Modeling Microenvironment Trajectories on Spatial Transcriptomics with NicheFlow

Kristiyan Sakalyan [*1] **Alessandro Palma** [*1 2] **Filippo Guerranti** [*1 3] **Fabian J. Theis** [1 2 3 4]
Stephan Günnemann [1 3]

Abstract

Understanding the evolution of cellular microenvironments is essential for deciphering tissue development and disease progression. While spatial transcriptomics now enables high-resolution mapping of tissue organization across space and time, current techniques that analyze cellular evolution operate at the single-cell level, overlooking critical spatial relationships. We introduce NicheFlow, a flow-based generative model that infers the temporal trajectory of cellular microenvironments across sequential spatial slides. By representing local cell neighborhoods as point clouds, NicheFlow jointly models the evolution of cell states and coordinates using optimal transport and Variational Flow Matching. Our approach successfully recovers both global spatial architecture and local microenvironment composition across diverse spatiotemporal datasets, from embryonic to brain development.

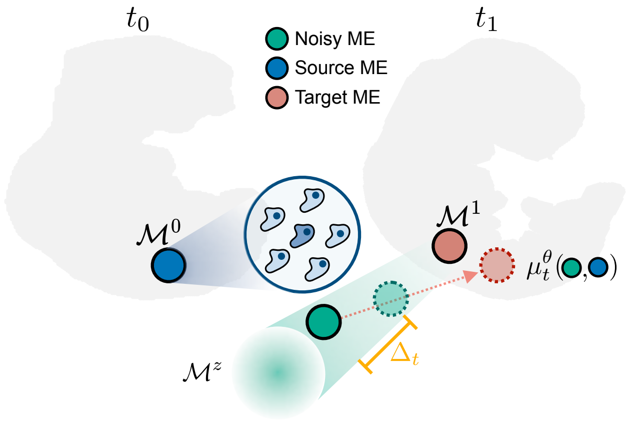


Figure 1. **Overview of NicheFlow.** At time t_1, we generate a target microenvironment $\mathcal{M}^1$ by transforming Gaussian noise $\mathcal{M}^z$ using a Variational Flow Matching model with a posterior μ_t^θ conditioned on a source microenvironment $\mathcal{M}^0$ at t_0. Source-target pairs are identified via entropic OT over pooled microenvironment coordinates and gene expression profiles.

1. Introduction

Uncovering the principles governing tissue organization across space and time remains one of the most fundamental challenges in biology, with profound implications for evolutionary and developmental biology (Mayr et al., 2019; Zinner et al., 2020). While individual cells form the basic units of biological systems, they operate not in isolation but as integral parts of spatially organized *microenvironments*, functionally distinct neighborhoods, or *niches*, shaped by cell-to-cell interactions and extracellular components (Ren et al., 2023b; Schaar et al., 2024; Liu et al., 2025). These spatial microenvironments influence crucial biological processes, from tumor progression to immune infiltration and tissue regeneration (Moncada et al., 2020; Larsson et al., 2021; Chen et al., 2022).

Spatial transcriptomics (ST) has transformed our ability to investigate these tissue architectures by providing single-cell resolution mapping of gene expression while preserving spatial context (Ståhl et al., 2016; Rodriques et al., 2019; Eng et al., 2019; Moses & Pachter, 2022). This technological breakthrough has enabled researchers to examine the molecular underpinnings of tissue organization with unprecedented detail. However, ST provides only static snapshots of inherently dynamic biological systems. *Time-resolved* spatial transcriptomics extends ST by capturing how gene expression patterns and cellular arrangements evolve across developmental stages or experimental time (Briggs et al., 2018; Wagner et al., 2018; Ren et al., 2023a).

[*] Equal contribution [1] School of Computation Information and Technology, Technical University of Munich, Germany [2] Institute of Computational Biology, Helmholtz Munich, Munich, Germany [3] Munich Data Science Institute, Technical University of Munich, Germany [4] TUM School of Life Sciences Weihenstephan, Technical University of Munich, Germany. Correspondence to: Kristiyan Sakalyan, Alessandro Palma, Filippo Guerranti <{kristiyan.sakalyan, alessandro.palma, filippo.guerranti}@tum.de>.

Proceedings of the Workshop on Generative AI for Biology at the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

This temporal dimension offers critical insights into the development of tissue organization that cannot be inferred from static observations alone.

Despite these technological advances, current computational methods fall short in modeling the evolution of tissue organization at the level of cellular microenvironments. Most approaches infer trajectories by modeling single-cell dynamics using velocity-based models (Abdelaal et al., 2024; Long et al., 2025; Shen et al., 2025) or optimal transport between individual cells (Klein et al., 2025; Bryan et al., 2025). While effective at capturing cell evolution, these cell-centric methods fundamentally miss the coordinated evolution of structured niches within tissues.

This limitation presents a critical research gap that we address with the following question:

How can we model the spatiotemporal evolution of cellular microenvironments while preserving both local neighborhood relationships and global tissue architecture?

To address this question, we directly model the dynamics of cellular neighborhoods as cohesive units rather than focusing on isolated cell trajectories. This approach aligns naturally with tissue-scale biological processes and enables principled learning of dynamics in structured, high-dimensional, and variably sized spatial domains.

We introduce **Niche Flow** Matching (**NicheFlow**) (Figure 1), a generative model for learning spatiotemporal dynamics of cellular niches from time-resolved spatial transcriptomics data. NicheFlow builds on recent advances in Flow Matching (FM) and Optimal Transport (OT) to operate over distributions of microenvironments, which we represent as point clouds. NicheFlow enables accurate modeling of global spatial architecture and local microenvironment composition within evolving tissues.

Our contributions include:

- **A microenvironment-centered trajectory inference paradigm** that shifts from modeling individual cells in time to modeling niches as point clouds, enabling simultaneous prediction of spatial coordinates and gene expression profiles while preserving local tissue context.
- **A factorized Variational Flow Matching (VFM) approach** with distributional families (Laplace for spatial coordinates, Gaussian for gene expression) that independently optimizes spatial and cell state dynamics, effectively modeling both spatial reconstruction and biological fidelity in time.
- **A spatially-aware sampling strategy** using OT between niche representations, enabling scalable training on large tissue sections while ensuring comprehensive coverage of heterogeneous regions.

Our approach consistently outperforms baselines in recovering cell-type organization and spatial structure across embryonic, brain development, and aging datasets. NicheFlow enables principled learning of dynamics in structured, high-dimensional, and variably sized spatial domains, a challenge with parallels in other spatiotemporal modeling domains beyond biology.

2. Related Work

NicheFlow is at the interface between generative models and spatiotemporal transcriptomic data.

FM and single-cell transcriptomics. We propose a model based on FM, a framework introduced by several seminal works (Liu et al., 2023; Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023). Specifically, we adopt a variational view of the FM objective, following Eijkelboom et al. (2024), but extend it to mixed-factorized distributions for point cloud generation. Our method, NicheFlow, combines FM with OT, a pairing that has proven effective in modeling cellular data (Tong et al., 2024a;b; Atanackovic et al., 2025; Klein et al., 2024). Unlike these models, however, we focus on point clouds of spatially-resolved transcriptomic profiles. Closest to our approach is Wasserstein FM for point cloud generation (Haviv et al., 2024), applied to reconstruct cellular niches. Yet, that work does not address joint generation of spatial coordinates and cellular states, nor OT-based temporal trajectory prediction, both central to our contribution.

Generative models for spatial transcriptomics. Generative models have been key to spatial tasks such as gene expression prediction from histology slides (Zhu et al., 2025; Wan et al., 2023), integration with dissociated single-cell data (Wan et al., 2023), and spatial imputation (Haviv et al., 2025; Li et al., 2024). More recently, LUNA (Yu et al., 2025a) demonstrated strong performance in predicting single-cell spatial coordinates using diffusion models (Ho et al., 2020) conditioned on transcription data. While related, our model addresses the distinct task of inferring niche trajectories, enabling the simultaneous generation of coordinates and cellular states.

Trajectory inference for spatial transcriptomics. Previous work has explored learning trajectories from spatial slides. Pham et al. (2023) proposed a graph-based spatiotemporal algorithm for pseudotime inference, while others leveraged tissue-resolved transcriptomics to estimate cell velocity (Abdelaal et al., 2024; Long et al., 2025; Shen et al., 2025). Closer to our approach, Klein et al. (2025) and Bryan et al. (2025) apply discrete OT to identify likely descendants of individual cells in spatially-resolved gene expression. However, unlike our mini-batch deep learning model, these methods do not operate on entire microenvironments and rely on exact OT at the single-cell level, leading to scalability limitations on larger slides.

3. Background

3.1. Optimal Transport with FM

FM (Lipman et al., 2023) is a generative model that transforms a source density p_0 into a target density p_1. It operates by learning a time-dependent velocity field $u_t(\boldsymbol{x})$ for $t \in [0,1]$, which generates a probability path $\{p_t\}_{t\in[0,1]}$. This path is constructed such that the marginals at time $t=0$ and $t=1$ match the source and target distributions, i.e., p_0 and p_1, respectively. The velocity field induces an Ordinary Differential Equation (ODE), whose solution $\phi_t(\boldsymbol{x})$ defines a *flow map* that transports samples from the source to the target distribution.

In practice, FM approximates $u_t(\boldsymbol{x})$ with a time-conditioned neural network $v_t^\theta(\boldsymbol{x})$. While the exact marginal velocity field $u_t(\boldsymbol{x})$ is intractable, it can be expressed in terms of data-conditioned velocity fields and a joint distribution $\pi(\boldsymbol{x}_0, \boldsymbol{x}_1)$ over the source and target samples $\boldsymbol{x}_0 \sim p_0$ and $\boldsymbol{x}_1 \sim p_1$:

$$u_t(\boldsymbol{x}) = \int u_t(\boldsymbol{x} \mid \boldsymbol{x}_0, \boldsymbol{x}_1) \cdot \frac{p_t(\boldsymbol{x} \mid \boldsymbol{x}_0, \boldsymbol{x}_1)\pi(\boldsymbol{x}_0, \boldsymbol{x}_1)}{p_t(\boldsymbol{x})} \, \mathrm{d}\boldsymbol{x}_0 \, \mathrm{d}\boldsymbol{x}_1 \, , \tag{1}$$

where $p_t(\boldsymbol{x} \mid \boldsymbol{x}_0, \boldsymbol{x}_1)$ is a pre-defined interpolating probability path. Here, we consider the tractable probability path $p_t(\boldsymbol{x} \mid \boldsymbol{x}_0, \boldsymbol{x}_1) = \delta(\boldsymbol{x} - g_t(\boldsymbol{x}_0, \boldsymbol{x}_1))$, where $g_t(\boldsymbol{x}_0, \boldsymbol{x}_1) = (1-t)\boldsymbol{x}_0 + t\boldsymbol{x}_1$ is a linear interpolation and δ denotes a Dirac delta function, representing a deterministic conditional path.

