Modeling Cellular State Transitions

An RNA Velocity–Guided Neural Stochastic Framework for Forward Simulation

Bob Zhao
The Webb Schools

Abstract

Single-cell RNA sequencing (scRNA-seq) represents cells as high-dimensional gene expression vectors and enables RNA velocity to estimate short-term state changes; however, existing methods primarily infer fate probabilities on observed cells states rather than simulate future trajectories. Here, I develop a stochastic framework for modeling cellular state transitions that enables forward simulation from a single snapshot. A neural network learns a continuous RNA velocity field in a reduced latent space, guiding stochastic transitions across both observed and arbitrarily proposed intermediate states. Transitions are proposed based on local similarity and velocity alignment and accepted via a Metropolis–Hastings–inspired criterion that enforces biologically plausible dynamics, with terminal states treated as absorbing. Evaluations on pancreas and hematopoiesis datasets show that simulated trajectories capture key directional trends and produce endpoint distributions aligned with observed future populations. This approach enables explicit and continuous prediction of cellular dynamics, with potential applications in disease prevention and forecasting disease progression with future research.

1 Background

1.1 Clinical Motivation and the Snapshot Problem

Many serious diseases, such as cancer, immune dysregulation, and degenerative disorders, occur through gradual and directional shifts in cellular gene expression. These transitions reflect changes that unfold over time, often long before overt clinical symptoms appear. Ideally, clinicians would continuously monitor these molecular trajectories to understand progression, predict outcomes, and guide intervention.

In practice, however, monitoring cellular evolution requires invasive tissue biopsies followed by single-cell RNA sequencing (scRNA-seq). Because biopsies are costly, risky, and cannot be performed at high frequency, most datasets consist of isolated molecular snapshots rather than continuous time-series measurements. As a result, the temporal dynamics of cellular progression remain largely unobserved.

This creates a fundamental challenge: biological processes are dynamic, yet measurements are sparse and static.

1.2 Unrealized Dynamical Potential in Single-Cell Data

Despite being collected at a single time point, scRNA-seq data contain rich structural information. Each cell can be represented as a high-dimensional gene expression profile, and transcriptionally similar cells occupy nearby regions in gene expression space. This geometric structure reflects developmental relationships, differentiation hierarchies, and disease-associated shifts.

However, most standard analyses treat these datasets as static point clouds. Clustering methods identify cell types, and trajectory inference algorithms estimate pseudotime orderings, but they do not explicitly model the underlying dynamical process that moves cells through state space.

The key question becomes:

Can a single snapshot contain information about where cells are going, not just where they are?

1.3 RNA Velocity: Local Directional Information

RNA velocity provides a partial answer to this question. Introduced by La Manno et al., RNA velocity leverages the relative abundance of unspliced and spliced mRNA transcripts to estimate the short-term rate of change of gene expression within individual cells.

For each observed cell, RNA velocity assigns a vector in gene expression space representing the predicted instantaneous direction of transcriptional change. Conceptually, this transforms each static point into a point with an associated directional arrow, providing local dynamical information.

In formal terms, if a cell's gene expression state is represented by a vector

$$x_i \in \mathbb{R}^G,$$

RNA velocity estimates a corresponding vector

$$v_i \in \mathbb{R}^G,$$

indicating the direction in which the cell is expected to move over short time scales.

Positive components of v_i indicate genes whose expression is increasing; negative components indicate decreasing expression.

1.4 Limitations of RNA Velocity in Current Frameworks

Although RNA velocity provides local directional estimates, it does not by itself define a global dynamical system. Velocity vectors are computed only at observed cell states, and therefore define directional information at discrete points rather than a continuous vector field.

Existing computational methods integrate RNA velocity into graph-based models of cellular transitions. These approaches estimate transition probabilities among observed cells and infer terminal fate biases. While biologically meaningful, such methods remain constrained to transitions within the empirical dataset and do not simulate explicit forward trajectories through unobserved regions of state space.

Thus, a conceptual gap remains between:

- local directional information (RNA velocity), and
- global forward simulation of cellular evolution.

Bridging this gap requires learning a continuous approximation of the transcriptional velocity field and defining a principled stochastic update mechanism that allows cells to evolve through latent space beyond the discrete observed states.

2 Model Concept

2.1 Cellular State as a High-Dimensional Vector

Single-cell RNA sequencing represents each cell as a vector of gene expression measurements. If G genes are measured, a cell can be written as

$$x \in \mathbb{R}^G,$$

where each coordinate corresponds to the normalized expression of a gene.

A dataset of N cells is therefore a finite sampling of points in high-dimensional gene expression space. Transcriptionally similar cells lie near one another, while distinct cell types occupy separated regions. Differentiation, development, and disease progression can be interpreted geometrically as movement through this state space.

From this viewpoint, cellular biology can be framed as a dynamical systems problem: cells evolve through gene expression space under underlying regulatory forces.

2.2 Latent Structure of the Transcriptomic Landscape

Although gene expression space is high-dimensional, biological variation is often structured along lower-dimensional axes. Principal component analysis (PCA) provides a latent representation that captures dominant sources of variation while reducing noise.

Let

$$s \in \mathbb{R}^d, \quad d \ll G,$$

denote the latent representation of a cell after projection into PCA space. In this framework, cellular evolution is modeled within this reduced latent space, which preserves large-scale transcriptional geometry while enabling tractable modeling.

