopoietic, Epithelial, Endothelial, and Early embryo. For each developmental time $t$ and broad lineage $l$, we average the preprocessed expression profiles of cells in that group to obtain a lineage-specific pseudobulk vector $\mu_{t,l}$. For lineages observed at both endpoints of a common 0.5-day interval, we define

$$r_{t,l} = \frac{\|\mu_{t+0.5,l} - \mu_{t,l}\|_2}{0.5}.$$

The interval-level empirical change rate is then summarized by the median across comparable lineages,

$$r_t = \mathrm{median}_{l \in \mathcal{L}_t}\, r_{t,l},$$

where $\mathcal{L}_t$ denotes the broad lineages observed at both endpoints of interval $t$.

The original mdCLA sampling contains two 0.25-day sub-intervals between E7.5 and E8.0. To keep a common 0.5-day duration in the main analysis, we use the E7.5 and E8.0 endpoints directly and recompute the lineage-specific displacements over E7.5–E8.0, rather than averaging or summing the two 0.25-day rates. The resulting lineage-aware empirical change rates span an 8.6-fold max/min range across development, showing that substantial temporal heterogeneity remains after accounting for broad lineage composition. This quantity is used to characterize heterogeneity in observed transcriptomic state changes and is not interpreted as a direct measurement of biological developmental speed.

# B Supplementary Method Details

## B.1 Network Architecture and Time Encoding

The full forecasting configuration uses latent dimension 64 and hidden dimension 256. Fourier time encoding uses $B = 4$ frequency bands, so Eq. (1) produces a 9-dimensional vector $\tau(t)$: one raw time coordinate and four sine–cosine pairs. The time encoding is auxiliary conditioning information and is not included in the 64-dimensional latent-state definition.

The encoder and forecasting decoder each contain three linear layers with two hidden layers of width 256, using SiLU activations and dropout $0.1$. Their dimensions are $(G+9) \to 256 \to 256 \to 64$ for the encoder and $73 \to 256 \to 256 \to G$ for the decoder. Thus, source expression is concatenated with $\tau(t_s)$ before encoding, whereas the 64-dimensional rollout endpoint is concatenated with $\tau(t^\star)$ before decoding. If the Late Organ Adapter is used, the decoder receives the adapter-refined 64-dimensional endpoint state before this target-time concatenation. Time encoding is injected at the network input rather than repeatedly at hidden layers.

The persistent field, fast-transition field, and diffusion network each receive the 73-dimensional concatenation $[z, \tau(t)]$ and use three linear layers with dimensions $73 \to 256 \to 256 \to 64$. Time encoding is concatenated once at the input of each dynamics network and is not re-injected into hidden layers. The diffusion MLP output is transformed element-wise by $\mathrm{softplus}(\cdot) + 10^{-4}$ before being used as the diffusion scale. The state–time gate receives the same 73-dimensional input and uses a two-layer MLP, $73 \to 256 \to 1$, followed by a sigmoid to produce a scalar learned gate. This learned scalar is then multiplied by the fixed temporal prior $p(t)$ defined over the raw scalar time. The evolving latent state itself remains 64-dimensional throughout rollout. We use $z_i(t)$ for the latent state of cell $i$ at developmental time $t$; $z_{i,k}$ denotes its numerical approximation at integration time $t_k$, and $\bar{z}_i(t^\star)$ denotes the endpoint after Late Organ Adapter refinement.

## B.2 Temporal Prior and Numerical Rollout

**Temporal prior.** The temporal prior in Eq. (3) provides smooth temporal modulation for the learned state–time gate. It is a deterministic function of the raw scalar developmental time $t$ and

does not use the Fourier encoding $\tau(t)$:

$$p(t) = \rho + (1-\rho)\sigma(\kappa(t - t_{\text{on}}))\,\sigma(\kappa(t_{\text{off}} - t))\,.$$

The configuration used in our experiments is

$$t_{\text{on}} = 7.5, \qquad t_{\text{off}} = 16.5, \qquad \rho = 0.05, \qquad \kappa = 8.$$

$t_{\text{on}}$ and $t_{\text{off}}$ define the main active range of the prior, $\kappa$ controls the smoothness of the boundaries, and $\rho$ sets the baseline response outside the interval. These parameters define the temporal modulation applied to Fast-Transition Dynamics.

**Numerical rollout.** Eq. (4) gives the Euler–Maruyama discretization for forward rollout. The experiments use $K = 16$ numerical steps and $\varepsilon = 10^{-8}$. At each integration step, $\xi_{i,k} \sim \mathcal{N}(0, I)$ is sampled independently; setting $\xi_{i,k} = 0$ yields deterministic rollout.

The primary forecasting settings use $t^\star > t_s$ and therefore $\Delta t > 0$. Masked temporal completion also supports $\Delta t < 0$ through a signed-step extension: the persistent update uses signed $\Delta t$ to determine numerical direction, while the fast-transition and diffusion magnitudes use $|\Delta t|$ and $\sqrt{|\Delta t| + \varepsilon}$, respectively.

### B.3 Task-Specific Readout Details

**Transcriptomic State Forecasting.** Given a source cell $x_i^{(t_s)}$, the shared developmental backbone first rolls it to the target time:

$$z_i(t^\star) = \Phi_{\theta_\Phi}\Big(t_s, t^\star; E_{\theta_E}([x_i^{(t_s)}, \tau(t_s)])\Big)\,.$$

The Late Organ Adapter routing probability is predicted from the endpoint:

$$r(z,t) = \text{softmax}(C_\omega(z, \tau(t)))\,,$$

and the refined endpoint is

$$\bar{z}_i(t^\star) = z_i(t^\star) + \beta q(t^\star) \sum_{o=1}^{M} r_o(z_i(t^\star), t^\star) A_o(z_i(t^\star), \tau(t^\star))$$

The final transcriptomic prediction is

$$\widehat{x}_i^{(t^\star)} = D_{\theta_D}([\bar{z}_i(t^\star), \tau(t^\star)]),$$

The predicted cells together form $\widehat{X}_{t^\star}$.

**Late Organ Adapter.** Both the routing classifier $C_\omega$ and each per-organ adapter $A_o$ receive the 73-dimensional concatenation $[z, \tau(t)]$. In contrast, the adapter activation strength is controlled separately by a deterministic sigmoid gate defined directly over the raw scalar developmental time $t$:

$$q(t) = \sigma\left(\frac{t - c}{0.25w}\right), \qquad c = 11.0, \quad w = 1.0.$$

The residual scale is fixed at $\beta = 0.25$. The gate $q(t)$ does not use $\tau(t)$; it transitions smoothly from low to high activation over approximately E10.5–E11.5 and retains a high weight at later times. After adapter refinement, the resulting 64-dimensional endpoint $\bar{z}_i(t^\star)$ is concatenated with $\tau(t^\star)$ to form the 73-dimensional input to the forecasting decoder.

Training and evaluation use self-predicted routing probabilities. Organ labels supervise the routing classifier during training, with stop-gradient separating this supervision from the shared developmental backbone.

The Late Organ Adapter uses two-stage training. Stage 1 trains the shared encoder, decoder, and multi-rate dynamics for up to 30 epochs. Stage 2 freezes the main developmental backbone and updates the routing classifier, organ adapters, and decoder; the classifier/adapter learning rate is $5 \times 10^{-4}$ and the decoder learning rate is $5 \times 10^{-5}$. Stage 2 runs for up to 20 epochs with a minimum of 8 epochs, early-stopping patience of 8, and minimum delta of $10^{-5}$.

**Cell Fate Supervision and Readout.** For an early cell $i$, let $c_i$ denote its clonal barcode, $\mathcal{I}(c_i)$ the descendants with the same barcode and terminal fate annotations, and $\mathcal{Y}$ the terminal fate categories. The soft fate target is

$$\pi_{i,y} = \frac{|\{j \in \mathcal{I}(c_i) : y_j = y\}|}{|\mathcal{I}(c_i)|}, \qquad y \in \mathcal{Y}, \tag{5}$$

with $y_i^{\mathrm{dom}} = \arg\max_{y \in \mathcal{Y}} \pi_{i,y}$.