Lipman et al. (2023) show that regressing the conditional field $u_t(\boldsymbol{x} \mid \boldsymbol{x}_0, \boldsymbol{x}_1)$ is equivalent to learning the marginal field $u_t(\boldsymbol{x})$ in expectation. Hence, the FM objective becomes the task of learning the velocity along the conditional probability path between any pair of source and target data points. For a linear conditional probability path, the velocity $u_t(\boldsymbol{x} \mid \boldsymbol{x}_0, \boldsymbol{x}_1)$ has a closed form, and the FM loss is:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathop{\mathbb{E}}_{\substack{t\sim\mathcal{U}[0,1] \\ (\boldsymbol{x}_0,\boldsymbol{x}_1)\sim\pi}} \left[\left\| v_t^\theta\left(g_t(\boldsymbol{x}_0, \boldsymbol{x}_1)\right) - \frac{\partial}{\partial t} g_t(\boldsymbol{x}_0, \boldsymbol{x}_1) \right\|_2^2 \right] . \tag{2}$$

In practice, the coupling $\pi(\boldsymbol{x}_0, \boldsymbol{x}_1)$ is instantiated using sample pairs drawn from a mini-batch estimate. When one chooses $\pi^\star$ as the OT coupling under a squared Euclidean cost between samples from p_0 and p_1, FM approximates the dynamic OT map between source and target densities (Pooladian et al., 2023; Tong et al., 2024a). Thus, given the solution samples $(\boldsymbol{x}_0, \boldsymbol{x}_1) \sim \pi^\star$ from the joint distribution approximately follow:

$$\boldsymbol{x}_0 \sim p_0, \; \boldsymbol{x}_1 \sim \delta\left(\boldsymbol{x}_1 - \phi_1^\theta(\boldsymbol{x}_0)\right) , \tag{3}$$

where ϕ_t^θ is the solution of the ODE with velocity field v_t^θ.

3.2. Generative OT on incomparable source and target spaces

Klein et al. (2024) generalize the OT FM formulation to settings where the source and target distributions are defined on incomparable spaces and propose an approach to generative entropic OT using FM. Given a standard normal noise distribution with samples $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$, the authors show that the following sampling procedure:

$$\boldsymbol{x}_0 \sim p_0, \; \boldsymbol{x}_1 \sim \delta\left(\boldsymbol{x}_1 - \phi_1^\theta(\boldsymbol{z} \mid \boldsymbol{x}_0)\right) , \tag{4}$$

defines a generative model that implicitly samples from an Entropic OT (EOT) coupling, where $\phi_t^\theta(\boldsymbol{z} \mid \boldsymbol{x}_0)$ is a FM model that maps noise to target samples, conditioned on source points. To achieve this, ϕ_t^θ is trained using source-target pairs $(\boldsymbol{x}_0, \boldsymbol{x}_1)$ drawn from the EOT coupling $\pi_\epsilon^\star$, with ϵ denoting the entropic regularization parameter, which the model aims to approximate.

Crucially, this formulation enables OT between distinct source and target spaces, *as $\boldsymbol{x}_0$ does not flow directly into $\boldsymbol{x}_1$, but instead conditions the generation of target samples from noise*.

3.3. Source-conditioned VFM

Consider the source-conditioned FM formulation in Section 3.2. Given a conditioning source $\boldsymbol{x}_0$ and noise-based generation, the marginal field in Equation (1) can be written as:

$$u_t(\boldsymbol{x} \mid \boldsymbol{x}_0) = \mathop{\mathbb{E}}_{p_t(\boldsymbol{x}_1 \mid \boldsymbol{x}, \boldsymbol{x}_0)} \left[u_t(\boldsymbol{x} \mid \boldsymbol{x}_1) \right] , \tag{5}$$

where we drop the conditioning on $\boldsymbol{x}_0$ in the velocity field, as u_t is entirely determined by the target $\boldsymbol{x}_1$ when generating from noise under linear probability paths (see the derivation in Equation (5)).

Since $u_t(\boldsymbol{x} \mid \boldsymbol{x}_1)$ is tractable (Lipman et al., 2023), one can recast FM as a variational inference problem, following Eijkelboom et al. (2024), by introducing a parameterized approximation $q_t^\theta(\boldsymbol{x}_1 \mid \boldsymbol{x}, \boldsymbol{x}_0)$ to the true posterior $p_t(\boldsymbol{x}_1 \mid \boldsymbol{x}, \boldsymbol{x}_0)$. Integrating the expected velocity in Equation (5) over $t \in [0,1]$ enables the generation of target points $\boldsymbol{x}_1$ from noise, conditioned on $\boldsymbol{x}_0$.

During training, the source-conditioned Variational Flow Matching (VFM) loss is:

$$\mathcal{L}_{\text{SC-VFM}}(\theta) = - \mathop{\mathbb{E}}_{\substack{t\sim\mathcal{U}[0,1] \\ (\boldsymbol{x}_0,\boldsymbol{x}_1)\sim\pi_\epsilon^\star \\ \boldsymbol{x}\sim p_t(\boldsymbol{x}\mid\boldsymbol{x}_1)}} \left[\log q_t^\theta(\boldsymbol{x}_1 \mid \boldsymbol{x}, \boldsymbol{x}_0) \right] , \tag{6}$$

where $\pi_\epsilon^\star$ is an entropic OT coupling modeling the joint distribution over source and target samples, and $p_t(\boldsymbol{x} \mid$

$\boldsymbol{x}_1$) interpolates between target samples and noise. In the generation phase, one samples $\boldsymbol{x}_0 \sim p_0$ and noise $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$, then simulates the marginal field in Equation (5) starting from $\phi_0(\boldsymbol{x}) = \boldsymbol{z}$ to generate a target sample $\boldsymbol{x}_1$.

Crucially, under the assumption that $u_t(\boldsymbol{x} \mid \boldsymbol{x}_1)$ is linear in $\boldsymbol{x}_1$, which holds when using straight-line interpolation paths, the marginal field in Equation (5) only depends on the posterior's first moment on $\boldsymbol{x}_1$:

$$u_t(\boldsymbol{x}) = u_t\left(\boldsymbol{x} \mid \mathbb{E}_{p_t(\boldsymbol{x}_1 \mid \boldsymbol{x}, \boldsymbol{x}_0)}\left[\boldsymbol{x}_1\right]\right) \tag{7}$$

This implies that the VFM objective reduces to matching the first moment of the approximate posterior $q_t^\theta(\boldsymbol{x}_1 \mid \boldsymbol{x}, \boldsymbol{x}_0)$ to that of the true posterior $p_t(\boldsymbol{x}_1 \mid \boldsymbol{x}, \boldsymbol{x}_0)$. As a result, the approximate posterior can be chosen fully factorized under a *mean field assumption*, since each dimension can be matched independently if the mean of the true posterior is preserved; see Appendix C.2 for more details.

4. NicheFlow

We introduce a flow-based generative OT model to infer the temporal evolution of spatially resolved cellular microenvironments. More specifically, given a spatial microenvironment represented as a point cloud of cell states with their coordinates, NicheFlow predicts the corresponding tissue structure at a later time point. To delineate our approach, we define a list of desiderata.

Generative model on structured data. Similar to prior work (Yu et al., 2025b; Haviv et al., 2024), we consider a generative model over structured point cloud data representing cellular microenvironments. This approach implicitly accounts for spatial correlations between cells, in contrast to models that study the evolution of spatial trajectories at the single-cell level (Klein et al., 2025).

Sub-regions and variable location. Crucially, for better memory efficiency, we do not consider an entire spatial slide for trajectory inference, but instead learn the dynamics of variably located sub-regions. This design choice enables scalability and flexibility in modeling functional regions across different parts of the screened tissue.

Changes in the number of nodes. To model the temporal evolution and densification of microenvironments, we allow the source and target regions to differ in the number of nodes. We adopt a formulation similar to Section 3.2, implementing OT FM between non-comparable spaces.

Flexible generative models for features and coordinates. We allow flexibility in the choice of generative models for the features and coordinates. To this end, we implement the approach described in Section 3.3, factorizing features and coordinates into separate posteriors from different families.

4.1. Data description and problem statement

We are given a sequence of time-resolved spatial transcriptomic measurements across biological processes such as development or aging. For simplicity, we formulate our problem in terms of two consecutive time points indexed by s, such that $s \in \{0, 1\}$, though the model can be extended to collections of more consecutive discrete temporal measurements. Each dataset at a time point is a full tissue slide that can be represented as an *attributed point cloud*:

$$\mathcal{P}_s = \{(\boldsymbol{c}_i^s, \boldsymbol{x}_i^s) \mid i = 1, \ldots, N_s\}, \tag{8}$$

where $\boldsymbol{c}_i^s \in \mathbb{R}^2$ denotes the 2D spatial coordinate of cell i at time s, and $\boldsymbol{x}_i^s \in \mathbb{R}^D$ denotes its associated feature vector, typically corresponding to its gene expression profile or a low-dimensional representation thereof. Thus, each dataset is an *attributed point set* in two-dimensional space.

To capture spatial context beyond individual cells, we define local *microenvironments* as fixed-radius neighborhoods. Specifically, for each cell $(\boldsymbol{c}_i^s, \boldsymbol{x}_i^s)$ at time s, we construct a neighborhood $\mathcal{M}_i^s$ consisting of all neighboring cells within a spatial radius r:

$$\mathcal{M}_i^s = \left\{(\boldsymbol{c}_j^s, \boldsymbol{x}_j^s) \,\middle|\, \|\boldsymbol{c}_j^s - \boldsymbol{c}_i^s\| \leq r\right\}. \tag{9}$$

Let $\{\mathcal{M}_i^0\}_{i=1}^{N_0}$ and $\{\mathcal{M}_j^1\}_{j=1}^{N_1}$ be collections of source and target microenvironments at consecutive time points. Our goal is to train a parameterized flow model ϕ_t^θ, with $t \in [0, 1]$, that generates target microenvironments conditioned on source point clouds. Specifically, to sample a target microenvironment with k cells conditioned on a variably sized source $\mathcal{M}^0$, we define sampling as:

$$\begin{aligned} \mathcal{M}^z = \{(\boldsymbol{c}_i^z, \boldsymbol{z}_i) \mid \boldsymbol{c}_i^z &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_2), \\ \boldsymbol{z}_i &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D),\ \forall i = 1, \ldots, k\}, \end{aligned} \tag{10}$$

$$\mathcal{M}^1 = \phi_1^\theta(\mathcal{M}^z \mid \mathcal{M}^0), \tag{11}$$

where $\mathcal{M}^z$ is a point cloud composed of noisy coordinates and features, and $\mathcal{M}^1$ is a generated prediction for the evolution of $\mathcal{M}^0$ at the next time point. As explained in Section 3.1 and Section 3.2, we want our generative model to parameterize some notion of optimal entropic coupling $\pi_\epsilon^\star$ between microenvironments across slides (see Section 4.2).

4.2. OT formulation

To train the flow map ϕ_t^θ to perform conditional EOT, we define a cost function that induces an optimal entropic coupling $\pi_\epsilon^\star$ between source and target point clouds. This coupling is used to sample pairs of source and target microenvironment mini-batches during training. While there is no established notion of optimal cost in this setting, we propose to compute OT using source and target microenvironment representations based on the weighted average of features and coordinates.