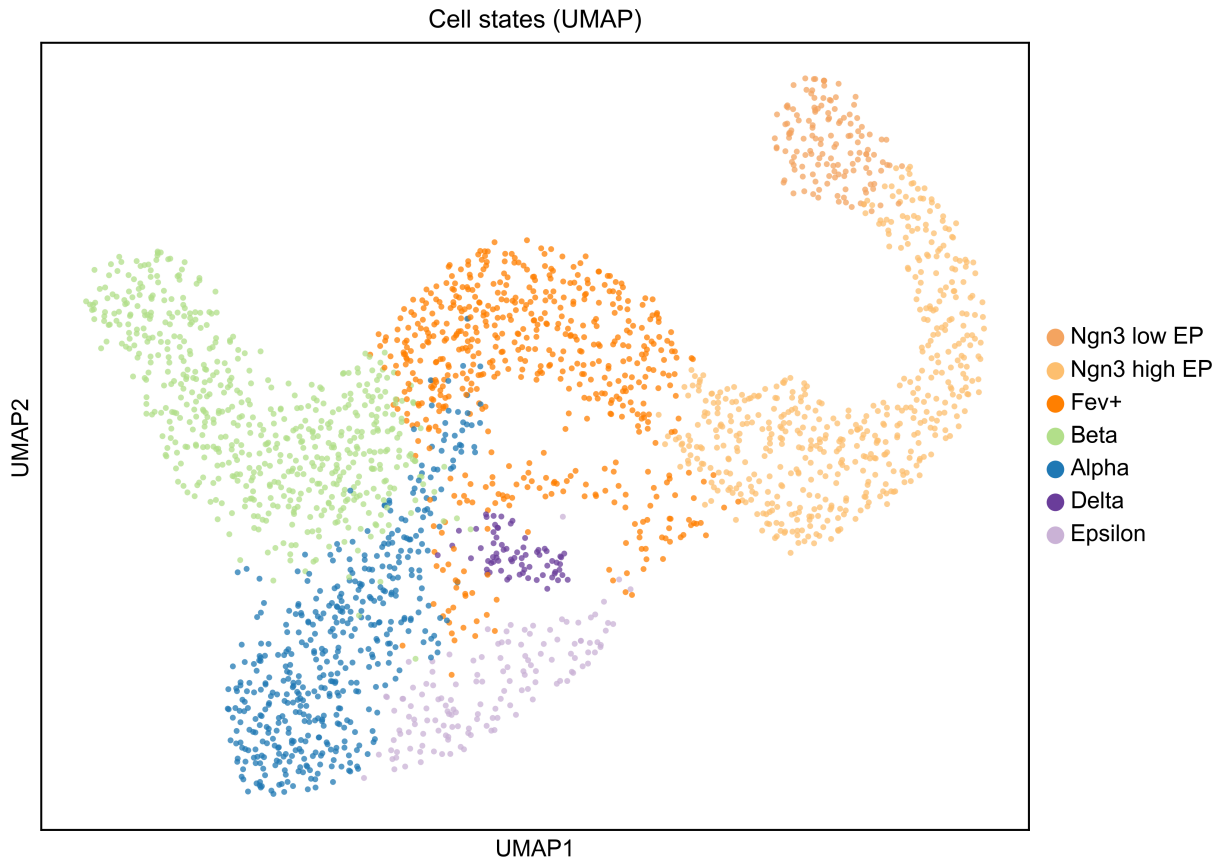

Figure 1: Pancreas Cell States in PCA Space

2.3 RNA Velocity as Directional Information

RNA velocity augments static cellular position with directional information. For each observed cell, RNA velocity estimates a vector indicating the short-term direction of transcriptional change.

Conceptually, this assigns to each observed point in state space an arrow describing local motion. The collection of these arrows provides a sampled representation of an underlying transcriptional flow.

However, RNA velocity is computed only at observed cell states. It therefore provides local directional information but does not by itself define a continuous dynamical system capable of generating trajectories through unobserved regions of state space.

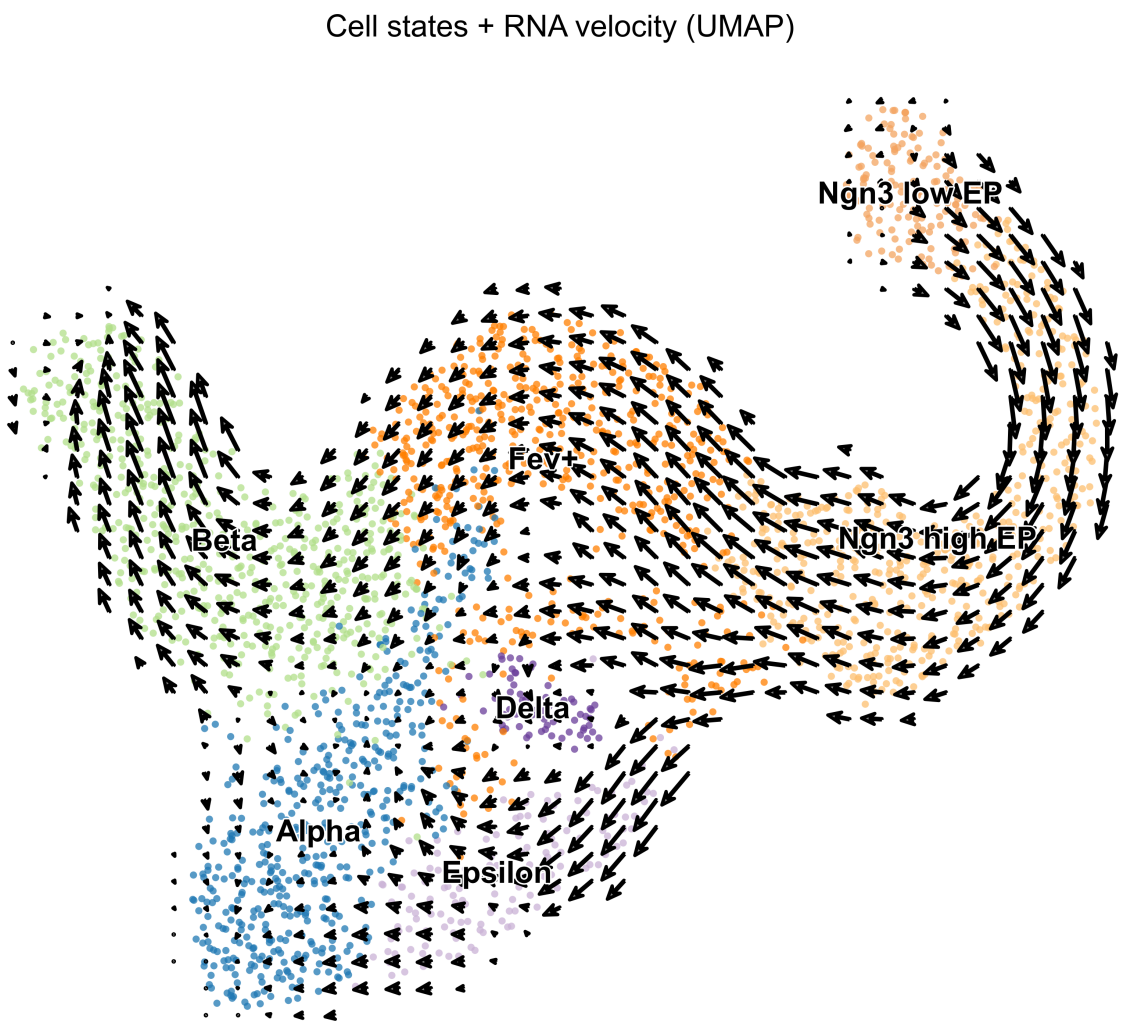

Figure 2: RNA Velocity Graph on Pancreas Cell States

2.4 From Local Arrows to Global Flow

The central idea of this work is to reinterpret RNA velocity not merely as a visualization tool, but as partial observations of a continuous vector field governing cellular evolution.

Rather than restricting analysis to transitions among observed cells, we treat latent space as a continuous landscape in which any point represents a possible transcriptional state. The observed RNA velocity vectors are then viewed as samples from an underlying dynamical field defined over this landscape.

By approximating this field and defining a stochastic update rule consistent with its directional structure, we can simulate explicit forward trajectories beginning from observed cells and extending into unobserved regions of latent space.

2.5 Population-Level Forward Simulation

Under this framework, cellular evolution is modeled as a stochastic process in latent space. Individual trajectories represent possible realizations of differentiation or progression, while ensembles of trajectories characterize the evolving distribution of cellular states.

Rather than predicting the exact fate of any single cell, the model aims to capture structured population-level movement over time. This shifts single-cell analysis from static fate inference toward explicit forward simulation of transcriptomic evolution.