Cell Fate Prediction uses the same backbone to roll each cell to terminal time $T$:

$$z_i(T) = \Phi_{\theta_\Phi}\left(t_s, T; E_{\theta_E}([x_i^{(t_s)}, \tau(t_s)])\right),$$
$$\widehat{\pi}_i = \mathrm{softmax}(h_\phi(z_i(T))).$$

The fate head $h_\phi$ operates directly on the 64-dimensional terminal latent state and does not concatenate $\tau(T)$. Thus, the rollout endpoint branches into two distinct readouts: forecasting concatenates the 9-dimensional target-time encoding before decoding, whereas fate prediction uses the unaugmented 64-dimensional latent state. $\widehat{\pi}_i$ is the predicted terminal fate distribution.

**Cell Fate Transfer.** The transfer model uses a 64-dimensional latent space, hidden dimension 256, four Fourier bands, and eight rollout steps. In all reported LARRY experiments, the organ-transfer auxiliary branch is disabled, so the fate head operates directly on the 64-dimensional terminal latent state without time encoding or organ-logit concatenation. The fate head contains three linear layers with dimensions $64 \to 256 \to 256 \to C$, where $C$ is the number of terminal fate categories. The two hidden layers use SiLU activations followed by dropout $0.1$; the output layer has no activation or dropout before the softmax used to obtain fate probabilities. Training uses AdamW with learning rate $5 \times 10^{-4}$ and weight decay $10^{-5}$. The fate head is first warmed up with the transferred developmental backbone frozen, followed by joint fine-tuning of the transferred encoder, multi-rate latent dynamics, and fate head. Clonal barcodes, terminal descendant expression, and fate labels define the supervision targets in Eq. (5). LARRY uses its own 2,447 HVGs. When the pretrained encoder is transferred, its input layer is remapped by gene name: weight columns of LARRY genes that also appear in the mdCLA HVG set are copied from the pretrained checkpoint, weight columns of unmatched genes are set to zero, and the Fourier time-encoding columns are kept unchanged. LARRY sampling days are used directly as developmental time ($t = 2, 4, 6$) without rescaling.

### B.4 Learning Objective Details

Let $\mathcal{O}$ denote all training-visible cell–time pairs $(x, t)$ in the current split and $\mathcal{A}$ the weak-lineage centroid pairs $(\mu_a, t_a, \mu_b, t_b)$ constructed from training-visible developmental states. All expectations below are estimated by mini-batch empirical means from the corresponding training data loaders. $\mathcal{L}_{\mathrm{Jac}}$ and $\mathcal{L}_{\mathrm{ft}}$ use the encoded latent states from the same mini-batches as reconstruction.

Same-time reconstruction preserves the transcriptomic state through the encoder–decoder:

$$\mathcal{L}_{\mathrm{rec}} = \mathbb{E}_{(x,t)\sim\mathcal{O}} \left\| D_{\theta_D}([E_{\theta_E}([x, \tau(t)]), \tau(t)]) - x \right\|_2^2.$$

Endpoint reconstruction constrains each decoded cross-time rollout endpoint toward its target centroid:

$$\mathcal{L}_{\mathrm{endpoint}} = \mathbb{E}_{\mathcal{A}} \left\| D_{\theta_D}([\Phi_{\theta_\Phi}(t_a, t_b; E_{\theta_E}([\mu_a, \tau(t_a)])), \tau(t_b)]) - \mu_b \right\|_2^2.$$

The anchor term aligns the rolled source centroid with the encoded target centroid in latent space:

$$\mathcal{L}_{\mathrm{anchor}} = \mathbb{E}_{\mathcal{A}} \left\| \Phi_{\theta_\Phi}(t_a, t_b; E_{\theta_E}([\mu_a, \tau(t_a)])) - E_{\theta_E}([\mu_b, \tau(t_b)]) \right\|_2^2.$$

Population-level distribution agreement is enforced with a RBF maximum mean discrepancy (MMD) (Gretton et al., 2012) between rollout endpoint latents $\{\Phi_{\theta_\Phi}(t_a, t_b; E_{\theta_E}([\mu_a, \tau(t_a)]))\}$ and the corresponding target latents $\{E_{\theta_E}([\mu_b, \tau(t_b)])\}$ within the same anchor mini-batch.

Jacobian regularization (Finlay et al., 2020) promotes local smoothness of the persistent field and is computed on encoded states from $\mathcal{O}$:

$$\mathcal{L}_{\mathrm{Jac}} = \mathbb{E}_{(x,t)\sim\mathcal{O}} \left\| \nabla_z v_{\mathrm{pers}}(z, t) \right\|_F^2 \Big|_{z = E_{\theta_E}([x, \tau(t)])}.$$

Fast-transition regularization controls the gated fast field:

$$\mathcal{L}_{\text{ft}} = \mathbb{E}_{(x,t)\sim\mathcal{O}} \left\| a(z,t) v_{\text{fast}}(z,t) \right\|_2^2, \qquad z = E_{\theta_E}([x, \tau(t)]).$$

This expectation is estimated on the same training mini-batches used for reconstruction.

The forecasting objective uses fixed coefficients:

$$\lambda_{\text{rec}} = 1, \quad \lambda_{\text{anchor}} = 1.5, \quad \lambda_{\text{dist}} = 0.25, \quad \lambda_{\text{reg}} = 5 \times 10^{-4}, \quad \lambda_{\text{ft}} = 5 \times 10^{-4}.$$

$\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{endpoint}}$ share the same $\lambda_{\text{rec}}$. The forecasting objective therefore consists of five weighted terms comprising the six loss components defined above.

**Weak lineage anchors.** For adjacent training-visible developmental states within the same coarse lineage, we compute the mean expression of each developmental time + lineage label group and add the centroid pair $(\mu_a, t_a, \mu_b, t_b)$ to $\mathcal{A}$. These anchors provide coarse cross-time directional supervision and are constructed from training-visible developmental states. Anchor supervision uses a fixed focus weighting: anchors between E7.5 and E16.5 receive weight 2 and all others receive weight 1; this weighting applies to $\mathcal{L}_{\text{anchor}}$.

**Cell Fate Prediction Objective and Weights.** The soft fate distribution is the primary supervision signal and dominant fate is an auxiliary classification target:

$$\mathcal{L}_{\text{fate}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{y\in\mathcal{Y}} \pi_{i,y} \log \widehat{\pi}_{i,y},$$

$$\mathcal{L}_{\text{dom}} = -\frac{1}{N} \sum_{i=1}^{N} \log \widehat{\pi}_{i,y_i^{\text{dom}}}.$$

The final objective is

$$\mathcal{L}_{\text{CFP}} = \lambda_{\text{fate}} \mathcal{L}_{\text{fate}} + \lambda_{\text{dom}} \mathcal{L}_{\text{dom}}, \tag{6}$$

with fixed weights

$$\lambda_{\text{fate}} = 3.0, \qquad \lambda_{\text{dom}} = 1.0.$$

The 3:1 weighting is used in all transfer runs.

# C Supplementary Experimental Settings

## C.1 Data and Preprocessing Details

Processed mdCLA data are available from OMIX (accession: OMIX014646; `https://ngdc.cncb.ac.cn/omix/release/OMIX014646`), and LARRY data are available from GEO (accession: GSE140802; `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE140802`). For both datasets, expression counts are library-size normalized to $10^4$ per cell and transformed with $\log(1+x)$. mdCLA uses a fixed highly variable gene (HVG) set as the input gene space with identical gene identity and ordering across temporal splits; LARRY uses its own HVG set, mapped by gene name onto the transferred encoder input as described in Appendix B.3.

HVG-space evaluation uses the same fixed gene set and ordering. A 50-dimensional PCA basis is fitted on training-visible cells for each split, then frozen and applied to held-out target cells. PCA fitting and anchor construction use training-visible data exclusively.

## C.2 Temporal Splits

**Inter-Time Interpolation.** Specified intermediate target stages are fully withheld during training, and the model reconstructs their cross-sectional transcriptomic snapshots from observed developmental stages.

**Strict Long-Horizon Extrapolation.** An independent frozen split fixes an earlier source stage and uses later target stages as future targets. Model fitting, anchor construction, and PCA fitting use training-visible data. The source/target lists are fixed by versioned split files.

**Masked Temporal Completion.** We additionally construct two masked temporal completion settings, Fetal Completion and Pre-Fetal Completion. Within the corresponding developmental period, $80\%$ of target observations are randomly masked and the remaining $20\%$ are visible; the model reconstructs the hidden transcriptomic states. These settings evaluate temporal completion within partially observed developmental periods. Developmental ranges, source/target construction, and the frozen masking protocol are fixed by the corresponding split files.