Specifically, we compute a pooled representation for each microenvironment in the source and target slides by averaging spatial coordinates and gene expression features, weighted by a tunable hyperparameter $\lambda \in [0,1]$ that balances spatial versus cellular state information:

$$\bar{\boldsymbol{m}}_i^s = \left[\frac{1-\lambda}{|\mathcal{M}_i^s|} \sum_{(\boldsymbol{c}_j^s, \boldsymbol{x}_j^s) \in \mathcal{M}_i^s} \boldsymbol{c}_j^s \; \Big\| \; \frac{\lambda}{|\mathcal{M}_i^s|} \sum_{(\boldsymbol{c}_j^s, \boldsymbol{x}_j^s) \in \mathcal{M}_i^s} \boldsymbol{x}_j^s \right], \tag{12}$$

where $\|$ denotes concatenation, $\bar{\boldsymbol{m}}_i^s \in \mathbb{R}^{2+D}$, and $s \in \{0,1\}$. We then apply EOT on the sets $\{\bar{\boldsymbol{m}}_i^0\}_{i=1}^{N_0}$ and $\{\bar{\boldsymbol{m}}_j^1\}_{j=1}^{N_1}$ using a squared Euclidean cost and regularization parameter ϵ, yielding the coupling $\pi^\star_{\epsilon,\lambda}$. During training, we sample matched pairs $(\mathcal{M}^0, \mathcal{M}^1) \sim \pi^\star_{\epsilon,\lambda}$ computed over mini-batches, and use them as supervision for learning the conditional generative model. A higher value of λ prioritizes feature similarity, while lower values favor proximity in coordinates (see Figure 7).

4.3. Mixed-factorized VFM

Once we have established a strategy for performing mini-batch OT, we proceed to describe our approach for learning the flow model ϕ_t^θ and simulating Equation (11). To this end, we adopt a variant of the VFM (Section 3.3), originally developed for graph generation, and adapt it to our point cloud setting.

In line with Section 3.2 and 3.3, we delineate an objective to train a parameterized, source-conditioned posterior over target point clouds $q_t^\theta(\mathcal{M}^1 \mid \mathcal{M}, \mathcal{M}^0)$, where $\mathcal{M}^0$ and $\mathcal{M}^1$ represent source and target niches, and $\mathcal{M}$ denotes a noisy point cloud at interpolation time t. Importantly, our posterior comes with the following characteristics: **(i)** the posterior is factorized across the single points in a point cloud; **(ii)** the posterior is factorized across cellular features and coordinate dimensions; and **(iii)** the family of posteriors can be chosen differently between cellular features and coordinates.

Following **(i)**, we model the variational distribution over $\mathcal{M}^1$ by factorizing it across individual points $(\boldsymbol{c}_1, \boldsymbol{x}_1) \in \mathcal{M}^1$. Moreover, we tackle **(ii)** and **(iii)** using the mean-field VFM assumption, modeling cellular state and positions separately (see Appendix C.1 and C.2 for theoretical justifications):

$$q_t^\theta(\mathcal{M}^1 \mid \mathcal{M}, \mathcal{M}^0) = \prod_{(\boldsymbol{c}_1, \boldsymbol{x}_1) \in \mathcal{M}^1} \prod_{k=1}^{2} \prod_{d=1}^{D} f_t^\theta(c_1^k \mid \mathcal{M}, \mathcal{M}^0) \cdot r_t^\theta(x_1^d \mid \mathcal{M}, \mathcal{M}^0) \,. \tag{13}$$

Here, f_t^θ and r_t^θ denote distinct approximate posterior families for cellular states (x_1^d) and spatial positions (c_1^k), respectively. We use a Laplace distribution for f_t^θ due to its concentration around the mean, which supports precise modeling of coordinate features, while a Gaussian distribution is used for r_t^θ.

As explained in Section 3.3, if one uses FM with straight probability paths, only the first moment of q_t^θ is required to simulate the generative field in Equation (5). Therefore, the posterior is replaced by a time-dependent predictor $(\bar{\boldsymbol{f}}_t^\theta, \bar{\boldsymbol{r}}_t^\theta) = \mu_t^\theta(\mathcal{M}, \mathcal{M}^0)$ of the mean features $\bar{\boldsymbol{r}}_t^\theta$ and coordinates $\bar{\boldsymbol{f}}_t^\theta$ of the target point clouds, learned minimizing the following loss:

$$\mathcal{L}_{\text{NicheFlow}}(\theta) = \mathop{\mathbb{E}}_{\substack{t \sim \mathcal{U}[0,1] \\ (\mathcal{M}^0, \mathcal{M}^1) \sim \pi^\star_{\epsilon,\lambda} \\ \mathcal{M} \sim p_t(\mathcal{M} \mid \mathcal{M}^1)}} \left[\sum_{(\boldsymbol{c}_1, \boldsymbol{x}_1) \in \mathcal{M}^1} \left(\|\boldsymbol{c}_1 - \bar{\boldsymbol{f}}_t^\theta\|_1 + \frac{1}{2} \|\boldsymbol{x}_1 - \bar{\boldsymbol{r}}_t^\theta\|_2^2 \right) \right] . \tag{14}$$

We derive the objective in Appendix C.4 and provide algorithms in Appendix E. Here, μ_t^θ inputs a noisy point cloud $\mathcal{M}$ and a source $\mathcal{M}^0$ and provides a mean prediction vector for coordinates and feature dimension, respectively indicated as $\bar{\boldsymbol{f}}_t^\theta$ and $\bar{\boldsymbol{r}}_t^\theta$. We implement it as a point cloud transformer (see Section 4.4).

We highlight a crucial aspect about Equation (14). While the single dimensions are fully factorized in the predictions from μ_t^θ, every feature's mean is a function of the whole noisy point cloud $\mathcal{M}$ as well as the target microenvironment $\mathcal{M}^0$. In other words, the predictions exploit structural information in the point cloud to predict the individual posterior mean of each dimension. Like most approaches, our method has modeling limitations that we outline in Appendix B.

4.4. Backbone architecture: Microenvironment transformer

To parameterize the conditional posterior mean μ_t^θ from Section 4.3, we use a permutation-invariant transformer architecture for variable-size point clouds in an encoder-decoder layout. The *condition encoder* processes the source microenvironment $\mathcal{M}^0$, while the *decoder* predicts the posterior mean from a noisy target $\mathcal{M} \sim p_t(\cdot \mid \mathcal{M}^1)$, conditioned on the output of the condition encoder. Each target point is represented by features $\boldsymbol{x} \in \mathbb{R}^D$ and spatial coordinates $\boldsymbol{c} \in \mathbb{R}^2$, embedded separately and concatenated, while the time t is encoded via sinusoidal embeddings and broadcast across points. The condition encoder embeds the source via self-attention, and the decoder applies self-attention on the target, followed by cross-attention to the encoded source, allowing each target point to attend to all source points. Decoder outputs are linearly projected to posterior mean estimates $(\bar{\boldsymbol{f}}_t^\theta, \bar{\boldsymbol{r}}_t^\theta)$.

5. Experiments

We propose quantitative and qualitative evaluations of our algorithm. Quantitatively, we test whether source-conditioned samples generated by our model preserve the biological

Table 1. Performance comparison across three biological datasets. Models are trained using Conditional Flow Matching (CFM), Gaussian VFM (GVFM), or Gaussian-Laplacian VFM (GLVFM). Results are reported as mean ± standard deviation over five evaluation runs on mouse embryonic development (MED), axolotl brain development (ABD), and mouse brain aging (MBA). For all experiments, we use a fixed value of $\lambda = 0.1$.

		MED			ABD			MBA		
Model	Obj.	1NN-F1 ↑	PSD ↓ (10^2)	SPD ↓ (10^2)	1NN-F1 ↑	PSD ↓ (10^2)	SPD ↓ (10^2)	1NN-F1 ↑	PSD ↓ (10^2)	SPD ↓ (10^2)
LUNA	—	0.540 ± 0.004	—	—	0.331 ± 0.003	—	—	0.222 ± 0.000	—	—
SPFlow	CFM	0.272 ± 0.001	1.693 ± 0.007	1.173 ± 0.006	0.190 ± 0.001	2.489 ± 0.004	1.715 ± 0.009	0.205 ± 0.000	1.835 ± 0.000	0.726 ± 0.001
	GVFM	0.259 ± 0.002	2.401 ± 0.008	1.137 ± 0.002	0.175 ± 0.001	3.345 ± 0.005	1.657 ± 0.013	0.181 ± 0.000	2.575 ± 0.003	0.735 ± 0.001
	GLVFM	0.251 ± 0.001	2.236 ± 0.004	1.150 ± 0.003	0.174 ± 0.001	2.869 ± 0.011	1.659 ± 0.014	0.194 ± 0.000	2.319 ± 0.004	0.760 ± 0.001
RPCFlow	CFM	0.546 ± 0.002	0.978 ± 0.004	1.110 ± 0.002	0.524 ± 0.001	**2.049 ± 0.004**	1.538 ± 0.007	0.271 ± 0.000	**1.544 ± 0.001**	0.717 ± 0.001
	GVFM	0.502 ± 0.001	1.158 ± 0.005	1.143 ± 0.002	0.478 ± 0.002	2.252 ± 0.002	1.553 ± 0.005	0.249 ± 0.000	1.752 ± 0.001	0.695 ± 0.001
	GLVFM	0.587 ± 0.001	0.978 ± 0.003	1.149 ± 0.002	0.553 ± 0.001	2.053 ± 0.006	1.546 ± 0.006	0.265 ± 0.000	1.722 ± 0.001	**0.691 ± 0.001**
NicheFlow	CFM	0.608 ± 0.004	0.992 ± 0.037	0.954 ± 0.019	0.604 ± 0.001	2.082 ± 0.014	1.252 ± 0.013	0.282 ± 0.001	1.558 ± 0.003	0.733 ± 0.004
	GVFM	0.595 ± 0.005	0.992 ± 0.038	0.987 ± 0.029	0.573 ± 0.002	2.218 ± 0.003	1.253 ± 0.006	0.268 ± 0.000	1.662 ± 0.003	0.705 ± 0.003
	GLVFM	**0.663 ± 0.002**	**0.897 ± 0.021**	**0.824 ± 0.013**	**0.629 ± 0.001**	2.074 ± 0.006	**1.235 ± 0.006**	**0.285 ± 0.001**	1.556 ± 0.001	0.694 ± 0.002

structure and shape of future tissue states. Qualitatively, we demonstrate that our approach accurately captures compositional shifts in substructural components and developmental trajectories across time.

5.1. Quantitative evaluation

Our first research question is to assess the impact of two core modeling choices in NicheFlow: **(i)** learning trajectories over spatial microenvironments, rather than independently for each cell and **(ii)** restricting source and target point clouds to spatially co-localized neighborhoods of cells, instead of sampling them randomly across the slide.

We use NicheFlow and baseline FM approaches that do not incorporate **(i)** and **(ii)** to simulate spatial trajectories conditioned on early time point observations. Assuming that spatial arrangements and the biological composition at later stages evolve from earlier slides, the global correspondence between predicted and true slides indicates the quality of the generative trajectory.

5.1.1. Training setup

Datasets. We assess model performance across three spatiotemporal datasets: **(i)** Mouse embryogenesis (Klein et al., 2025; Chen et al., 2022) and **(ii)** the axolotl brain development (Wei et al., 2022), two Stereo-seq datasets profiling the spatially-resolved cellular development of a mouse embryo and axolotl brain across three (E9.5, E10.5 and E11.5) and five time points, respectively. We also consider the **(iii)** mouse brain aging dataset (Sun et al., 2025), profiled with MERFISH (Chen et al., 2015) across twenty time points (see Appendices F.1 and F.2).

Dataset construction. For each dataset and time point $s \in \mathcal{S}$, we construct a set of cellular microenvironments by applying the fixed-radius neighborhood definition introduced in Section 4.1. Each microenvironment $\mathcal{M}_i^s$ is centered at cell i and contains all cells within a fixed radius r. This results in:

$$\boldsymbol{\mathcal{M}}^s := \{\mathcal{M}_i^s \mid i = 1, \ldots, N_s\},$$

where N_s is the number of cells in the tissue at time s, and each $\mathcal{M}_i^s$ is an attributed point cloud encoding both spatial and gene expression information. We standardize coordinates for cross-time comparability and reduce the normalized gene expression to its top 50 Principal Components (PC).