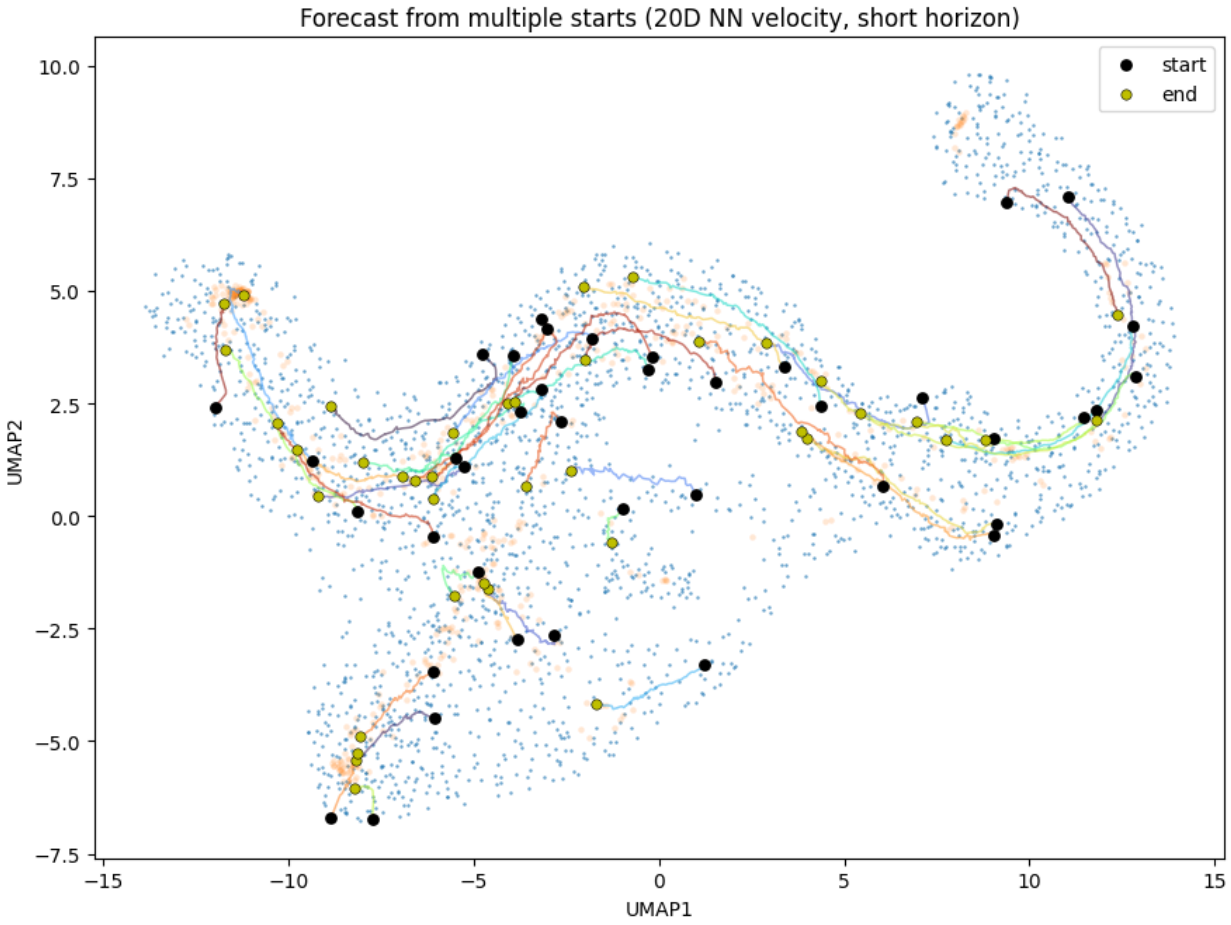

Figure 3: Simulated Pancreas Cell State Transitions

3 Methodology

3.1 Data

Two single-cell RNA sequencing datasets were used in this study.

Pancreas Dataset. A publicly available pancreas dataset provided within the scVelo framework was used during model development. This dataset contains preprocessed single-cell transcriptomic profiles with associated spliced and unspliced transcript counts, enabling RNA velocity estimation.

Bone Marrow Time-Course Dataset. To evaluate forward predictive performance on independent data, we used a human hematopoietic stem and progenitor cell (HSPC) time-course dataset from the Gene Expression Omnibus (GEO), accession GSE193517. This dataset includes cells sampled at multiple experimental time points, allowing direct comparison between simulated future states and empirically observed downstream populations.

The pancreas dataset was used for model training and internal validation, while the bone marrow dataset was reserved for independent forward-evolution evaluation.

3.2 Data Representation

All datasets were stored in AnnData (`.h5ad`) format and processed using the Scanpy and scVelo libraries in Python.

For each dataset:

- `adata.X` stores normalized gene expression values.
- `adata.layers["spliced"]` and `adata.layers["unspliced"]` store transcript counts required for RNA velocity estimation.
- `adata.obs` stores cell-level metadata, including time-point annotations when available.

Each cell is represented as a gene expression vector

$$x_i \in \mathbb{R}^G,$$

where G denotes the number of genes retained after filtering.

3.3 Preprocessing and Latent Representation

Single-cell RNA sequencing data consist of gene expression count matrices across N cells and G genes. Raw counts vary substantially across cells due to differences in sequencing depth, requiring normalization before analysis.

3.3.1 Normalization and Log Transformation

For each cell i, let c_{ig} denote the observed count for gene g. The total library size of the cell is

$$L_i = \sum_{g=1}^{G} c_{ig}.$$

Counts are normalized to a common depth L_0:

$$\tilde{c}_{ig} = \frac{L_0}{L_i} c_{ig}.$$

To stabilize variance and reduce skewness, log transformation is applied:

$$x_{ig} = \log(1 + \tilde{c}_{ig}).$$

The resulting matrix $X \in \mathbb{R}^{N \times G}$ represents normalized gene expression.

Highly variable genes are then selected to restrict analysis to biologically informative features, reducing dimensionality and noise.

3.3.2 Principal Component Analysis

Because G is large, modeling directly in gene space is computationally inefficient and prone to noise. Principal component analysis (PCA) is applied to obtain a lower-dimensional latent representation.

Let X denote the centered expression matrix. PCA finds orthonormal directions $w_1, \ldots, w_d$ that maximize projected variance. Equivalently, these are the top eigenvectors of the empirical covariance matrix:

$$\Sigma = \frac{1}{N} X^\top X.$$

Collecting these directions into a matrix

$$W \in \mathbb{R}^{G \times d},$$

the latent representation of cell i is

$$s_i = X_i W \in \mathbb{R}^d.$$

All subsequent modeling and simulation are conducted in this PCA latent space, which preserves dominant transcriptional structure while suppressing high-frequency noise.

3.4 RNA Velocity Estimation

RNA velocity estimates the short-term rate of change of gene expression using spliced (s) and unspliced (u) transcript counts.

For each gene, a simplified dynamical interpretation models transcription, splicing, and degradation processes. Under this framework, the instantaneous rate of change of spliced mRNA can be approximated as

$$v_g \approx \beta_g u_g - \gamma_g s_g,$$

where β_g and γ_g represent gene-specific splicing and degradation parameters estimated from the dataset.

For each cell i, collecting gene-level velocities produces a velocity vector

$$v_i \in \mathbb{R}^G.$$

These velocity vectors are projected into PCA space using the same loading matrix W:

$$\tilde{v}_i = v_i W \in \mathbb{R}^d.$$

Thus, each observed cell is represented in latent space by a pair

$$(s_i, \tilde{v}_i),$$

where s_i denotes position and $\tilde{v}_i$ denotes local directional change. These pairs form the foundation for learning a continuous velocity field and simulating forward cellular trajectories.

