Flow Matching for Single-Cell Dynamic Biology

From Unpaired Snapshots to Population Transport

Anonymous authors

Abstract

Single-cell dynamic biology aims to understand how cellular populations develop, transition, and respond to perturbations over time. Because standard single-cell assays are destructive and unpaired, the central problem is not to recover true single-cell trajectories, but to infer distributional transport between population states. This perspective naturally aligns with generative modeling, and in particular with Flow Matching, which learns continuous velocity fields that transform one distribution into another through simulation-free velocity regression that bypasses numerical ODE solving during training. In this tutorial, we present a practitioner-oriented exploration of Flow Matching for dynamic biology, connecting snapshot marginals, optimal transport, conditional velocity regression, and the modeling assumptions that determine biological interpretation. We organize the discussion around geometric, probabilistic, and conditioning choices, and discuss how each shapes time-course reconstruction and perturbation-response prediction, enabling practitioners to diagnose modeling failures and select assumptions matched to their biological question. Code will be made available upon publication.

Keywords: Flow Matching, single-cell dynamic biology, snapshot data, distributional transport, optimal transport, generative modeling

1 Learning Biological Dynamics as Flow

In this tutorial, we aim to introduce a mathematical framework for analyzing the biological dynamics in single-cell studies. We start by elucidating the biological motivation behind characterizing the temporal dynamics of cell states. Next, we examine the type of data generated by current single-cell sequencing technologies. Due to the inherently destructive nature of standard measurements, what we obtain is not a continuous trajectory of an individual cell, but rather a series of unpaired population snapshots. By clarifying this limitation of the data, we present the issue of matching unpaired cellular populations. This leads us to an open question that will guide our discussion throughout the tutorial: how can we mathematically connect two or more unpaired distributions of single-cell populations?

1.1 Why Dynamic Biology Matters

Over the past decade, single-cell transcriptomics has enabled researchers to analyze tissue heterogeneity at single-cell resolution, overcoming the limitations of traditional population-averaged bulk analysis. Large-scale efforts such as the Human Cell Atlas have provided detailed reference maps of cellular diversity across organs, developmental stages, and disease contexts (Aldridge

and Teichmann, 2020; Rood et al., 2025). These atlases effectively address fundamental questions about what cell types exist in a tissue, which marker genes characterize them, and how the composition of cellular communities varies between healthy and diseased samples.

However, while constructing static cell atlases is scientifically valuable, it cannot fully illuminate the underlying biological mechanisms. The core functions of biological systems depend on dynamic transitions between states, not just isolated states. An atlas may reveal the coexistence of progenitor and terminal cell states within a tissue, but it cannot explain the processes governing these transitions. For instance, in hematopoiesis, a bone marrow atlas can categorize hematopoietic stem cells, progenitor stages, and mature blood lineages such as monocytes and erythrocytes. Yet it cannot address critical questions: when does a progenitor commit to the monocyte lineage instead of the erythroid lineage? Is there a point of no return in the differentiation process? Which intermediate states does a cell traverse on its path to a terminal fate?

Unlike static atlases that catalog existing states, dynamic analyses investigate how states emerge, bifurcate, and lead to lineage commitment. This focus motivates trajectory-inference and fate-probability methods, which organize single-cell data into developmental pathways, branching processes, and fate decisions (Trapnell et al., 2014; Saelens et al., 2019; Lange et al., 2022). Dynamic evolution is a recurring theme in biology: during development, progenitor cells traverse intermediate states before reaching terminal fates; during immune responses, naive cells differentiate into effector, memory, or exhausted populations; during disease progression, cells can deviate from normal transcriptional programs. Moreover, under drug or genetic perturbations, an initial cell population can be redirected toward different evolutionary endpoints, as demonstrated by high-throughput perturbation studies and perturbation-response models (Dixit et al., 2016; Replogle et al., 2022; Bunne et al., 2023).

A static atlas is akin to a high-resolution photograph of biology, while a dynamic view poses a movie-like question: how do states evolve continuously over time? In essence, a static atlas highlights differences: which cell states, marker genes, and population proportions change across time points or experimental conditions. In contrast, dynamic analysis delves into the process: how does one population transform into another over time? This shift in perspective offers several advantages. For instance, in experiments with sparse time points or only a starting and ending point, static comparisons can only indicate differences in cellular composition between the endpoints. However, dynamic modeling can interpolate the missing steps, inferring potential intermediate population states, branching sequences, and transient transcriptional programs. This turns the gap between two snapshots into a dynamic process, from which hypotheses can be generated and validation experiments designed (Schiebinger et al., 2019; Tong et al., 2020; Huguet et al., 2022; Klein et al., 2025b). Additionally, a dynamic model that learns transferable patterns from existing time series or perturbation responses can predict population behaviors at unsampled time points, unmeasured doses, or novel perturbations. With limited data, it can generate candidate hypotheses and guide prioritization for validation experiments (Yeo et al., 2021; Bunne et al., 2023).

These operations share a common computational premise: given two or more unpaired cellular population snapshots, we construct a mathematical relation to connect them. To model

this kind of problem and data, the first key question is: what form of dynamic observation can current single-cell sequencing technologies actually provide?

1.2 From Destructive Assays to Population Snapshots

Ideally, we would like to measure the same cell repeatedly without perturbing it, obtaining a true "molecular movie" across time points or perturbation conditions:

$$(x_i(t_0),\ x_i(t_1),\ \ldots,\ x_i(t_K)).$$

Each frame would record the state of cell i at one time. If we had enough such longitudinal single-cell records, learning single-cell dynamic trajectories would reduce to a supervised prediction problem: estimate a later state from an earlier state.

Standard single-cell sequencing technologies, however, do not provide such molecular movies. To measure the transcriptome of a cell, the standard protocol must lyse the cell. Sequencing destroys the cell, preventing repeated measurement. This technical barrier, often described as sequencing by destructive lysis, changes the mathematical structure of the modeling problem at a fundamental level (Tritschler et al., 2019; La Manno et al., 2018; Bunne et al., 2024).

Many existing time-series datasets are described as recording dynamic evolution at day 0, day 2, and day 4, but they do not track the same cells across time. Instead, each time point destructively samples a new batch of cells from the population. A cell observed at day 2 cannot be the day-2 state of some specific cell that was measured at day 0. The time labels in these data are real, but they do not record cell-to-cell correspondences across time points. They record the population-level distributional state of different cells sampled at different times (Schiebinger et al., 2019; Yeo et al., 2021).

This limitation changes the modeling object itself. Single-cell dynamic modeling is no longer a supervised prediction problem on paired samples $(x_i^{(t_0)}, x_i^{(t_1)})$. It is the problem of connecting two unpaired populations. The computational problem shifts from predicting the future trajectory of one specific cell to characterizing how an entire cell population distribution evolves over time.

1.3 Snapshot Marginals and the Inverse Problem

State space. Each cell is represented as a point x in a continuous state space $\mathcal{X} \subseteq \mathbb{R}^d$, obtained by projecting raw measurements through a representation map ϕ:

$$x_i = \phi(g_i) \in \mathcal{X}.$$

This map ϕ determines the "base space" on which downstream dynamic models operate. It may be the low-dimensional coordinates produced by principal component analysis (PCA) (Jolliffe, 2002), or a latent representation learned by a variational autoencoder (VAE), such as the scVI model (Lopez et al., 2018). Although different choices of state space induce different geometries, for the general discussion in this section and the derivations that follow, we assume that $\mathcal{X}$ is a preselected and fixed metric space. Section 4 returns to the effect of state-space choice on the model.

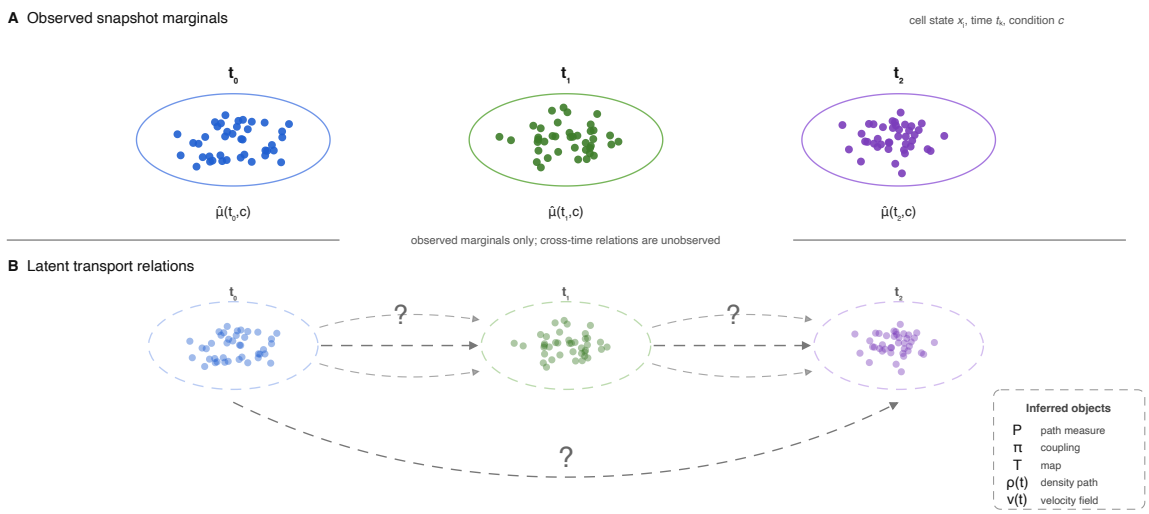

Figure 1: Observed snapshot marginals and latent cross-time relations. (a) Time-series single-cell data provide empirical marginal samples $\widehat{\mu}_{t_k,c}$ at sampled times and conditions; cross-time cell identities and correspondences are not observed. (b) Path measures, cross-time couplings or joint distributions, transport maps, continuous intermediate distributions, and velocity fields specify possible population-level relations among the observed marginals.

Snapshot marginals. Under destructive measurement, a single-cell experiment captures, at a particular time t_k and condition c, a set of cell indices $I_{t_k,c}$. We formalize all cell states observed at that time as an empirical marginal distribution $\widehat{\mu}_{t_k,c}$:

$$\widehat{\mu}_{t_k,c} = \frac{1}{n_{k,c}} \sum_{i \in I_{t_k,c}} \delta_{x_i},$$

where $n_{k,c} = |I_{t_k,c}|$. Here δ_x is the Dirac measure at the observed cell state. Its meaning is simple: without any additional assumptions, our knowledge of the population is limited to these observed discrete sample points.

Mathematically, each $\widehat{\mu}_{t_k,c}$ is a finite-sample approximation of an underlying true but unobserved population distribution $\mu_{t_k,c}$. These distributions are called marginals because they record only the cross-sectional information of cell states at that time and lose the longitudinal associations across time.

The observed-latent boundary. A collection of empirical marginal samples $\{\widehat{\mu}_{t_k,c}\}_{k,c}$, indexed by time and condition, is observable, but cell identities across time points, individual trajectories connecting different snapshots, and joint probability distributions between snapshots are not observed. This boundary, between the marginals that can be measured and the dynamic objects that must be inferred, is the conceptual starting point of the entire framework. Figure 1 summarizes it visually.

Path measures and the population process. Suppose that cell states change over time according to some process X_t. At the population level, this means that each cell traces a path $\omega : [t_0, t_K] \to \mathcal{X}$ along the continuous time axis.

Let $\Omega = C([t_0, t_K], \mathcal{X})$ be the path space consisting of all possible continuous paths. A complete population dynamics can be formalized as a probability measure $P \in \mathcal{P}(\Omega)$ on this path space.

This path measure P is the ultimate modeling target. If it truly reflects the biological evolution, then its "projection" at any sampled time t_k should agree with the snapshot that we observe. In mathematical terms, the pushforward of P at time t_k should satisfy the marginal constraint

$$(e_{t_k})_{\#}P = \mu_{t_k,c}, \quad k = 0, \ldots, K,$$

where $e_t(\omega) = \omega(t)$ is the evaluation map that sends a complete path to its state at time t.

The population-dynamics inverse problem. Single-cell dynamic modeling is an **underdetermined inverse problem**.

There are typically infinitely many candidate path measures P that satisfy the marginal constraints. Even if we consider only two adjacent time points t_0 and t_1, the data give us μ_0 and μ_1, and stop short of telling us which part of μ_0 corresponds to which part of μ_1. This missing correspondence is called a **temporal coupling** γ_{t_0,t_1}:

$$\gamma_{t_0,t_1} = (e_{t_0}, e_{t_1})_{\#}P \in \mathcal{P}(\mathcal{X} \times \mathcal{X}).$$

The marginals of this joint distribution should satisfy $(p_0)_{\#}\gamma_{t_0,t_1} = \mu_0$ and $(p_1)_{\#}\gamma_{t_0,t_1} = \mu_1$. Because γ cannot be directly observed, any joint distribution that connects the two marginals is mathematically possible.

Given marginal constraints, single-cell dynamic modeling needs a criterion—mathematical or biological—to select, from infinitely many candidate dynamics, the one most consistent with biological logic.

What is fixed and what remains open. Three objects are now fixed:

- **State space:** each cell is represented as a point x_i in $\mathcal{X}$.
- **Observed data:** empirical marginal distributions $\widehat{\mu}_{t_k,c}$, indexed by time and condition.
- **Basic inverse problem:** constraining a latent path measure P from finitely many temporal marginals.

What remains open is which P should be chosen, how the missing coupling γ_{t_0,t_1} between time points should be constrained, and whether endpoint marginals are sufficient or the distributional path at intermediate times must also be modeled. The final question is what mathematical principle can turn this underdetermined problem into a learnable and interpretable model.

The simplest structure that makes the problem tractable is a deterministic pushforward map $T_{\#}\mu_0 = \mu_1$.

2 Distribution Transport from Optimal Transport to Continuous Flows

The snapshot inverse problem is underdetermined. Infinitely many path measures fit the same marginals. To pick one we need a selection principle. The simplest candidate is a deterministic map T with $T_{\#}\mu_0 = \mu_1$. It matches the target marginal but it does not specify which source state goes where or how the population travels in between.

2.1 Static Optimal Transport and Endpoint-Level Distribution Matching

We already know that marginal constraints alone cannot determine a correspondence. We need an additional principle for choosing one. The first natural principle is geometric: if two cell states are close in the modeling space, moving the same amount of mass from one state to the other should be cheaper. Static optimal transport writes this intuition as an endpoint-level optimization problem.

2.1.1 From Maps to Couplings

Why relax a map into a coupling? To make the snapshot inverse problem tractable, the preceding section assumed that the cross-time connection could be described by a deterministic map T, a strict function that would specify that a cell in state x today must become $T(x)$ tomorrow.

This assumption is too restrictive for biology. First, during development and evolution, a progenitor-like cell can differentiate into several distinct fates with different probabilities. Second, the data come from destructive sequencing, so an absolute same-cell before-and-after correspondence is unavailable at the single-cell level.

If we insist on a deterministic map T, we mathematically forbid one-to-many fate decisions and many-to-one convergence. The question must become softer. Instead of asking exactly where each cell x went, we ask how much probability mass is assigned from state x to state y.

The rigidity of the classical Monge problem. This shift is precisely the classical move from Monge's problem to Kantorovich's relaxation (Monge, 1781; Kantorovich, 1942; Peyré and Cuturi, 2019). The original Monge problem, which searches for a deterministic map T, can be written as

$$\min_T \int_{\mathcal{X}} c(x, T(x))\, d\mu_0(x) \qquad \text{s.t.} \qquad T_{\#}\mu_0 = \mu_1.$$

Here $c(x, y)$ is the ground cost. It can be interpreted as the physical or geometric energy needed to transport one unit of mass from x to y. Integrating this cost over all cells gives the total cost that we would like to minimize. The constraint $T_{\#}\mu_0 = \mu_1$ is the pushforward constraint introduced above: after transport, the final population shape must match the observed target distribution μ_1.

The difficulty is that the Monge form is rigid in two related ways. Biologically, a deterministic map assigns each source state x to a single target state $T(x)$. It cannot represent a progenitor-like state that distributes its probability mass over several possible descendant states.

Computationally, the same restriction is awkward for empirical single-cell measures. If the source and target populations are finite sets of cells, a map must move the entire mass attached to each source cell to one target location. Exact matching is possible only when the target weights can be assembled from whole source-cell masses. Unequal sample sizes do not by themselves rule out a Monge map, but they expose a divisibility constraint that is absent once we allow mass splitting.

Kantorovich's relaxation with mass splitting. Kantorovich's key relaxation breaks this rigidity by allowing mass splitting. Rather than searching for a single function T, we search for a joint distribution $\pi(x_0, x_1)$. In optimal transport, this joint distribution is called a coupling. Intuitively, $\pi(x_0, x_1)$ is a logistics table: it records how much mass is assigned from source state x_0 to target state x_1. Because it is a probability allocation table, a single source state can naturally distribute mass to multiple target states.

The set of all legal allocation plans is denoted by $\Pi(\mu_0, \mu_1)$:

$$\Pi(\mu_0, \mu_1) = \{\pi \in \mathcal{P}(\mathcal{X} \times \mathcal{X}) : (\mathrm{proj}_0)_\# \pi = \mu_0,\ (\mathrm{proj}_1)_\# \pi = \mu_1\}.$$

Legal means that, no matter how the table reallocates mass internally, its source-side marginal, obtained by projecting onto the first coordinate, must equal the initial cell population μ_0. Likewise, its target-side marginal must equal the target population μ_1. This set is nonempty because the independent product coupling $\mu_0 \otimes \mu_1$ is always admissible; here $\otimes$ denotes the product measure, meaning that $X_0 \sim \mu_0$ and $X_1 \sim \mu_1$ are sampled independently. It is also usually very large, containing infinitely many possible correspondences.

Within this broader set, the classical Monge map appears as a special case of the Kantorovich framework. If the logistics table contains no splitting and all mass lies on the graph of a map T, then

$$\pi = (\mathrm{Id}, T)_\# \mu_0,$$

and the coupling collapses back to a deterministic transport map.

Admissible couplings are not unique. At this point, couplings seem to give cell evolution and differentiation a much more suitable mathematical form. The admissible set $\Pi(\mu_0, \mu_1)$, however, only declares which correspondences are mathematically allowed; it does not tell us which one to choose. Without a geometric criterion, the set includes infinitely many possibilities. The independent coupling $\mu_0 \otimes \mu_1$ satisfies the marginal constraints perfectly, but it ignores all distances in the cell-state space. We still need an explicit criterion for choosing the allocation table that is biologically and geometrically most plausible.

2.1.2 Kantorovich OT and the Sinkhorn Algorithm

The OT criterion finds the least-effort transport plan. We have narrowed the problem to choosing among admissible couplings in $\Pi(\mu_0, \mu_1)$. Static optimal transport adds the first explicit criterion: use the geometry of the state space. If moving mass from x_0 to x_1 incurs cost

$c(x_0, x_1)$, then the preferred coupling minimizes total transport cost (Peyré and Cuturi, 2019):

$$\pi^\star = \arg\min_{\pi \in \Pi(\mu_0,\mu_1)} \int_{\mathcal{X}\times\mathcal{X}} c(x_0, x_1)\, d\pi(x_0, x_1).$$

In empirical single-cell analysis, the source and target are finite cell sets. If the source cells are $\{x_i\}_{i=1}^n$, the target cells are $\{y_j\}_{j=1}^m$, and their weights are $a \in \Delta^n$ and $b \in \Delta^m$, then the coupling is a nonnegative matrix P. The discrete Kantorovich problem is

$$P^\star = \arg\min_{P \in \Pi(a,b)} \sum_{i,j} C_{ij} P_{ij},$$

where

$$C_{ij} = c(x_i, y_j), \qquad \Pi(a,b) = \left\{ P \in \mathbb{R}_+^{n\times m} : P\mathbf{1}_m = a,\ P^\top \mathbf{1}_n = b \right\}.$$

Thus static OT produces an endpoint-level soft correspondence matrix whose rows and columns match the observed source and target marginals.

Why the Wasserstein distance matters The Wasserstein distance answers a practical question: how much probability mass must be moved, and how far, to transform one population into another. When the ground cost is a power of a metric, $c(x, y) = d(x, y)^p$, the minimized OT cost defines the p-Wasserstein distance (Peyré and Cuturi, 2019):

$$W_p(\mu_0, \mu_1) = \left(\inf_{\pi \in \Pi(\mu_0,\mu_1)} \int d(x, y)^p\, d\pi(x, y) \right)^{1/p}.$$

This is especially useful for single-cell dynamics, where source and target populations may occupy different regions of state space. Pointwise divergences such as KL can be ill behaved without overlapping support or reliable density estimates (Arjovsky et al., 2017). The Wasserstein distance instead compares snapshot distributions through the geometry of the modeling space.

From theory to practice with entropic regularization and Sinkhorn. Solving the exact discrete OT linear program is often too expensive for single-cell datasets. The standard scalable compromise is entropic regularization (Cuturi, 2013):

$$P_\varepsilon^\star = \arg\min_{P \in \Pi(a,b)} \sum_{ij} C_{ij} P_{ij} + \varepsilon \sum_{ij} P_{ij}(\log P_{ij} - 1).$$

The entropy term regularizes the transport linear program into a smooth, strictly convex problem whose optimizer admits a matrix-scaling form. Let

$$K_{ij} = \exp(-C_{ij}/\varepsilon).$$

The optimality conditions imply that the regularized plan can be written as

$$P_\varepsilon^\star = \operatorname{diag}(u)\, K\, \operatorname{diag}(v),$$

Table 1: Coupling diagnostics for EB time 1 to time 2.

Method	ε	Expected cost	Normalized cost	Entropy	Effective support	Converged	Backend
Independent	–	299.60	1.185	11.04	62500.00	–	independent
Sinkhorn OT	0.001	182.49	0.722	6.47	642.98	no	scaling fallback
Sinkhorn OT	0.01	147.96	0.585	6.35	572.08	no	POT
Sinkhorn OT	0.05	161.36	0.638	8.26	3853.01	yes	POT
Sinkhorn OT	0.1	182.41	0.722	9.42	12275.48	yes	POT
Sinkhorn OT	1.0	277.71	1.099	11.00	59826.43	yes	POT
Sinkhorn OT	5.0	295.37	1.169	11.04	62394.88	yes	POT

where u and v are positive scaling vectors chosen so that the row and column sums match a and b. A derivation is given in the supplementary material. Enforcing these marginal constraints gives the Sinkhorn updates, with division understood elementwise:

$$u^{(\ell+1)} = \frac{a}{Kv^{(\ell)}}, \qquad v^{(\ell+1)} = \frac{b}{K^\top u^{(\ell+1)}}.$$

Alternating these row and column rescalings constructs the regularized transport plan using matrix-vector products rather than a full linear-program solver.