Batching. We train NicheFlow with mini-batches of source and target cellular point clouds. To ensure spatial diversity during training, we sample individual batches uniformly from within discrete regions of the slides computed with K-Means clustering over the 2D coordinates (see Figure 8 for a visualization with different K values). From these regions, we collect M source and target microenvironments and resample $N \leq M$ matching pairs from the entropic OT coupling $(\boldsymbol{\mathcal{M}}^0, \boldsymbol{\mathcal{M}}^1) \sim \pi^\star_{\epsilon,\lambda}$ as described in Section 3.1, where $\boldsymbol{\mathcal{M}}^0$ and $\boldsymbol{\mathcal{M}}^1$ denote the sampled sets (see Section 4.2 for details on our OT coupling).

Evaluation data. For consistent and reproducible evaluation, we discretize each tissue into a fixed 2D grid and define evaluation microenvironments as fixed-radius neighborhoods around the nearest cells to each grid point. This guarantees full spatial coverage and ensures deterministic comparison across methods. See Appendix F.5 for details.

Multiple time-point. NicheFlow predicts piecewise trajectories between subsequent time points. Instead of learning one flow for each couple of subsequent slides, we train a single model with additional conditioning on source and target labels (see Appendix F.6).

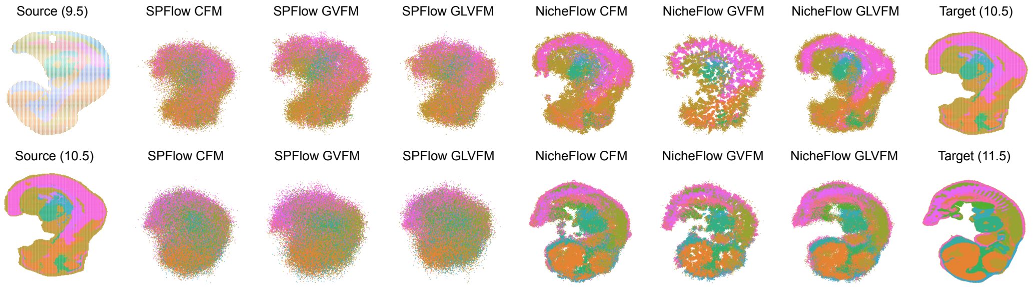


Figure 2. Qualitative comparison of generated samples on the embryonic development dataset (9.5-11.5 days). We show source and target samples alongside predictions from SPFlow and NicheFlow with different objectives.

5.1.2. Quantitative evaluation metrics

Spatial structure. We quantify coordinate generation accuracy using two asymmetric distance metrics. The *point-to-shape distance* (PSD) measures how far predicted coordinates deviate from the true structure by computing the mean squared distance from each generated point to its nearest ground truth counterpart. In contrast, the *shape-to-point distance* (SPD) evaluates how well the generated points cover the target region by averaging the squared distance from each ground truth point to its nearest generated point (see Appendix F.3 for a mathematical formulation of the metrics).

Cell-type organization. To assess how well the model reconstructs the spatial organization of different cell types, we use a *1-nearest-neighbor* (1NN) classification setup. Since the model generates only gene expression profiles and spatial coordinates, we assign cell type labels to generated cells using a classifier trained on ground truth gene expression data (see Appendix F.4). Each predicted cell is then matched to its nearest real cell, and we report the weighted F1 score (1NN-F1).

5.1.3. Models and results

Baselines. We compare against *SPFlow*, a standard FM-based model that predicts temporal trajectories across slides at a single-cell level using an MLP-based velocity field. We also consider *RPCFlow*, which has the same backbone as NicheFlow, but conditions on randomly sampled point clouds instead of radius neighborhoods. We also include *LUNA* (Yu et al., 2025a), a diffusion model for spatial reconstruction from dissociated cells. Note that LUNA does not model temporal dynamics and only generates coordinates from noise with their respective biological annotations. Therefore, we use it as a reference for spatial generation accuracy via the 1NN-F1 metric rather than a proper baseline.

Ablations. We assess different training objectives by comparing standard Conditional Flow Matching (CFM) from (Tong et al., 2024a) with two variational formulations modeling posteriors over coordinates and features: Gaussian-only (GVFM) and Gaussian-Laplace (GLVFM). The former uses Gaussian posteriors for coordinates and features. The latter uses the factorized formulation in Section 4.3. For NicheFlow, we use a fixed $\lambda = 0.1$ and 64 batch sampling regions chosen from the ablations in Appendix D.3 and D.4.

Results. Our quantitative evaluation (Table 1) demonstrates that *NicheFlow* trained with the GLVFM objective consistently achieves strong performance across both spatial and semantic metrics. It outperforms all baselines in reconstructing spatial structure (PSD, SPD) and cell-type organization (1NN-F1) on developmental datasets, while remaining competitive on aging data. These results highlight the importance of structured microenvironment modeling for capturing the spatiotemporal dynamics of complex tissues. We complement our quantitative results visually in Figure 2 and Figure 4 and 5 in the Appendix, where we show that SPFlow fails to capture tissue-level organization, producing blurry and spatially incoherent samples. In contrast, NicheFlow generates predictions that preserve spatial structure and cell-type organization, despite learning only from local microenvironments. In Appendix D.2 and Appendix D.5, we additionally demonstrate that, unlike RPCFlow, NicheFlow produces conditionally consistent outputs with the source.

5.2. Qualitative evaluation and biological analysis

We explore the capabilities of NicheFlow on the spatial trajectory inference task through qualitative and biological assessments. Specifically, we focus on validating whether our model captures compositional changes within fixed spatial structures and developmental trajectories.

Experimental setup. We train the model as described in Section 5.1.1. Using the mouse embryonic dataset (Chen et al., 2022), we select specific microenvironments as source niches for which we want to study the trajectory over time. Specifically, we propose two scenarios for the application of NicheFlow depending on the choice of the OT parameter λ (see Section 4.2):

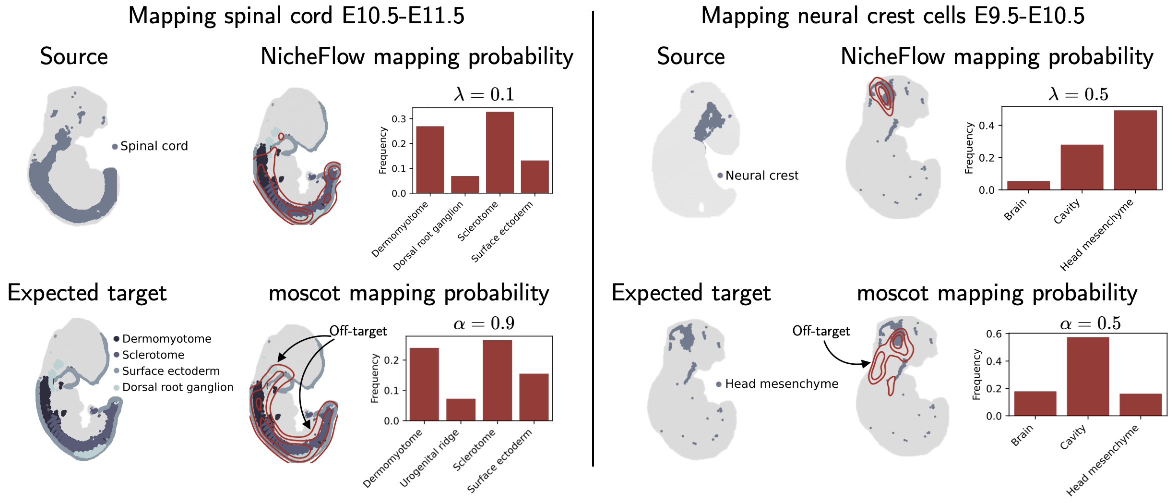


Figure 3. Left and right panels show the mapping of spinal cord (E10.5 to E11.5) and head neural crest cells (E9.5 to E10.5). In each panel, the left column shows source cells and expected targets, and the right column shows density contours of the most likely mapped regions. Bar plots display transition probabilities to the most likely descendant cell types. For NicheFlow, contours represent the proportion of samples in generated point clouds assigned to real cell coordinates across 10 samples.

1. *Compositional changes in fixed structures across time*. We choose the evolution of the *spinal cord* of the embryo from E10.5 to E11.5 as an example and set a high value of λ to prioritize the preservation of the spatial location in the trajectory.
2. *Spatial and cellular development of immature cells*. Developing cells may displace to different areas of the embryo, requiring a higher value of λ to account for gene expression. As a case study, we inspect how neural crest cells in the head evolve into mesenchymal and cranial structures.

Baseline. We compare NicheFlow with the spatiotemporal framework in moscot (Klein et al., 2025), which models spatial trajectories at the single-cell level. In contrast to NicheFlow, moscot integrates spatial coordinates directly into the OT formulation using a Fused Gromov-Wasserstein cost (Titouan et al., 2019), where the hyperparameter α controls the trade-off between spatial and feature-based distances (see Appendix D.6). A high α increases the influence of spatial distances, whereas in our framework this role is played by the hyperparameter λ (see Section 4.2). Notably, moscot relies on exact OT and learns a transition matrix between source and target samples. As such, it is not a generative model over point clouds like NicheFlow. However, given the overlap in downstream tasks, we consider the comparison relevant.

Evaluation and results. For both scenarios (1) and (2), we have prior knowledge of the ground truth regions that the source microenvironments are expected to occupy at later developmental stages, as well as their corresponding biological compositions. For both moscot and NicheFlow, we assess whether the transported mass of source samples concentrates within the correct anatomical region at the target time point, and whether the predicted descendant cell types are biologically consistent (see Appendix D.6). Results are summarized in Figure 3. When modeling the evolution of the spinal cord, moscot assigns considerable mass to unrelated regions such as the urogenital ridge and branchial arches, whereas NicheFlow correctly maps source niches to the maturing spinal cord. Similarly, NicheFlow captures the differentiation of neural crest cells into mesenchymal and cranial tissues within the head region, while moscot exhibits substantial off-target leakage towards lower regions.

6. Conclusions

We introduce NicheFlow, a point-cloud-based generative model designed to capture the spatiotemporal dynamics of cellular niches in time-resolved spatial transcriptomics data. Unlike methods that model single-cell trajectories independently, NicheFlow implicitly captures spatial correlations between cells by learning trajectories on variably sized local neighborhoods. To this end, we combine OT with a new version of VFM that factorizes features and coordinates into distinct posteriors from different distribution families. We showed that NicheFlow outperforms standard FM approaches at reconstructing spatial context from previous time points and improves the mapping of biological structures in time over established exact OT approaches. We envision that NicheFlow will also enable future extensions to perturbation modeling and multi-modal measurements.

Acknowledgments

A.P. is supported by the Helmholtz Association under the joint research school Munich School for Data Science (MUDS). A.P., F.G., S.G. and F.J.T. also acknowledge support from the German Federal Ministry of Education and Research (BMBF) through grant numbers 031L0289A and 031L0289C. F.J.T. acknowledges support from the Helmholtz Association's Initiative and Networking Fund via the CausalCellDynamics project (grant number Interlabs-0029).