3.5 Learning a Continuous Latent-Space Velocity Field

RNA velocity provides directional estimates only at observed latent states

$$\{(s_i, \tilde{v}_i)\}_{i=1}^N,$$

where $s_i \in \mathbb{R}^d$ is the PCA representation of cell i, and $\tilde{v}_i \in \mathbb{R}^d$ is the RNA velocity vector projected into the same PCA space.

These vectors define local directional information at discrete sampled states. However, to simulate trajectories through unobserved regions of latent space, we require a continuous function that approximates the underlying transcriptional dynamics.

3.5.1 Regression Formulation

We model the latent velocity field as a function

$$f_\theta : \mathbb{R}^d \to \mathbb{R}^d,$$

parameterized by θ, such that

$$f_\theta(s_i) \approx \tilde{v}_i.$$

The problem reduces to supervised regression on paired samples:

$$\{(s_i, \tilde{v}_i)\}_{i=1}^N.$$

Once trained, f_θ defines a continuous approximation of the RNA velocity field and can be evaluated at any latent state $s \in \mathbb{R}^d$, including states not present in the original dataset.

3.5.2 Neural Network Architecture

The function f_θ is implemented as a fully connected multilayer perceptron (MLP) with two hidden layers.

The architecture is:

$$d \longrightarrow 64 \longrightarrow 64 \longrightarrow d.$$

Each hidden layer consists of 64 units with ReLU activation:

$$\phi(z) = \max(0, z).$$

Formally, the network computes:

$$f_\theta(s) = W_3\,\phi\Big(W_2\,\phi(W_1 s + b_1) + b_2\Big) + b_3,$$

where W_ℓ and b_ℓ denote learnable weights and biases.

The output dimension matches the latent dimension d, ensuring that the network predicts a velocity vector in PCA space for every input state.

3.5.3 Training Objective

Parameters θ are optimized to minimize the mean squared error between predicted and true projected velocities:

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \|f_\theta(s_i) - \tilde{v}_i\|_2^2.$$

Optimization is performed using the Adam algorithm with mini-batch stochastic gradient descent.

3.5.4 Model Performance

Model accuracy is evaluated by comparing predicted latent-space velocities $f_\theta(s_i)$ against scVelo-derived projected velocities $\tilde{v}_i$ on held-out validation data.

Performance is quantified using the coefficient of determination:

$$R^2 = 1 - \frac{\sum_i \|f_\theta(s_i) - \tilde{v}_i\|_2^2}{\sum_i \|\tilde{v}_i - \bar{v}\|_2^2},$$

where $\bar{v}$ denotes the mean velocity vector across validation samples.

The trained model achieves

$$R^2 \approx 0.905,$$

indicating strong agreement between predicted and observed RNA velocity directions and magnitudes in latent space.

3.5.5 Interpretation

After training, f_θ serves as a continuous approximation of the transcriptional vector field over PCA space. Unlike the original discrete velocity estimates, this learned field can be evaluated at arbitrary latent states, enabling forward simulation beyond observed cell positions.

This transformation from discrete arrows to a continuous neural velocity field is the core enabling step of the framework.

3.6 Velocity-Guided Stochastic Proposal Rule

Once a continuous latent-space velocity field

$$f_\theta : \mathbb{R}^d \to \mathbb{R}^d$$

has been learned, cellular evolution is modeled as a stochastic discrete-time process in PCA space.

Let $s_t \in \mathbb{R}^d$ denote the latent state of a cell at simulation step t.

3.6.1 Proposal Step

At each step, a candidate next state is generated by moving in the direction of the learned velocity field with added stochasticity:

$$s' = s_t + \eta f_\theta(s_t) + \sigma \varepsilon_t,$$

where:

- $\eta > 0$ is the step size controlling deterministic progression along the velocity direction,
- $\sigma \geq 0$ controls the magnitude of stochastic exploration,
- $\varepsilon_t \sim \mathcal{N}(0, I_d)$ is isotropic Gaussian noise.

The deterministic term $\eta f_\theta(s_t)$ advances the state along the learned transcriptional flow, while the stochastic term allows exploration of nearby regions in latent space.

3.6.2 Directional Alignment

Not all proposed transitions are biologically plausible. To evaluate consistency with the underlying velocity field, we compute the cosine alignment between the proposed displacement and the local velocity direction.

Define the proposed displacement:

$$\Delta s = s' - s_t.$$

Let the predicted velocity at the current state be:

$$v_t = f_\theta(s_t).$$

The cosine alignment is:

$$\cos(\theta_t) = \frac{\langle \Delta s, v_t \rangle}{\|\Delta s\|_2 \, \|v_t\|_2}.$$

This quantity satisfies:

$$\cos(\theta_t) = \begin{cases} +1 & \text{if perfectly aligned with velocity,} \\ 0 & \text{if orthogonal,} \\ -1 & \text{if directly opposed.} \end{cases}$$

3.6.3 Acceptance Probability

To probabilistically regulate transitions, cosine alignment is transformed into an acceptance probability using a logistic function:

$$\alpha(s_t \to s') = \frac{1}{1 + \exp(-\kappa \cos(\theta_t))},$$

where $\kappa > 0$ controls sensitivity to directional alignment.

Proposals strongly aligned with the velocity field have high acceptance probability, while transitions opposing the velocity direction are suppressed.

3.6.4 Acceptance–Rejection Update

The candidate state s' is accepted with probability $\alpha(s_t \to s')$. If accepted,

$$s_{t+1} = s',$$

otherwise,

$$s_{t+1} = s_t.$$

This defines a stochastic Markov process in latent space guided by the learned velocity field.

3.7 Acceptance Mechanism

The stochastic proposal rule generates candidate transitions that combine deterministic motion along the learned velocity field with isotropic noise. However, not all such proposals are biologically plausible. In particular, noise may induce transitions that oppose the local transcriptional direction.

To regulate these transitions, we introduce a probabilistic acceptance criterion based on directional alignment.

3.7.1 Directional Consistency Measure

Let the current latent state be $s_t \in \mathbb{R}^d$ and the proposed state be s'. Define the displacement vector:

$$\Delta s = s' - s_t.$$

Let the predicted local velocity be:

$$v_t = f_\theta(s_t).$$

We measure directional consistency using cosine similarity:

$$\gamma_t = \frac{\langle \Delta s, v_t \rangle}{\|\Delta s\|_2 \, \|v_t\|_2}.$$

The quantity $\gamma_t \in [-1, 1]$ encodes alignment:

- $\gamma_t \approx 1$: displacement follows velocity.
- $\gamma_t \approx 0$: orthogonal movement.
- $\gamma_t \approx -1$: displacement opposes velocity.

3.7.2 Logistic Acceptance Rule

Rather than enforcing strict rejection of misaligned steps, we convert alignment into a smooth acceptance probability:

$$\alpha(s_t \to s') = \frac{1}{1 + \exp(-\kappa \gamma_t)},$$

where $\kappa > 0$ controls sensitivity to alignment.
This mapping ensures:

$$\alpha \to 1 \quad \text{as } \gamma_t \to 1,$$
$$\alpha \to 0 \quad \text{as } \gamma_t \to -1.$$

Thus, transitions strongly aligned with the velocity field are almost always accepted, while strongly opposing transitions are suppressed but not deterministically forbidden.