The parameter ε controls both numerical behavior and modeling sharpness. As $\varepsilon \to 0$, the plan approaches exact OT and concentrates mass on low-cost pairs, but numerical instability becomes more likely. As ε grows, convergence is easier but the plan becomes diffuse and approaches the independent coupling. We usually tune ε on a held-out coupling diagnostic before trusting the plan for training.

An embryoid body example. To make the theory concrete, we use embryoid body (EB) differentiation data (Moon et al., 2019) and compare the cell populations from time point 1 to time point 2. The cost is computed in a 20-dimensional principal-component space. Figure 2 shows the learned coupling in three complementary views, together with the ε sweep in panel (d). The sweep makes a point worth noting. ε controls a trade-off between geometric sharpness and numerical stability, and the right value depends on the cost scale of the data. For display only, cell expression profiles are projected to a PHATE visualization embedding (Moon et al., 2019), which is also used to order cells in the coarse block view; all transport costs remain computed in PC-20. Table 1 reports the full coupling diagnostics.

Table 1 reports diagnostics for uniform-marginal couplings P between 250 EB source cells at time 1 and 250 target cells at time 2. Let $C^{\text{raw}}_{ij} = \|x^{(1)}_i - x^{(2)}_j\|^2_2$ be the PC-20 squared-distance cost matrix, and let $c_{\text{ref}} = \text{median}\{C^{\text{raw}}_{ij} : C^{\text{raw}}_{ij} > 0\} = 252.759$ be the positive median cost for this diagnostic. The raw expected cost is $\langle C^{\text{raw}}, P\rangle = \sum_{ij} C^{\text{raw}}_{ij} P_{ij}$, and the normalized cost is $\langle C^{\text{raw}}/c_{\text{ref}}, P\rangle$. Entropy is the natural-log Shannon entropy $H(P) = -\sum_{ij:P_{ij}>0} P_{ij} \log P_{ij}$, and effective support is $\exp(H(P))$, so it measures coupling diffuseness rather than accuracy. The independent baseline uses $P_{ij} = (1/250)(1/250)$. POT denotes Python Optimal Transport's `ot.sinkhorn` (Flamary et al., 2021); the scaling fallback is our iterative row–column scaling on the normalized Gibbs kernel. A Sinkhorn run is marked converged only when the final marginal L_1 error is at most 10^{-4} and no captured warning indicates non-convergence or numerical failure.

The diagnostics show the expected trade-off. The independent coupling spreads each source cell nearly uniformly over all target cells, giving high expected cost and maximal effective sup-

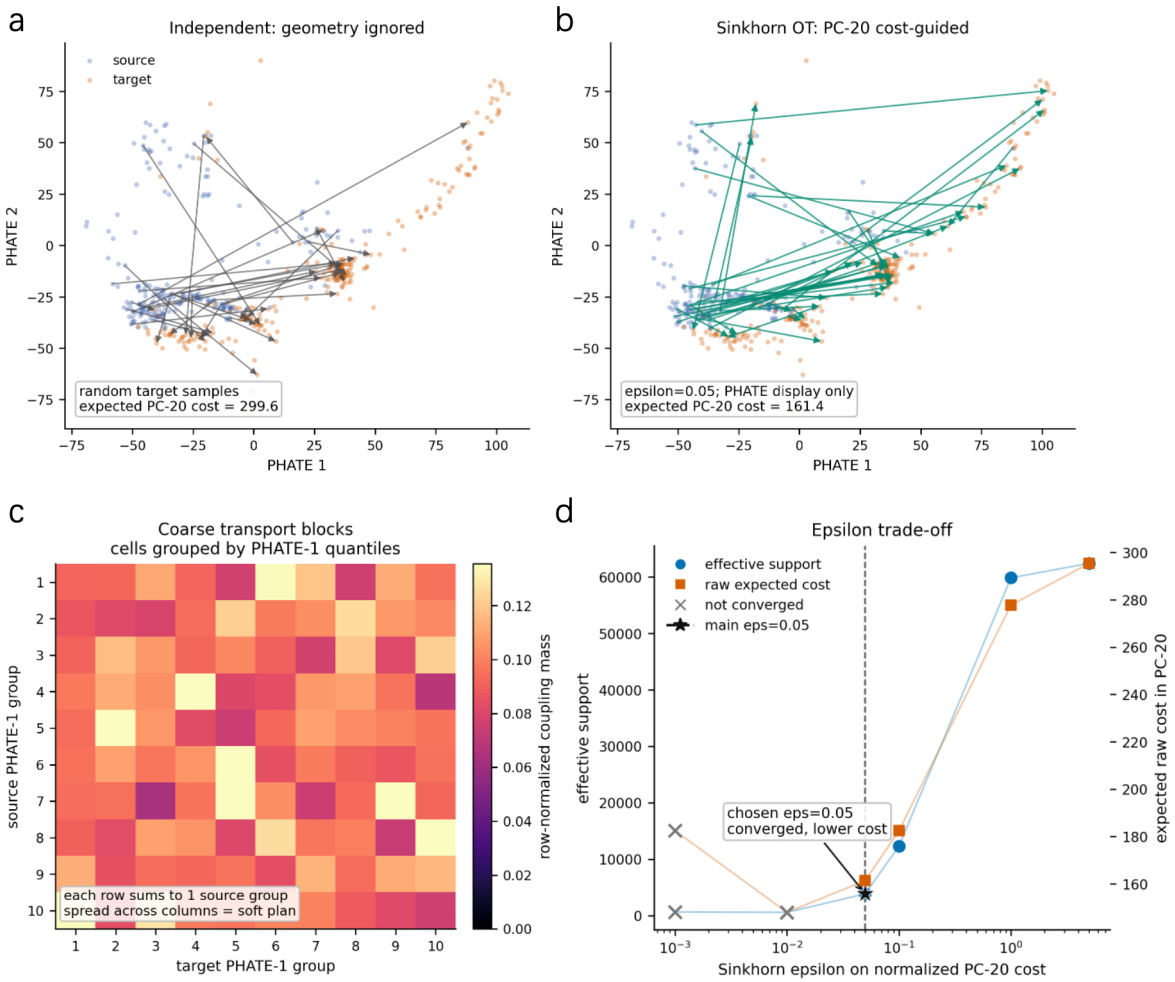

Figure 2: Static entropic OT endpoint transport on EB snapshots. (a) Independent source–target coupling ignores geometry and produces high expected transport cost in PC-20. (b) Sinkhorn OT with $\varepsilon = 0.05$ uses the PC-20 ground cost to concentrate mass on lower-cost endpoint correspondences. (c) Coarse block view of coupling mass between PHATE-ordered source and target groups. (d) ε sweep: effective support, expected PC-20 cost, and numerical convergence. The chosen setting is $\varepsilon = 0.05$.

port. Large ε values behave similarly because entropy dominates. We set $\varepsilon = 0.05$ as a practical compromise: it lowers expected cost and effective support while still converging. Very small ε values push the cost lower but fail to converge, illustrating why practical OT in single-cell data must balance geometric sharpness against numerical stability.

2.1.3 Biological Interpretation of an OT Plan

Reading an OT plan After the derivation and computation above, we have a satisfactory optimal logistics table, the matrix $P^\star$. The question is how to extract biological meaning from such a large matrix. We can treat it as a soft endpoint correspondence. Cell evolution often permits one-to-many outcomes. For a particular source-time cell x_i, the normalized row P_{ij}/a_i describes its possible descendant distribution. Conversely, for a target-time cell y_j, the normalized column P_{ij}/b_j describes its possible ancestor distribution. This is the central logic by which

Table 2: OT plan sensitivity across PC dimensions.

PC dimension	ε	Expected cost	Entropy	Effective support	L_1 distance to PC-20 plan
5	0.05	67.78	8.48	4808.31	1.303
10	0.05	108.14	8.28	3956.63	0.814
20	0.05	157.68	8.30	4017.56	0.000
50	0.05	227.99	8.52	4999.14	0.573

Waddington-OT uses coupling matrices to visualize fate probabilities (Schiebinger et al., 2019).

The OT plan is a model-implied correspondence under a specific set of assumptions. No physical line is measured directly in the biological experiment. Its final form depends deeply on subjective modeling choices: which genes or latent variables define the state space, which metric defines the ground cost, and what value of the entropic regularization parameter ε is used. Changing any of these choices can rewrite the allocation table.

To make this sensitivity visible, we run a simple comparison that changes only the PCA dimension while holding the other settings, including $\varepsilon = 0.05$, fixed. Table 2 reports the result.

The rightmost column shows that changing the PCA dimension shifts the final allocation plan substantially, as measured by L_1 distance to the PC-20 baseline. Expected cost and effective support also vary strongly. The conclusion is that an OT plan is highly sensitive to how we build the state space.

One more distinction will matter throughout the tutorial. In this section, static OT is a standalone method for matching distributions: it takes two marginal distributions as input and returns a coupling matrix. Later, especially in Section 4, OT appears in a different role as an endpoint-pairing strategy inside Flow Matching training.

Endpoint coupling does not determine dynamics. An OT plan gives endpoint correspondence probabilities. It stops short of describing the actual dynamics of cell evolution. Even if $P^\star$, or equivalently the joint distribution $\pi(x_0, x_1)$, answered perfectly which starts and ends are connected, it would still say nothing about the process in between. It does not contain intermediate density paths, time parameterization, or the velocity field that describes the direction of cellular motion. An OT coupling defines a static start-end relation. Moving from that endpoint relation to a continuous population process requires another layer of structure.

Sampling a pair of corresponding cells (X_0, X_1) from the coupling π gives only the boundary conditions of a trajectory. To turn the two isolated endpoints into a continuous path, we must additionally specify a path kernel

$$K_\tau(dx \mid x_0, x_1).$$

Only after introducing this extra kernel have we answered the dynamic question of where the cell should be at intermediate bridge time τ given the start and end.

The simplest path choice is a straight-line interpolation:

$$X_\tau = (1-\tau)X_0 + \tau X_1, \qquad \dot{X}_\tau = X_1 - X_0.$$

But this is only a mathematical convenience. We could also construct curved deterministic bridges or stochastic Brownian bridges. They satisfy the same endpoint coordinates but describe

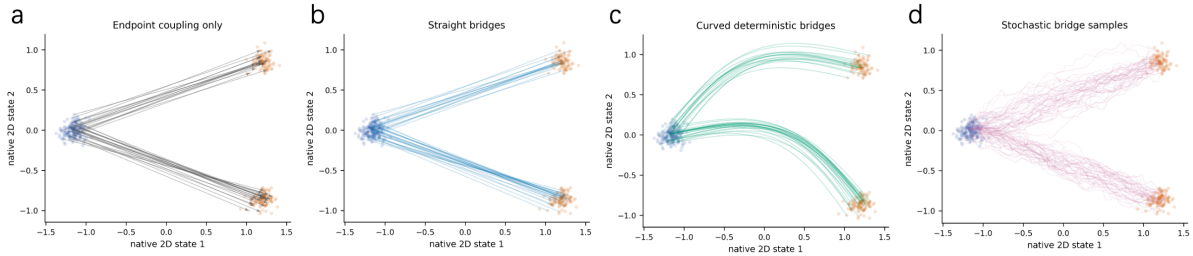

Figure 3: Same endpoint pairs under three path families. (a) OT plan endpoint correspondences, shown as fixed source–target arrows. (b) Straight deterministic bridges on the same pairs. (c) Curved deterministic bridges on the same pairs, changing intermediate density, time parameterization, and velocity field. (d) Stochastic Brownian bridge samples on the same pairs.

Table 3: Path diagnostics for the same endpoint coupling.

Path family	Mean endpoint distance	Mean path length	Energy proxy	Midpoint deviation	Midpoint spread	Pairs	Steps
Straight bridge	2.485	2.485	6.19	0.000	0.000	80	51
Curved deterministic bridge	2.485	2.714	7.41	0.497	0.021	80	51
Brownian stochastic bridge	2.485	3.076	11.32	0.152	0.470	80	51

Energy proxy uses the same discrete kinetic-action approximation as action (defined in the supplementary material), computed as $\sum_k \|\Delta x_k\|^2/\Delta t$ at the reported number of steps. Midpoint deviation: Euclidean distance between the path midpoint and the straight-chord midpoint. Midpoint spread: standard deviation across stochastic bridge samples at the midpoint.

different intermediate states and path geometries.

To demonstrate this point, we use a simple two-dimensional branching toy dataset. We fix the same endpoint pairs sampled from an OT coupling and then impose three different path rules: straight, curved, and Brownian. Figure 3 separates the endpoint relation from the imposed path family, and Table 3 reports how the resulting path diagnostics change.

But once the path assumption changes, the mean path length and the energy proxy differ substantially. The straight bridge has the lowest energy. The curved bridge pays extra cost because it detours. The Brownian bridge pays still more because random fluctuations lengthen the route. This is the message of Figure 3: the same endpoints can support very different trajectories.

The induced intermediate distribution

$$\mu_\tau(dx) = \int K_\tau(dx \mid x_0, x_1)\, d\pi(x_0, x_1)$$

is determined by two separate choices. Static OT helps choose the first object, the correspondence π. It does not choose the second object, the path kernel K_τ.

Static OT provides endpoint correspondence only. Dynamic optimal transport then supplies the missing dynamics, answering how intermediate densities evolve and what velocity field drives them.

2.2 Dynamic Optimal Transport and Density Paths with Velocity Fields

Static OT gives only a logistics table between start and end. A continuous biological population dynamic needs two additional objects.

First, a continuous process cannot look only at the two endpoints. We need to know the

population distribution at any intermediate time t. This gives the first required object, a time-indexed density path $\rho_t(x)$. Second, a density path alone is not enough. We also need to know how cells move to their next positions: at every point in the space, how fast are they moving and in which direction? This requires a velocity field $v_t(x)$.

Dynamic optimal transport puts these two objects, the density path and the velocity field, into one mathematical framework and binds them by the physical principle of mass conservation.

2.2.1 Two Languages for Dynamics, from Particle Trajectories to Density Fields

When we try to pass from two discrete snapshots to a continuous population process, the most intuitive idea is to watch one particular cell and follow its trajectory over time. In physics, this particle-tracking perspective is called the Lagrangian view.

Because destructive single-cell sequencing prevents tracking the same cell across time points, we must move from the language of individual particle trajectories to the language of distributional transformation—the Eulerian view, the other classical perspective from fluid mechanics.

In the Eulerian view, we no longer label and follow one cell. Instead, like an observer standing at a fixed position in a river, we fix a coordinate x in the state space and observe how many cells are present there at time t, represented by $\rho_t(x)$, and how fast and in what direction cells are flowing there, represented by $v_t(x)$. This change of viewpoint moves us from unobserved single-cell trajectories to continuous fields that can be modeled at the population level.

We can still begin from the Lagrangian side to see the microscopic relation. If an ideal particle trajectory X_t follows the velocity field v_t, then it satisfies the ODE

$$\frac{dX_t}{dt} = v_t(X_t).$$

If the velocity field is regular enough to define a flow map Φ_t, meaning the map that sends an initial position x_0 to its position at time t, then the initial population density ρ_0 is pushed forward to the intermediate population density:

$$\rho_t = (\Phi_t)_{\#}\rho_0.$$

The change-of-variables formula gives

$$\rho_t(\Phi_t(x_0)) = \rho_0(x_0)\left|\det(\nabla_{x_0}\Phi_t(x_0))\right|^{-1}.$$

Intuitively, if the flow map stretches a region, the cell density there is diluted; if it compresses a region, the density increases.

2.2.2 The Continuity Equation

Once the velocity field pushes the density forward, the density path ρ_t and velocity field v_t cannot be chosen independently. If water flows out of a region of a pool, the water level there must fall; if water flows in, the water level must rise. The same mass-conservation idea gives the continuity equation:

$$\partial_t\rho_t(x) + \nabla\cdot\big(\rho_t(x)v_t(x)\big) = 0.$$

The quantity $\rho_t(x)v_t(x)$ is the flux: the amount of cell mass crossing unit area per unit time. The divergence operator $\nabla\cdot$ measures net local outflow or inflow. In plain language, the local rate of density change at x, $\partial_t\rho_t$, must cancel the net rate at which cell mass flows out of that location, $\nabla\cdot(\rho_t v_t)$.

This equation is not an arbitrary constraint. It is the inevitable consequence of defining density evolution by a flow map. Let the flow map be Φ_t and its Jacobian be $J_t = \nabla_{x_0}\Phi_t(x_0)$. Change of variables gives

$$\rho_t(\Phi_t(x_0))\det(J_t) = \rho_0(x_0).$$

Differentiating this identity in time and using Jacobi's formula together with the product rule for divergence yields the continuity equation. A concise derivation is given in the supplementary material. The equation acts like a strict consistency condition: density and velocity must describe the same population evolution.

The continuity equation above is balanced, meaning it assumes that total cell mass is conserved. In real biological dynamics such as embryonic development, cells proliferate and die, so total mass can change. Such cases require unbalanced transport with source terms. In this section, however, the balanced continuity equation is a useful simplifying model that keeps the mathematical line clear.

2.2.3 Benamou–Brenier Path Choice by Minimum Action

The continuity equation ties density paths and velocity fields together and removes many physically impossible motions. But there are still infinitely many pairs (ρ_t, v_t) that satisfy the equation and connect the same endpoints. Just as in static OT, we again face a large admissible set and need a criterion. In static OT, the ruler was the endpoint transport cost. In continuous dynamics, we want a path whose overall motion costs the least effort.

Benamou and Brenier gave such a principle in 2000 (Benamou and Brenier, 2000). They showed that the usual W_2 optimal transport distance can be written equivalently as a dynamic variational problem over continuous fields. Static OT and dynamic OT are two descriptions of the same W_2 geometry. Under squared Euclidean cost, the minimum action equals the static OT cost. A compact statement appears in the supplementary material. Specifically, among all (ρ_t, v_t) satisfying the continuity equation and the endpoint boundary conditions, we choose the one minimizing total kinetic action:

$$W_2^2(\mu_0,\mu_1) = \inf_{\rho_t, v_t} \int_0^1 \int_{\mathcal{X}} \rho_t(x)\|v_t(x)\|^2\, dx\, dt$$

subject to

$$\partial_t\rho_t + \nabla\cdot(\rho_t v_t) = 0, \qquad \rho_0 = \mu_0, \quad \rho_1 = \mu_1.$$

Here $\|v_t(x)\|^2$ is squared speed, and $\rho_t(x)$ is the local mass weight. The space-time integral is the weighted total kinetic energy of the population motion. Minimizing it penalizes unnecessary detours, local swirling that moves quickly without advancing the global transport, and abrupt changes in speed.

This low-action criterion does not mean that real biology must move in straight lines. It is a

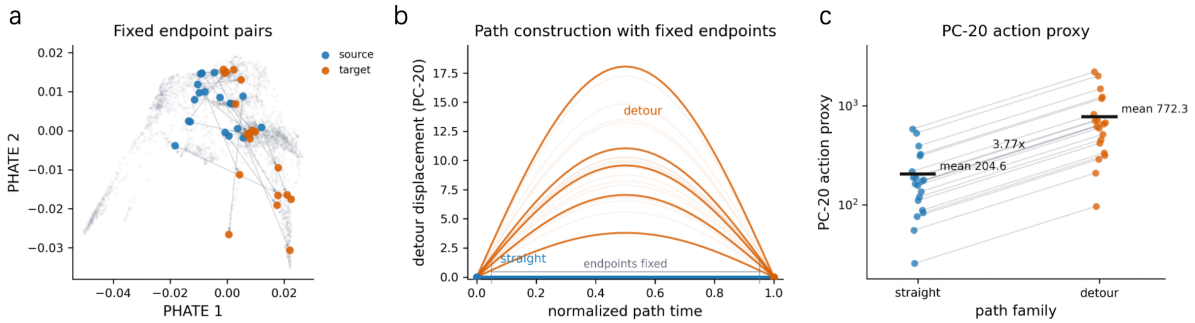

Figure 4: Dynamic OT low-action diagnostic on fixed EB endpoint pairs. (a) Twenty source–target endpoint pairs from the same PC-20 OT coupling, displayed in PHATE; gray segments mark paired endpoints, not trajectories. (b) Straight and detour paths in a path-local PC-20 coordinate: straight paths have zero perpendicular displacement, detours leave the chord and return, penalized by squared-velocity action. (c) Kinetic-action proxy in PC-20 for straight versus detour paths across 41 steps; gray lines connect matched pairs, black bars denote means. Detour paths increase action by roughly 3.8×.

mathematical prior: in the absence of richer observations, we prefer short, smooth, low-energy population motion. It is the distribution-transport analogue of the physical principle of least action.

There is also a single-particle intuition. If endpoints are fixed and the cost is the time integral of a convex function of velocity, most commonly squared velocity, Jensen's inequality says that the lowest-action motion keeps the average velocity constant. This gives a constant-speed straight path in Euclidean space. The short proof is included in the supplementary material.

For an intuitive EB demonstration, we compare straight and detour paths in Figure 4. We do not solve an expensive PDE. Instead, we take fixed endpoint pairs sampled from the PC-20 OT coupling and change only the intermediate path construction. The straight construction follows the endpoint chord; the detour construction leaves the chord in a perpendicular PC-20 direction and returns to the same target. The kinetic-action proxy increases by almost a factor of four, supporting the intuition behind the minimum-action principle: detours are expensive.

Dynamic OT provides the target object—a continuous velocity field $v_t(x)$ that transports probability mass through the population—and the minimum-action criterion. But the Benamou–Brenier formulation is an infinite-dimensional PDE variational problem. It is not a stochastic-gradient objective on finite samples, and its solution is tied to one source–target pair instead of a parameterized model that generalizes. A direct approach is to parameterize the velocity field as a neural network $v_\theta(x, t)$ and learn it from data.

2.3 Continuous Normalizing Flows as Learnable Continuous Transport

Dynamic OT gives the target objects, a density path ρ_t and a velocity field v_t. It remains an abstract variational problem and has no trainable parameters yet. We parameterize the velocity field directly with a neural network. Continuous Normalizing Flows (CNFs) build on this idea and turn the dynamic-transport object into an implementable neural model.

2.3.1 Neural Velocity Fields and ODE Flows

So far, v_t has been an abstract function. To make it operational, we introduce a neural network $v_\theta(x, t)$, where θ denotes trainable weights. Given a current state x and time t, the network outputs a velocity vector. This is the continuous-depth limit of normalizing flows and Neural ODEs introduced by Chen et al. (Chen et al., 2018).

Once we replace the abstract velocity field with a neural network, we define a Neural ODE:

$$\frac{dX_t}{dt} = v_\theta(X_t, t), \qquad X_0 \sim \rho_0.$$

This equation says that at time $t = 0$, we sample an initial cell state X_0 from ρ_0. We then integrate forward under the velocity prescribed by v_θ. As time evolves, we can compute the state X_t at any intermediate time.