### C.3 Baseline Compatibility

All baselines use the same frozen splits, shared evaluation genes, and unified evaluator. NMAE, Sinkhorn, DES, and CSS are computed on the same target groups. DES compares predicted and observed differential-expression signatures relative to the task-specific reference state. scNODE and cfDiffusion are scored as unpaired population samples without source-cell correspondence; the evaluator does not compute DES in this mode, so DES is reported as N/A for both methods. Pre-Fetal Completion requires rolling later observations back to earlier stages. The PRESCIENT benchmark adapter always simulates forward from a source stage 0.5 days before the target, and its growth-coupled SDE has no direct time-reversed counterpart; scNODE always integrates forward from the earliest training time point. Neither method is therefore evaluated in Pre-Fetal Completion. Unavailable evaluator outputs are reported as N/A in figures and as "–" in tables.

### C.4 Metric Definitions

**Evaluation dimensions and spaces.** The four forecasting metrics capture complementary properties of the predicted target population. NMAE evaluates population-mean transcriptomic-state reconstruction; Sinkhorn divergence evaluates agreement between predicted and observed empirical cell-state distributions; DES evaluates recovery of differential-expression changes relative to the task-specific reference state; and CSS evaluates gene–gene co-expression structure. NMAE and Sinkhorn are computed in both the fixed 50-dimensional PCA space and the shared HVG expression space. DES and CSS are computed only in HVG space because they depend on gene identities or gene-pair relationships that are not directly preserved by PCA projection. The PCA basis is fitted exclusively on training-visible cells and then frozen for held-out evaluation. Lower values are better for NMAE, Sinkhorn, and CSS, whereas higher values are better for DES.

**NMAE.** For a matched target-time group $g$, let $v_g$ and $\widehat{v}_g$ denote the observed and predicted population means. NMAE is

$$\mathrm{NMAE} = \frac{\sum_g \|\widehat{v}_g - v_g\|_1}{\sum_g \|v_g\|_1 + \epsilon_{\mathrm{den}}}, \qquad \epsilon_{\mathrm{den}} = 10^{-12}. \tag{7}$$

$\mathrm{NMAE}_{\mathrm{PCA}}$ and $\mathrm{NMAE}_{\mathrm{HVG}}$ are computed in the fixed 50-dimensional PCA space and shared HVG expression space, respectively; lower values are better.

**Sinkhorn divergence.** At target time $t$, let

$$\widehat{P}_t = \frac{1}{m_t}\sum_{i=1}^{m_t} \delta_{\widehat{x}_{t,i}}, \qquad P_t = \frac{1}{n_t}\sum_{j=1}^{n_t} \delta_{x_{t,j}} \tag{8}$$

denote the predicted and observed empirical cell-state distributions, respectively. We use the debiased Sinkhorn divergence from entropic optimal transport,

$$S_\eta(P,Q) = \mathrm{OT}_\eta(P,Q) - \frac{1}{2}\,\mathrm{OT}_\eta(P,P) - \frac{1}{2}\,\mathrm{OT}_\eta(Q,Q). \tag{9}$$

$\mathrm{Sinkhorn}_{\mathrm{PCA}}$ and $\mathrm{Sinkhorn}_{\mathrm{HVG}}$ are computed in the fixed 50-dimensional PCA space and HVG space and aggregated across test target times; lower values are better.

**DES.** We use the Differential Expression Score (DES) from the Virtual Embryo Challenge evaluator.[1] For each matched target-time group $g$, the observed target is compared with its task-specific

[1] https://virtualembryo.ai/challenge/evaluation

reference state to define the true differentially expressed genes using a two-sided Mann–Whitney $U$ test, Benjamini–Hochberg FDR control at $\alpha = 0.05$, and an absolute mean-effect threshold of $0.25$. The resulting set is split into true up- and down-regulated genes, $U_g^{\mathrm{t}}$ and $D_g^{\mathrm{t}}$.

Let $n_u = |U_g^{\mathrm{t}}|$ and $n_d = |D_g^{\mathrm{t}}|$. The prediction is ranked by its mean expression change relative to the same reference, and the top $n_u$ and bottom $n_d$ genes define $U_g^{\mathrm{p}}$ and $D_g^{\mathrm{p}}$. The signed overlap is

$$O_g = \frac{|U_g^{\mathrm{p}} \cap U_g^{\mathrm{t}}| + |D_g^{\mathrm{p}} \cap D_g^{\mathrm{t}}|}{|U_g^{\mathrm{t}}| + |D_g^{\mathrm{t}}|}. \tag{10}$$

To discount overlap explainable from reference expression alone, the evaluator computes a chance overlap $O_g^{\mathrm{chance}}$ from reference-only rankings and applies

$$\mathrm{DES}_g = \frac{O_g - O_g^{\mathrm{chance}}}{1 - O_g^{\mathrm{chance}}}. \tag{11}$$

$\mathrm{DES}_{\mathrm{HVG}}$ is aggregated across matched groups in the shared HVG evaluation space; higher values are better. A score of 1 denotes exact recovery of the target differential-expression set, 0 corresponds to the reference-expression chance baseline, and negative values indicate worse-than-chance signed gene recovery.

**CSS.** For each matched target-time group $g$, let $\mathcal{P}$ be the gene-pair set in HVG space. For dataset $X$ and gene pair $(i, j)$, define the $p = 0.5$ variogram

$$v_X(i, j) = \frac{1}{|X|} \sum_{x \in X} |x_i - x_j|^{1/2}.$$

The group-level co-expression structure discrepancy is

$$\mathrm{CSS}_g = \frac{1}{|\mathcal{P}|} \sum_{(i,j) \in \mathcal{P}} \left( v_{\widehat{X}_g}(i, j) - v_{X_g}(i, j) \right)^2. \tag{12}$$

$\mathrm{CSS}_{\mathrm{HVG}}$ is averaged over matched groups; lower values are better.

**Dominant top-1 fate accuracy.** Cell Fate Prediction additionally reports dominant top-1 fate accuracy:

$$\mathrm{Acc}_{\mathrm{dom}} = \frac{1}{N} \sum_i \mathbf{1} \left[ \arg\max_y \widehat{\pi}_{i,y} = y_i^{\mathrm{dom}} \right]. \tag{13}$$

Higher values are better.

### C.5 Implementation Details

The shared developmental backbone uses batch size 2048, latent dimension 64, hidden dimension 256, four Fourier bands, and 16 rollout steps. The backbone is optimized with Adam using initial learning rate $5 \times 10^{-4}$. Early stopping uses training-visible validation cells. Baseline runs use the official training configurations for each method and share the same frozen evaluation splits.

### C.6 Hyperparameter and Temporal-Prior Sensitivity

Table 2 summarizes the one-factor-at-a-time sensitivity study using a fixed random seed (seed 1) and point estimates without error bars. The final benchmark and component-ablation results are evaluated over multiple seeds as described in their corresponding tables. Boldface in the Value column indicates the value used by the final scChrono configuration; the sensitivity sweep is diagnostic and individual alternatives can be preferable for particular metrics or task settings.

The sensitivity study examines whether the reported results depend on a narrow choice of model configuration. We vary latent dimension, hidden dimension, rollout steps $K$, and anchor weight $\lambda_{\mathrm{anchor}}$ one factor at a time while keeping the remaining settings fixed. Across the tested values, alternative settings often improve individual metrics, but no single alternative consistently dominates the final configuration across tasks and evaluation dimensions.