3.7.3 Update Rule

Given a proposal s', we sample

$$u \sim \text{Uniform}(0, 1).$$

The update is:

$$s_{t+1} = \begin{cases} s', & \text{if } u < \alpha(s_t \to s'), \\ s_t, & \text{otherwise.} \end{cases}$$

This defines a Markov process in latent space in which stochastic exploration is constrained by learned transcriptional directionality.

3.8 Trajectory Simulation and Population Forecasting

The proposal and acceptance rules define a stochastic Markov process in latent space. Starting from an initial cell state $s_0 \in \mathbb{R}^d$, a simulated trajectory is generated by iteratively applying the update rule

$$s_{t+1} = \begin{cases} s', & \text{with probability } \alpha(s_t \to s'), \\ s_t, & \text{otherwise,} \end{cases}$$

where s' is proposed using the velocity-guided rule and α is the alignment-based acceptance probability.

3.8.1 Initialization

To generate biologically meaningful trajectories, simulations are initialized from selected starting cells. In particular, progenitor-like or early-state cells are natural starting points for forward differentiation modeling. For population-level forecasting, large ensembles of trajectories are initialized from many cells sampled from the initial population distribution.

3.8.2 Stopping Criteria and Terminal States

Trajectories are terminated using practical stopping criteria that reflect either convergence of the dynamics or computational limits. In this framework, a trajectory stops when any of the following occurs:

1. **Velocity magnitude becomes negligible:**

$$\|f_\theta(s_t)\|_2 < \varepsilon,$$

   indicating that the local predicted dynamics are near-stationary.

2. **A maximum step count is reached:** $t = T_{\max}$.

3. **An absorbing terminal region is reached:** once a state is classified as terminal, it is treated as absorbing so that subsequent updates do not move the state.

This design allows endpoint distributions to be collected across many runs and interpreted as predicted future population occupancy in latent space.

3.8.3 Large-Scale Simulation

To estimate population-level behavior, many trajectories are simulated in parallel. Given an initial set of latent states $\{s_0^{(j)}\}_{j=1}^M$, each trajectory produces a terminal state $s_T^{(j)}$. The collection of endpoints

$$\{s_T^{(j)}\}_{j=1}^M$$

defines an empirical forecast distribution over latent space.

In experiments, simulations were capped at a finite number of steps due to hardware constraints, but were still sufficient to produce structured, non-uniform endpoint distributions rather than isotropic drift.

3.9 UMAP Visualization of States and Simulated Endpoints

While modeling and simulation are performed in PCA latent space, results are visualized in two dimensions using Uniform Manifold Approximation and Projection (UMAP). UMAP is used strictly as a visualization tool and does not affect the simulation dynamics.

3.9.1 Fitting UMAP on Observed Cells

Let $\{s_i\}_{i=1}^N \subset \mathbb{R}^d$ be the observed PCA latent states. A UMAP embedding is fit using these observed states:

$$\Phi : \mathbb{R}^d \to \mathbb{R}^2.$$

This produces embedded coordinates

$$u_i = \Phi(s_i) \in \mathbb{R}^2.$$

3.9.2 Embedding Simulated Trajectories

Given simulated latent states $\{s_t\}$ or terminal endpoints $\{s_T^{(j)}\}$, the same learned embedding map is used to project them into two dimensions:

$$u_t = \Phi(s_t), \qquad u_T^{(j)} = \Phi(s_T^{(j)}).$$

This enables direct comparison between simulated states and the geometry of the observed cell manifold in a shared 2D visualization.

3.9.3 Endpoint Density Visualization

To summarize population-level forecasts, simulated terminal points are visualized as a density field in UMAP space. If $\{u_T^{(j)}\}_{j=1}^M$ denotes embedded endpoints, a smoothed density estimate $\hat{\rho}(u)$ can be constructed, and plotted as a heatmap where intensity corresponds to predicted future occupancy.

This produces a direct visual representation of the model's forecast: regions of high density correspond to latent states that the dynamics predict the population will occupy at a future horizon.

Suggested figures.

- *Single-trajectory visualization:* one simulated trajectory overlaid on the observed UMAP embedding.

- *Ensemble visualization:* multiple trajectories from random starting states showing directional structure.
- *Endpoint density map:* heatmap of simulated endpoints in UMAP space, interpreted as predicted future population distribution.

4 Results and Validation

4.1 Velocity Prediction Accuracy

Before performing forward simulations, we evaluate whether the neural network accurately reconstructs RNA velocity in PCA latent space. Because the learned velocity field f_θ is the foundation of all subsequent trajectory modeling, its fidelity must be established independently.

4.1.1 Evaluation Setup

The dataset was partitioned into training and held-out validation subsets. The network was trained to minimize mean squared error between predicted and scVelo-derived projected velocities:

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \| f_\theta(s_i) - \tilde{v}_i \|_2^2.$$

Evaluation was performed on validation samples not used during training.

4.1.2 Coefficient of Determination

Prediction accuracy is quantified using the coefficient of determination:

$$R^2 = 1 - \frac{\sum_i \| f_\theta(s_i) - \tilde{v}_i \|_2^2}{\sum_i \| \tilde{v}_i - \bar{v} \|_2^2},$$

where $\bar{v}$ denotes the mean projected velocity vector over validation samples.

An R^2 value near 1 indicates strong agreement between predicted and true velocity magnitudes and directions.

4.1.3 Results

The trained model achieved

$$R^2 \approx 0.905$$

on held-out validation data.

Figure 4 shows predicted versus true latent-space velocities. Each point corresponds to a velocity component across validation samples. The strong linear correlation demonstrates that the neural network accurately reconstructs both magnitude and directional structure of RNA velocity in PCA space.

