KOOPCELL: KOOPMAN-BASED GENERATIVE MODEL FOR LEARNING SINGLE-CELL DYNAMICS FROM DISTRIBUTION SNAPSHOTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning population dynamics from temporally sparse, unpaired distribution snapshots is a fundamental challenge in developmental biology. Recent approaches based on neural differential equations and flow matching can interpolate between observed population snapshots, but may struggle to extrapolate beyond the training horizon and often lack an explicit mechanism for modeling developmental branching. We propose KoopCell, a unified generative framework based on Koopman–Mori–Zwanzig theory that jointly learns representations and predictive linear latent dynamics. Theoretically, using the weak continuity equation, we derive a closed-form least-squares estimator for the Koopman generator from distribution snapshots and establish convergence guarantees under suitable assumptions. To model branching dynamics, we further develop KoopCell-M, which incorporates non-Markovian memory into the latent Koopman dynamics through a Markovian embedding. Experiments on synthetic systems and three scRNA-seq datasets demonstrate the ability of our framework to recover Koopman spectra, model branching through memory, and scale to predicting high-dimensional gene expression distributions, achieving state-of-the-art performance among the evaluated methods.

1 INTRODUCTION

Inferring population dynamics from temporally sparse, unpaired distribution snapshots is a fundamental challenge in machine learning and the natural sciences, with applications in biology (Schiebinger et al., 2019; Moon et al., 2019), ecology (Ayala et al., 1973), and epidemiology (Wang et al., 2021; Kosma et al., 2023). In particular, advances in high-throughput single-cell RNA sequencing (scRNA-seq) (Hwang et al., 2018; Chen et al., 2019) have made this problem increasingly important in single-cell biology. By profiling population-level gene expression at multiple time points, scRNA-seq enables the study of cellular developmental dynamics, the identification of key driver genes, and the prediction of future states. However, due to the destructive nature of scRNA-seq measurements, only a cross-sectional sample from the population-level gene-expression distribution is available, without cell-level correspondence across time. Inferring the underlying continuous population dynamics from such observations therefore remains a substantial computational challenge.

Neural dynamical systems have been widely used to model cellular dynamics from population snapshots. Huguet et al. (2022); Zhang et al. (2024) use neural ordinary differential equations (ODEs) (Chen et al., 2018), whereas Yeo et al. (2021); Jiang & Wan (2024) employ neural stochastic differential equations (SDEs) to capture biological stochasticity in latent space. More recently, Zhang et al. (2025); Wang et al. (2025); Ling et al. (2026) have introduced unbalanced dynamical optimal transport to model cell proliferation and death. Although these methods perform well under their respective dynamical assumptions and can interpolate missing distributions within the observed time horizon, they do not effectively extrapolate beyond the observed time points or explicitly account for branching dynamics. Both capabilities are essential for predicting future cell-population states and representing the lineage differentiation that is crucial for understanding cellular development.

In this work, we propose KoopCell, a Koopman-based generative model for population dynamics. Koopman theory (Koopman, 1931; Koopman & von Neumann, 1932) provides a unifying perspective for modeling nonlinear dynamics by lifting the system state into a space of observables (functions of

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.

the system state), in which the evolution is governed by a linear operator. Neural Koopman methods (Lusch et al., 2018) approximate these observables with an encoder and represent their evolution using a finite-dimensional linear operator, but they generally rely on paired state transitions or trajectories. In contrast, KoopCell jointly learns the observable representation and its linear dynamics directly from temporally sparse, unpaired distribution snapshots. We theoretically formulate the estimation of the latent linear propagator through the weak continuity equation (Lu et al., 2024) and integrate it with a generative VAE model in an alternating learning procedure, enabling population prediction both within and beyond the observed time horizon. A single Markovian linear flow, however, may be insufficient to represent branching processes such as the differentiation of stem cells into multiple specialized cell types. We therefore extend KoopCell using the Mori–Zwanzig formalism, which accounts for the influence of unresolved variables through non-Markovian memory (Mori, 1965; Zwanzig, 1973). By embedding the suggested memory dynamics into an augmented latent space (Hijón et al., 2010; Menier et al., 2025), we develop tractable variational inference for the hidden memory and a corresponding alternating learning algorithm for the memory-augmented model, KoopCell-M. This memory mechanism enables KoopCell-M to model the divergence of a population into distinct developmental branches. Our main contributions are summarized as follows:

- We develop a theoretical and computational framework for identifying a finite-dimensional Koopman representation from unpaired distribution snapshots. Based on the weak continuity equation, we characterize the closed-form estimator, establish its convergence properties, and jointly learn the associated observable representation within a generative model.
- Furthermore, we introduce KoopCell-M, a Mori–Zwanzig-inspired extension that represents non-Markovian memory in an augmented latent space, providing a mechanistic formulation and a practical learning algorithm for population dynamics with branching.
- We evaluate KoopCell and KoopCell-M on synthetic toy systems and three high-dimensional scRNA-seq datasets. The experiments validate Koopman spectral recovery and the ability to model branching dynamics, while demonstrating state-of-the-art extrapolation performance in predicting cellular population distributions from sparse, unpaired measurements.

2 PRELIMINARIES

Problem Formulation and Notation. We consider a time series of data distributions ρ_t ($t \in \mathcal{T} \subset \mathbb{R}^+$), from each of which we observe n_t independent samples (e.g., cells) $\{\boldsymbol{x}_{t,i}\}_{i=1}^{n_t}$, with $\boldsymbol{x}_{t,i} \sim \rho_t$ and $\boldsymbol{x}_{t,i} \in \mathcal{X} = \mathbb{R}^G$. We aim to learn the population dynamics that explain the evolution of these distribution snapshots. A typical application is to recover cellular developmental dynamics from time-series scRNA-seq data, where cells sampled from ρ_t form a gene expression matrix $\boldsymbol{X}_t \in \mathbb{R}^{n_t \times G}$, and G is the number of genes. Given the high dimensionality of the observations, a common approach is to model temporal dynamics in a D-dimensional latent space $\mathcal{Z} = \mathbb{R}^D$, with $D \ll G$. Rather than specifying a heuristic dynamical structure in the latent space, we use Koopman theory to guide the learning of latent coordinates with approximately linear evolution. Throughout, we use uppercase state symbols for random variables and lowercase symbols for their realizations.

Koopman Theory and Koopman Learning. For an autonomous system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$ with flow F_t, an observable is a function $g : \mathcal{X} \to \mathbb{R}$ of the system state. For non-autonomous systems, augmenting the state with time ($\dot{t} = 1$) yields an autonomous formulation. The Koopman operator evolves observables by $[U(t)g](\boldsymbol{x}) = g(F_t\boldsymbol{x})$ and is linear even when the state-space dynamics are nonlinear (Koopman, 1931; Koopman & von Neumann, 1932). We assume that the operators $U(t) : \mathcal{H} \to \mathcal{H}$ form a strongly continuous semigroup on a Hilbert space $\mathcal{H}$ of observables, with generator L and domain $D(L)$. For sufficiently smooth $g \in D(L)$, $Lg = \boldsymbol{f} \cdot \nabla g$ and $\frac{\mathrm{d}}{\mathrm{d}t}U(t)g = U(t)Lg$. Let $\boldsymbol{X}_t = F_t\boldsymbol{X}_0$ with $\boldsymbol{X}_0 \sim \rho_0$. If $\mathcal{H}_D = \mathrm{span}\{g_1, \ldots, g_D\} \subset D(L)$ is *invariant* under L, then $L\boldsymbol{g} = \boldsymbol{A}\boldsymbol{g}$ for a matrix $\boldsymbol{A}$, so $\boldsymbol{Z}_t = \boldsymbol{g}(\boldsymbol{X}_t)$ satisfies $\dot{\boldsymbol{Z}}_t = \boldsymbol{A}\boldsymbol{Z}_t$, where $\boldsymbol{g} = (g_1, \ldots, g_D)^\top$. With the observable map $\boldsymbol{g}$ fixed, extended dynamic mode decomposition (EDMD) (Williams et al., 2015) estimates a propagator from paired states $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, $\boldsymbol{y}_i = F_\Delta\boldsymbol{x}_i$, by solving a regression problem $\min_{\boldsymbol{K}} \sum_i \|\boldsymbol{g}(\boldsymbol{y}_i) - \boldsymbol{K}\boldsymbol{g}(\boldsymbol{x}_i)\|_2^2$. Koopman representation learning extends EDMD by learning the observable map together with the latent dynamics. Specifically, an encoder $\boldsymbol{x} \mapsto \boldsymbol{z} = \boldsymbol{g}(\boldsymbol{x}) \in \mathbb{R}^D$ is jointly trained with a linear latent propagator and a decoder to obtain coordinates that support both approximately linear evolution and state reconstruction (Otto & Rowley, 2019; Lusch et al., 2018). Our work extends this framework to learning generative latent linear dynamics from temporally sparse, unpaired distribution snapshots.

Mori–Zwanzig Formalism and Memory Embedding. The reduced finite-dimensional linear dynamics above are exact when the span of the chosen observables is invariant under the Koopman generator L. When this linear closure fails, the evolution of the resolved coordinates $\boldsymbol{Z}_t = \boldsymbol{g}(\boldsymbol{X}_t) \in \mathbb{R}^D$ generally involves unresolved observables outside the chosen observable subspace $\mathcal{H}_D$. The Mori–Zwanzig formalism, originally developed in statistical mechanics, accounts for their influence through memory and forcing terms (Mori, 1965; Zwanzig, 1973). Specifically, under suitable regularity, Mori's linear projection onto $\mathcal{H}_D$ yields a generalized Langevin equation (Lin et al., 2021):

$$\dot{\boldsymbol{Z}}_t = \boldsymbol{A}\boldsymbol{Z}_t + \int_0^t \boldsymbol{K}(t-s)\boldsymbol{Z}_s \,\mathrm{d}s + \boldsymbol{\eta}_t.$$

Here $\boldsymbol{A}$ describes the instantaneous linear contribution, $\boldsymbol{K}$ is the memory kernel, and $\boldsymbol{\eta}_t$ carries the influence of unresolved initial conditions. A tractable closure approximates the kernel by $\boldsymbol{K}(u) \approx \boldsymbol{B}e^{\boldsymbol{\Lambda}u}\boldsymbol{C}$ and represents the unresolved influence through auxiliary hidden memory variables $\boldsymbol{H}_t \in \mathbb{R}^{d_h}$ (Lei et al., 2016; Menier et al., 2025). This leads to the augmented dynamics $\mathrm{d}\boldsymbol{Z}_t = (\boldsymbol{A}\boldsymbol{Z}_t + \boldsymbol{B}\boldsymbol{H}_t)\,\mathrm{d}t$ and $\mathrm{d}\boldsymbol{H}_t = (\boldsymbol{C}\boldsymbol{Z}_t + \boldsymbol{\Lambda}\boldsymbol{H}_t)\,\mathrm{d}t + \boldsymbol{\Sigma}_h\,\mathrm{d}\boldsymbol{W}_t$, where $\boldsymbol{W}_t$ is standard Brownian motion and $\boldsymbol{\Sigma}_h$ controls the hidden noise. Eliminating $\boldsymbol{H}_t$ yields the kernel $\boldsymbol{K}(u) = \boldsymbol{B}e^{\boldsymbol{\Lambda}u}\boldsymbol{C}$ and a temporally correlated forcing determined by $\boldsymbol{H}_0$ and the hidden noise. The augmented state $(\boldsymbol{Z}_t, \boldsymbol{H}_t)$ thus evolves as a Markovian system, while $\boldsymbol{Z}_t$ alone generally retains memory. This Markovian embedding provides the underlying mechanism for modeling branching dynamics in our framework.

Wasserstein Flow and Optimal Transport. Time series of distribution snapshots are often modeled as probability flows in Wasserstein space. We write $\mathcal{P}_2(\mathbb{R}^D)$ for the space of probability measures with finite second moment and W_2 for the Wasserstein-2 distance, defined by $W_2^2(\mu,\nu) = \inf_{\pi\in\Pi(\mu,\nu)} \int \|\boldsymbol{z}-\boldsymbol{y}\|^2 \mathrm{d}\pi(\boldsymbol{z},\boldsymbol{y})$, where $\Pi(\mu,\nu)$ is the set of couplings. We use $\mu[\psi] = \int \psi(\boldsymbol{z})\,\mathrm{d}\mu(\boldsymbol{z})$ and write $F_{\#}\mu$ for the distribution of $F(\boldsymbol{Z})$ when $\boldsymbol{Z} \sim \mu$. The Wasserstein flow $t \mapsto \mu_t$, with $\mu_\cdot \in AC^2([0,T];\mathcal{P}_2(\mathbb{R}^D))$, is an absolutely continuous curve in Wasserstein space that describes the evolution of probability measures over time. For almost every t, there exists a unique canonical velocity $\boldsymbol{v}_t \in T_{\mu_t}\mathcal{P}_2(\mathbb{R}^D)$[1] satisfying $\partial_t\mu_t + \nabla\cdot(\boldsymbol{v}_t\mu_t) = 0$ in the distributional sense (Ambrosio et al., 2008). Among all velocity fields generating the same measure curve, $\boldsymbol{v}_t$ has minimum $L^2(\mu_t)$ norm $\left(\int \|\boldsymbol{v}_t\|^2 \mathrm{d}\mu_t\right)^{1/2}$. We denote by $P_\mu : L^2(\mu;\mathbb{R}^D) \to T_\mu\mathcal{P}_2(\mathbb{R}^D)$ the $L^2(\mu)$-orthogonal projection onto this tangent space. For a zero-mass distribution σ, we define the negative Sobolev norm $\|\sigma\|_{-1,\mu} = \sup_{\int\|\nabla\varphi\|^2\mathrm{d}\mu\le 1} \langle\sigma,\varphi\rangle$. Here $\langle\sigma,\varphi\rangle$ denotes the duality pairing of a distribution with a test function. Equivalently, when finite, $\|\sigma\|_{-1,\mu}$ is the minimum $L^2(\mu)$ norm of a tangent field $\boldsymbol{w}$ satisfying $\sigma = -\nabla\cdot(\boldsymbol{w}\mu)$; see Lemma A.2 in Appendix A.1.

3 METHODS

3.1 KOOPMAN-BASED VAE FOR GENERATIVE MODELING OF POPULATION DYNAMICS

Motivated by Koopman theory, we jointly learn an encoder that maps cellular states to observables with approximately linear evolution, a linear propagator governing their dynamics, and a decoder reconstructing the cellular states from latent representations. Specifically, we use a VAE (Kingma & Welling, 2014), widely used for single-cell gene expression modeling (Grønbech et al., 2020), with encoder $q_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}, t) = \mathcal{N}(\boldsymbol{m}_{\boldsymbol{\theta}}(\boldsymbol{x},t), \boldsymbol{S}_{\boldsymbol{\theta}}(\boldsymbol{x},t))$ and decoder $\mathrm{Dec}_{\boldsymbol{\phi}}$, where $\boldsymbol{z} \in \mathbb{R}^D$. We concatenate t with $\boldsymbol{x}$ to encode the augmented state. Including the constant observable 1 yields affine dynamics $\dot{\boldsymbol{Z}}_t = \boldsymbol{A}\boldsymbol{Z}_t + \boldsymbol{b}_z$. For simplicity, we use $\dot{\boldsymbol{Z}}_t = \boldsymbol{A}\boldsymbol{Z}_t$ below and describe the detailed implementation in Appendix B.1.

Unlike traditional settings where the Koopman model is learned from densely sampled, paired trajectories, we are given only temporally sparse, unpaired distribution snapshots. We address the challenge of learning the Koopman operator from snapshots through the following two alternating steps: (1) with the observables (latent coordinates) fixed, we estimate the linear matrix that best fits the evolution of the latent distributions, providing an analogue of EDMD estimation for population snapshots; (2) with the estimated linear propagator fixed, we refine the latent representation to better fit the dynamics. By alternating between propagator estimation and representation refinement, we ultimately learn the Koopman representation and its linear dynamics generator simultaneously.

[1]Here, $T_\mu\mathcal{P}_2 = \overline{\{\nabla\varphi : \varphi \in C_c^\infty(\mathbb{R}^D)\}}^{L^2(\mu)}$ is the tangent space; the overline denotes closure in $L^2(\mu)$.

Estimation of the linear propagator. We fix a set of observables $\{g_1, \dots, g_D\}$, which induces a Wasserstein flow of latent distributions $\mu_t = \boldsymbol{g}_{\#}\rho_t \in \mathcal{P}_2(\mathbb{R}^D)$, $t \in [0, T]$. We aim to find a linear generator $\boldsymbol{A}$ such that $(e^{\boldsymbol{A}\Delta t})_{\#}\mu_t \approx \mu_{t+\Delta t}$. Exact propagation under this linear flow satisfies the continuity equation $\partial_t \mu_t + \nabla_{\boldsymbol{z}} \cdot (\boldsymbol{A}\boldsymbol{z}\,\mu_t) = 0$. Accordingly, we introduce the *weak population residual* $r_{\boldsymbol{A}}(t) = \partial_t \mu_t + \nabla_{\boldsymbol{z}} \cdot (\boldsymbol{A}\boldsymbol{z}\,\mu_t)$ to quantify the mismatch between the linear dynamics and the evolution of μ_t. We then seek an $\boldsymbol{A}$ minimizing $\mathcal{J}_\mu(\boldsymbol{A}) = \frac{1}{2}\int_0^T \|r_{\boldsymbol{A}}(t)\|^2_{-1,\mu_t}\mathrm{d}t$. However, directly minimizing the residual is intractable, as it requires the density derivative $\partial_t \mu_t$. To this end, we use the equivalent min–max formulation in the dual test space, as established in Theorem A.3 (Appendix A.1):

$$\boldsymbol{A}^* = \arg\min_{\boldsymbol{A}} \mathcal{J}_\mu(\boldsymbol{A}) = \arg\min_{\boldsymbol{A}} \sup_{\varphi \in \mathbb{V}_D} \left\{ \mathcal{R}_D(\boldsymbol{A}; \varphi) - \frac{1}{2}\|\varphi\|^2_{\mathbb{V}_D} \right\}, \tag{1}$$
$$\text{where} \quad \mathcal{R}_D(\boldsymbol{A}; \varphi) = \mu_T[\varphi(T, \cdot)] - \mu_0[\varphi(0, \cdot)] - \int_0^T \mu_t\big[\partial_t \varphi(t, \boldsymbol{z}) + \nabla_{\boldsymbol{z}} \varphi(t, \boldsymbol{z})^\top \boldsymbol{A}\boldsymbol{z}\big]\,\mathrm{d}t.$$

Here $\mathbb{V}_D = L^2(0, T; \dot{H}^1(\mu_t))$ is the Hilbert space with $\|\varphi\|^2_{\mathbb{V}_D} = \int_0^T \mu_t[\|\nabla_{\boldsymbol{z}}\varphi\|^2]\mathrm{d}t$. The following theorem shows that minimizing $\mathcal{J}_\mu$ gives the best approximation of the canonical Wasserstein velocity $\boldsymbol{v}_t$ by projected linear fields in $L^2(\mathrm{d}t\,\mathrm{d}\mu_t)$. Furthermore, restricting the test space $\mathbb{V}_D$ to finitely many functions yields a least-squares problem which admits a closed-form solution.

Theorem 3.1. *Assume $\mu_\cdot \in AC^2([0, T]; \mathcal{P}_2(\mathbb{R}^D))$, and let $\boldsymbol{v}_t$ be its canonical velocity. Then, for every $\boldsymbol{A} \in \mathbb{R}^{D\times D}$,*

$$\mathcal{J}_\mu(\boldsymbol{A}) = \sup_{\varphi \in \mathbb{V}_D} \left\{ \mathcal{R}_D(\boldsymbol{A}; \varphi) - \tfrac{1}{2}\|\varphi\|^2_{\mathbb{V}_D} \right\} = \frac{1}{2}\int_0^T \|\boldsymbol{v}_t - P_{\mu_t}(\boldsymbol{A}\boldsymbol{z})\|^2_{L^2(\mu_t)}\mathrm{d}t. \tag{2}$$

Hence the minimization gives the orthogonal best approximation of the canonical velocity by projected linear velocity fields. Furthermore, restrict the supremum to $\mathbb{V}_{D,m} = \mathrm{span}\{\varphi_1, \dots, \varphi_m\}$ and define

$$\begin{aligned} y_j &= \mu_T[\varphi_j(T)] - \mu_0[\varphi_j(0)] - \int_0^T \mu_t[\partial_t \varphi_j(t)]\mathrm{d}t, \\ \boldsymbol{H}_j &= \int_0^T \mu_t[\nabla_{\boldsymbol{z}}\varphi_j(t, \boldsymbol{z})\boldsymbol{z}^\top]\mathrm{d}t, \qquad M_{ij} = \int_0^T \mu_t[\nabla_{\boldsymbol{z}}\varphi_i(t, \boldsymbol{z})^\top \nabla_{\boldsymbol{z}}\varphi_j(t, \boldsymbol{z})]\mathrm{d}t. \end{aligned} \tag{3}$$

Let $\boldsymbol{a} = \mathrm{vec}(\boldsymbol{A})$, let row j of $\boldsymbol{G}$ be $\mathrm{vec}(\boldsymbol{H}_j)^\top$, and let $\boldsymbol{y} = (y_1, \dots, y_m)^\top$. If $\boldsymbol{M} \succ 0$, restricting Equation (1) to $\mathbb{V}_{D,m}$ gives the weighted least-squares objective $\mathcal{J}_{D,m}(\boldsymbol{A}) = \frac{1}{2}(\boldsymbol{y} - \boldsymbol{G}\boldsymbol{a})^\top \boldsymbol{M}^{-1}(\boldsymbol{y} - \boldsymbol{G}\boldsymbol{a})$. If $\boldsymbol{G}$ has full column rank, its unique minimizer is $\mathrm{vec}(\widehat{\boldsymbol{A}}_{D,m}) = (\boldsymbol{G}^\top \boldsymbol{M}^{-1}\boldsymbol{G})^{-1}\boldsymbol{G}^\top \boldsymbol{M}^{-1}\boldsymbol{y}$. Finally, under the test-enrichment and uniqueness assumptions in Appendix A.2, these minimizers converge to $\boldsymbol{A}^*$, the unique population minimizer of Equation (1) as $m \to \infty$.

We provide the proof of Theorem 3.1 in Appendices A.1 and A.2. In practice, we use random Fourier tests to construct $\mathbb{V}_{D,m}$, reducing the min–max optimization to a weighted least-squares problem. For irregular snapshots at $0 = t_0 < t_1 < \cdots < t_N = T$, we formulate weak equations on each adjacent interval $I_k = [t_k, t_{k+1}]$ and approximate the integrals by the trapezoidal rule, $\int_{I_k} f(t)\mathrm{d}t \simeq \frac{t_{k+1}-t_k}{2}\{f(t_k) + f(t_{k+1})\}$. The integrands $f(\cdot) = \mu_\cdot[\cdot]$ are expectations under the latent distributions, approximated by empirical averages over the encoded snapshots. Finally, equations over different intervals are combined to form the overall regression problem, with ridge regularization used to stabilize the solve. Implementation details are provided in Appendix B.1.2. We further show in a Koopman-invariant RKHS that the minimum weak population residual vanishes as $D \to \infty$ (Appendix A.3.2), derive Wasserstein prediction bounds (Appendix A.4.1), and prove locally uniform strong semigroup convergence (Appendix A.4.4), under the stated approximation, stability, and identifiability assumptions. We also discuss theoretical limitations in Appendix A.4.4.

Updating the representation under the linear flow. Given the estimated propagator $\boldsymbol{A} = \widehat{\boldsymbol{A}}_{D,m}$, we update the representation to better fit the linear dynamics. To regularize the VAE representation, we use a prior transported by the linear flow: the initial prior $p_0 = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_D)$ evolves into $p_t = \mathcal{N}(\boldsymbol{0}, e^{\boldsymbol{A}t}e^{\boldsymbol{A}^\top t})$. We thus combine squared reconstruction error with the KL divergence (Csiszár, 1975) between the encoder posterior and this dynamic prior:

$$\mathcal{L}_{\mathrm{VAE}} = \mathbb{E}_{t,\boldsymbol{x}}\left[\frac{1}{2\sigma_x^2 d_x}\mathbb{E}_{\boldsymbol{Z}\sim q_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x},t)}\|\boldsymbol{x} - \mathrm{Dec}_{\boldsymbol{\phi}}(\boldsymbol{Z})\|^2 + \beta\,\mathrm{KL}\big(q_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}, t)\,\|\,p_t\big)\right]. \tag{4}$$

Here d_x is the expression dimension, σ_x^2 is the fixed observation variance, and $\mathbb{E}_{t,\boldsymbol{x}}$ averages over training times and cells. To encourage accurate population prediction in both latent and expression space, we augment $\mathcal{L}_{\mathrm{VAE}}$ with the population-matching loss $\mathcal{L}_{\mathrm{pop}} = \mathbb{E}_{s<t\in\mathcal{T}}[\lambda_z \mathcal{S}_\varepsilon(\mu_t, \widehat{\mu}_{s\to t}) + \lambda_x \mathcal{S}_\varepsilon(\rho_t, \widehat{\rho}_{s\to t})]$, where $\mu_t = \int q_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}, t)\mathrm{d}\rho_t(\boldsymbol{x})$ is the encoder-induced population, $\widehat{\mu}_{s\to t} = (e^{\boldsymbol{A}(t-s)})_{\#}\mu_s$ is the latent distribution propagated from μ_s to time t, and $\widehat{\rho}_{s\to t} = (\mathrm{Dec}_{\boldsymbol{\phi}})_{\#}\widehat{\mu}_{s\to t}$ is the decoded prediction. The expectation averages over all forward pairs of training times, and $\mathcal{S}_\varepsilon$ is the debiased Sinkhorn divergence with quadratic cost. We further regularize the encoder with the weak population residual loss $\mathcal{L}_{\mathrm{weak}} = \mathcal{J}_{D,m}(\widehat{\boldsymbol{A}}_{D,m})$, which penalizes violations of the weak continuity equation and encourages latent coordinates in which population evolution is better approximated by the estimated linear dynamics (Appendix B.1.2). The linear flow is evaluated by matrix exponentials, enabling prediction at arbitrary times without costly numerical ODE integration.

Training. Following Zhang et al. (2024), we pretrain the VAE with a standard-normal Gaussian prior, then alternate between generator estimation and representation learning. During each representation update, we hold $\boldsymbol{A}$ fixed and minimize $\mathcal{L}_{\mathrm{KC}} = \mathcal{L}_{\mathrm{VAE}} + \mathcal{L}_{\mathrm{pop}} + \lambda_{\mathrm{weak}}\mathcal{L}_{\mathrm{weak}}$ over the VAE parameters $\boldsymbol{\theta}, \boldsymbol{\phi}$. Algorithm 1 summarizes the training procedure of KoopCell.

3.2 KoopCell-M: Memory-Augmented Koopman Dynamics for Branching

Learning a linear latent flow is appealing, but it constrains the modeling of branching during cellular differentiation, in which an initially homogeneous population differentiates into multiple cell types. Specifically, an ODE with a Lipschitz vector field cannot exactly transport a source distribution supported on M disjoint, compact, path-connected components to a target with $N \neq M$ such components (Zhao et al., 2026, Proposition 3.3). Thus, when the initial population has connected latent support, the linear dynamics followed by a Lipschitz decoder cannot produce fate populations with disconnected support. Moreover, the linear flow itself cannot create new latent density modes. Although nonlinear decoding can introduce additional modes, bounding the decoder Lipschitz constant limits the amplification of latent separation (Appendix B.2.1). To model branching through an explicit dynamical mechanism, Zhao et al. (2026) introduce non-Markovian delay dynamics with distinct initial histories for the same state, allowing trajectories to diverge. Such non-Markovian dynamics also arise within Koopman theory through the Mori–Zwanzig formalism (Section 2), which describes how unresolved observables contribute memory to the resolved linear dynamics. In cellular differentiation, such unresolved influences may reflect heritable fate biases that are not fully captured by scRNA-seq (Weinreb et al., 2020). This connection motivates us to introduce memory into KoopCell, using non-Markovian dynamics to model branching within the same Koopman framework.

Memory embedding and branching. To obtain a tractable formulation of this non-Markovian model, we use the Markovian embedding from Section 2 to evolve the augmented latent state $(\boldsymbol{Z}_t, \boldsymbol{H}_t)$:

$$\mathrm{d}\boldsymbol{Z}_t = (\boldsymbol{A}\boldsymbol{Z}_t + \boldsymbol{B}\boldsymbol{H}_t + \boldsymbol{b}_z)\,\mathrm{d}t, \qquad \mathrm{d}\boldsymbol{H}_t = (\boldsymbol{C}\boldsymbol{Z}_t + \boldsymbol{\Lambda}\boldsymbol{H}_t + \boldsymbol{b}_h)\,\mathrm{d}t + \boldsymbol{\Sigma}_h\,\mathrm{d}\boldsymbol{W}_t. \tag{5}$$

Here $\boldsymbol{Z}_t \in \mathbb{R}^D$ is the *visible latent state* encoded from cellular gene expression, and $\boldsymbol{H}_t \in \mathbb{R}^{d_h}$ contains auxiliary memory variables. A joint Gaussian initialization of $(\boldsymbol{Z}_0, \boldsymbol{H}_0)$ yields Gaussian visible $\boldsymbol{Z}$-marginals. To enable branching in the latent distribution, we parameterize hidden initial distributions by a discrete variable $R \in \mathscr{R}$. Each value of R indexes a candidate root-to-terminal path in a directed acyclic graph (DAG) of coarse biological states. We detail the state annotations and graph construction in Appendix B.2.1 and show the resulting DAG for the mouse cell reprogramming dataset (SC; Schiebinger et al., 2019) in Figure 5. With R specifying the developmental path, we define a generative model with latent variables $(R, \boldsymbol{Z}_t, \boldsymbol{H}_t)$. We specify the initial distribution of the augmented dynamics in Equation (5) as

$$R \sim \mathrm{Cat}(\boldsymbol{\pi}), \qquad \boldsymbol{Z}_0 \mid R = r \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_D), \qquad \boldsymbol{H}_0 \mid \boldsymbol{Z}_0 = \boldsymbol{z}, R = r \sim \mathcal{N}(\boldsymbol{F}_r\boldsymbol{z} + \boldsymbol{a}_r, \boldsymbol{S}_r). \tag{6}$$

Under this parametrization, the augmented process is Gaussian conditional on R. The marginal distribution of $\boldsymbol{Z}_t$ is therefore a Gaussian mixture, $p_t(\boldsymbol{z}) = \sum_{r\in\mathscr{R}} \pi_r \mathcal{N}(\boldsymbol{z}; \boldsymbol{m}^z_{t,r}, \boldsymbol{V}^{zz}_{t,r})$, with propagated means $\boldsymbol{m}^z_{t,r}$ and covariances $\boldsymbol{V}^{zz}_{t,r}$ (Appendix B.2.2). Although all paths share the same visible initial distribution p_0, coupling through $\boldsymbol{B}\boldsymbol{H}_t$ can turn hidden heterogeneity into visible separation, producing multiple density modes when component means become sufficiently separated relative to their spread. We construct a toy example in Section 4.2 to illustrate how introducing memory enables the dynamics to generate distinct branches from an initially unimodal population (Figure 2).

Learning the representation and memory dynamics. As in KoopCell, we first pretrain the VAE, then alternate between dynamics estimation and representation learning. With the representation and

Table 1: Hard-task Wasserstein-2 distance (↓) at held-out time points (mean ± sample standard deviation over three training seeds). Shaded labels distinguish interpolation from extrapolation within each dataset. The best and second-best means are shaded green (bold) and blue, respectively.

Prediction	Data	t	scNODE	LGP-OT	MIOFlow	PRESCIENT	PI-SDE	VGFM	KoopCell	KoopCell-M
Interpolation	ZB	3	577.71 ± 3.69	577.91 ± 0.64	594.71 ± 32.03	597.49 ± 1.76	$\mathbf{567.30 \pm 1.41}$	575.86 ± 3.22	572.32 ± 2.14	588.91 ± 6.04
	ZB	5	511.58 ± 5.63	504.45 ± 1.73	519.32 ± 7.48	540.72 ± 1.52	492.43 ± 2.19	499.66 ± 0.81	497.80 ± 3.83	$\mathbf{488.84 \pm 9.27}$
	ZB	7	440.65 ± 6.47	411.20 ± 3.07	452.30 ± 0.77	476.68 ± 1.16	416.80 ± 3.36	434.99 ± 1.37	412.99 ± 1.50	$\mathbf{408.79 \pm 1.57}$
	ZB	9	514.24 ± 5.97	475.65 ± 2.60	539.92 ± 11.44	527.09 ± 1.25	474.41 ± 3.51	522.60 ± 2.83	474.92 ± 1.55	$\mathbf{468.53 \pm 2.72}$
Extrapolation	ZB	11	653.56 ± 8.98	605.14 ± 1.78	654.64 ± 2.17	656.41 ± 4.17	612.74 ± 3.51	635.96 ± 2.73	595.82 ± 0.48	$\mathbf{588.73 \pm 0.77}$
	ZB	12	719.28 ± 12.36	665.78 ± 4.77	720.67 ± 17.25	700.68 ± 4.58	679.61 ± 4.11	720.54 ± 10.97	648.91 ± 2.47	$\mathbf{639.78 \pm 0.72}$
Interpolation	DR	3	443.40 ± 5.78	433.15 ± 2.43	446.21 ± 8.45	475.42 ± 1.27	432.23 ± 2.64	455.79 ± 6.96	$\mathbf{429.70 \pm 3.54}$	430.41 ± 3.72
	DR	5	468.39 ± 0.68	465.22 ± 1.10	468.02 ± 1.97	492.41 ± 1.06	$\mathbf{457.57 \pm 2.85}$	466.01 ± 0.99	463.32 ± 0.60	466.04 ± 1.46
	DR	7	534.69 ± 8.30	519.13 ± 0.49	533.38 ± 1.03	541.43 ± 0.77	519.12 ± 0.67	526.89 ± 0.67	522.50 ± 2.41	$\mathbf{518.49 \pm 1.41}$
Extrapolation	DR	9	602.99 ± 5.41	567.25 ± 0.94	617.05 ± 5.86	604.33 ± 0.80	577.97 ± 0.83	592.17 ± 5.07	558.89 ± 2.21	$\mathbf{556.80 \pm 1.27}$
	DR	10	593.05 ± 2.68	570.52 ± 5.15	673.08 ± 24.93	565.45 ± 0.37	603.87 ± 2.94	653.24 ± 19.74	563.86 ± 3.26	$\mathbf{557.68 \pm 3.56}$
	DR	11	726.16 ± 5.89	713.37 ± 4.70	860.26 ± 87.00	716.82 ± 0.96	751.16 ± 5.70	851.07 ± 22.16	$\mathbf{694.53 \pm 4.21}$	699.49 ± 4.56
Interpolation	SC	6	54.44 ± 0.28	48.66 ± 0.47	53.87 ± 4.29	64.70 ± 0.88	$\mathbf{47.35 \pm 0.04}$	48.94 ± 0.34	50.20 ± 1.58	49.49 ± 0.63
	SC	8	58.45 ± 0.94	$\mathbf{52.79 \pm 0.22}$	63.28 ± 3.71	66.35 ± 0.52	53.03 ± 0.26	53.68 ± 0.80	54.00 ± 0.68	53.44 ± 0.28
	SC	10	103.23 ± 1.21	96.40 ± 0.28	110.39 ± 5.17	108.01 ± 0.22	$\mathbf{95.14 \pm 1.39}$	108.46 ± 1.37	96.72 ± 0.35	96.99 ± 0.32
	SC	12	142.30 ± 1.36	128.31 ± 1.26	156.91 ± 7.55	152.00 ± 2.76	137.35 ± 0.78	166.69 ± 3.40	127.47 ± 0.92	$\mathbf{127.33 \pm 0.37}$
Extrapolation	SC	16	128.25 ± 3.90	108.53 ± 1.37	160.34 ± 21.35	134.07 ± 0.64	125.81 ± 1.99	159.34 ± 1.88	109.78 ± 1.73	$\mathbf{105.61 \pm 1.48}$
	SC	17	142.40 ± 4.78	114.45 ± 1.06	195.20 ± 44.48	142.04 ± 1.57	147.12 ± 3.60	161.57 ± 2.74	118.54 ± 4.86	$\mathbf{111.87 \pm 0.64}$
	SC	18	131.79 ± 5.62	118.72 ± 3.29	203.72 ± 60.02	131.42 ± 1.30	140.50 ± 5.75	169.55 ± 4.90	109.89 ± 5.23	$\mathbf{108.69 \pm 2.28}$
	SC	19	145.91 ± 8.78	132.99 ± 4.67	243.97 ± 92.21	136.76 ± 1.76	160.55 ± 7.53	181.55 ± 9.57	121.87 ± 8.65	$\mathbf{117.16 \pm 3.84}$

memory statistics fixed, we estimate $\boldsymbol{A}, \boldsymbol{b}_z$ in closed form using the same least-squares formulation, after subtracting the memory contribution from the weak continuity equations (Appendix B.2.3).

With $\boldsymbol{A}, \boldsymbol{b}_z$ fixed, we jointly update the representation and remaining memory parameters using the distribution-matching objective $\mathcal{L}_{\text{pop}}$ over all training pairs $s < t$, as in KoopCell. Prediction from a snapshot at s additionally requires inferring R and $\boldsymbol{H}_s$. Our generative model supports tractable variational inference: given $\boldsymbol{z} \sim q_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}, s)$, we compute the path probabilities $q_s(r \mid \boldsymbol{z}, \boldsymbol{x})$ and Gaussian memory conditionals $p(\boldsymbol{H}_s \mid \boldsymbol{Z}_s = \boldsymbol{z}, R = r)$ analytically (Appendix B.2.2). We propagate the inferred augmented population under Equation (5), with transition means and covariances evaluated by matrix exponentials. The predicted $\boldsymbol{Z}_t$-marginal is $\widehat{\mu}_{s\to t}$, and decoding gives $\widehat{\rho}_{s\to t} = (\text{Dec}_{\boldsymbol{\phi}})_{\#}\widehat{\mu}_{s\to t}$. To encourage rollouts to follow the biological pathway specified by R, we introduce a state classifier $c_{\boldsymbol{\omega}}(c \mid \boldsymbol{z})$. We jointly train the VAE and classifier using the snapshot objective $\mathcal{L}_{\text{snap}}$, which combines reconstruction, KL regularization, and supervised classification on annotated training cells (Appendix B.2.1). We then evaluate $c_{\boldsymbol{\omega}}$ along each rollout and use the predicted state probabilities to compute the path consistency loss $\mathcal{L}_{\text{path}}$, which encourages progression through biological states in the order specified by R (Equation (77) in Appendix B.2.2). The overall training objective is $\mathcal{L}_{\text{snap}} + \mathcal{L}_{\text{pop}} + \lambda_R\mathcal{L}_{\text{path}} + \lambda_{\text{weak}}\mathcal{L}_{\text{weak}}$. Appendix B.2.2 derives these objectives, and Algorithm 2 summarizes the alternating learning procedure.