Table 2: Sensitivity to key scChrono hyperparameters. Each factor is varied independently using seed 1 while all other settings are fixed; bold indicates the final configuration.

| Hyperparameter | Value | $\mathrm{NMAE}_{\mathrm{PCA}} \downarrow$ | $\mathrm{Sinkhorn}_{\mathrm{PCA}} \downarrow$ | $\mathrm{NMAE}_{\mathrm{HVG}} \downarrow$ | $\mathrm{Sinkhorn}_{\mathrm{HVG}} \downarrow$ | $\mathrm{DES}_{\mathrm{HVG}} \uparrow$ | $\mathrm{CSS}_{\mathrm{HVG}} \downarrow$ |
|---|---|---|---|---|---|---|---|
| *Inter-Time Interpolation* | | | | | | | |
| Latent dimension | 32 | 0.646 | 12.900 | 0.320 | 191.514 | 0.266 | 0.022 |
| | **64** | 0.626 | 12.077 | 0.301 | 186.896 | 0.247 | 0.020 |
| | 128 | 0.650 | 12.046 | 0.293 | 187.161 | 0.238 | 0.019 |
| Hidden dimension | 128 | 0.657 | 13.573 | 0.295 | 193.976 | 0.304 | 0.019 |
| | **256** | 0.626 | 12.077 | 0.301 | 186.896 | 0.247 | 0.020 |
| | 512 | 0.657 | 12.740 | 0.353 | 188.155 | 0.247 | 0.023 |
| Rollout steps $K$ | 8 | 0.639 | 12.201 | 0.297 | 189.389 | 0.185 | 0.019 |
| | **16** | 0.626 | 12.077 | 0.301 | 186.896 | 0.247 | 0.020 |
| | 32 | 0.641 | 12.365 | 0.296 | 188.234 | 0.249 | 0.019 |
| Anchor weight $\lambda_{\mathrm{anchor}}$ | 1.0 | 0.634 | 11.910 | 0.303 | 185.284 | 0.248 | 0.019 |
| | **1.5** | 0.626 | 12.077 | 0.301 | 186.896 | 0.247 | 0.020 |
| | 2.0 | 0.670 | 12.633 | 0.322 | 191.487 | 0.174 | 0.020 |
| *Fetal Completion* | | | | | | | |
| Latent dimension | 32 | 0.480 | 14.320 | 0.115 | 256.510 | 0.585 | 0.014 |
| | **64** | 0.466 | 14.343 | 0.114 | 256.170 | 0.561 | 0.014 |
| | 128 | 0.504 | 14.654 | 0.122 | 257.474 | 0.541 | 0.013 |
| Hidden dimension | 128 | 0.504 | 15.349 | 0.116 | 266.881 | 0.601 | 0.014 |
| | **256** | 0.466 | 14.343 | 0.114 | 256.170 | 0.561 | 0.014 |
| | 512 | 0.509 | 14.931 | 0.113 | 253.310 | 0.541 | 0.014 |
| Rollout steps $K$ | 8 | 0.481 | 14.493 | 0.116 | 256.372 | 0.562 | 0.014 |
| | **16** | 0.466 | 14.343 | 0.114 | 256.170 | 0.561 | 0.014 |
| | 32 | 0.485 | 14.331 | 0.112 | 255.587 | 0.487 | 0.014 |
| Anchor weight $\lambda_{\mathrm{anchor}}$ | 1.0 | 0.496 | 14.580 | 0.115 | 256.689 | 0.545 | 0.013 |
| | **1.5** | 0.466 | 14.343 | 0.114 | 256.170 | 0.561 | 0.014 |
| | 2.0 | 0.497 | 14.327 | 0.115 | 256.404 | 0.491 | 0.014 |
| *Pre-Fetal Completion* | | | | | | | |
| Latent dimension | 32 | 0.395 | 7.730 | 0.310 | 139.950 | 0.324 | 0.021 |
| | **64** | 0.516 | 9.719 | 0.346 | 148.507 | 0.296 | 0.024 |
| | 128 | 0.662 | 14.004 | 0.410 | 177.953 | 0.234 | 0.029 |
| Hidden dimension | 128 | 0.501 | 12.623 | 0.461 | 171.556 | 0.262 | 0.032 |
| | **256** | 0.516 | 9.719 | 0.346 | 148.507 | 0.296 | 0.024 |
| | 512 | 0.550 | 12.675 | 0.397 | 160.986 | 0.321 | 0.029 |
| Rollout steps $K$ | 8 | 0.534 | 9.778 | 0.329 | 146.340 | 0.257 | 0.023 |
| | **16** | 0.516 | 9.719 | 0.346 | 148.507 | 0.296 | 0.024 |
| | 32 | 0.493 | 9.713 | 0.329 | 147.089 | 0.279 | 0.023 |
| Anchor weight $\lambda_{\mathrm{anchor}}$ | 1.0 | 0.572 | 11.784 | 0.383 | 162.920 | 0.255 | 0.027 |
| | **1.5** | 0.516 | 9.719 | 0.346 | 148.507 | 0.296 | 0.024 |
| | 2.0 | 0.432 | 8.915 | 0.304 | 144.236 | 0.257 | 0.022 |

Table 3: Sensitivity to the temporal-prior window. Results use seed 1; bold indicates the interval used in the final model.

| Temporal-prior window $[t_{\mathrm{on}}, t_{\mathrm{off}}]$ | $\mathrm{NMAE}_{\mathrm{PCA}} \downarrow$ | $\mathrm{Sinkhorn}_{\mathrm{PCA}} \downarrow$ | $\mathrm{NMAE}_{\mathrm{HVG}} \downarrow$ | $\mathrm{Sinkhorn}_{\mathrm{HVG}} \downarrow$ | $\mathrm{CSS}_{\mathrm{HVG}} \downarrow$ |
|---|---|---|---|---|---|
| $[7.5, 14.5]$ | 0.644 | 12.103 | 0.299 | 187.846 | 0.019 |
| $[\mathbf{7.5}, \mathbf{16.5}]$ | 0.626 | 12.077 | 0.301 | 186.896 | 0.020 |
| $[7.5, 19.0]$ | 0.635 | 12.079 | 0.299 | 186.545 | 0.019 |

Table 4: Practical computational cost on mdCLA Inter-Time Interpolation. Total time includes data loading, preprocessing, training, and evaluation; peak CPU memory excludes GPU memory.

| Method | Training Time | Total Time | Peak CPU Memory |
|---|---|---|---|
| cfDiffusion | 8 h 9 min | 8 h 20 min | 126 GB |
| moscot | 19 min | 33 min | 124 GB |
| PRESCIENT | 2 h 27 min | 2 h 48 min | 138 GB |
| scDiffEq | 3 h 20 min | 3 h 38 min | 125 GB |
| scNODE | 10 min | 11 min | 124 GB |
| TrajectoryNet | 6 h 39 min | 7 h 04 min | 124 GB |
| PlainODE | 25 min | 33 min | 124 GB |
| PlainSDE | 27 min | 34 min | 124 GB |
| scChrono | 1 h 32 min | 1 h 58 min | 126 GB |

In Inter-Time Interpolation, most settings remain within a relatively similar range. For example, increasing the latent dimension from 64 to 128 reduces $\mathrm{NMAE}_{\mathrm{HVG}}$ from $0.301$ to $0.293$, but increases $\mathrm{NMAE}_{\mathrm{PCA}}$ from $0.626$ to $0.650$. Hidden dimension 128 increases $\mathrm{DES}_{\mathrm{HVG}}$ to $0.304$, while both Sinkhorn metrics become worse than under the final hidden dimension of 256. Similar trade-offs appear for rollout steps and anchor weight: $\lambda_{\mathrm{anchor}} = 1.0$ improves both Sinkhorn metrics, but does not improve the remaining evaluation dimensions consistently.

Fetal Completion shows a similar pattern. The final latent dimension of 64 gives the lowest $\mathrm{NMAE}_{\mathrm{PCA}}$, whereas hidden dimension 128 gives a higher $\mathrm{DES}_{\mathrm{HVG}}$. Increasing $K$ to 32 slightly improves several transcriptomic-state reconstruction and population-distribution matching metrics but reduces $\mathrm{DES}_{\mathrm{HVG}}$ from $0.561$ to $0.487$. Pre-Fetal Completion is more sensitive to the tested hyperparameters: latent dimension 32 improves several metrics relative to 64, and $\lambda_{\mathrm{anchor}} = 2.0$ improves several transcriptomic-state reconstruction and population-distribution matching metrics, but these gains do not remain consistent across the other task settings. We therefore keep one shared configuration across all tasks instead of tuning each task separately.

Because the temporal prior directly modulates Fast-Transition Dynamics, we also vary the temporal-prior window in Inter-Time Interpolation. Table 3 shows that the main transcriptomic-state reconstruction and population-distribution matching metrics remain in a narrow range across representative shorter and longer windows: $\mathrm{NMAE}_{\mathrm{PCA}}$ remains approximately $0.626$–$0.644$, while $\mathrm{Sinkhorn}_{\mathrm{PCA}}$ remains approximately $12.077$–$12.103$. These results indicate that the Inter-Time Interpolation results are not critically dependent on a single exact temporal-prior boundary within the tested range.

The sensitivity results therefore show task- and metric-specific trade-offs rather than dependence on one isolated hyperparameter value. The final scChrono configuration—latent dimension 64, hidden dimension 256, $K = 16$, $\lambda_{\mathrm{anchor}} = 1.5$, and temporal-prior window $[7.5, 16.5]$—is used consistently across the evaluated forecasting tasks rather than being separately tuned for each task.