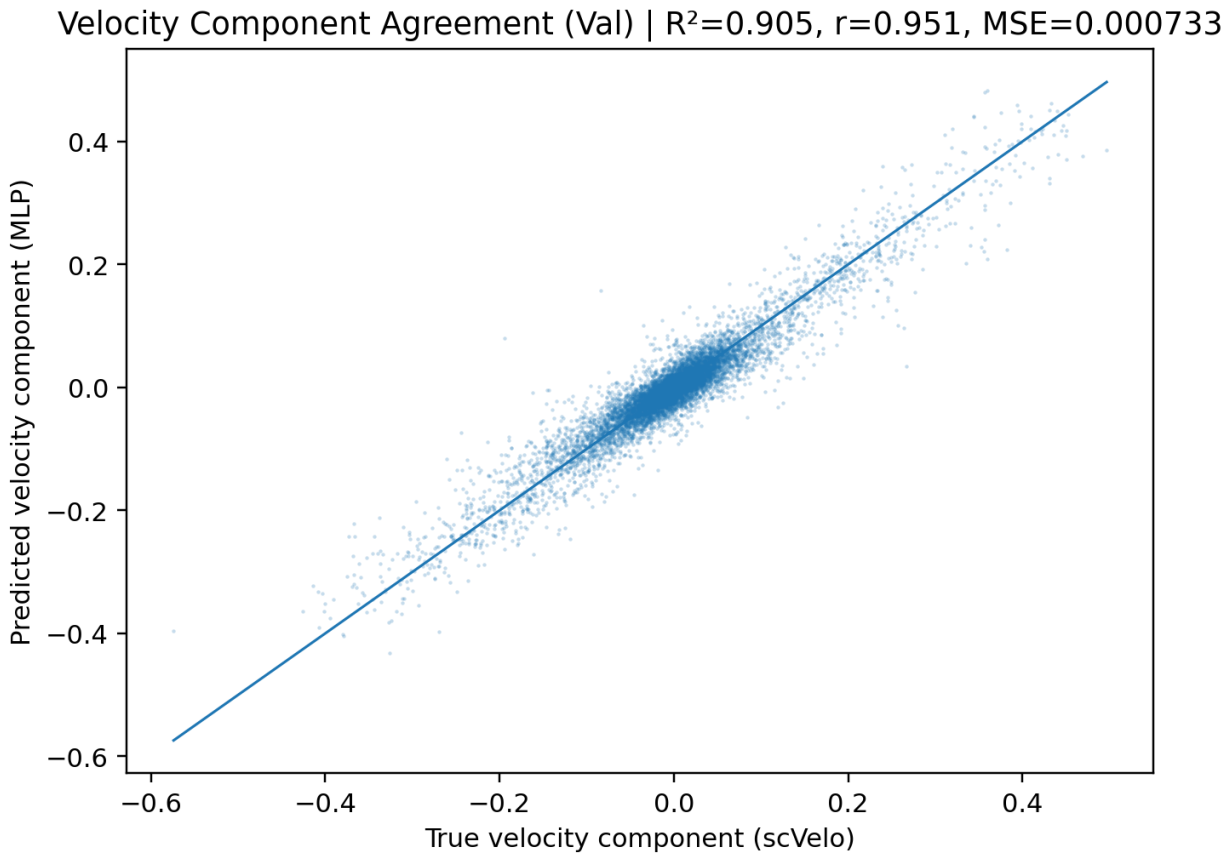

Figure 4: Predicted RNA Velocity Vs. True RNA Velocity

4.1.4 Interpretation

This result demonstrates that the learned function f_θ captures the local transcriptional vector field with high fidelity. Because the model generalizes to held-out samples, the learned field reflects structured biological signal rather than simple memorization of the training data.

Accurate reconstruction of latent-space velocity is a necessary condition for biologically meaningful forward simulation. The high R^2 value supports the use of f_θ as a continuous approximation of the RNA velocity field.

4.2 Endpoint Prediction Comparison with CellRank

A central capability of CellRank is the prediction of terminal fates through absorption probabilities in a transition kernel constructed over observed cells. Because endpoint inference is a core function of CellRank, it provides a natural benchmark for evaluating the long-term behavior of the proposed neural velocity–guided simulation framework.

4.2.1 CellRank Endpoint Behavior

CellRank integrates RNA velocity with a neighborhood graph to construct a discrete transition kernel over observed cell states. Random walks performed under this kernel converge toward terminal regions, and absorption probabilities quantify the likelihood of reaching specific fates.

Figure 5 shows representative random walks generated using the CellRank transition kernel. Trajectories remain constrained to observed cells and accumulate in biologically meaningful terminal regions of the manifold.

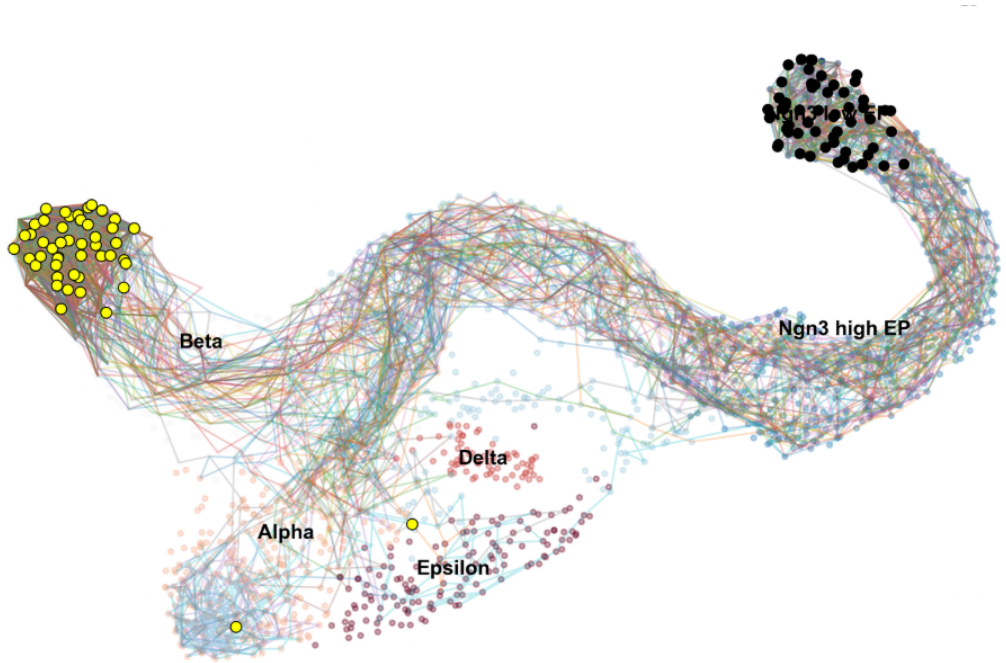

Figure 5: Random walks generated using CellRank's transition kernel. Trajectories evolve among observed cell states and converge toward terminal regions.

4.2.2 Neural Velocity–Guided Endpoint Simulation

In contrast, the proposed framework models evolution in continuous PCA latent space. Rather than restricting transitions to observed cells, trajectories evolve according to the learned velocity field with stochastic exploration and directional acceptance constraints.

To evaluate long-term behavior, large ensembles of trajectories were simulated and their terminal states collected. Figure 6 shows the distribution of simulated endpoints in UMAP space.

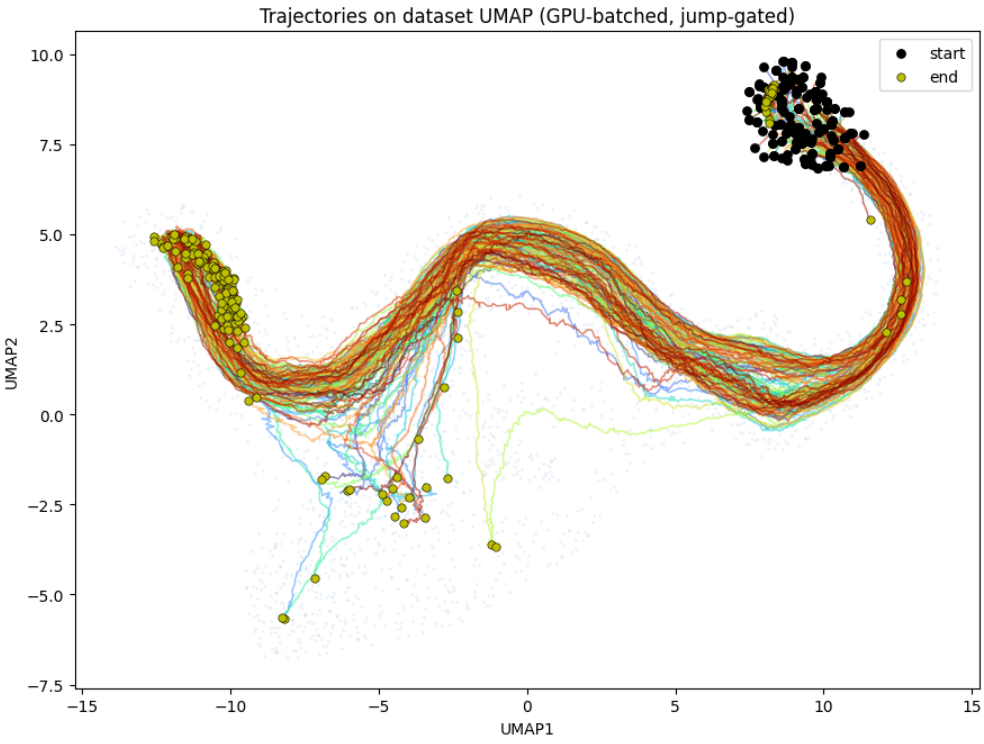

Figure 6: Simulated terminal states generated by the neural velocity–guided stochastic model. Color intensity reflects endpoint density.