4 Experiments

We evaluate our models on synthetic systems and three single-cell RNA-sequencing datasets. Through these experiments, we answer the following three questions: **Q1.** Can KoopCell recover the underlying Koopman spectrum from sparse, unpaired snapshots, and does this recovery support stable extrapolation? **Q2.** How does incorporating memory improve the modeling of branching dynamics? **Q3.** Can our models scale to high-dimensional scRNA-seq data and learn from sparse, irregular snapshots to accurately predict gene expression distributions within and beyond the training horizon?

4.1 Spectral Recovery from Unpaired Snapshots

To test whether KoopCell can recover the underlying Koopman operator from unpaired snapshots, and whether the recovered spectral structure translates into more accurate temporal prediction, we consider two toy systems with known Koopman spectra that are widely studied in the literature:

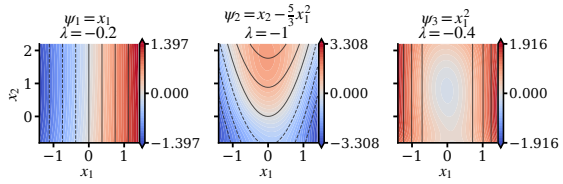
(Toy I) Following Tu et al. (2014), we consider $\dot{x}_1 = -0.2x_1$ and $\dot{x}_2 = -(x_2 - x_1^2)$. The system has the attracting slow manifold $x_2 = \frac{5}{3}x_1^2$ and admits the exact Koopman observables $\Psi_1 = x_1$, $\Psi_2 = x_2 - \frac{5}{3}x_1^2$, and $\Psi_3 = x_1^2$. Consequently, $\dot{\boldsymbol{\Psi}} = \boldsymbol{A}\boldsymbol{\Psi}$ with $\boldsymbol{A} = \text{diag}(-0.2, -1, -0.4)$.

(Toy II) Following Otto & Rowley (2019), we consider the unforced Duffing equation $\ddot{q} = -\delta\dot{q} - q(\beta + \alpha q^2)$ with $\delta = 0.5$, $\beta = -1$, and $\alpha = 1$. The system has two stable equilibria at $q = \pm 1$, with the conjugate intrawell eigenpair $\lambda_{1,2} = (-1 \pm \sqrt{31}i)/4$. Across the two wells, it also admits an eigenfunction with zero eigenvalue that separates the invariant basins.

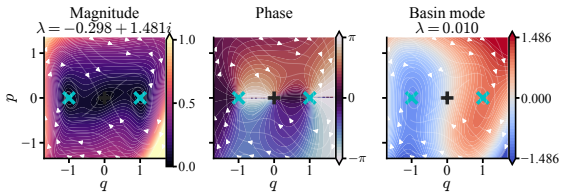
We generate 2D unpaired snapshots by solving these equations over short training horizons during which the populations remain far from equilibrium. For Toy I, training comprises 9 snapshots at $t \in [0, 1.2]$, and extrapolation is evaluated at 0.2-time-unit intervals from $t = 1.4$ to 4.0. For Toy II, training comprises 10 snapshots at $t = 0, 0.5, \ldots, 4.5$, and extrapolation is evaluated at $t \in \{5, 6, 8, 10, 12\}$. We also synthesize a 64D version of each dataset using $\boldsymbol{y} = \mathrm{softplus}(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{c}) + \boldsymbol{\epsilon} \in \mathbb{R}^{64}$, with $\boldsymbol{x} \in \mathbb{R}^2$ and Gaussian observation noise $\boldsymbol{\epsilon}$. Since the underlying systems are autonomous, we omit t from the encoder input. We compare KoopCell with scNODE (Zhang et al., 2024), which parameterizes the latent dynamics with a neural ODE. Both methods use the same latent dimension of $D = 3$ across all experiments. After training, we roll out each model from the observed initial distribution beyond the training horizon and report the mean sliced Wasserstein distance (SWD) to the ground-truth snapshots. Complete protocols and full results are provided in Appendix C.

Spectral recovery. Table 2 reports the recovered continuous-time spectrum of the learned matrix $\boldsymbol{A}$, while Figure 1 shows the corresponding eigenfunctions. KoopCell consistently recovers the Koopman spectrum with low MAE, and, for Toy I, the learned observables reproduce the level-set geometry of the ground-truth observables. For Toy II, we visualize the learned complex eigenfunction through its magnitude $|\Psi|$ and phase $\angle\Psi$ (Figure 1(b), left and middle). The magnitude decays toward zero at both attractors, $q = \pm 1$, while the phase reveals the oscillatory relaxation within each basin. Meanwhile, the eigenmode associated with the zero eigenvalue acts as an indicator that separates the two basins of attraction. Overall, these results demonstrate KoopCell's ability to recover the underlying spectral structure from unpaired, temporally sparse distribution snapshots.

Spectral structure supports stable extrapolation. Across all settings, KoopCell achieves lower extrapolation SWD than scNODE (Table 2). Together with the spectral recovery results, this suggests that learning the underlying Koopman spectral structure helps capture the long-term evolution of the system and supports more stable extrapolation beyond the training horizon.



(a) Toy I: learned eigenfunctions aligned with $\{x_1, x_2 - \frac{5}{3}x_1^2, x_1^2\}$.



(b) Toy II: learned oscillatory and basin eigenmodes.

Figure 1: Koopman eigenmodes learned from unpaired 2D snapshots. In (a), colors show the learned modes and black contours show the analytic eigenfunctions. In (b), the complex mode separates intrawell amplitude and phase, whereas the near-zero mode separates the two attraction basins; sink and saddle markers are included only for visualization.

Table 2: Spectral recovery (MAE) and extrapolation comparison with scNODE (SWD ↓).

		Toy I		Toy II	
Recovered Spectrum		$\{-0.195, -0.423, -0.977\}$		$\{0.010, -0.298 \pm 1.481i\}$	
Spectrum MAE		0.0167		0.0710	
	Model	Toy I		Toy II	
		2D	**64D**	**2D**	**64D**
SWD	scNODE	0.1936	0.2011	0.1076	0.0277
	KoopCell	**0.0665**	**0.0641**	**0.0592**	**0.0175**

4.2 Modeling Branching Dynamics with KoopCell-M

We compare KoopCell and KoopCell-M on a stochastic pitchfork system in which an initially unimodal population separates into two branches. A fixed nonlinear map produces 64D noisy observations at nine training times. We initialize both models with the same pretrained autoencoder and jointly train their autoencoders and dynamics using the same population-matching objective. We impose the same Lipschitz bound on both decoders to limit the amplification of latent separation

during decoding. Figure 2 overlays the population rollouts, while Table 3 summarizes SWD and branch-coordinate W_1 over all held-out times; definitions of both metrics and experimental details are provided in Appendix D. KoopCell-M more closely reproduces the separation of the two branches and achieves lower errors on both metrics, as expected.

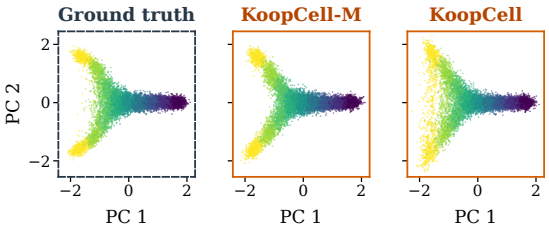


Figure 2: Population rollouts at all nine training times in a shared PCA plane (1,024 cells per time). Colors progress from purple to yellow with time; all panels use identical axes.

Table 3: Distributional prediction errors averaged over held-out times (mean ± SD over three optimization seeds).

Model	SWD ↓	Branch W_1 ↓
KoopCell	0.1557 ± 0.0146	0.2245 ± 0.0069
KoopCell-M	$\mathbf{0.1124 \pm 0.0046}$	$\mathbf{0.0866 \pm 0.0026}$

4.3 Modeling High-dimensional Single-Cell Population Dynamics

To assess the scalability of KoopCell and KoopCell-M, we evaluate them on three high-dimensional single-cell RNA-sequencing datasets: zebrafish embryogenesis (ZB) (Farrell et al., 2018), Drosophila embryogenesis (DR) (Calderon et al., 2022), and mouse iPSC reprogramming (SC) (Schiebinger et al., 2019). Dataset descriptions, preprocessing, and visualizations are provided in Appendix E.1.

Tasks and evaluation protocol. Following Zhang et al. (2024), we evaluate our models on three tasks of increasing difficulty to assess their ability to predict gene expression distributions at held-out time points, both within and beyond the training horizon. For each task, we train on the populations at the retained time points. At evaluation, we sample cells from the initial snapshot, propagate their encoded states forward, and decode the predicted populations at the held-out times. Generated and observed distributions are compared in gene-expression space using the Wasserstein distance W_2. The tasks differ in which snapshots are withheld: **(1) Easy (interpolation)** removes selected intermediate snapshots to assess prediction within the training horizon; **(2) Medium (extrapolation)** removes several final snapshots to assess prediction beyond that horizon; and **(3) Hard (interpolation and extrapolation)** removes both intermediate and several final snapshots to assess the ability to recover population dynamics from highly sparse, unpaired distribution snapshots. Figure 12 shows the training and held-out time points.

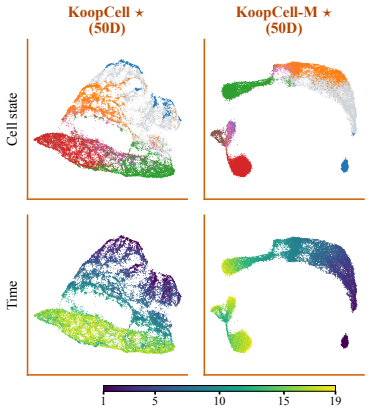


Figure 3: Latent representations of SC cells colored by reference state (top; stromal: green, iPSC: red) and snapshot index (bottom).

Baselines. We compare against several representative models for temporal dynamics of single-cell populations, including **scNODE** (Zhang et al., 2024), **LGP-OT** (Balik & Lähdesmäki, 2026), **MIOFlow** (Huguet et al., 2022), **PRESCIENT** (Yeo et al., 2021), **PI-SDE** (Jiang & Wan, 2024), and **VGFM** (Wang et al., 2025). KoopCell(-M), scNODE and LGP-OT jointly learn an informative latent representation and the latent dynamics, while other models learn dynamics in a fixed latent space obtained by dimensionality reduction. We tune all models to their optimal hyperparameters (Appendix E.2) and report the mean and std over three training seeds.

Interpolation and extrapolation performance. KoopCell achieves strong predictive performance across all three datasets and tasks (Table 1; Appendix Tables 10 and 11). While some baselines attain comparable or slightly lower errors at individual interpolation times, KoopCell achieves lower average W_2 distances than all baselines on both the medium task and the extrapolation portion of the hard task. Incorporating memory further enhances predictive performance compared to the linear model (KoopCell). Relative to the strongest baseline for each dataset, KoopCell-M reduces the average error by 3.6–7.3% on the medium task and 2.0–6.6% on the extrapolation portion of the hard task. These results highlight the ability of our models to learn population dynamics from highly sparse, unpaired

snapshots and extrapolate reliably beyond the training horizon. Appendix Figures 13–15 provide the corresponding visualization comparisons of predicted cell distributions.

Dynamics Shapes Representation. Figure 3 compares latent representations learned by KoopCell and KoopCell-M on the SC dataset under the hard-task setting. The complete comparison of latent representations, including those from PCA, scNODE, and MIOFlow, appears in Appendix Figure 16. We match cells to published Waddington-OT reference states and label the initial population as mouse embryonic fibroblasts (MEFs), leaving unmatched or ambiguous cells gray. Reprogramming involves divergence toward stromal states or mesenchymal-to-epithelial transition (MET), from which iPSCs and alternative neural- and trophoblast-like states emerge (Schiebinger et al., 2019). KoopCell-M more clearly separates stromal and iPSC populations with distinct temporal progressions, whereas these late populations overlap substantially in KoopCell. These qualitative patterns are consistent with the branching structure of the memory-augmented model.

Biological Downstream Analysis. Following Zhang et al. (2024), we examine lineage-associated genes in ZB using KoopCell-M fitted to all snapshots, focusing on Presomitic Mesoderm (PSM) and Hindbrain. Using the driver-gene statistic from CellRank (Weiler et al., 2024), we correlate preterminal gene expression with predicted fate probabilities and recover *TBX16*/*MSGN1* for PSM and *SOX3*/*SOX19A* for Hindbrain (Appendix Figure 17). We further characterize dynamic gene-expression programs along model-predicted trajectories over 8–12 hours post fertilization (hpf). Gene modules with similar predicted expression-rate profiles are enriched for muscle and neural development in the PSM and Hindbrain lineages, respectively. Appendix E.5 provides methods and additional results. Together, these analyses support the biological interpretability of our model.

5 Related Work

Learning scRNA-seq dynamics from distribution snapshots. Several approaches have been proposed to infer temporal dynamics from unpaired distributions. TrajectoryNet (Tong et al., 2020) fits continuous normalizing flows regularized by dynamic OT in fixed PCA coordinates, while PRESCIENT and PI-SDE (Yeo et al., 2021; Jiang & Wan, 2024) model potential-driven and Hamilton–Jacobi-regularized SDEs, respectively. MIOFlow (Huguet et al., 2022) fits neural ODEs in a pretrained geodesic embedding; scNODE (Zhang et al., 2024) jointly trains a VAE and latent neural ODE through dynamic regularization, and LGP-OT (Balik & Lähdesmäki, 2026) learns a latent Gaussian process and a decoder using an OT objective. Additionally, proliferation and death are accommodated through unbalanced OT (Zhang et al., 2025; Sha et al., 2024; Sun et al., 2025). VGFM (Wang et al., 2025) and WFR-FM (Peng et al., 2026) further combine unbalanced OT with flow matching (Lipman et al., 2023; Tong et al., 2024a;b) for scalable joint velocity and growth estimation.

Koopman learning and memory. Koopman theory (Koopman, 1931) describes nonlinear dynamics through the linear evolution of observables, motivating the joint learning of autoencoders and latent linear propagators for dynamical prediction and time-series generation (Otto & Rowley, 2019; Lusch et al., 2018; Liu et al., 2023; Naiman et al., 2024). While these methods learn from paired time-series data, we develop theory and algorithms for generative Koopman learning from unpaired distribution snapshots. Finding observables with exact linear closure remains challenging. Recent work uses the Mori–Zwanzig formalism (Mori, 1965; Zwanzig, 1961) to learn non-Markovian memory closures to account for unresolved variables and improve dynamical prediction (Lin et al., 2021; Menier et al., 2025; Gupta et al., 2025; Lu et al., 2026). Building on this principle, we learn a memory closure through an augmented Markovian system to model branching population dynamics from snapshots.

6 Conclusions

We introduced KoopCell, a unified framework for learning Koopman generative dynamics from unpaired distribution snapshots. The weak population residual in Wasserstein space yields a closed-form least-squares estimator of the linear generator with convergence guarantees, while a Markovian memory embedding further enables KoopCell-M to capture branching dynamics. Our theoretical and empirical findings demonstrate the ability to recover Koopman spectra and improve extrapolation of gene-expression distributions from sparse, unpaired measurements, supporting quantitative analysis of developmental dynamics beyond the observed time window. Future directions include applications to multi-omics data and learning stochastic Koopman operators from distribution snapshots.

AI USE STATEMENT

In this work, we used generative AI tools to assist with code implementation and debugging, including baseline implementations. We also used these tools to help verify the correctness and technical details of our mathematical arguments and algorithms, and to support synthetic data generation, experimental visualizations, and language polishing. We did not use generative AI tools to generate research ideas, develop the theoretical framework, formulate the main mathematical results, or design the core algorithms. We reviewed all AI-assisted work and take full responsibility for the final content, including the mathematical arguments, code, data, figures, and text.

ETHICS STATEMENT

This work develops computational methods for modeling cellular population dynamics. Our experiments use synthetic data and publicly available single-cell datasets from zebrafish, Drosophila, and mouse, and involve no new biological sample collection or human participants. We anticipate no foreseeable adverse societal effects from this work.

REPRODUCIBILITY STATEMENT

We facilitate reproducibility by documenting the assumptions and proofs of our theoretical results in Appendix A, and the estimation and training procedures for KoopCell and KoopCell-M, including pseudocode, in Appendix B. Synthetic data generation and experimental protocols are described in Appendices C and D. Single-cell dataset sources and preprocessing are provided in Appendix E.1, with training and held-out time points, evaluation metrics in Appendix E.1.1. Baseline configurations and model-selection procedures are documented in Appendix E.2, and the biological downstream analyses are detailed in Appendix E.5. We will release our code as open source upon acceptance of the paper.

REFERENCES

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient Flows in Metric Spaces and in the Space of Probability Measures*. Lectures in Mathematics ETH Zürich. Springer, 2. ed edition, 2008. ISBN 978-3-7643-8722-8 978-3-7643-8721-1. OCLC: 254181287.

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

Francisco J Ayala, Michael E Gilpin, and Joan G Ehrenfeld. Competition between species: theoretical models and experimental tests. *Theoretical Population Biology*, 4(3):331–356, 1973.

Mehmet Yigit Balik and Harri Lähdesmäki. Modeling temporal scRNA-seq data with latent Gaussian process and optimal transport. In *Forty-Third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=8RE5i7g4uB.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 57(1):289–300, January 1995. ISSN 1467-9868. doi: 10.1111/j.2517-6161.1995.tb02031.x. URL http://dx.doi.org/10.1111/j.2517-6161.1995.tb02031.x.

Elizabeth I. Boyle, Shuai Weng, Jeremy Gollub, Heng Jin, David Botstein, J. Michael Cherry, and Gavin Sherlock. GO::TermFinder—open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. *Bioinformatics*, 20(18):3710–3715, August 2004. ISSN 1367-4803. doi: 10.1093/bioinformatics/bth456. URL http://dx.doi.org/10.1093/bioinformatics/bth456.

Diego Calderon, Ronnie Blecher-Gonen, Xingfan Huang, Stefano Secchia, James Kentro, Riza M. Daza, Beth Martin, Alessandro Dulja, Christoph Schaub, Cole Trapnell, Erica Larschan, Kate M. O'Connor-Giles, Eileen E. M. Furlong, and Jay Shendure. The continuum of *Drosophila* embryonic development at single-cell resolution. *Science*, 377(6606):eabn5800, 2022. doi: 10.1126/science.abn5800. URL https://www.science.org/doi/abs/10.1126/science.abn5800.

Geng Chen, Baitang Ning, and Tieliu Shi. Single-cell RNA-Seq technologies and related computational data analysis. *Frontiers in Genetics*, 10:317, 2019.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31, 2018.

Shawn J Cokus, Maricruz De La Torre, Eric F Medina, Jeffrey P Rasmussen, Joselyn Ramirez-Gutierrez, Alvaro Sagasti, and Fang Wang. Tissue-specific transcriptomes reveal gene expression trajectories in two maturing skin epithelial layers in zebrafish embryos. *G3 Genes|Genomes|Genetics*, 9(10):3439–3452, October 2019. ISSN 2160-1836. doi: 10.1534/g3.119.400402. URL http://dx.doi.org/10.1534/g3.119.400402.

Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, pp. 146–158, 1975.

C. Domínguez Conde, C. Xu, L. B. Jarvis, D. B. Rainbow, S. B. Wells, T. Gomes, S. K. Howlett, O. Suchanek, K. Polanski, H. W. King, L. Mamanova, N. Huang, P. A. Szabo, L. Richardson, L. Bolt, E. S. Fasouli, K. T. Mahbubani, M. Prete, L. Tuck, N. Richoz, Z. K. Tuong, L. Campos, H. S. Mousa, E. J. Needham, S. Pritchard, T. Li, R. Elmentaite, J. Park, E. Rahmani, D. Chen, D. K. Menon, O. A. Bayraktar, L. K. James, K. B. Meyer, N. Yosef, M. R. Clatworthy, P. A. Sims, D. L. Farber, K. Saeb-Parsy, J. L. Jones, and S. A. Teichmann. Cross-tissue immune cell analysis reveals tissue-specific features in humans. *Science*, 376(6594), May 2022. ISSN 1095-9203. doi: 10.1126/science.abl5197. URL https://doi.org/10.1126/science.abl5197.

Klaus-Jochen Engel and Rainer Nagel. *One-Parameter Semigroups for Linear Evolution Equations*, volume 194 of *Graduate Texts in Mathematics*. Springer, 2000. doi: 10.1007/b97696.

Jeffrey A. Farrell, Yiqun Wang, Samantha J. Riesenfeld, Karthik Shekhar, Aviv Regev, and Alexander F. Schier. Single-cell reconstruction of developmental trajectories during zebrafish embryogenesis. *Science*, 360(6392):eaar3131, 2018. doi: 10.1126/science.aar3131. URL https://www.science.org/doi/abs/10.1126/science.aar3131.

Gene Ontology Consortium. The Gene Ontology knowledgebase in 2026. *Nucleic Acids Research*, 54(D1):D1779–D1792, January 2026. ISSN 1362-4962. doi: 10.1093/nar/gkaf1292. URL http://dx.doi.org/10.1093/nar/gkaf1292.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning – ICML '06*, ICML '06, pp. 369–376. ACM Press, 2006. doi: 10.1145/1143844.1143891. URL http://dx.doi.org/10.1145/1143844.1143891.

Christopher Heje Grønbech, Maximillian Fornitz Vording, Pascal N Timshel, Casper Kaae Sønderby, Tune H Pers, and Ole Winther. scVAE: variational auto-encoders for single-cell gene expression data. *Bioinformatics*, 36(16):4415–4422, 2020.

Vincent Guan, Joseph Janssen, Hossein Rahmani, Andrew Warren, Stephen Zhang, Elina Robeva, and Geoffrey Schiebinger. Identifying drift, diffusion, and causal structure from temporal snapshots. *arXiv preprint arXiv:2410.22729*, 2024.

Priyam Gupta, Peter Schmid, Denis Sipp, Taraneh Sayadi, and Georgios Rigas. Mori–Zwanzig latent space Koopman closure for nonlinear autoencoder. *Proceedings of the Royal Society A*, 481(2313): 20240259, 2025.

Carmen Hijón, Pep Español, Eric Vanden-Eijnden, and Rafael Delgado-Buscalioni. Mori–Zwanzig formalism as a practical computational tool. *Faraday Discussions*, 144:301–322, 2010. doi: 10.1039/B902479B.

Guillaume Huguet, Daniel Sumner Magruder, Alexander Tong, Oluwadamilola Fasina, Manik Kuchroo, Guy Wolf, and Smita Krishnaswamy. Manifold interpolating optimal-transport flows for trajectory inference. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=ahAEhOtVif.

Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. Single-cell RNA sequencing technologies and bioinformatics pipelines. *Experimental & Molecular Medicine*, 50(8):1–14, 2018.

Masahiro Ikeda, Isao Ishikawa, and Corbinian Schlosser. Koopman and Perron–Frobenius operators on reproducing kernel Banach spaces. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 32(12), 2022.

Qi Jiang and Lin Wan. A physics-informed neural SDE network for learning cellular dynamics from time-series scRNA-seq data. *Bioinformatics*, 40(Supplement_2):ii120–ii127, 09 2024. ISSN 1367-4811. doi: 10.1093/bioinformatics/btae400. URL https://doi.org/10.1093/bioinformatics/btae400.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Frederik Köhne, Friedrich M. Philipp, Manuel Schaller, Anton Schiela, and Karl Worthmann. L^∞-error bounds for approximations of the Koopman operator by kernel extended dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 24(1):501–529, 2025. doi: 10.1137/24M1650120. URL https://doi.org/10.1137/24M1650120.

Bernard O Koopman. Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

Bernard O Koopman and J. von Neumann. Dynamical systems of continuous spectra. *Proceedings of the National Academy of Sciences*, 18(3):255–263, 1932.

Milan Korda and Igor Mezić. On convergence of extended dynamic mode decomposition to the Koopman operator. *Journal of Nonlinear Science*, 28(2):687–710, 2018. ISSN 1432-1467. doi: 10.1007/s00332-017-9423-0. URL https://doi.org/10.1007/s00332-017-9423-0.

Chrysoula Kosma, Giannis Nikolentzos, George Panagopoulos, Jean-Marc Steyaert, and Michalis Vazirgiannis. Neural ordinary differential equations for modeling epidemic spreading. *Transactions on Machine Learning Research*, 2023.

Vladimir R Kostic, Pietro Novelli, Andreas Maurer, Carlo Ciliberto, Lorenzo Rosasco, and Massimiliano Pontil. Learning dynamical systems via Koopman operator regression in reproducing kernel Hilbert spaces. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=Tz1lknIPVfp.

Huan Lei, Nathan A. Baker, and Xiantao Li. Data-driven parameterization of the generalized Langevin equation. *Proceedings of the National Academy of Sciences*, 113(50):14183–14188, November 2016. ISSN 1091-6490. doi: 10.1073/pnas.1609587113. URL http://dx.doi.org/10.1073/pnas.1609587113.

Yen Ting Lin, Yifeng Tian, Daniel Livescu, and Marian Anghel. Data-driven learning for the Mori–Zwanzig formalism: A generalization of the Koopman learning framework. *SIAM Journal on Applied Dynamical Systems*, 20(4):2558–2601, 2021.

Yue Ling, Peiqi Zhang, Zhenyi Zhang, and Peijie Zhou. CellStream: Dynamical optimal transport informed embeddings for reconstructing cellular trajectories from snapshots data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 746–754, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with Koopman predictors. *Advances in Neural Information Processing Systems*, 36:12271–12290, 2023.

Liwei Lu, Zhijun Zeng, Yan Jiang, Yi Zhu, and Pipi Hu. Weak collocation regression method: fast reveal hidden stochastic dynamics from high-dimensional aggregate data. *Journal of Computational Physics*, 502:112799, 2024.

Wanfeng Lu, He Ma, Wei Lin, and Qunxi Zhu. Interpretable functional Koopman learning with non-Markovian closure for spatiotemporal systems. In *Forty-Third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=lquDiBCgNJ.

Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1):4950, 2018.

Alyssa J. Manning and David Kimelman. Tbx16 and Msgn1 are required to establish directional cell migration of zebrafish mesodermal progenitors. *Developmental Biology*, 406(2):172–185, October 2015. ISSN 0012-1606. doi: 10.1016/j.ydbio.2015.09.001. URL http://dx.doi.org/10.1016/j.ydbio.2015.09.001.

Emmanuel Menier, Sebastian Kaltenbach, Mouadh Yagoubi, Marc Schoenauer, and Petros Koumoutsakos. Interpretable learning of effective dynamics for multiscale systems. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 481(2306), 2025.

Kevin R Moon, David Van Dijk, Zheng Wang, Scott Gigante, Daniel B Burkhardt, William S Chen, Kristina Yim, Antonia van den Elzen, Matthew J Hirn, Ronald R Coifman, et al. Visualizing structure and transitions in high-dimensional biological data. *Nature Biotechnology*, 37(12):1482–1492, 2019.

Hazime Mori. Transport, collective motion, and Brownian motion. *Progress of Theoretical Physics*, 33(3):423–455, 1965.

Ilan Naiman, N Benjamin Erichson, Pu Ren, Michael W Mahoney, and Omri Azencot. Generative modeling of regular and irregular time series data via Koopman VAEs. In *International Conference on Learning Representations*, volume 2024, pp. 50547–50573, 2024.

Daniel P. S. Osborn, Kuoyu Li, Stephen J. Cutty, Andrew C. Nelson, Fiona C. Wardle, Yaniv Hinits, and Simon M. Hughes. Fgf-driven Tbx protein activities directly induce myf5 and myod to initiate zebrafish myogenesis. *Development*, 147(8):dev184689, April 2020. ISSN 0950-1991. doi: 10.1242/dev.184689. URL http://dx.doi.org/10.1242/dev.184689.

Samuel E. Otto and Clarence W. Rowley. Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, 2019. doi: 10.1137/18M1177846. URL https://doi.org/10.1137/18M1177846.

Qiangwei Peng, Zihan Wang, Junda Ying, Yuhao Sun, Qing Nie, Lei Zhang, Tiejun Li, and Peijie Zhou. WFR-FM: Simulation-free dynamic unbalanced optimal transport. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=1nqu7bK1mm.

Christy Cortez Rossi, Takao Kaji, and Kristin Bruk Artinger. Transcriptional control of Rohon–Beard sensory neuron development at the neural plate border. *Developmental Dynamics*, 238(4):931–943, March 2009. ISSN 1097-0177. doi: 10.1002/dvdy.21915. URL http://dx.doi.org/10.1002/dvdy.21915.

Leyla Ruzicka, Douglas G Howe, Sridhar Ramachandran, Sabrina Toro, Ceri E Van Slyke, Yvonne M Bradford, Anne Eagle, David Fashena, Ken Frazer, Patrick Kalita, Prita Mani, Ryan Martin, Sierra Taylor Moxon, Holly Paddock, Christian Pich, Kevin Schaper, Xiang Shao, Amy Singer, and Monte Westerfield. The Zebrafish Information Network: new support for non-coding genes, richer Gene Ontology annotations and the Alliance of Genome Resources. *Nucleic Acids Research*, 47(D1):D867–D873, January 2019. ISSN 1362-4962. doi: 10.1093/nar/gky1090. URL http://dx.doi.org/10.1093/nar/gky1090.

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.

Yuhao Sun, Zhenyi Zhang, Zihan Wang, Tiejun Li, and Peijie Zhou. Variational regularized unbalanced optimal transport: Single network, least action. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Iguyg0LULD.

Monica Tambalo, Richard Mitter, and David G. Wilkinson. A single cell transcriptome atlas of the developing zebrafish hindbrain. *Development*, 147(6):dev184143, March 2020. ISSN 0950-1991. doi: 10.1242/dev.184143. URL http://dx.doi.org/10.1242/dev.184143.

Alexander Tong, Jessie Huang, Guy Wolf, David van Dijk, and Smita Krishnaswamy. TrajectoryNet: A dynamic optimal transport network for modeling cellular dynamics. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024a. ISSN 2835-8856. URL https://openreview.net/forum?id=CD9Snc73AW. Expert Certification.

Alexander Y. Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-free Schrödinger bridges via score and flow matching. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li (eds.), *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 1279–1287. PMLR, 02–04 May 2024b. URL https://proceedings.mlr.press/v238/tong24a.html.

Jonathan H. Tu, Clarence W. Rowley, Dirk M. Luchtenburg, Steven L. Brunton, and J. Nathan Kutz. On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, 1(2):391–421, 2014. ISSN 2158-2491. doi: 10.3934/jcd.2014.1.391. URL https://www.aimsciences.org/article/id/1dfebc20-876d-4da7-8034-7cd3c7ae1161.

Dongyi Wang, Yuanwei Jiang, Zhenyi Zhang, Xiang Gu, Peijie Zhou, and Jian Sun. Joint velocity-growth flow matching for single-cell dynamics modeling. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 160552–160588. Curran Associates, Inc., 2025. doi: 10.52202/085713-5363. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/eb1bad7a84ef68a64f1afd6577725d45-Paper-Conference.pdf.

Xiaomei Wang, Qi An, Zilong He, and Wei Fang. A literature review of social network analysis in epidemic prevention and control. *Complexity*, 2021(1):3816221, 2021.

Philipp Weiler, Marius Lange, Michal Klein, Dana Pe'er, and Fabian Theis. CellRank 2: unified fate mapping in multiview single-cell data. *Nature Methods*, 21(7):1196–1205, 2024. ISSN 1548-7105. doi: 10.1038/s41592-024-02303-9. URL http://dx.doi.org/10.1038/s41592-024-02303-9.

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), February 2020. ISSN 1095-9203. doi: 10.1126/science.aaw3381. URL http://dx.doi.org/10.1126/science.aaw3381.

Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

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.

Zhenyi Zhang, Tiejun Li, and Peijie Zhou. Learning stochastic dynamics from snapshots through regularized unbalanced optimal transport. In *International Conference on Learning Representations*, volume 2025, pp. 19888–19919, 2025.

Bolin Zhao, Xiaoyu Zhang, Yuting Dong, Xin Lu, Wei Lin, and Qunxi Zhu. Delay flow matching. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=6lH1XblLpo.

Robert Zwanzig. Memory effects in irreversible thermodynamics. *Physical Review*, 124(4):983, 1961.

Robert Zwanzig. Nonlinear generalized Langevin equations. *Journal of Statistical Physics*, 9(3): 215–220, 1973.

APPENDIX CONTENTS

A Theory **17**
- A.1 Dual formulation and geometry of the weak population residual . . . 17
  - A.1.1 Canonical velocity and the space–time test space . . . 17
  - A.1.2 Residual functional and minimum-flux geometry . . . 18
- A.2 Finite tests, closed-form regression, and convergence under test enrichment . . . 19
- A.3 Convergence of the weak population residual . . . 22
  - A.3.1 Preliminaries: Koopman-invariant RKHS and linear feature dynamics . . . 22
  - A.3.2 Convergence of the weak population residual . . . 22
- A.4 From weak residual consistency to strong semigroup convergence . . . 24
  - A.4.1 The weak residual controls probability prediction . . . 24
  - A.4.2 Distributional identifiability of the linear propagator . . . 25
  - A.4.3 From distributional compactness to strong operator compactness . . . 26
  - A.4.4 Locally uniform strong convergence of the learned propagators . . . 27

B Algorithmic Details **29**
- B.1 KoopCell: Koopman dynamics and alternating training . . . 29
  - B.1.1 VAE representation learning . . . 29
  - B.1.2 Empirical weak residual and closed-form dynamics estimation . . . 30
- B.2 KoopCell-M: memory-augmented branching dynamics and alternating training . . . 32
  - B.2.1 Memory-augmented dynamics and developmental paths . . . 32
  - B.2.2 Variational inference and representation learning . . . 35
  - B.2.3 Closed-form dynamics estimation and alternating training . . . 37

C Spectral Recovery Experiments **39**
- C.1 Toy 1: finite-dimensional nonlinear flow . . . 39
- C.2 Toy 2: damped Duffing flow . . . 42
- C.3 Rollout-error accumulation . . . 43

D Branching Dynamics Experiments **44**

E Single-Cell Experimental Details **45**
- E.1 Single-cell datasets and preprocessing . . . 45
  - E.1.1 Population prediction and evaluation . . . 46
- E.2 Implementation of Baseline Models . . . 47
- E.3 Model configurations . . . 48
- E.4 Additional Single-Cell Results . . . 49
  - E.4.1 Predicted cell populations at held-out time points . . . 50
  - E.4.2 Visualization of the learned latent representations . . . 54
- E.5 Biological Downstream Analysis . . . 54

A THEORY

This appendix develops the theoretical foundations of estimating the linear generator based on the weak population residual, introduced in Section 3.1. Sections A.1–A.2 prove Theorem 3.1: minimizing the weak population residual gives the best projected linear approximation to the canonical Wasserstein velocity, and finite tests yield a least-squares regression problem whose minimizers converge under dense test enrichment and uniqueness. We then consider increasing the observable dimension in a Koopman-invariant RKHS. Section A.3 shows that the minimum weak population residual vanishes as $D \to \infty$; Section A.4 relates this consistency to Wasserstein prediction bounds and locally uniform strong convergence of the learned propagators, under the respective approximation, stability, and identifiability assumptions. Algorithmic details and pseudocode are provided in Appendix B.

A.1 DUAL FORMULATION AND GEOMETRY OF THE WEAK POPULATION RESIDUAL

Throughout Sections A.1–A.2, the observable dimension D is fixed. Following Section 3.1, we write $\mu_t \in \mathcal{P}_2(\mathbb{R}^D)$ for the latent distribution induced by the chosen observables at time t. We work over real Hilbert spaces; complex-valued tests can be handled by separating real and imaginary parts.

A.1.1 CANONICAL VELOCITY AND THE SPACE–TIME TEST SPACE

Assume

$$\mu_\cdot \in AC^2([0,T];\mathcal{P}_2(\mathbb{R}^D)). \tag{7}$$

Then there is a unique minimum-norm velocity $\boldsymbol{v}_t \in T_{\mu_t}\mathcal{P}_2$ such that

$$\partial_t \mu_t + \nabla\cdot(\boldsymbol{v}_t\mu_t) = 0, \qquad \int_0^T \|\boldsymbol{v}_t\|^2_{L^2(\mu_t)}\mathrm{d}t < \infty. \tag{8}$$

Here

$$T_\mu\mathcal{P}_2 = \overline{\{\nabla\phi : \phi \in C_c^\infty(\mathbb{R}^D)\}}^{L^2(\mu)}, \tag{9}$$

and P_μ denotes the $L^2(\mu)$-orthogonal projection onto this tangent space. We collect these spaces along the curve in

$$\mathscr{T}_\mu = L^2(0,T;T_{\mu_t}\mathcal{P}_2)\,, \qquad \|\boldsymbol{q}\|^2_{\mathscr{T}_\mu} = \int_0^T\int \|\boldsymbol{q}_t(\boldsymbol{z})\|^2\mathrm{d}\mu_t(\boldsymbol{z})\mathrm{d}t. \tag{10}$$

Let $\mathscr{D}$ consist of C^1 space–time potentials compactly supported in space uniformly in t. We identify two potentials whose spatial gradients agree in $L^2(\mathrm{d}t\,\mathrm{d}\mu_t)$ and define $\mathbb{V}_D$ as the completion under

$$\|\varphi\|^2_{\mathbb{V}_D} = \int_0^T\int \|\nabla_{\boldsymbol{z}}\varphi(t,\boldsymbol{z})\|^2\mathrm{d}\mu_t(\boldsymbol{z})\mathrm{d}t. \tag{11}$$

Lemma A.1 (Density of smooth space–time gradients)**.** *The map $\varphi \mapsto \nabla_{\boldsymbol{z}}\varphi$ extends to an isometric isomorphism from $\mathbb{V}_D$ onto $\mathscr{T}_\mu$. In particular, finite sums*

$$\sum_{\ell=1}^{L}\eta_\ell(t)\nabla\psi_\ell(\boldsymbol{z}), \qquad \eta_\ell \in C^1([0,T]), \quad \psi_\ell \in C_c^\infty(\mathbb{R}^D), \tag{12}$$

are dense in $\mathscr{T}_\mu$.