### C.7 Computational Cost and Complexity

Table 4 reports practical computational cost on mdCLA Inter-Time Interpolation; we also report the corresponding asymptotic complexity. scChrono requires 1 h 32 min for training and 1 h 58 min end-to-end. This places its wall-clock cost above the lighter moscot, scNODE, PlainODE, and PlainSDE baselines, but below cfDiffusion, PRESCIENT, scDiffEq, and TrajectoryNet. Peak CPU memory is 126 GB, within the 124–138 GB range observed across the compared methods. Because each method uses its recommended model configuration and optimization procedure, these measurements characterize practical resource requirements rather than a controlled efficiency comparison at matched predictive accuracy.

Table 5: Asymptotic training-time and memory complexity.

| Method | Training-Time Complexity | Memory Complexity |
|---|---|---|
| cfDiffusion | $\mathcal{O}(E_{\text{vae}}Bgh + E_{\text{diff}}Bh^2)$ | $\mathcal{O}(B(g+h) + gh + h^2)$ |
| moscot | $\mathcal{O}(In_sn_td)$ | $\mathcal{O}(n_sn_t)$ |
| PRESCIENT | $\mathcal{O}(EB(gh + Kh^2))$ | $\mathcal{O}(B(g+h) + Kh + h^2)$ |
| scDiffEq | $\mathcal{O}(EB(gh + Kh^2))$ | $\mathcal{O}(B(g+h) + Kh + h^2)$ |
| scNODE | $\mathcal{O}(EB(gh + Kh^2))$ | $\mathcal{O}(B(g+h) + Kh + h^2)$ |
| TrajectoryNet | $\mathcal{O}(EB(gh + Kh^2))$ | $\mathcal{O}(B(g+h) + Kh + h^2)$ |
| PlainODE | $\mathcal{O}(EB(gh + Kh^2))$ | $\mathcal{O}(B(g+h) + Kh + h^2)$ |
| PlainSDE | $\mathcal{O}(EB(gh + Kh^2))$ | $\mathcal{O}(B(g+h) + Kh + h^2)$ |
| scChrono | $\mathcal{O}(EB(gh + Kh^2))$ | $\mathcal{O}(B(g+h) + Kh + h^2)$ |

*Notation:* $n_s, n_t$ are source and target population sizes; $d$ is the OT feature dimension; $I$ is the number of OT iterations; $E$ is the number of optimization updates; $B$ is batch size; $g$ is input-gene dimension; $h$ is hidden dimension; $K$ is the number of numerical integration steps; and $E_{\text{vae}}, E_{\text{diff}}$ denote the VAE and diffusion optimization updates for cfDiffusion.

The practical measurements are consistent with the theoretical scaling in Table 5. For a single pass over $N$ cells, the dominant scChrono computation scales as $\mathcal{O}\big(N(gh + Kh^2)\big)$; with $E$ mini-batch optimization updates of size $B$, the corresponding training complexity is $\mathcal{O}(EB(gh+Kh^2))$, with memory complexity $\mathcal{O}(B(g + h) + Kh + h^2)$. This is the same leading-order scaling as the ODE/SDE-style neural baselines. The multi-rate decomposition therefore adds practical constant-factor cost without changing the leading-order dependence on data size, hidden dimension, or roll-out resolution. These asymptotic expressions describe scaling behavior rather than wall-clock time, which also depends on solver choice, implementation, hardware utilization, and optimization settings.

The practical runtime and asymptotic analysis provide complementary views of computational cost. scChrono is not the fastest method, but its multi-rate dynamics remain within the same overall memory range and asymptotic complexity class as the main neural ODE/SDE baselines while requiring less wall-clock training time than several of the more computationally expensive comparison methods.

# D Supplementary Results

## D.1 Primary Forecasting Results

Table 6 reports the exact mean ± SD values corresponding to the primary forecasting comparison in Figure 3. In Inter-Time Interpolation, scChrono is best on all six metrics; relative to the strongest competing value for each metric, $\text{Sinkhorn}_{\text{PCA}}$, $\text{NMAE}_{\text{HVG}}$, and $\text{Sinkhorn}_{\text{HVG}}$ improve by 21.9%, 16.0%, and 11.8%, while $\text{NMAE}_{\text{PCA}}$ and $\text{CSS}_{\text{HVG}}$ improve by 3.5% and 4.5%, and $\text{DES}_{\text{HVG}}$ increases from $0.251$ to $0.258$.

Under Strict Long-Horizon Extrapolation, scChrono is best on five of six metrics. Relative to the strongest competing values, $\text{Sinkhorn}_{\text{PCA}}$, $\text{NMAE}_{\text{HVG}}$, $\text{Sinkhorn}_{\text{HVG}}$, and $\text{CSS}_{\text{HVG}}$ improve by 16.1%, 14.1%, 7.4%, and 4.5%, respectively, while $\text{NMAE}_{\text{PCA}}$ improves by 0.1%. $\text{DES}_{\text{HVG}}$ is the only exception: PRESCIENT obtains $-0.008 \pm 0.037$, compared with $-0.071 \pm 0.096$ for scChrono.

## D.2 Temporal Completion Results

Table 7 reports the two masked temporal completion settings using the same six metrics as the primary forecasting evaluation. In Fetal Completion, scChrono achieves the best value on all six metrics. Relative to the strongest competing value for each metric, $\text{NMAE}_{\text{PCA}}$, $\text{Sinkhorn}_{\text{PCA}}$, $\text{NMAE}_{\text{HVG}}$, and $\text{Sinkhorn}_{\text{HVG}}$ improve by **38.2%**, **30.5%**, **28.8%**, and **11.6%**, respectively, while $\text{CSS}_{\text{HVG}}$ is reduced by **17.6%**. For $\text{DES}_{\text{HVG}}$, scChrono reaches $0.557 \pm 0.016$, compared with $0.144 \pm 0.013$ for the strongest competing method, PRESCIENT.

The same pattern holds in Pre-Fetal Completion, where scChrono again leads all six metrics. Compared with the strongest competing values, $\text{NMAE}_{\text{PCA}}$ and $\text{Sinkhorn}_{\text{PCA}}$ improve by **14.3%** and **12.6%**, $\text{NMAE}_{\text{HVG}}$ and $\text{Sinkhorn}_{\text{HVG}}$ improve by **33.6%** and **20.3%**, and $\text{CSS}_{\text{HVG}}$ is reduced by **16.7%**. $\text{DES}_{\text{HVG}}$ increases from $0.054 \pm 0.009$ for the strongest competitor, TrajectoryNet,

Table 6: Complete results on the two primary forecasting settings. Values are mean ± SD over available seeds and are reported to three decimal places; moscot is deterministic for a fixed split, so a single result is reported without SD; "–" denotes unavailable evaluations. Bold indicates the best value at the displayed precision.