4.2.3 Comparison and Interpretation

Although the modeling mechanisms differ substantially—discrete graph transitions in CellRank versus continuous latent dynamics in the present framework—the resulting endpoint distributions exhibit qualitative agreement. Simulated trajectories accumulate in similar regions of state space identified as terminal by CellRank.

This comparison demonstrates that the learned continuous velocity field preserves biologically meaningful directional structure. Importantly, the neural framework achieves endpoint prediction without explicitly constructing a transition kernel or computing absorption probabilities. Instead, terminal distributions emerge naturally from repeated stochastic simulation guided by the learned vector field.

Thus, while CellRank estimates fate probabilities over observed states, the present model produces explicit trajectory-level realizations that converge toward comparable terminal

regions.

The qualitative agreement between the two approaches indicates that the neural velocity field preserves biologically relevant directional structure rather than generating random drift.

4.3 Pancreas Population Simulation

To evaluate large-scale behavior of the learned velocity field, forward simulations were initialized from all observed cell states in the pancreas dataset.

4.3.1 Simulation Setup

Each observed latent state s_i was used as an initial condition. For each starting cell, stochastic trajectories were simulated for a fixed number of steps using the velocity-guided proposal and acceptance mechanism described previously.

Due to hardware memory constraints, trajectories were capped at a finite number of steps. Nevertheless, the simulation horizon was sufficient to observe structured redistribution of the population in latent space.

4.3.2 Endpoint Distribution

Terminal states from all simulated trajectories were collected and embedded into the shared UMAP visualization space. The resulting distribution is shown in Figure 7.

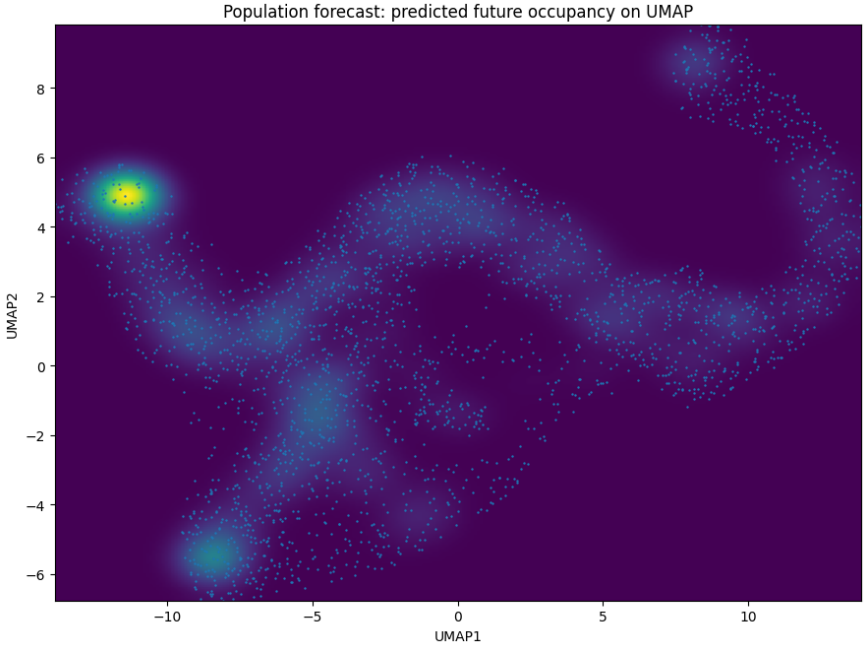

Figure 7: Population-level forecast generated by simulating forward trajectories from all observed pancreas cells. Color intensity represents density of simulated terminal states in UMAP space.

4.3.3 Results

The simulated terminal states do not disperse uniformly across latent space. Instead, trajectories concentrate in specific regions, indicating that the learned velocity field induces structured forward dynamics rather than isotropic drift.

Regions with high endpoint density correspond to transcriptionally coherent clusters, suggesting convergence toward stable or terminal transcriptional programs.

4.3.4 Interpretation

This experiment demonstrates that the neural velocity–guided stochastic framework captures population-level directional structure. When applied to the entire observed manifold, the model produces coherent redistribution consistent with biologically meaningful progression patterns.

Importantly, this behavior emerges without explicit graph construction or predefined terminal labels. The endpoint distribution is an emergent property of the learned continuous velocity field and stochastic simulation dynamics.

4.4 Independent Time-Course Validation: Bone Marrow Dataset

To evaluate whether the learned velocity field generalizes beyond the dataset used during development, the framework was tested on an independent bone marrow time-course dataset (GEO accession GSE193517).

This dataset contains hematopoietic stem and progenitor cells (HSPCs) measured at multiple experimental time points, enabling direct comparison between simulated forward evolution and empirically observed future populations.

4.4.1 Experimental Design

Only Day 0 cells were used as initial conditions for simulation.

No information from later time points was used during trajectory generation. In particular, the Day 4 cell population was excluded from the simulation process and reserved exclusively for validation.

Each Day 0 cell was projected into PCA space and evolved forward using the learned neural velocity field and stochastic proposal–acceptance mechanism described previously.

Terminal simulated states were collected after a fixed simulation horizon and embedded into the shared PCA–UMAP visualization space.

4.4.2 Comparison with Observed Day 4 Cells

Figure 8 shows the overlay of:

- Observed Day 0 cells,
- Observed Day 4 cells,
- Simulated endpoints generated from Day 0 initial states.