If the velocity field is regular enough, this evolution induces a deterministic flow map ψ_t^θ. The map sends the initial state x to its state at time t:

$$\frac{d}{dt}\psi_t^\theta(x) = v_\theta(\psi_t^\theta(x), t), \qquad \psi_0^\theta(x) = x.$$

When all cells move along this flow map, the population distribution changes as well:

$$\rho_t^\theta = (\psi_t^\theta)_\# \rho_0.$$

The formerly abstract velocity field has now become a trainable model controlled by neural-network parameters θ.

CNFs connect the objects discussed throughout this section. At the start of the section, we asked for a pushforward map T such that $T_\# \mu_0 = \mu_1$. In a CNF, the endpoint map ψ_1^θ plays the role of T. Dynamic OT asked for a density path ρ_t and a velocity field v_t. In a CNF, the ODE-induced distributions ρ_t^θ supply the density path, and the neural network v_θ supplies the velocity field. CNFs give a unified neural ODE implementation of the transport objects introduced above.

2.3.2 Density Tracking and the Instantaneous Change of Variables

The goal in single-cell dynamic modeling is not only to simulate how an individual sample moves along a path. We also want to track how the whole population density changes. This requires more than an ODE for sample transport; it also requires a density evolution rule.

CNFs provide this rule through the instantaneous change-of-variables formula:

$$\frac{d}{dt} \log \rho_t^\theta(X_t) = -\nabla \cdot v_\theta(X_t, t).$$

This formula is the continuous-time consequence of the Jacobian determinant of the flow map; the supplementary material gives the derivation. Its physical meaning is clear: the rate of change of log density along a cell trajectory depends entirely on the divergence of the velocity field at that point. Divergence measures whether local volume expands or contracts. If the field diverges outward, volume expands, nearby cells are pulled apart, and log density decreases because of

Table 4: CNF solver-in-loop training bottleneck.

Method	Steps	Batch size	Mean step time (ms)	Median step time (ms)	Mean NFE/step	Total wall time (s)	Final loss	Final distribution MMD
Solver-in-the-loop Neural ODE training proxy	100	256	42.41	25.33	16.0	4.28	0.0118	0.00079

NFE: number of function evaluations per training step. MMD: maximum mean discrepancy, defined in the supplementary material. Lower final loss and MMD are better.

the minus sign. If the field contracts inward, cells are compressed and density increases.

This formula lets CNFs build a full training pipeline for real data: a learnable neural velocity field can transport samples while also accounting for how probability density evolves.

2.3.3 Training Cost

CNF training on real data brings a heavy computational burden.

In likelihood-based CNF training, we often need both the future state of a sample and its log density. This requires an augmented Neural ODE:

$$\frac{d}{dt}\begin{bmatrix} X_t \\ \log \rho_t(X_t) \end{bmatrix} = \begin{bmatrix} v_\theta(X_t, t) \\ -\nabla \cdot v_\theta(X_t, t) \end{bmatrix}.$$

Each training step then follows a computationally expensive procedure. First, sample an initial cell state X_0. Next, call a numerical ODE solver and integrate the augmented ODE forward to $t = 1$, obtaining both a predicted endpoint X_1 and the log density $\log \rho_1(X_1)$. Only after this rollout can the loss be evaluated against the target distribution. Finally, gradients must backpropagate through the entire ODE solution process to update network parameters.

This control flow hides several costs. First, the ODE rollout sits inside the training loop: every loss evaluation requires many forward integration steps, each calling the neural network v_θ. Second, computing divergence in high dimension is expensive. Even the Hutchinson trace estimator used in FFJORD-style CNFs, which stochastically approximates the divergence, introduces estimator variance into the training signal (Grathwohl et al., 2019). Third, gradients must pass through the solver. Whether one stores intermediate states directly or uses adjoint sensitivity methods, training becomes slower and more memory intensive.

For single-cell evolution, this burden is especially severe. Empirical marginals are discrete and noisy; continuously tracked trajectories are unavailable. A continuous dynamic model is desirable, but not at the cost of a full ODE simulation inside every parameter update.

CNFs identify the correct model form—a learnable velocity field that transports distributions through an ODE flow—but their traditional training procedure is too heavy. The goal is to keep the neural velocity field while removing the expensive ODE solver from the inner loop of every training update.

To quantify this solver-in-the-loop control flow, we ran a minimal two-dimensional toy experiment. Even in this small model, placing the ODE solver inside the training loop forces 16 neural velocity evaluations per update (Table 4). Scaling this control flow to high-dimensional biological data would make the cost grow quickly.

3 Flow Matching for Efficient Training of Continuous Flows

CNFs identify a learnable velocity field that transports distributions through an ODE flow. But their training loop is too heavy to scale to large-scale single-cell data. Flow Matching keeps the object and removes the bottleneck (Lipman et al., 2023). It replaces full trajectory rollout with local velocity regression at randomly sampled points.

3.1 The Core Idea of Replacing Simulation with Regression

Instead of asking a model to roll out an entire trajectory and then correcting the endpoint, Flow Matching asks the model to predict the instantaneous velocity at a randomly chosen point on a prescribed probability path. The supervision is local: a time t, an intermediate state x_t, and a target velocity at that point.

3.1.1 Probability Paths and Target Velocities

Let p_0 and p_1 denote the source and target population distributions in a chosen state space $\mathcal{X}$. A *probability path* is a family of distributions $\{p_t\}_{t\in[0,1]}$ interpolating between them. At each intermediate time, $p_t(x)$ describes the population shape implied by the chosen interpolation.

The corresponding target velocity field, denoted $u_t(x)$, is the local vector field that transports the density along this path. As in Section 2.2.2, consistency between a density path and a velocity field is expressed by the continuity equation:

$$\partial_t p_t(x) + \nabla \cdot \big(p_t(x) u_t(x)\big) = 0.$$

This equation describes conservation of probability mass. The velocity field appearing in it is not the true physical velocity of an individual cell; it is a modeling target induced by the probability path we choose.

3.1.2 The Flow Matching Objective

The ideal Flow Matching loss has a clean formal expression, but in real high-dimensional data we do not know the target velocity $u_t(x)$ in closed form. Formally, the loss is the mean-squared error between the learned velocity and the target velocity along the probability path:

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t\sim\mathrm{Unif}(0,1),\, X_t\sim p_t}\left[\|v_\theta(X_t,t) - u_t(X_t)\|^2\right].$$

The loss samples a time t, samples a state X_t from the corresponding intermediate distribution, and regresses the model output to the local target velocity. Nothing in this expression requires rolling out the current model from $t = 0$ to $t = 1$. The gap between this clean objective and practical training is what motivates the distinction between a conditional training path and the marginal velocity field learned by the network.

3.1.3 The Marginal Velocity

Let Z denote the hidden construction variable used to generate a training path. Later, Z will be a pair of source and target cells. For now, it can be any variable indexing a simple conditional path. The marginal velocity at a location x is the posterior average of all conditional velocities whose paths pass through x at time t:

$$u_t(x) = \mathbb{E}[u_t(X_t \mid Z) \mid X_t = x] = \frac{\int u_t(x \mid z)\, p_t(x \mid z)\, p(z)\, dz}{\int p_t(x \mid z)\, p(z)\, dz}.$$

This formula has two consequences.

First, it explains the computational obstruction. Evaluating $u_t(x)$ exactly would require the density of the entire intermediate population and the posterior contribution of every conditional path passing through x. This is not available in single-cell state spaces with thousands of cells and tens to thousands of dimensions. The ideal loss $\mathcal{L}_{\mathrm{FM}}$ is not directly executable.

Second, it explains why conditional supervision does not simply teach the network to memorize artificial paths. The construction variable Z is used to sample an intermediate point and a target velocity, but the network typically receives only (x_t, t), or (x_t, t, c) when experimental conditions are included. If many conditional paths pass through the same local region, the network must output one shared velocity for that region. Under squared loss, the optimal shared prediction is the posterior mean of the conditional velocities, which is exactly the marginal velocity above.

This distinction is especially important in snapshot biology. A conditional path is a mathematical construction used for supervision. It is not an observed lineage, and it should not be read as the molecular history of a particular cell.

3.1.4 Sampling and Inference

Flow Matching is often called simulation-free. The statement is correct only for training. During training, a mini-batch can be formed by sampling intermediate times and states, evaluating v_θ, and computing a local regression loss. The model's ODE is not unrolled inside the loss.

At sampling time, the learned field must still be integrated:

$$\widehat{X}_1 = \mathrm{ODESolve}\Big(v_\theta, \widehat{X}_0, 0, 1\Big), \qquad \widehat{X}_0 \sim \widehat{\mu}_0.$$

Thus Flow Matching changes the computational structure of training. It does not remove the need to choose a numerical solver when generating future cell states.

3.2 Conditional Flow Matching from Intractable to Sampleable

The ideal target $u_t(x)$ is a marginal object. Conditional Flow Matching makes the training problem practical by replacing this unavailable target with conditional velocities that are easy to sample and differentiate.

3.2.1 From Marginal Velocity to Conditional Velocity

Let $Z \sim p(z)$ be a conditional construction variable. In the endpoint-pair setting used throughout this section,

$$z = (x_0, x_1),$$

where x_0 is drawn from the source snapshot and x_1 from the target snapshot. We use z for the hidden path-construction variable and c for optional experimental conditions such as perturbation label, dose, batch, donor, or patient background. The condition c indexes observable experimental context, whereas z indexes the hidden path construction.

For a fixed z, define a conditional probability path

$$X_t \mid Z = z \sim p_t(\cdot \mid z).$$

Because this path is designed by the modeler, it can be chosen to have a tractable velocity $u_t(x \mid z)$. The conditional velocity satisfies its own continuity equation:

$$\partial_t p_t(x \mid z) + \nabla \cdot \big(p_t(x \mid z) u_t(x \mid z)\big) = 0.$$

Mixing the conditional paths recovers the marginal path,

$$p_t(x) = \int p_t(x \mid z) p(z)\, dz,$$

and the marginal velocity is the posterior conditional average already shown in Section 3.1.3. The conditional velocity is the accessible training label; the marginal velocity is the shared object represented by the network.

Figure 5 summarizes these objects on a two-dimensional toy problem. It separates three ideas that will recur throughout the paper: local velocity regression averages over hidden path identities, conditional paths mix into a marginal field, and the learned field becomes a population prediction only after ODE sampling.

3.2.2 The Conditional Loss and Gradient Equivalence

The trainable CFM loss samples the conditional variable first and then samples a point along the corresponding conditional path:

$$t \sim \mathrm{Unif}(0,1), \qquad z \sim p(z), \qquad x \sim p_t(\cdot \mid z).$$

The network is trained against the conditional velocity:

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t,z,x}\left[\|v_\theta(x,t) - u_t(x \mid z)\|^2\right].$$

Every quantity in this objective is available at training time once the conditional path family is chosen.

The reason this objective trains the same marginal field is the orthogonal decomposition of

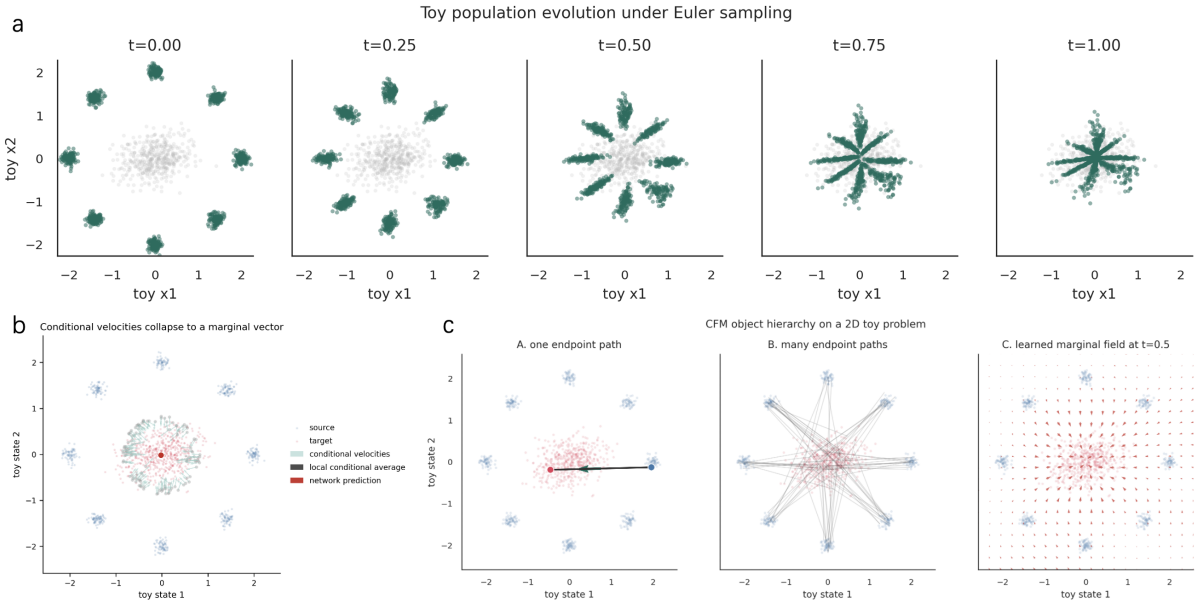

Figure 5: Conceptual anatomy of Flow Matching and Conditional Flow Matching on a two-dimensional toy problem. (a) After a velocity field is learned, Euler sampling pushes a source population through normalized model time; these are generated solver states, not observed intermediate snapshots. (b) Conditional velocity labels from different hidden path identities collapse to a local posterior average because the network does not observe Z. (c) A fixed endpoint path gives one conditional velocity label, many endpoint paths define the marginal intermediate population, and the learned neural field represents the marginal velocity.

squared loss. Since $v_\theta(x,t)$ does not observe the hidden Z, the conditional expectation

$$u_t(x) = \mathbb{E}[u_t(X_t \mid Z) \mid t, X_t = x]$$

is the best shared prediction under mean-squared error. Equivalently,

$$\mathbb{E}\|v_\theta(X_t,t) - u_t(X_t \mid Z)\|^2 = \mathbb{E}\|v_\theta(X_t,t) - u_t(X_t)\|^2 + \mathbb{E}\|u_t(X_t) - u_t(X_t \mid Z)\|^2 .$$

The first term is the ideal Flow Matching loss, and the second term depends only on the conditional path family, not on θ. Therefore

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathcal{L}_{\mathrm{FM}}(\theta) + C, \qquad \nabla_\theta \mathcal{L}_{\mathrm{CFM}}(\theta) = \nabla_\theta \mathcal{L}_{\mathrm{FM}}(\theta).$$

the supplementary material gives the full derivation. The important point is conceptual: CFM is not a heuristic endpoint shortcut. For a fixed conditional path family, it provides an executable estimator of the same marginal velocity-regression objective.

3.2.3 A Minimal Instance of Endpoint-Pair Linear CFM

The simplest CFM instance uses endpoint pairs. Let

$$(x_0, x_1) \sim \pi, \qquad t \sim \mathrm{Unif}(0,1),$$

Table 5: Objects in endpoint-pair Conditional Flow Matching. The first four rows are modeling choices; the last two are algorithmic execution steps.

Object	Determined by	Role
State space $\mathcal{X}$	preprocessing / representation choice	geometry for distances, paths, and velocities
Coupling π	pairing or transport strategy	endpoint-pair distribution
Conditional path $p_t(\cdot \mid z)$	path design	intermediate-state construction
Conditional velocity $u_t(\cdot \mid z)$	analytic derivative	local regression target
Velocity field $v_\theta(x, t, c)$	learned network	shared marginal vector field
ODE solver	inference-time choice	turns local velocities into a pushforward map

where π is a coupling between the source and target snapshots. In this section we use the simplest independent pairing. Section 4 studies how changing π, for example through optimal transport, changes the learned field.

For a fixed endpoint pair, take the conditional path to be the straight line

$$x_t = (1-t)x_0 + tx_1.$$

The corresponding conditional velocity is constant:

$$u_t(x_t \mid x_0, x_1) = x_1 - x_0.$$

Substituting this path into the CFM objective yields

$$\mathcal{L}_{\text{pair-CFM}}(\theta) = \mathbb{E}_{(x_0,x_1)\sim\pi,\, t\sim\text{Unif}(0,1)} \left[\left\| v_\theta((1-t)x_0 + tx_1, t) - (x_1 - x_0) \right\|^2 \right].$$

The training loop has three operations:

$$\text{sample pair} \longrightarrow \text{interpolate} \longrightarrow \text{regress velocity}.$$

We caution against reading the gray chords in Figure 6 as observed lineage. They are conditional training constructions, produced by independent endpoint pairing in the 20D PC space and then displayed in PHATE for visualization. Many displayed chords cross branches or connect distant regions. This shows that even the simplest endpoint-pair CFM already contains a modeling choice through the coupling π. Section 4 replaces this baseline coupling with OT, minibatch OT, and reflow-style alternatives.

The ingredients introduced so far can be summarized as follows.

3.3 The Full Pipeline of Network, Training, and Sampling

A minimal CFM implementation has three components: a velocity network, a local-regression training loop, and an ODE sampler. The same velocity field is used in both stages; only the computational role changes.

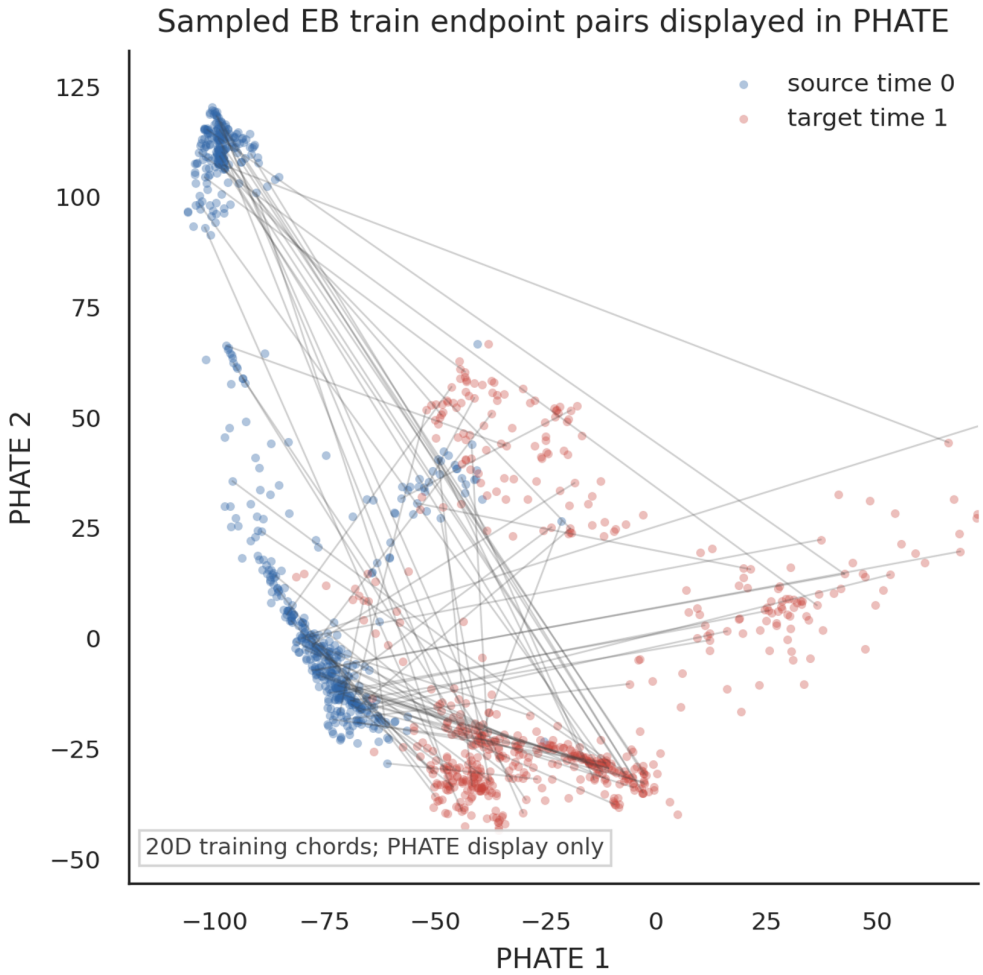

Figure 6: Endpoint-pair CFM construction on the EB example. Source and target cells are paired in the 20-dimensional PC training space, while PHATE is used only to display the sampled endpoint pairs. Each gray segment connects a sampled pair in the display; the actual interpolation $x_t = (1-t)x_0 + tx_1$ and regression label $x_1 - x_0$ are defined in PC space, not in PHATE coordinates.

3.3.1 The Velocity Network

In code, the velocity network has the interface

$$v_\theta : \mathbb{R}^d \times [0,1] \times \mathcal{C} \to \mathbb{R}^d, \qquad (x_t, t, c) \mapsto v_\theta(x_t, t, c).$$

The input state x_t may be gene-expression coordinates, PCA scores, a learned latent representation, or another fixed state-space representation. The output must have the same dimension as the state:

$$\text{shape}(v_\theta(x_t, t, c)) = \text{shape}(x_t).$$

This is neither a classifier nor an endpoint predictor. It is an instantaneous velocity field.

The simplest baseline concatenates state and time and applies an MLP,

$$v_\theta(x_t, t) = \text{MLP}_\theta([x_t, t]).$$

Algorithm 1 Endpoint-pair CFM training

Require: Pair sampler π, velocity network v_θ, learning rate η, batch size B

for each training step **do**

  Sample endpoint pairs $\{(x_0^{(i)}, x_1^{(i)}, c^{(i)})\}_{i=1}^{B} \sim \pi$

  Sample $s^{(i)} \sim \mathrm{Unif}(0,1)$

  Set $x_s^{(i)} = (1 - s^{(i)})x_0^{(i)} + s^{(i)}x_1^{(i)}$

  Set $u_s^{(i)} = x_1^{(i)} - x_0^{(i)}$

  Compute

$$\mathcal{L} = \frac{1}{B}\sum_{i=1}^{B}\left\| v_\theta(x_s^{(i)}, s^{(i)}, c^{(i)}) - u_s^{(i)} \right\|^2$$

  Update $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}$

end for

When experimental conditions matter, the same interface extends to

$$z_t = \big[E_x(x_t), E_t(t), E_c(c)\big], \qquad v_\theta(x_t, t, c) = g_\theta(z_t),$$

where E_t may be a time embedding and E_c may encode perturbation, dose, batch, or donor. This conditional form is the natural bridge from a single source-target example to perturbation and multi-context dynamics in Section 5.1.