Proof. The gradient map is an isometry from the quotient of $\mathscr{D}$ into $\mathscr{T}_\mu$: spatial mollification approximates each C_c^1 potential by smooth compactly supported potentials with uniformly convergent gradients. To prove density, let $\boldsymbol{q} \in \mathscr{T}_\mu$ be orthogonal to every field in Equation (12). For fixed $\psi \in C_c^\infty$, set

$$F_\psi(t) = \int \boldsymbol{q}_t(\boldsymbol{z})^\top\nabla\psi(\boldsymbol{z})\mathrm{d}\mu_t(\boldsymbol{z}).$$

Cauchy–Schwarz and boundedness of $\nabla\psi$ imply $F_\psi \in L^1(0,T)$. Orthogonality to $\eta(t)\nabla\psi(\boldsymbol{z})$ for every smooth η therefore implies $F_\psi(t) = 0$ for almost every t.

For each integer $m \geq 1$, choose a countable C^1-norm dense subset of $C_c^\infty(B_m(0))$, viewing its members as functions on $\mathbb{R}^D$, and enumerate their union as $\{\psi_n\}$. Such subsets exist by separability of $C^1(\overline{B_m(0)})$. Every $\psi \in C_c^\infty(\mathbb{R}^D)$ can then be approximated by members supported in one fixed ball, with $\|\nabla\psi_n - \nabla\psi\|_\infty \to 0$ along a suitable sequence. Since μ_t is a probability measure, $\|\nabla\psi_n - \nabla\psi\|_{L^2(\mu_t)} \leq \|\nabla\psi_n - \nabla\psi\|_\infty$. Thus the same countable family generates a dense gradient class for every t. Removing the union of the null sets obtained above, we have, for almost every t,

$$\int \boldsymbol{q}_t^\top \nabla\psi_n \,\mathrm{d}\mu_t = 0 \qquad \text{for every } n.$$

Since $\boldsymbol{q}_t \in T_{\mu_t}\mathcal{P}_2$, density forces $\boldsymbol{q}_t = 0$. Hence the orthogonal complement of the range is trivial, proving density and the claimed isomorphism after completion. □

This lemma allows optimization over completed test potentials to be identified with optimization over all square-integrable tangent velocity fields, as used in Theorem A.3 below.

A.1.2 Residual functional and minimum-flux geometry

For $\boldsymbol{A} \in \mathbb{R}^{D\times D}$ define

$$r_{\boldsymbol{A}}(t) = \partial_t\mu_t + \nabla\cdot(\boldsymbol{A}\boldsymbol{z}\,\mu_t). \tag{13}$$

For a smooth test $\varphi \in \mathscr{D}$, define $\mathcal{R}_D(\boldsymbol{A};\varphi) := \langle r_{\boldsymbol{A}}, \varphi\rangle$. Integration by parts gives

$$\mathcal{R}_D(\boldsymbol{A};\varphi) = \mu_T[\varphi(T)] - \mu_0[\varphi(0)] - \int_0^T \mu_t\big[\partial_t\varphi(t) + \nabla_{\boldsymbol{z}}\varphi(t)^\top \boldsymbol{A}\boldsymbol{z}\big]\,\mathrm{d}t. \tag{14}$$

Using Equation (8), the same quantity is

$$\mathcal{R}_D(\boldsymbol{A};\varphi) = \int_0^T \int \nabla_{\boldsymbol{z}}\varphi(t,\boldsymbol{z})^\top(\boldsymbol{v}_t(\boldsymbol{z}) - \boldsymbol{A}\boldsymbol{z})\mathrm{d}\mu_t(\boldsymbol{z})\mathrm{d}t. \tag{15}$$

Thus it extends continuously to $\mathbb{V}_D$.

For a zero-mass distribution σ, define

$$\|\sigma\|_{-1,\mu} = \sup_{\phi\in C_c^\infty(\mathbb{R}^D)} \left\{\langle\sigma,\phi\rangle : \int \|\nabla\phi\|^2\mathrm{d}\mu \leq 1\right\}. \tag{16}$$

Lemma A.2 (Minimum-flux representation)**.** *If $\boldsymbol{w} \in L^2(\mu;\mathbb{R}^D)$ and $\sigma = -\nabla\cdot(\boldsymbol{w}\mu)$, then*

$$\|\sigma\|_{-1,\mu} = \|P_\mu\boldsymbol{w}\|_{L^2(\mu)}. \tag{17}$$

Moreover, $P_\mu\boldsymbol{w}$ is the unique min-norm vector field among all $L^2(\mu)$ fields satisfying $-\nabla\cdot(\boldsymbol{q}\mu) = \sigma$. Conversely, if $\|\sigma\|_{-1,\mu} < \infty$, a tangent field satisfying this constraint exists.

Proof. For every smooth ϕ,

$$\langle\sigma,\phi\rangle = \int \nabla\phi^\top\boldsymbol{w}\,\mathrm{d}\mu = \int \nabla\phi^\top P_\mu\boldsymbol{w}\,\mathrm{d}\mu,$$

because $\boldsymbol{w} - P_\mu\boldsymbol{w}$ is orthogonal to the tangent space. Cauchy–Schwarz proves the upper bound in Equation (17). Conversely, if $P_\mu\boldsymbol{w} \neq 0$, choose gradients converging in $L^2(\mu)$ to $P_\mu\boldsymbol{w}$ and normalize them. Their pairings converge to $\|P_\mu\boldsymbol{w}\|$, proving equality. If $P_\mu\boldsymbol{w} = 0$, all pairings vanish and equality is immediate.

If $\boldsymbol{q}$ has the same weighted divergence, then $\boldsymbol{q} - \boldsymbol{w}$ is orthogonal to all smooth gradients, hence $P_\mu\boldsymbol{q} = P_\mu\boldsymbol{w}$. Orthogonal decomposition gives

$$\|\boldsymbol{q}\|_{L^2(\mu)}^2 = \|P_\mu\boldsymbol{w}\|_{L^2(\mu)}^2 + \|(I - P_\mu)\boldsymbol{q}\|_{L^2(\mu)}^2.$$

The minimum is therefore attained uniquely at $\boldsymbol{q} = P_\mu\boldsymbol{w}$.

For the converse, finiteness of $\|\sigma\|_{-1,\mu}$ makes $\nabla\phi \mapsto \langle\sigma,\phi\rangle$ a well-defined bounded linear functional on smooth gradients. It extends continuously to $T_\mu\mathcal{P}_2$, so the Riesz representation theorem gives a unique $\boldsymbol{q} \in T_\mu\mathcal{P}_2$ with $\langle\sigma,\phi\rangle = \int \boldsymbol{q}^\top\nabla\phi\,\mathrm{d}\mu$ for every $\phi \in C_c^\infty(\mathbb{R}^D)$. Thus $\sigma = -\nabla\cdot(\boldsymbol{q}\mu)$, and the preceding argument applies. □

Theorem A.3 (Dual representation, exact geometry, and population optimum)**.** *For every* $\boldsymbol{A} \in \mathbb{R}^{D\times D}$,

$$\mathcal{J}_D(\boldsymbol{A}) := \frac{1}{2}\int_0^T \|r_{\boldsymbol{A}}(t)\|_{-1,\mu_t}^2 \mathrm{d}t \tag{18}$$

$$= \sup_{\varphi\in\mathbb{V}_D}\left\{\mathcal{R}_D(\boldsymbol{A};\varphi) - \frac{1}{2}\|\varphi\|_{\mathbb{V}_D}^2\right\} \tag{19}$$

$$= \frac{1}{2}\int_0^T \|\boldsymbol{v}_t - P_{\mu_t}(\boldsymbol{A}\boldsymbol{z})\|_{L^2(\mu_t)}^2 \mathrm{d}t. \tag{20}$$

A minimizer $\boldsymbol{A}_D^\dagger$ *exists. It satisfies*

$$\int_0^T \int \big(\boldsymbol{v}_t(\boldsymbol{z}) - P_{\mu_t}(\boldsymbol{A}_D^\dagger \boldsymbol{z})\big)\boldsymbol{z}^\top \mathrm{d}\mu_t(\boldsymbol{z})\mathrm{d}t = 0. \tag{21}$$

If

$$P_{\mu_t}(\boldsymbol{B}\boldsymbol{z}) = 0 \text{ in } L^2(\mu_t) \text{ for a.e. } t \quad\Longrightarrow\quad \boldsymbol{B} = 0, \tag{22}$$

then the minimizing matrix is unique. Every minimizer satisfies

$$\mathcal{J}_D(\boldsymbol{A}) = \mathcal{J}_D(\boldsymbol{A}_D^\dagger) + \frac{1}{2}\int_0^T \|P_{\mu_t}((\boldsymbol{A}-\boldsymbol{A}_D^\dagger)\boldsymbol{z})\|_{L^2(\mu_t)}^2 \mathrm{d}t. \tag{23}$$

Proof. The canonical equation gives $r_{\boldsymbol{A}} = -\nabla\cdot((\boldsymbol{v}_t - \boldsymbol{A}\boldsymbol{z})\mu_t)$. Lemma A.2 and $P_{\mu_t}\boldsymbol{v}_t = \boldsymbol{v}_t$ yield, for almost every t, $\|r_{\boldsymbol{A}}(t)\|_{-1,\mu_t} = \|\boldsymbol{v}_t - P_{\mu_t}(\boldsymbol{A}\boldsymbol{z})\|_{L^2(\mu_t)}$. This proves Equation (20). To obtain Equation (19), set $\boldsymbol{h}_t = \boldsymbol{v}_t - P_{\mu_t}(\boldsymbol{A}\boldsymbol{z})$. For smooth φ, Equation (15) and orthogonal projection give $\mathcal{R}_D(\boldsymbol{A};\varphi) = \langle \boldsymbol{h}, \nabla_{\boldsymbol{z}}\varphi\rangle_{\mathscr{T}_\mu}$, where the inner product integrates over both space and time. This identity extends continuously to $\mathbb{V}_D$. Lemma A.1 makes the correspondence $\varphi \mapsto \boldsymbol{q} = \nabla_{\boldsymbol{z}}\varphi$ an isometric isomorphism onto $\mathscr{T}_\mu$. Thus $\|\varphi\|_{\mathbb{V}_D} = \|\boldsymbol{q}\|_{\mathscr{T}_\mu}$, and taking the supremum over $\varphi \in \mathbb{V}_D$ is equivalent to taking it over all $\boldsymbol{q} \in \mathscr{T}_\mu$. Completing the square gives

$$\langle \boldsymbol{h}, \boldsymbol{q}\rangle_{\mathscr{T}_\mu} - \tfrac{1}{2}\|\boldsymbol{q}\|_{\mathscr{T}_\mu}^2 = \tfrac{1}{2}\|\boldsymbol{h}\|_{\mathscr{T}_\mu}^2 - \tfrac{1}{2}\|\boldsymbol{q}-\boldsymbol{h}\|_{\mathscr{T}_\mu}^2.$$

The maximum is attained at $\boldsymbol{q} = \boldsymbol{h}$ and equals $\frac{1}{2}\|\boldsymbol{h}\|_{\mathscr{T}_\mu}^2 = \mathcal{J}_D(\boldsymbol{A})$ by Equation (20), proving Equation (19).

Define $\mathsf{S}_D\boldsymbol{B} = [t \mapsto P_{\mu_t}(\boldsymbol{B}\boldsymbol{z})] \in \mathscr{T}_\mu$. This is a bounded linear map because $\|\mathsf{S}_D\boldsymbol{B}\|_{\mathscr{T}_\mu}^2 \le \|\boldsymbol{B}\|_F^2 \int_0^T \int \|\boldsymbol{z}\|^2 \mathrm{d}\mu_t \mathrm{d}t$. Its range is finite dimensional and closed, and $\mathcal{J}_D(\boldsymbol{A}) = \frac{1}{2}\|\boldsymbol{v} - \mathsf{S}_D\boldsymbol{A}\|_{\mathscr{T}_\mu}^2$. Projecting $\boldsymbol{v}$ onto this range therefore gives a minimizing matrix $\boldsymbol{A}_D^\dagger$, whose projected velocity $\mathsf{S}_D\boldsymbol{A}_D^\dagger$ is unique. Orthogonality to every $\mathsf{S}_D\boldsymbol{B}$ gives

$$\int_0^T \int (\boldsymbol{v}_t - P_{\mu_t}(\boldsymbol{A}_D^\dagger \boldsymbol{z}))^\top P_{\mu_t}(\boldsymbol{B}\boldsymbol{z})\mathrm{d}\mu_t \mathrm{d}t = 0.$$

The first factor is tangent, so $P_{\mu_t}(\boldsymbol{B}\boldsymbol{z})$ may be replaced by $\boldsymbol{B}\boldsymbol{z}$. Since the last identity holds for every $\boldsymbol{B}$, it is equivalent to Equation (21). The Pythagorean theorem gives Equation (23); injectivity in Equation (22) makes the matrix representative unique. □

A.2 Finite tests, closed-form regression, and convergence under test enrichment

Let

$$\mathbb{V}_{D,1} \subset \mathbb{V}_{D,2} \subset \cdots \subset \mathbb{V}_D, \qquad \overline{\bigcup_{m\ge 1} \mathbb{V}_{D,m}} = \mathbb{V}_D, \tag{24}$$

where $\mathbb{V}_{D,m} = \mathrm{span}\{\varphi_1, \ldots, \varphi_m\}$. We choose the test potentials φ_j from the class $\mathscr{D}$ defined above; such a dense family exists by Lemma A.1.

Define

$$y_j = \mu_T[\varphi_j(T)] - \mu_0[\varphi_j(0)] - \int_0^T \mu_t[\partial_t \varphi_j(t)]\mathrm{d}t, \tag{25}$$

$$\boldsymbol{H}_j = \int_0^T \mu_t[\nabla_{\boldsymbol{z}} \varphi_j(t,\boldsymbol{z})\boldsymbol{z}^\top]\mathrm{d}t, \tag{26}$$

$$M_{ij} = \int_0^T \mu_t[\nabla_{\boldsymbol{z}} \varphi_i(t,\boldsymbol{z})^\top \nabla_{\boldsymbol{z}} \varphi_j(t,\boldsymbol{z})]\mathrm{d}t. \tag{27}$$

Let $\boldsymbol{a} = \mathrm{vec}(\boldsymbol{A})$, let row j of $\boldsymbol{G}$ be $\mathrm{vec}(\boldsymbol{H}_j)^\top$, and set $\boldsymbol{y} = (y_1, \ldots, y_m)^\top$.

Lemma A.4. *If $\boldsymbol{M} \succ 0$, then*

$$\sup_{\varphi \in \mathbb{V}_{D,m}} \left\{ \mathcal{R}_D(\boldsymbol{A};\varphi) - \frac{1}{2}\|\varphi\|^2_{\mathbb{V}_D} \right\} = \frac{1}{2}(\boldsymbol{y} - \boldsymbol{G}\boldsymbol{a})^\top \boldsymbol{M}^{-1}(\boldsymbol{y} - \boldsymbol{G}\boldsymbol{a}). \tag{28}$$

The maximizing test is

$$\varphi^\star_{\boldsymbol{A},m} = \sum_{j=1}^m [\boldsymbol{M}^{-1}(\boldsymbol{y} - \boldsymbol{G}\boldsymbol{a})]_j \varphi_j. \tag{29}$$

If $\boldsymbol{G}$ has full column rank, the unique matrix minimizer is

$$\mathrm{vec}(\boldsymbol{A}_{D,m}) = (\boldsymbol{G}^\top \boldsymbol{M}^{-1}\boldsymbol{G})^{-1}\boldsymbol{G}^\top \boldsymbol{M}^{-1}\boldsymbol{y}. \tag{30}$$

Proof. Write $\varphi = \sum_{j=1}^m c_j \varphi_j$, with $\boldsymbol{c} = (c_1, \ldots, c_m)^\top$. For each basis function, the definitions of y_j and $\boldsymbol{H}_j$ give

$$\begin{aligned}\mathcal{R}_D(\boldsymbol{A};\varphi_j) &= y_j - \int_0^T \mu_t\left[\nabla_{\boldsymbol{z}} \varphi_j(t,\boldsymbol{z})^\top \boldsymbol{A}\boldsymbol{z}\right] \mathrm{d}t \\ &= y_j - \langle \boldsymbol{H}_j, \boldsymbol{A}\rangle_F = y_j - \mathrm{vec}(\boldsymbol{H}_j)^\top \boldsymbol{a} = y_j - (\boldsymbol{G}\boldsymbol{a})_j.\end{aligned}$$

Therefore, by linearity of $\mathcal{R}_D(\boldsymbol{A};\cdot)$,

$$\mathcal{R}_D(\boldsymbol{A};\varphi) = \sum_{j=1}^m c_j \mathcal{R}_D(\boldsymbol{A};\varphi_j) = \boldsymbol{c}^\top(\boldsymbol{y} - \boldsymbol{G}\boldsymbol{a}).$$

Moreover, since $\boldsymbol{M}$ is the Gram matrix of $\{\varphi_j\}_{j=1}^m$ under the $\mathbb{V}_D$ inner product,

$$\|\varphi\|^2_{\mathbb{V}_D} = \sum_{i,j=1}^m c_i c_j \int_0^T \mu_t\left[\nabla_{\boldsymbol{z}} \varphi_i(t,\boldsymbol{z})^\top \nabla_{\boldsymbol{z}} \varphi_j(t,\boldsymbol{z})\right] \mathrm{d}t = \boldsymbol{c}^\top \boldsymbol{M}\boldsymbol{c}.$$

Completing the square gives

$$\begin{aligned}\boldsymbol{c}^\top(\boldsymbol{y} - \boldsymbol{G}\boldsymbol{a}) - \frac{1}{2}\boldsymbol{c}^\top \boldsymbol{M}\boldsymbol{c} = &-\frac{1}{2}(\boldsymbol{c} - \boldsymbol{M}^{-1}(\boldsymbol{y} - \boldsymbol{G}\boldsymbol{a}))^\top \boldsymbol{M}(\boldsymbol{c} - \boldsymbol{M}^{-1}(\boldsymbol{y} - \boldsymbol{G}\boldsymbol{a})) \\ &+ \frac{1}{2}(\boldsymbol{y} - \boldsymbol{G}\boldsymbol{a})^\top \boldsymbol{M}^{-1}(\boldsymbol{y} - \boldsymbol{G}\boldsymbol{a}).\end{aligned}$$

The first term is nonpositive and vanishes exactly at the coefficients in Equation (29), proving Equation (28). Differentiating the remaining quadratic in $\boldsymbol{a}$ gives $\boldsymbol{G}^\top \boldsymbol{M}^{-1}\boldsymbol{G}\boldsymbol{a} = \boldsymbol{G}^\top \boldsymbol{M}^{-1}\boldsymbol{y}$. Full column rank makes the normal matrix positive definite and yields Equation (30). □

Remark (Weak population residuals as regression residuals). For fixed $\boldsymbol{A}$, $\mathcal{R}_D(\boldsymbol{A};\cdot)$ is the continuous linear functional obtained by testing the continuity-equation residual $r_{\boldsymbol{A}}$ in Equation (13). Its value $\mathcal{R}_D(\boldsymbol{A};\varphi_j)$ measures the signed discrepancy between the moment evolution of μ_t and the contribution of the candidate drift $\boldsymbol{A}\boldsymbol{z}$, as expressed in Equation (14). With the moments defined above,

$$\mathcal{R}_D(\boldsymbol{A};\varphi_j) = y_j - \langle \boldsymbol{H}_j, \boldsymbol{A}\rangle_F = y_j - (\boldsymbol{G}\boldsymbol{a})_j, \qquad j = 1, \ldots, m.$$

Thus $\boldsymbol{y} - \boldsymbol{G}\boldsymbol{a}$ collects these scalar weak residuals, and $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{a}$ means that the continuity equation is satisfied against every test in $\mathbb{V}_{D,m}$. When $\boldsymbol{M} \succ 0$, Lemma A.4 gives the weighted least-squares objective $\frac{1}{2}(\boldsymbol{y} - \boldsymbol{G}\boldsymbol{a})^\top \boldsymbol{M}^{-1}(\boldsymbol{y} - \boldsymbol{G}\boldsymbol{a})$ for estimating $\boldsymbol{A}$. In Appendix B.1.2, we implement this regression by approximating $\boldsymbol{y}$, $\boldsymbol{G}$, and $\boldsymbol{M}$ with empirical sample averages and temporal quadrature.

Lemma A.4 gives the estimator for a fixed finite test space. We next show that, as the nested test spaces become dense in $\mathbb{V}_D$, these estimators converge to the unique minimizer of the full weak population residual objective. To compare the two objectives, we use the Riesz representation of the continuous functional $\mathcal{R}_D(\boldsymbol{A};\cdot)$: for each $\boldsymbol{A}$, there is a unique $h_{\boldsymbol{A}} \in \mathbb{V}_D$ satisfying $\mathcal{R}_D(\boldsymbol{A};\varphi) = \langle h_{\boldsymbol{A}}, \varphi\rangle_{\mathbb{V}_D}$ for every $\varphi \in \mathbb{V}_D$.

Writing $\boldsymbol{a} = \mathrm{vec}(\boldsymbol{A})$ and letting h_0 correspond to $\boldsymbol{A} = 0$, define the linear operator $\mathsf{T}_D : \mathbb{R}^{D^2} \to \mathbb{V}_D$ by

$$\langle \mathsf{T}_D\boldsymbol{a}, \varphi\rangle_{\mathbb{V}_D} = \int_0^T \mu_t[\nabla_{\boldsymbol{z}}\varphi(t,\boldsymbol{z})^\top \boldsymbol{A}\boldsymbol{z}]\mathrm{d}t, \qquad \varphi \in \mathbb{V}_D. \tag{31}$$

Cauchy–Schwarz and the finite integrated second moment make this definition well posed by the Riesz theorem, with $\|\mathsf{T}_D\boldsymbol{a}\|_{\mathbb{V}_D} \le (\int_0^T \mu_t[\|\boldsymbol{z}\|^2]\mathrm{d}t)^{1/2}\|\boldsymbol{a}\|_2$. The residual is affine in $\boldsymbol{A}$, so $h_{\boldsymbol{A}} = h_0 - \mathsf{T}_D\boldsymbol{a}$. Under the gradient identification of Lemma A.1, $\nabla_{\boldsymbol{z}} h_0(t,\cdot) = \boldsymbol{v}_t$, $\nabla_{\boldsymbol{z}}(\mathsf{T}_D\boldsymbol{a})(t,\cdot) = P_{\mu_t}(\boldsymbol{A}\boldsymbol{z})$, and thus $\nabla_{\boldsymbol{z}} h_{\boldsymbol{A}}(t,\cdot) = \boldsymbol{v}_t - P_{\mu_t}(\boldsymbol{A}\boldsymbol{z})$, with these identities understood in $\mathscr{T}_\mu$.

The Riesz representation allows us to complete the square in the dual objective:

$$\mathcal{R}_D(\boldsymbol{A};\varphi) - \tfrac{1}{2}\|\varphi\|^2_{\mathbb{V}_D} = \tfrac{1}{2}\|h_{\boldsymbol{A}}\|^2_{\mathbb{V}_D} - \tfrac{1}{2}\|\varphi - h_{\boldsymbol{A}}\|^2_{\mathbb{V}_D}.$$

The supremum over $\mathbb{V}_D$ is therefore attained at $\varphi = h_{\boldsymbol{A}}$. For the finite-test problem, let $Q_m : \mathbb{V}_D \to \mathbb{V}_{D,m}$ be the orthogonal projection. Since $h_{\boldsymbol{A}} - Q_m h_{\boldsymbol{A}}$ is orthogonal to $\mathbb{V}_{D,m}$, every $\varphi \in \mathbb{V}_{D,m}$ satisfies $\langle h_{\boldsymbol{A}}, \varphi\rangle_{\mathbb{V}_D} = \langle Q_m h_{\boldsymbol{A}}, \varphi\rangle_{\mathbb{V}_D}$. The same calculation with $Q_m h_{\boldsymbol{A}}$ in place of $h_{\boldsymbol{A}}$ shows that the restricted supremum is attained at $\varphi = Q_m h_{\boldsymbol{A}}$, which belongs to $\mathbb{V}_{D,m}$. Consequently,

$$\mathcal{J}_D(\boldsymbol{a}) = \tfrac{1}{2}\|h_{\boldsymbol{A}}\|^2_{\mathbb{V}_D}, \qquad \mathcal{J}_{D,m}(\boldsymbol{a}) = \tfrac{1}{2}\|Q_m h_{\boldsymbol{A}}\|^2_{\mathbb{V}_D}. \tag{32}$$

Thus increasing the test space approximates the full residual representative by its projections $Q_m h_{\boldsymbol{A}}$. To compare the minimizers, write $\boldsymbol{C} = \mathsf{T}_D^*\mathsf{T}_D$ and $\boldsymbol{d} = \mathsf{T}_D^* h_0$ for the normal-equation matrix and vector of the full problem, and $\boldsymbol{C}_m = \mathsf{T}_D^* Q_m \mathsf{T}_D$ and $\boldsymbol{d}_m = \mathsf{T}_D^* Q_m h_0$ for their finite-test counterparts.

Theorem A.5 (Convergence under test-space enrichment)**.** *Suppose that the nested test spaces satisfy Equation* (24) *and that the population minimizer $\boldsymbol{A}_D^\dagger$ is unique. Let $\boldsymbol{A}_{D,m}$ minimize, over $\boldsymbol{A} \in \mathbb{R}^{D\times D}$, the objective obtained by restricting the dual supremum to $\mathbb{V}_{D,m}$, and set $\boldsymbol{a}^\dagger = \mathrm{vec}(\boldsymbol{A}_D^\dagger)$ and $\boldsymbol{a}_m = \mathrm{vec}(\boldsymbol{A}_{D,m})$. Then $\boldsymbol{C} \succ 0$, and, for all sufficiently large m, $\boldsymbol{C}_m \succ 0$ and the finite-test minimizer is unique. For these m,*

$$\|\boldsymbol{a}_m - \boldsymbol{a}^\dagger\|_2 \le \frac{2}{\lambda_{\min}(\boldsymbol{C})}\left(\|\boldsymbol{d}_m - \boldsymbol{d}\|_2 + \|\boldsymbol{C}_m - \boldsymbol{C}\|_{\mathrm{op}}\|\boldsymbol{a}^\dagger\|_2\right). \tag{33}$$

Consequently,

$$\boldsymbol{A}_{D,m} \longrightarrow \boldsymbol{A}_D^\dagger \qquad \text{as } m \to \infty. \tag{34}$$

Proof. For each m, the range of $Q_m\mathsf{T}_D$ is finite dimensional and hence closed. Projecting $Q_m h_0$ onto this range therefore yields a finite-test minimizer. The quadratic representations above yield the normal equations

$$\boldsymbol{C}\boldsymbol{a}^\dagger = \boldsymbol{d}, \qquad \boldsymbol{C}_m\boldsymbol{a}_m = \boldsymbol{d}_m. \tag{35}$$

Density in Equation (24) gives $Q_m q \to q$ for every $q \in \mathbb{V}_D$. Since the domain of T_D is finite dimensional, this convergence is uniform after composition with T_D. To see it directly, let $\boldsymbol{e}_1, \dots, \boldsymbol{e}_{D^2}$ be an orthonormal basis of $\mathbb{R}^{D^2}$. For $\boldsymbol{a} = \sum_q a_q \boldsymbol{e}_q$ with $\|\boldsymbol{a}\|_2 = 1$,

$$\begin{aligned}\|(I - Q_m)\mathsf{T}_D\boldsymbol{a}\| &\le \sum_q |a_q|\,\|(I - Q_m)\mathsf{T}_D\boldsymbol{e}_q\| \\ &\le D\max_q \|(I - Q_m)\mathsf{T}_D\boldsymbol{e}_q\| \longrightarrow 0.\end{aligned}$$

Consequently,

$$\|\boldsymbol{C}_m - \boldsymbol{C}\|_{\mathrm{op}} = \|\mathsf{T}_D^*(Q_m - I)\mathsf{T}_D\|_{\mathrm{op}} \to 0, \tag{36}$$

$$\|\boldsymbol{d}_m - \boldsymbol{d}\|_2 = \|\mathsf{T}_D^*(Q_m - I)h_0\|_2 \to 0. \tag{37}$$

Uniqueness of $\boldsymbol{A}_D^\dagger$ is equivalent to injectivity of T_D. Because its domain is finite dimensional, $\boldsymbol{C} = \mathsf{T}_D^* \mathsf{T}_D \succ 0$. Put $\kappa = \lambda_{\min}(\boldsymbol{C})$. For all sufficiently large m, $\|\boldsymbol{C}_m - \boldsymbol{C}\| \le \kappa/2$, and Weyl's inequality gives $\boldsymbol{C}_m \succeq (\kappa/2)\boldsymbol{I}$. Thus $\boldsymbol{C}_m$ is invertible and $\|\boldsymbol{C}_m^{-1}\| \le 2/\kappa$.

Subtracting the equations in Equation (35) yields

$$\boldsymbol{C}_m(\boldsymbol{a}_m - \boldsymbol{a}^\dagger) = (\boldsymbol{d}_m - \boldsymbol{d}) - (\boldsymbol{C}_m - \boldsymbol{C})\boldsymbol{a}^\dagger.$$

Multiplying by $\boldsymbol{C}_m^{-1}$ proves Equation (33). Its right-hand side tends to zero by Equation (36)–Equation (37), which proves Equation (34). □

A.3 Convergence of the weak population residual

Our goal in this section is to prove that as the number of compatible observables increases, the *minimum weak population residual* converges to zero. This is different from Section A.2, where the observable dimension D is fixed and only the test space is enlarged. As a beginning, we first define the observable space and the infinite-dimensional feature coordinates in which its dynamics become linear.

A.3.1 Preliminaries: Koopman-invariant RKHS and linear feature dynamics

Koopman operators on reproducing-kernel spaces have been studied from both operator-theoretic and statistical perspectives, including boundedness, spectral approximation, and learning from data (Ikeda et al., 2022; Kostic et al., 2022). From now on, let $\mathcal{X}$ be a separable metric space with a flow F_t, and let $\mathcal{H}$ be a separable real RKHS with kernel k (Aronszajn, 1950). The feature map $\Phi(\boldsymbol{x}) := k(\boldsymbol{x}, \cdot)$ satisfies $f(\boldsymbol{x}) = \langle f, \Phi(\boldsymbol{x})\rangle_{\mathcal{H}}$ for every $f \in \mathcal{H}$; we assume that Φ is strongly measurable. Suppose the Koopman operators $U(t)f = f \circ F_t$ preserve $\mathcal{H}$ and form a C_0-semigroup with generator L.

Wendland native spaces provide a concrete example of Koopman invariance: these RKHSs are norm-equivalent to Sobolev spaces on bounded Lipschitz domains, and composition preserves them when the state map is a sufficiently smooth diffeomorphism with bounded derivatives and Jacobian determinant bounded away from zero (Köhne et al., 2025).

Furthermore, write $\mathcal{Z} = \mathcal{H}$ for the feature space. Since $\mathcal{H}$ is reflexive, $T(t) := U(t)^*$ is also a C_0-semigroup, with generator $\mathsf{A} = L^*$ on its adjoint domain, possibly unbounded. The reproducing identity gives $\langle f, T(t)\Phi(\boldsymbol{x})\rangle_{\mathcal{H}} = \langle U(t)f, \Phi(\boldsymbol{x})\rangle_{\mathcal{H}} = f(F_t\boldsymbol{x})$. As this holds for every $f \in \mathcal{H}$, the features evolve linearly, following:

$$\Phi(F_t\boldsymbol{x}) = T(t)\Phi(\boldsymbol{x}).$$

For an orthonormal basis $\{e_j\}_{j\ge1}$ of $\mathcal{H}$, let P_D be the projector onto $\mathcal{Z}_D = \operatorname{span}\{e_1, \dots, e_D\}$. Since $\langle e_j, \Phi(\boldsymbol{x})\rangle_{\mathcal{H}} = e_j(\boldsymbol{x})$,

$$P_D\Phi(\boldsymbol{x}) = \sum_{j=1}^{D} e_j(\boldsymbol{x})e_j \quad \longleftrightarrow \quad \boldsymbol{g}_D(\boldsymbol{x}) = \big(e_1(\boldsymbol{x}), \dots, e_D(\boldsymbol{x})\big)^\top.$$

Thus we may view the D-dimensional encoder as a projection of the same feature map $\Phi(\boldsymbol{x})$. For a random initial state $\boldsymbol{X}_0$, define $\mu_t := \operatorname{Law}(\Phi(F_t\boldsymbol{X}_0))$ and $\mu_t^D := (P_D)_{\#}\mu_t$. Here μ_t denotes the full feature distribution, while μ_t^D, identified with a measure on $\mathbb{R}^D$, corresponds to the latent distribution denoted by μ_t in Section 3.1 and Sections A.1–A.2. The superscript makes its dependence on the observable dimension explicit. These projected laws can be compared in the common space $\mathcal{Z}$ as D increases, with $T(t)$ providing the exact linear evolution for the subsequent convergence analysis.

A.3.2 Convergence of the weak population residual

Fix $\tau > 0$ and $\boldsymbol{X}_0 \sim \rho_0$ on $(\Omega, \mathbb{P})$. For $t \in [0, 2\tau]$, set $\boldsymbol{Z}_0 = \Phi(\boldsymbol{X}_0)$, $\boldsymbol{Z}_t = T(t)\boldsymbol{Z}_0 = \Phi(F_t\boldsymbol{X}_0)$, $\mu_t = \operatorname{Law}(\boldsymbol{Z}_t)$, and $\mu_t^D = (P_D)_{\#}\mu_t$. Assume that $\boldsymbol{Z}_0 \in D(\mathsf{A})$ almost surely and $\mathbb{E}(\|\boldsymbol{Z}_0\|^2 + \|\mathsf{A}\boldsymbol{Z}_0\|^2) < \infty$. We write $\|\cdot\|_{L^2}$ for the norm on $L^2(\Omega; \mathcal{Z})$ and $\|\cdot\|_{\mathbb{L}}$ for the norm on $\mathbb{L} = L^2([0, 2\tau] \times \Omega; \mathcal{Z})$.

The semigroup preserves $D(\mathsf{A})$ and satisfies $\mathsf{A}T(t)\boldsymbol{z} = T(t)\mathsf{A}\boldsymbol{z}$ and $T(t)\boldsymbol{z} - \boldsymbol{z} = \int_0^t T(r)\mathsf{A}\boldsymbol{z}\,\mathrm{d}r$ for $\boldsymbol{z} \in D(\mathsf{A})$ (Engel & Nagel, 2000, Chapter II). Strong continuity and uniform boundedness give

$M_\tau := \sup_{0\le t\le 2\tau}\|T(t)\| < \infty$, so

$$\int_0^{2\tau} \mathbb{E}(\|\boldsymbol{Z}_t\|^2 + \|\mathsf{A}\boldsymbol{Z}_t\|^2)\,\mathrm{d}t \le 2\tau M_\tau^2 \mathbb{E}(\|\boldsymbol{Z}_0\|^2 + \|\mathsf{A}\boldsymbol{Z}_0\|^2) < \infty.$$

The same bound and dominated convergence show that $t \mapsto \boldsymbol{Z}_t$ is continuous in $L^2(\Omega;\mathcal{Z})$. For $s \le t$, coupling $P_D\boldsymbol{Z}_s$ and $P_D\boldsymbol{Z}_t$ and using $\boldsymbol{Z}_t - \boldsymbol{Z}_s = \int_s^t \mathsf{A}\boldsymbol{Z}_r\,\mathrm{d}r$ yields

$$W_2(\mu_s^D, \mu_t^D) \le \|P_D(\boldsymbol{Z}_t - \boldsymbol{Z}_s)\|_{L^2} \le \int_s^t \|\mathsf{A}\boldsymbol{Z}_r\|_{L^2}\,\mathrm{d}r.$$

Since the last integrand belongs to $L^2(0,2\tau)$, $\mu_\cdot^D \in AC^2([0,2\tau];\mathcal{P}_2(\mathcal{Z}_D))$. The weak-continuity framework of the preceding subsections therefore applies, with weak population objective

$$\mathcal{J}_D(\boldsymbol{B}) = \frac{1}{2}\int_0^{2\tau} \|\partial_t\mu_t^D + \nabla\cdot(\boldsymbol{B}\boldsymbol{y}\,\mu_t^D)\|^2_{-1,\mu_t^D}\,\mathrm{d}t. \tag{38}$$

Theorem A.3 ensures that a minimizer $\boldsymbol{A}_D \in \operatorname{argmin}_{\boldsymbol{B}\in\mathcal{L}(\mathcal{Z}_D)} \mathcal{J}_D(\boldsymbol{B})$ exists; uniqueness is unnecessary for the statements below.