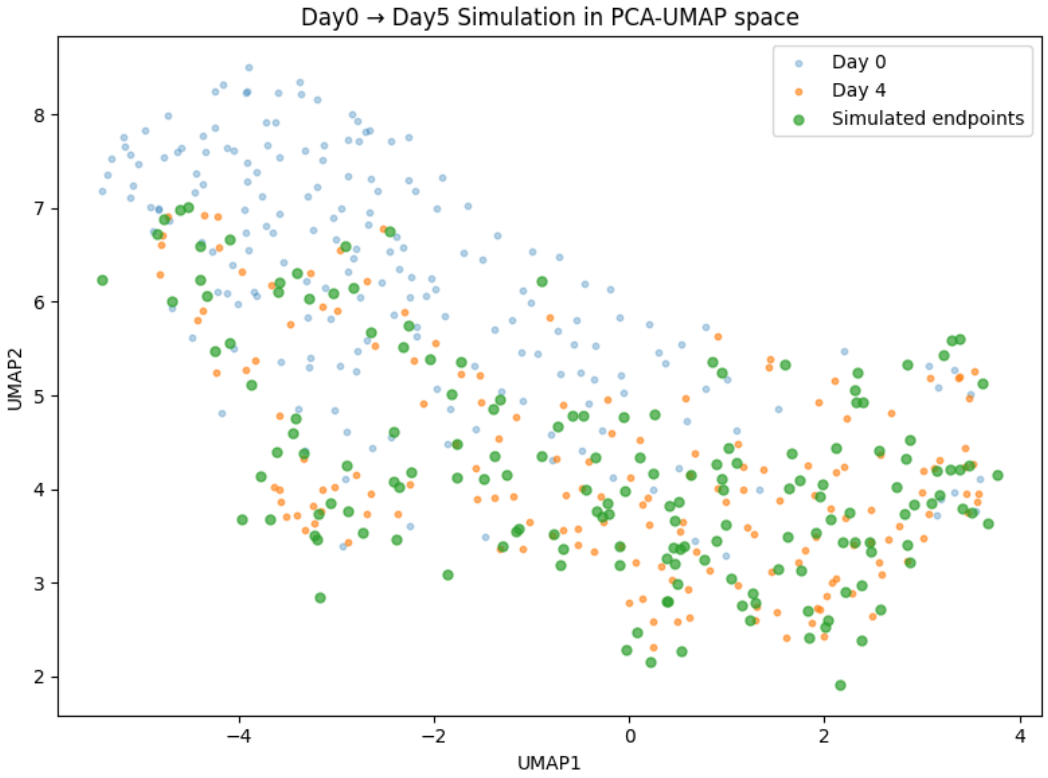

Figure 8: Forward simulation validation on the bone marrow time-course dataset. Simulated endpoints (generated from Day 0 cells only) are overlaid with observed Day 4 cells in the shared UMAP embedding.

4.4.3 Results

Simulated terminal states exhibit substantial spatial overlap with the empirically observed Day 4 population. Although differences in local density and dispersion remain, the predicted distribution shifts from the Day 0 manifold toward regions occupied by the true downstream cell states.

Importantly, this redistribution emerges without explicit training on Day 4 data. The model infers forward movement using only local RNA velocity information and the learned continuous velocity field.

4.4.4 Interpretation

This experiment provides independent validation that the neural velocity–guided stochastic framework captures structured forward transcriptomic evolution rather than merely reproducing training-set geometry.

While the model does not predict exact single-cell identities at future time points, it successfully approximates the distributional shift of the cellular population over time.

This result demonstrates that local RNA velocity information, when extended into a continuous latent vector field and coupled with stochastic simulation, can generate biologically meaningful forward forecasts across independent datasets.

5 Discussion and Future Work

The proposed framework demonstrates that RNA velocity can be extended from local directional estimation to trajectory-level forward simulation through continuous latent vector field learning and stochastic dynamics. While results on pancreas and independent bone marrow datasets support the validity of this approach, several limitations and extensions remain.

5.1 Improving Velocity Field Accuracy

The learned neural velocity field currently minimizes mean squared error between predicted and projected velocities. Although this yields strong agreement ($R^2 \approx 0.905$), further refinement is possible.

Future work may incorporate:

- Directional loss terms that explicitly penalize angular deviation rather than only magnitude error.
- Regularization strategies to enforce smoothness of the learned vector field.
- Alternative architectures, such as neural ordinary differential equation (Neural ODE) models, to better capture continuous transcriptional flow.

Improving velocity field fidelity would directly enhance long-horizon simulation stability and biological realism.

5.2 Scaling Simulation and Hardware Constraints

Due to hardware memory limitations, trajectory length and population size were capped during experiments. Running larger-scale simulations on higher-performance hardware would allow:

- Longer trajectory horizons,
- Higher-resolution endpoint distributions,
- More robust population-level forecasts.

Such scaling would provide clearer insight into long-term convergence behavior and attractor structure in latent space.

5.3 Mapping Simulation Steps to Biological Time

Currently, simulation steps represent abstract dynamical iterations rather than calibrated biological time units. Establishing a principled relationship between simulation step size and real temporal progression remains an open problem.

Future work may explore:

- Calibrating step size using experimental time-course data,
- Matching simulated velocity magnitudes to observed temporal gene expression shifts,
- Learning time-aware scaling factors directly from multi-timepoint datasets.

A calibrated time mapping would transform the framework from qualitative forward modeling into quantitatively interpretable temporal prediction.

5.4 Anomaly Detection and Pathological State Monitoring

Because the framework produces explicit trajectories in latent space, it provides a natural platform for identifying abnormal or unstable transitions.

Future extensions could incorporate:

- Density-based anomaly detection in latent space,
- Threshold-based monitoring of marker genes,
- Identification of trajectories that diverge from healthy differentiation paths.

Such capabilities may support computational flagging of pathological progression in disease contexts.

6 Conclusion

This work introduces a neural velocity–guided stochastic framework for modeling cellular state transitions in continuous latent space. By learning a continuous approximation of the RNA velocity field and coupling it with a directionally regulated stochastic simulation mechanism, the framework extends single-cell analysis beyond static clustering and fate probability estimation.

Validation results demonstrate that:

- The learned velocity field accurately reconstructs scVelo-derived latent velocities,
- Simulated trajectories produce structured endpoint distributions comparable to CellRank,
- Forward simulations initialized from Day 0 cells shift toward empirically observed downstream populations in an independent time-course dataset.

Rather than predicting deterministic single-cell fates, the framework captures population-level transcriptomic evolution through explicit trajectory generation. This shifts single-cell analysis from static inference toward dynamic forward modeling.

Overall, the results suggest that local RNA velocity information, when extended through continuous vector field learning and stochastic simulation, provides a computationally scalable pathway for forecasting cellular population dynamics from a single molecular snapshot.

References

1. La Manno, G., Soldatov, R., Zeisel, A., et al. RNA velocity of single cells. *Nature* 560, 494–498 (2018). https://doi.org/10.1038/s41586-018-0414-6

2. Bergen, V., Lange, M., Peidli, S., et al. Generalizing RNA velocity to transient cell states through dynamical modeling. *Nature Biotechnology* 38, 1408–1414 (2020). https://doi.org/10.1038/s41587-020-0591-3

3. Lange, M., Bergen, V., Klein, M., et al. CellRank for directed single-cell fate mapping. *Nature Methods* 19, 159–170 (2022). https://doi.org/10.1038/s41592-021-01346-6

4. McInnes, L., Healy, J., & Melville, J. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint* arXiv:1802.03426 (2018).

5. Wolf, F. A., Angerer, P., & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology* 19, 15 (2018).

6. Qiu, X., Martin-Rufino, J. D., Weng, C., & Hosseinzadeh, S. Mapping Transcriptomic Vector Field of Single Cells. Gene Expression Omnibus (GEO), Accession GSE193517 (2022). National Center for Biotechnology Information. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE193517

AI Assistance Disclosure

Artificial intelligence tools were used solely to improve clarity, organization, and grammatical precision of the written manuscript. All research design, mathematical modeling, data analysis, code implementation, experimental validation, and scientific conclusions were developed and executed independently by the author.