| Method | $\mathrm{NMAE}_{\mathrm{PCA}}\downarrow$ | $\mathrm{Sinkhorn}_{\mathrm{PCA}}\downarrow$ | $\mathrm{NMAE}_{\mathrm{HVG}}\downarrow$ | $\mathrm{Sinkhorn}_{\mathrm{HVG}}\downarrow$ | $\mathrm{DES}_{\mathrm{HVG}}\uparrow$ | $\mathrm{CSS}_{\mathrm{HVG}}\downarrow$ |
|---|---|---|---|---|---|---|
| *Inter-Time Interpolation* | | | | | | |
| cfDiffusion | – | – | – | – | – | – |
| moscot | 1.396 | 41.616 | 0.575 | 380.140 | 0.121 | 0.027 |
| PRESCIENT | $0.717 \pm 0.013$ | $17.202 \pm 0.061$ | $0.381 \pm 0.003$ | $225.985 \pm 0.316$ | $0.223 \pm 0.033$ | $0.023 \pm 0.000$ |
| scDiffEq | $2.147 \pm 0.013$ | $43.845 \pm 0.236$ | $1.113 \pm 0.007$ | $322.766 \pm 1.814$ | $-0.024 \pm 0.003$ | $0.066 \pm 0.001$ |
| scNODE | $4.480 \pm 0.419$ | $94.019 \pm 17.502$ | $1.276 \pm 0.132$ | $927.221 \pm 234.545$ | – | $0.127 \pm 0.022$ |
| TrajectoryNet | $0.684 \pm 0.005$ | $15.988 \pm 0.118$ | $0.387 \pm 0.000$ | $215.297 \pm 0.197$ | $0.251 \pm 0.008$ | $0.022 \pm 0.000$ |
| PlainODE | $4.580 \pm 0.047$ | $742.205 \pm 7.205$ | $5.483 \pm 0.034$ | $4937.341 \pm 103.274$ | $-0.207 \pm 0.006$ | $0.831 \pm 0.003$ |
| PlainSDE | $4.549 \pm 0.034$ | $719.694 \pm 7.430$ | $5.402 \pm 0.041$ | $4554.265 \pm 73.465$ | $-0.211 \pm 0.004$ | $0.804 \pm 0.006$ |
| **scChrono** | $\mathbf{0.660 \pm 0.028}$ | $\mathbf{12.484 \pm 0.332}$ | $\mathbf{0.320 \pm 0.018}$ | $\mathbf{189.825 \pm 2.951}$ | $\mathbf{0.258 \pm 0.058}$ | $\mathbf{0.021 \pm 0.001}$ |
| *Strict Long-Horizon Extrapolation* | | | | | | |
| cfDiffusion | – | – | – | – | – | – |
| moscot | – | – | – | – | – | – |
| PRESCIENT | $0.826 \pm 0.035$ | $37.255 \pm 0.933$ | $0.419 \pm 0.011$ | $387.628 \pm 2.766$ | $\mathbf{-0.008 \pm 0.037}$ | $0.022 \pm 0.001$ |
| scDiffEq | $2.009 \pm 0.083$ | $91.318 \pm 1.123$ | $0.804 \pm 0.037$ | $620.117 \pm 5.437$ | $-0.136 \pm 0.036$ | $0.059 \pm 0.003$ |
| scNODE | $17.121 \pm 6.786$ | $2316.306 \pm 1334.627$ | $4.331 \pm 1.914$ | $10704.234 \pm 6846.160$ | – | $0.691 \pm 0.360$ |
| TrajectoryNet | $1.382 \pm 0.257$ | $53.421 \pm 9.810$ | $0.531 \pm 0.052$ | $419.128 \pm 18.451$ | $-0.097 \pm 0.065$ | $0.039 \pm 0.006$ |
| PlainODE | $7555.114 \pm 10865.354$ | $(6.628 \pm 10.804) \times 10^{17}$ | $3700.293 \pm 4609.450$ | $(3.602 \pm 7.859) \times 10^{18}$ | $-0.160 \pm 0.050$ | $214.726 \pm 279.487$ |
| PlainSDE | $1466.134 \pm 3108.968$ | $(7.241 \pm 16.192) \times 10^{15}$ | $720.940 \pm 1500.341$ | $(1.489 \pm 3.329) \times 10^{16}$ | $-0.154 \pm 0.066$ | $91.078 \pm 188.553$ |
| **scChrono** | $\mathbf{0.825 \pm 0.037}$ | $\mathbf{31.251 \pm 1.211}$ | $\mathbf{0.360 \pm 0.012}$ | $\mathbf{359.108 \pm 2.585}$ | $-0.071 \pm 0.096$ | $\mathbf{0.021 \pm 0.002}$ |

Table 7: Complete results on the two masked temporal completion settings. Values are mean ± SD over available seeds and are reported to three decimal places; moscot is deterministic for a fixed split, so a single result is reported without SD; "–" denotes unavailable evaluations. Bold indicates the best value.

| Method | $\mathrm{NMAE}_{\mathrm{PCA}}\downarrow$ | $\mathrm{Sinkhorn}_{\mathrm{PCA}}\downarrow$ | $\mathrm{NMAE}_{\mathrm{HVG}}\downarrow$ | $\mathrm{Sinkhorn}_{\mathrm{HVG}}\downarrow$ | $\mathrm{DES}_{\mathrm{HVG}}\uparrow$ | $\mathrm{CSS}_{\mathrm{HVG}}\downarrow$ |
|---|---|---|---|---|---|---|
| *Fetal Completion* | | | | | | |
| cfDiffusion | $3.021 \pm 0.042$ | $39.828 \pm 0.405$ | $0.893 \pm 0.004$ | $491.024 \pm 1.741$ | – | $0.070 \pm 0.001$ |
| moscot | 2.483 | 37.437 | 0.420 | 381.237 | $-0.042$ | 0.020 |
| PRESCIENT | $0.790 \pm 0.008$ | $20.694 \pm 0.095$ | $0.163 \pm 0.002$ | $290.022 \pm 0.500$ | $0.144 \pm 0.013$ | $0.017 \pm 0.000$ |
| scDiffEq | $2.751 \pm 0.125$ | $40.532 \pm 0.882$ | $0.561 \pm 0.028$ | $381.115 \pm 3.715$ | $0.068 \pm 0.016$ | $0.041 \pm 0.002$ |
| scNODE | $30.335 \pm 14.655$ | $1419.475 \pm 954.531$ | $3.890 \pm 1.546$ | $7864.394 \pm 5055.571$ | – | $0.580 \pm 0.275$ |
| TrajectoryNet | $1.416 \pm 0.167$ | $22.540 \pm 0.575$ | $0.242 \pm 0.015$ | $295.430 \pm 1.369$ | $0.016 \pm 0.063$ | $0.020 \pm 0.001$ |
| PlainODE | $4.409 \pm 0.074$ | $952.088 \pm 22.085$ | $2.412 \pm 0.095$ | $4862.550 \pm 174.871$ | $-0.135 \pm 0.004$ | $1.037 \pm 0.017$ |
| PlainSDE | $4.447 \pm 0.064$ | $894.169 \pm 18.202$ | $2.426 \pm 0.075$ | $4648.194 \pm 130.853$ | $-0.136 \pm 0.004$ | $0.965 \pm 0.012$ |
| **scChrono** | $\mathbf{0.488 \pm 0.028}$ | $\mathbf{14.388 \pm 0.070}$ | $\mathbf{0.116 \pm 0.003}$ | $\mathbf{256.359 \pm 0.277}$ | $\mathbf{0.557 \pm 0.016}$ | $\mathbf{0.014 \pm 0.000}$ |
| *Pre-Fetal Completion* | | | | | | |
| cfDiffusion | $1.026 \pm 0.005$ | $13.396 \pm 0.043$ | $0.788 \pm 0.007$ | $203.783 \pm 0.859$ | – | $0.043 \pm 0.000$ |
| moscot | 1.776 | 105.514 | 1.025 | 818.727 | $-0.131$ | 0.056 |
| PRESCIENT | – | – | – | – | – | – |
| scDiffEq | $2.010 \pm 0.004$ | $65.373 \pm 0.738$ | $1.517 \pm 0.003$ | $354.043 \pm 1.304$ | $-0.004 \pm 0.021$ | $0.090 \pm 0.000$ |
| scNODE | – | – | – | – | – | – |
| TrajectoryNet | $0.631 \pm 0.027$ | $12.538 \pm 0.110$ | $0.533 \pm 0.013$ | $196.906 \pm 0.810$ | $0.054 \pm 0.009$ | $0.030 \pm 0.001$ |
| PlainODE | $5.124 \pm 0.079$ | $637.051 \pm 16.659$ | $7.980 \pm 0.059$ | $4986.590 \pm 197.127$ | $-0.177 \pm 0.004$ | $0.814 \pm 0.006$ |
| PlainSDE | $5.141 \pm 0.088$ | $620.879 \pm 8.359$ | $7.988 \pm 0.085$ | $4936.703 \pm 311.125$ | $-0.174 \pm 0.002$ | $0.795 \pm 0.018$ |
| **scChrono** | $\mathbf{0.541 \pm 0.015}$ | $\mathbf{10.961 \pm 0.887}$ | $\mathbf{0.354 \pm 0.019}$ | $\mathbf{156.944 \pm 6.283}$ | $\mathbf{0.269 \pm 0.038}$ | $\mathbf{0.025 \pm 0.002}$ |

to $0.269 \pm 0.038$ for scChrono. These results extend the primary forecasting results to partially observed developmental periods with large temporal gaps.

### D.3 Additional Ablation Results

Throughout the supplementary ablation tables, Fast-Transition Dynamics (FTD), Persistent Dynamics (PD), Late Organ Adapter (LOA), and Weak Lineage Anchors (WLA) are abbreviated as indicated; **Full** denotes the complete scChrono model.