The capacity ablation is treated as an appendix diagnostic. Tiny networks underfit, while the medium VelocityMLP (a four-hidden-layer MLP with 256 units per layer and 208,148 parameters) achieves the lowest validation MSE and endpoint MMD in this run. Increasing to a larger model does not consistently enhance endpoint quality. Consequently, we opt for a medium network as a sufficient implementation choice, avoiding a diversion into architecture search.

3.3.2 The Training Loop

For endpoint-pair linear CFM, a mini-batch step is a direct implementation of the loss in Section 3.2.3. Algorithm 1 summarizes the resulting local-regression loop.

There is no ODE rollout, no likelihood integration, and no discriminator in this loop. In the full EB run used in this section, the CFM model was trained in a 20-dimensional PC state space from time 0 to time 1, using 1,440 source and 1,440 target training cells, with a held-out validation set of 360 cells per endpoint. The PHATE embedding is used only for visualization; it is not used for training or metric computation. The main VelocityMLP has 20-dimensional input and output states, four hidden layers of width 256, and 208,148 trainable parameters. Over 10,000 local-regression steps, the training loss decreases from 13.06 to 3.20; the fixed-grid validation velocity MSE drops quickly, then drifts upward and ends near 5.15. This is only a local velocity-regression diagnostic, not endpoint likelihood or endpoint distribution fit.

The cost diagnostic makes the computational distinction concrete. With the same 600 optimization steps, local CFM training takes about 1.5 seconds and uses one network evaluation per step, whereas the adjoint endpoint baseline takes about 53.46 seconds and averages 101.4 right-hand-side evaluations per step. This shows that once an ODE solve is placed inside every update, the solver dominates the training cost.

Algorithm 2 Fixed-step ODE sampling

Require: Trained velocity field v_θ, source state x_0, condition c, number of steps K, solver type

Set $\Delta t = 1/K$
for $k = 0, \ldots, K-1$ **do**
  Set $t_k = k/K$
  if solver type is Euler **then**
    $x_{k+1} \leftarrow x_k + \Delta t\, v_\theta(x_k, t_k, c)$
  else if solver type is midpoint **then**
    $x_{k+\frac{1}{2}} \leftarrow x_k + \frac{\Delta t}{2} v_\theta(x_k, t_k, c)$
    $x_{k+1} \leftarrow x_k + \Delta t\, v_\theta(x_{k+\frac{1}{2}}, t_k + \frac{\Delta t}{2}, c)$
  end if
end for

Ensure: Predicted endpoint x_K

3.3.3 Sampling and the Flow Map

After training, the learned field defines an ODE

$$\frac{dX_t}{dt} = v_\theta(X_t, t, c), \qquad X_0 \sim \widehat{\mu}_{0,c}.$$

For single-cell snapshot modeling, the initial distribution is usually a real observed source population, not Gaussian noise. The learned flow map is

$$\Phi_\theta^{0\to 1}(x_0) = x_0 + \int_0^1 v_\theta(X_t, t, c)\, dt, \qquad \widehat{\mu}_{1,c}^{\,\theta} = (\Phi_\theta^{0\to 1})_{\#}\widehat{\mu}_{0,c}.$$

Thus the local velocity field becomes a population prediction only after ODE integration.

Real experimental time and model time should also be separated. If a biological segment runs from real time s_i to s_j, the model often uses normalized time

$$\tau = \frac{s - s_i}{s_j - s_i}.$$

The real-time velocity rescales as

$$\frac{dX_s}{ds} = \frac{1}{s_j - s_i} v_\theta(X_s, \tau, c).$$

The internal time variable τ is relative progress through a chosen bridge, not automatically an hour, day, or cell-cycle unit.

Algorithm 2 shows two fixed-step solvers used to turn the learned velocity field into predicted cell states. Adaptive embedded Runge–Kutta solvers such as Dormand–Prince choose step sizes to satisfy error tolerances and report a data-dependent number of function evaluations (Dormand and Prince, 1980).

In the EB example below, Euler sampling is performed in the 20D PC state space with normalized model time. The intermediate panels are generated solver states along the rollout, not observed intermediate biological snapshots.

Figure 7 connects local velocity regression to population sampling. Starting from the ob-

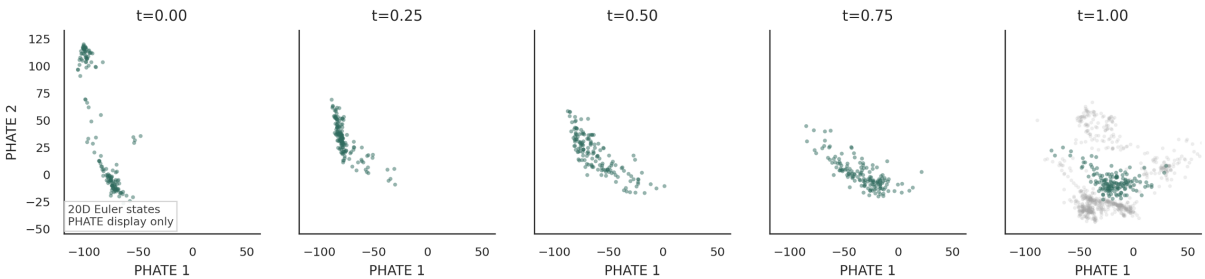

Figure 7: Generated EB population rollout after Euler integration of the learned velocity field. The ODE is solved in the 20D PC state space from observed source cells at normalized model time $t = 0$. Panels show selected generated solver states along the rollout; the gray points in the final panel denote the observed target snapshot. PHATE is used only for visualization, not for training, integration, or metric computation.

served source snapshot, Euler integration moves the generated 20D states toward the region occupied by the observed target after projection to PHATE. The final generated cloud overlaps part of the target distribution but does not cover every target structure and still shows local displacement. Because this is a PHATE-projected display, the figure serves as a qualitative check of the sampling direction; concrete endpoint quality still needs to be judged by metrics in the 20D PC space.

3.3.4 Solver Error and Endpoint-Fit Error

Sampling introduces a numerical question that is separate from the modeling question. Let D be a distributional discrepancy such as MMD or sliced W_2 (defined in the supplementary material). For a fixed trained velocity field and a solver setting a, solver error compares the generated endpoint with a high-accuracy numerical reference:

$$\mathrm{err}_{\mathrm{solver}}(a) = D\Big(\widehat{\mu}_{1,c}^{(a)}, \widehat{\mu}_{1,c}^{\mathrm{ref}}\Big) .$$

Endpoint-fit error instead compares the generated endpoint with the observed target snapshot:

$$\mathrm{err}_{\mathrm{fit}}(a) = D\Big(\widehat{\mu}_{1,c}^{(a)}, \widehat{\mu}_{1,c}^{\mathrm{obs}}\Big) .$$

These two errors have different causes. Solver error can be reduced by adding steps or using a higher-order method. Endpoint-fit error reflects the learned field and the modeling assumptions behind it: coupling, path family, state space, conditioning, architecture, and optimization. More solver steps cannot repair a poor biological modeling choice.

Figure 8 shows the distinction. In panel (a), the Euler endpoint cloud changes substantially at very small step counts and then stabilizes as K increases. This visual stabilization reflects reduced numerical sensitivity. But the stabilized endpoint still misses part of the observed target structure. This gap is a modeling and endpoint-fit issue, and extra solver effort alone will not close it. Panel (b) makes the same point quantitatively: increasing NFE beyond a modest solver effort does not monotonically improve MMD to the observed target, and midpoint or adaptive integration does not by itself repair the endpoint mismatch. Once numerical integration is

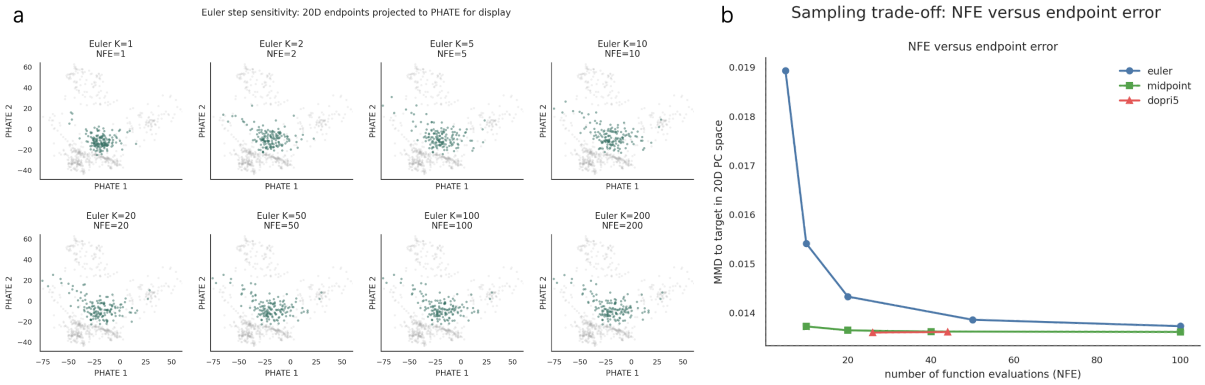

Figure 8: Sampling diagnostics for the trained EB 20D PC velocity field. (a) Euler step sensitivity: generated endpoints in the 20D PC state space, projected to PHATE; gray points mark the observed target. Increasing K stabilizes the generated endpoint cloud, but the stabilized cloud still covers only part of the target structure. (b) Endpoint-fit MMD to the observed target in 20D PC space as a function of the number of function evaluations (NFE).

accurate enough for the learned field, the remaining endpoint behavior is dominated by the velocity field and by the assumptions used to construct the training targets: coupling, path geometry, state space, and conditioning.

Flow Matching localizes training to per-point regression. Numerical integration is deferred to inference. Its velocity labels still depend on modeling assumptions about endpoint pairing, representation space, path geometry, and biological constraints—the subject of Section 4. If endpoint mismatch stays flat as NFE grows, the remaining error is in the assumptions, and more solver steps will not help.

4 Couplings, State Spaces, and Biological Assumptions

The previous sections treated the Flow Matching pipeline as a fixed recipe. Given a pair of endpoints (x_0, x_1), the model trains on a linear bridge $x_t = (1-t)x_0 + tx_1$ with velocity supervision $u_t = x_1 - x_0$. This section opens three assumptions that the recipe depends on. The coupling decides which cells get paired as endpoints. The state space decides what straight and near mean. Biological constraints decide whether the flow must account for proliferation, stochasticity, or prior structure.

But we already know that in real single-cell snapshot data, we observe population distributions at different time points. We may have a source distribution μ_0 and a target distribution μ_1, but the data do not record which early cell evolved into which later cell. The CFM objective is complete only after we specify the endpoint coupling:

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{(x_0,x_1)\sim\pi,\, t\sim U(0,1)} \left[\|v_\theta(x_t, t) - (x_1 - x_0)\|^2 \right].$$

Here v_θ is the learned velocity field, t is normalized bridge time, and x_t is the linearly interpolated intermediate state. The critical object is π: the joint distribution over endpoint pairs. Its two marginals must equal the observed source and target distributions, but the data do not directly tell us how individuals from the two marginals should be connected.

Three choices remain open.

1. **Coupling.** Who pairs with whom? The endpoint joint distribution $\pi \in \Pi(\mu_0, \mu_1)$ determines which velocity targets the network sees.

2. **State space.** In which coordinate system do we define cell states, compute distances, and construct straight lines? The representation space determines the meaning of cost, coupling, and "straight line."

3. **Biological assumptions.** Does the flow merely redistribute normalized probability mass, or does it describe proliferation, death, stochasticity, and biological priors such as RNA velocity, markers, lineage, or gene regulatory network (GRN) constraints?

The dependency chain is

$$r \longrightarrow C_r \longrightarrow \pi_r \longrightarrow (x_0, x_1) \longrightarrow (x_t, u_t) \longrightarrow v_\theta \longrightarrow \text{claim}.$$

The map r sends raw sequencing observations to model states, for example PCA coordinates, a VAE latent space, or gene-program scores. Once r is fixed, it defines a cost matrix C_r, which guides a coupling π_r. The coupling determines sampled endpoints; the endpoints determine intermediate states and velocity labels; the neural network fits the conditional mean of those labels at the population level.

The core message is clear: the meaning of a learned flow is relative to its coupling, state space, and biological assumptions. Altering the coupling affects velocity supervision; modifying the representation influences cost and straight lines; and introducing growth, stochasticity, or priors changes the supported biological claims.

4.1 Coupling Shapes the Velocity Field

Every CFM training step requires an endpoint pair (x_0, x_1). Single-cell snapshots provide the marginals μ_0 and μ_1, but not the cross-time correspondence. Any joint distribution $\pi \in \Pi(\mu_0, \mu_1)$ satisfying the marginal constraints is a valid endpoint sampler. This choice matters because, in the infinite-sample population limit, the squared-loss optimum for a fixed coupling satisfies

$$v^\star(x, t) = \mathbb{E}[X_1 - X_0 \mid X_t = x], \qquad X_t = (1-t)X_0 + tX_1, \quad (X_0, X_1) \sim \pi.$$

The network can output only one vector at a given (x, t), so it learns the conditional mean of all endpoint velocities whose bridges pass through that point. Different couplings change which endpoint pairs generate samples, which conditional bridges meet locally, and which target velocities are averaged.

4.1.1 Independent Coupling: The Simplest Baseline

The simplest endpoint sampler is the independent product coupling:

$$\pi_{\text{ind}} = \mu_0 \otimes \mu_1.$$

This means that $X_0 \sim \mu_0$ and $X_1 \sim \mu_1$ are sampled independently and then randomly paired. The procedure assumes no prior geometric association between source and target cells. It is easy to implement: at each training step, sample a source batch and a target batch, then combine endpoints. The cost is that it ignores the physical and geometric intuition that nearby states are more likely to be connected.

Suppose the source snapshot contains early progenitor-like cells and the target snapshot contains two terminal branches. Under independent coupling, each source cell can be connected with equal probability to all target cells, including states that are far away or biologically belong to another branch. This is mathematically legal because the target marginal remains μ_1, but the geometry of the supervision can be poor.

One way to quantify this is the average chord length:

$$\mathbb{E}_{\pi_{\text{ind}}}\left[\|X_1 - X_0\|^2\right].$$

If the source and target distributions contain clusters or branching structure, blind pairing creates many long and crossing chords. The conditional-mean formula then becomes a warning:

$$v^\star(x,t) = \mathbb{E}[X_1 - X_0 \mid X_t = x].$$

If many bridges with different or opposite directions pass near the same intermediate point, the right-hand side averages conflicting velocity signals. The resulting vector may not correspond to any clear biological trajectory.

Independent coupling serves as a reference. It shows how far a continuous flow model can go using only endpoint marginals, with no distance, lineage, or local-continuity prior. When the data contain strong geometry, the baseline exposes long-distance chords, crossing paths, and local velocity averaging. Adding a pairing prior addresses this; we compare it directly with OT-CFM below.

4.1.2 OT Coupling Improves Pairing with Geometric Information

OT-CFM keeps the CFM loss unchanged and replaces the endpoint sampler with a geometry-aware coupling (Tong et al., 2024a). For empirical source and target snapshots,

$$\widehat{\mu}_0 = \sum_{i=1}^{n_0} a_i \delta_{x_i^0}, \qquad \widehat{\mu}_1 = \sum_{j=1}^{n_1} b_j \delta_{x_j^1},$$

we define a cost matrix in the chosen representation space:

$$C_{ij} = c(x_i^0, x_j^1) = \|x_i^0 - x_j^1\|^2.$$

Here x_i^0 denotes the i-th source cell state, x_j^1 denotes the j-th target cell state, and a_i, b_j are their empirical weights. This cost matrix is a mathematical quantity induced by the selected state representation and distance metric.

An OT coupling is a nonnegative matrix $\pi = (\pi_{ij})$, where π_{ij} denotes how much probability

mass is assigned from source cell i to target cell j. Feasible couplings satisfy

$$U(a,b) = \left\{\pi \geq 0 : \pi\mathbf{1} = a,\ \pi^\top \mathbf{1} = b\right\}.$$

In practice, a smooth entropic OT plan is often used:

$$\pi_\varepsilon^\star = \arg\min_{\pi \in U(a,b)} \left[\langle C, \pi\rangle + \varepsilon \sum_{ij} \pi_{ij}(\log \pi_{ij} - 1)\right].$$

The term $\langle C, \pi\rangle$ favors low-cost, geometrically nearby endpoint pairs; the entropy term controls how diffuse the plan is. Small ε produces sharper, more local pairings; large ε makes the plan more diffuse and closer to independent coupling. The regularized problem is usually solved by Sinkhorn iterations (Cuturi, 2013). We write $\pi_{\mathrm{OT}} := \pi_\varepsilon^\star$ for this geometry-aware endpoint sampler.

After computing $\pi_\varepsilon^\star$, OT-CFM uses the same velocity regression interface as ordinary CFM. At each training step, sample (i, j) from the plan, set

$$x_0 = x_i^0, \qquad x_1 = x_j^1,$$

construct $x_t = (1-t)x_0 + tx_1$, and train against $x_1 - x_0$:

$$\mathcal{L}_{\text{OT-CFM}}(\theta) = \mathbb{E}_{(x_0,x_1)\sim\pi_{\mathrm{OT}},\, t\sim U(0,1)} \left[\|v_\theta(x_t, t) - (x_1 - x_0)\|^2\right].$$

The only structural change is the sampling source for (x_0, x_1). The optimized object remains the network parameter θ.

Algorithm 3 writes this as the same local-regression loop as Algorithm 1, with the independent pair sampler replaced by an entropic OT plan. The displayed version uses a full empirical plan for clarity; the minibatch OT-CFM implementation computes the same Sinkhorn step on each sampled mini-batch.

We prefer this construction because it connects directly to dynamic OT. Under squared Euclidean cost, if $\pi^\star$ is the true W_2 optimal coupling, then

$$X_t = (1-t)X_0 + tX_1, \qquad (X_0, X_1) \sim \pi^\star$$

induces displacement interpolation in Wasserstein geometry. The Benamou–Brenier dynamic formulation identifies this marginal path with a minimum-action transport path between endpoint distributions (Benamou and Brenier, 2000). OT-CFM does more than shorten visual chords. It lifts static OT geometry into a learnable continuous velocity field inside a simulation-free CFM training interface. Figure 9 visualizes this endpoint-level effect of replacing independent pairing with the geometry-aware sampler π_{OT}.

4.1.3 Rectified Flow from Endpoint Pairing to Path Geometry Optimization

Rectified Flow exposes the same geometry from a different angle (Liu et al., 2023). OT-CFM starts with optimal transport and uses low-cost coupling to define endpoint pairs. Rectified Flow

Algorithm 3 OT-CFM endpoint-pair training

Require: Source cells $\{x_i^0\}_{i=1}^{n_0}$, target cells $\{x_j^1\}_{j=1}^{n_1}$, weights a, b, entropic parameter ε, velocity network v_θ, learning rate η, batch size B

Compute $C_{ij} = \|x_i^0 - x_j^1\|^2$ in the chosen representation space
Compute $\pi_{\mathrm{OT}} \leftarrow \mathrm{Sinkhorn}(C, a, b, \varepsilon)$
for each training step **do**
  Sample endpoint index pairs $\{(i_m, j_m)\}_{m=1}^{B} \sim \pi_{\mathrm{OT}}$
  Set $x_0^{(m)} = x_{i_m}^0$ and $x_1^{(m)} = x_{j_m}^1$
  Sample $s^{(m)} \sim \mathrm{Unif}(0,1)$
  Set $x_s^{(m)} = (1 - s^{(m)})x_0^{(m)} + s^{(m)}x_1^{(m)}$
  Set $u_s^{(m)} = x_1^{(m)} - x_0^{(m)}$
  Compute

$$\mathcal{L} = \frac{1}{B}\sum_{m=1}^{B}\left\|v_\theta(x_s^{(m)}, s^{(m)}) - u_s^{(m)}\right\|^2$$

  Update $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}$
end for

Ensure: Trained velocity field v_θ

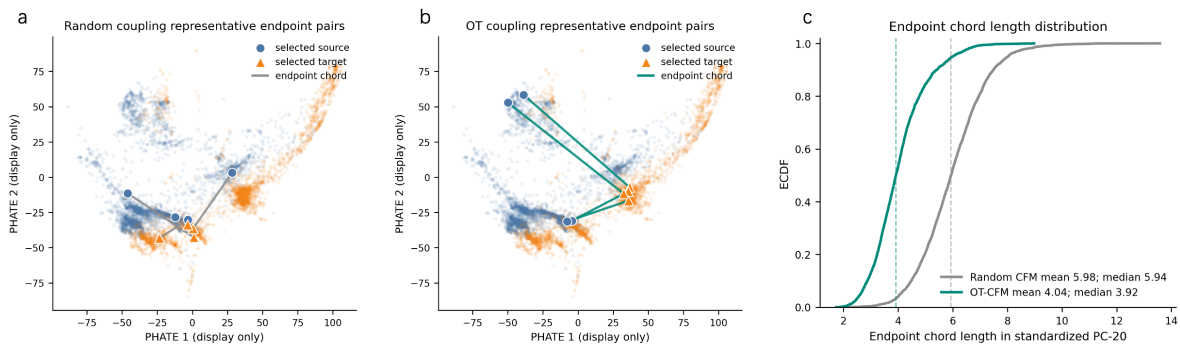

Figure 9: Endpoint coupling changes the endpoint pairs, and with them the straight-line CFM velocity labels, before training. (a) Random CFM pairs EB source and target snapshots independently of geometry, producing long, crossing supervision chords. (b) OT-CFM samples endpoint pairs from the geometry-aware coupling π_{OT} computed in standardized PC-20 space, yielding shorter local chords between source and target cells. (c) The endpoint chord-length ECDF, measured in standardized PC-20, shows that OT-CFM shifts supervision toward shorter endpoint distances, reducing the mean chord length from 5.98 for Random CFM to 4.04 for OT-CFM. PHATE coordinates in (a,b) are for visualization only; all chord lengths and coupling distances are computed in standardized PC-20.

starts from a more general coupling and asks how to make the learned ODE flow straighter and easier to integrate.