References

Abdelaal, T., Grossouw, L. M., Pasterkamp, R. J., Lelieveldt, B. P., Reinders, M. J., and Mahfouz, A. Sirv: Spatial inference of rna velocity at the single-cell resolution. *NAR genomics and bioinformatics*, 6(3):lqae100, 2024.

Albergo, M. S. and Vanden-Eijnden, E. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2023.

Atanackovic, L., Zhang, X., Amos, B., Blanchette, M., Lee, L. J., Bengio, Y., Tong, A., and Neklyudov, K. Meta flow matching: Integrating vector fields on the wasserstein manifold. In *The Thirteenth International Conference on Learning Representations*, 2025.

Briggs, J. A., Weinreb, C., Wagner, D. E., Megason, S., Peshkin, L., Kirschner, M. W., and Klein, A. M. The dynamics of gene expression in vertebrate embryogenesis at single-cell resolution. *Science*, 360(6392), June 2018.

Bryan, J. P., Farhi, S. L., and Cleary, B. Accurate trajectory inference in time-series spatial transcriptomics with structurally-constrained optimal transport. *bioRxiv*, pp. 2025–03, 2025.

Chen, A., Liao, S., Cheng, M., Ma, K., Wu, L., Lai, Y., Qiu, X., Yang, J., Xu, J., Hao, S., Wang, X., Lu, H., Chen, X., Liu, X., Huang, X., Li, Z., Hong, Y., Jiang, Y., Peng, J., Liu, S., Shen, M., Liu, C., Li, Q., Yuan, Y., Wei, X., Zheng, H., Feng, W., Wang, Z., Liu, Y., Wang, Z., Yang, Y., Xiang, H., Han, L., Qin, B., Guo, P., Lai, G., Muñoz-Cánoves, P., Maxwell, P. H., Thiery, J. P., Wu, Q.-F., Zhao, F., Chen, B., Li, M., Dai, X., Wang, S., Kuang, H., Hui, J., Wang, L., Fei, J.-F., Wang, O., Wei, X., Lu, H., Wang, B., Liu, S., Gu, Y., Ni, M., Zhang, W., Mu, F., Yin, Y., Yang, H., Lisby, M., Cornall, R. J., Mulder, J., Uhlén, M., Esteban, M. A., Li, Y., Liu, L., Xu, X., and Wang, J. Spatiotemporal transcriptomic atlas of mouse organogenesis using DNA nanoball-patterned arrays. *Cell*, 185(10):1777–1792.e21, May 2022.

Chen, K. H., Boettiger, A. N., Moffitt, J. R., Wang, S., and Zhuang, X. Spatially resolved, highly multiplexed rna profiling in single cells. *Science*, 348(6233):aaa6090, 2015.

Eijkelboom, F., Bartosh, G., Naesseth, C. A., Welling, M., and van de Meent, J.-W. Variational flow matching for graph generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Eng, C.-H. L., Lawson, M., Zhu, Q., Dries, R., Koulena, N., Takei, Y., Yun, J., Cronin, C., Karp, C., Yuan, G.-C., and Cai, L. Transcriptome-scale super-resolved imaging in tissues by RNA seqFISH+. *Nature*, 568(7751):235–239, March 2019.

Haviv, D., Pooladian, A.-A., Pe'er, D., and Amos, B. Wasserstein flow matching: Generative modeling over families of distributions. *arXiv preprint arXiv:2411.00698*, 2024.

Haviv, D., Remšík, J., Gatie, M., Snopkowski, C., Takizawa, M., Pereira, N., Bashkin, J., Jovanovich, S., Nawy, T., Chaligne, R., et al. The covariance environment defines cellular niches for spatial inference. *Nature Biotechnology*, 43(2):269–280, 2025.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Klein, D., Uscidda, T., Theis, F., and Cuturi, M. Genot: Entropic (gromov) wasserstein flow matching with applications to single-cell genomics. *Advances in Neural Information Processing Systems*, 37:103897–103944, 2024.

Klein, D., Palla, G., Lange, M., Klein, M., Piran, Z., Gander, M., Meng-Papaxanthos, L., Sterr, M., Saber, L., Jing, C., et al. Mapping cells through time and space with moscot. *Nature*, pp. 1–11, 2025.

Larsson, L., Frisén, J., and Lundeberg, J. Spatially resolved transcriptomics adds a new dimension to genomics. *Nature Methods*, 18(1):15–18, January 2021.

Li, K., Li, J., Tao, Y., and Wang, F. stdiff: a diffusion model for imputing spatial transcriptomics through single-cell transcriptomics. *Briefings in Bioinformatics*, 25(3): bbae171, 2024.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Liu, S., Li, X., Gu, Z., Wu, J., Jia, S., Shi, J., Dai, Y., Wu, Y., Yan, H., Zhang, J., You, Y., Xue, X., Liu, L., Lang, J., Wang, X., and Leng, J. Single-cell and spatial transcriptomic profiling revealed niche interactions sustaining growth of endometriotic lesions. *Cell Genomics*, 5(1): 100737, January 2025.

Liu, X., Gong, C., and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023.

Long, W., Liu, T., Xue, L., and Zhao, H. spvelo: Rna velocity inference for multi-batch spatial transcriptomics data. *bioRxiv*, pp. 2025–03, 2025.

Mayr, U., Serra, D., Liberali, P., Klein, A., and Treutlein, B. Exploring single cells in space and time during tissue development, homeostasis and regeneration. *Development*, 146(12), 2019.

Moncada, R., Barkley, D., Wagner, F., Chiodin, M., Devlin, J. C., Baron, M., Hajdu, C. H., Simeone, D. M., and Yanai, I. Integrating microarray-based spatial transcriptomics and single-cell RNA-seq reveals tissue architecture in pancreatic ductal adenocarcinomas. *Nature Biotechnology*, 38(3):333–342, January 2020.

Moses, L. and Pachter, L. Museum of spatial transcriptomics. *Nature Methods*, 19(5):534–546, March 2022.

Pham, D., Tan, X., Balderson, B., Xu, J., Grice, L. F., Yoon, S., Willis, E. F., Tran, M., Lam, P. Y., Raghubar, A., et al. Robust mapping of spatiotemporal trajectories and cell–cell interactions in healthy and diseased tissues. *Nature communications*, 14(1):7739, 2023.

Pooladian, A.-A., Ben-Hamu, H., Domingo-Enrich, C., Amos, B., Lipman, Y., and Chen, R. T. Q. Multisample flow matching: Straightening flows with minibatch couplings. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28100–28127. PMLR, 23–29 Jul 2023.

Ren, J., Zhou, H., Zeng, H., Wang, C. K., Huang, J., Qiu, X., Sui, X., Li, Q., Wu, X., Lin, Z., Lo, J. A., Maher, K., He, Y., Tang, X., Lam, J., Chen, H., Li, B., Fisher, D. E., Liu, J., and Wang, X. Spatiotemporally resolved transcriptomics reveals the subcellular RNA kinetic landscape. *Nature Methods*, 20(5):695–705, April 2023a.

Ren, Y., Huang, Z., Zhou, L., Xiao, P., Song, J., He, P., Xie, C., Zhou, R., Li, M., Dong, X., Mao, Q., You, C., Xu, J., Liu, Y., Lan, Z., Zhang, T., Gan, Q., Yang, Y., Chen, T., Huang, B., Yang, X., Xiao, A., Ou, Y., Su, Z., Chen, L., Zhang, Y., Ju, Y., Zhang, Y., and Wang, Y. Spatial transcriptomics reveals niche-specific enrichment and vulnerabilities of radial glial stem-like cells in malignant gliomas. *Nature Communications*, 14(1), February 2023b.

Rodriques, S. G., Stickels, R. R., Goeva, A., Martin, C. A., Murray, E., Vanderburg, C. R., Welch, J., Chen, L. M., Chen, F., and Macosko, E. Z. Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution. *Science*, 363(6434):1463–1467, March 2019.

Schaar, A. C., Tejada-Lapuerta, A., Palla, G., Gutgesell, R., Halle, L., Minaeva, M., Vornholz, L., Dony, L., Drummer, F., Bahrami, M., and Theis, F. J. Nicheformer: a foundation model for single-cell and spatial omics. *bioRxiv*, pp. 2024–04, April 2024.

Shen, X., Zuo, L., Ye, Z., Yuan, Z., Huang, K., Li, Z., Yu, Q., Zou, X., Wei, X., Xu, P., et al. Inferring cell trajectories of spatial transcriptomics via optimal transport analysis. *Cell Systems*, 16(2), 2025.

Ståhl, P. L., Salmén, F., Vickovic, S., Lundmark, A., Navarro, J. F., Magnusson, J., Giacomello, S., Asp, M., Westholm, J. O., Huss, M., Mollbrink, A., Linnarsson, S., Codeluppi, S., Borg, A., Pontée, F., Costea, P. I., Sahlen, P., Mulder, J., Bergmann, O., Lundeberg, J., and Frisen, J. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, 353(6294): 78–82, July 2016.

Sun, E. D., Zhou, O. Y., Hauptschein, M., Rappoport, N., Xu, L., Navarro Negredo, P., Liu, L., Rando, T. A., Zou, J., and Brunet, A. Spatial transcriptomic clocks reveal cell proximity effects in brain ageing. *Nature*, 638(8049): 160–171, Feb 2025.

Titouan, V., Courty, N., Tavenard, R., and Flamary, R. Optimal transport for structured data with application on graphs. In *International Conference on Machine Learning*, pp. 6275–6284. PMLR, 2019.

Tong, A., FATRAS, K., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Wolf, G., and Bengio, Y. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024a.

Tong, A. Y., Malkin, N., Fatras, K., Atanackovic, L., Zhang, Y., Huguet, G., Wolf, G., and Bengio, Y. 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*, pp. 1279–1287. PMLR, 02–04 May 2024b.

Wagner, D. E., Weinreb, C., Collins, Z. M., Briggs, J. A., Megason, S. G., and Klein, A. M. Single-cell mapping of gene expression landscapes and lineage in the zebrafish embryo. *Science*, 360(6392):981–987, June 2018.

Wan, X., Xiao, J., Tam, S. S. T., Cai, M., Sugimura, R., Wang, Y., Wan, X., Lin, Z., Wu, A. R., and Yang, C. Integrating spatial and single-cell transcriptomics data using deep generative models with spatialscope. *Nature Communications*, 14(1):7848, 2023.

Wei, X., Fu, S., Li, H., Liu, Y., Wang, S., Feng, W., Yang, Y., Liu, X., Zeng, Y.-Y., Cheng, M., Lai, Y., Qiu, X., Wu, L., Zhang, N., Jiang, Y., Xu, J., Su, X., Peng, C., Han, L., Lou, W. P.-K., Liu, C., Yuan, Y., Ma, K., Yang, T., Pan, X., Gao, S., Chen, A., Esteban, M. A., Yang, H., Wang, J., Fan, G., Liu, L., Chen, L., Xu, X., Fei, J.-F., and Gu, Y. Single-cell stereo-seq reveals induced progenitor cells involved in axolotl brain regeneration. *Science*, 377 (6610):eabp9444, 2022.

Yu, T., Ekbote, C., Morozov, N., Fan, J., Frossard, P., d'Ascoli, S., and Brbić, M. Tissue reassembly with generative ai. *bioRxiv*, 2025a.

Yu, T., Ekbote, C., Morozov, N., Fan, J., Frossard, P., d'Ascoli, S., and Brbić, M. Tissue reassembly with generative ai. *bioRxiv*, pp. 2025–02, 2025b.