Lemma A.6 (Comparison with the projected path velocity)**.** *Under the assumptions above, every $\boldsymbol{B} \in \mathcal{L}(\mathcal{Z}_D)$ satisfies*

$$\sqrt{2\mathcal{J}_D(\boldsymbol{B})} \le \|P_D\mathsf{A}\boldsymbol{Z} - \boldsymbol{B}P_D\boldsymbol{Z}\|_{\mathbb{L}}.$$

Proof. Set $\boldsymbol{Y}_t = P_D\boldsymbol{Z}_t$ and $\boldsymbol{V}_t = P_D\mathsf{A}\boldsymbol{Z}_t$. Conditioning $\boldsymbol{V}$ on $(t,\boldsymbol{Y}_t)$ under the probability measure $(2\tau)^{-1}\mathrm{d}t\,\mathrm{d}\mathbb{P}$ gives a jointly measurable field $\boldsymbol{w}_t^D(\boldsymbol{y}) = \mathbb{E}[\boldsymbol{V}_t \mid \boldsymbol{Y}_t = \boldsymbol{y}]$ for almost every t. Conditional Jensen's inequality gives $\|\boldsymbol{w}^D(\boldsymbol{Y})\|_{\mathbb{L}} \le \|\boldsymbol{V}\|_{\mathbb{L}} < \infty$. For $\psi \in C_c^\infty(\mathcal{Z}_D)$, the chain rule along $\boldsymbol{Y}_t$ gives

$$\mathbb{E}\psi(\boldsymbol{Y}_t) - \mathbb{E}\psi(\boldsymbol{Y}_s) = \int_s^t \int \langle\nabla\psi(\boldsymbol{y}), \boldsymbol{w}_r^D(\boldsymbol{y})\rangle\,\mathrm{d}\mu_r^D(\boldsymbol{y})\,\mathrm{d}r.$$

Here Fubini's theorem applies because $\nabla\psi$ is bounded and $\boldsymbol{V} \in \mathbb{L}$. Thus $\boldsymbol{w}^D$ satisfies the projected continuity equation $\partial_t\mu_t^D + \nabla_{\boldsymbol{y}}\cdot(\boldsymbol{w}_t^D\mu_t^D) = 0$ in the sense of distributions on $(0,2\tau)\times\mathcal{Z}_D$. Writing $P_{\mu_t^D}$ for the orthogonal projection onto its Wasserstein tangent space, the canonical velocity is $P_{\mu_t^D}\boldsymbol{w}_t^D$. Theorem A.3 and contraction of this projection imply

$$2\mathcal{J}_D(\boldsymbol{B}) = \int_0^{2\tau} \|P_{\mu_t^D}(\boldsymbol{w}_t^D - \boldsymbol{B}\boldsymbol{y})\|^2_{L^2(\mu_t^D)}\,\mathrm{d}t \le \|\boldsymbol{w}^D(\boldsymbol{Y}) - \boldsymbol{B}\boldsymbol{Y}\|_{\mathbb{L}}^2.$$

Since $\boldsymbol{w}_t^D(\boldsymbol{Y}_t) - \boldsymbol{B}\boldsymbol{Y}_t = \mathbb{E}[\boldsymbol{V}_t - \boldsymbol{B}\boldsymbol{Y}_t \mid \boldsymbol{Y}_t]$, another application of conditional Jensen gives $\|\boldsymbol{w}^D(\boldsymbol{Y}) - \boldsymbol{B}\boldsymbol{Y}\|_{\mathbb{L}} \le \|\boldsymbol{V} - \boldsymbol{B}\boldsymbol{Y}\|_{\mathbb{L}}$, proving the claim. □

To approximate the projected velocity, we use the bounded difference quotient $C_h = (T(h) - I)/h$ before projecting onto $\mathcal{Z}_D$. This avoids applying the unbounded generator to $P_D\boldsymbol{Z}_t$.

Theorem A.7 (Convergence of the weak population residual)**.** *Under the assumptions above, every choice of weak population minimizers $\boldsymbol{A}_D$ satisfies $\mathcal{J}_D(\boldsymbol{A}_D) \to 0$ as $D \to \infty$. More precisely, define $a(h) = \|(\mathsf{A} - C_h)\boldsymbol{Z}\|_{\mathbb{L}}$ and $b_D = \|(I - P_D)\boldsymbol{Z}\|_{\mathbb{L}}$. Then $a(h) \to 0$ as $h \downarrow 0$, $b_D \to 0$ as $D \to \infty$, and*

$$\sqrt{2\mathcal{J}_D(\boldsymbol{A}_D)} \le a(h) + \|C_h\|b_D \qquad (h > 0).$$

Proof. For fixed $h > 0$, C_h is bounded on $\mathcal{Z}$, so $\boldsymbol{B}_{D,h} = P_DC_h|_{\mathcal{Z}_D}$ is an admissible comparison matrix. The velocity error decomposes as

$$P_D\mathsf{A}\boldsymbol{Z} - P_DC_hP_D\boldsymbol{Z} = P_D(\mathsf{A} - C_h)\boldsymbol{Z} + P_DC_h(I - P_D)\boldsymbol{Z}.$$

The preceding lemma, the triangle inequality in $\mathbb{L}$, and $\|P_D\| \le 1$ therefore give

$$\sqrt{2\mathcal{J}_D(\boldsymbol{A}_D)} \le \sqrt{2\mathcal{J}_D(\boldsymbol{B}_{D,h})} \le a(h) + \|C_h\|b_D.$$

Strong convergence of P_D gives $(I - P_D)\boldsymbol{Z}_t \to 0$ pointwise, while $\|(I - P_D)\boldsymbol{Z}_t\|^2 \le \|\boldsymbol{Z}_t\|^2$ is integrable over time and probability. Dominated convergence yields $b_D \to 0$.

For $\boldsymbol{z} \in D(\mathsf{A})$, $C_h\boldsymbol{z} = h^{-1}\int_0^h T(r)\mathsf{A}\boldsymbol{z}\,\mathrm{d}r \to \mathsf{A}\boldsymbol{z}$ by strong continuity. Fix $h_0 > 0$ and let $M_0 = \sup_{0\le r\le h_0}\|T(r)\| < \infty$. For $0 < h \le h_0$, $\|C_h\boldsymbol{z}\| \le M_0\|\mathsf{A}\boldsymbol{z}\|$, hence $\|(\mathsf{A} - C_h)\boldsymbol{Z}_t\|^2 \le (1 + M_0)^2\|\mathsf{A}\boldsymbol{Z}_t\|^2$. The finite orbit energy established above makes this bound integrable, so dominated convergence gives $a(h) \to 0$.

The bound on the minimum now proves the result by first fixing h and then increasing D. Specifically, for any $\varepsilon > 0$, choose h with $a(h) < \varepsilon/2$, and then choose D sufficiently large that $b_D < \varepsilon/[2(1 + \|C_h\|)]$. It follows that $\sqrt{2\mathcal{J}_D(\boldsymbol{A}_D)} < \varepsilon$. □

A.4 From weak residual consistency to strong semigroup convergence

Section A.3.2 establishes $\mathcal{J}_D(\boldsymbol{A}_D) \to 0$ for the weak population minimizers $\boldsymbol{A}_D$. In this section, under a uniform stability bound on $e^{t\boldsymbol{A}_D}$, we first show that this vanishing weak residual implies uniform Wasserstein convergence of the predicted distributions on the training interval. The subsequent subsections establish the stronger conclusion of semigroup convergence at every initial feature vector: compactness provides strong operator subsequences, and finite-time linear determinacy (FT) identifies their limits with $T(t)$.

Assumption A.8 (Finite-time stability of weak population minimizers)**.** For the selected minimizers $\boldsymbol{A}_D \in \operatorname{argmin}_{\boldsymbol{B}\in\mathcal{L}(\mathcal{Z}_D)} \mathcal{J}_D(\boldsymbol{B})$, there exists $M \ge 1$, independent of D, such that

$$\sup_D \sup_{0\le t\le\tau} \|e^{t\boldsymbol{A}_D}\|_{\mathcal{L}(\mathcal{Z}_D)} \le M.$$

This assumption bounds the finite-time evolution generated by the selected minimizers. It is an additional hypothesis beyond convergence of the weak residual and does not require a uniform bound on $\|\boldsymbol{A}_D\|$.

A.4.1 The weak residual controls probability prediction

The weak residual measures the minimum correction to a candidate linear velocity needed to transport a prescribed distribution curve. Propagating this correction gives a prediction bound on any subinterval.

Lemma A.9 (The weak residual controls finite-time prediction)**.** *Let $\alpha_\cdot \in AC^2([s,t];\mathcal{P}_2(\mathbb{R}^d))$ and $\boldsymbol{B} \in \mathbb{R}^{d\times d}$. For $r \in [s,t]$, write $r_{\boldsymbol{B}}(r) = \partial_r\alpha_r + \nabla_{\boldsymbol{y}}\cdot(\boldsymbol{B}\boldsymbol{y}\,\alpha_r)$. Then*

$$W_2((e^{(t-s)\boldsymbol{B}})_\#\alpha_s, \alpha_t) \le \int_s^t \|e^{(t-r)\boldsymbol{B}}\|\,\|r_{\boldsymbol{B}}(r)\|_{-1,\alpha_r}\,\mathrm{d}r.$$

Proof. Let $\boldsymbol{v}_r$ be the canonical velocity of α_r and set $\boldsymbol{u}_r = \boldsymbol{v}_r - P_{\alpha_r}(\boldsymbol{B}\boldsymbol{y})$, where P_{α_r} is the orthogonal projection onto the Wasserstein tangent space. Lemma A.2 and Theorem A.3 give $-\nabla_{\boldsymbol{y}}\cdot(\boldsymbol{u}_r\alpha_r) = r_{\boldsymbol{B}}(r)$ and $\|\boldsymbol{u}_r\|_{L^2(\alpha_r)} = \|r_{\boldsymbol{B}}(r)\|_{-1,\alpha_r}$ for almost every r. Hence $\partial_r\alpha_r + \nabla_{\boldsymbol{y}}\cdot((\boldsymbol{B}\boldsymbol{y} + \boldsymbol{u}_r)\alpha_r) = 0$ in the sense of distributions. Choose a jointly Borel representative of $\boldsymbol{u}$. The corrected velocity $\boldsymbol{B}\boldsymbol{y} + \boldsymbol{u}_r$ has finite squared energy because $\boldsymbol{v} \in L^2(\mathrm{d}r\,\mathrm{d}\alpha_r)$, tangent projection is contractive, and $\alpha_\cdot$ has bounded second moments on $[s,t]$.

The superposition principle therefore gives a probability measure on absolutely continuous paths $\boldsymbol{\gamma}$ with marginals α_r and $\dot{\boldsymbol{\gamma}}_r = \boldsymbol{B}\boldsymbol{\gamma}_r + \boldsymbol{u}_r(\boldsymbol{\gamma}_r)$ almost everywhere (Ambrosio et al., 2008, Theorem 8.2.1). Variation of constants yields

$$\boldsymbol{\gamma}_t - e^{(t-s)\boldsymbol{B}}\boldsymbol{\gamma}_s = \int_s^t e^{(t-r)\boldsymbol{B}}\boldsymbol{u}_r(\boldsymbol{\gamma}_r)\,\mathrm{d}r.$$

The pair $(e^{(t-s)\boldsymbol{B}}\boldsymbol{\gamma}_s, \boldsymbol{\gamma}_t)$ couples the two distributions in the claim. Minkowski's integral inequality and the marginal identity $\mathrm{Law}(\boldsymbol{\gamma}_r) = \alpha_r$ give

$$\Big(\mathbb{E}\|\boldsymbol{\gamma}_t - e^{(t-s)\boldsymbol{B}}\boldsymbol{\gamma}_s\|^2\Big)^{1/2} \le \int_s^t \|e^{(t-r)\boldsymbol{B}}\|\,\|\boldsymbol{u}_r\|_{L^2(\alpha_r)}\,\mathrm{d}r.$$

Substituting the weak residual norm proves the bound. □

To compare the learned linear propagator in the common space $\mathcal{Z}$, define

$$K_D(t) = \iota_D e^{t\boldsymbol{A}_D} P_D, \qquad S_D(t) = K_D(t) + (I - P_D),$$

where $\iota_D : \mathcal{Z}_D \hookrightarrow \mathcal{Z}$ is inclusion. $K_D(t)$ evolves the projected features, while $S_D(t)$ also retains the omitted unresolved feature space. On $\mathcal{Z}_D \oplus \mathcal{Z}_D^{\perp}$, $S_D(t)$ acts as $e^{t\boldsymbol{A}_D} \oplus I$, so it is a C_0-semigroup with $S_D(0) = I$ and $\|S_D(t)\| \le \max\{\|e^{t\boldsymbol{A}_D}\|, 1\} \le M$ for $0 \le t \le \tau$. We use S_D only for the subsequent operator-convergence argument; K_D remains the propagator over the finite feature subspace, with $K_D(0) = P_D$.

The following result combines the weak residual bound with the error from projecting $\boldsymbol{Z}_t$. It is uniform in both the starting time and the prediction lag, as needed later to compare operators on all source distribution laws.

Theorem A.10. *Under Theorem A.7 and Assumption A.8, let* $q_D = \sup_{0\le r\le 2\tau} \|(I - P_D)\boldsymbol{Z}_r\|_{L^2}$, *with the* L^2 *norm defined in Section A.3.2. Then* $q_D \to 0$ *and*

$$\sup_{s,t\in[0,\tau]} W_2^{\mathcal{Z}}(K_D(t)_{\#}\mu_s, \mu_{s+t}) \le q_D + M\sqrt{2\tau \mathcal{J}_D(\boldsymbol{A}_D)} \longrightarrow 0,$$
$$\sup_{s,t\in[0,\tau]} W_2^{\mathcal{Z}}(S_D(t)_{\#}\mu_s, \mu_{s+t}) \le 2q_D + M\sqrt{2\tau \mathcal{J}_D(\boldsymbol{A}_D)} \longrightarrow 0.$$

Prediction from the initial law over the full training interval satisfies

$$\sup_{0\le t\le 2\tau} W_2^{\mathcal{Z}}(K_D(t)_{\#}\mu_0, \mu_t) \le q_D + M^2\sqrt{4\tau \mathcal{J}_D(\boldsymbol{A}_D)} \longrightarrow 0.$$

Proof. Strong convergence of P_D and dominated convergence imply $\|(I - P_D)\boldsymbol{V}\|_{L^2} \to 0$ for every $\boldsymbol{V} \in L^2(\Omega; \mathcal{Z})$. Since $r \mapsto \boldsymbol{Z}_r$ is continuous in this space, its image on $[0, 2\tau]$ is compact. For any $\varepsilon > 0$, choose a finite ε-net $\{\boldsymbol{Z}_{r_j}\}_{j=1}^m$ of this image. Contraction of $I - P_D$ gives $q_D \le \varepsilon + \max_j \|(I - P_D)\boldsymbol{Z}_{r_j}\|_{L^2}$; letting $D \to \infty$ and then $\varepsilon \downarrow 0$ proves $q_D \to 0$.

For $s, t \in [0, \tau]$, apply Lemma A.9 to $\mu_\cdot^D$ on $[s, s + t] \subset [0, 2\tau]$. With $r_{\boldsymbol{A}_D}(r) = \partial_r \mu_r^D + \nabla_{\boldsymbol{y}} \cdot (\boldsymbol{A}_D \boldsymbol{y}\, \mu_r^D)$, stability and Cauchy–Schwarz yield

$$W_2((e^{t\boldsymbol{A}_D})_{\#}\mu_s^D, \mu_{s+t}^D) \le M \int_s^{s+t} \|r_{\boldsymbol{A}_D}(r)\|_{-1,\mu_r^D}\, \mathrm{d}r \le M\sqrt{2t \mathcal{J}_D(\boldsymbol{A}_D)}.$$

The last inequality uses $\int_0^{2\tau} \|r_{\boldsymbol{A}_D}(r)\|_{-1,\mu_r^D}^2\, \mathrm{d}r = 2\mathcal{J}_D(\boldsymbol{A}_D)$. Isometric inclusion into $\mathcal{Z}$ preserves this Wasserstein distance, and coupling $P_D\boldsymbol{Z}_{s+t}$ with $\boldsymbol{Z}_{s+t}$ adds at most q_D. This proves the bound for K_D since $t \le \tau$. Likewise, $S_D(t)\boldsymbol{Z}_s - K_D(t)\boldsymbol{Z}_s = (I - P_D)\boldsymbol{Z}_s$, so the coupling induced by $\boldsymbol{Z}_s$ adds at most another q_D, proving the bound for S_D.

For $0 \le r \le 2\tau$, the semigroup law gives $\|e^{r\boldsymbol{A}_D}\| = \|(e^{(r/2)\boldsymbol{A}_D})^2\| \le M^2$. Applying the same argument on $[0, t]$ therefore bounds the full-interval prediction error by $q_D + M^2\sqrt{2t\mathcal{J}_D(\boldsymbol{A}_D)}$, which is at most the stated bound for $t \le 2\tau$. All three bounds vanish because $q_D \to 0$ and $\mathcal{J}_D(\boldsymbol{A}_D) \to 0$ by Theorem A.7. □

A.4.2 Distributional identifiability of the linear propagator

Theorem A.10 shows that a vanishing weak residual yields convergent distribution predictions. To establish convergence of the linear propagators themselves, we also need the distribution dynamics to identify the underlying map. This is a fundamental issue when inferring dynamics from unpaired snapshots: distinct trajectories can produce the same distribution at every time. For linear SDEs with additive noise in finite dimensions, Guan et al. (2024) characterize the obstruction to identifiability through generalized rotational symmetries of the initial distribution. For example, a rotational flow and a stationary flow preserve the same isotropic Gaussian distribution, despite moving individual states differently.

In our setting, the subsequent operator-convergence argument therefore requires the family $\{\mu_s : 0 \le s \le \tau\}$ to determine $T(t)$ uniquely for each lag $t \in [0, \tau]$. We impose this requirement directly on bounded linear maps on $\mathcal{Z}$ through the following assumption.

Assumption A.11 (Finite-time linear determinacy (FT)). For every lag $t \in [0, \tau]$ and every bounded linear map $B : \mathcal{Z} \to \mathcal{Z}$,

$$B_{\#}\mu_s = \mu_{s+t} \quad \text{for all } s \in [0, \tau] \qquad \Longrightarrow \qquad B = T(t). \tag{39}$$

For a fixed lag, FT identifies the propagator from its action on the entire family of source distributions. At $t = 0$, it rules out nonidentity linear symmetries shared by all these distributions. Allowing every bounded linear candidate B, including noninvertible ones, ensures that the condition applies to the operator limits considered below. FT supplies uniqueness; stability and compactness will provide the remaining ingredients for convergence. The next proposition relates FT to the linear span of the distribution supports and gives a sufficient condition in terms of their means.

Proposition A.12 (Total linear support and a sufficient criterion for FT). *Let $\bar{\mu} = \tau^{-1} \int_0^\tau \mu_s \, \mathrm{d}s$. Under FT,*

$$\overline{\operatorname{span}(\operatorname{supp} \bar{\mu})} = \mathcal{Z}. \tag{40}$$

A sufficient condition for FT is

$$\overline{\operatorname{span}\{\boldsymbol{m}_s : 0 \le s \le \tau\}} = \mathcal{Z}, \qquad \boldsymbol{m}_s = \int \boldsymbol{z} \, \mathrm{d}\mu_s(\boldsymbol{z}). \tag{41}$$

Proof. If Equation (40) failed, there would be a nonzero $\boldsymbol{h}$ with $\langle \boldsymbol{h}, \boldsymbol{z} \rangle = 0$ for $\bar{\mu}$-almost every $\boldsymbol{z}$. Consequently

$$\int_0^\tau \mathbb{E}|\langle \boldsymbol{h}, \boldsymbol{Z}_s \rangle|^2 \mathrm{d}s = 0.$$

The integrand is nonnegative and continuous in s, because $\boldsymbol{Z}_s$ is L^2-continuous. It is therefore zero for every $s \in [0, \tau]$. The bounded linear map $B\boldsymbol{z} = \boldsymbol{z} + \langle \boldsymbol{h}, \boldsymbol{z} \rangle \boldsymbol{h}$ fixes $\boldsymbol{Z}_s$ almost surely for each s, and hence $B_{\#}\mu_s = \mu_s$ for all s. But $B \neq I$, contradicting FT at $t = 0$. This proves the first claim.

For the sufficient condition, take B satisfying the premise of FT for a fixed lag t. Finite second moments allow Bochner expectations, and bounded linear maps commute with them. Thus

$$B\boldsymbol{m}_s = \boldsymbol{m}_{s+t} = T(t)\boldsymbol{m}_s \qquad (0 \le s \le \tau).$$

The bounded operator $B - T(t)$ vanishes on the dense linear span in Equation (41), so it vanishes everywhere. This proves FT. □

A.4.3 From distributional compactness to strong operator compactness

Theorem A.10 establishes W_2 convergence of the pushforward measures $S_D(t)_{\#}\mu_s$ to μ_{s+t} as the weak residual vanishes. To obtain convergence of the linear propagators, we next show that relative compactness of $\{(B_n)_{\#}\nu\}$ in W_2 yields strongly convergent operator subsequences when (B_n) is uniformly bounded and $\operatorname{supp} \nu$ has dense linear span in $\mathcal{Z}$. In our setting, stability supplies the bound, and Proposition A.12 establishes the support condition for the mixture $\bar{\mu} = \tau^{-1} \int_0^\tau \mu_s \, \mathrm{d}s$. The following lemma provides the required compactness; the next subsection uses FT to identify each operator limit with the true propagator $T(t)$.

Lemma A.13 (From W_2 compactness to strong operator compactness). *Let $\mathcal{Z}$ be a separable real Hilbert space and let $\nu \in \mathcal{P}_2(\mathcal{Z})$ satisfy $\overline{\operatorname{span}(\operatorname{supp} \nu)} = \mathcal{Z}$. Suppose $B_n \in \mathcal{L}(\mathcal{Z})$, $\sup_n \|B_n\| \le R < \infty$, and $\{(B_n)_{\#}\nu\}$ is relatively compact in W_2. Then every subsequence of (B_n) has a further subsequence and a bounded linear map B, $\|B\| \le R$, such that*

$$\|B_n\boldsymbol{z} - B\boldsymbol{z}\| \longrightarrow 0 \qquad \textit{for every } \boldsymbol{z} \in \mathcal{Z} \tag{42}$$

along that further subsequence.

Proof. Start with any subsequence and retain the index n. Choose a countable dense set $\{\boldsymbol{q}_j\}$ in $\mathcal{Z}$. The uniform operator bound allows a diagonal extraction for which $\langle B_n \boldsymbol{q}_i, \boldsymbol{q}_j \rangle$ converges for every i, j. By density and the same bound, these limits extend to a bilinear form $b(\boldsymbol{x}, \boldsymbol{y})$ with $|b(\boldsymbol{x}, \boldsymbol{y})| \le R\|\boldsymbol{x}\|\|\boldsymbol{y}\|$. Riesz representation gives $B \in \mathcal{L}(\mathcal{Z})$ with $b(\boldsymbol{x}, \boldsymbol{y}) = \langle B\boldsymbol{x}, \boldsymbol{y} \rangle$ and $\|B\| \le R$, so B_n converges to B in the weak operator topology. Extracting once more using the W_2 relative compactness gives

$$B_n \xrightarrow{\mathrm{WOT}} B, \qquad (B_n)_{\#}\nu \xrightarrow{W_2} \lambda \quad \text{for some } \lambda \in \mathcal{P}_2(\mathcal{Z}). \tag{43}$$

For every $\boldsymbol{h} \in \mathcal{Z}$, weak operator convergence and dominated convergence imply

$$\int e^{i\langle \boldsymbol{h}, \boldsymbol{y}\rangle} \, \mathrm{d}\lambda(\boldsymbol{y}) = \lim_n \int e^{i\langle \boldsymbol{h}, B_n \boldsymbol{z}\rangle} \, \mathrm{d}\nu(\boldsymbol{z}) = \int e^{i\langle \boldsymbol{h}, B\boldsymbol{z}\rangle} \, \mathrm{d}\nu(\boldsymbol{z}).$$

The first equality follows from W_2 convergence, which implies weak convergence of the measures $(B_n)_{\#}\nu$ to λ. Thus λ and $B_{\#}\nu$ have the same characteristic functional. Restricting $\boldsymbol{h}$ to each finite span of an orthonormal basis gives equality of all finite-coordinate distributions. These coordinates generate the Borel sigma field of $\mathcal{Z}$, since it is separable, and hence $\lambda = B_{\#}\nu$.

Convergence in W_2 also gives convergence of second moments, so $\int \|B_n \boldsymbol{z}\|^2 \, \mathrm{d}\nu(\boldsymbol{z}) \to \int \|B\boldsymbol{z}\|^2 \, \mathrm{d}\nu(\boldsymbol{z})$. Indeed, the difference between the square roots of these moments is bounded by $W_2((B_n)_{\#}\nu, B_{\#}\nu)$, by the reverse triangle inequality in any transport coupling. For each fixed $\boldsymbol{z}$, weak operator convergence gives $\langle B_n \boldsymbol{z}, B\boldsymbol{z}\rangle \to \|B\boldsymbol{z}\|^2$, and $|\langle B_n \boldsymbol{z}, B\boldsymbol{z}\rangle| \le R^2 \|\boldsymbol{z}\|^2$ is ν-integrable. Dominated convergence therefore gives convergence of the corresponding integrals. Expanding the square yields

$$\int \|(B_n - B)\boldsymbol{z}\|^2 \, \mathrm{d}\nu(\boldsymbol{z}) = \int \big(\|B_n \boldsymbol{z}\|^2 + \|B\boldsymbol{z}\|^2 - 2\langle B_n \boldsymbol{z}, B\boldsymbol{z}\rangle\big) \, \mathrm{d}\nu(\boldsymbol{z}) \longrightarrow 0. \tag{44}$$

To deduce pointwise convergence on $\operatorname{supp}\nu$ from convergence in $L^2(\nu; \mathcal{Z})$, fix $\boldsymbol{z} \in \operatorname{supp}\nu$ and $\delta > 0$. The open ball $\mathbb{B}_\delta(\boldsymbol{z})$ has positive ν-measure. For $\boldsymbol{y}$ in this ball, $\|(B_n - B)\boldsymbol{z}\| \le 2R\delta + \|(B_n - B)\boldsymbol{y}\|$. Averaging over the ball and applying Cauchy–Schwarz gives

$$\|(B_n - B)\boldsymbol{z}\| \le 2R\delta + \frac{\big(\int \|(B_n - B)\boldsymbol{y}\|^2 \, \mathrm{d}\nu(\boldsymbol{y})\big)^{1/2}}{\nu(\mathbb{B}_\delta(\boldsymbol{z}))^{1/2}}.$$

Letting $n \to \infty$ and then $\delta \downarrow 0$ proves convergence at every point of $\operatorname{supp}\nu$ along the same subsequence. Linearity extends it to the span of this support, which is dense by assumption. Finally, for any $\boldsymbol{z} \in \mathcal{Z}$ and any $\boldsymbol{x}$ in this span, $\|(B_n - B)\boldsymbol{z}\| \le 2R\|\boldsymbol{z} - \boldsymbol{x}\| + \|(B_n - B)\boldsymbol{x}\|$. Approximating $\boldsymbol{z}$ by such $\boldsymbol{x}$ proves Equation (42). The starting subsequence was arbitrary, so the asserted strong operator relative compactness follows. □

In the next subsection, we apply the lemma with $\nu = \bar{\mu}$ and $B_n = S_{D_n}(t_n)$. Theorem A.10 and W_2 continuity of $t \mapsto \mu_t$ establish the required compactness of $\{(S_{D_n}(t_n))_{\#}\bar{\mu}\}$, even when the time lags t_n vary. Along subsequences with $t_n \to t$, FT identifies the strong operator limit B through the identities $B_{\#}\mu_s = \mu_{s+t}$ for all $s \in [0, \tau]$.

A.4.4 Locally uniform strong convergence of the learned propagators

Theorem A.10 establishes convergence of the pushforward measures $S_D(t)_{\#}\mu_s$, and Lemma A.13 converts W_2 compactness of these measures into strong operator subsequences. We now combine these results with the finite-time linear determinacy condition in Assumption A.11 to identify every subsequential limit with $T(t)$. Allowing the time lags to vary along each subsequence will establish convergence uniformly on $[0, \tau]$ for every fixed $\boldsymbol{z} \in \mathcal{Z}$. The semigroup property then extends the result to any finite time interval.

Theorem A.14 (Locally uniform strong convergence under stability and FT)**.** *In the setting of Sections A.3.1 and A.3.2, assume $\boldsymbol{Z}_0 \in D(\mathsf{A})$ almost surely and $\mathbb{E}(\|\boldsymbol{Z}_0\|^2 + \|\mathsf{A}\boldsymbol{Z}_0\|^2) < \infty$. Let P_D be the increasing finite-rank orthogonal projections onto $\mathcal{Z}_D$, with $P_D\boldsymbol{z} \to \boldsymbol{z}$ for every $\boldsymbol{z} \in \mathcal{Z}$, and choose $\boldsymbol{A}_D \in \operatorname{argmin}_{\boldsymbol{B} \in \mathcal{L}(\mathcal{Z}_D)} \mathcal{J}_D(\boldsymbol{B})$ for the weak population objective Equation* (38)*. Suppose the selected minimizers satisfy the finite-time stability condition in Assumption A.8, and the distribution family $\{\mu_s\}_{0 \le s \le 2\tau}$ satisfies FT (Assumption A.11). Then, for every $\boldsymbol{z} \in \mathcal{Z}$ and $H > 0$,*

$$\sup_{0 \le t \le H} \|K_D(t)\boldsymbol{z} - T(t)\boldsymbol{z}\| \longrightarrow 0 \qquad (D \to \infty). \tag{45}$$

The same conclusion holds for $S_D(t) = K_D(t) + (I - P_D)$.

Proof. Step 1: Wasserstein convergence of the averaged measures. Theorem A.10 gives

$$e_D := \sup_{s,t \in [0,\tau]} W_2(S_D(t)_{\#}\mu_s, \mu_{s+t}) \longrightarrow 0. \tag{46}$$

For $t \in [0, \tau]$, define

$$\bar{\mu} = \frac{1}{\tau}\int_0^\tau \mu_s \, \mathrm{d}s, \qquad \bar{\mu}_t = \frac{1}{\tau}\int_0^\tau \mu_{s+t} \, \mathrm{d}s.$$

Both measures belong to $\mathcal{P}_2(\mathcal{Z})$ because $r \mapsto \boldsymbol{Z}_r$ is continuous in $L^2(\Omega; \mathcal{Z})$ on $[0, 2\tau]$. Proposition A.12 gives $\overline{\mathrm{span}(\mathrm{supp}\, \bar{\mu})} = \mathcal{Z}$. For $r, t \in [0, \tau]$, using the same $\boldsymbol{Z}_0$ and an independent uniform time in $[0, \tau]$ to couple the averaged measures yields

$$W_2(\bar{\mu}_t, \bar{\mu}_r)^2 \le \frac{1}{\tau}\int_0^\tau \|\boldsymbol{Z}_{s+t} - \boldsymbol{Z}_{s+r}\|_{L^2}^2 \, \mathrm{d}s \longrightarrow 0 \qquad (t \to r).$$

Here the convergence follows from uniform L^2 continuity of $\boldsymbol{Z}_r$ on $[0, 2\tau]$. Averaging the bounds in Equation (46) gives

$$W_2(S_D(t)_\# \bar{\mu}, \bar{\mu}_t)^2 \le \frac{1}{\tau}\int_0^\tau W_2(S_D(t)_\# \mu_s, \mu_{s+t})^2 \, \mathrm{d}s \le e_D^2. \tag{47}$$

To justify this inequality, first approximate each time integral by a Riemann sum on a uniform partition. Averaging optimal couplings for the finitely many pairs of measures proves the inequality for these sums. For fixed D, t, the discrete averages converge in W_2 to $S_D(t)_\#\bar{\mu}$ and $\bar{\mu}_t$: for s in a partition interval with left endpoint s_j, use the pairs $(S_D(t)\boldsymbol{Z}_s, S_D(t)\boldsymbol{Z}_{s_j})$ and $(\boldsymbol{Z}_{s+t}, \boldsymbol{Z}_{s_j+t})$, respectively. Uniform L^2 continuity and boundedness of $S_D(t)$ make the coupling costs vanish. The integrand on the right is continuous in s, since both curves of measures are W_2-continuous. Passing to the limit proves Equation (47).

Step 2: Strong operator subsequences for convergent time lags. Take arbitrary sequences $D_n \to \infty$ and $t_n \in [0, \tau]$. After passing to a subsequence, compactness of $[0, \tau]$ gives $t_n \to t_*$. By Equation (47) and continuity of $t \mapsto \bar{\mu}_t$,

$$W_2((S_{D_n}(t_n))_\#\bar{\mu}, \bar{\mu}_{t_*}) \le e_{D_n} + W_2(\bar{\mu}_{t_n}, \bar{\mu}_{t_*}) \longrightarrow 0.$$

Assumption A.8 gives $\|S_{D_n}(t_n)\| \le M$. Since $\mathrm{supp}\, \bar{\mu}$ has dense linear span, Lemma A.13 applies and provides a further subsequence, with the same notation, such that

$$S_{D_n}(t_n)\boldsymbol{z} \longrightarrow B\boldsymbol{z} \quad \text{for every } \boldsymbol{z} \in \mathcal{Z}, \qquad B \in \mathcal{L}(\mathcal{Z}), \quad \|B\| \le M. \tag{48}$$

Step 3: Identification of subsequential limits under FT. Fix $s \in [0, \tau]$. The pair $(S_{D_n}(t_n)\boldsymbol{Z}_s, B\boldsymbol{Z}_s)$ gives

$$W_2((S_{D_n}(t_n))_\#\mu_s, B_\#\mu_s)^2 \le \mathbb{E}\|(S_{D_n}(t_n) - B)\boldsymbol{Z}_s\|^2 \longrightarrow 0.$$

Indeed, Equation (48) gives pointwise convergence, and the integrand is bounded by $4M^2\|\boldsymbol{Z}_s\|^2$, which is integrable. Combining this limit with Equation (46) and W_2 continuity of $r \mapsto \mu_r$ gives

$$W_2(B_\#\mu_s, \mu_{s+t_*}) \le W_2(B_\#\mu_s, (S_{D_n}(t_n))_\#\mu_s) + e_{D_n} + W_2(\mu_{s+t_n}, \mu_{s+t_*}) \longrightarrow 0.$$

The subsequence in Equation (48) is independent of s and converges at every $\boldsymbol{z} \in \mathcal{Z}$. Thus the same argument applies to every $s \in [0, \tau]$ without further extraction, establishing $B_\#\mu_s = \mu_{s+t_*}$ for all such s. Assumption A.11 therefore implies $B = T(t_*)$, including $B = I$ when $t_* = 0$.

Step 4: Uniform strong convergence on $[0, \tau]$. If convergence failed to be uniform for some fixed $\boldsymbol{z} \in \mathcal{Z}$, there would exist $\varepsilon > 0$, $D_n \to \infty$, and $t_n \in [0, \tau]$ with $\|S_{D_n}(t_n)\boldsymbol{z} - T(t_n)\boldsymbol{z}\| \ge \varepsilon$. Steps 2–3 give a further subsequence for which $t_n \to t_*$ and $S_{D_n}(t_n)\boldsymbol{z} \to T(t_*)\boldsymbol{z}$. Strong continuity gives $T(t_n)\boldsymbol{z} \to T(t_*)\boldsymbol{z}$, a contradiction. Consequently,

$$\sup_{0 \le t \le \tau} \|S_D(t)\boldsymbol{z} - T(t)\boldsymbol{z}\| \longrightarrow 0 \qquad \text{for every } \boldsymbol{z} \in \mathcal{Z}. \tag{49}$$

Step 5: Extension to arbitrary finite time intervals. For each fixed integer $k \ge 0$, $S_D(\tau)^k\boldsymbol{z} \to T(\tau)^k\boldsymbol{z}$. This follows inductively from Equation (49) and $\|S_D(\tau)\| \le M$, using

$$S_D(\tau)^{k+1}\boldsymbol{z} - T(\tau)^{k+1}\boldsymbol{z} = S_D(\tau)\big(S_D(\tau)^k\boldsymbol{z} - T(\tau)^k\boldsymbol{z}\big) + \big(S_D(\tau) - T(\tau)\big)T(\tau)^k\boldsymbol{z}.$$

For $t = k\tau + r$ with $0 \le r < \tau$, the semigroup property gives

$$\|S_D(t)\boldsymbol{z} - T(t)\boldsymbol{z}\| \le M\|S_D(\tau)^k\boldsymbol{z} - T(\tau)^k\boldsymbol{z}\| + \|(S_D(r) - T(r))T(\tau)^k\boldsymbol{z}\|.$$

For $t \in [0, H]$, only finitely many k occur. The first term tends to zero uniformly in r, and Equation (49), applied to the finitely many fixed vectors $T(\tau)^k \boldsymbol{z}$, makes the second term vanish uniformly in r. Hence $\sup_{0\le t\le H} \|S_D(t)\boldsymbol{z} - T(t)\boldsymbol{z}\| \to 0$. Finally, $S_D(t) - K_D(t) = I - P_D$ gives

$$\sup_{0\le t\le H} \|K_D(t)\boldsymbol{z} - T(t)\boldsymbol{z}\| \le \sup_{0\le t\le H} \|S_D(t)\boldsymbol{z} - T(t)\boldsymbol{z}\| + \|(I - P_D)\boldsymbol{z}\| \longrightarrow 0,$$

which proves Equation (45). □

The theorem establishes strong operator convergence locally uniformly in time, including on finite intervals beyond $[0, 2\tau]$. The extension relies on the global semigroup property after stability and FT have established convergence on $[0, \tau]$. Korda & Mezić (2018) analyze EDMD, which fits the Koopman action on a finite observable space by least squares from paired state observations. For a fixed observable space, the infinite-sample limit projects each evolved observable back onto that space in the L^2 inner product induced by the sampling measure. Under their assumptions, strong operator convergence then follows as the observable space is enlarged. Here, convergence follows from the weak population objective, compactness of pushforward measures, and FT.

Limitations. Our convergence analysis as $D \to \infty$ assumes an underlying autonomous vector field whose induced distributions form an absolutely continuous Wasserstein flow. It concerns minimizers of the exact weak population residual over a compatible family of observables obtained from nested projections of a fixed Koopman-invariant RKHS. Under the regularity conditions in Sections A.3.1–A.3.2, the stability condition in Assumption A.8, and finite-time linear determinacy (FT) in Assumption A.11, the feature propagators $K_D(t)$ and $S_D(t)$ converge strongly to $T(t) = U(t)^*$, locally uniformly in time. In practice, only finite samples at sparse observation times are available, so we discretize the time integrals using trapezoidal quadrature. We further use ridge regularization and exponential moving average (EMA) updates to stabilize the estimated generator during alternating optimization of neural representations and dynamics (Appendix B). The analysis does not control the resulting sampling, quadrature, regularization, or optimization errors, nor establish convergence of this training procedure or of independently learned neural representations as D increases. Strong convergence of the adjoints to $U(t)$, operator-norm convergence, and spectral convergence also remain outside these guarantees.