For any coupling $\pi \in \Pi(\mu_0, \mu_1)$, Rectified Flow uses

$$X_t = (1-t)X_0 + tX_1, \qquad (X_0, X_1) \sim \pi,$$

and regresses the corresponding straight-line velocity:

$$\mathcal{L}_{\mathrm{RF}}(\theta) = \int_0^1 \mathbb{E}_{(X_0,X_1)\sim\pi}\left[\|v_\theta(X_t,t) - (X_1 - X_0)\|^2\right] dt.$$

Algorithm 4 Rectified Flow with optional reflow

Require: Initial pair sampler $\pi^{(0)}$, source distribution $\widehat{\mu}_0$, velocity networks $v_{\theta^{(r)}}$, learning rate η, batch size B, reflow rounds R, ODE solver

for $r = 0, \ldots, R$ **do**
  for each training step in round r **do**
    Sample endpoint pairs $\{(x_0^{(m)}, x_1^{(m)})\}_{m=1}^{B} \sim \pi^{(r)}$
    Sample $s^{(m)} \sim \mathrm{Unif}(0,1)$
    Set $x_s^{(m)} = (1 - s^{(m)})x_0^{(m)} + s^{(m)}x_1^{(m)}$
    Set $u_s^{(m)} = x_1^{(m)} - x_0^{(m)}$
    Compute

$$\mathcal{L}^{(r)} = \frac{1}{B}\sum_{m=1}^{B}\left\| v_{\theta^{(r)}}(x_s^{(m)}, s^{(m)}) - u_s^{(m)} \right\|^2$$

    Update $\theta^{(r)} \leftarrow \theta^{(r)} - \eta \nabla_{\theta^{(r)}} \mathcal{L}^{(r)}$
  end for
  if $r < R$ **then**
    Sample source states $\{\widetilde{x}_0^{(m)}\}_{m=1}^{B} \sim \widehat{\mu}_0$
    Integrate $\dot{z}_t = v_{\theta^{(r)}}(z_t, t)$ from $t = 0$ to $t = 1$ with $z_0 = \widetilde{x}_0^{(m)}$
    Set $\widetilde{x}_1^{(m)} = z_1^{(m)}$
    Define $\pi^{(r+1)}$ as the empirical sampler over $\{(\widetilde{x}_0^{(m)}, \widetilde{x}_1^{(m)})\}$
  end if
end for

Ensure: Final velocity field $v_{\theta^{(R)}}$

The population optimum is again

$$v^\star(x,t) = \mathbb{E}[X_1 - X_0 \mid X_t = x]\,.$$

This formula explains the problem: training bridges may cross and may assign different directions to the same region, but inference uses a shared, single-valued ODE velocity field.

Rectification can be understood as path straightening. Train a first velocity field $v^{(0)}$, sample $Z_0 \sim \mu_0$, and integrate

$$\frac{dZ_t}{dt} = v^{(0)}(Z_t, t), \qquad Z_1 = \Phi_1^{(0)}(Z_0).$$

The model induces new endpoint pairs

$$(Z_0, \Phi_1^{(0)}(Z_0)).$$

Reflow trains the next model on these induced pairs with the same straight-line regression. The procedure re-expresses a crossing or curved transport process as a field that is closer to straight and friendlier to coarse-step solvers such as Euler. The diagnostics below separate this solver-geometry role from endpoint-fit and biological-lineage claims.

Algorithm 4 summarizes the recursive version used for reflow diagnostics. Round $r = 0$ trains from the initial endpoint sampler, and each later round trains on endpoint pairs induced by integrating the previous learned field.

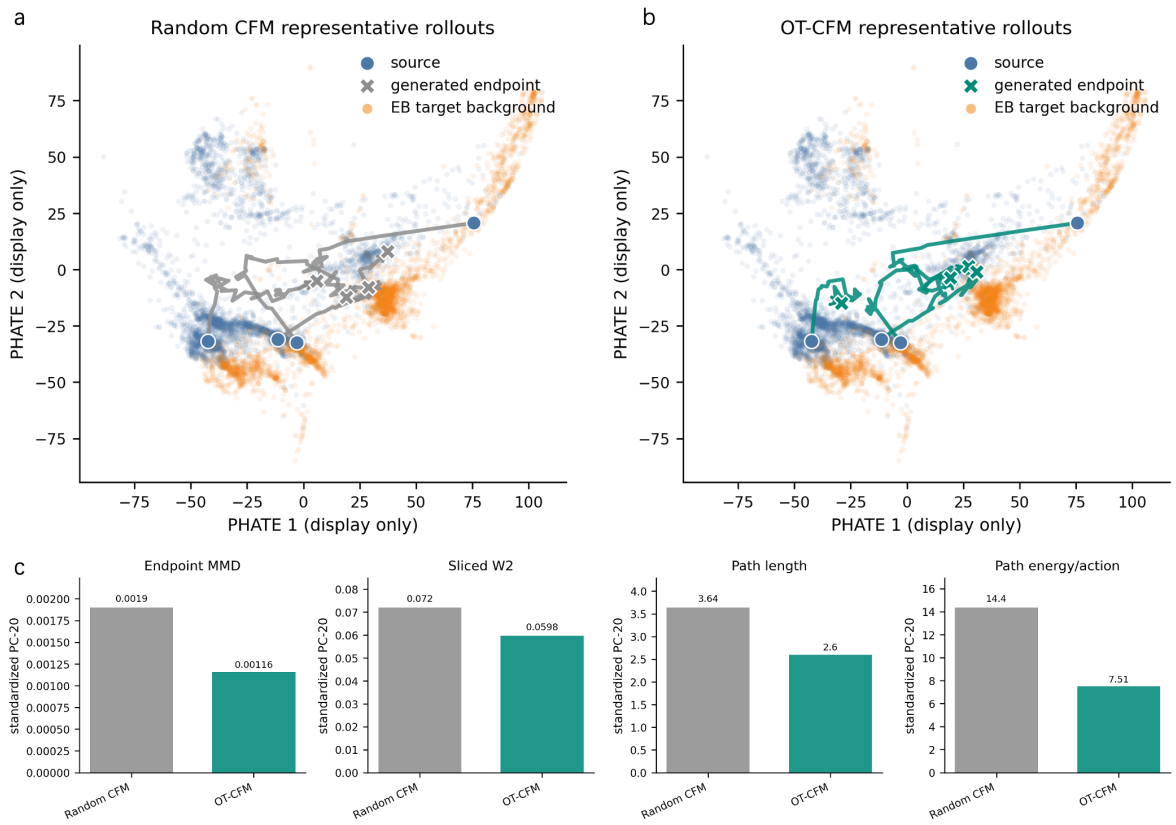

Figure 10: Random CFM versus OT-CFM rollouts and diagnostics on the EB bridge. (a) Random CFM rollouts from matched EB source cells produce dispersed, irregular generated endpoints. (b) OT-CFM rollouts from the same sources produce more localized endpoints and smoother paths. (c) Endpoint and path diagnostics in standardized PC-20 under Random CFM and OT-CFM.

4.1.4 Coupling Diagnostics and Discussion

OT-CFM converts static OT geometry into a more local, lower-energy velocity field with better endpoint matching. Table 6 compares Random CFM, OT-CFM, and two reflow rounds on the same EB bridge. The pair-level change is visible first. Mean endpoint chord length drops from 5.98 to 4.04. This reaches the learned flow. Endpoint MMD drops from 0.00190 to 0.00116, sliced W_2 from 0.0720 to 0.0598, path length from 3.64 to 2.60, and action from 14.35 to 7.51. Figure 10 shows representative rollouts and the main endpoint and path diagnostics.

Table 6: Path-geometry diagnostics on the EB bridge. Metrics are evaluated in standardized PC-20 unless otherwise noted.

Method	MMD	Sliced W_2	Length	Action	Straightness	Tortuosity	kNN off-manifold	NFE 4 error
Random CFM	0.00190	0.0720	3.64	14.35	5.58	0.363	2.44	0.992
OT-CFM	0.00116	0.0598	2.60	7.51	0.448	0.032	2.49	0.312
Reflow 1	0.00160	0.0656	2.48	6.80	0.047	0.003	2.55	0.124
Reflow 2	0.00201	0.0714	2.45	6.62	0.033	0.002	2.55	0.103

Reflow should be read differently. It does not improve endpoint fit beyond OT-CFM, but it dramatically improves straightness and few-step stability. In the PC-20 diagnostics, straight-

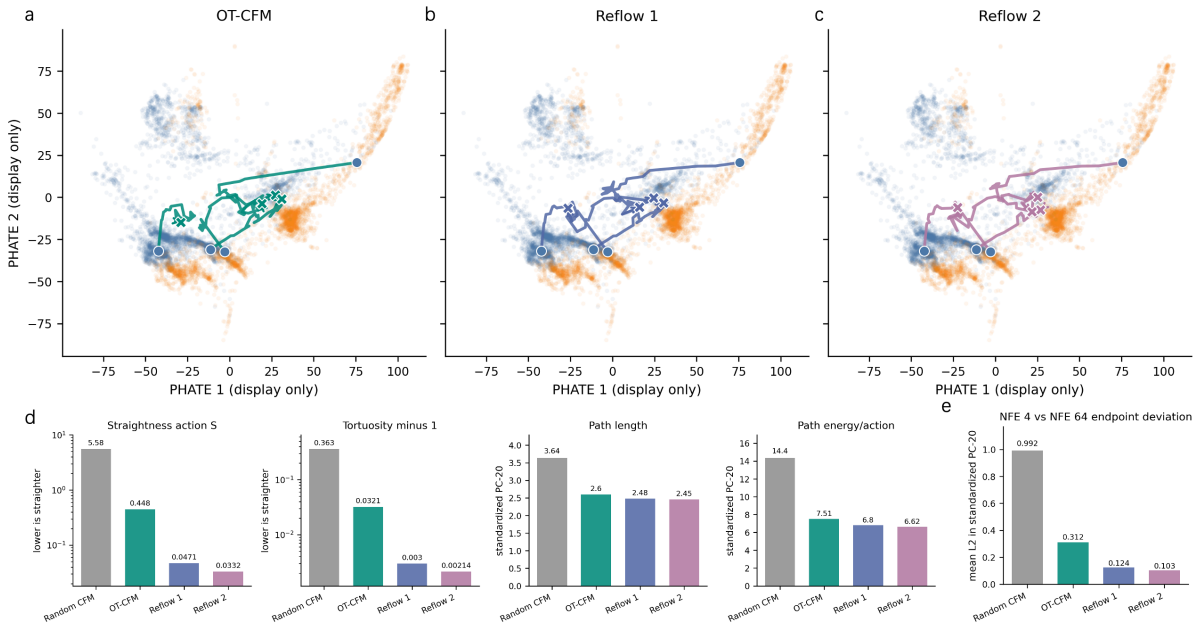

Figure 11: Path geometry under OT-CFM and two reflow rounds on the EB bridge. (a–c) Representative rollouts under OT-CFM, Reflow 1, and Reflow 2, displayed in PHATE for visualization only. (d) Path-geometry diagnostics in standardized PC-20 for Random CFM, OT-CFM, Reflow 1, and Reflow 2: straightness action, tortuosity minus 1, path length, and kinetic action. (e) Few-step solver stability measured as the mean PC-20 L^2 deviation between NFE 4 and NFE 64 endpoints.

ness action drops from 0.448 for OT-CFM to 0.047 and 0.033 after one and two reflow rounds, while tortuosity drops from 0.0321 to 0.0030 and 0.00214. Path length and action decrease more modestly after OT-CFM. The NFE= 4 coarse-step error relative to NFE= 64 drops from 0.312 to 0.124 and 0.103. Figure 11 visualizes this distinction: representative PHATE trajectories are useful for intuition, but the PC-20 straightness and few-step stability panels carry the quantitative claim.

Several limitations are equally important, and one deserves emphasis because it is easy to overlook. Off-manifold kNN distance does not improve with OT-CFM or Reflow: the four values are 2.44, 2.49, 2.55, and 2.55. A very straight trajectory can still pass through regions where no real cell has ever been observed. Low action or high straightness cannot be directly equated with a biological path closer to the data manifold. Controlled toy branching also shows that OT coupling reduces target fate mass error (0.040 versus 0.0575) and rare-fate mass error (0.020 versus 0.0288), but does not reduce branch leakage (0.0975 versus 0.0938).

OT-CFM is a direct geometric coupling choice for single-cell snapshots, while Reflow is a complementary view focused on ODE straightness and few-step integration. Both remain model diagnostics. In our EB runs Reflow mainly buys straightness and few-step stability, and endpoint fit barely moves. All results in this subsection occur in PC-20 training space. Change the state space, and low cost, straight line, and off-manifold all change.

4.2 State Representation and the Geometry of Flow Matching

Coupling changes CFM velocity supervision. The same loss, architecture, and linear interpolation produce different supervised velocities $u_t = x_1 - x_0$ when endpoint pairing changes. The choice of state space is equally critical: in which space is the cost for low-cost coupling computed? If the space changes, so do nearness, distance, straight lines, and velocity directions. State representation is not ordinary preprocessing; it is part of the Flow Matching learning problem.

4.2.1 The Representation Function Defines the Supervision Problem

Single-cell sequencing gives sparse cell-by-gene count vectors plus metadata such as batch, donor, condition, time point, and library size. Flow Matching first chooses a representation map

$$x_i = r(y_i, m_i) \in \mathcal{X}_r.$$

Here y_i is the molecular observation, m_i is metadata, r is the representation function, and $\mathcal{X}_r$ is the induced state space. It may be PCA scores, VAE or scVI latents, gene-program scores, or a latent space learned by a multimodal or graph model. Choosing r means choosing the geometric space of Flow Matching.

In that space, the empirical marginal at time t and condition c is

$$\widehat{\mu}_{t,c}^{(r)} = \frac{1}{n_{t,c}} \sum_{i:\, t_i = t,\, c_i = c} \delta_{r(y_i, m_i)}.$$

The model sees empirical distributions in representation space. Raw count space never enters the learning problem. All later objects inherit this choice:

$$r \implies \mathcal{X}_r \implies C_r \implies \pi_r \implies \gamma_r(t; x_0, x_1) \implies u_r(t; x_0, x_1) \implies v_\theta.$$

First, r defines distances:

$$c_r(i, j) = \|r(y_i, m_i) - r(y_j, m_j)\|^2.$$

Second, distance defines OT or nearest-neighbor coupling:

$$\pi_r^\star = \arg\min_{\pi \in U(a,b)} \left[\langle C_r, \pi \rangle + \varepsilon H(\pi)\right].$$

Third, the coupling defines sampled endpoint pairs. Fourth, the representation defines the bridge itself:

$$\gamma_r(t; x_0, x_1) = (1-t)x_0 + t x_1, \qquad u_r(t; x_0, x_1) = x_1 - x_0.$$

This line is a Euclidean chord in $\mathcal{X}_r$ and carries no biological trajectory meaning on its own. Under squared loss, the learned field is

$$v_r^\star(x, t) = \mathbb{E}_{\pi_r}\left[u_r(t; X_0, X_1) \mid \gamma_r(t; X_0, X_1) = x\right].$$

The same cells and the same CFM objective can define different supervision distributions and

Table 7: Representation coupling diagnostics. The same cells and Sinkhorn setup produce substantially different couplings under PCA-30 and Program-4 geometry.

Representation	ε	L^1 vs other	top-15 overlap	Effective support	Entropy
PCA-30	0.05	1.750	0.0428	24,629	10.11
Program-4	0.05	1.750	0.0428	144,631	11.88

Table 8: State-space model metrics. Native metrics are scoped to their own representation; shared gene-program readouts support only readout-level comparison.

Training space	Native MMD	Program MMD	Program MAE	Fate mass error
PCA-30	0.00591	0.00994	0.0327	0.025
Program-4	0.00108	0.00126	0.0409	0.050

different learned flows under different r. The next diagnostic keeps the cells and OT setup fixed and changes only this representation map.

4.2.2 Representation Geometry Changes the Biological Question

Representation space enters the supervision chain by defining distances, couplings, endpoint pairs, paths, and velocity labels. This choice of geometry determines what biological question the model is learning. This viewpoint is consistent with recent representation-aware transport work: FlatVI and CellStream emphasize that latent representations should be evaluated by their downstream interpolation and trajectory geometry, not only by reconstruction, clustering, or visualization (Palma et al., 2025; Ling et al., 2025); GENOT makes cost functions, mass constraints, and transport families explicit (Klein et al., 2024). To interpret a learned flow as biological dynamics, one must first specify the representation geometry in which the flow lives.

Figure 12 and Tables 7–8 show this on controlled toy data and EB diagnostics. We use the same source and target cells and the same OT settings, changing only the state representation. PCA-30 compresses expression into 30 principal components, preserving major expression variation and giving a stable compact training space. We do not expect PCA-20 to be the right default for every tissue or modality. Program-4 represents each cell by four predefined gene-program scores, making directions easier to read as progenitor, major fate, or rare fate signals. That interpretability is itself a prior: the model learns transport in the predefined program space rather than discovering the programs without supervision.

The L^1 distance between the PCA-30 and Program-4 couplings is 1.750, and the top-15 coupling overlap is only 0.0428. The two representations allocate transport mass to very different source-target pairs. Downstream biological narratives change because endpoint-pair supervision has already changed. In shared 4D gene-program readout, the PCA-30 and Program-4 models also give different distribution fits and fate-mass errors, again showing that the supported terminal-fate composition and program-activation summaries depend on representation.

The same warning appears on real EB data, as shown in Figure 12c–d. A PHATE-induced diagnostic coupling looks shorter in the 2D display: its mean PHATE pair distance is 49.76, compared with 68.67 for the PC-20 coupling projected to PHATE. But in PC-20 training space, the PHATE-induced coupling has mean pair distance 5.95, farther than the PC-20 coupling's 4.03. Table 9 reports the corresponding training-space and display-space distances. Visual

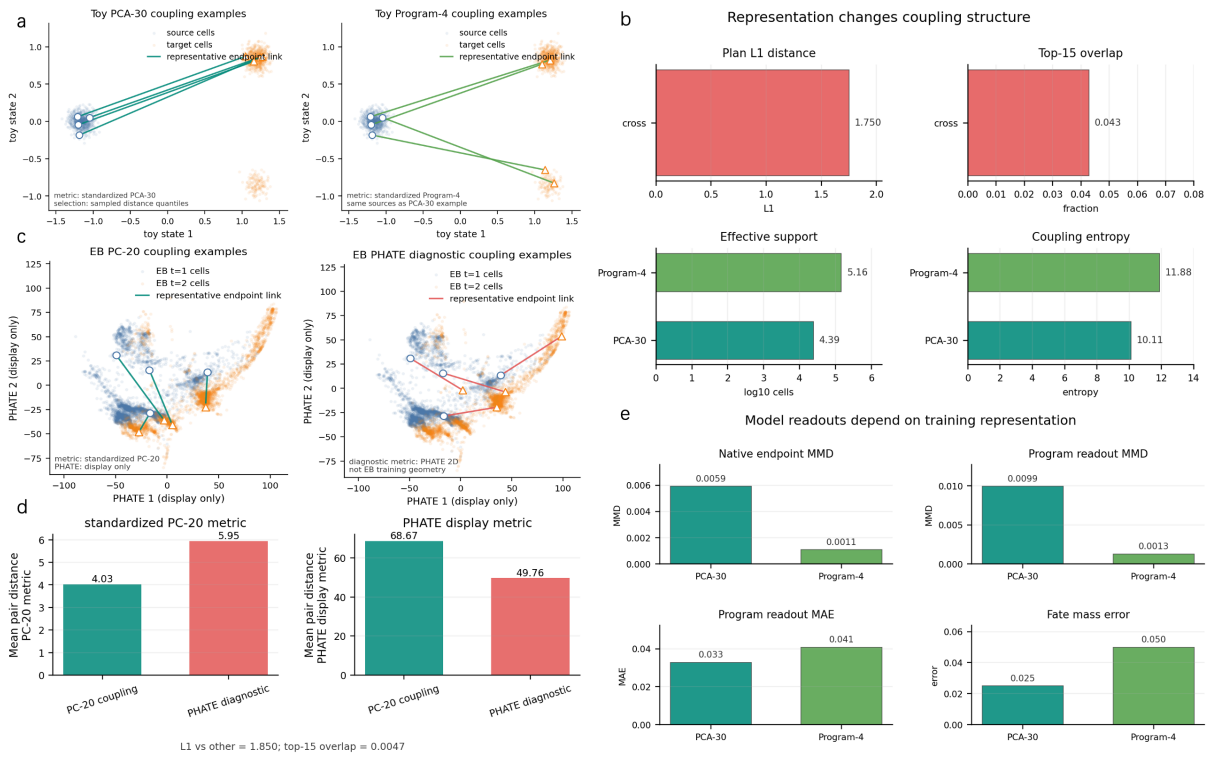

Figure 12: Representation choice changes the coupling, the training geometry, and the supported readout claims. (a) Controlled toy example with the same source and target cells under PCA-30 and Program-4 state representations. The OT setup is fixed, but representative endpoint links change because distances are computed in different representation spaces. (b) Coupling-level diagnostics for the toy example. The two representation-induced couplings have large L^1 distance and very small top-15 target overlap; effective support is shown on a $\log_{10}$ cell scale, and Program-4 has larger support and higher coupling entropy. (c) Real EB diagnostic example comparing a standardized PC-20 coupling with a PHATE-induced diagnostic coupling. Both panels are displayed in PHATE for visualization, but only the left coupling is computed in the PC-20 training space; the right coupling is a contrastive display-space diagnostic and is not used as the EB training geometry. (d) Training-space and display-space distances can disagree. The PC-20 coupling has lower mean pair distance in standardized PC-20, while the PHATE diagnostic coupling appears shorter in PHATE display space. Visual proximity in PHATE does not imply low training-space transport cost. (e) Downstream toy model readouts also depend on the chosen state space. Native endpoint MMD is scoped within each representation and should not be ranked across spaces directly; shared Program-4 readouts provide a common biological summary, where the two trained spaces yield different program errors and fate-mass errors. Together, these panels show that the representation map r defines not only preprocessing, but the cost, coupling, and velocity supervision of the learned flow.

Table 9: EB representation coupling diagnostics. The PHATE-induced coupling is shorter in display space but farther in PC-20 training space.

Coupling geometry	L^1 vs other	top-15 overlap	PC-20 distance	PHATE distance
PC-20 coupling	1.850	0.0047	4.03	68.67
PHATE diagnostic	1.850	0.0047	5.95	49.76

nearness and training-space nearness are different objects.

4.2.3 The Straight-Line Bridge Is Also a Representation-Space Assumption

Coupling defines endpoint relations, but Flow Matching must also define the conditional path between endpoints. Standard CFM uses Euclidean straight interpolation,

$$x_t = (1 - t)x_0 + tx_1,$$

but this straight line is only straight in the current representation. Metric Flow Matching, MIOFlow, and TrajectoryNet challenge this default from complementary angles: data-induced metrics and geodesic latent paths try to keep paths closer to the data manifold, while dynamic OT path-energy views remind us that endpoint matching does not imply biological path realism (Kapuśniak et al., 2024; Huguet et al., 2022; Tong et al., 2020).

Figure 13 fixes the same PC-20 OT endpoint pairs and changes only the path construction. The first path family is the standard PC-20 straight chord. The second is a 20D kNN graph diagnostic path that follows a piecewise path along the local adjacency graph of observed cells. PHATE is used only for display; off-manifold statistics are computed in PC-20/20D space.