Zhu, S., Zhu, Y., Tao, M., and Qiu, P. Diffusion generative modeling for spatially resolved gene expression inference from histology images. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zinner, M., Lukonin, I., and Liberali, P. Design principles of tissue organisation: How single cells coordinate across scales. *Current opinion in cell biology*, 67:37–45, 2020.

A. Broader impacts

This work addresses fundamental challenges in spatial transcriptomics by modeling complex spatial and compositional changes in developing tissues. We demonstrate how efficient representations of high-dimensional spatial cellular data can advance the understanding of developmental trajectories and microenvironment dynamics. We anticipate releasing NicheFlow as an open-source, user-friendly tool to enable broad application in spatial biology studies. Given its use with biological data, NicheFlow may also be applied in sensitive contexts involving clinical or patient information.

B. Limitations

Our approach relies on fixed Optimal Transport (OT) feature weighting by a parameter λ during training (see Section 4.2), limiting flexibility at inference and potentially constraining certain biological analyses. Moreover, radius-based niche definitions may also be sub-optimal for small or irregularly shaped microenvironments, where the radius captures excessive spatial context and does not allow fine-grained modeling of the functional region's evolution. Additionally, the model assumes that spatial slides can be aligned with respect to each other in time, and requires normalization-based pre-processing. Future work will be directed towards rotational and translational invariant spatial constraints. Finally, while the learned flow models cell population dynamics, it does not explicitly capture biological events such as division or death.

C. Mixed-Factorized Variational Flow Matching

C.1. Theoretical aspects of Variational Flow Matching

Variational Flow Matching (VFM) (Eijkelboom et al., 2024) relies on the observation that one can write the time-resolved marginal vector field $u_t(\boldsymbol{x})$ in FM as the expected conditional field $u_t(\boldsymbol{x} \mid \boldsymbol{x}_1)$ under the posterior $p_t(\boldsymbol{x}_1 \mid \boldsymbol{x})$ as:

$$u_t(\boldsymbol{x}) = \mathbb{E}_{p_t(\boldsymbol{x}_1|\boldsymbol{x})} \left[u_t(\boldsymbol{x} \mid \boldsymbol{x}_1) \right] \ . \tag{15}$$

Since $u_t(\boldsymbol{x} \mid \boldsymbol{x}_1)$ has a closed form and $u_t(\boldsymbol{x})$ is all that we need to generate the probability path p_t from noise to data, this opens the door to a new interpretation of the objective as a variational inference problem, where we approximate $p_t(\boldsymbol{x}_1 \mid \boldsymbol{x})$ with a variational posterior $q_t^\theta(\boldsymbol{x}_1 \mid \boldsymbol{x})$. In other words, one can optimize the following objective:

$$\mathcal{L}_{\text{VFM}}(\theta) = -\mathbb{E}_{t\sim\mathcal{U}[0,1],\boldsymbol{x}_1\sim p_1(\boldsymbol{x}_1),\boldsymbol{x}\sim p_t(\boldsymbol{x}|\boldsymbol{x}_1)} \left[\log q_t^\theta(\boldsymbol{x}_1 \mid \boldsymbol{x}) \right] \ ,$$

where $p_1(\boldsymbol{x})$ is the data distribution and $p_t(\boldsymbol{x} \mid \boldsymbol{x}_1)$ a straight probability path that maps data to noise. When $u_t(\boldsymbol{x} \mid \boldsymbol{x}_1)$ is linear in $\boldsymbol{x}_1$, this model formulation acquires convenient properties listed below.

Mean parameterization. The expected conditional field under the posterior only depends on the posterior mean:

$$\mathbb{E}_{p_t(\boldsymbol{x}_1|\boldsymbol{x})} \left[u_t(\boldsymbol{x} \mid \boldsymbol{x}_1) \right] = u_t(\boldsymbol{x} \mid \mathbb{E}_{p_t(\boldsymbol{x}_1|\boldsymbol{x})} \left[\boldsymbol{x}_1 \right]) \ ,$$

suggesting that it is sufficient to parameterize the posterior mean to simulate data under the marginal flow. The posterior mean can be regressed against real samples $\boldsymbol{x}_1$ during training, acting as a denoiser.

Equivalence between posterior and approximate posterior formulation. From the previous point, it follows that the expectation of the conditional field is the same under the true and approximate posterior, as long as their first moments match.

Efficient simulation. Given a parameterized posterior mean μ_t^θ, simulating the generative field in Equation (15) is efficient under the linearity condition. For example, in the standard FM setting with straight paths (Lipman et al., 2023), the marginal generative field becomes:

$$u_t(\boldsymbol{x}) = \mathbb{E}_{q_t^\theta(\boldsymbol{x}_1|\boldsymbol{x})} \left[u_t(\boldsymbol{x} \mid \boldsymbol{x}_1) \right] \tag{16}$$

$$= u_t \left(\boldsymbol{x} \mid \mathbb{E}_{q_t^\theta(\boldsymbol{x}_1|\boldsymbol{x})} \left[\boldsymbol{x}_1 \right] \right) \tag{17}$$

$$= \frac{\mu_t^\theta(\boldsymbol{x}) - \boldsymbol{x}}{1-t} \ . \tag{18}$$

which can be easily simulated in the range $t \in [0, 1]$.

C.2. Factorized posterior

Similar to Eijkelboom et al. (2024), in our work we use a fully factorized posterior, where individual dimensions can follow different families of distributions with finite moments (see Section 4.3). Notably, a factorized approximate posterior over $\boldsymbol{x}_1$ is allowed as a choice for q_t^θ, since the only requirement to simulate $u_t(\boldsymbol{x})$ is for $q_t^\theta(\boldsymbol{x}_1 \mid \boldsymbol{x})$ to match the expectation of $p_t(\boldsymbol{x}_1 \mid \boldsymbol{x})$ over $\boldsymbol{x}_1$, irrespectively of higher moments or correlations between factors.

In this regard, it is useful to consider the following proposition.

Proposition C.1. *Let $\boldsymbol{x}_1 \in \mathbb{R}^D$ be a D-dimensional target data point, $p_t(\boldsymbol{x}_1 \mid \boldsymbol{x})$ the posterior probability path conditioned on a noisy point $\boldsymbol{x} \sim p_t(\boldsymbol{x})$, and $u_t(\boldsymbol{x} \mid \boldsymbol{x}_1)$ the conditional velocity field. Assume that $u_t(\boldsymbol{x} \mid \boldsymbol{x}_1)$ is linear in $\boldsymbol{x}_1$. Then, for any dimension $d \in \{1, \ldots, D\}$, the following holds:*

$$\mathbb{E}_{p_t(\boldsymbol{x}_1|\boldsymbol{x})}[x_1^d] = \mathbb{E}_{p_t(x_1^d|\boldsymbol{x})}[x_1^d] \tag{19}$$

$$u_t(x^d) = u_t\left(x^d \mid \mathbb{E}_{p_t(x_1^d|\boldsymbol{x})}[x_1^d]\right) , \tag{20}$$

where x^d refers to the d^{th} scalar dimension of the vector $\boldsymbol{x}$.

Proof. We begin by proving Equation (19) using marginalization:

$$\begin{aligned}\mathbb{E}_{p_t(\boldsymbol{x}_1|\boldsymbol{x})}[x_1^d] &= \int x_1^d \, p_t(\boldsymbol{x}_1 \mid \boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}_1 \\ &= \int x_1^d \left(\int p_t(\boldsymbol{x}_1 \mid \boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}_1^{\setminus d} \right) \mathrm{d}x_1^d \\ &= \int x_1^d \, p_t(x_1^d \mid \boldsymbol{x}) \, \mathrm{d}x_1^d = \mathbb{E}_{p_t(x_1^d|\boldsymbol{x})}[x_1^d] \, . \end{aligned} \tag{21}$$

In the vector formulation, the notation $\mathbb{E}_{p_t(\boldsymbol{x}_1|\boldsymbol{x})}[x_1^d]$ implicitly omits the multiple integrals over all components of $\boldsymbol{x}_1$. The step-by-step derivation below makes these integrals explicit for clarity.

Next, we prove Equation (20). Under the assumption that the conditional velocity field $u_t(\boldsymbol{x} \mid \boldsymbol{x}_1)$ is linear in $\boldsymbol{x}_1$, we have:

$$\begin{aligned} u_t(x^d) &= \mathbb{E}_{p_t(\boldsymbol{x}_1|\boldsymbol{x})} \left[u_t(x^d \mid \boldsymbol{x}_1) \right] \\ &\overset{(1)}{=} \mathbb{E}_{p_t(\boldsymbol{x}_1|\boldsymbol{x})} \left[u_t(x^d \mid x_1^d) \right] \\ &= \mathbb{E}_{p_t(\boldsymbol{x}_1|\boldsymbol{x})} \left[\frac{x_1^d - x^d}{1-t} \right] \\ &= \frac{\mathbb{E}_{p_t(\boldsymbol{x}_1|\boldsymbol{x})}[x_1^d] - x^d}{1-t} \\ &\overset{Equation\ (21)}{=} \frac{\mathbb{E}_{p_t(x_1^d|\boldsymbol{x})}[x_1^d] - x^d}{1-t} \\ &= u_t\left(x^d \mid \mathbb{E}_{p_t(x_1^d|\boldsymbol{x})}[x_1^d]\right) \, . \end{aligned} \tag{22}$$

In other words, the expected value under the posterior at an individual feature d does not depend on the other features and has an influence only on the d-th dimension of the conditional vector field. This flexibility allows each dimension's approximate posterior to be chosen from a potentially different distributional family, as long as the first moment exists and is correctly parameterized.

C.3. Marginal field derivation in source-conditioned VFM

When applying source conditioning to VFM, the marginal conditional vector field given a source $\boldsymbol{x}_0$ is:

$$u_t(\boldsymbol{x} \mid \boldsymbol{x}_0) = \int u_t(\boldsymbol{x} \mid \boldsymbol{x}_1) \, \frac{p_t(\boldsymbol{x} \mid \boldsymbol{x}_1) \, \pi(\boldsymbol{x}_1 \mid \boldsymbol{x}_0)}{p_t(\boldsymbol{x} \mid \boldsymbol{x}_0)} \, \mathrm{d}\boldsymbol{x}_1 \tag{23}$$

where the $p_t(\boldsymbol{x} \mid \boldsymbol{x}_1)$ is a probability path interpolating observations $\boldsymbol{x}_1$ with noise. Note that we omit $\boldsymbol{x}_0$ from the probability path and conditional velocity as they are fully determined by $\boldsymbol{x}_1$ under linear conditional probability paths. Hence, $\boldsymbol{x}_0$ controls the field biasing the choice of $\boldsymbol{x}_1$ under the conditional OT coupling π.

Furthermore, we can rewrite the marginal as an expectation:

$$\int u_t(\boldsymbol{x} \mid \boldsymbol{x}_1) \frac{p_t(\boldsymbol{x} \mid \boldsymbol{x}_1)\, \pi(\boldsymbol{x}_1 \mid \boldsymbol{x}_0)}{p_t(\boldsymbol{x} \mid \boldsymbol{x}_0)} \,\mathrm{d}\boldsymbol{x}_1 = \mathbb{E}_{p_t(\boldsymbol{x}_1 \mid \boldsymbol{x}, \boldsymbol{x}_0)} \left[u_t(\boldsymbol{x} \mid \boldsymbol{x}_1) \right] \tag{24}$$

where we used that $p_t(\boldsymbol{x} \mid \boldsymbol{x}_1) = p_t(\boldsymbol{x} \mid \boldsymbol{x}_0, \boldsymbol{x}_1)$.