B Algorithmic Details

This appendix develops the training objectives and algorithms for KoopCell and its memory-augmented extension, KoopCell-M, introduced in Sections 3.1 and 3.2. We first describe how KoopCell alternates between VAE representation learning and closed-form estimation of the latent dynamics. We then introduce the memory embedding and developmental-path conditioning in KoopCell-M, detailing the resulting inference and training procedures. Throughout, $\rho_{t_k} = n_k^{-1} \sum_{i=1}^{n_k} \delta_{\boldsymbol{x}_{k,i}}$ denotes an observed snapshot, with $\boldsymbol{x}_{k,i} \in \mathbb{R}^{d_x}$ and $\mathcal{T} = \{0 = t_0 < \cdots < t_N = T\}$ the training times. We retain $\boldsymbol{\theta}, \boldsymbol{\phi}$ for the encoder and decoder parameters and D for the latent dimension. Held-out snapshots are used only for evaluation.

B.1 KoopCell: Koopman dynamics and alternating training

KoopCell alternates between learning a VAE representation for a given latent dynamics generator and estimating the linear dynamics generator in the current latent coordinates. We describe the VAE objective first, then derive the empirical generator update from the finite-test formulation proposed in Theorem 3.1.

B.1.1 VAE representation learning

For a cell with expression $\boldsymbol{x}$ observed at time t, we use a posterior encoder $q_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}, t) = \mathcal{N}(\boldsymbol{m}_{\boldsymbol{\theta}}(\boldsymbol{x}, t), \boldsymbol{S}_{\boldsymbol{\theta}}(\boldsymbol{x}, t))$ with diagonal covariance $\boldsymbol{S}_{\boldsymbol{\theta}}$. A shared MLP takes $(\boldsymbol{x}, t)$ as input and predicts the mean and log-variance. The observation model is $p_{\boldsymbol{\phi}}(\boldsymbol{x} \mid \boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}; \mathrm{Dec}_{\boldsymbol{\phi}}(\boldsymbol{z}), \sigma_x^2 \boldsymbol{I}_{d_x})$, with fixed variance σ_x^2. Reconstruction and population matching use reparameterized samples $\boldsymbol{z} = \boldsymbol{m}_{\boldsymbol{\theta}} + \boldsymbol{S}_{\boldsymbol{\theta}}^{1/2} \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_D)$, and predicted expression is given by $\mathrm{Dec}_{\boldsymbol{\phi}}(\boldsymbol{z})$. The resulting encoded population is $\mu_t = \int q_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}, t) \mathrm{d}\rho_t(\boldsymbol{x})$.

Given the linear latent flow $\mathcal{F}_t(\boldsymbol{z}) = \boldsymbol{F}_t\boldsymbol{z} + \boldsymbol{c}_t$ (Appendix B.1.2), we regularize the representation with the transported prior $p_t = (\mathcal{F}_t)_{\#}\mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_D) = \mathcal{N}(\boldsymbol{c}_t, \boldsymbol{\Gamma}_t)$, where $\boldsymbol{\Gamma}_t = \boldsymbol{F}_t\boldsymbol{F}_t^\top$. Here μ_t denotes the encoder-induced population and p_t the generative prior. We write $q_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}, t)$ for the encoder distribution and $q_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x}, t)$ for its density at $\boldsymbol{z}$. Within fixed-time expectations, we suppress time subscripts on random states. The resulting generative model $p_{\boldsymbol{\phi},t}(\boldsymbol{x}, \boldsymbol{z}) = p_{\boldsymbol{\phi}}(\boldsymbol{x} \mid \boldsymbol{z})p_t(\boldsymbol{z})$ yields $p_{\boldsymbol{\phi},t}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{Z}\sim q_{\boldsymbol{\theta}}(\cdot\mid\boldsymbol{x},t)}[p_{\boldsymbol{\phi}}(\boldsymbol{x} \mid \boldsymbol{Z})p_t(\boldsymbol{Z})/q_{\boldsymbol{\theta}}(\boldsymbol{Z} \mid \boldsymbol{x}, t)]$. Jensen's inequality therefore gives the evidence lower bound (ELBO)

$$\log p_{\boldsymbol{\phi},t}(\boldsymbol{x}) \geq \mathcal{E}_{\mathrm{KC}}(\boldsymbol{x}, t) := \mathbb{E}_{q_{\boldsymbol{\theta}}} \log p_{\boldsymbol{\phi}}(\boldsymbol{x} \mid \boldsymbol{Z}) - \mathrm{KL}\big(q_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}, t) \,\|\, p_t\big). \tag{50}$$

The Gaussian observation model gives a squared reconstruction error up to an additive constant. Following Equation (4), we normalize it by the expression dimension: $\mathcal{L}_{\mathrm{rec}} = -d_x^{-1}\mathbb{E}_{t,\boldsymbol{x}}\mathbb{E}_{q_{\boldsymbol{\theta}}} \log p_{\boldsymbol{\phi}}(\boldsymbol{x} \mid \boldsymbol{Z})$, where $\mathbb{E}_{t,\boldsymbol{x}}$ averages equally over training times and uniformly over cells within each time. Writing $\boldsymbol{m} = \boldsymbol{m}_{\boldsymbol{\theta}}(\boldsymbol{x}, t)$ and $\boldsymbol{S} = \boldsymbol{S}_{\boldsymbol{\theta}}(\boldsymbol{x}, t)$, we compute the Gaussian KL divergence analytically:

$$\mathrm{KL}\big(\mathcal{N}(\boldsymbol{m}, \boldsymbol{S}) \,\|\, p_t\big) = \tfrac{1}{2}\left[\mathrm{tr}(\boldsymbol{\Gamma}_t^{-1}\boldsymbol{S}) + (\boldsymbol{m} - \boldsymbol{c}_t)^\top\boldsymbol{\Gamma}_t^{-1}(\boldsymbol{m} - \boldsymbol{c}_t) - D + \log\frac{\det\boldsymbol{\Gamma}_t}{\det\boldsymbol{S}}\right]. \tag{51}$$

The training objective $\mathcal{L}_{\mathrm{VAE}} = \mathcal{L}_{\mathrm{rec}} + \beta\mathbb{E}_{t,\boldsymbol{x}}\,\mathrm{KL}(q_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}, t) \,\|\, p_t)$ thus equals the negative ELBO divided by d_x when $\beta = 1/d_x$; otherwise, β controls the strength of the variational regularization.

To learn representations whose evolution under the latent flow matches the observed populations, we augment $\mathcal{L}_{\mathrm{VAE}}$ with prediction losses in latent and expression space. For every forward pair $(s, t) \in \mathcal{P} = \{(s, t) \in \mathcal{T}^2 : s < t\}$, we form $\widehat{\mu}_{s\to t} = (\mathcal{F}_{t-s})_{\#}\mu_s$ and $\widehat{\rho}_{s\to t} = (\mathrm{Dec}_{\boldsymbol{\phi}})_{\#}\widehat{\mu}_{s\to t}$, and average the prediction losses over these pairs:

$$\mathcal{L}_{\mathrm{pop}} = \frac{1}{|\mathcal{P}|}\sum_{(s,t)\in\mathcal{P}}\big[\lambda_z\mathcal{S}_\varepsilon(\mu_t, \widehat{\mu}_{s\to t}) + \lambda_x\mathcal{S}_\varepsilon(\rho_t, \widehat{\rho}_{s\to t})\big]. \tag{52}$$

Here $\mathcal{S}_\varepsilon$ is the debiased Sinkhorn divergence with quadratic cost. With the dynamics fixed, we update $\boldsymbol{\theta}, \boldsymbol{\phi}$ by minimizing $\mathcal{L}_{\mathrm{KC}} = \mathcal{L}_{\mathrm{VAE}} + \mathcal{L}_{\mathrm{pop}} + \lambda_{\mathrm{weak}}\mathcal{L}_{\mathrm{weak}}$, where the weak-residual loss is defined below. Reconstruction and prediction losses update both the encoder and decoder; the weak-residual term updates only the encoder.

To initialize the representation before alternating training, we first pretrain the VAE with a standard-normal prior at every training time, minimizing

$$\mathcal{L}_{\mathrm{pre}} = \mathcal{L}_{\mathrm{rec}} + \beta_{\mathrm{pre}}\mathbb{E}_{t,\boldsymbol{x}}\,\mathrm{KL}\big(q_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}, t) \,\|\, \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_D)\big), \tag{53}$$

with a gradually increased β_{pre}. The pretrained encoder then provides the initial latent coordinates for the linear dynamics estimation described next.

B.1.2 Empirical weak residual and closed-form dynamics estimation

With the representation fixed, we now derive a closed-form estimator for the linear generator that best fits the evolution of the latent distributions. For notational simplicity, Section 3.1 presents the method using $\dot{\boldsymbol{Z}}_t = \boldsymbol{A}\boldsymbol{Z}_t$. In the implementation, we include the constant observable 1, giving $\dot{\boldsymbol{Z}}_t = \boldsymbol{A}\boldsymbol{Z}_t + \boldsymbol{b}_z$ and coefficients $\boldsymbol{\Theta} = [\boldsymbol{A}, \boldsymbol{b}_z] \in \mathbb{R}^{D\times(D+1)}$. Writing $\widetilde{\boldsymbol{z}} = (\boldsymbol{z}^\top, 1)^\top$ and $\widetilde{\boldsymbol{A}} = \left(\begin{smallmatrix}\boldsymbol{A} & \boldsymbol{b}_z\\ \boldsymbol{0}^\top & 0\end{smallmatrix}\right)$, we evaluate the flow through

$$\exp(\Delta\widetilde{\boldsymbol{A}}) = \begin{pmatrix}\boldsymbol{F}_\Delta & \boldsymbol{c}_\Delta\\ \boldsymbol{0}^\top & 1\end{pmatrix}, \qquad \mathcal{F}_\Delta(\boldsymbol{z}) = \boldsymbol{F}_\Delta\boldsymbol{z} + \boldsymbol{c}_\Delta. \tag{54}$$

The constant coordinate stays equal to one, and only the upper D rows are estimated.

Theorem 3.1 shows that restricting the dual supremum to a finite test space gives a least-squares regression objective for estimating the generator; Lemma A.4 in Appendix A.2 provides the corresponding derivation. For the linear drift $\boldsymbol{\Theta}\widetilde{\boldsymbol{z}}$, the moment definitions in Equation (3) require only replacing $\boldsymbol{z}^\top$ by $\widetilde{\boldsymbol{z}}^\top$ in $\boldsymbol{H}_j$; $\boldsymbol{y}$ and $\boldsymbol{M}$ are unchanged. Thus, with $\boldsymbol{a} = \mathrm{vec}(\boldsymbol{\Theta})$, the residual vector remains $\boldsymbol{y} - \boldsymbol{G}\boldsymbol{a}$ and the objective is $\frac{1}{2}(\boldsymbol{y} - \boldsymbol{G}\boldsymbol{a})^\top\boldsymbol{M}^{-1}(\boldsymbol{y} - \boldsymbol{G}\boldsymbol{a})$ when $\boldsymbol{M} \succ 0$. We next construct its empirical counterpart from the observed snapshots by forming these moments on adjacent time intervals.

Specifically, on each interval $I_k = [t_k, t_{k+1}]$, we use a smooth spatial test ψ_j with bounded values and gradients. The residual identity Equation (14) in Appendix A.1, applied to $\varphi_{kj}(t, \boldsymbol{z}) = \mathbf{1}_{I_k}(t)\psi_j(\boldsymbol{z})$, gives[2]

$$\mathcal{R}_D(\boldsymbol{\Theta}; \varphi_{kj}) = \mu_{t_{k+1}}[\psi_j] - \mu_{t_k}[\psi_j] - \left\langle \boldsymbol{\Theta}, \int_{I_k} \mu_t[\nabla\psi_j(\boldsymbol{z})\widetilde{\boldsymbol{z}}^\top]\mathrm{d}t \right\rangle_F. \tag{55}$$

Here the endpoint difference supplies y_{kj}, and the time integral supplies $\boldsymbol{H}_{kj}$ in the finite-test construction. The Gram matrix defined in Equation (27) is block diagonal, $\boldsymbol{M} = \mathrm{diag}(\boldsymbol{M}_0, \ldots, \boldsymbol{M}_{N-1})$; each block is $(\boldsymbol{M}_k)_{ij} = \int_{I_k} \mu_t[\nabla\psi_i^\top \nabla\psi_j]\mathrm{d}t$.

In the implementation, we use random Fourier tests $\cos(\boldsymbol{\xi}_j^\top \boldsymbol{z})$ and $\sin(\boldsymbol{\xi}_j^\top \boldsymbol{z})$ to construct this finite test space. Frequencies are sampled once from $\mathcal{N}(\boldsymbol{0}, s^2\boldsymbol{I}_D/D)$ in equal-sized groups for $s \in \{0.5, 1, 2, 4\}$, then held fixed during training. The standard bank contains 1,024 frequencies and hence 2,048 tests. Their gradients, $-\boldsymbol{\xi}_j \sin(\boldsymbol{\xi}_j^\top \boldsymbol{z})$ and $\boldsymbol{\xi}_j \cos(\boldsymbol{\xi}_j^\top \boldsymbol{z})$, allow all required moments to be evaluated directly from encoded cells.

For the empirical regression, we choose the encoder means as the deterministic observables to which the theorem is applied. At each training time t_k, we encode the observed cells as $\boldsymbol{z}_{k,i} = \boldsymbol{m}_{\boldsymbol{\theta}}(\boldsymbol{x}_{k,i}, t_k)$, yielding the empirical latent distribution $\mu_{t_k}^{\mathrm{mean}} = (\boldsymbol{m}_{\boldsymbol{\theta}}(\cdot, t_k))_{\#}\rho_{t_k}$. With $\widetilde{\boldsymbol{z}}_{k,i} = (\boldsymbol{z}_{k,i}^\top, 1)^\top$, the endpoint statistics are

$$u_{kj} = \frac{1}{n_k}\sum_i \psi_j(\boldsymbol{z}_{k,i}), \qquad \boldsymbol{D}_{kj} = \frac{1}{n_k}\sum_i \nabla\psi_j(\boldsymbol{z}_{k,i})\widetilde{\boldsymbol{z}}_{k,i}^\top. \tag{56}$$

In Equation (55), u_{kj} replaces the endpoint expectation $\mu_{t_k}[\psi_j]$, and $\boldsymbol{D}_{kj}$ replaces the integrand $\mu_{t_k}[\nabla\psi_j(\boldsymbol{z})\widetilde{\boldsymbol{z}}^\top]$ at t_k. Approximating the time integral by the trapezoidal rule, with $\Delta t_k = t_{k+1} - t_k$, therefore gives

$$y_{kj} = u_{k+1,j} - u_{kj}, \qquad \boldsymbol{H}_{kj} = \frac{\Delta t_k}{2}(\boldsymbol{D}_{kj} + \boldsymbol{D}_{k+1,j}), \qquad \boldsymbol{G}_{(k,j),:} = \mathrm{vec}(\boldsymbol{H}_{kj})^\top. \tag{57}$$

Following the regression interpretation in the remark after Lemma A.4, the empirical residual over interval $I_k = [t_k, t_{k+1}]$ is $\widehat{\mathcal{R}}_{kj}(\boldsymbol{\Theta}) = y_{kj} - \langle \boldsymbol{\Theta}, \boldsymbol{H}_{kj}\rangle_F = y_{kj} - (\boldsymbol{G}\boldsymbol{a})_{(k,j)}$. This is the empirical, time-discretized approximation of $\mathcal{R}_D(\boldsymbol{\Theta}; \varphi_{kj})$ in Equation (55). The construction accommodates irregular observation times; further, if the moment integrand has two bounded time derivatives, the interval quadrature error is $\mathcal{O}(\Delta t_k^3)$. Applying the same quadrature to the Gram matrix gives

$$(\boldsymbol{M}_k)_{ij} = \frac{\Delta t_k}{2}\left\{\mu_{t_k}^{\mathrm{mean}}[\nabla\psi_i^\top\nabla\psi_j] + \mu_{t_{k+1}}^{\mathrm{mean}}[\nabla\psi_i^\top\nabla\psi_j]\right\}. \tag{58}$$

Finally, we stack these empirical residuals over all interval–test pairs (k, j) to obtain $\boldsymbol{y} - \boldsymbol{G}\boldsymbol{a}$, the residual vector in the finite-test objective of Equation (28).

With these empirical quantities and $\boldsymbol{W} = (\boldsymbol{M} + \epsilon_M \boldsymbol{I})^{-1}$, we define $\mathcal{L}_{\mathrm{weak}} = (\boldsymbol{G}\boldsymbol{a} - \boldsymbol{y})^\top \boldsymbol{W}(\boldsymbol{G}\boldsymbol{a} - \boldsymbol{y})$. To stabilize the estimation when $\boldsymbol{G}$ is poorly conditioned, we additionally introduce a ridge penalty:

$$\widehat{\boldsymbol{a}} = \arg\min_{\boldsymbol{a}} \left\{\tfrac{1}{2}(\boldsymbol{G}\boldsymbol{a} - \boldsymbol{y})^\top \boldsymbol{W}(\boldsymbol{G}\boldsymbol{a} - \boldsymbol{y}) + \tfrac{\lambda}{2}\|\boldsymbol{a}\|_2^2\right\} = (\boldsymbol{G}^\top \boldsymbol{W}\boldsymbol{G} + \lambda\boldsymbol{I})^{-1}\boldsymbol{G}^\top \boldsymbol{W}\boldsymbol{y}, \qquad \lambda > 0. \tag{59}$$

We form the regression statistics from all training cells and solve the normal equations using preconditioned conjugate gradients. The fitted coefficients $\widehat{\boldsymbol{\Theta}} = \mathrm{reshape}(\widehat{\boldsymbol{a}})$ are then used to update the dynamics before refining the representation. However, the pretrained latent distributions need not follow a common linear flow, so directly using $\widehat{\boldsymbol{\Theta}}$ in the representation objective can induce abrupt updates and distort the pretrained latent manifold. We therefore introduce the closed-form dynamics generator gradually through an exponential moving average, $\boldsymbol{\Theta} \leftarrow \alpha\boldsymbol{\Theta} + (1-\alpha)\widehat{\boldsymbol{\Theta}}$, with $\alpha = 0.99$ in the real single-cell dynamics learning procedure.

With the updated $\boldsymbol{\Theta}$ held fixed, we optimize the VAE with the objective in Appendix B.1.1, recomputing the weak residual on minibatches. Algorithm 1 summarizes the alternating procedure. In both algorithms, sg stops gradients, $\|\boldsymbol{v}\|_{\boldsymbol{W}}^2 = \boldsymbol{v}^\top \boldsymbol{W}\boldsymbol{v}$, and $\mathrm{AdamW}(\boldsymbol{w}, \nabla_{\boldsymbol{w}}\mathcal{L})$ denotes one optimizer

[2]The interval test is understood by smooth approximation in $\mathbb{V}_D$, using the continuous residual extension in Equation (15).

Algorithm 1 KoopCell: alternating closed-form generator estimation and representation learning

Input: Training snapshots $\{(t_k, \rho_{t_k})\}_{k=0}^{N}$; fixed Fourier tests $\{\psi_j\}_{j=1}^{m}$; batch size n_{b}; $\lambda, \epsilon_M > 0$; $\alpha \in [0, 1)$; loss weights

Output: Encoder $q_{\boldsymbol{\theta}}$, decoder $\mathrm{Dec}_{\boldsymbol{\phi}}$, and linear dynamics parameters $\boldsymbol{\Theta} = [\boldsymbol{A}, \boldsymbol{b}_z]$

Stage 1: VAE pretraining

for each pretraining iteration **do**
  $\mathcal{B}_k \leftarrow \mathrm{Sample}(\rho_{t_k}, n_{\mathrm{b}}), \quad k = 0, \dots, N$ ▷ Independent sampling at each time
  $\mathcal{L}_{\mathrm{pre}} \leftarrow \mathrm{PretrainLoss}(\mathcal{B}; \boldsymbol{\theta}, \boldsymbol{\phi}, \beta_{\mathrm{pre}})$ ▷ Standard-normal VAE; Equation (53)
  $(\boldsymbol{\theta}, \boldsymbol{\phi}) \leftarrow \mathrm{AdamW}((\boldsymbol{\theta}, \boldsymbol{\phi}), \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathcal{L}_{\mathrm{pre}})$ ▷ Pretrain encoder and decoder
end for

Stage 2: Alternating dynamics estimation and representation learning

$\boldsymbol{\Theta} \leftarrow \mathbf{0}$
for each training iteration **do**
  (a) Closed-form dynamics estimation
  $\bar{\boldsymbol{z}}_{k,i} \leftarrow \mathrm{sg}(\boldsymbol{m}_{\boldsymbol{\theta}}(\boldsymbol{x}_{k,i}, t_k)), \quad k = 0, \dots, N,\ i = 1, \dots, n_k$
  $(\boldsymbol{G}, \boldsymbol{y}, \boldsymbol{M}) \leftarrow \mathrm{WeakStats}(\{\bar{\boldsymbol{z}}_{k,i}\}; \mathcal{T}, \{\psi_j\}); \boldsymbol{W} \leftarrow (\boldsymbol{M} + \epsilon_M \boldsymbol{I})^{-1}$
  $\widehat{\boldsymbol{a}} \leftarrow \mathrm{PCG}(\boldsymbol{G}^\top \boldsymbol{W} \boldsymbol{G} + \lambda \boldsymbol{I}, \boldsymbol{G}^\top \boldsymbol{W} \boldsymbol{y}); \widehat{\boldsymbol{\Theta}} \leftarrow \mathrm{reshape}(\widehat{\boldsymbol{a}})$ ▷ closed-form estimation; Equation (59)
  $\boldsymbol{\Theta} \leftarrow \mathrm{sg}(\alpha \boldsymbol{\Theta} + (1 - \alpha) \widehat{\boldsymbol{\Theta}})$ ▷ EMA update
  (b) Representation update
  $\mathcal{B}_k \leftarrow \mathrm{Sample}(\rho_{t_k}, n_{\mathrm{b}}), \quad k = 0, \dots, N$ ▷ Fresh minibatches for VAE updates
  $\bar{\boldsymbol{z}}^{\mathcal{B}}_{k,i} \leftarrow \boldsymbol{m}_{\boldsymbol{\theta}}(\boldsymbol{x}_{k,i}, t_k), \quad \boldsymbol{x}_{k,i} \in \mathcal{B}_k; \boldsymbol{\epsilon}_{k,i} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_D)$
  $\boldsymbol{z}_{k,i} \leftarrow \bar{\boldsymbol{z}}^{\mathcal{B}}_{k,i} + \boldsymbol{S}_{\boldsymbol{\theta}}(\boldsymbol{x}_{k,i}, t_k)^{1/2} \boldsymbol{\epsilon}_{k,i}$ ▷ Reparameterized posterior samples
  $(\boldsymbol{F}_\Delta, \boldsymbol{c}_\Delta) \leftarrow \mathrm{Transition}(\boldsymbol{\Theta}, \Delta), \quad \Delta \in \mathcal{T} \cup \{t - s : (s, t) \in \mathcal{P}\}$ ▷ Calculate the transition matrix
  $p_{t_k} \leftarrow \mathcal{N}(\boldsymbol{c}_{t_k}, \boldsymbol{F}_{t_k} \boldsymbol{F}_{t_k}^\top)$ ▷ Transported Gaussian prior
  $\mathcal{L}_{\mathrm{VAE}} \leftarrow \mathrm{VAELoss}(\mathcal{B}; \boldsymbol{\theta}, \boldsymbol{\phi}, \{p_{t_k}\})$ ▷ Reconstruction + KL; Equation (4)
  for each $(s, t) \in \mathcal{P}$ **in parallel do**
    $\widehat{\boldsymbol{z}}_{s \to t,i} \leftarrow \boldsymbol{F}_{t-s} \boldsymbol{z}_{s,i} + \boldsymbol{c}_{t-s}; \widehat{\boldsymbol{x}}_{s \to t,i} \leftarrow \mathrm{Dec}_{\boldsymbol{\phi}}(\widehat{\boldsymbol{z}}_{s \to t,i})$
  end for
  $\mathcal{L}_{\mathrm{pop}} \leftarrow \mathrm{PopulationLoss}(\mathcal{B}, \{\boldsymbol{z}_{k,i}\}, \{\widehat{\boldsymbol{z}}_{s \to t,i}, \widehat{\boldsymbol{x}}_{s \to t,i}\})$ ▷ Population matching; Equation (52)
  $(\boldsymbol{G}_{\mathcal{B}}, \boldsymbol{y}_{\mathcal{B}}, \boldsymbol{M}_{\mathcal{B}}) \leftarrow \mathrm{WeakStats}(\{\bar{\boldsymbol{z}}^{\mathcal{B}}_{k,i}\}; \mathcal{T}, \{\psi_j\})$
  $\boldsymbol{W}_{\mathcal{B}} \leftarrow (\boldsymbol{M}_{\mathcal{B}} + \epsilon_M \boldsymbol{I})^{-1}; \mathcal{L}_{\mathrm{weak}} \leftarrow \|\boldsymbol{G}_{\mathcal{B}} \mathrm{vec}(\boldsymbol{\Theta}) - \boldsymbol{y}_{\mathcal{B}}\|^2_{\boldsymbol{W}_{\mathcal{B}}}$ ▷ Weak population residual loss
  $\mathcal{L}_{\mathrm{KC}} \leftarrow \mathcal{L}_{\mathrm{VAE}} + \mathcal{L}_{\mathrm{pop}} + \lambda_{\mathrm{weak}} \mathcal{L}_{\mathrm{weak}}$ ▷ Combined training objective
  $(\boldsymbol{\theta}, \boldsymbol{\phi}) \leftarrow \mathrm{AdamW}((\boldsymbol{\theta}, \boldsymbol{\phi}), \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathcal{L}_{\mathrm{KC}})$ ▷ Update encoder and decoder
end for
return $q_{\boldsymbol{\theta}}, \mathrm{Dec}_{\boldsymbol{\phi}}, \boldsymbol{A}, \boldsymbol{b}_z$

update of parameters $\boldsymbol{w}$ using the indicated gradient. In Algorithm 1, $\mathrm{WeakStats}(\{\boldsymbol{z}_{k,i}\}; \mathcal{T}, \{\psi_j\})$ returns $(\boldsymbol{G}, \boldsymbol{y}, \boldsymbol{M})$ using Equation (56), Equation (57), and Equation (58); the same operation is used for full snapshots and minibatches. $\mathrm{Transition}(\boldsymbol{\Theta}, \Delta)$ returns $(\boldsymbol{F}_\Delta, \boldsymbol{c}_\Delta)$ via Equation (54), and $\mathrm{PCG}(\boldsymbol{Q}, \boldsymbol{r})$ solves $\boldsymbol{Q}\boldsymbol{a} = \boldsymbol{r}$ by preconditioned conjugate gradients. The loss routines evaluate the referenced objectives on the indicated minibatches $\mathcal{B} = \{\mathcal{B}_k\}_{k=0}^{N}$. Sample draws n_{b} cells independently from each snapshot. The pairwise predictions depend only on the source samples and fixed dynamics, so they can be computed in parallel.

For evaluation, we sample $\boldsymbol{Z}_0$ from the encoder distribution of the initial observed population snapshot and return $\mathrm{Dec}_{\boldsymbol{\phi}}(\mathcal{F}_t(\boldsymbol{Z}_0))$ at each requested time.

B.2 KoopCell-M: memory-augmented branching dynamics and alternating training

This section develops the memory extension introduced in Section 3.2. We first introduce the augmented latent dynamics for modeling branching, then derive the probabilistic model and training objectives. The resulting algorithm retains the two alternating steps of KoopCell: closed-form dynamics estimation followed by representation learning.

B.2.1 Memory-augmented dynamics and developmental paths

Zhao et al. (2026, Proposition 3.3) establishes that an ODE with a Lipschitz vector field cannot exactly transport between distributions supported on different numbers of disjoint compact, path-connected

components. In particular, a linear flow cannot split a connected initial population into disconnected fate populations; continuous decoding also preserves connectedness. When branching is understood as the emergence of separated density modes, a linear latent flow has a further limitation: it preserves Gaussianity, so a single Gaussian population remains unimodal. Although nonlinear decoding can introduce additional density modes, a decoder with Lipschitz constant L_{dec} limits the amplification of latent separation:

$$\|\operatorname{Dec}_{\boldsymbol{\phi}}(\mathcal{F}_t(\boldsymbol{z})) - \operatorname{Dec}_{\boldsymbol{\phi}}(\mathcal{F}_t(\boldsymbol{z}'))\| \le L_{\mathrm{dec}}\|e^{\boldsymbol{A}t}\|\,\|\boldsymbol{z}-\boldsymbol{z}'\|. \tag{60}$$

To model branching in the latent dynamics, we draw on the Mori–Zwanzig formalism: unresolved observables influence the resolved dynamics through memory (Section 2). Specifically, we augment the *visible latent state* $\boldsymbol{Z}_t \in \mathbb{R}^D$, which represents gene expression, with *memory variables* $\boldsymbol{H}_t \in \mathbb{R}^{d_h}$ that influence the dynamics of $\boldsymbol{Z}_t$. The decoder reconstructs gene expression from $\boldsymbol{Z}_t$ alone. The memory variables can represent fate-relevant influences not captured by the measured expression state (Weinreb et al., 2020).

Further, a linear augmented process with a single joint Gaussian initialization would still have Gaussian marginals. To generate distinct developmental branches, we therefore introduce a discrete variable $R \in \mathscr{R}$ indexing heterogeneous hidden initial conditions:

$$R \sim \mathrm{Cat}(\boldsymbol{\pi}), \qquad \boldsymbol{Z}_0 \mid R=r \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_D), \qquad \boldsymbol{H}_0 \mid \boldsymbol{Z}_0=\boldsymbol{z}, R=r \sim \mathcal{N}(\boldsymbol{F}_r\boldsymbol{z}+\boldsymbol{a}_r, \boldsymbol{S}_r). \tag{61}$$

Here π_r are normalized mixture weights and $\boldsymbol{S}_r \succ 0$. Conditional on $R=r$, we evolve the augmented state $(\boldsymbol{Z}_t, \boldsymbol{H}_t)$ by

$$\mathrm{d}\boldsymbol{Z}_t = (\boldsymbol{A}\boldsymbol{Z}_t + \boldsymbol{B}\boldsymbol{H}_t + \boldsymbol{b}_z)\,\mathrm{d}t, \qquad \mathrm{d}\boldsymbol{H}_t = (\boldsymbol{C}\boldsymbol{Z}_t + \boldsymbol{\Lambda}\boldsymbol{H}_t + \boldsymbol{b}_h(r))\,\mathrm{d}t + \boldsymbol{\Sigma}_h\,\mathrm{d}\boldsymbol{W}_t, \tag{62}$$

where $\boldsymbol{W}_t$ is a standard Brownian motion. Motivated by the use of decaying memory in related models (Lei et al., 2016; Menier et al., 2025), we parameterize $\boldsymbol{\Lambda} = (\boldsymbol{L}-\boldsymbol{L}^\top)/2 - \mathrm{diag}(\delta + \mathrm{softplus}(\boldsymbol{d}))$, with $\delta > 0$, making the homogeneous $\boldsymbol{H}$ block dissipative. Through the coupling $\boldsymbol{B}\boldsymbol{H}_t$, cells with similar visible states but different hidden initial conditions can follow distinct latent evolution. Each component remains Gaussian, but their mixture can develop multiple modes when their separation exceeds their spread. With nondegenerate component covariances, the mixture retains connected support; branching here refers to the separation of latent density modes.

To connect this augmented model with the Mori–Zwanzig formulation in Section 2, we eliminate $\boldsymbol{H}_t$ from Equation (62), obtaining a non-Markovian generalized Langevin equation for $\boldsymbol{Z}_t$:

$$\begin{aligned}\dot{\boldsymbol{Z}}_t &= \boldsymbol{A}\boldsymbol{Z}_t + \boldsymbol{b}_z + \int_0^t \boldsymbol{B}e^{\boldsymbol{\Lambda}(t-s)}\boldsymbol{C}\boldsymbol{Z}_s\,\mathrm{d}s + \boldsymbol{\eta}_r(t),\\ \boldsymbol{\eta}_r(t) &= \boldsymbol{B}e^{\boldsymbol{\Lambda}t}\boldsymbol{H}_0 + \int_0^t \boldsymbol{B}e^{\boldsymbol{\Lambda}(t-s)}\boldsymbol{b}_h(r)\,\mathrm{d}s + \boldsymbol{B}\int_0^t e^{\boldsymbol{\Lambda}(t-s)}\boldsymbol{\Sigma}_h\,\mathrm{d}\boldsymbol{W}_s.\end{aligned} \tag{63}$$

The kernel $\boldsymbol{K}(u) = \boldsymbol{B}e^{\boldsymbol{\Lambda}u}\boldsymbol{C}$ describes dependence on the past history, while $\boldsymbol{\eta}_r(t)$ collects the effects of the hidden initial state, bias, and noise. This decomposition into memory and forcing is motivated by Mori–Zwanzig reduction (Lin et al., 2021).

Example B.1 (Emergence of branching in the memory-augmented model)**.** Let $R \in \{-1, 1\}$ have equal probabilities, $Z_0 \mid R=r \sim \mathcal{N}(0,1)$, and $H_0 \mid Z_0=z, R=r \sim \mathcal{N}(-z+ra, \sigma_H^2)$, with $0 < \sigma_H < a < \sqrt{1+\sigma_H^2}$. Both initial marginals are unimodal: Z_0 is standard Gaussian, and H_0 is an equal mixture of $\mathcal{N}(\pm a, 1+\sigma_H^2)$. Consider the augmented dynamics $\mathrm{d}Z_t = (-Z_t + H_t)\,\mathrm{d}t$ and $\mathrm{d}H_t = (Z_t - H_t)\,\mathrm{d}t$ in Equation (62). Here $A = \Lambda = -1$ and $B = C = 1$, so eliminating H_t gives the memory kernel $K(u) = e^{-u}$.

Writing $\kappa_t = (1-e^{-2t})/2$, the solution is $Z_t = e^{-2t}Z_0 + \kappa_t(Z_0 + H_0)$. Conditional on $R=r$, the initialization makes $Z_0 + H_0$ independent of Z_0, with law $\mathcal{N}(ra, \sigma_H^2)$. Thus the visible density is

$$p_t(z) = \tfrac{1}{2}\mathcal{N}(z; -a\kappa_t, v_t) + \tfrac{1}{2}\mathcal{N}(z; a\kappa_t, v_t), \qquad v_t = e^{-4t} + \sigma_H^2\kappa_t^2. \tag{64}$$

The components coincide at $t=0$. When $a^2\kappa_t^2 > v_t$, $p_t''(0) = (a^2\kappa_t^2 - v_t)v_t^{-2}\mathcal{N}(0; a\kappa_t, v_t) > 0$; symmetry and decay at infinity then imply a maximum on each side of zero. This condition holds for $t > \frac{1}{2}\log(1 + 2/\sqrt{a^2-\sigma_H^2})$. As $t \to \infty$, the component means approach $\pm a/2$ and their variance approaches $\sigma_H^2/4$, so the two modes persist.

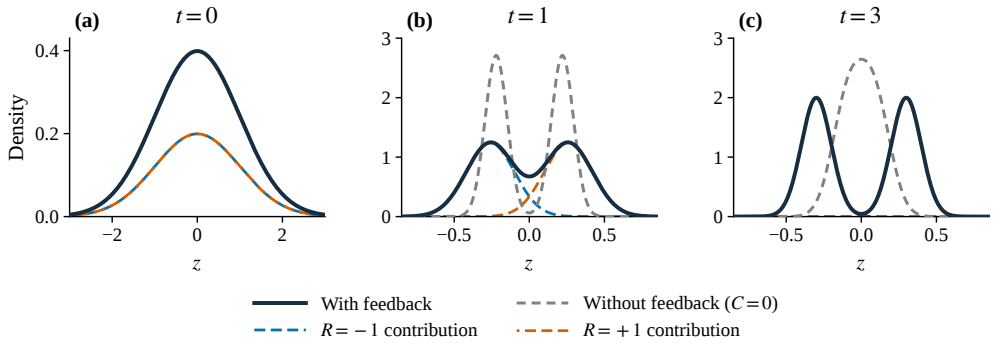


Figure 4: **Memory feedback sustains visible branching.** Analytic densities in Example B.1, with $a = 0.6$ and $\sigma_H = 0.2$. Both initial marginals are unimodal. The solid curve is the visible mixture with $K(u) = e^{-u}$; colored dashed curves illustrate its weighted components. The gray dashed curve removes only the feedback ($C = 0$), keeping the same initialization. Both models develop two modes at $t = 1$, but the comparison returns to a single mode by $t = 3$. Axis scales vary across panels.

To isolate the role of memory feedback, set $C = 0$ while keeping the other coefficients and initialization fixed. The component means then become $\pm ate^{-t}$, with common variance $e^{-2t}[(1-t)^2+\sigma_H^2 t^2]$. Since $a^2 < 1 + \sigma_H^2$, this comparison becomes unimodal again for sufficiently large t, as illustrated in Figure 4. The joint initialization is non-Gaussian: R encodes latent fate information, and the memory feedback sustains the resulting visible modes.

In this construction, R parameterizes hidden heterogeneity. For cellular development, we use R to represent *candidate developmental pathways*, whose progression is supervised using biological state annotations.