We measure two diagnostics. The `density-radius percentile` of an intermediate point reports how large its local neighborhood radius is relative to observed cells; larger values indicate lower-density support. The `kNN radius` directly measures distance to nearby observed cells. A positive `straight-minus-graph` value means that, for the same endpoint pair and time point, the straight chord midpoint is farther from data support than the graph path.

The difference is systematic. At $t = 0.25, 0.5, 0.75$, the mean straight-minus-graph density-radius percentiles are 7.41, 6.62, and 6.23; the corresponding kNN-radius differences are 0.145, 0.125, and 0.113. Positive-value proportions are about 57% to 65%. Even with fixed coupling, the path family changes intermediate states and velocity labels. Low CFM loss means the model fits the specified supervised bridge; it does not mean that the bridge's intermediate states are biologically realistic.

4.3 Biologizing Flow Matching with Proliferation, Stochasticity, and Prior Constraints

The preceding sections treated Flow Matching as a balanced transport model: define coupling and paths in a state space, then train a velocity field to push normalized probability mass from one time point to another. Cellular systems also divide and die, fate choices can be stochastic, and regulatory networks, perturbations, spatial microenvironments, and lineage information shape how the flow should be interpreted. Recent work increasingly treats growth, death, and transport as one coupled learning problem rather than separate post-hoc corrections (Sha et al., 2024; Zhang et al., 2025; Wang et al., 2025; Peng et al., 2026). The open question is not whether to model growth, but which mass convention the data actually supports.

4.3.1 Non-Conservative Flow with Proliferation, Death, and Abundance Changes

The default continuity equation describes a probability flow with total mass fixed at 1:

$$\partial_t \rho_t(x) + \nabla \cdot (\rho_t(x) v_t(x)) = 0.$$

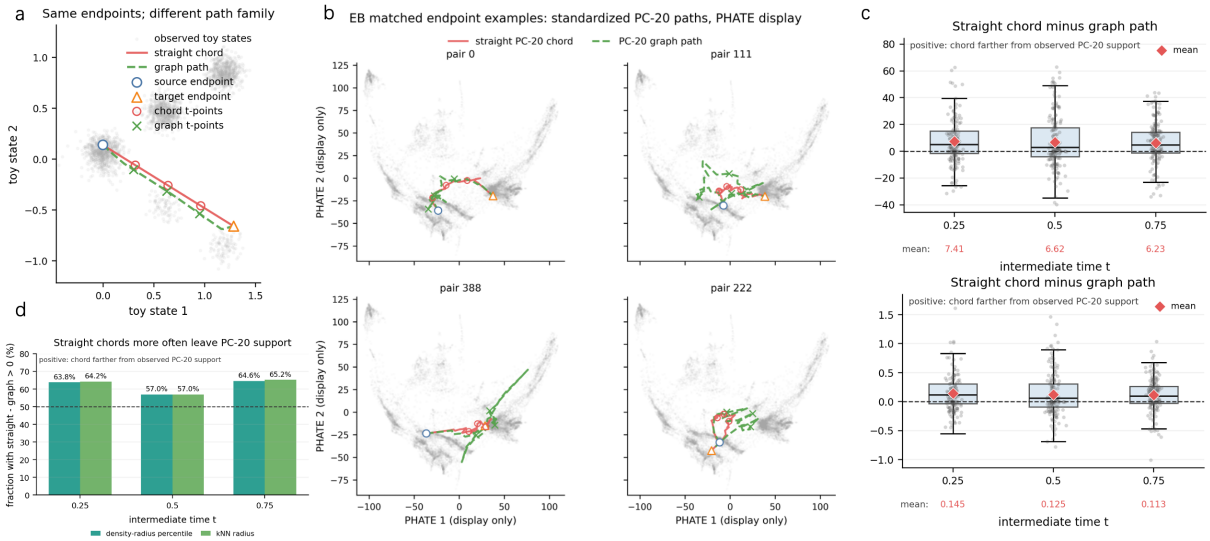

Figure 13: The path family is a modeling assumption even when endpoint pairs are fixed. (a) Toy illustration with the same source and target endpoint. A Euclidean straight chord and a graph-based diagnostic path pass through different intermediate states, so they would generate different conditional bridge samples and velocity labels for Flow Matching. (b) Matched EB endpoint examples. For the same fixed PC-20 OT endpoint pairs, the straight PC-20 chord and the 20-dimensional kNN graph diagnostic path can traverse different regions when projected to PHATE. These panels are qualitative displays only: the paths are constructed and evaluated in standardized PC-20, and PHATE is not a training geometry or lineage ground truth. (c) Paired EB support-distance diagnostics. For each intermediate bridge time, the plotted value is the straight-chord diagnostic minus the graph-path diagnostic. Positive values mean that the straight chord lies farther from observed PC-20 support. Both density-radius percentile and kNN-radius diagnostics have positive mean deltas at $t = 0.25, 0.5, 0.75$, indicating that straight interpolation tends to place intermediate states farther from the observed data support than the graph diagnostic path. (d) Frequency of positive support-distance deltas. Across intermediate times, more than half of the endpoint examples have positive straight-minus-graph deltas for both diagnostics. Even with the endpoint coupling held fixed, changing the path family changes the intermediate states and the supervised velocity labels. The standard straight-line bridge is itself a representation-space assumption.

In this setting, probability mass in a region changes only through inflow and outflow. Real biology also has local abundance change through proliferation and death. To model real cell numbers or externally calibrated abundance, let $p_t(x)$ be an unnormalized density with total abundance

$$p_t(x) \geq 0, \qquad M_t = \int p_t(x)\, dx.$$

The continuity equation becomes

$$\partial_t p_t(x) + \nabla \cdot (p_t(x) v_t(x)) = s_t(x),$$

where $s_t(x)$ is the local net source or sink. For local birth-death dynamics, a multiplicative growth form is natural:

$$s_t(x) = g_t(x) p_t(x),$$

which gives

$$\partial_t p_t(x) + \nabla \cdot (p_t(x) v_t(x)) = g_t(x) p_t(x).$$

The particle view is that positions move according to v_t, while mass weights change according to $d \log m_t / dt = g_t(X_t)$. the supplementary material gives the weak change-of-variables derivation for this statement. Integrating over state space under no-flux boundary conditions yields

$$\frac{dM_t}{dt} = \int g_t(x) p_t(x)\, dx \qquad \text{or, for additive source,} \qquad \frac{dM_t}{dt} = \int s_t(x)\, dx.$$

Here v_t explains movement of existing mass, while g_t or s_t explains abundance change. Local proportional increase can arise from transport, proliferation in that region, death elsewhere, or normalization effects.

From the OT perspective, this is the transition from balanced to unbalanced OT (Chizat et al., 2018a). Static unbalanced OT relaxes marginal constraints and penalizes untransported, created, or deleted mass:

$$\mathcal{UOT}(\mu, \nu) = \inf_{\gamma \geq 0} \int c(x, y)\, d\gamma(x, y) + \lambda_1 D((\pi_x)_{\#}\gamma \mid \mu) + \lambda_2 D((\pi_y)_{\#}\gamma \mid \nu).$$

Dynamic unbalanced transport writes movement and growth costs together. A Wasserstein–Fisher–Rao-type energy combines transport and Fisher–Rao growth costs (Chizat et al., 2018b):

$$\inf_{p,v,g} \int_0^1 \int \left(\|v_t(x)\|^2 + a\, g_t(x)^2 \right) p_t(x)\, dx\, dt, \quad \text{s.t.} \quad \partial_t p_t + \nabla \cdot (p_t v_t) = g_t p_t.$$

Compared with balanced FM, the model must decide not only where cell states move but also which mass is retained, expanded, or decayed.

Recent work has developed this direction. TrajectoryNet connects continuous normalizing flows with dynamic OT and incorporates growth/death, manifold-density, and local-velocity priors (Tong et al., 2020). TIGON uses dynamic unbalanced OT and WFR distance to learn velocity, growth, and cellular dynamics (Sha et al., 2024). DeepRUOT learns continuous unbalanced stochastic dynamics from snapshots; VGFM formulates joint velocity-growth flow matching; WFR-FM unifies Flow Matching with dynamic unbalanced OT under WFR geometry (Zhang et al., 2025; Wang et al., 2025; Peng et al., 2026).

Empirical considerations on sampling depth and mass convention. The mathematical framework is strong, but it requires a data condition: destructive scRNA-seq sample counts at each time point must reflect, or at least be proportional to, the true total abundance. If Day 1 contains 5000 sequenced cells and Day 2 contains 8000, that does not by itself imply a 60% increase in real cell number. The difference may come from loading, capture efficiency, dissociation bias, sequencing depth, filtering, or design. Without external abundance calibration, most single-cell time points should be normalized into cell-state distributions rather than treated as absolute post-growth cell counts.

The EB experiments illustrate the issue. Raw observed depth differs across time points, with totals $(2381, 4163, 3278, 3665, 3332)$. Equal-depth subsampling fixes each time point at 2381 cells, removing sampling-depth differences and leaving state-bin composition changes. Ta-

Table 10: Sampling-depth sensitivity summary for EB state-bin diagnostics. State bins are coarse k-means bins in standardized PC-20, not validated cell types.

Diagnostic	Result
Observed cell counts across time	$(2381, 4163, 3278, 3665, 3332)$
Equal-depth setting	2381 cells per time point
Time-bridge/bin comparisons	24
Sensitive to equal-depth bootstrap	16
Raw proxy inside equal-depth 95% interval	8
Claim boundary	Sampling depth, not absolute biological abundance

Table 11: WFR-FM raw-depth versus equal-depth agreement averaged across seeds and evaluation times.

Spearman rank	Top expanding overlap	Top shrinking overlap	Sign agreement
0.96	0.98	0.87	0.78

ble 10 summarizes the diagnostic: 16 of 24 time-bridge/state-bin comparisons are sensitive to downsampling, and only 8 raw-count proxies fall within the equal-depth bootstrap 95% interval. Raw count changes alone are not evidence for real proliferation, apoptosis, or clone expansion.

We also tested WFR-FM under raw observed-depth and equal-depth settings. The ranking structure is stable: average Spearman rank correlation is 0.96, top expanding overlap is 0.98, and top shrinking overlap is 0.87. But absolute calibration and local signs depend on mass convention. The unweighted mean growth at time 0 changes from 0.62 under raw depth to approximately 0.01 under equal depth; at time 4 it changes from -0.20 to approximately -0.02. Overall sign agreement is about 0.78, and at times 0 and 2 it is only 0.375. Table 11 reports the aggregate agreement scores. Figure 14 combines these checks: the raw-depth and equal-depth views differ in mass convention, and WFR-FM growth readouts inherit that convention. Growth-aware FM is mathematically meaningful, but its learned g_t is defined under the input mass convention.

4.3.2 Stochastic Flow: Randomness in Cell Paths

Individual cell-state evolution may be stochastic rather than deterministic. Cells have transcriptional bursting, molecular reaction noise, regulatory-network fluctuations, cell-cycle effects, and microenvironmental perturbations. Even conditioned on $X_t = x$, the real system may not have a unique next direction. A more natural object can be a conditional transition distribution $P(X_{t+\Delta t} \mid X_t = x)$ or a path distribution $P(X_{0:T} \mid X_0)$ (Elowitz et al., 2002; Zechner et al., 2020).

Standard FM or CFM usually learns a deterministic ODE sampler:

$$\frac{dX_t}{dt} = v_\theta(X_t, t).$$

Given X_0, the path is unique once v_θ is fixed. This is useful for population-average transport, but it assumes that current state and velocity field are enough to determine the next state. An SDE extension writes

$$dX_t = b_\theta(X_t, t)\, dt + \sigma_\theta(X_t, t)\, dW_t,$$

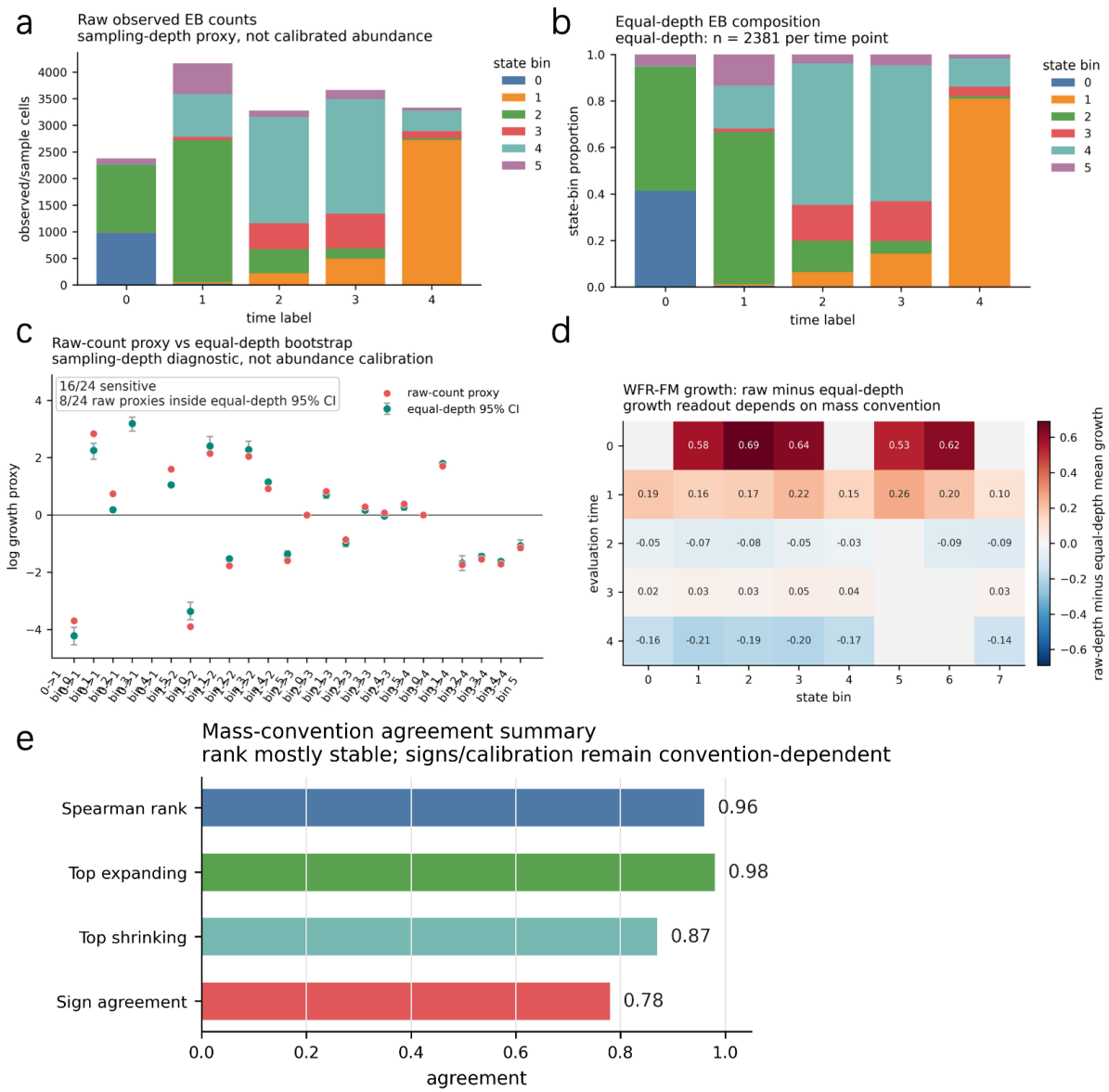

Figure 14: Sampling depth and mass convention bound growth-aware Flow Matching claims. (a) Raw observed EB cell counts vary across destructive snapshots and should be treated as sampling-depth proxies rather than calibrated biological abundance. (b) Equal-depth subsampling fixes each time point to the same total depth and preserves a state-bin composition view while removing total-depth differences. (c) Raw-count growth proxies are sensitive to equal-depth bootstrap intervals: 16 of 24 bridge/bin comparisons are sensitive, and only 8 of 24 raw proxies fall inside the equal-depth 95% interval. (d) WFR-FM growth readouts also change under the mass convention, shown as raw-depth minus equal-depth mean growth across evaluation times and state bins. (e) Agreement summaries show that rank structure is mostly stable, with Spearman rank agreement 0.96, top expanding overlap 0.98, and top shrinking overlap 0.87, but sign agreement is lower at 0.78. These panels support a mass-convention sensitivity claim only.

where b_θ is drift and $\sigma_\theta dW_t$ is diffusion. The next state from the same current state remains a distribution. At the population level, this can represent wider intermediate distributions and stochastic transition hypotheses.

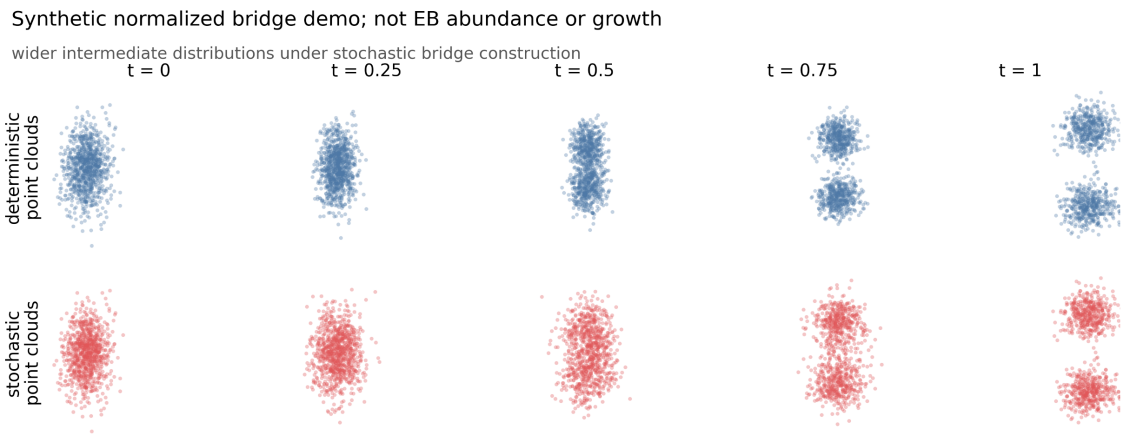

Figure 15: Stochasticity is a separate bridge assumption from mass convention. This synthetic normalized demonstration fixes the same endpoint marginals and total mass while comparing deterministic and stochastic bridge constructions. The deterministic panels show narrower intermediate point clouds, whereas the stochastic panels show wider intermediate distributions under Brownian-bridge-like noise. This is a synthetic demonstration; it does not use observed single-cell sample paths.

For isotropic diffusion, the density evolves by the Fokker–Planck equation (Risken, 1996)

$$\partial_t \rho_t(x) = -\nabla \cdot (\rho_t(x) b_t(x)) + \frac{1}{2}\sigma_t^2 \Delta \rho_t(x).$$

The drift term transports mass; the diffusion term spreads probability. Unlike growth, diffusion changes path randomness and normalized density spread while preserving total mass under suitable boundary conditions.

Many stochastic processes can connect the same endpoint marginals. A Schrödinger bridge chooses the path distribution closest in KL divergence to a reference stochastic process, such as Brownian motion or a diffusion with prior drift (Léonard, 2014). It is a stochastic generative bridge constrained by endpoint marginals. Single-cell methods increasingly use this view. PRESCIENT models differentiation as a stochastic process on a learned potential landscape (Yeo et al., 2021); scDiffEq learns drift and state-dependent diffusion and evaluates against lineage-traced data (Vinyard et al., 2025); DynNet and Probability Flow Matching connect stochastic paths to biophysical GRN dynamics and perturbation mechanisms (Dou et al., 2026; Maddu et al., 2026); MIOFlow 2.0 combines stochasticity, branching, growth, and spatial influence (Sun et al., 2026). Simulation-free stochastic bridge objectives such as $[SF]^2$M and Action Matching learn stochastic dynamics from unpaired temporal marginals (Tong et al., 2024b; Neklyudov et al., 2023).

The demonstration in Figure 15 fixes the same source distribution, target distribution, and normalized mass. The deterministic bridge gives narrow intermediate point clouds, while Brownian-bridge-like noise produces broader intermediate distributions at $t = 0.25, 0.5, 0.75$. The same endpoint marginals can be connected by different bridge families. This is separate from the mass-convention issue above: stochasticity changes the path distribution under fixed normalized mass, whereas WFR-style growth modeling changes how total mass is represented.

4.3.3 Flow Matching with Integrated Biological Priors

A third direction is to integrate flow matching with biological priors already present in single-cell experiments. RNA velocity, spliced/unspliced counts, and metabolic labeling provide local directional information that can encourage $v_\theta(x, t)$ to align with transcriptional dynamics (La Manno et al., 2018; Bergen et al., 2020; Lange et al., 2022). Marker genes, gene programs, and cell-type annotations can constrain trends along predicted paths. GRN priors can impose structure, sparsity, or sign constraints on the drift field. Spatial transcriptomics, CITE-seq, cell-cell communication features, and microenvironment annotations can influence local transition rules. Lineage barcodes, clone information, and live imaging provide external evidence for selected fate relations.

The most direct combination is regularization:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{FM}} + \sum_j \lambda_j \mathcal{R}_j^{\text{bio}}.$$

Here $\mathcal{R}_j^{\text{bio}}$ may encode RNA velocity alignment, marker monotonicity, GRN consistency, or spatial smoothness. Another route is to change the coupling, cost, or state-space geometry, for example by using graph distance, manifold metrics, or spatial neighborhoods to define more biologically structured path costs (Kapuśniak et al., 2024; Huguet et al., 2022; Bunne et al., 2024). These routes are compatible: a model can include RNA velocity regularization, GRN consistency, and manifold-aware geometry at the same time. Table 12 summarizes the claim boundaries used by the diagnostics in this section.

Table 12: Biological-assumption boundary checklist. All entries are diagnostic only.

Assumption	Status	Claim boundary
State labels	Coarse k-means bins	Diagnostic bins, not validated cell types
Cell counts	Observed snapshot counts	Sampling depth, not absolute abundance
Subsampling	Equal-depth bootstrap repeats = 50	Stability diagnostic only
Adjacent bridge OT	PC-20 endpoint diagnostics	Not trained CFM and not observed lineage

The learned vector field is meaningful only relative to the geometry and assumptions that produced its velocity labels. Flow Matching can turn snapshots into useful population-flow simulators, but these choices—coupling, state space, path family, mass convention, and biological priors—must be declared before interpreting a learned flow.

5 Flow Matching in Real Biological Tasks

The previous sections have developed Flow Matching as a way to transport one snapshot into another. Real experiments rarely stop at a single pair. Time-course studies provide a sequence of destructive snapshots $\widehat{\mu}_{t_0}, \widehat{\mu}_{t_1}, \ldots, \widehat{\mu}_{t_K}$ rather than a single source–target pair. Training one bridge is no longer the only goal. The harder problem is assembling local bridge supervision into coherent population dynamics over experimental time.