**Primary component-ablation details.** Table 1 shows that FTD has little effect under Inter-Time Interpolation but becomes consistently beneficial under Strict Long-Horizon Extrapolation. The PD-only control provides a stronger contrast. In Inter-Time Interpolation, $\mathrm{NMAE}_{\mathrm{PCA}}$ increases from 0.660 to 1.711, $\mathrm{Sinkhorn}_{\mathrm{PCA}}$ from 12.484 to 110.017, $\mathrm{NMAE}_{\mathrm{HVG}}$ from 0.320 to 1.373, and $\mathrm{Sinkhorn}_{\mathrm{HVG}}$ from 189.825 to 580.007, while $\mathrm{DES}_{\mathrm{HVG}}$ decreases from 0.258 to 0.023 and $\mathrm{CSS}_{\mathrm{HVG}}$ increases from 0.021 to 0.083. This unusually large Inter-Time degradation reflects an

Table 8: Parameter-matched controls under Strict Long-Horizon Extrapolation. The matched-single-drift model uses hidden width 406 to closely match the parameter count of PD, FTD, and the gate in Full scChrono. Results are mean ± SD over five seeds.

| Variant | $\mathrm{NMAE}_{\mathrm{PCA}} \downarrow$ | $\mathrm{Sinkhorn}_{\mathrm{PCA}} \downarrow$ | $\mathrm{NMAE}_{\mathrm{HVG}} \downarrow$ | $\mathrm{Sinkhorn}_{\mathrm{HVG}} \downarrow$ | $\mathrm{DES}_{\mathrm{HVG}} \uparrow$ | $\mathrm{CSS}_{\mathrm{HVG}} \downarrow$ |
|---|---|---|---|---|---|---|
| **Full** | $\mathbf{0.825 \pm 0.037}$ | $\mathbf{31.251 \pm 1.211}$ | $\mathbf{0.360 \pm 0.012}$ | $\mathbf{359.108 \pm 2.585}$ | $\mathbf{-0.071 \pm 0.096}$ | $\mathbf{0.021 \pm 0.002}$ |
| Matched-single-drift | $0.967 \pm 0.093$ | $35.228 \pm 1.639$ | $0.474 \pm 0.071$ | $375.154 \pm 6.042$ | $-0.205 \pm 0.063$ | $0.028 \pm 0.005$ |
| No gate | $0.939 \pm 0.020$ | $33.678 \pm 0.608$ | $0.457 \pm 0.020$ | $369.671 \pm 2.544$ | $-0.253 \pm 0.015$ | $0.026 \pm 0.003$ |
| −FTD | $0.908 \pm 0.037$ | $33.451 \pm 1.704$ | $0.444 \pm 0.032$ | $369.541 \pm 8.072$ | $-0.168 \pm 0.079$ | $0.027 \pm 0.002$ |

Table 9: FTD gate controls under Strict Long-Horizon Extrapolation. Full combines the learned gate and temporal prior; Gate only removes the temporal prior, Prior only removes the learned gate, and No gate removes both. Results are mean ± SD over five seeds.

| Variant | $\mathrm{NMAE}_{\mathrm{PCA}} \downarrow$ | $\mathrm{Sinkhorn}_{\mathrm{PCA}} \downarrow$ | $\mathrm{NMAE}_{\mathrm{HVG}} \downarrow$ | $\mathrm{Sinkhorn}_{\mathrm{HVG}} \downarrow$ | $\mathrm{DES}_{\mathrm{HVG}} \uparrow$ | $\mathrm{CSS}_{\mathrm{HVG}} \downarrow$ |
|---|---|---|---|---|---|---|
| **Full** | $\mathbf{0.825 \pm 0.037}$ | $\mathbf{31.251 \pm 1.211}$ | $\mathbf{0.360 \pm 0.012}$ | $\mathbf{359.108 \pm 2.585}$ | $\mathbf{-0.071 \pm 0.096}$ | $\mathbf{0.021 \pm 0.002}$ |
| Gate only | $0.913 \pm 0.049$ | $34.464 \pm 2.633$ | $0.445 \pm 0.038$ | $378.015 \pm 16.599$ | $-0.156 \pm 0.070$ | $0.028 \pm 0.003$ |
| Prior only | $0.964 \pm 0.060$ | $36.666 \pm 2.404$ | $0.433 \pm 0.026$ | $373.690 \pm 8.313$ | $-0.261 \pm 0.030$ | $\mathbf{0.021 \pm 0.001}$ |
| No gate | $0.939 \pm 0.020$ | $33.678 \pm 0.608$ | $0.457 \pm 0.020$ | $369.671 \pm 2.544$ | $-0.253 \pm 0.015$ | $0.026 \pm 0.003$ |

observed numerical instability of the drift-only configuration without diffusion or FTD, rather than a reporting artifact. Under Strict Long-Horizon Extrapolation, PD only remains worse than Full but stays on the same numerical scale as the other component ablations, indicating that the pronounced instability is specific to the Inter-Time setting.

LOA and WLA show smaller and more task-dependent effects. Removing LOA changes most primary-setting metrics only marginally. Removing WLA under Strict Long-Horizon Extrapolation worsens several reconstruction and distribution metrics, whereas DES and CSS do not change in the same direction. These auxiliary components therefore produce more mixed effects than the dynamics decomposition itself.

**Capacity-matched dynamics control.** To separate architectural structure from model size, we compare Full scChrono with a single-drift control whose hidden width is increased to 406. This produces 221,334 parameters in the single drift network, closely matching the 221,569 parameters used by PD, FTD, and the gate together in the Full model. Table 8 shows that Full scChrono remains better across all six Strict Long-Horizon Extrapolation metrics under this matched parameter budget. The two-branch no-gate control and the standard −FTD ablation are included for reference. In the corresponding control rerun, PlainSDE was numerically unstable, with extremely large NMAE and Sinkhorn values, and is therefore not included as a directly comparable numeric row.

**Gate-mechanism controls.** We next isolate the two factors that modulate FTD. The gate-only variant removes the fixed temporal prior, the prior-only variant removes the learned gate network, and the no-gate variant removes both. Table 9 shows that Full scChrono is best on five metrics and tied with Prior only on $\mathrm{CSS}_{\mathrm{HVG}}$. Neither the learned gate nor the temporal prior alone reproduces the Full result, indicating that state-dependent modulation and temporal conditioning play complementary roles in the strict long-horizon setting.

**Horizon-wise analysis of FTD.** To distinguish the role of FTD in the jointly trained model from its direct contribution during a fixed rollout, we perform two horizon-wise analyses under Strict Long-Horizon Extrapolation. For each of ten target horizons ($\Delta t = 0.5, 1.0, \ldots, 5.0$), we define a degradation $\Delta m$ such that positive values always indicate worse performance after removing FTD: for lower-is-better metrics, $\Delta m = m_{-\mathrm{FTD}} - m_{\mathrm{Full}}$, whereas for $\mathrm{DES}_{\mathrm{HVG}}$, $\Delta m = m_{\mathrm{Full}} - m_{-\mathrm{FTD}}$. We then compute the Spearman correlation between $\Delta t$ and $\Delta m$, with two-sided $p$-values from the exact permutation distribution for ten horizons, whose smallest attainable value is $5.5 \times 10^{-7}$.

The first analysis compares independently retrained Full and −FTD models, using five seeds for both. Table 10 shows that the performance gap widens with horizon for the five NMAE, Sinkhorn, and CSS metrics, with correlations ranging from $\rho = 0.867$ to $1.000$ and all exact permutation $p \leq 0.0022$. The clearest trend is observed for $\mathrm{NMAE}_{\mathrm{HVG}}$ ($\rho = 1.000$): the gap between Full and

Table 10: Horizon-wise FTD analysis under Strict Long-Horizon Extrapolation. Spearman $\rho$ measures the association between forecast horizon and performance degradation after removing FTD. Retrained ablation and inference-time masking isolate training-level and direct rollout effects, respectively. $p$-values are two-sided and computed from the exact permutation distribution over ten horizons.

| | Retrained −FTD ablation | | Inference-time FTD masking | |
|---|---|---|---|---|
| Metric | $\rho$ | $p$ | $\rho$ | $p$ |
| $\mathrm{NMAE}_{\mathrm{HVG}}$ | 1.000 | $5.5\times10^{-7}$ | −0.939 | $2.1\times10^{-4}$ |
| $\mathrm{Sinkhorn}_{\mathrm{HVG}}$ | 0.976 | $2.1\times10^{-5}$ | 0.927 | $3.3\times10^{-4}$ |
| $\mathrm{CSS}_{\mathrm{HVG}}$ | 0.988 | $5.5\times10^{-6}$ | −0.915 | $5.3\times10^{-4}$ |
| $\mathrm{Sinkhorn}_{\mathrm{PCA}}$ | 0.988 | $5.5\times10^{-6}$ | 0.697 | 0.031 |
| $\mathrm{NMAE}_{\mathrm{PCA}}$ | 0.867 | 0.0022 | 0.321 | 0.37 |
| $\mathrm{DES}_{\mathrm{HVG}}$ | 0.539 | 0.114 | −0.877 | 0.0016 |