Biological state annotation. We use coarse biological states to model developmental paths, encompassing broad cell identities and developmental-stage categories. Initial training labels are obtained by matching cell IDs to published annotations for DR (Calderon et al., 2022) and WOT cell sets for SC (Schiebinger et al., 2019). For ZB, we annotate Louvain clusters at the latest training time using marker expression, following the terminal-cluster annotation approach of Farrell et al. (2018). We also define early developmental states using sampling time. For example, the first two training snapshots are labeled Blastula progenitor in ZB and Blastoderm in DR; the initial SC population is labeled MEF. For ZB and SC, we fit a logistic regression classifier to the initially labeled training cells and predict states for the remaining training cells. Logistic regression is also used for supervised cell annotation in CellTypist (Domínguez Conde et al., 2022). We calibrate the predicted probabilities and use only assignments that pass confidence, support, and consistency filters as additional supervision. For DR, state supervision uses only labels obtained from the published annotations or assigned using sampling time. Classifier fitting and label supervision use only training cells.

Candidate developmental paths. We use these annotated biological states as nodes to define a candidate directed acyclic graph (DAG). We specify its core edges using published developmental relationships (Farrell et al., 2018; Calderon et al., 2022; Schiebinger et al., 2019) and retain them as biological priors. Additional edges are selected using optimal transport between adjacent training snapshots, subject to biological compatibility and support criteria. An edge represents an admissible transition between coarse states. Enumerating root-to-terminal paths defines $\mathscr{R}$, with $r = (c_1, \ldots, c_{\ell_r})$ specifying an ordered developmental pathway. Each path parameterizes a hidden initial law in Equation (61). The path identity R remains fixed along a trajectory, while the biological state progresses along the pathway. We further introduce a path consistency loss to encourage generated trajectories to follow the developmental order specified by R (Equation (77) in Appendix B.2.2). Figure 5 shows the graph for SC, whose four root-to-terminal paths define the candidate values of R.

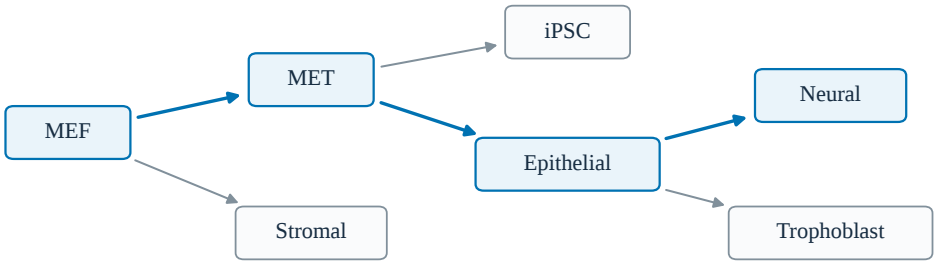


Figure 5: **Candidate developmental paths for SC.** Nodes represent seven coarse biological states, and edges represent the developmental transitions described by Schiebinger et al. (2019). The graph defines four root-to-terminal paths. The highlighted sequence MEF → MET → Epithelial → Neural illustrates one value of R. Each path parameterizes a hidden initial distribution and specifies the developmental order used in the path consistency loss (Equation (77)).

B.2.2 Variational inference and representation learning

The augmented latent dynamics define a time-dependent prior for learning the VAE representation. At a known sampling time t, the observed variable is a single-cell expression vector $\boldsymbol{X}_t$; the latent variables are the developmental path R, the visible state $\boldsymbol{Z}_t$, and the memory state $\boldsymbol{H}_t$.

Generative model. We draw $R \sim \mathrm{Cat}(\boldsymbol{\pi})$ and evolve $\boldsymbol{Y}_t = (\boldsymbol{Z}_t^\top, \boldsymbol{H}_t^\top)^\top$ from the path-dependent initialization in Equation (61) under Equation (62). This gives the Gaussian prior $p_{\boldsymbol{\vartheta},t}(\boldsymbol{z}, \boldsymbol{h} \mid r)$. Gene expression is generated from $\boldsymbol{Z}_t$ alone through the same observation model as KoopCell:

$$\begin{aligned} p_{\boldsymbol{\phi},\boldsymbol{\vartheta},t}(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{h}, r) &= \pi_r \, p_{\boldsymbol{\vartheta},t}(\boldsymbol{z}, \boldsymbol{h} \mid r) \, p_{\boldsymbol{\phi}}(\boldsymbol{x} \mid \boldsymbol{z}), \\ p_{\boldsymbol{\phi}}(\boldsymbol{x} \mid \boldsymbol{z}) &= \mathcal{N}\big(\boldsymbol{x}; \mathrm{Dec}_{\boldsymbol{\phi}}(\boldsymbol{z}), \sigma_x^2 \boldsymbol{I}_{d_x}\big). \end{aligned} \tag{65}$$

Here σ_x^2 is fixed, as in Appendix B.1.1, and $\boldsymbol{\vartheta}$ collects memory and initial-distribution parameters; dependence of the prior on $\boldsymbol{\Theta} = [\boldsymbol{A}, \boldsymbol{b}_z]$ is left implicit.

Inference model. Given expression $\boldsymbol{x}$ at time t, we approximate the joint posterior $p_{\boldsymbol{\phi},\boldsymbol{\vartheta},t}(\boldsymbol{z}, \boldsymbol{h}, r \mid \boldsymbol{x})$ by

$$q_t(\boldsymbol{z}, \boldsymbol{h}, r \mid \boldsymbol{x}) = q_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x}, t) \, q_t(r \mid \boldsymbol{z}, \boldsymbol{x}) \, q_t^H(\boldsymbol{h} \mid \boldsymbol{z}, r, \boldsymbol{x}). \tag{66}$$

The Gaussian encoder $q_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}, t) = \mathcal{N}(\boldsymbol{m}_{\boldsymbol{\theta}}(\boldsymbol{x}, t), \boldsymbol{S}_{\boldsymbol{\theta}}(\boldsymbol{x}, t))$ infers the visible latent state; $q_t(r \mid \boldsymbol{z}, \boldsymbol{x})$ assigns path probabilities, and $q_t^H(\boldsymbol{h} \mid \boldsymbol{z}, r, \boldsymbol{x})$ infers the memory state conditional on $\boldsymbol{z}$ and r. Using the factorization in Equation (65), Jensen's inequality gives

$$\begin{aligned} \log p_{\boldsymbol{\phi},\boldsymbol{\vartheta},t}(\boldsymbol{x}) &\geq \mathbb{E}_{q_t}\left[\log \frac{p_{\boldsymbol{\phi},\boldsymbol{\vartheta},t}(\boldsymbol{x}, \boldsymbol{Z}, \boldsymbol{H}, R)}{q_t(\boldsymbol{Z}, \boldsymbol{H}, R \mid \boldsymbol{x})}\right] \\ &= \mathbb{E}_{q_{\boldsymbol{\theta}}} \log p_{\boldsymbol{\phi}}(\boldsymbol{x} \mid \boldsymbol{Z}) - \mathrm{KL}\big(q_t(\cdot \mid \boldsymbol{x}) \,\|\, p_{\boldsymbol{\vartheta},t}\big), \end{aligned} \tag{67}$$

where $p_{\boldsymbol{\vartheta},t}(\boldsymbol{z}, \boldsymbol{h}, r) = \pi_r p_{\boldsymbol{\vartheta},t}(\boldsymbol{z}, \boldsymbol{h} \mid r)$. We next specify the memory and path posteriors in Equation (66) to evaluate this bound.

Conditional on $R = r$, the Gaussian initialization and linear augmented dynamics give a jointly Gaussian prior for $(\boldsymbol{Z}_t, \boldsymbol{H}_t)$. The conditional distribution $p_{\boldsymbol{\vartheta}}(\boldsymbol{h} \mid \boldsymbol{z}, r, t)$ is therefore Gaussian and can be computed in closed form. Because the observation model depends only on $\boldsymbol{Z}_t$, Equation (65) implies $p(\boldsymbol{H}_t \mid \boldsymbol{Z}_t, R, \boldsymbol{X}_t) = p(\boldsymbol{H}_t \mid \boldsymbol{Z}_t, R)$. We therefore use this Gaussian conditional as the memory posterior, setting $q_t^H(\boldsymbol{h} \mid \boldsymbol{z}, r, \boldsymbol{x}) = p_{\boldsymbol{\vartheta}}(\boldsymbol{h} \mid \boldsymbol{z}, r, t)$. To derive its mean and covariance, write $\mathbf{K} = \left(\begin{smallmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{\Lambda} \end{smallmatrix}\right)$, $\boldsymbol{b}_r = (\boldsymbol{b}_z^\top, \boldsymbol{b}_h(r)^\top)^\top$, and $\boldsymbol{\Sigma} = (\boldsymbol{0}, \boldsymbol{\Sigma}_h^\top)^\top$. The transition is $\boldsymbol{Y}_{t+\Delta} \mid \boldsymbol{Y}_t, R = r \sim \mathcal{N}(\boldsymbol{\Phi}_\Delta \boldsymbol{Y}_t + \boldsymbol{J}_\Delta \boldsymbol{b}_r, \boldsymbol{Q}_\Delta)$, with

$$\boldsymbol{\Phi}_\Delta = e^{\mathbf{K}\Delta}, \qquad \boldsymbol{J}_\Delta = \int_0^\Delta e^{\mathbf{K}s} \, \mathrm{d}s, \qquad \boldsymbol{Q}_\Delta = \int_0^\Delta e^{\mathbf{K}s} \boldsymbol{\Sigma}\boldsymbol{\Sigma}^\top e^{\mathbf{K}^\top s} \, \mathrm{d}s. \tag{68}$$

The initialization in Equation (61) gives

$$\boldsymbol{m}_{0,r} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{a}_r \end{pmatrix}, \qquad \boldsymbol{V}_{0,r} = \begin{pmatrix} \boldsymbol{I}_D & \boldsymbol{F}_r^\top \\ \boldsymbol{F}_r & \boldsymbol{F}_r \boldsymbol{F}_r^\top + \boldsymbol{S}_r \end{pmatrix}. \tag{69}$$

Hence $\boldsymbol{Y}_t \mid R = r \sim \mathcal{N}(\boldsymbol{m}_{t,r}, \boldsymbol{V}_{t,r})$, where $\boldsymbol{m}_{t,r} = \boldsymbol{\Phi}_t \boldsymbol{m}_{0,r} + \boldsymbol{J}_t \boldsymbol{b}_r$ and $\boldsymbol{V}_{t,r} = \boldsymbol{\Phi}_t \boldsymbol{V}_{0,r} \boldsymbol{\Phi}_t^\top + \boldsymbol{Q}_t$. Gaussian conditioning gives $q_t^H = \mathcal{N}(\bar{\boldsymbol{h}}_{t,r}(\boldsymbol{z}), \boldsymbol{V}_{t,r}^{h|z})$, with

$$\bar{\boldsymbol{h}}_{t,r}(\boldsymbol{z}) = \boldsymbol{m}_{t,r}^h + \boldsymbol{V}_{t,r}^{hz} (\boldsymbol{V}_{t,r}^{zz})^{-1} (\boldsymbol{z} - \boldsymbol{m}_{t,r}^z), \qquad \boldsymbol{V}_{t,r}^{h|z} = \boldsymbol{V}_{t,r}^{hh} - \boldsymbol{V}_{t,r}^{hz} (\boldsymbol{V}_{t,r}^{zz})^{-1} \boldsymbol{V}_{t,r}^{zh}, \tag{70}$$

Having specified the memory conditional q_t^H, we complete the inference model by defining the path probabilities $q_t(r \mid \boldsymbol{z}, \boldsymbol{x})$. The same joint Gaussian prior gives the visible marginal $p_{\boldsymbol{\vartheta},t}(\boldsymbol{z} \mid r) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{m}_{t,r}^z, \boldsymbol{V}_{t,r}^{zz})$. We combine this density with the path prior π_r and an annotation weight for biological states, $\chi_r(\boldsymbol{x})$, then normalize over candidate paths:

$$q_t(r \mid \boldsymbol{z}, \boldsymbol{x}) = \frac{\pi_r p_{\boldsymbol{\vartheta},t}(\boldsymbol{z} \mid r) \chi_r(\boldsymbol{x})}{\sum_{r' \in \mathscr{R}} \pi_{r'} p_{\boldsymbol{\vartheta},t}(\boldsymbol{z} \mid r') \chi_{r'}(\boldsymbol{x})}. \tag{71}$$

Here $\chi_r(\boldsymbol{x})$ sums the annotation probabilities over biological states on path r: an accepted state label restricts support to paths containing that state, soft annotations give soft weights, and an unlabeled cell uses $\chi_r(\boldsymbol{x}) = 1$. At least one compatible path is required for an annotated cell.

ELBO and training objective. With both posteriors specified, we return to the bound in Equation (67). Write $q_{\boldsymbol{\theta},t}(\boldsymbol{z}, r \mid \boldsymbol{x}) = q_{\boldsymbol{\theta}}(\boldsymbol{z} \mid \boldsymbol{x}, t) q_t(r \mid \boldsymbol{z}, \boldsymbol{x})$ and $p_{\boldsymbol{\vartheta},t}(\boldsymbol{z}, r) = \pi_r p_{\boldsymbol{\vartheta},t}(\boldsymbol{z} \mid r)$. Since $q_t^H(\boldsymbol{h} \mid \boldsymbol{z}, r, \boldsymbol{x}) = p_{\boldsymbol{\vartheta}}(\boldsymbol{h} \mid \boldsymbol{z}, r, t)$, the joint-density ratio in the bound becomes

$$\begin{aligned} \frac{p_{\boldsymbol{\phi},\boldsymbol{\vartheta},t}(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{h}, r)}{q_t(\boldsymbol{z}, \boldsymbol{h}, r \mid \boldsymbol{x})} &= \frac{p_{\boldsymbol{\phi}}(\boldsymbol{x} \mid \boldsymbol{z})\, p_{\boldsymbol{\vartheta},t}(\boldsymbol{z}, r)\, p_{\boldsymbol{\vartheta}}(\boldsymbol{h} \mid \boldsymbol{z}, r, t)}{q_{\boldsymbol{\theta},t}(\boldsymbol{z}, r \mid \boldsymbol{x})\, p_{\boldsymbol{\vartheta}}(\boldsymbol{h} \mid \boldsymbol{z}, r, t)} \\ &= \frac{p_{\boldsymbol{\phi}}(\boldsymbol{x} \mid \boldsymbol{z})\, p_{\boldsymbol{\vartheta},t}(\boldsymbol{z}, r)}{q_{\boldsymbol{\theta},t}(\boldsymbol{z}, r \mid \boldsymbol{x})}. \end{aligned} \tag{72}$$

The common memory conditional cancels, so the ratio no longer depends on $\boldsymbol{h}$. Taking its logarithm and expectation therefore integrates out $\boldsymbol{H}$ and gives

$$\begin{aligned} \log p_{\boldsymbol{\phi},\boldsymbol{\vartheta},t}(\boldsymbol{x}) \geq \mathcal{E}_{\mathrm{M}}(\boldsymbol{x}, t) &:= \mathbb{E}_{q_{\boldsymbol{\theta}}} \log p_{\boldsymbol{\phi}}(\boldsymbol{x} \mid \boldsymbol{Z}) + \mathbb{E}_{q_{\boldsymbol{\theta},t}} \log \frac{p_{\boldsymbol{\vartheta},t}(\boldsymbol{Z}, R)}{q_{\boldsymbol{\theta},t}(\boldsymbol{Z}, R \mid \boldsymbol{x})} \\ &= \mathbb{E}_{q_{\boldsymbol{\theta}}} \log p_{\boldsymbol{\phi}}(\boldsymbol{x} \mid \boldsymbol{Z}) - \mathrm{KL}\big(q_{\boldsymbol{\theta},t}(\cdot \mid \boldsymbol{x}) \,\|\, p_{\boldsymbol{\vartheta},t}\big). \end{aligned} \tag{73}$$

Expanding the remaining KL over the discrete paths yields

$$\mathrm{KL}(q_{\boldsymbol{\theta},t} \,\|\, p_{\boldsymbol{\vartheta},t}) = \mathbb{E}_{\boldsymbol{Z} \sim q_{\boldsymbol{\theta}}} \sum_{r \in \mathscr{R}} q_t(r \mid \boldsymbol{Z}, \boldsymbol{x}) \log \frac{q_{\boldsymbol{\theta}}(\boldsymbol{Z} \mid \boldsymbol{x}, t) q_t(r \mid \boldsymbol{Z}, \boldsymbol{x})}{\pi_r \mathcal{N}(\boldsymbol{Z}; \boldsymbol{m}_{t,r}^z, \boldsymbol{V}_{t,r}^{zz})}. \tag{74}$$

We enumerate the paths and estimate the expectation with reparameterized encoder samples. Thus the conditional hidden posterior is analytic, while the mixture KL is evaluated by sampling over $\boldsymbol{Z}$.

To encourage generated trajectories to follow the candidate developmental pathway represented by R, we introduce a path consistency loss based on the probabilities over biological states predicted by a latent state classifier $c_{\boldsymbol{\omega}}(c \mid \boldsymbol{z})$. We train the classifier using the annotations above by minimizing the cross-entropy loss $\mathcal{L}_{\mathrm{state}} = -\mathbb{E}_{t,\boldsymbol{x},c} \mathbb{E}_{\boldsymbol{Z} \sim q_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}, t)} \log c_{\boldsymbol{\omega}}(c \mid \boldsymbol{Z})$, where the outer expectation averages over training cells with supervised biological state labels c, and the inner expectation is over their encoder samples. We combine $\mathcal{L}_{\mathrm{state}}$ with the normalized reconstruction loss and KL regularization from Appendix B.1.1 to form the snapshot objective

$$\mathcal{L}_{\mathrm{snap}} = \mathcal{L}_{\mathrm{rec}} + \beta \mathbb{E}_{t,\boldsymbol{x}} \,\mathrm{KL}\big(q_{\boldsymbol{\theta},t}(\cdot \mid \boldsymbol{x}) \,\|\, p_{\boldsymbol{\vartheta},t}\big) + \lambda_c \mathcal{L}_{\mathrm{state}}. \tag{75}$$

As in KoopCell, we also match predicted and observed populations over all forward training pairs $s < t$. In KoopCell-M this requires inferring memory at the source time. The inferred augmented source population is

$$\widetilde{\mu}_s(\mathrm{d}\boldsymbol{z}, \mathrm{d}\boldsymbol{h}, r) = \int q_{\boldsymbol{\theta}}(\mathrm{d}\boldsymbol{z} \mid \boldsymbol{x}, s)\, q_s(r \mid \boldsymbol{z}, \boldsymbol{x})\, p_{\boldsymbol{\vartheta}}(\mathrm{d}\boldsymbol{h} \mid \boldsymbol{Z}_s = \boldsymbol{z}, R = r)\, \mathrm{d}\rho_s(\boldsymbol{x}). \tag{76}$$

We propagate $\widetilde{\mu}_s$ to time t using Equation (68), keeping r fixed along each trajectory. We denote the predicted $\boldsymbol{Z}_t$-marginal by $\widehat{\mu}_{s \to t}$ and map this latent population to gene-expression space through the decoder: $\widehat{\rho}_{s \to t} = (\mathrm{Dec}_{\boldsymbol{\phi}})_{\#} \widehat{\mu}_{s \to t}$. We compare these predictions with the corresponding populations at time t using $\mathcal{L}_{\mathrm{pop}}$ defined in Equation (52).

Path consistency loss. For a candidate developmental pathway $r = (c_1, \dots, c_{\ell_r})$, we sum over admissible timings of transitions between successive biological states. On a grid $0 = u_1 < \cdots < u_M = T$, let $\mathcal{I}_r$ contain index sequences with $i_1 = 1$, $i_m - i_{m-1} \in \{0, 1\}$, and $1 \le i_m \le \ell_r$. The classifier assigns the path-consistency probability

$$\mathcal{P}_r(\boldsymbol{Z}_{u_1:u_M}) = \sum_{(i_1,\dots,i_M)\in\mathcal{I}_r} \prod_{m=1}^{M} c_{\boldsymbol{\omega}}(c_{i_m} \mid \boldsymbol{Z}_{u_m}), \tag{77}$$

and we minimize $\mathcal{L}_{\text{path}} = -\sum_r \bar{q}_T(r)\mathbb{E}\log\mathcal{P}_r(\boldsymbol{Z}_{u_1:u_M})$. Here trajectories start from uniformly sampled initial cells conditional on each path, and $\bar{q}_T(r)$ is the detached mean path probability at the latest training time. We evaluate Equation (77) by forward dynamic programming, adapting the alignment marginalization in connectionist temporal classification (CTC; Graves et al., 2006) to the constraints in $\mathcal{I}_r$. The recursion uses no blank symbol and sums over all final path positions, allowing trajectories to end before the terminal state. The path loss updates both the classifier and the parameters governing the rollout. In Algorithm 2, $\boldsymbol{Z}^{(r)}_{u_1:u_M}$ denotes a batch of simulated visible trajectories conditional on $R = r$.

B.2.3 Closed-form dynamics estimation and alternating training

We now derive the closed-form update of $\boldsymbol{\Theta} = [\boldsymbol{A}, \boldsymbol{b}_z]$ with the representation and memory statistics fixed. Let $\widetilde{\mu}_t$ be the inferred augmented population in Equation (76), whose $\boldsymbol{Z}$-marginal is μ_t, and write $\widetilde{\boldsymbol{z}} = (\boldsymbol{z}^\top, 1)^\top$ and $\widetilde{\boldsymbol{Z}}_t = (\boldsymbol{Z}_t^\top, 1)^\top$. For a smooth test $\varphi(t, \boldsymbol{z})$, substituting the visible drift $\boldsymbol{\Theta}\widetilde{\boldsymbol{z}} + \boldsymbol{B}\boldsymbol{h}$ into the definition of the weak population residual in Equation (14) (Appendix A.1) gives

$$\begin{aligned}\mathcal{R}_D^{\mathrm{M}}(\boldsymbol{\Theta};\varphi) &= \mu_T[\varphi(T)] - \mu_0[\varphi(0)] \\ &\quad - \int_0^T \sum_{r\in\mathscr{R}} \int \big[\partial_t\varphi(t,\boldsymbol{z}) + \nabla_{\boldsymbol{z}}\varphi(t,\boldsymbol{z})^\top(\boldsymbol{\Theta}\widetilde{\boldsymbol{z}} + \boldsymbol{B}\boldsymbol{h})\big]\,\widetilde{\mu}_t(\mathrm{d}\boldsymbol{z},\mathrm{d}\boldsymbol{h},r)\,\mathrm{d}t \\ &= \mathcal{R}_D(\boldsymbol{\Theta};\varphi) - \int_0^T \sum_{r\in\mathscr{R}} \int \nabla_{\boldsymbol{z}}\varphi(t,\boldsymbol{z})^\top \boldsymbol{B}\boldsymbol{h}\,\widetilde{\mu}_t(\mathrm{d}\boldsymbol{z},\mathrm{d}\boldsymbol{h},r)\,\mathrm{d}t,\end{aligned} \tag{78}$$

where $\mathcal{R}_D(\boldsymbol{\Theta};\varphi)$ is the population residual used in Appendix B.1.2. Using Equation (76) and the conditional mean $\int \boldsymbol{h}\, p_{\boldsymbol{\vartheta}}(\mathrm{d}\boldsymbol{h} \mid \boldsymbol{z}, r, t) = \bar{\boldsymbol{h}}_{t,r}(\boldsymbol{z})$ from Equation (70) gives

$$\begin{aligned}&\sum_{r\in\mathscr{R}} \int \nabla_{\boldsymbol{z}}\varphi(t,\boldsymbol{z})^\top \boldsymbol{B}\boldsymbol{h}\,\widetilde{\mu}_t(\mathrm{d}\boldsymbol{z},\mathrm{d}\boldsymbol{h},r) \\ &\quad = \iint \nabla_{\boldsymbol{z}}\varphi(t,\boldsymbol{z})^\top \boldsymbol{B}\left[\sum_{r\in\mathscr{R}} q_t(r \mid \boldsymbol{z},\boldsymbol{x})\bar{\boldsymbol{h}}_{t,r}(\boldsymbol{z})\right] q_{\boldsymbol{\theta}}(\mathrm{d}\boldsymbol{z} \mid \boldsymbol{x}, t)\,\rho_t(\mathrm{d}\boldsymbol{x}) \\ &\quad = \mathbb{E}\big[\nabla_{\boldsymbol{z}}\varphi(t,\boldsymbol{Z}_t)^\top \boldsymbol{B}\bar{\boldsymbol{h}}_t(\boldsymbol{Z}_t,\boldsymbol{X}_t)\big],\end{aligned} \tag{79}$$

where $\bar{\boldsymbol{h}}_t(\boldsymbol{z},\boldsymbol{x}) = \sum_r q_t(r \mid \boldsymbol{z},\boldsymbol{x})\bar{\boldsymbol{h}}_{t,r}(\boldsymbol{z})$ and the expectation is over $\boldsymbol{X}_t \sim \rho_t$, $\boldsymbol{Z}_t \sim q_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{X}_t, t)$.

Applying the spatial test ψ_j on $I_k = [t_k, t_{k+1}]$, as in Equation (55), gives $\mathcal{R}_D^{\mathrm{M}}(\boldsymbol{\Theta};\varphi_{kj}) = y_{kj} - \langle\boldsymbol{\Theta}, \boldsymbol{H}_{kj}\rangle_F - c_{kj}$, where

$$\begin{aligned}y_{kj} &= \mu_{t_{k+1}}[\psi_j] - \mu_{t_k}[\psi_j], \qquad \boldsymbol{H}_{kj} = \int_{I_k} \mathbb{E}[\nabla\psi_j(\boldsymbol{Z}_t)\widetilde{\boldsymbol{Z}}_t^\top]\mathrm{d}t, \\ c_{kj} &= \int_{I_k} \mathbb{E}[\nabla\psi_j(\boldsymbol{Z}_t)^\top \boldsymbol{B}\bar{\boldsymbol{h}}_t(\boldsymbol{Z}_t,\boldsymbol{X}_t)]\mathrm{d}t, \qquad y^{\mathrm{M}}_{kj} = y_{kj} - c_{kj}.\end{aligned} \tag{80}$$

We estimate the expectations from encoded cells and approximate the time integrals using the trapezoidal rule, as in KoopCell (Equation (57)). Stacking the interval–test pairs gives the residual vector $\boldsymbol{y}^{\mathrm{M}} - \boldsymbol{G}\boldsymbol{a}$, with $\boldsymbol{a} = \mathrm{vec}(\boldsymbol{\Theta})$ and $\boldsymbol{G}_{(k,j),:} = \mathrm{vec}(\boldsymbol{H}_{kj})^\top$. Similarly, we solve the regularized regression used in KoopCell, replacing $\boldsymbol{y}$ with $\boldsymbol{y}^{\mathrm{M}} = \boldsymbol{y} - \boldsymbol{c}$:

$$\widehat{\boldsymbol{a}} = \arg\min_{\boldsymbol{a}} \left\{\tfrac{1}{2}\|\boldsymbol{y}^{\mathrm{M}} - \boldsymbol{G}\boldsymbol{a}\|^2_{\boldsymbol{W}} + \tfrac{\lambda}{2}\|\boldsymbol{a}\|_2^2\right\} = (\boldsymbol{G}^\top\boldsymbol{W}\boldsymbol{G} + \lambda\boldsymbol{I})^{-1}\boldsymbol{G}^\top\boldsymbol{W}\boldsymbol{y}^{\mathrm{M}}. \tag{81}$$

Algorithm 2 KoopCell-M: memory augmented latent dynamics for modeling branching

Input: Training snapshots $\mathcal{D} = \{(t_k, \rho_{t_k})\}_{k=0}^{N}$ with biological state annotations; candidate paths $\mathscr{R}$; fixed Fourier tests $\{\psi_j\}_{j=1}^{m}$; batch size n_{b}; $\lambda, \epsilon_M > 0$; $\alpha \in [0, 1)$; loss weights

Output: Encoder $\boldsymbol{q}_{\boldsymbol{\theta}}$, decoder $\mathrm{Dec}_{\boldsymbol{\phi}}$, state classifier $c_{\boldsymbol{\omega}}$, and dynamics parameters $\boldsymbol{\Theta} = [\boldsymbol{A}, \boldsymbol{b}_z]$, $\boldsymbol{\vartheta}$

Stage 1: VAE and state classifier pretraining

for each pretraining iteration **do**
  $\mathcal{B}_k \leftarrow \mathrm{Sample}(\rho_{t_k}, n_{\mathrm{b}}), \quad k = 0, \ldots, N$ ▷ Independent sampling at each time
  $\mathcal{L} \leftarrow \mathrm{PretrainLoss}(\mathcal{B}; \boldsymbol{\theta}, \boldsymbol{\phi}) + \lambda_c \mathrm{StateLoss}(\mathcal{B}; \boldsymbol{\theta}, \boldsymbol{\omega})$ ▷ VAE loss: Equation (53)
  $(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}) \leftarrow \mathrm{AdamW}((\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}), \nabla_{\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\omega}} \mathcal{L})$ ▷ Pretrain the VAE and state classifier
end for

Stage 2: Alternating dynamics estimation and representation learning

$\boldsymbol{\Theta} \leftarrow \mathbf{0}$; initialize $\boldsymbol{B}, \boldsymbol{C}, \boldsymbol{\Lambda}, \boldsymbol{\Sigma}_h$ and $\{\pi_r, \boldsymbol{F}_r, \boldsymbol{a}_r, \boldsymbol{S}_r, \boldsymbol{b}_h(r)\}_{r \in \mathscr{R}}$
$\boldsymbol{\eta} \leftarrow (\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\vartheta}, \boldsymbol{\omega})$ ▷ Load pretrained encoder, decoder, and classifier from Stage 1
for each training iteration **do**
  (a) Closed-form dynamics estimation ($\boldsymbol{\eta}$ fixed)
  $\boldsymbol{z}^{\mathrm{all}} \leftarrow \mathrm{sg}(\mathrm{Encode}(\mathcal{D}; \boldsymbol{\theta}))$ ▷ Reparameterized samples from all training cells
  $(\boldsymbol{q}^{\mathrm{all}}, \mathcal{C}^{\mathrm{all}}) \leftarrow \mathrm{Infer}(\mathcal{D}, \boldsymbol{z}^{\mathrm{all}}; \mathrm{sg}(\boldsymbol{\Theta}, \boldsymbol{\vartheta}))$
  ▷ $\boldsymbol{q}$: path probabilities (Equation (71)); $\mathcal{C}$: Gaussian moments and conditional coefficients (Equation (70))
  $(\boldsymbol{G}, \boldsymbol{y}^{\mathrm{M}}, \boldsymbol{M}) \leftarrow \mathrm{MemoryStats}(\boldsymbol{z}^{\mathrm{all}}, \boldsymbol{q}^{\mathrm{all}}, \mathcal{C}^{\mathrm{all}}, \mathrm{sg}(\boldsymbol{B}))$
  ▷ Compute $\boldsymbol{G}, \boldsymbol{y}^{\mathrm{M}}, \boldsymbol{M}$ by sample averaging and trapezoidal quadrature; Equation (80), Equation (58)
  $\boldsymbol{W} \leftarrow (\boldsymbol{M} + \epsilon_M \boldsymbol{I})^{-1}$; $\widehat{\boldsymbol{a}} \leftarrow \mathrm{PCG}(\boldsymbol{G}^\top \boldsymbol{W} \boldsymbol{G} + \lambda \boldsymbol{I}, \boldsymbol{G}^\top \boldsymbol{W} \boldsymbol{y}^{\mathrm{M}})$
  $\widehat{\boldsymbol{\Theta}} \leftarrow \mathrm{reshape}(\widehat{\boldsymbol{a}})$; $\boldsymbol{\Theta} \leftarrow \mathrm{sg}(\alpha \boldsymbol{\Theta} + (1 - \alpha)\widehat{\boldsymbol{\Theta}})$ ▷ EMA update
  (b) Representation and memory update ($\boldsymbol{\Theta}$ fixed)
  $\mathcal{B}_k \leftarrow \mathrm{Sample}(\rho_{t_k}, n_{\mathrm{b}}), \quad k = 0, \ldots, N$; $\boldsymbol{\epsilon}_{k,i} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_D)$
  $\boldsymbol{z}_{k,i} \leftarrow \boldsymbol{m}_{\boldsymbol{\theta}}(\boldsymbol{x}_{k,i}, t_k) + \boldsymbol{S}_{\boldsymbol{\theta}}(\boldsymbol{x}_{k,i}, t_k)^{1/2} \boldsymbol{\epsilon}_{k,i}$ ▷ Reparameterized posterior samples
  $(\boldsymbol{\Phi}_\Delta, \boldsymbol{J}_\Delta, \boldsymbol{Q}_\Delta) \leftarrow \mathrm{OUTransition}(\boldsymbol{\Theta}, \boldsymbol{\vartheta}, \Delta), \quad \Delta \in \mathcal{T} \cup \{t - s : (s, t) \in \mathcal{P}\}$
  $(\boldsymbol{q}, \mathcal{C}) \leftarrow \mathrm{Infer}(\mathcal{B}, \boldsymbol{z}; \boldsymbol{\Theta}, \boldsymbol{\vartheta})$ ▷ Path probabilities and Gaussian conditionals
  $\mathcal{L}_{\mathrm{snap}} \leftarrow \mathrm{SnapshotLoss}(\mathcal{B}, \boldsymbol{z}, \boldsymbol{q}, \mathcal{C}; \boldsymbol{\eta})$ ▷ Reconstruction, KL, state supervision; Equation (75)
  for each $(s, t) \in \mathcal{P}$ **in parallel do**
    $\boldsymbol{h}_{s,i,r} \sim \mathcal{N}(\bar{\boldsymbol{h}}_{s,r}(\boldsymbol{z}_{s,i}), \boldsymbol{V}_{s,r}^{h|z})$ for all i, r ▷ Infer memory at the source time s
    $\widehat{\boldsymbol{y}}_{s \to t,i,r} \leftarrow \boldsymbol{\Phi}_{t-s}(\boldsymbol{z}_{s,i}^\top, \boldsymbol{h}_{s,i,r}^\top)^\top + \boldsymbol{J}_{t-s} \boldsymbol{b}_r + \boldsymbol{\epsilon}_{i,r}, \boldsymbol{\epsilon}_{i,r} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}_{t-s})$
    $\widehat{\mu}_{s \to t} \leftarrow |\mathcal{B}_s|^{-1} \sum_{i,r} q_{s,i,r} \delta_{[\widehat{\boldsymbol{y}}_{s \to t,i,r}]_{\boldsymbol{z}}}$; $\widehat{\rho}_{s \to t} \leftarrow (\mathrm{Dec}_{\boldsymbol{\phi}})_{\#} \widehat{\mu}_{s \to t}$
  end for
  $\mathcal{L}_{\mathrm{pop}} \leftarrow \mathrm{PopulationLoss}(\mathcal{B}, \boldsymbol{z}, \{\widehat{\mu}_{s \to t}, \widehat{\rho}_{s \to t}\})$ ▷ Population matching; Equation (52)
  $\boldsymbol{Z}_{u_1:u_M}^{(r)} \leftarrow \mathrm{PathRollout}(\mathcal{B}_0, r; \boldsymbol{\theta}, \boldsymbol{\Theta}, \boldsymbol{\vartheta}), \quad r \in \mathscr{R}$
  $\bar{\boldsymbol{q}}_T \leftarrow \mathrm{sg}(|\mathcal{B}_N|^{-1} \sum_i \boldsymbol{q}_{N,i,\cdot})$; $\mathcal{L}_{\mathrm{path}} \leftarrow \mathrm{PathLoss}(\{\boldsymbol{Z}_{u_1:u_M}^{(r)}\}, \bar{\boldsymbol{q}}_T; \boldsymbol{\omega})$
  $(\boldsymbol{G}_{\mathcal{B}}, \boldsymbol{y}_{\mathcal{B}}^{\mathrm{M}}, \boldsymbol{M}_{\mathcal{B}}) \leftarrow \mathrm{MemoryStats}(\boldsymbol{z}, \mathrm{sg}(\boldsymbol{q}), \mathrm{sg}(\mathcal{C}), \mathrm{sg}(\boldsymbol{B}))$ ▷ Equation (80)
  $\boldsymbol{W}_{\mathcal{B}} \leftarrow (\boldsymbol{M}_{\mathcal{B}} + \epsilon_M \boldsymbol{I})^{-1}$
  $\mathcal{L}_{\mathrm{weak}} \leftarrow \|\boldsymbol{y}_{\mathcal{B}}^{\mathrm{M}} - \boldsymbol{G}_{\mathcal{B}} \mathrm{vec}(\boldsymbol{\Theta})\|_{\boldsymbol{W}_{\mathcal{B}}}^2$ ▷ Retain encoder gradients through $\boldsymbol{z}$
  $\mathcal{L}_{\mathrm{M}} \leftarrow \mathcal{L}_{\mathrm{snap}} + \mathcal{L}_{\mathrm{pop}} + \lambda_R \mathcal{L}_{\mathrm{path}} + \lambda_{\mathrm{weak}} \mathcal{L}_{\mathrm{weak}}$ ▷ Combined objective; Equation (82)
  $\boldsymbol{\eta} \leftarrow \mathrm{AdamW}(\boldsymbol{\eta}, \nabla_{\boldsymbol{\eta}} \mathcal{L}_{\mathrm{M}})$ ▷ Update VAE, memory dynamics, and classifier
end for
return $\boldsymbol{q}_{\boldsymbol{\theta}}, \mathrm{Dec}_{\boldsymbol{\phi}}, c_{\boldsymbol{\omega}}, \boldsymbol{\Theta}, \boldsymbol{\vartheta}$

We evaluate the path probabilities $q_t(r \mid \boldsymbol{z}, \boldsymbol{x})$ and conditional memory means $\bar{\boldsymbol{h}}_{t,r}(\boldsymbol{z})$ under the current model and hold them fixed together with the memory coupling matrix $\boldsymbol{B}$ when updating $\boldsymbol{\Theta}$.