The second setting is perturbation response prediction. Here the motion of a cell population depends on external conditions: drug identity, dose, gene knockout, stimulation, donor, cell line,

or tissue context. The same state x, at the same transport progress τ, may require different directions under different perturbations:

$$v_\theta(x, \tau, c_1) \neq v_\theta(x, \tau, c_2).$$

The task is to learn how the flow changes with time and condition, a direction now studied explicitly in Flow Matching models for complex systems and cell dynamics (Rohbeck et al., 2025; Richter et al., 2026). Section 5.1 studies how pairwise Flow Matching extends to a shared time-conditioned field. Section 5.2 studies how perturbation conditions enter a conditional velocity field and how the quality of the condition encoding determines the claim supported by the evaluation split.

5.1 Single-Cell Time-Course Dynamics from Local Bridges to a Shared Time Field

In a time-course single-cell experiment, the object of interest is not the future of any measured cell. The observations are destructive population snapshots at experimental times. We start from a sequence of empirical marginals

$$\widehat{\mu}_{t_0}, \widehat{\mu}_{t_1}, \ldots, \widehat{\mu}_{t_K},$$

and ask for a population-level dynamics whose pushforward distributions agree with these marginals. Flow-based time-course models and dynamic OT methods increasingly make this distinction explicit: local endpoint matching is useful, but multi-timepoint data must ultimately be placed on a common temporal axis (Schiebinger et al., 2019; Tong et al., 2020; Huguet et al., 2022; Yeo et al., 2021; Bunne et al., 2024; Klein et al., 2025a).

Recall the endpoint-pair Flow Matching recipe. We sample a source–target pair (X_0, X_1) from a coupling π, form a simple interpolation $X_\tau = (1-\tau)X_0 + \tau X_1$, and regress the network $v_\theta(X_\tau, \tau)$ to the path velocity $X_1 - X_0$. The most direct multi-timepoint extension is to cut the time series into adjacent intervals and train an independent local bridge for each pair,

$$\widehat{\mu}_{t_i} \xrightarrow{v_\theta^{(i,i+1)}} \widehat{\mu}_{t_{i+1}}.$$

Here $v_\theta^{(i,i+1)}$ denotes a velocity field trained only for the interval $[t_i, t_{i+1}]$; the superscript is an interval label, not a power. The corresponding local flow map is denoted by $\Phi^{(i,i+1)}$ below. This is mathematically valid: each interval is again a two-snapshot problem. But it learns a collection of separate transports. No single dynamics over experimental time emerges from it.

The first issue is the meaning of time. In a single bridge, $\tau \in [0, 1]$ is only normalized progress along an interpolation path. It is not the physical sampling time. If every interval has its own private τ, then identical values of τ across intervals do not refer to a shared biological clock, especially when experimental sampling times are uneven. A model may learn how to move cells inside each interval without learning one time-indexed vector field.

The second issue is global consistency. Local endpoint fit does not guarantee that adjacent intervals meet smoothly. At a shared boundary t_i, the terminal velocity predicted by the bridge

from t_{i-1} to t_i can differ sharply from the initial velocity predicted by the bridge from t_i to t_{i+1}. When a long trajectory is produced by composing local maps,

$$\Phi_{t_0 \to t_K} = \Phi^{(K-1,K)} \circ \cdots \circ \Phi^{(1,2)} \circ \Phi^{(0,1)},$$

small deviations at early intermediate populations also become inputs that later local models did not see during training. This is the long-horizon drift problem: a set of locally plausible bridges need not form a stable developmental simulator.

A natural remedy is to pool all local supervision into one shared velocity field defined on experimental time:

$$\frac{dX_t}{dt} = v_\theta(X_t, t).$$

For any sampled time pair $t_i < t_j$, choose a coupling

$$(X_i, X_j) \sim \pi_{ij}, \qquad \pi_{ij} \in \Pi(\widehat{\mu}_{t_i}, \widehat{\mu}_{t_j}),$$

and construct

$$x_\tau = (1-\tau)X_i + \tau X_j, \qquad t_\tau = (1-\tau)t_i + \tau t_j, \qquad u_{ij} = \frac{X_j - X_i}{t_j - t_i}.$$

The interpolation variable τ still builds the bridge sample, but the network sees t_τ, the corresponding location on the experimental time axis. The multi-timepoint objective is

$$\mathcal{L}_{\text{time}}(\theta) = \mathbb{E}_{(i,j),\,(X_i,X_j)\sim\pi_{ij},\,\tau\sim U(0,1)} \left[\|v_\theta(x_\tau, t_\tau) - u_{ij}\|^2 \right].$$

This objective retains the simulation-free training advantage of Flow Matching. Training still consists of local velocity regression; the difference is that all sampled intervals constrain a single $v_\theta(x, t)$.

The choice of time pairs now becomes a modeling decision. If only adjacent pairs (t_i, t_{i+1}) are sampled, the supervision stays close to the observed experimental order and avoids forcing the model to jump over intermediate structure. However, adjacent-only supervision may not sufficiently constrain long rollouts. If skip pairs (t_i, t_j) with $j > i+1$ are added, the shared field receives longer-range constraints, which may improve endpoint rollout but can also blur intermediate dynamics. The trade-off is empirical.

We test this trade-off in an embryoid-body time-course experiment with five timepoints. Training is performed in a 20-dimensional PC space and the visualization is projected to PHATE. The hidden timepoint is $t = 2$: it is removed from the training losses and used to test population-level recovery. We also evaluate the seen endpoint $t = 4$, reached by rolling out from $t = 0$, as a diagnostic of long-horizon stability. We compare three designs: independent pairwise local bridges, a shared adjacent-only field, and a shared adjacent+skip field.

Figure 16 summarizes these three time-course designs and their main diagnostics, and Table 13 reports the corresponding numerical values.

Sharing a time-conditioned field helps hidden-time recovery. Pairwise local bridges give MMD 0.0836 and sliced W_2 0.408 at the unseen $t = 2$ population. The shared adjacent-only

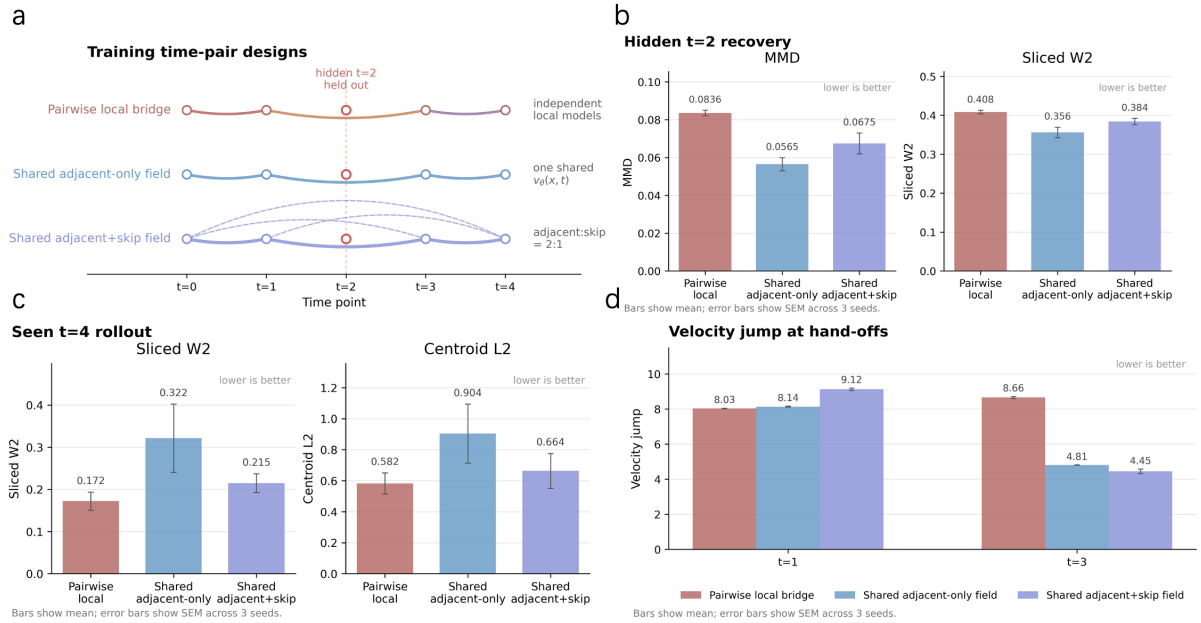

Figure 16: Multi-timepoint Flow Matching designs and diagnostics on EB snapshots. (a) Training schemes: independent pairwise local bridges, shared adjacent-only field, and shared adjacent+skip field (2:1 weighting). (b–d) Hidden-time recovery, long-horizon rollout, and velocity-jump metrics. Bars show means across seeds with SEM; all metrics computed in PC training space.

Table 13: Time-course recovery and hand-off diagnostics. Entries are means over three seeds. Lower is better for MMD, Sliced W_2, centroid error, and velocity jump. Velocity jump is a hand-off diagnostic only.

	Hidden $t = 2$			Seen $t = 4$			Velocity jump	
Method	MMD	Sliced W_2	Centroid	MMD	Sliced W_2	Centroid	$t = 1$	$t = 3$
Pairwise local bridge	0.0836	0.408	1.128	0.00687	0.172	0.582	8.03	8.66
Shared adjacent-only field	0.0565	0.356	1.139	0.0141	0.322	0.904	8.14	4.81
Shared adjacent+skip field	0.0675	0.384	1.221	0.00890	0.215	0.664	9.12	4.45

field cuts these to 0.0565 and 0.356. Adding skip pairs also improves over pairwise, with MMD 0.0675 and sliced W_2 0.384, but is slightly worse than adjacent-only on this interpolation task. Adding more supervision does not automatically improve hidden-time interpolation quality.

The seen $t = 4$ endpoint tells a different story. Within the shared-field family, adding weighted skip supervision improves the long rollout: Sliced W_2 decreases from 0.322 to 0.215, and centroid error decreases from 0.904 to 0.664, corresponding to reductions of about 33.1% and 26.6%, respectively (computed as (baseline − model)/baseline). The pairwise local bridge remains a strong endpoint-composition baseline, but it does so by stitching separate interval models rather than by learning one smooth time field. This distinction matters for the scientific claim: endpoint arrival and global dynamical consistency are not the same property.

Finally, the velocity-jump diagnostic makes the stitching problem visible. Pairwise local bridges show large hand-off jumps at both $t = 1$ and $t = 3$, around 8 in mean L_2 norm. Shared fields do not eliminate all discontinuity in this finite experiment, and the early hand-off remains large. They do, however, substantially reduce the late hand-off at $t = 3$, from 8.66

to 4.81 for adjacent-only and 4.45 for adjacent+skip. The conclusion is specific: ordinary Flow Matching gives efficient local velocity supervision; time-course modeling requires organizing that supervision on a shared experimental time axis, and skip pairs can help long-horizon rollout while sacrificing some hidden-time interpolation quality.

5.2 Perturbation Response Prediction from a Single Bridge to a Conditional Velocity Field

We now turn to a second biological task: perturbation response prediction. If an experiment contains only one treatment, such as vehicle control and one drug-treated A549 population, the problem can still be written as an ordinary endpoint bridge,

$$\widehat{\mu}_{\text{control}} \longrightarrow \widehat{\mu}^{\text{resp}}.$$

The control population is the source distribution, the treated population is the target distribution, and the same endpoint-coupling and path-construction machinery from earlier sections can be used to create local velocity labels.

This conditional form reflects a broader shift. The field is moving from predicting an average expression change toward modeling the full distributional response as a function of perturbation context (Bunne et al., 2023; Lotfollahi et al., 2023; Klein et al., 2025a). Real perturbation experiments rarely stop at one condition. Perturb-seq and genome-scale Perturb-seq place many genetic perturbations into a single-cell readout framework (Dixit et al., 2016; Replogle et al., 2022). sci-Plex combines small-molecule screening with single-cell transcriptomics, measuring chemical responses across cell lines and dose conditions (Srivatsan et al., 2020). In such data, there is not one treated endpoint. For each perturbation condition c, such as a compound–dose pair, a gene knockout, or a stimulation protocol, there is a response population:

$$\widehat{\mu}^{\text{resp}}_{c_1}, \widehat{\mu}^{\text{resp}}_{c_2}, \ldots, \widehat{\mu}^{\text{resp}}_{c_m}.$$

The target distribution has become a function of the condition.

This is because the model may see the same intermediate cell state x at the same normalized bridge progress τ. Under one compound, the appropriate direction may move toward a cell-cycle arrest program. Under another, it may move toward a stress-response program. Under a weak or inactive compound, it may barely move at all. A perturbation model needs to allow

$$v(x, \tau, c_1), \qquad v(x, \tau, c_2), \qquad v(x, \tau, c_3)$$

to point in different directions.

The first naive solution is to ignore the condition and pool all treated cells into one response distribution,

$$\widehat{\mu}_{\text{control}} \xrightarrow{v_\theta(x,\tau)} \widehat{\mu}^{\text{resp}}_{\text{all}}.$$

This uses the largest treated sample, but it discards the experimental label that explains why different response modes exist. Strong drugs, weak drugs, different mechanisms of action, and different doses are all collapsed into one target. The learned field can only represent an average

response. It may compress distinct response modes together, or learn a weak motion close to control if many perturbations have small effects.

The second naive solution is the opposite extreme: train a separate flow for each perturbation condition,

$$\widehat{\mu}_{\text{control}} \xrightarrow{v_\theta^{(c)}(x,\tau)} \widehat{\mu}_c^{\text{resp}}.$$

Each model fits its own target without averaging across conditions. Yet a large perturbation experiment is split into many isolated small tasks, and for a held-out compound there is no model to call. Different perturbations often share statistical structure: doses of the same compound move along related response directions; chemically or mechanistically related compounds can activate similar transcriptional modules; many perturbations share stress, cell-cycle, or apoptosis programs. Separate models cannot share this structure, and for a held-out compound there is no $v_\theta^{(c)}$ to call at all.

Perturbation response prediction needs an intermediate design: one model that shares parameters across conditions while still allowing the velocity direction to depend on the condition. The natural conditional Flow Matching form is

$$\frac{dX_\tau}{d\tau} = v_\theta(X_\tau, \tau, e(c), h_s), \qquad \tau \in [0,1].$$

Here τ is the normalized transport progress inside a bridge from a control endpoint to a response endpoint. The perturbation information, such as dose, compound identity, chemical structure, or gene perturbation, enters through $e(c)$. The source context, such as donor, cell line, tissue background, or batch, enters through h_s.

For a source context s, write the empirical control distribution and the response distribution under condition c as

$$\widehat{\mu}_{0,s} = \frac{1}{n_{0,s}} \sum_{i=1}^{n_{0,s}} \delta_{x_i^{(0,s)}}, \qquad \widehat{\mu}_{c,s}^{\text{resp}} = \frac{1}{n_{c,s}} \sum_{i=1}^{n_{c,s}} \delta_{x_i^{(c,s)}}.$$

For each (c, s), choose an endpoint coupling

$$(X_0, X_1) \mid c, s \sim \pi_{c,s}, \qquad \pi_{c,s} \in \Pi(\widehat{\mu}_{0,s}, \widehat{\mu}_{c,s}^{\text{resp}}).$$

The source endpoint X_0 is sampled from the control cells in context s, and the response endpoint X_1 is sampled from treated cells under condition c in the same context. An independent coupling adds no geometric pairing prior. An OT coupling assumes that the response is relatively local in the chosen state space and so prefers nearby control–response pairs. CellOT follows this logic by framing the prediction of treated responses from unpaired control cells as a neural optimal transport problem (Bunne et al., 2023).

With the linear endpoint path

$$X_\tau = (1-\tau)X_0 + \tau X_1, \qquad u_\tau = X_1 - X_0,$$

the conditional Flow Matching loss is

$$\mathcal{L}_{\text{cond}}(\theta) = \mathbb{E}_{c,s,\tau,(X_0,X_1)\sim\pi_{c,s}} \left[\| v_\theta(X_\tau, \tau, e(c), h_s) - u_\tau \|^2 \right].$$

The same network now sees all perturbation conditions, so it can share statistical strength. At the same time, the inputs $e(c)$ and h_s prevent the model from collapsing back to one unconditional average response.

The condition does not have to enter only through the velocity network. It can also affect the endpoint coupling $\pi_{c,s}$, the path family between endpoints, or even the source/base distribution. In the experiments below we use the minimal design: the condition enters $v_\theta(x, \tau, e(c), h_s)$, while the coupling and path family are kept simple. This isolates the role of condition encoding.

The effectiveness of conditional Flow Matching in perturbation prediction depends strongly on what information is present in $e(c)$. The simplest encoding is a compound one-hot vector plus dose. This lets the model memorize the response pattern of each compound observed during training and learn interpolation or extrapolation along the numeric dose. But a one-hot vector only says which training compound is being referenced. It does not tell the model which compounds are chemically or mechanistically related. When a completely unseen compound appears at test time, it has no natural position in the one-hot dictionary.

This is the motivation for structured condition encoding. CPA and chemCPA emphasize that predicting unseen drugs or new condition combinations requires perturbation representations with transferable information, rather than only discrete identifiers (Lotfollahi et al., 2023; Hetzel et al., 2022). For chemical perturbations, a natural choice is to compute RDKit2D molecular descriptors with RDKit, an open-source cheminformatics toolkit, from SMILES (Simplified Molecular Input Line Entry System) strings (Landrum, 2016; Weininger, 1988). Such descriptors do not imply that the model has discovered the mechanism of action. They simply place a held-out compound in a condition space where similarity to training compounds can be compared. If two histone deacetylase (HDAC) inhibitors are close in descriptor space, the model at least has a channel through which this similarity can influence the response direction.

We study this issue on an A549 chemical-perturbation subset of sci-Plex 3. The dataset contains vehicle controls and treated cells for eight compounds, each measured at four doses. Models are trained and evaluated in a split-aware PCA representation, so held-out doses or held-out compounds do not participate in fitting the representation for their split.

We compare four designs. M1 is unconditional Flow Matching, which pools all responses and learns one average flow. M2 trains one flow per compound; it can fit seen compounds but cannot share information across compounds. M3 is conditional Flow Matching with compound one-hot plus dose. M4 is chemistry-aware conditional Flow Matching with RDKit2D descriptors plus dose.

The two evaluation splits ask different questions. In the held-out highest dose split, every test compound has appeared in training, but its highest dose has not. The model has seen the compound and its lower-dose responses, and must extrapolate to the unseen highest dose. In the held-out compound split, the compound itself is completely removed from training; all four doses of that compound are held out. The model must transfer from seen compounds to an unseen chemical entity.

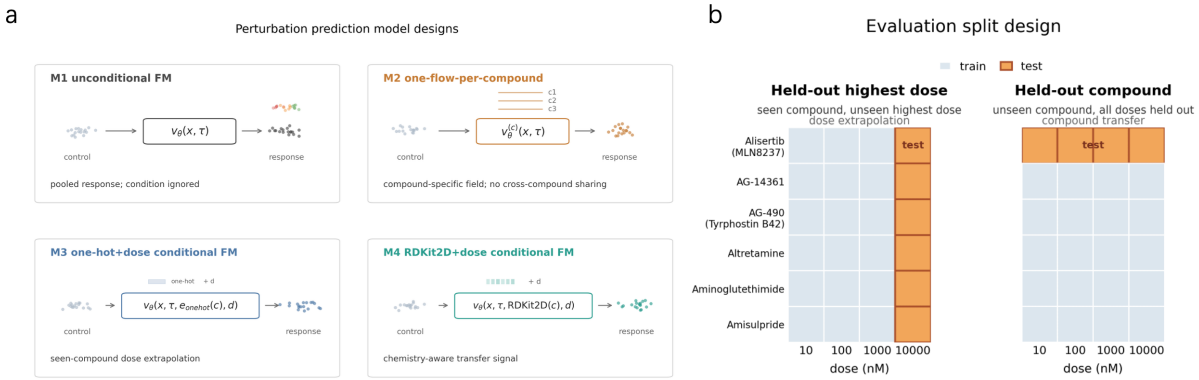

Figure 17: Perturbation-response model designs and evaluation splits for the sci-Plex 3 A549 chemical-perturbation subset with vehicle controls and eight compounds measured at four doses. (a) Models M1–M4 define increasingly conditioned Flow Matching designs: M1 ignores perturbation labels and learns an unconditional average response flow, M2 fits separate compound-specific local flows without cross-compound sharing, M3 learns a shared conditional field using one-hot compound identity and dose, and M4 replaces compound identity with RDKit2D chemical descriptors plus dose to support chemistry-aware transfer. (b) Evaluation separates two claims: the held-out highest-dose split tests dose extrapolation for compounds observed at lower doses, whereas the held-out-compound split tests transfer to a chemical identity and all of its doses removed from training.

Table 14: sci-Plex 3 split definitions. Vehicle cells are available only in training; test cells are treated response cells deliberately held out by dose or by compound.

Split	What is unseen in the test set?	Train vehicle	Train treated	Test treated	Test compounds/doses
Held-out highest dose	Highest dose for each otherwise seen compound	250	5714	1892	8 compounds / 1 dose
Held-out compound	One compound, with all four doses removed from training	250	6606	1000	1 compound / 4 doses

Table 15: sci-Plex 3 perturbation response metrics. Lower is better. The table reports the split-specific comparisons used in the text; simple baselines are included to make the claim boundary explicit. M2 is not applicable to the held-out compound split because the held-out compound has no compound-specific flow.

Split	Method	MMD	Sliced W_2	Mean abs. error	Centroid error
Held-out highest dose	M1 unconditional FM	0.0225	0.395	0.144	0.978
Held-out highest dose	M2 one-flow-per-compound	0.0242	0.381	0.164	1.101
Held-out highest dose	M3 one-hot+dose conditional FM	**0.0175**	**0.356**	0.156	1.054
Held-out highest dose	Vehicle as prediction	0.0219	0.372	0.165	1.146
Held-out highest dose	Mean-shift baseline	0.0250	0.381	**0.141**	**0.974**
Held-out compound	M1 unconditional FM	0.0983	1.001	0.346	2.599
Held-out compound	M3 one-hot+dose conditional FM	0.0841	0.965	0.368	2.736
Held-out compound	M4 RDKit2D+dose conditional FM	**0.0619**	0.803	0.461	3.293
Held-out compound	Vehicle as prediction	0.0635	**0.785**	**0.330**	**2.327**
Held-out compound	Mean-shift baseline	0.0777	0.824	0.341	2.506
Held-out compound	Nearest-chemistry baseline	0.0867	0.833	0.352	2.690

Figure 17 summarizes the modeling choices and the split logic. The schematic is deliberately split-aware: M3 is expected to be meaningful in the highest-dose split because the compound identity has appeared during training, while M4 is the only learned model here with structured information about an unseen compound. Table 14 adds the sample counts behind the schematic.