C.4. Gaussian and Laplacian Mixed-Factorized VFM

We define a hybrid Variational Flow Matching (VFM) model using a fully factorized variational distribution over individual points in the target microenvironment $\mathcal{M}^1$. Following the mean-field assumption, the variational distribution factorizes over spatial and feature dimensions:

$$\begin{aligned} q_t^\theta(\mathcal{M}^1 \mid \mathcal{M}, \mathcal{M}^0) &= \prod_{(\boldsymbol{c}_1, \boldsymbol{x}_1) \in \mathcal{M}^1} q_t^\theta(\boldsymbol{c}_1, \boldsymbol{x}_1 \mid \mathcal{M}, \mathcal{M}^0) \qquad (25) \\ &= \prod_{(\boldsymbol{c}_1, \boldsymbol{x}_1) \in \mathcal{M}^1} \left(\prod_{k=1}^{2} f_t^\theta(c_1^k \mid \mathcal{M}, \mathcal{M}^0) \cdot \prod_{d=1}^{D} r_t^\theta(x_1^d \mid \mathcal{M}, \mathcal{M}^0) \right) \qquad (26) \end{aligned}$$

In line with Eijkelboom et al. (2024), using FM with straight paths enables us to efficiently simulate the marginal generating field using the *first moment* of the posterior distribution. In other words, for a fully factorized posterior, we only need to parameterize a mean predictor. In our setting, the mean prediction is a neural network μ_t^θ as a time-condition function of a noisy microenvironment $\mathcal{M}$ and a source $\mathcal{M}^0$ with outputs:

$$(\bar{\boldsymbol{f}}_t^\theta, \bar{\boldsymbol{r}}_t^\theta) = \mu_t^\theta(\mathcal{M}, \mathcal{M}^0)$$

where $\bar{f}_t^{\theta,k}$ and $\bar{r}_t^{\theta,d}$ are predictions for the expected values for the k^{th} coordinate and d^{th} cell feature. Then, we choose a parameterization for the variational factors at time $t \in [0, 1]$ as follows:

$$\begin{aligned} x_1^d &\sim \mathcal{N}(\bar{r}_t^{\theta,d}, 1), \qquad (27) \\ c_1^k &\sim \text{Laplace}(\bar{f}_t^{\theta,k}, 1), \qquad (28) \end{aligned}$$

Substituting into the negative log-likelihood yields:

$$\begin{aligned} &- \log(q_t^\theta(\mathcal{M}^1 \mid \mathcal{M}, \mathcal{M}^0)) \qquad (29) \\ &= - \log \left(\prod_{(\boldsymbol{c}_1, \boldsymbol{x}_1) \in \mathcal{M}^1} \left(\prod_{k=1}^{2} f_t^\theta(c_1^k \mid \mathcal{M}, \mathcal{M}^0) \cdot \prod_{d=1}^{D} r_t^\theta(x_1^d \mid \mathcal{M}, \mathcal{M}^0) \right) \right) \qquad (30) \\ &= \sum_{(\boldsymbol{c}_1, \boldsymbol{x}_1) \in \mathcal{M}^1} \left(\sum_{k=1}^{2} \left(\log 2 + |c_1^k - \bar{f}_t^{\theta,k}| \right) + \sum_{d=1}^{D} \left(\frac{1}{2} \log(2\pi) + \frac{1}{2} \left(x_1^d - \bar{r}_t^{\theta,d} \right)^2 \right) \right) \\ &= \sum_{(\boldsymbol{c}_1, \boldsymbol{x}_1) \in \mathcal{M}^1} \left(\|\boldsymbol{c}_1 - \bar{\boldsymbol{f}}_t^\theta\|_1 + \frac{1}{2} \|\boldsymbol{x}_1 - \bar{\boldsymbol{r}}_t^\theta\|_2^2 \right) + \text{const w.r.t. } \theta \qquad (31) \end{aligned}$$

This results in a loss consisting of an ℓ_1 error on spatial coordinates and a mean squared error on gene expression features, consistent with the hybrid variational design.

D. Additional results

D.1. Qualitative results

We provide additional qualitative results to support our experimental evaluation. Specifically, our Appendix contains the following additional evidence:

- Qualitative generation of the axolotl brain development and mouse brain aging trajectories.
- Evidence on the correct modeling of anatomical structures and cell type composition on the axolotl brain development dataset.
- Visual depiction of the discrete coupling used for mini-batch OT using different values of λ.
- Ablation studies on the number of discretized regions used for the uniform microenvironment sampling and the hyperparameter λ, which controls the importance of the feature versus the coordinate component in the static OT matching between mini-batches.
- Comparison between microenvironment-based and random sampling strategies.
- Comparisons with moscot on previously unconsidered structures on the mouse embryonic data.
- Evidence of the effectiveness of source conditioning on the moscot dataset.

D.1.1. Slide reconstruction in the axolotl brain development and mouse brain aging datasets

We provide additional visualizations of the generated samples on the axolotl brain development dataset, presented in Figure 4, and on the mouse brain aging dataset, presented in Figure 5.

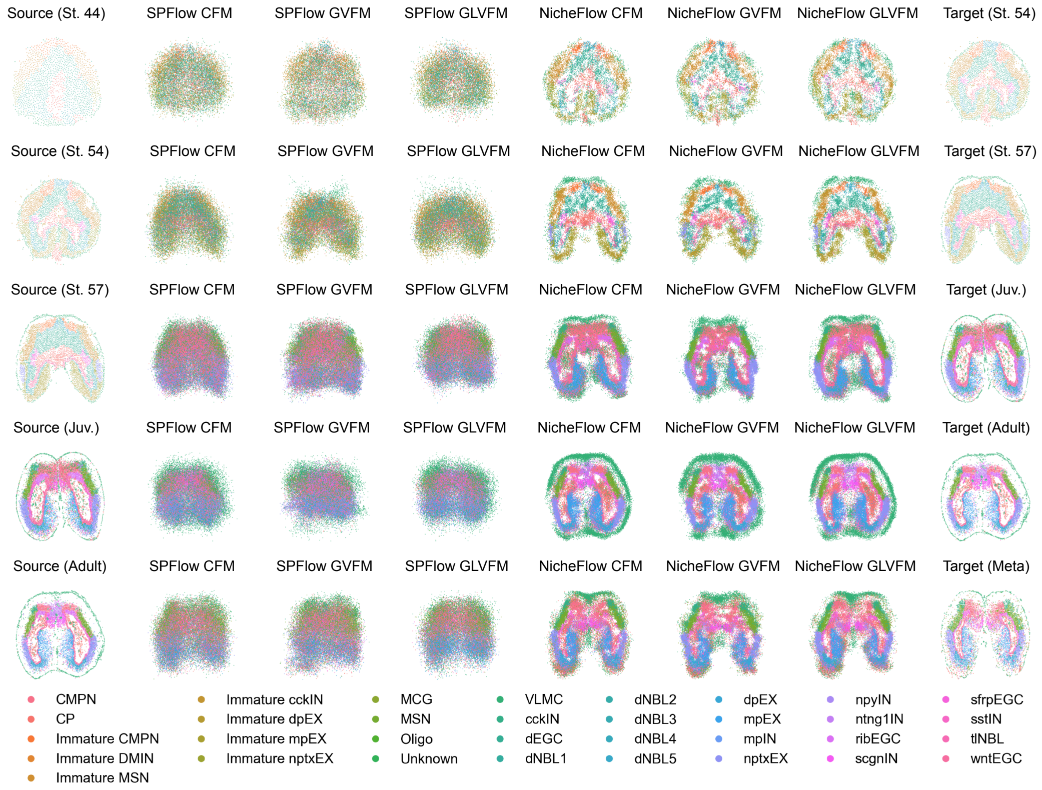


Figure 4. Qualitative comparison of generated samples on the axolotl brain development dataset (Stage 44, 54, 57, Juvenile, Adult, Meta). We show source and target samples alongside predictions from SPFlow and NicheFlow with different objectives. NicheFlow captures the spatial structure and cell-type organization more faithfully across developmental stages.

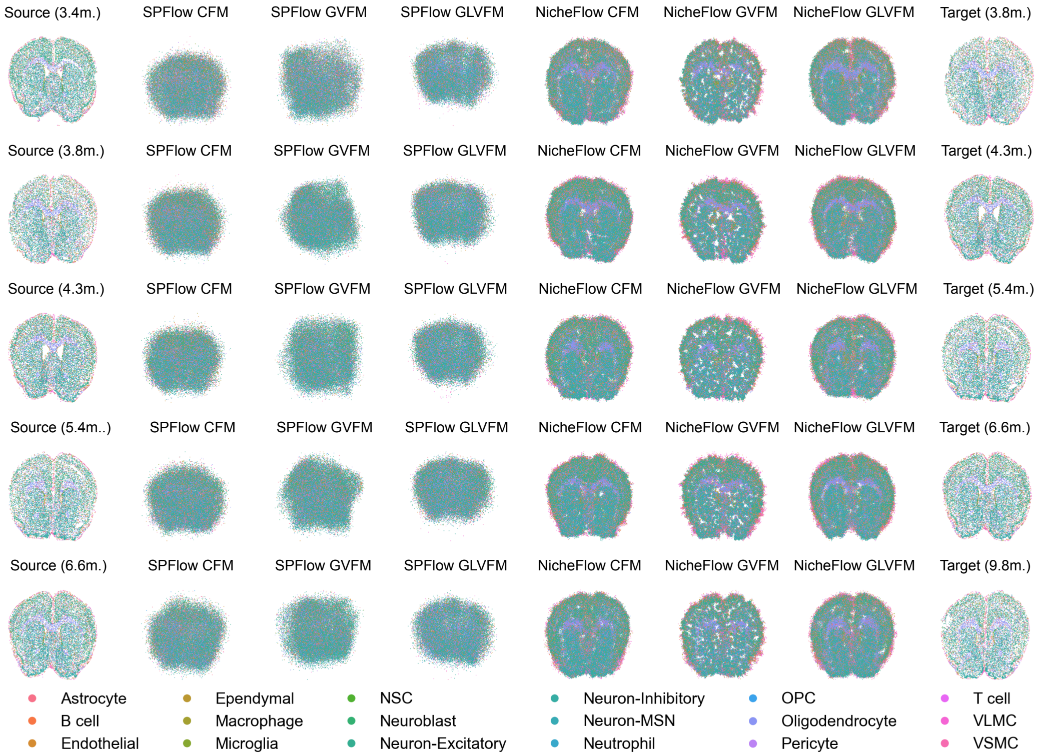


Figure 5. Qualitative comparison of generated samples on the mouse brain aging dataset (3.4, 3.8, 4.3, 5.4, 6.6, 9.8 months). We show source and target samples alongside predictions from SPFlow and NicheFlow with different objectives. NicheFlow captures the spatial structure and cell-type organization more faithfully across developmental stages.

Similar to Figure 2, in Figure 4 and Figure 5, we qualitatively show that NicheFlow with microenvironment sampling strategy and mixed-factorized VFM is the best approach for reconstructing mouse brain trajectories in time.