Finally, we combine this closed-form update of $\boldsymbol{\Theta}$ with gradient-based learning of the representation and memory dynamics, following the alternating training scheme adopted in KoopCell (Algorithm 1). We first jointly pretrain the VAE and state classifier by minimizing $\mathcal{L}_{\mathrm{pre}} + \lambda_c \mathcal{L}_{\mathrm{state}}$, with $\mathcal{L}_{\mathrm{pre}}$ defined in Equation (53). In Stage 2, we alternate between the following two updates. First, we form the weak statistics, solve Equation (81), and update $\boldsymbol{\Theta} \leftarrow \alpha \boldsymbol{\Theta} + (1 - \alpha)\widehat{\boldsymbol{\Theta}}$. Second, with $\boldsymbol{\Theta}$ fixed, we update $\boldsymbol{\eta} = (\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\vartheta}, \boldsymbol{\omega})$ by minimizing

$$\mathcal{L}_{\mathrm{M}} = \mathcal{L}_{\mathrm{snap}} + \mathcal{L}_{\mathrm{pop}} + \lambda_R \mathcal{L}_{\mathrm{path}} + \lambda_{\mathrm{weak}} \mathcal{L}_{\mathrm{weak}}, \qquad \mathcal{L}_{\mathrm{weak}} = \|\boldsymbol{y}^{\mathrm{M}} - \boldsymbol{G}\boldsymbol{a}\|_{\boldsymbol{W}}^2. \tag{82}$$

We summarize the training procedure in Algorithm 2. The routine StateLoss evaluates the supervised classification loss $\mathcal{L}_{\mathrm{state}}$ in Equation (75). In Infer, we compute path probabilities $\boldsymbol{q}$ using Equation (71), and Gaussian moments and conditional coefficients $\mathcal{C}$ using Equation (68) and Equa-

tion (70). In MemoryStats, we form $(\boldsymbol{G}, \boldsymbol{y}^{\mathrm{M}}, \boldsymbol{M})$ from Equation (80) and Equation (58) using empirical averages and the trapezoidal rule. For the minibatch weak loss, we hold $(\boldsymbol{q}, \mathcal{C}, \boldsymbol{B})$ fixed while retaining gradients through $\boldsymbol{z}$. Rollouts and the path consistency loss follow Equation (68) and Equation (77), respectively. The notation sg and PCG follows Algorithm 1.

C Spectral Recovery Experiments

This section gives the data-generation and evaluation protocols for the two deterministic toy models in Section 4.1. Let Φ_t denote the flow of the corresponding dynamical system and let ρ_0 be its initial population distribution. For each toy model, the dataset consists of empirical population snapshots $\{\mathcal{X}_{t_k}\}_{k=0}^{K}$, where each $\mathcal{X}_{t_k} = \{\boldsymbol{x}_{t_k}^{(i)}\}_{i=1}^{N_k}$ contains independent samples from the time marginal $\rho_{t_k} = (\Phi_{t_k})_{\#}\rho_0$. To generate the empirical snapshot at time t_k, we independently draw a population of initial states from ρ_0, solve the governing ODE from time 0 to t_k, and retain only the terminal states. Repeating this construction independently at every observation time removes cross-time cell correspondence and yields the unpaired population snapshots used for training and evaluation.

C.1 Toy 1: finite-dimensional nonlinear flow

Dynamics and snapshot data generation. We consider the two-dimensional nonlinear flow

$$\dot{x}_1 = \mu x_1, \qquad \dot{x}_2 = \lambda(x_2 - x_1^2), \tag{83}$$

with $(\mu, \lambda) = (-0.2, -1)$. The origin $(0, 0)$ is globally attracting, and the invariant parabola $x_2 = \frac{\lambda}{\lambda - 2\mu} x_1^2 = \frac{5}{3} x_1^2$ is an attracting slow manifold. Indeed, the observables

$$\psi_1(\boldsymbol{x}) = x_1, \qquad \psi_2(\boldsymbol{x}) = x_2 - \frac{5}{3} x_1^2, \qquad \psi_3(\boldsymbol{x}) = x_1^2 \tag{84}$$

form a finite-dimensional Koopman-invariant subspace and satisfy $\dot{\psi}_j = \lambda_j \psi_j$ with $(\lambda_1, \lambda_2, \lambda_3) = (-0.2, -1, -0.4)$. The transverse coordinate ψ_2 therefore decays faster than the coordinates that parameterize the slow manifold.

We define ρ_0 by $x_1(0) \sim \frac{1}{2}\mathcal{N}(-1, 0.2^2) + \frac{1}{2}\mathcal{N}(1, 0.2^2)$ and $x_2(0) = x_1(0)^2 + \eta$, where $\eta \sim \mathcal{N}(0, 0.8^2)$. For each selected time, 3,000 new initial states are sampled from this law and advanced with the analytic solution of Equation (83). The nine training snapshots are observed at $t \in \{0, 0.1, 0.2, 0.3, 0.4, 0.55, 0.7, 0.9, 1.2\}$; independent target snapshots are generated every 0.2 time units from $t = 1.4$ to $t = 4.0$ for extrapolation. Figure 6 illustrates the generated snapshots, where we observe a rapid contraction toward the slow manifold.

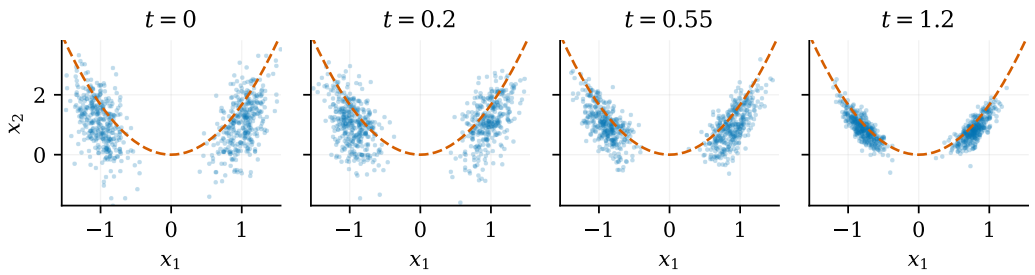


Figure 6: Toy 1: generated population snapshots. Each panel is an independently generated sample from the corresponding time marginal. The dashed curve denotes the attracting slow manifold.

Experimental settings. We consider three settings. The learned-observable and high-dimensional settings are briefly described in Section 4.1, whereas the known-observable setting is presented only in this appendix due to space constraints. The first is the *known-observable* setting. Here we

use the analytically specified observable map $\boldsymbol{z} = (x_1, x_2, x_1^2)^\top$, whose components span a finite-dimensional Koopman-invariant subspace in which the dynamics are exactly linear. We therefore fix $\boldsymbol{z}$ as the encoder and estimate only the linear generator, thereby isolating spectral recovery from representation learning. Second, in the *learned-observable* setting, the model receives only the two physical coordinates (x_1, x_2) and jointly learns a three-dimensional observable and its corresponding linear propagator. Third, in the *high-dimensional* setting, we generate 64-dimensional observations by applying a fixed injective map:

$$\boldsymbol{Y} = \mathrm{softplus}(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{c}) + \boldsymbol{\epsilon}_Y \in \mathbb{R}^{64}, \qquad \boldsymbol{\epsilon}_Y \sim \mathcal{N}(\boldsymbol{0}, 0.03^2 \boldsymbol{I}). \tag{85}$$

Here $\boldsymbol{x} = (x_1, x_2)^\top$ is the physical state and $\boldsymbol{W} \in \mathbb{R}^{64\times 2}$ has full column rank. Only $\boldsymbol{Y}$ is provided to either model; $\boldsymbol{x}$, $\boldsymbol{W}$, and $\boldsymbol{c}$ remain unknown. This setting evaluates representation learning and population prediction in the observed 64D space, and we do not report an inferred spectrum for it.

The learned-observable models use 2,048 multiscale random Fourier test functions and a three-dimensional latent space. KoopCell alternates the weak generator solve with representation updates for 500 epochs. scNODE receives the same snapshots and latent dimension, replacing the linear latent generator with an autonomous neural ODE. Table 4 reports the recovered spectrum only for the two settings in which the physical observable structure is available for an unambiguous comparison. The linear observable alignment used to compare learned and analytic eigenfunctions is applied only after training and does not enter either the objective or the rollout.

Table 4: Toy 1 spectral recovery with known and learned observables. Parentheses contain absolute eigenvalue errors.

$\lambda_\star$	Known observable	Learned observable
-0.2	-0.1983 (0.0017)	-0.1952 (0.0048)
-1.0	-0.9949 (0.0051)	-0.9774 (0.0226)
-0.4	-0.3893 (0.0107)	-0.4227 (0.0227)

Rollout and evaluation. Each rollout starts from the observed population at $t = 0$. KoopCell encodes this population, advances every latent state with the learned latent linear dynamics, and decodes at the requested time; scNODE instead integrates its learned latent neural ODE. At each horizon, we compute the sliced Wasserstein distance (SWD) between the predicted population and the independently generated target snapshot. We use physical-space SWD for the learned-observable setting and native observation-space SWD for the 64D setting, and average the metric over all extrapolation times. Figure 7 compares the population rollouts of KoopCell and scNODE. Predicted distributions from KoopCell agree more closely with the ground-truth distributions.

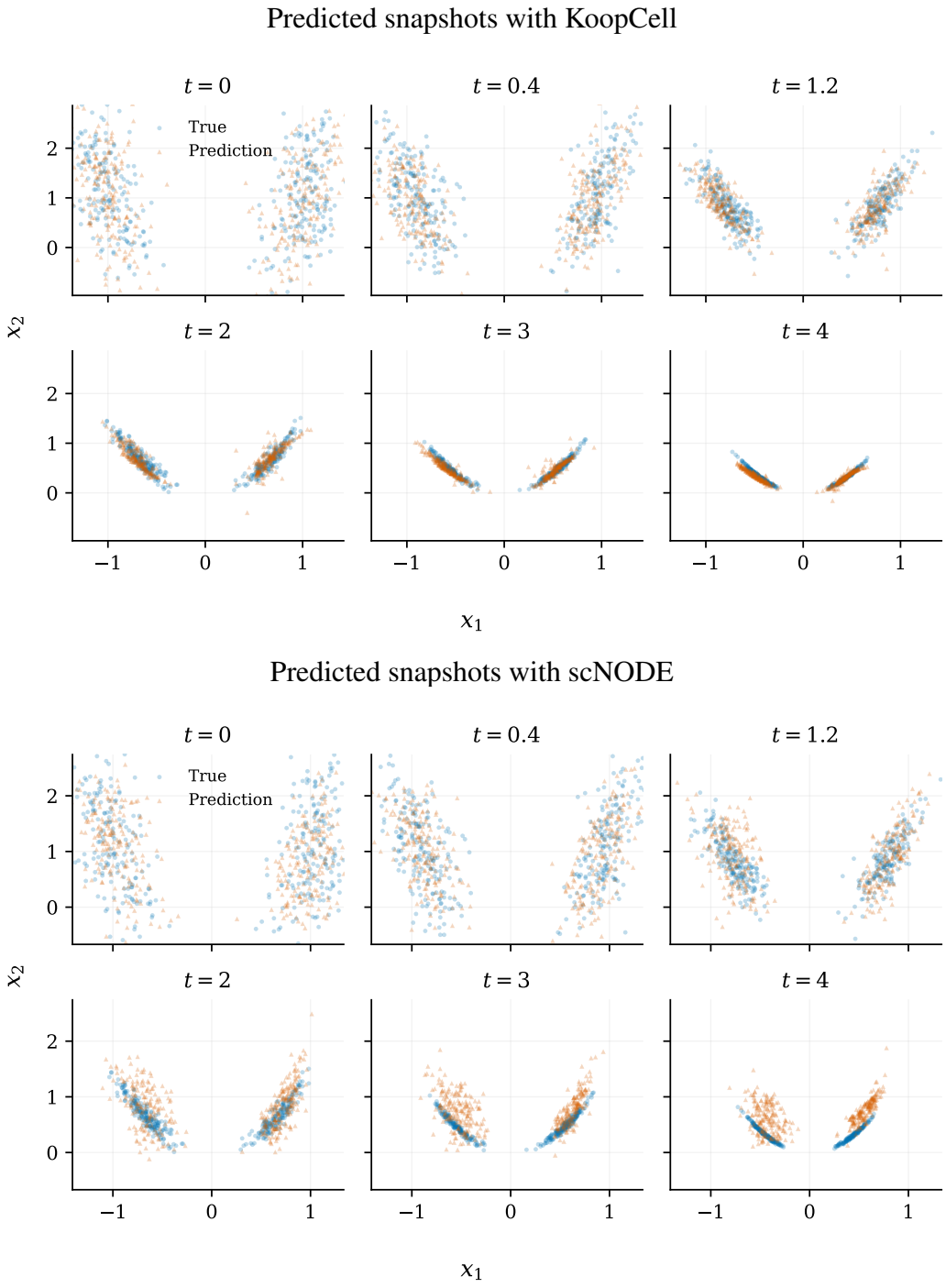


Figure 7: Predicted snapshots with KoopCell/scNODE. Both rollouts start from the observed Toy 1 population at $t = 0$. Blue circles denote independently generated target snapshots, and orange triangles denote predictions. Times after $t = 1.2$ are extrapolations.

C.2 Toy 2: damped Duffing flow

Dynamics and snapshot data generation. We consider the damped, unforced Duffing system studied by Otto & Rowley (2019),

$$\dot{q} = p, \qquad \dot{p} = -0.5p + q - q^3. \tag{86}$$

Its energy $H(q,p) = \frac{1}{2}p^2 - \frac{1}{2}q^2 + \frac{1}{4}q^4$ satisfies $\frac{\mathrm{d}}{\mathrm{d}t}H(q(t),p(t)) = -0.5p(t)^2 \le 0$, so the flow is dissipative. The origin is a saddle point, while $(q,p) = (\pm 1, 0)$ are stable spiral equilibria. The linearization at either stable equilibrium has eigenvalues $(-1 \pm \sqrt{31}i)/4$. The corresponding global Koopman eigenfunctions do not have closed-form expressions. As described by Otto & Rowley (2019), the magnitude and phase of the complex eigenfunction provide action–angle-like coordinates over each basin of attraction: the magnitude vanishes at the stable equilibrium, while the phase tracks the damped rotation. A further eigenfunction with eigenvalue zero is constant along trajectories and takes distinct values in the two basins of attraction.

For the initial population, we assign equal mass to two connected annular sectors, one in each basin. For a sign $s \in \{-1, +1\}$, we sample $\theta \sim \mathcal{U}[-2.20, 1.10]$ and $r \sim \mathcal{U}[0.16, 0.48]$, set $(q,p) = (s(1 + r\cos\theta), -1.35sr\sin\theta)$, and retain samples with $H(q,p) < -0.025$. This construction covers a range of intrabasin amplitudes and phases while keeping the population away from the energy barrier. Each time marginal contains 3,000 independently resampled initial states propagated with a high-accuracy numerical solver. Training uses the ten marginals at $t = 0, 0.5, \ldots, 4.5$; evaluation uses independently generated extrapolation targets at $t \in \{5, 6, 8, 10, 12\}$.

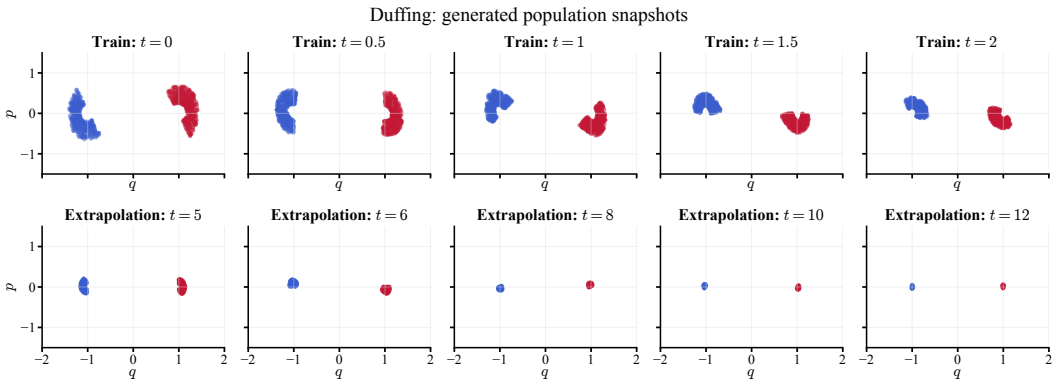


Figure 8: Toy 2 Duffing System: generated population snapshots. The first row shows training distribution marginals and the second row shows extrapolation ground truth. Each panel is generated independently from the same initial population law.

Experimental settings. We use two settings. In the *learned-observable* setting, KoopCell receives the physical snapshots (q,p) and jointly learns a three-dimensional observable, its linear generator, and a nonlinear decoder. Both the encoder and decoder are two-layer MLPs with hidden widths 64–64, and training uses 2,048 multiscale random Fourier test functions for 500 epochs. Because the relevant eigenfunctions are not available analytically, the objective of this experiment is to determine whether snapshot data recover their characteristic behavior: a complex mode encoding decaying intrabasin amplitude and phase, and a near-zero mode that remains constant within each basin and distinguishes the two long-term outcomes. Figure 1 visualizes these learned modes, and Table 5 compares their eigenvalues with the values implied by the Duffing dynamics.

In the *high-dimensional* setting, the same physical marginals are observed through a fixed injective map

$$\boldsymbol{Y} = \mathrm{softplus}(\boldsymbol{W}[q,p]^\top + \boldsymbol{c}) + \boldsymbol{\epsilon}_Y \in \mathbb{R}^{64}, \qquad \boldsymbol{\epsilon}_Y \sim \mathcal{N}(\boldsymbol{0}, 0.02^2\boldsymbol{I}). \tag{87}$$

Neither method receives (q,p) or the observation map during training. Both KoopCell and scNODE use a three-dimensional latent state and are evaluated by their population predictions in the native 64D observation space.

Table 5: Toy 2 spectral recovery in the learned-observable setting.

Mode	Reference λ	Estimated $\widehat{\lambda}$	Absolute error
Oscillatory pair	$-0.2500 \pm 1.3919i$	$-0.2979 \pm 1.4811i$	0.1012
Basin mode	0	0.0105	0.0105

Rollout and evaluation. We follow the same protocol as in the Toy 1 experiment. Figure 9 compares the population rollouts produced by KoopCell and scNODE. KoopCell more faithfully reproduces the target population over the full prediction horizon through $t = 12$.

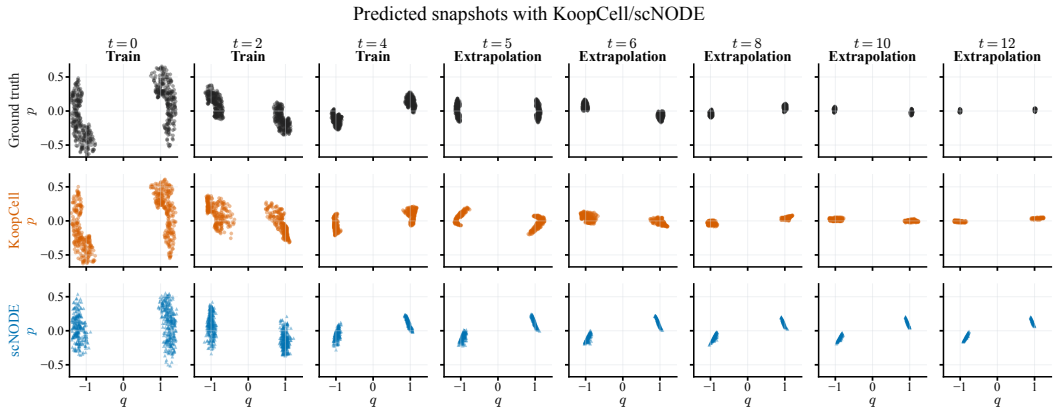


Figure 9: Predicted Duffing snapshots with KoopCell/scNODE. Rows show independently generated ground truth, KoopCell predictions, and scNODE predictions; columns progress from training times to extrapolation through $t = 12$.

C.3 Rollout-error accumulation

Figure 10 compares the rollout errors of KoopCell and scNODE over time for both toy models. The vertical dotted line marks the final training snapshot, and the shaded region denotes extrapolation. On Toy 1, scNODE's error increases steadily beyond the training window, whereas KoopCell's error remains low throughout extrapolation. On Toy 2, KoopCell also maintains lower rollout errors than scNODE throughout the evaluated horizon.

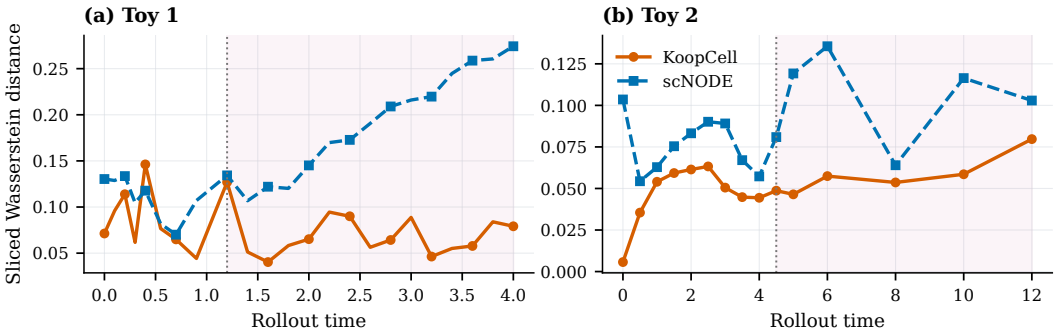


Figure 10: Accumulation of population rollout error. Left: Toy 1. Right: Toy 2. Each curve reports the per-time SWD for a rollout initialized from the observed population at $t = 0$.

D BRANCHING DYNAMICS EXPERIMENTS

This section gives the data-generation, training, and evaluation protocols for the branching experiment in Section 4.2.

Dynamics and snapshot data generation. We generate a noisy pitchfork process with progression coordinate s and branch coordinate b:

$$\mathrm{d}s = 0.21\,\mathrm{d}t, \qquad \mathrm{d}b = 1.5(2.4sb - b^3)\,\mathrm{d}t + 0.15\,\mathrm{d}W_t, \tag{88}$$

where $s_0 \sim \mathcal{N}(-1.05, 0.07^2)$ and $b_0 \sim \mathcal{N}(0, 0.04^2)$ are independent. As s becomes positive, the central equilibrium of the branch drift loses stability and two stable branches emerge. We simulate 4,000 trajectories with an Euler–Maruyama step of 0.0025, updating s before evaluating the branch drift. To simulate high-dimensional data from this two-dimensional process, we use the nonlinear observation map $\boldsymbol{x} = 2[\boldsymbol{Q}\boldsymbol{u} + 0.2\{\tanh(\boldsymbol{W}\boldsymbol{u} + \boldsymbol{c}) - \tanh(\boldsymbol{c})\}] + \boldsymbol{\epsilon} \in \mathbb{R}^{64}$, where $\boldsymbol{u} = (s/1.2, b/1.8)^\top$, $\boldsymbol{Q}^\top\boldsymbol{Q} = \boldsymbol{I}_2$, $\|\boldsymbol{W}\|_2 = 1$, $c_j \sim \mathcal{U}[-0.5, 0.5]$, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, 0.09^2\boldsymbol{I}_{64})$ is independent observation noise. Each snapshot is independently shuffled, and training uses only unpaired populations, without trajectory correspondences or branch labels. The nine training times are $0, 1.2, 2.4, 3.6, 4.8, 5.8, 6.8, 7.8, 9.2$; interpolation is evaluated at $4.2, 8.3$ and extrapolation at $9.5, 10$.

Models and training. At each training time, 3,000 cells are used for fitting, 500 for validation, and 500 are reserved for testing. We use the means and standard deviations estimated from fitting cells to standardize observations, and rescale time as $\tau = t/9.2$. Both models use a Gaussian encoder with a 10-dimensional latent state. The encoder and decoder each have two hidden layers of width 128, with SiLU and tanh activation functions, respectively. Decoder weights are spectrally normalized with output scale 4 to control the decoder's Lipschitz constant in both models and thus limit the amplification of latent differences during decoding. KoopCell uses the linear latent flow $\dot{\boldsymbol{Z}}_\tau = \boldsymbol{A}\boldsymbol{Z}_\tau + \boldsymbol{b}_z$ with a stable parameterization of $\boldsymbol{A}$. KoopCell-M uses the augmented dynamics in Equation (62) (Appendix B.2.1), with visible state $\boldsymbol{Z}_\tau \in \mathbb{R}^{10}$ and memory state $\boldsymbol{H}_\tau \in \mathbb{R}^{24}$. In this experiment, $\boldsymbol{\Sigma}_h = \boldsymbol{0}$ and the hidden bias $\boldsymbol{b}_h$ is shared across components. Following the conditional-mixture construction in Equation (61), we use two hidden components with $R \sim \mathrm{Cat}(\pi_1, \pi_2)$ and $\boldsymbol{H}_0 \mid \boldsymbol{Z}_0 = \boldsymbol{z}, R = r \sim \mathcal{N}(\boldsymbol{m}_r(\boldsymbol{z}), \boldsymbol{S}(\boldsymbol{z}))$. Here the mixture probabilities π_r are learned, and neural networks parameterize the conditional means $\boldsymbol{m}_r(\boldsymbol{z})$ and diagonal covariance $\boldsymbol{S}(\boldsymbol{z})$.

To isolate the effect of the dynamical model from that of the generator-estimation procedure, we use a common gradient-based training protocol for both models. For this controlled comparison, the closed-form generator updates in Algorithms 1 and 2 are replaced by joint optimization of the representation and dynamics, using the same reconstruction and population-matching objectives. We first pretrain a shared VAE for 2,400 Adam steps (learning rate 0.002, 256 cells per time), then initialize both models from this VAE and train each for 8,000 steps, with 128 cells per time and a cosine-decayed learning rate starting at 0.001.

Evaluation. After training, we encode the initial observed population, propagate its latent states with the learned dynamics, and decode the predicted populations at the evaluation times. We compare these predictions with the corresponding ground-truth populations using the following metrics:

- **Sliced Wasserstein distance (SWD).** The root mean square of one-dimensional W_2 distances along 512 shared random directions in standardized 64D observation space.
- **Energy distance.** The energy distance between predicted and observed populations in the same standardized space, normalized by $\sqrt{64} = 8$.
- **Squared maximum mean discrepancy (MMD^2).** We use an RBF kernel with bandwidth set to the median pairwise distance among fitting cells in standardized observation space.
- **Branch-coordinate W_1.** After branching, the two populations occupy positive and negative values of the branch coordinate b. We fit a linear probe $\widehat{b}(\boldsymbol{x}) = \boldsymbol{w}^\top\boldsymbol{x} + c$ to training observations and their known b values, then apply it to the predicted 64D observations. The W_1 distance between the estimated and true b distributions measures discrepancies in branch locations, widths, and relative population sizes.

- **Central mass error.** The interval $|b| < 0.5$ defines a fixed central region around the initial, unbranched population. Using the same linear probe, we compute the absolute difference between predicted and true fractions of cells in this region. This measures whether the model reproduces the amount of population remaining between the two branches.

Table 6 reports prediction errors averaged over the corresponding training, interpolation, and extrapolation times.

Table 6: Detailed branching prediction errors (mean $\pm$ standard deviation across three optimization seeds). Lower is better; the better mean is shown in bold.

Times	Model	SWD	Energy $\times 10^3$	MMD$^2 \times 10^3$	Branch W_1	Mass error
Training times	KoopCell	0.0882 ± 0.0047	2.281 ± 0.561	0.831 ± 0.225	0.068 ± 0.003	0.0264 ± 0.0007
	KoopCell-M	$\mathbf{0.0798 \pm 0.0004}$	$\mathbf{1.273 \pm 0.022}$	$\mathbf{0.410 \pm 0.011}$	$\mathbf{0.044 \pm 0.002}$	$\mathbf{0.0122 \pm 0.0017}$
Interpolation	KoopCell	0.0994 ± 0.0083	2.804 ± 0.960	1.058 ± 0.383	0.105 ± 0.004	0.0519 ± 0.0006
	KoopCell-M	$\mathbf{0.0848 \pm 0.0003}$	$\mathbf{1.258 \pm 0.055}$	$\mathbf{0.437 \pm 0.018}$	$\mathbf{0.067 \pm 0.003}$	$\mathbf{0.0201 \pm 0.0013}$
Extrapolation	KoopCell	0.2120 ± 0.0212	10.790 ± 2.713	4.359 ± 0.952	0.344 ± 0.010	0.0951 ± 0.0049
	KoopCell-M	$\mathbf{0.1400 \pm 0.0089}$	$\mathbf{2.393 \pm 0.568}$	$\mathbf{1.012 \pm 0.267}$	$\mathbf{0.106 \pm 0.006}$	$\mathbf{0.0016 \pm 0.0002}$

E Single-Cell Experimental Details

E.1 Single-cell datasets and preprocessing

We use the processed benchmark released with scNODE (Zhang et al., 2024) on Figshare[3], which contains three high-dimensional scRNA-seq datasets. ZB comprises 38,731 cells from 12 stages of zebrafish embryogenesis, spanning 3.3–12 hours post fertilization (Farrell et al., 2018). DR contains 27,386 cells from 11 overlapping collection windows spanning 0–20 hours of Drosophila embryogenesis (Calderon et al., 2022). SC follows the reprogramming of mouse embryonic fibroblasts toward induced pluripotent stem cells (Schiebinger et al., 2019); we use the reduced benchmark subset of 23,619 cells in 19 temporal groups, formed by pooling samples collected within each integer day from day 0 to day 18. Table 7 shows the number of cells at each time point in each dataset.

We consider three prediction tasks of increasing difficulty: *easy* withholds intermediate snapshots for interpolation; *medium* withholds the final snapshots for extrapolation; and *hard* combines intermediate and final held-out snapshots, leaving sparser observations for training. Figure 12 shows the training and held-out snapshots for each dataset and task.

For each task, we use the 2,000 highly variable genes (HVGs) selected in the benchmark. For ZB and DR, we normalize each cell's counts over these genes to a total of 10^4 and apply a log transformation: $x_{ig} = \log(1 + 10^4 c_{ig} / \sum_{h=1}^{2000} c_{ih})$, where c_{ig} is the count for gene g in cell i. For SC, we use the normalized expression values provided in the benchmark. We index the observed populations in chronological order as $t = 1, \ldots, T$.

Figure 11 visualizes all observed cells using the hard-task gene sets. For each dataset, we reduce the expression profiles to 50 principal components and then construct a two-dimensional UMAP embedding, with cells colored by snapshot index.

Table 7: Number of observed cells at each time point. Indices start at $t = 1$; dashes indicate indices beyond the observed horizon.

Dataset	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
ZB	311	200	1158	1467	5716	1026	4101	6178	5442	7114	1614	4404	–	–	–	–	–	–	–
DR	3944	2801	947	3139	2074	2147	3311	1692	1856	1491	3984	–	–	–	–	–	–	–	–
SC	800	560	1371	1413	1608	1377	1153	1156	2420	886	732	712	747	708	1444	2061	2106	1622	743

[3] https://doi.org/10.6084/m9.figshare.25601610.v1.

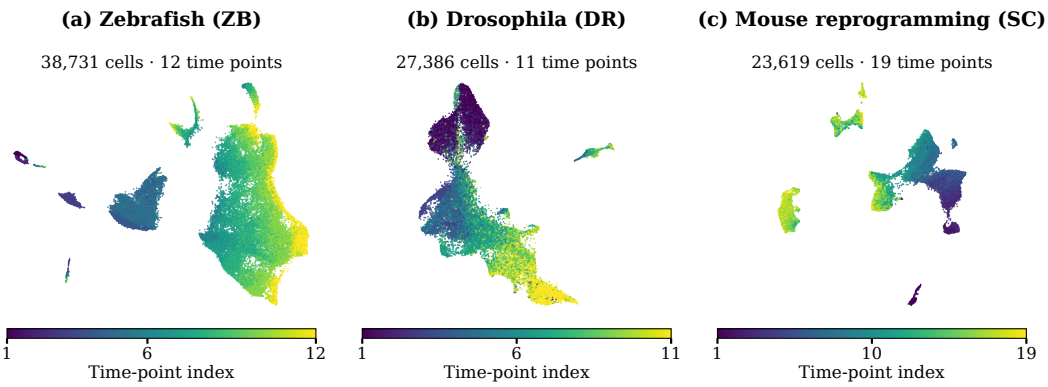


Figure 11: Observed populations in (a) ZB, (b) DR, and (c) SC. Each point represents a cell in a UMAP embedding of gene expression, colored by snapshot index.

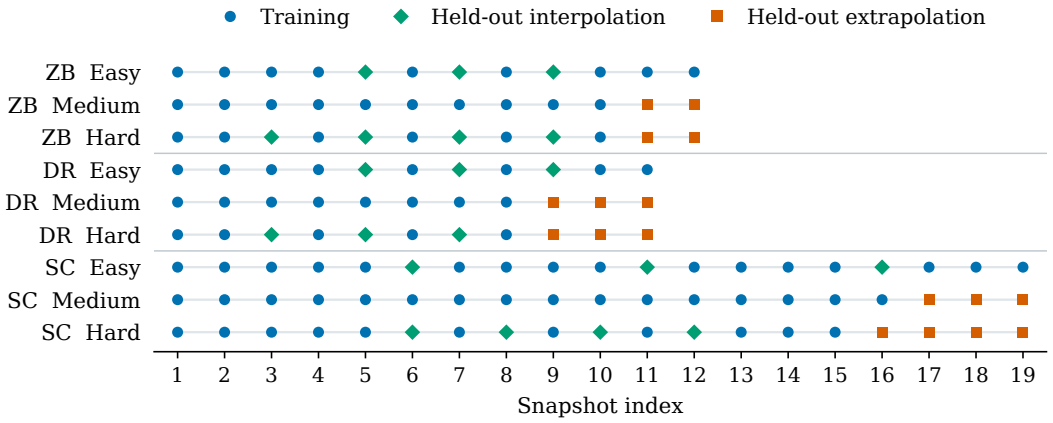


Figure 12: Training and held-out snapshots for each prediction task.

E.1.1 Population prediction and evaluation

For a consistent evaluation across methods, KoopCell, KoopCell-M, and all baselines each generate 2,000 cells at every held-out time point. We compare these predicted populations with all observed cells at the corresponding time point in the processed 2,000-gene space.

Every task retains the earliest observed population as the source for prediction. For KoopCell and KoopCell-M, we uniformly sample 2,000 cells from this population, with replacement when fewer than 2,000 cells are available, and draw their initial latent states from the encoder posterior. KoopCell propagates these states with its learned linear dynamics. KoopCell-M additionally samples a path R and memory state $\boldsymbol{H}_0$ for each initial latent state, then propagates the augmented state while keeping R fixed. We decode the visible latent states at each evaluation time to obtain the predicted gene expression distribution.

In the single-cell experiments, we evaluate predictions using the quadratic optimal transport cost

$$\frac{1}{2}W_2^2(\mu,\nu) = \inf_{\pi\in\Pi(\mu,\nu)} \int \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2 \,\mathrm{d}\pi(\boldsymbol{x},\boldsymbol{y}).$$

We report and summarize the results as mean $\pm$ sample standard deviation over three training seeds.

E.2 Implementation of Baseline Models

We compare KoopCell and KoopCell-M with six representative models for modeling temporal single-cell population dynamics. For scNODE and LGP-OT, the released repositories provide tuned configurations for the same datasets and prediction tasks, so we use those configurations directly. We select the hyperparameters of MIOFlow, PRESCIENT, PI-SDE and VGFM by training-only three-fold cross-validation, following the protocol adopted in the scNODE benchmark. Specifically, at every training time point, we randomly partition the cells into three approximately equal subsets, fit the model on two subsets, and validate on the remaining subset. We rotate the held-out subset over the three folds and select the configuration with the lowest mean validation optimal-transport distance across folds and training time points. The test time points defined by each prediction task remain inaccessible during model selection. After selecting a configuration, we retrain the model using all training cells and evaluate it on the held-out test time points.

scNODE. scNODE combines a variational autoencoder with a latent neural ordinary differential equation (ODE), and jointly optimizes the latent representation and continuous dynamics using population-level distribution matching and dynamic regularization (Zhang et al., 2024). We use the task-specific settings released as `tunedOurPars`: the latent dimension is fixed as 50, while the encoder, decoder, and ODE-network architectures depend on the dataset and prediction task. Because these settings were tuned on the same nine dataset–task combinations considered in our real world scRNA-seq experiments (section 4.3), we do not perform an additional search.

LGP-OT. LGP-OT represents population dynamics with a heteroscedastic Gaussian process in a learned latent space, approximates the process with Hilbert-space basis functions, and trains a decoder by matching generated and observed cell populations with a Sinkhorn optimal-transport objective (Balik & Lähdesmäki, 2026). We adopt the released task-specific `modelParams` configurations, which use a 32-dimensional latent space and select the decoder architecture and number of basis functions for each dataset and task. We therefore do not retune LGP-OT.

MIOFlow. MIOFlow first learns a geometry-preserving representation with a geodesic autoencoder and then fits neural-ODE dynamics using optimal-transport and density-regularization losses (Huguet et al., 2022). We select the geodesic-autoencoder dimension from $\{10, 50, 100\}$, the partition-encoder architecture from $\{[50, 100], [50, 100, 100]\}$, the ODE architecture from $\{[16], [16, 16], [16, 32, 16]\}$, and the density-loss coefficient from $\{1, 25, 50, 75, 100\}$ using three-fold cross-validation.

PRESCIENT. PRESCIENT learns a neural potential-energy landscape whose gradient defines the drift of a stochastic process, thereby generating cell-population trajectories in a fixed 50-dimensional PCA space (Yeo et al., 2021). We tune the Gaussian-noise standard deviation over $[0, 1]$, the energy-regularization strength over $[0, 0.1]$, the gradient-clipping threshold over $[0, 1]$, and the number of hidden layers over $\{1, 2, 3\}$ by three-fold cross-validation.

PI-SDE. PI-SDE learns a physics-informed stochastic differential equation in a fixed 50-dimensional PCA space by combining distribution matching with a potential-based action regularizer (Jiang & Wan, 2024). We fix the potential function to a two-layer fully connected network with 400 units per layer and Softplus activations. We tune only the diffusion parameterization among a fixed scalar (`const`), a learnable constant vector (`const_param`), and a state-dependent neural network (`Mlp`); the scalar initialization is fixed to 0.1. We also fix the learning rate to 0.005, the action-regularization coefficient to 0.5, and the sampled-cell fraction per training time point to 0.1, and select the diffusion type by three-fold cross-validation.