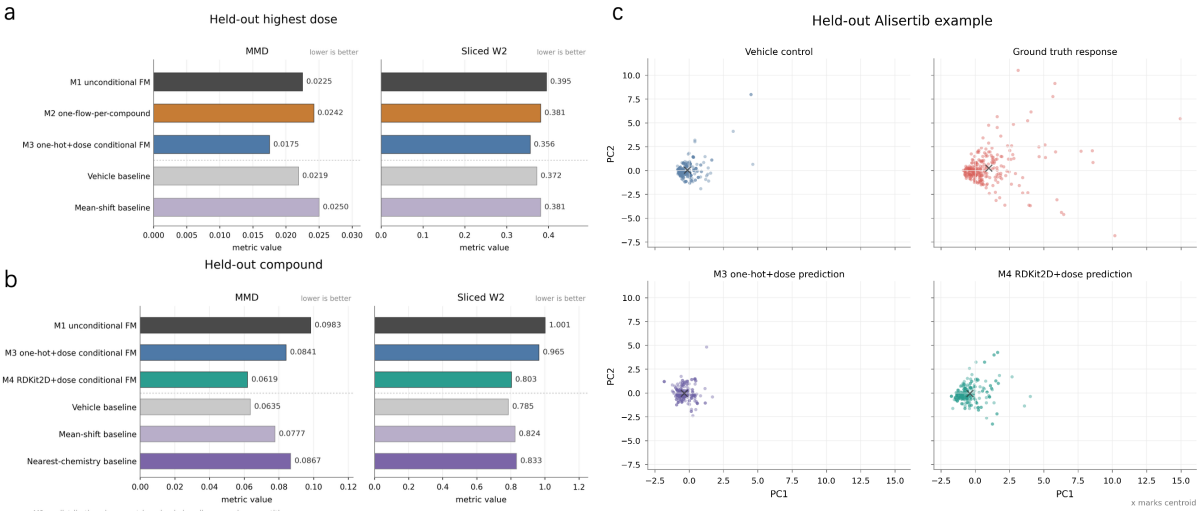

Figure 18: sci-Plex 3 perturbation prediction results. (a) In the held-out highest-dose split, MMD and sliced W_2 compare seen-compound dose extrapolation; M3 one-hot+dose conditional FM is the best learned model on both metrics. (b) In the held-out-compound split, M4 RDKit2D+dose conditional FM improves over M3 on MMD and sliced W_2, while simple baselines remain competitive. (c) A held-out Alisertib example shows the qualitative counterpart: the ground-truth response is broad and tail-heavy, M3 stays close to a compact control-like region, and M4 shifts and spreads the predicted population in a more response-like direction without recovering the full tail structure.

Figure 18 visualizes the two distribution-shape metrics emphasized in the text and the qualitative held-out Alisertib example. Table 15 gives the full numerical diagnostics, including mean absolute error and centroid error.

We first consider the held-out highest dose split. For every compound, the highest dose is removed from training while lower doses remain observed. M3, the one-hot+dose conditional model, performs best on the two distribution-shape metrics emphasized here: MMD 0.0175 and Sliced W_2 0.356 (Figure 18). It improves over M1 unconditional FM (0.0225/0.395) and M2 one-flow-per-compound (0.0242/0.381). The interpretation should be split-specific. Since the test compounds are already known to the model, one-hot identity is informative: it allows the shared field to use low-dose responses of the same compound when extrapolating to the highest dose. The improvement over M1 shows that a single average perturbation direction is insufficient. The improvement over M2 shows that sharing parameters across compounds is useful for this seen-compound dose-extrapolation task.

The held-out compound split is harder. Here all doses of one compound are removed from training, so a one-hot identifier has no transferable meaning. M4, which uses RDKit2D descriptors plus dose, improves over M3 on MMD and Sliced W_2: MMD decreases from 0.0841 to 0.0619, a reduction of about 26.4%, and Sliced W_2 decreases from 0.965 to 0.803, a reduction of about 16.8%. This shows that structured chemical encoding gives the conditional velocity field information that one-hot encoding does not provide for unseen compounds.

M4 does not dominate all diagnostics: its mean absolute error and centroid error are worse than M3, and simple vehicle or mean-shift baselines remain competitive on several metrics. The Alisertib panel in Figure 18 gives the qualitative version of the same boundary. The

observed response is more dispersed than vehicle and extends into a longer tail in the PC display. The M3 no-chemistry prediction remains close to the compact control-like region. The M4 chemistry-aware prediction leaves this region and produces a more visible shift and spread, but still fails to reconstruct the full tail and outlier structure. The figure and table agree: structured chemical conditions are useful signals for unseen-compound prediction, but the current RDKit2D representation, data scale, and training protocol are not enough to make the model a reliable zero-shot perturbation simulator.

Viewed more broadly, this experiment reflects a direction that is becoming clear in single-cell perturbation prediction: the field is moving from predicting only an average expression change toward conditional modeling of population distribution transfer. CellOT frames the prediction of perturbed populations from control populations as neural optimal transport (Bunne et al., 2023). CPA and chemCPA encode perturbation, dose, and cellular covariates in deep generative models to predict new condition combinations or unseen drug responses (Lotfollahi et al., 2023; Hetzel et al., 2022). Recent Flow Matching models continue this trajectory. CellFlow uses Flow Matching to model single-cell phenotypes induced by complex perturbations (Klein et al., 2025a), and scDFM casts perturbation prediction as distributional Flow Matching, explicitly emphasizing the distribution-level fit of perturbed expression profiles (Yu et al., 2026).

This trajectory also clarifies what future work must improve. Perturbation representations need to move beyond discrete identifiers: one-hot encodings support seen-compound extrapolation, but unseen compounds require transferable chemical structure, targets, or pathway information. Cell-state representations must be reliable—if the chosen state space primarily reflects batch effects, the learned flow may capture superficial shifts rather than perturbation-specific programs. Evaluation protocols must state the scope of supported claims: what was held out and what the metric can test. Systematic evaluations have shown that common perturbation-prediction metrics can be affected by systematic variation and that simple baselines can approach complex models under some protocols (Viñas Torné et al., 2025). Progress depends on richer perturbation representations, reliable cell-state representations, and split-aware evaluation that states exactly what was held out.

6 Conclusions

This tutorial review has framed single-cell dynamic biology as a problem of distributional transport under limited observation. Standard single-cell assays provide destructive and unpaired snapshots. The primary object is not an observed trajectory of a particular cell but a population-level connection among empirical marginal distributions. This distinction is the conceptual starting point for using Flow Matching in this setting.

Flow Matching enters the story as a practical way to learn continuous transport. Static optimal transport supplies a language for endpoint couplings. Dynamic optimal transport introduces density paths, velocity fields, and action. Continuous normalizing flows turn velocity fields into neural ODE models. Conditional Flow Matching makes training executable by replacing expensive simulation-in-the-loop objectives with local velocity regression. The result is a learned population velocity field whose rollout can map one empirical cell-state distribution toward another.

The central message is that a learned velocity field is meaningful only relative to its modeling assumptions. Five choices define the claim. **Endpoint coupling** decides which source and target cells provide velocity supervision. **Representation geometry** decides what straight and near mean. **The path family** decides the intermediate states used for training. **Mass convention** decides whether abundance changes are growth or just renormalized composition. **Stochasticity and priors** decide whether the model is an average deterministic transport or a richer family of transitions. None of these are implementation details. They are the scientific claim.

The same principle applies to real biological tasks. In time-course data, independent local bridges can match adjacent endpoints but do not automatically form one coherent dynamics over experimental time. Shared time-conditioned fields expose the need to organize local supervision on a common temporal axis. In perturbation-response prediction, a conditional velocity field can share statistical strength across conditions, but its ability to generalize depends on the information contained in the condition encoding and on the evaluation split. In both cases, endpoint fit, solver stability, path geometry, and biological interpretation must be reported as separate diagnostics.

Flow Matching is not focused on recovering individual cell lineages; rather, it serves as a framework for developing and testing hypotheses about population dynamics using snapshot data. To ensure that a learned flow is scientifically meaningful, three fundamental principles must be followed: explicitly articulate every transport assumption, carefully select representations for states and conditions, and ensure that each claim is substantiated by appropriate diagnostic evidence.

Data availability

The synthetic toy branching datasets used in the tutorial figures are generated by the accompanying code and require no external download. The embryoid body examples use the public PHATE embryoid body dataset (Moon et al., 2019), available from Mendeley Data with DOI 10.17632/v6n743h5ng.1. The perturbation-response examples use the sci-Plex 3 dataset (Srivatsan et al., 2020) through the scPerturb processed file `SrivatsanTrapnell2020_sciplex3.h5ad`, available from Zenodo record 13350497 with DOI 10.5281/zenodo.13350497; compound SMILES annotations are derived from the LINCS perturbation metadata available from GEO series GSE92742. Processed tutorial subsets, filtering metadata, and figure-reproduction scripts will be provided with the accompanying code release.

References

Sarah Aldridge and Sarah A. Teichmann. Single cell transcriptomics comes of age. *Nature Communications*, 11:4307, 2020. doi: 10.1038/s41467-020-18158-5.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, vol-

ume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 2017. URL https://proceedings.mlr.press/v70/arjovsky17a.html.

Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000. doi: 10.1007/s002110050002.

Volker Bergen, Marius Lange, Stefan Peidli, F. Alexander Wolf, and Fabian J. Theis. Generalizing rna velocity to transient cell states through dynamical modeling. *Nature Biotechnology*, 38(12):1408–1414, 2020. doi: 10.1038/s41587-020-0591-3.

Charlotte Bunne, Stefan G. Stark, Gabriele Gut, Jacobo Sarabia del Castillo, Mitch Levesque, Kjong-Van Lehmann, Lucas Pelkmans, Andreas Krause, and Gunnar Ratsch. Learning single-cell perturbation responses using neural optimal transport. *Nature Methods*, 20:1759–1768, 2023. doi: 10.1038/s41592-023-01969-x.

Charlotte Bunne, Geoffrey Schiebinger, Andreas Krause, Aviv Regev, Marco Cuturi, et al. Optimal transport for single-cell and spatial omics. *Nature Reviews Methods Primers*, 4:58, 2024. doi: 10.1038/s43586-024-00334-2.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, 2018.

Lénaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Unbalanced optimal transport: Dynamic and kantorovich formulation. *Journal of Functional Analysis*, 274(11):3090–3123, 2018a. doi: 10.1016/j.jfa.2018.03.008.

Lénaïc Chizat, Bernhard Schmitzer, Gabriel Peyré, and François-Xavier Vialard. An interpolating distance between optimal transport and fisher–rao metrics. *Foundations of Computational Mathematics*, 18:1–44, 2018b. doi: 10.1007/s10208-016-9331-y.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, 2013.

Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P. Fulco, Livnat Jerby-Arnon, Nemanja D. Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, et al. Perturb-seq: Dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. *Cell*, 167(7):1853–1866.e17, 2016. doi: 10.1016/j.cell.2016.11.038.

J. R. Dormand and P. J. Prince. A family of embedded runge-kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26, 1980. doi: 10.1016/0771-050X(80)90013-3.

Jingyu Dou, Wentao Lyu, Feng Chen, Qing Nie, and Chunhe Li. Inferring stochastic dynamics by biophysical neural ODE using single-cell transcriptomics. *Nature Communications*, 2026. doi: 10.1038/s41467-026-73257-z.

Michael B. Elowitz, Arnold J. Levine, Eric D. Siggia, and Peter S. Swain. Stochastic gene expression in a single cell. *Science*, 297(5584):1183–1186, 2002. doi: 10.1126/science.1070919.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T. H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. POT: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. URL https://jmlr.org/papers/v22/20-451.html.

Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: Free-form continuous dynamics for scalable reversible generative models. In *International Conference on Learning Representations*, 2019.

Leon Hetzel, Simon Böhm, Niki Kilbertus, Stephan Günnemann, Mohammad Lotfollahi, and Fabian J. Theis. Predicting cellular responses to novel drug perturbations at a single-cell resolution. In *Advances in Neural Information Processing Systems*, 2022.

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 *Advances in Neural Information Processing Systems*, volume 35, pages 29705–29718, 2022. doi: 10.52202/068431-2154.

Ian T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer New York, New York, NY, 2 edition, 2002. ISBN 978-0-387-95442-4. doi: 10.1007/b98835.

Leonid V. Kantorovich. On the translocation of masses. In *Doklady Akademii Nauk SSSR*, volume 37, pages 199–201, 1942.

Kacper Kapuśniak, Peter Potaptchik, Teodora Reu, Leo Zhang, Alexander Tong, Michael Bronstein, Avishek Joey Bose, and Francesco Di Giovanni. Metric flow matching for smooth interpolations on the data manifold. In *Advances in Neural Information Processing Systems*, volume 37, pages 135011–135042, 2024. doi: 10.52202/079017-4291.

Dominik Klein, Théo Uscidda, Fabian Theis, and Marco Cuturi. GENOT: Entropic (Gromov) wasserstein flow matching with applications to single-cell genomics. In *Advances in Neural Information Processing Systems*, volume 37, pages 103897–103944, 2024. doi: 10.52202/079017-3301.

Dominik Klein, Jonas Simon Fleck, Daniil Bobrovskiy, Lea Zimmermann, Soeren Becker, Alessandro Palma, Leander Dony, Alejandro Tejada-Lapuerta, Guillaume Huguet, Hsiu-Chuan Lin, Nadezhda Azbukina, Fatima Sanchis-Calleja, Theo Uscidda, Artur Szalata, Manuel Gander, Aviv Regev, Barbara Treutlein, J. Gray Camp, and Fabian J. Theis. Cellflow enables generative single-cell phenotype modeling with flow matching. *bioRxiv*, 2025a. doi: 10.1101/2025.04.11.648220.

Dominik Klein, Giovanni Palla, Marius Lange, Michal Klein, Zoe Piran, Manuel Gander, Laetitia Meng-Papaxanthos, Michael Sterr, Lama Saber, et al. Mapping cells through time and space with moscot. *Nature*, 638:1065–1075, 2025b. doi: 10.1038/s41586-024-08453-2.

Gioele La Manno, Ruslan Soldatov, Amit Zeisel, Emelie Braun, Hannah Hochgerner, Viktor Petukhov, Katja Lidschreiber, Maria E. Kastriti, Peter Lönnerberg, Alessandro Furlan, et al. Rna velocity of single cells. *Nature*, 560(7719):494–498, 2018. doi: 10.1038/s41586-018-0414-6.

Greg Landrum. RDKit: Open-source cheminformatics software, 2016. URL https://www.rdkit.org.

Marius Lange, Volker Bergen, Michal Klein, Manu Setty, Bernhard Reuter, Mostafa Bakhti, Heiko Lickert, Meshal Ansari, Janine Schniering, Herbert B. Schiller, et al. Cellrank for directed single-cell fate mapping. *Nature Methods*, 19:159–170, 2022. doi: 10.1038/s41592-021-01346-6.

Christian Léonard. A survey of the schrödinger problem and some of its connections with optimal transport. *Discrete and Continuous Dynamical Systems*, 34(4):1533–1574, 2014. doi: 10.3934/dcds.2014.34.1533.

Yue Ling, Peiqi Zhang, Zhenyi Zhang, and Peijie Zhou. CellStream: Dynamical optimal transport informed embeddings for reconstructing cellular trajectories from snapshots data. arXiv preprint arXiv:2511.13786, 2025. URL https://arxiv.org/abs/2511.13786.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Romain Lopez, Jeffrey Regier, Michael B. Cole, Michael I. Jordan, and Nir Yosef. Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15(12):1053–1058, 2018. doi: 10.1038/s41592-018-0229-2.

Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Leon Hetzel, Yuge Ji, Ignacio L. Ibarra, Sanjay R. Srivatsan, Mohsen Naghipourfar, Riza M. Daza, Beth Martin, Jay Shendure, Jose L. McFaline-Figueroa, Pierre Boyeau, F. Alexander Wolf, Nafissa Yakubova, Stephan Günnemann, Cole Trapnell, David Lopez-Paz, and Fabian J. Theis. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular Systems Biology*, 19:e11517, 2023. doi: 10.15252/msb.202211517.

Suryanarayana Maddu, Victor Chardès, and Michael J. Shelley. Learning biophysical models of gene regulation with probability flow matching. arXiv preprint arXiv:2604.25062, 2026. URL https://arxiv.org/abs/2604.25062.

Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*, pages 666–704, 1781.

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, Natalia B.

Ivanova, Guy Wolf, and Smita Krishnaswamy. Visualizing structure and transitions in high-dimensional biological data. *Nature Biotechnology*, 37(12):1482–1492, 2019. doi: 10.1038/s41587-019-0336-3.

Kirill Neklyudov, Rob Brekelmans, Daniel Severo, and Alireza Makhzani. Action matching: Learning stochastic dynamics from samples. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Alessandro Palma, Sergei Rybakov, Leon Hetzel, Stephan Günnemann, and Fabian J. Theis. Enforcing latent euclidean geometry in single-cell VAEs for manifold interpolation. arXiv preprint arXiv:2507.11789, 2025. URL https://arxiv.org/abs/2507.11789.

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 *International Conference on Learning Representations*, 2026.

Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

Joseph M. Replogle, Reuben A. Saunders, Angela N. Pogson, Jeffrey A. Hussmann, Alexander Lenail, Alina Guna, Liam Mascibroda, Eric J. Wagner, Karen Adelman, Gila Lithwick-Yanai, et al. Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq. *Cell*, 185(14):2559–2575.e28, 2022. doi: 10.1016/j.cell.2022.05.013.

Till Richter, Weixu Wang, Alessandro Palma, and Fabian J. Theis. Generative models of cell dynamics: From neural odes to flow matching. *Communications Biology*, 9:352, 2026. doi: 10.1038/s42003-026-09758-w.

Hannes Risken. *The Fokker-Planck Equation: Methods of Solution and Applications*, volume 18 of *Springer Series in Synergetics*. Springer Berlin Heidelberg, 2 edition, 1996. doi: 10.1007/978-3-642-61544-3.

Martin Rohbeck, Edward De Brouwer, Charlotte Bunne, Jan-Christian Hutter, Anne Biton, Kelvin Y. Chen, Aviv Regev, and Romain Lopez. Modeling complex system dynamics with flow matching across time and conditions. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=hwnObmOTrV.

Jennifer E. Rood, Samantha Wynne, Lucia Robson, Anna Hupalowska, John Randell, Sarah A. Teichmann, Aviv Regev, et al. The human cell atlas from a cell census to a unified foundation model. *Nature*, 637:1065–1071, 2025. doi: 10.1038/s41586-024-08338-4.

Wouter Saelens, Robrecht Cannoodt, Helena Todorov, and Yvan Saeys. A comparison of single-cell trajectory inference methods. *Nature Biotechnology*, 37(5):547–554, 2019. doi: 10.1038/s41587-019-0071-9.

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.e22, 2019. doi: 10.1016/j.cell.2019.01.006.

Yutong Sha, Yuchi Qiu, Peijie Zhou, Qing Nie, et al. Reconstructing growth and dynamic trajectories from single-cell transcriptomics data. *Nature Machine Intelligence*, 6:25–39, 2024. doi: 10.1038/s42256-023-00763-w.

Sanjay R. Srivatsan, Jose L. McFaline-Figueroa, Vijay Ramani, Lauren Saunders, Junyue Cao, Jonathan Packer, Hannah A. Pliner, Dana L. Jackson, Riza M. Daza, Lena Christiansen, Fan Zhang, Frank Steemers, Jay Shendure, and Cole Trapnell. Massively multiplex chemical transcriptomics at single-cell resolution. *Science*, 367(6473):45–51, 2020. doi: 10.1126/science.aax6234.

Xingzhi Sun, João Felipe Rocha, Brett Phelan, Dhananjay Bhaskar, Guillaume Huguet, Yanlei Zhang, D. S. Magruder, Alexander Tong, Ke Xu, Oluwadamilola Fasina, Mark Gerstein, Guy Wolf, Natalia Ivanova, Christine L. Chaffer, and Smita Krishnaswamy. MIOFlow 2.0: A unified framework for inferring cellular stochastic dynamics from single cell and spatial transcriptomics data. arXiv preprint arXiv:2603.22564, 2026. URL https://arxiv.org/abs/2603.22564.

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*, pages 9526–9536, 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.

Alexander 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 *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, 2024b.

Cole Trapnell, Davide Cacchiarelli, Jayson Grimsby, Pragya Pokharel, Shuqiang Li, Michael Morse, Niall J. Lennon, Kenneth J. Livak, Tarjei S. Mikkelsen, and John L. Rinn. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology*, 32(4):381–386, 2014. doi: 10.1038/nbt.2859.

Sophie Tritschler, Maren Büttner, David S. Fischer, Marius Lange, Volker Bergen, Heiko Lickert, and Fabian J. Theis. Concepts and limitations for learning developmental trajectories from single cell genomics. *Development*, 146(12):dev170506, 2019. doi: 10.1242/dev.170506.

Ramon Viñas Torné, Maciej Wiatrak, Zoe Piran, Shuyang Fan, Liangze Jiang, Sarah A. Teichmann, Mor Nitzan, and Maria Brbić. Systema: A framework for evaluating genetic perturbation response prediction beyond systematic variation. *Nature Biotechnology*, 44(6):1050–1059, 2025. doi: 10.1038/s41587-025-02777-8.

Michael E. Vinyard, Anders W. Rasmussen, Ruitong Li, Allon M. Klein, Gad Getz, and Luca Pinello. Learning cell dynamics with neural differential equations. *Nature Machine Intelligence*, 7(12):1969–1984, 2025. doi: 10.1038/s42256-025-01150-3.

Dongyi Wang, Yuanwei Jiang, Zhenyi Zhang, Xiang Gu, Peijie Zhou, and Jian Sun. Joint velocity-growth flow matching for single-cell dynamics modeling. arXiv preprint arXiv:2505.13413, 2025. URL https://arxiv.org/abs/2505.13413.

David Weininger. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988. doi: 10.1021/ci00057a005.

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*, 12:3222, 2021. doi: 10.1038/s41467-021-23518-w.

Chenglei Yu, Chuanrui Wang, Bangyan Liao, and Tailin Wu. scDFM: Distributional flow matching model for robust single-cell perturbation prediction. In *International Conference on Learning Representations*, 2026.

Christoph Zechner, Elisa Nerli, and Caren Norden. Stochasticity and determinism in cell fate decisions. *Development*, 147(14):dev181495, 2020. doi: 10.1242/dev.181495.

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