Table 11: Component ablation on the two masked temporal completion settings. Results are mean ± SD over five seeds; the − prefix denotes component removal.

| Variant | $\mathrm{NMAE}_{\mathrm{PCA}}\downarrow$ | $\mathrm{Sinkhorn}_{\mathrm{PCA}}\downarrow$ | $\mathrm{NMAE}_{\mathrm{HVG}}\downarrow$ | $\mathrm{Sinkhorn}_{\mathrm{HVG}}\downarrow$ | $\mathrm{DES}_{\mathrm{HVG}}\uparrow$ | $\mathrm{CSS}_{\mathrm{HVG}}\downarrow$ |
|---|---|---|---|---|---|---|
| *Fetal Completion* | | | | | | |
| **Full** | $0.488 \pm 0.028$ | $14.388 \pm 0.070$ | $0.116 \pm 0.003$ | $256.359 \pm 0.277$ | $0.557 \pm 0.016$ | $0.014 \pm 0.000$ |
| −FTD | $0.494 \pm 0.016$ | $14.559 \pm 0.040$ | $0.114 \pm 0.004$ | $257.305 \pm 0.326$ | $0.558 \pm 0.021$ | $0.014 \pm 0.000$ |
| PD only | $3.384 \pm 0.416$ | $85.315 \pm 8.401$ | $0.438 \pm 0.038$ | $594.199 \pm 41.009$ | $0.049 \pm 0.101$ | $0.055 \pm 0.006$ |
| −LOA | $0.516 \pm 0.023$ | $15.118 \pm 0.089$ | $0.121 \pm 0.003$ | $259.428 \pm 0.475$ | $0.542 \pm 0.018$ | $0.014 \pm 0.000$ |
| −WLA | $0.439 \pm 0.010$ | $16.757 \pm 0.154$ | $0.115 \pm 0.001$ | $266.769 \pm 0.631$ | $0.597 \pm 0.015$ | $0.014 \pm 0.000$ |
| *Pre-Fetal Completion* | | | | | | |
| **Full** | $0.541 \pm 0.015$ | $10.961 \pm 0.887$ | $0.354 \pm 0.019$ | $156.944 \pm 6.283$ | $0.269 \pm 0.038$ | $0.025 \pm 0.002$ |
| −FTD | $0.349 \pm 0.058$ | $9.368 \pm 2.944$ | $0.291 \pm 0.064$ | $155.341 \pm 30.551$ | $0.477 \pm 0.024$ | $0.022 \pm 0.004$ |
| PD only | $11.122 \pm 3.850$ | $1993.294 \pm 1026.660$ | $13.026 \pm 4.305$ | $7059.950 \pm 2742.369$ | $-0.139 \pm 0.013$ | $0.865 \pm 0.270$ |
| −LOA | $0.710 \pm 0.046$ | $19.746 \pm 3.628$ | $0.479 \pm 0.058$ | $252.897 \pm 56.311$ | $0.248 \pm 0.028$ | $0.034 \pm 0.004$ |
| −WLA | $6.507 \pm 7.937$ | $7699.662 \pm 14795.095$ | $10.493 \pm 15.001$ | $529234.986 \pm 825461.101$ | $0.138 \pm 0.022$ | $0.922 \pm 1.381$ |

−FTD is approximately $0.01$ at $\Delta t = 0.5$ and reaches approximately $0.21$ at $\Delta t = 5.0$, as shown in Figure 4. $\mathrm{Sinkhorn}_{\mathrm{HVG}}$ shows a similarly strong horizon association ($\rho = 0.976$, $p = 2.1 \times 10^{-5}$), whereas $\mathrm{DES}_{\mathrm{HVG}}$ has a weaker, non-significant trend ($\rho = 0.539$, $p = 0.114$). The retrained ablation therefore becomes progressively more separated from Full scChrono as the rollout extends farther from the fixed source stage.

The second analysis keeps the trained Full model fixed and suppresses the effective FTD contribution only during inference. Its changes are much smaller and vary by metric: $\mathrm{Sinkhorn}_{\mathrm{HVG}}$ and $\mathrm{Sinkhorn}_{\mathrm{PCA}}$ retain positive horizon correlations, $\mathrm{NMAE}_{\mathrm{PCA}}$ shows no clear relationship, and $\mathrm{NMAE}_{\mathrm{HVG}}$, $\mathrm{CSS}_{\mathrm{HVG}}$, and $\mathrm{DES}_{\mathrm{HVG}}$ move in the opposite direction. This contrast separates two effects that are otherwise easy to conflate. The retrained gap between Full and −FTD reflects how the overall developmental dynamics are jointly optimized when the fast-transition branch is present or absent during training, whereas inference-time masking measures the immediate contribution of that branch in an already trained model. This explains why the retrained gap keeps widening at late horizons. There, the fixed temporal prior has largely switched off the instantaneous fast-transition term, but the dynamics learned with FTD during training still differ.

Table 1 reports the two primary forecasting settings. Here we evaluate the same five ablation variants on the two masked temporal completion settings, Fetal Completion and Pre-Fetal Completion. All six metrics are reported as mean ± SD over five seeds with three decimal places.

Table 11 shows how the components behave within partially observed developmental periods. In Fetal Completion, removing FTD changes the metrics by only about 0–1.7%, whereas retaining PD only degrades $\mathrm{NMAE}_{\mathrm{PCA}}$ from $0.488$ to $3.384$, $\mathrm{Sinkhorn}_{\mathrm{PCA}}$ from $14.388$ to $85.315$, and $\mathrm{NMAE}_{\mathrm{HVG}}$ from $0.116$ to $0.438$. Because PD only removes both diffusion and FTD, whereas removing FTD alone changes the metrics by at most $1.7\%$, most of this degradation is attributable to the absence of stochastic diffusion.

In Pre-Fetal Completion, removing FTD produces larger average improvements: $\mathrm{NMAE}_{\mathrm{PCA}}$, $\mathrm{Sinkhorn}_{\mathrm{PCA}}$, $\mathrm{NMAE}_{\mathrm{HVG}}$, $\mathrm{Sinkhorn}_{\mathrm{HVG}}$, and $\mathrm{CSS}_{\mathrm{HVG}}$ improve by approximately $35.5\%$, $14.5\%$, $17.8\%$, $1.0\%$, and $12.0\%$, respectively, and $\mathrm{DES}_{\mathrm{HVG}}$ increases from $0.269$ to $0.477$. Be-

Table 12: Effect of developmental-backbone transfer on dominant top-1 fate accuracy. Transfer and Scratch use the same downstream architecture and training protocol and differ only in backbone initialization.

| Initialization | Accuracy (mean $\pm$ SD) |
|---|---|
| Scratch | $0.578 \pm 0.013$ |
| Transfer | $\mathbf{0.614 \pm 0.009}$ |

cause $20\%$ of target-period observations remain visible, the persistent/stochastic dynamics and observed target structure provide strong completion signals, reducing the contribution of the fast-transition correction in this setting.

The auxiliary components show stronger task dependence in the completion settings. In Pre-Fetal Completion, removing **LOA** degrades $\mathrm{NMAE}_{\mathrm{PCA}}$ and $\mathrm{Sinkhorn}_{\mathrm{HVG}}$ by $31.2\%$ and $61.1\%$, respectively. **WLA** strongly affects population-level alignment in the same setting: removing it changes $\mathrm{Sinkhorn}_{\mathrm{HVG}}$ from $156.944$ to $529234.986$ and $\mathrm{CSS}_{\mathrm{HVG}}$ from $0.025$ to $0.922$, and lowers $\mathrm{DES}_{\mathrm{HVG}}$ from $0.269$ to $0.138$. These results indicate that LOA and WLA primarily support late-state refinement and long-range distribution alignment in partially observed temporal completion.

### D.4 Fate-Transfer Experiments

Table 12 reports the Transfer–Scratch comparison under the same downstream architecture and training protocol.

Transfer and Scratch use the same architecture, split, fate objective, and training protocol, with different backbone initialization.