VGFM. VGFM jointly models cellular velocity and growth through flow matching and unbalanced optimal transport (Wang et al., 2025). We fit PCA using training cells only and explore raw and standardized PCA representations, dimensions in $\{50, 100\}$, Tanh and LeakyReLU activation functions, entropy regularization $\epsilon \in \{0.01, 0.03\}$, and source-marginal relaxation $\tau \in \{5, 10\}$. The reported results use standardized PCA with 50 components for the easy and hard tasks and 100 components for the medium task. All reported configurations use Tanh activations, $\epsilon = 0.01$, and $\tau = 10$. The velocity and growth networks have five and three hidden layers, respectively, with 512 units per

hidden layer. Training comprises 2,000 pretraining epochs followed by 50 trajectory-training epochs, with learning rates 10^{-3} and 10^{-4}, respectively.

KoopCell(-M), scNODE, LGP-OT, MIOFlow, PRESCIENT, PI-SDE and VGFM are all implemented and trained on NVIDIA Tesla V100 GPUs with 32 GB of memory.

E.3 Model configurations

Table 8: Settings shared across the three datasets and prediction tasks. Stage 1 and Stage 2 denote VAE pretraining and dynamics/representation alternating training stages, respectively.

Parameter	**KoopCell**	**KoopCell-M**
Visible latent dimension D	50	50
Encoder/decoder MLP activation	ReLU	ReLU
Training epochs (Stage 1 / Stage 2)	500 / 4,000	500 / 4,000
Batch size (Stage 1 / Stage 2)	256 / 128	256 / 128
Variational coefficient β	0.01	0.01
Number of Fourier test functions	2,048	2,048
Sinkhorn batch size	512	256
State-classifier hidden width	–	128
Path-loss coefficient λ_R	–	0.1

Table 9: Selected MLP architectures and loss weights. Encoder and decoder entries list hidden-layer widths; d_h is the memory dimension.

KoopCell							
Task	**Data**	**Encoder MLP**	**Decoder MLP**	d_h	λ_z	λ_x	λ_{weak}
Easy	ZB	256–256–256	256–256–256	–	1	1.5	1
	DR	512–256–256	256–256–512	–	1	5	1
	SC	512–256–256	256–256–512	–	1	3	1
Medium	ZB	256–256–256	256–256–256	–	1	1.5	1
	DR	256–256–256	256–256–256	–	1	5	1
	SC	256–256–256	256–256–256	–	1	3	1
Hard	ZB	256–256–256	256–256–256	–	1	1.5	1
	DR	256–256–256	256–256–256	–	1	5	1
	SC	256–256–256	256–256–256	–	1	3	1
KoopCell-M							
Task	**Data**	**Encoder MLP**	**Decoder MLP**	d_h	λ_z	λ_x	λ_{weak}
Easy	ZB	512–256–256	256–256–512	16	5	10	10^5
	DR	512–256–256	256–256–512	32	2	10	10^4
	SC	512–256–256	256–256–512	16	10	10	7,500
Medium	ZB	512–256–256	256–256–512	32	5	10	10^5
	DR	512–256–256	256–256–512	50	5	10	10^4
	SC	512–256–256	256–256–512	32	5	5	7,500
Hard	ZB	256–256–256	256–256–256	16	5	10	10^5
	DR	256–256–256	256–256–256	32	2	10	10^4
	SC	256–256–256	256–256–256	16	1	3	7,500

E.4 Additional Single-Cell Results

Table 10: Easy-task Wasserstein-2 distance (↓) at held-out time points (mean ± sample standard deviation over three seeds). The best and second-best means are shaded green (bold) and blue, respectively.

Method	**ZB/easy task**		
	Interpolation		
	$t = 5$	$t = 7$	$t = 9$
scNODE	435.03 ± 7.52	393.17 ± 3.84	473.80 ± 4.00
LGP-OT	436.27 ± 3.20	362.05 ± 2.20	432.68 ± 1.19
MIOFlow	447.71 ± 5.63	415.37 ± 5.93	491.57 ± 5.63
PRESCIENT	475.52 ± 8.25	447.97 ± 6.87	507.70 ± 2.65
PI-SDE	$\mathbf{422.71 \pm 1.60}$	369.90 ± 2.79	435.65 ± 3.68
VGFM	443.05 ± 5.89	391.59 ± 3.52	490.49 ± 4.32
KoopCell	431.53 ± 6.02	366.01 ± 1.99	436.71 ± 1.02
KoopCell-M	428.05 ± 3.79	$\mathbf{355.84 \pm 2.31}$	$\mathbf{426.25 \pm 1.34}$

Method	**DR/easy task**		
	Interpolation		
	$t = 5$	$t = 7$	$t = 9$
scNODE	348.05 ± 3.93	410.98 ± 3.28	499.14 ± 5.93
LGP-OT	346.08 ± 0.54	400.04 ± 0.63	471.29 ± 2.28
MIOFlow	345.69 ± 0.55	404.59 ± 1.80	497.52 ± 2.97
PRESCIENT	365.36 ± 0.65	428.44 ± 0.67	500.78 ± 0.58
PI-SDE	$\mathbf{335.03 \pm 1.02}$	392.45 ± 0.72	474.66 ± 1.54
VGFM	343.37 ± 3.20	406.06 ± 2.83	500.79 ± 1.07
KoopCell	339.74 ± 2.81	396.39 ± 6.49	464.00 ± 2.58
KoopCell-M	340.94 ± 0.34	$\mathbf{389.83 \pm 1.27}$	$\mathbf{462.52 \pm 1.06}$

Method	**SC/easy task**		
	Interpolation		
	$t = 6$	$t = 11$	$t = 16$
scNODE	57.42 ± 0.96	134.96 ± 5.05	118.18 ± 3.13
LGP-OT	51.16 ± 0.31	124.83 ± 0.71	100.55 ± 1.35
MIOFlow	55.96 ± 0.73	137.79 ± 0.78	122.78 ± 1.98
PRESCIENT	68.12 ± 0.67	144.99 ± 0.61	122.43 ± 1.48
PI-SDE	50.83 ± 0.18	128.23 ± 1.44	101.86 ± 1.09
VGFM	51.94 ± 2.60	152.69 ± 3.70	160.06 ± 1.12
KoopCell	52.40 ± 1.09	$\mathbf{124.38 \pm 1.09}$	96.33 ± 0.51
KoopCell-M	$\mathbf{50.52 \pm 0.70}$	124.65 ± 0.93	$\mathbf{95.54 \pm 0.70}$

Table 11: Medium-task Wasserstein-2 distance (↓) at held-out time points (mean ± sample standard deviation over three seeds). The best and second-best means are shaded green (bold) and blue, respectively.

Method	**ZB/medium task**	
	Extrapolation	
	$t = 11$	$t = 12$
scNODE	611.92 ± 5.22	692.19 ± 5.75
LGP-OT	570.90 ± 5.15	642.06 ± 9.06
MIOFlow	630.14 ± 13.26	755.77 ± 18.34
PRESCIENT	629.81 ± 3.64	677.10 ± 3.37
PI-SDE	572.99 ± 1.83	642.34 ± 4.18
VGFM	607.60 ± 4.59	665.91 ± 12.69
KoopCell	558.09 ± 2.51	605.61 ± 3.34
KoopCell-M	$\mathbf{548.66 \pm 2.12}$	$\mathbf{603.10 \pm 5.57}$

Method	**DR/medium task**		
	Extrapolation		
	$t = 9$	$t = 10$	$t = 11$
scNODE	561.97 ± 16.65	573.46 ± 13.14	715.56 ± 24.23
LGP-OT	514.14 ± 1.59	520.71 ± 15.78	661.35 ± 4.50
MIOFlow	558.93 ± 5.41	614.22 ± 21.61	803.22 ± 62.71
PRESCIENT	551.35 ± 0.75	511.82 ± 0.22	669.97 ± 1.78
PI-SDE	525.17 ± 1.02	543.32 ± 2.89	708.72 ± 2.27
VGFM	554.89 ± 5.43	670.18 ± 41.27	901.81 ± 72.49
KoopCell	505.50 ± 4.18	521.45 ± 12.38	644.53 ± 10.65
KoopCell-M	$\mathbf{495.08 \pm 1.65}$	$\mathbf{496.61 \pm 3.11}$	$\mathbf{643.43 \pm 5.02}$

Method	**SC/medium task**		
	Extrapolation		
	$t = 17$	$t = 18$	$t = 19$
scNODE	130.31 ± 3.95	124.75 ± 2.33	130.34 ± 2.47
LGP-OT	115.15 ± 1.66	122.09 ± 3.98	126.47 ± 3.44
MIOFlow	149.96 ± 14.49	154.17 ± 23.86	173.96 ± 39.70
PRESCIENT	145.16 ± 2.48	135.73 ± 2.14	136.07 ± 2.06
PI-SDE	122.47 ± 5.89	115.87 ± 3.50	126.16 ± 4.15
VGFM	164.07 ± 1.59	168.12 ± 1.03	174.05 ± 1.34
KoopCell	$\mathbf{110.59 \pm 0.65}$	111.27 ± 5.20	115.10 ± 5.88
KoopCell-M	114.75 ± 0.39	$\mathbf{109.09 \pm 2.17}$	$\mathbf{113.48 \pm 2.22}$

E.4.1 Predicted cell populations at held-out time points

Figures 13–15 compare the predicted populations from individual training runs. All methods share the saved PCA–UMAP projection for each dataset and task. We display up to 2,000 cells per time point, resampling VGFM particles according to their normalized growth weights.

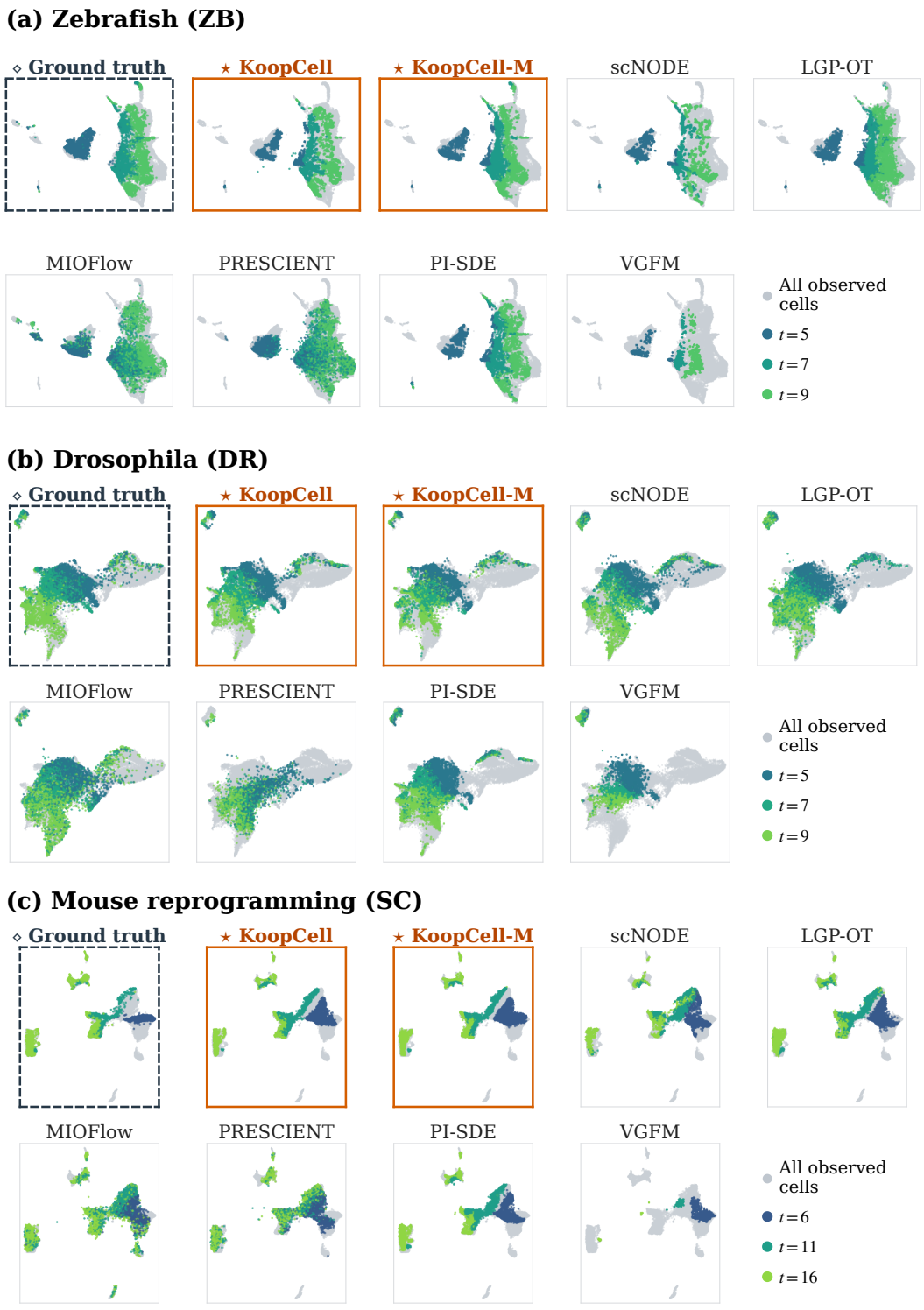


Figure 13: Easy-task predictions for ZB, DR, and SC. Colors indicate held-out time points; gray points show all observed cells. Diamonds and dark dashed frames identify ground truth; stars and orange frames mark KoopCell and KoopCell-M. All methods within each dataset share the same projection and axis ranges.

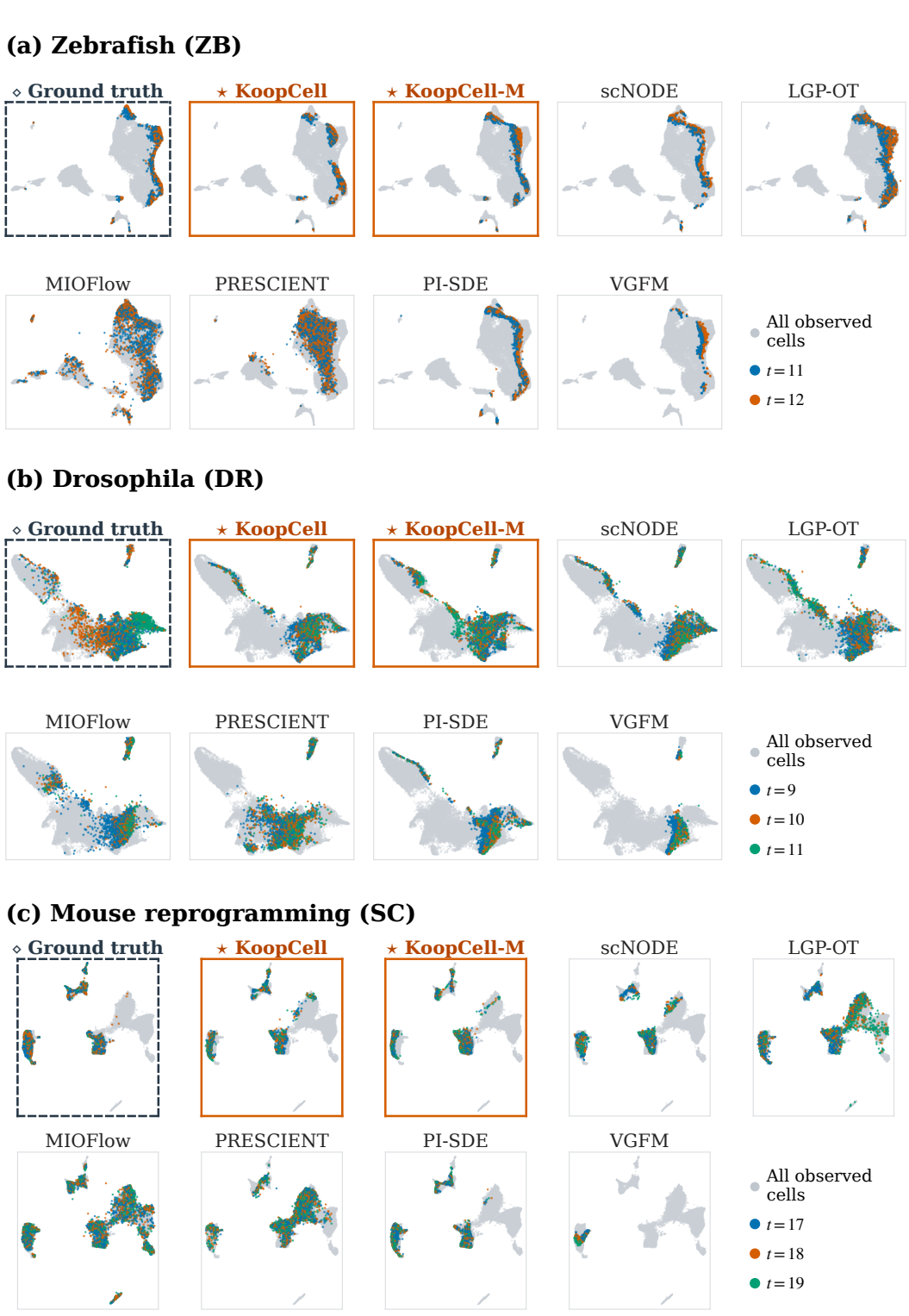


Figure 14: Medium-task predictions for ZB, DR, and SC. Colors indicate held-out time points; gray points show all observed cells. Diamonds and dark dashed frames identify ground truth; stars and orange frames mark KoopCell and KoopCell-M. All methods within each dataset share the same projection and axis ranges.

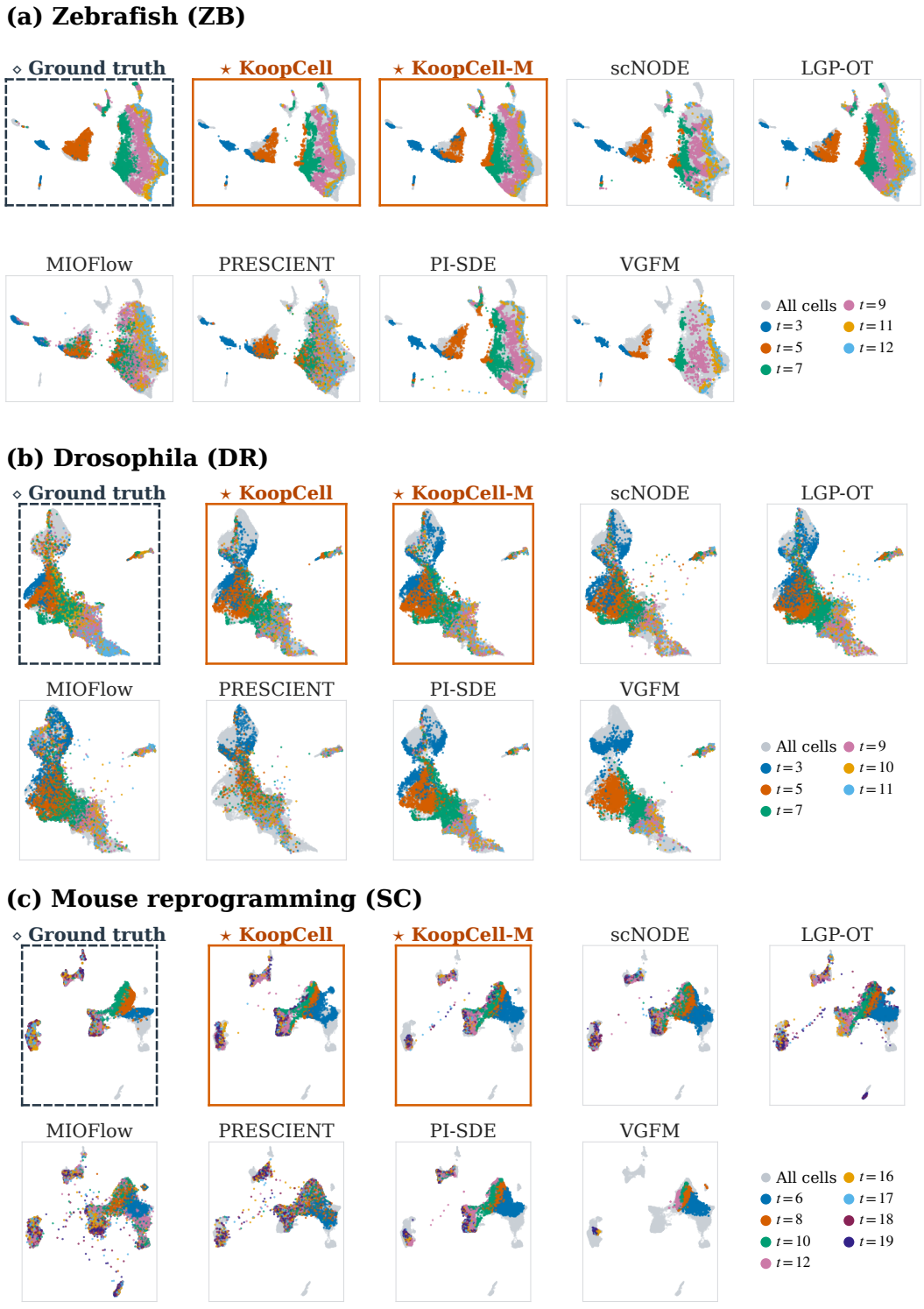


Figure 15: Hard-task predictions for ZB, DR, and SC, with all held-out interpolation and extrapolation time points overlaid in each panel. Colors indicate time points; gray points show all observed cells. Diamonds and dark dashed frames identify ground truth; stars and orange frames mark KoopCell and KoopCell-M. All methods within each dataset share the UMAP embedding used in Figure 11 and the same axis ranges.

E.4.2 Visualization of the learned latent representations

Figure 16 compares the representations of all 23,619 observed SC cells under the hard-task setting. Each method has its own UMAP projection; both rows use the same coordinates for that method. Reference states are obtained by matching cell identifiers to the published Waddington-OT cell sets and labeling the initial population as MEF. Unmatched or ambiguous cells remain gray; classifier predictions are not used to color these panels.

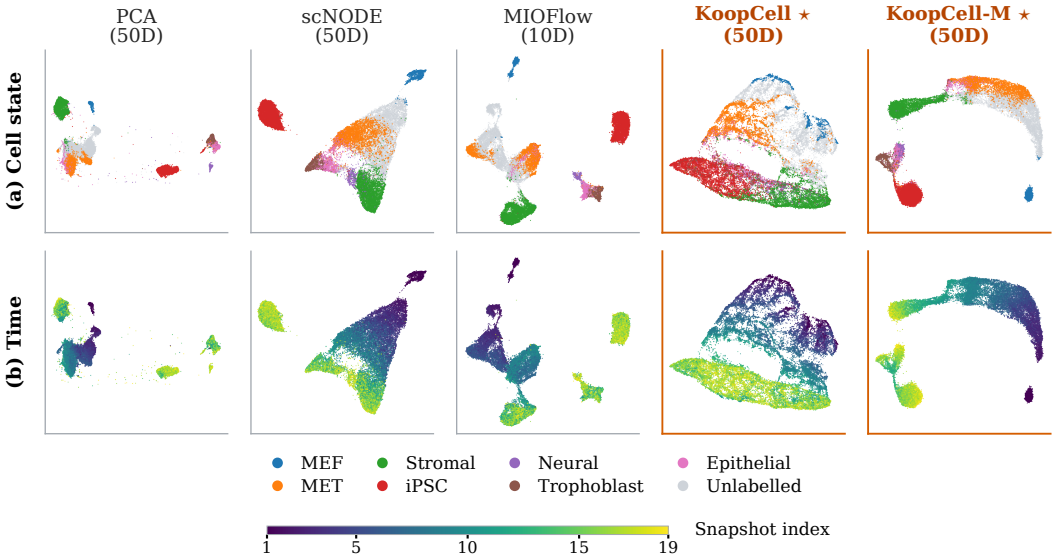


Figure 16: Latent representations of SC cells under the hard-task setting, colored by (a) reference cell state and (b) snapshot time index. All five methods show the same observed cells. Each method uses its own UMAP coordinates, shared between the two rows. The KoopCell and KoopCell-M columns are also shown in Figure 3.

E.5 Biological Downstream Analysis

We investigate how the population dynamics learned by KoopCell-M relate to lineage-associated genes and developmental expression programs. Using KoopCell-M trained on all ZB snapshots, we focus on Presomitic Mesoderm (PSM) and Hindbrain development, following Zhang et al. (2024).

Lineage-associated gene identification. We first estimate the probability that each cell contributes to a terminal cell type at 12 hpf. For a cell observed at time t_i, we encode $\boldsymbol{z}_i = \boldsymbol{m}_{\boldsymbol{\theta}}(\boldsymbol{x}_i, t_i)$ and infer its path probabilities and conditional memory distribution from the generative model. We then construct a directed graph between successive snapshots by propagating augmented states $(\boldsymbol{Z}, \boldsymbol{H})$ with the Gaussian transition in Equation (68) and assigning the propagated mass to neighboring states. Distances are scaled by the corresponding model covariance, and the path identity is retained across transitions. Reference annotations of all terminal cell types define the endpoints. Let $\boldsymbol{P}_{\ell,r}$ denote the transition matrix from snapshot ℓ to $\ell+1$ along path r, and let $\boldsymbol{F}_{\ell,r}(j, L)$ denote the probability of reaching terminal cell type L from augmented state j at snapshot ℓ, conditional on path r. The probability matrices satisfy

$$\boldsymbol{F}_{\ell,r} = \boldsymbol{P}_{\ell,r}\boldsymbol{F}_{\ell+1,r}, \qquad \boldsymbol{F}_{K,r}(j, L) = \mathbf{1}(c_j = L), \tag{89}$$

where K indexes the terminal snapshot and c_j is the reference label of terminal cell j. Averaging over the inferred memory and path distributions gives the cell-level fate probability f_{iL}.

Following the lineage-driver statistic of CellRank (Weiler et al., 2024), we rank genes by their Pearson correlation with the predicted probability of reaching lineage L:

$$r_{gL} = \mathrm{Corr}_{i:t_i<T}(x_{ig}, f_{iL}), \tag{90}$$

where x_{ig} is observed log-expression and T denotes the terminal time. To examine whether these fate-associated genes also distinguish the observed terminal populations, we compare log-normalized expression between 540 PSM and 511 Hindbrain cells at 12 hpf. Table 12 reports the mean expression difference Δ_g (PSM minus Hindbrain) for the four highest-ranked genes per lineage, selected by r_{gL}. Positive and negative values indicate higher expression in PSM and Hindbrain, respectively.

Recovery of developmental markers. *MSGN1* and *TBX16* rank first and fourth for PSM, with correlations of 0.665 and 0.500, respectively (Table 12). Both genes participate in zebrafish mesodermal morphogenesis and differentiation (Manning & Kimelman, 2015). For Hindbrain, *SOX19A* and *SOX3* rank first and second ($r = 0.568$ and 0.540), consistent with their expression in the developing neuroectoderm (Rossi et al., 2009). All eight genes show higher terminal expression in their corresponding lineage. *APELA* shows a modest terminal expression difference ($\Delta_g = -0.243$). In the UMAP projection, the observed expression patterns of these genes align with the annotated PSM and Hindbrain populations (Figure 17).

Table 12: Lineage-associated genes and terminal expression differences. The four highest-ranked genes are shown for each lineage. r_{gL} is the preterminal expression–fate correlation used for ranking; Δ_g is mean observed log-expression in PSM minus Hindbrain at 12 hpf.

Lineage	Gene	Rank	r_{gL}	Δ_g
PSM	*MSGN1*	1	0.665	4.388
PSM	*TBX6L*	2	0.559	1.434
PSM	*RBM38*	3	0.513	2.160
PSM	*TBX16*	4	0.500	3.844
Hindbrain	*SOX19A*	1	0.568	−3.670
Hindbrain	*SOX3*	2	0.540	−3.552
Hindbrain	*SI:CH211-152C2.3*	3	0.472	−3.770
Hindbrain	*APELA*	4	0.371	−0.243

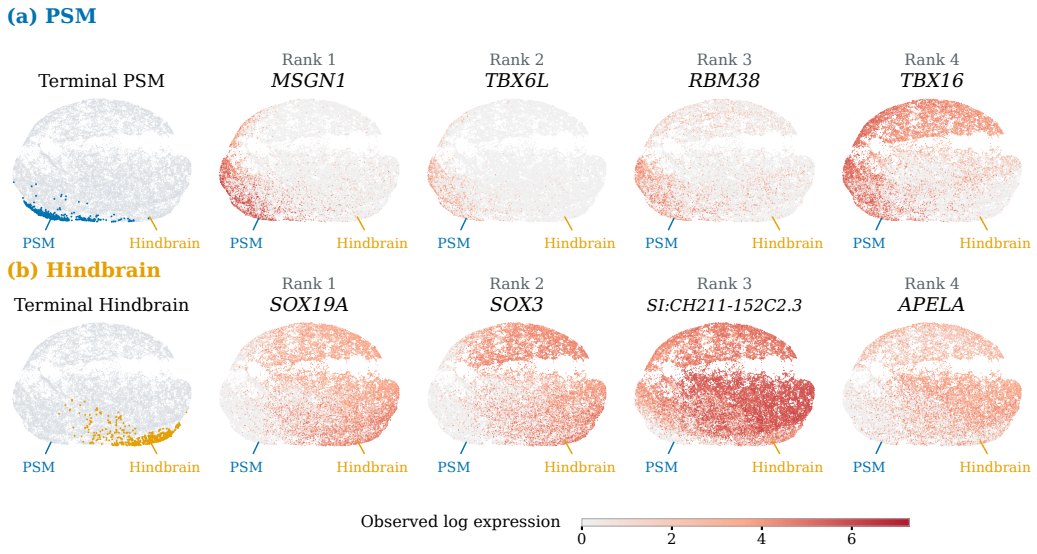


Figure 17: Observed expression of lineage-associated genes on a UMAP projection of the latent representations learned by KoopCell-M for ZB cells. Rows show the four highest-ranked genes for (a) PSM and (b) Hindbrain. The left panels highlight the corresponding annotated populations at 12 hpf; the remaining panels show observed log-expression across all cells using a common color scale. All panels share the same UMAP coordinates. Arrows indicate the terminal populations.

Dynamic gene-expression programs. We next identify groups of genes with similar predicted expression-rate profiles along developmental trajectories. Starting from cells observed at 8 hpf, we infer their path and memory distributions and propagate the augmented dynamics to 12 hpf. The decoder maps each trajectory to log-normalized gene-expression space. Because noise enters only the memory variables, the predicted expression rate for gene g follows the chain rule:

$$u_g(t) = \nabla_{\boldsymbol{z}} \operatorname{Dec}_{\boldsymbol{\phi},g}(\boldsymbol{Z}_t)^\top (\boldsymbol{A}\boldsymbol{Z}_t + \boldsymbol{B}\boldsymbol{H}_t + \boldsymbol{b}_z). \tag{91}$$

We compute this rate using Jacobian–vector products and average it separately for each lineage. We first give each source cell equal total weight, distributed across paths according to their inferred probabilities and equally among trajectory samples along each path. Let α_a denote the resulting weight of trajectory a, whose path is r_a. To determine its contribution to lineage L, we compare its endpoint $\boldsymbol{z}_T^a$ at $T = 12$ hpf with annotated terminal cells. We define $b_L(\boldsymbol{z}_T^a, r_a)$ as the distance-weighted fraction of neighboring cells labeled L, with distances scaled by the terminal covariance for path r_a. Normalizing the products $\alpha_a b_L(\boldsymbol{z}_T^a, r_a)$ gives the lineage-specific weights and mean expression rates:

$$w_{aL} = \frac{\alpha_a b_L(\boldsymbol{z}_T^a, r_a)}{\sum_{a'} \alpha_{a'} b_L(\boldsymbol{z}_T^{a'}, r_{a'})}, \qquad \bar{u}_{gL}(t) = \sum_a w_{aL} u_g^a(t). \tag{92}$$

For each lineage, we select the 100 genes with the largest mean expression rates over 8–12 hpf, emphasizing net upregulation over this interval. We standardize each rate profile across time and apply Ward clustering to obtain three modules, ordered by the temporal center of their above-average rates. To identify the biological functions associated with these programs, we test enrichment of Gene Ontology (GO) biological-process terms (Gene Ontology Consortium, 2026) in each lineage's top-100 genes and in each of its three modules. We map input gene symbols to Zebrafish Information Network (ZFIN) identifiers using official symbols and aliases (Ruzicka et al., 2019). The background comprises the 1,549 uniquely mapped input genes with GO biological-process annotations; each tested gene set is restricted to this same background. A one-sided hypergeometric test measures whether a term annotates more genes in a tested set than expected from its frequency in the background (Boyle et al., 2004). Benjamini–Hochberg correction (Benjamini & Hochberg, 1995) is applied jointly across the tested terms and gene sets in both lineages, with $q < 0.05$ as the significance threshold.

Developmental functions of the expression programs. The PSM modules are enriched for genes involved in muscle differentiation and development (Figure 18). The first module contains *MYF5*, *MYOD1*, *PRDM1A*, and *RBM24A*, and is enriched for positive regulation of muscle cell differentiation ($q = 6.91 \times 10^{-4}$). This is consistent with the role of *MYF5* and *MYOD1* in initiating zebrafish myogenesis (Osborn et al., 2020). The second includes *ACTA1A*, *DESMA*, and *DMRT2A*, with enrichment for skeletal muscle organ development ($q = 1.43 \times 10^{-2}$). In Hindbrain, the first module includes *FGF3*, *HER6*, and *NEUROG1*, with enrichment for regulation of neuron differentiation ($q = 8.57 \times 10^{-3}$). The second contains *SOX3*, *SOX21A*, *PAX6A*, and *GBX2*, and is enriched for nervous system development ($q = 2.47 \times 10^{-5}$) and brain development ($q = 7.68 \times 10^{-4}$). These enrichments are consistent with the neural progenitor and differentiation programs described in the developing zebrafish hindbrain (Tambalo et al., 2020). The third module in each lineage is enriched for intermediate filament organization and includes the keratin genes *KRT4* and *KRT5* (Cokus et al., 2019). Together, the recovered markers and lineage-associated functional enrichments support the biological interpretability of the learned population dynamics.

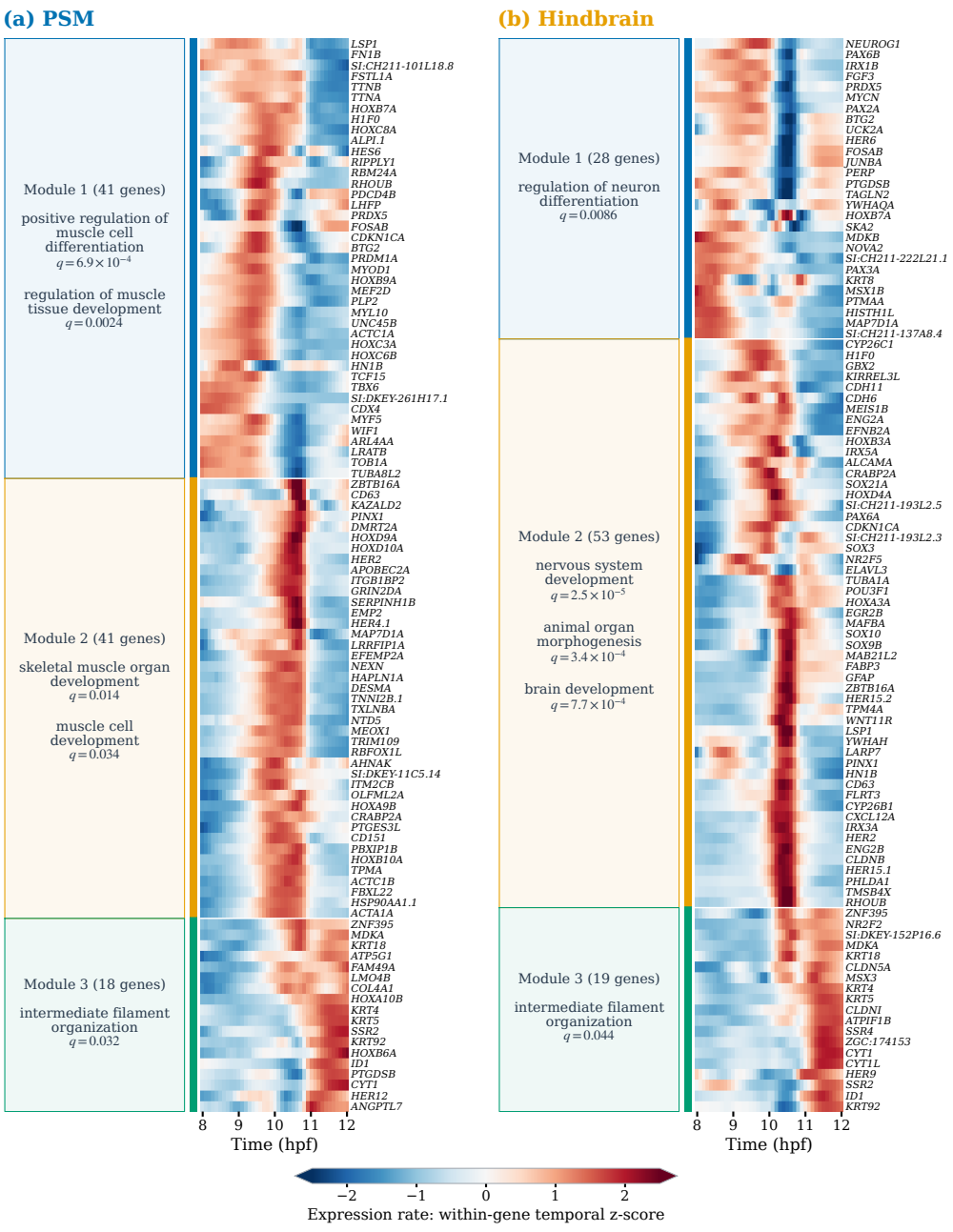


Figure 18: Dynamic gene-expression programs along KoopCell-M trajectories over 8–12 hpf. Each lineage contains the 100 genes with the largest mean predicted rates of change in log-expression, grouped into three Ward modules. Colors show within-gene temporal z-scores of expression rates, with red indicating rates above the gene's temporal mean and blue indicating rates below it. Boxes report representative GO biological-process terms and adjusted q-values. PSM programs are enriched for muscle development and Hindbrain programs for neural development; both include a module containing keratin genes.