D.2. Conditional generation

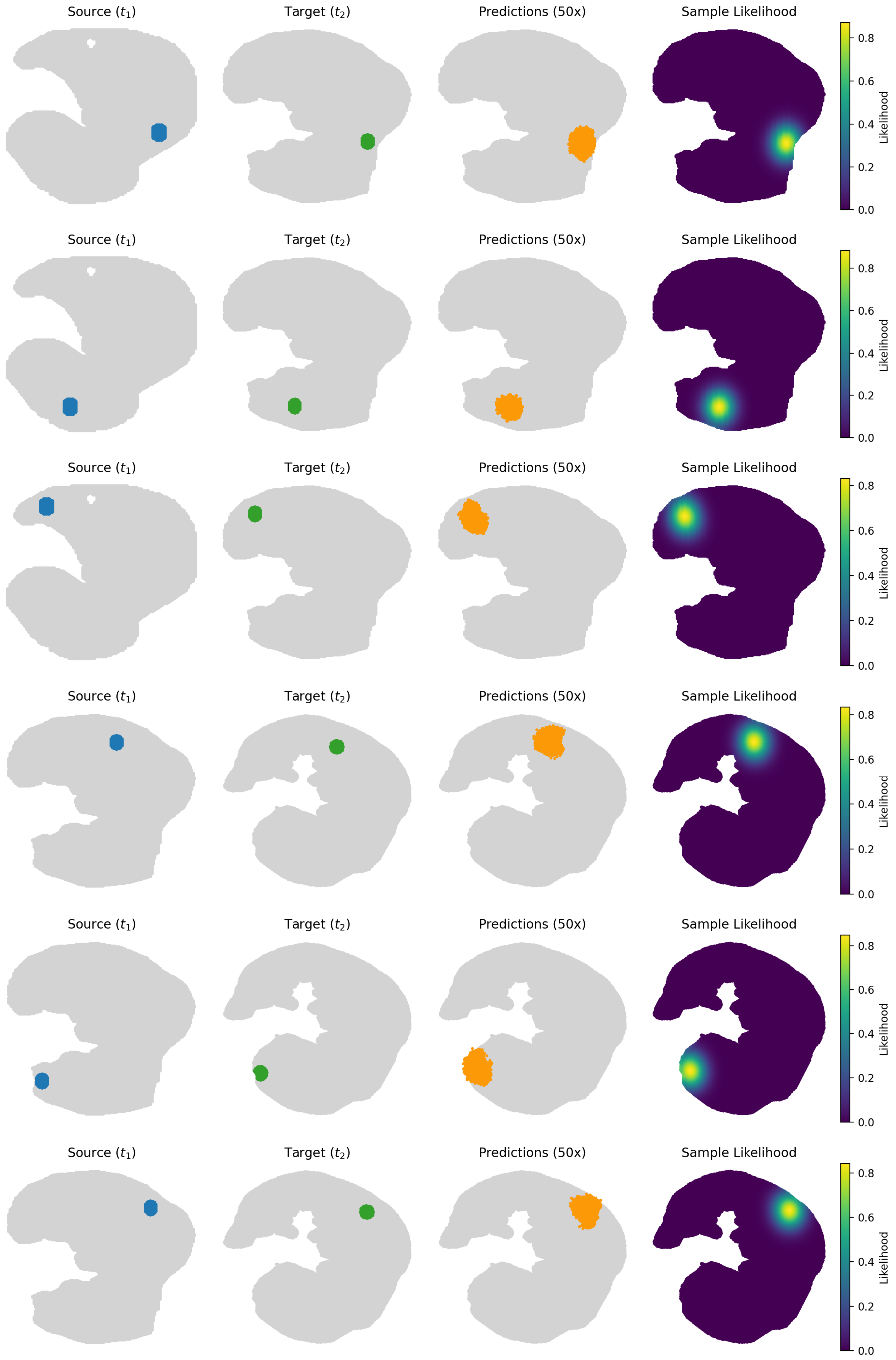


Figure 6. Qualitative evaluation of conditional generation with NicheFlow on the embryonic development dataset. Each row corresponds to a different source microenvironment at time t_1 (blue), shown alongside the ground truth target microenvironment at time t_2 (green). The third column displays 50 samples (orange) generated conditionally by the model, while the fourth column visualizes a kernel density estimate of the sample likelihood over the spatial domain. We use $\lambda = 0.1$ to prioritize coordinate preservation, as it provides better interpretable evidence of actual source conditioning.

To assess whether the model accurately conditions on input microenvironments, we visualize the spatial distribution of generated samples given a fixed source microenvironment. Figure 6 illustrates such cases on the embryonic development dataset for $\lambda = 0.1$, showing the input region at time t_1, the corresponding target at time t_2, and 50 independently generated samples from NicheFlow. Directly computing the likelihood of ground truth cell coordinates under our generative model is intractable, as it would require evaluating the density of an implicitly defined distribution over point clouds. Instead, we approximate the spatial likelihood using kernel density estimation (KDE) over Monte Carlo samples drawn from the model. Given a set of generated coordinates $\{\hat{\boldsymbol{c}}_i\}_{i=1}^N$, we estimate the likelihood at a ground truth location $\boldsymbol{c}$ as:

$$\hat{p}(\boldsymbol{c}) = \frac{1}{N}\sum_{i=1}^{N} \exp\left(-\frac{\|\boldsymbol{c} - \hat{\boldsymbol{c}}_i\|^2}{2\sigma^2}\right), \tag{32}$$

where σ is a fixed bandwidth parameter. The resulting KDE heatmap visualizes the spatial concentration of source-conditioned generated samples, allowing us to qualitatively assess whether the model produces consistent predictions conditioned on the source microenvironment.

D.3. OT Ablation Study

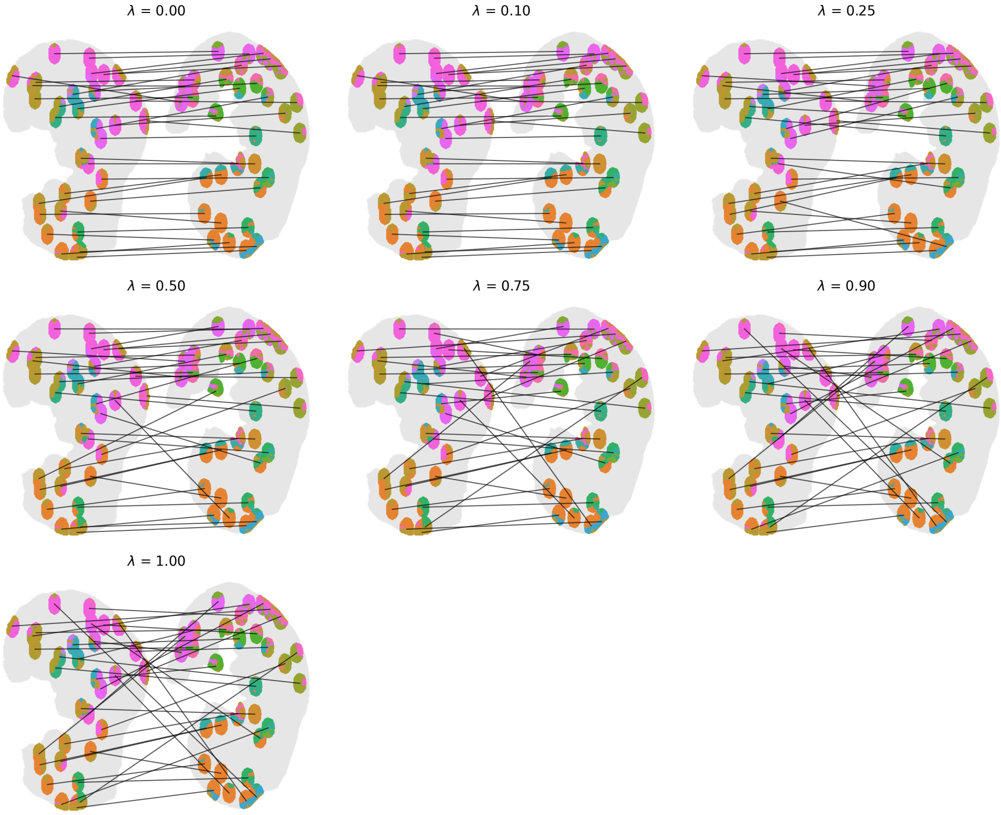


Figure 7. Visualization of OT couplings computed under varying values of the pooling parameter λ in eq. (12), which balances spatial coordinates and gene expression in microenvironment matching. Lower λ values prioritize spatial proximity, resulting in more dispersed and less structured alignments, while intermediate values yield tighter, biologically consistent mappings. Very high λ settings ignore spatial context and may lead to implausible long-range matches.

We begin by emphasizing that the optimal choice of the OT feature-weighting parameter λ in Equation (12) depends on the downstream application. This parameter determines the relative importance assigned to gene expression versus spatial coordinates during optimal transport. For developmental processes such as organogenesis or regeneration, higher values of λ prioritize transcriptional similarity, facilitating the reconstruction of continuous differentiation trajectories and capturing fate-driven transitions, which may span large spatial distances. In contrast, for applications that aim to monitor changes within a fixed spatial region, such as shifts in cell-type composition over time, a lower λ is more appropriate. In such cases, emphasizing spatial locality helps avoid spurious long-range transport assignments based on transcriptomic similarity.

Table 2. Ablation study of the feature weighting factor λ in eq. (12). We evaluate NicheFlow with the GLVFM objective across three datasets: mouse embryonic development (MED), axolotl brain development (ABD), and mouse brain aging (MBA). Results are reported as mean ± standard deviation over five evaluation runs.

λ	MED			ABD			MBA		
	1NN-F1 ↑	PSD ↓ (10^2)	SPD ↓ (10^2)	1NN-F1 ↑	PSD ↓ (10^2)	SPD ↓ (10^2)	1NN-F1 ↑	PSD ↓ (10^2)	SPD ↓ (10^2)
1.0	0.589 ± 0.003	0.916 ± 0.021	0.991 ± 0.039	0.605 ± 0.002	2.086 ± 0.009	1.461 ± 0.029	**0.287 ± 0.000**	1.563 ± 0.003	0.818 ± 0.005
0.9	0.596 ± 0.006	0.928 ± 0.024	1.044 ± 0.054	0.591 ± 0.002	2.149 ± 0.006	1.412 ± 0.020	0.282 ± 0.000	1.577 ± 0.005	0.825 ± 0.010
0.75	0.588 ± 0.006	0.881 ± 0.009	1.101 ± 0.028	0.602 ± 0.002	2.108 ± 0.009	1.372 ± 0.013	0.281 ± 0.000	1.575 ± 0.003	0.792 ± 0.009
0.5	0.611 ± 0.004	0.912 ± 0.007	0.982 ± 0.023	0.608 ± 0.002	2.147 ± 0.007	1.315 ± 0.011	0.282 ± 0.000	1.602 ± 0.002	0.726 ± 0.005
0.25	0.628 ± 0.003	0.923 ± 0.007	0.862 ± 0.027	0.617 ± 0.003	2.106 ± 0.003	1.300 ± 0.021	0.283 ± 0.000	1.600 ± 0.002	0.728 ± 0.002
0.1	**0.663 ± 0.002**	0.897 ± 0.021	**0.824 ± 0.013**	**0.629 ± 0.001**	**2.074 ± 0.006**	1.235 ± 0.006	0.285 ± 0.001	1.556 ± 0.001	**0.694 ± 0.002**
0.0	0.649 ± 0.002	**0.880 ± 0.011**	0.836 ± 0.008	0.617 ± 0.001	2.127 ± 0.003	**1.202 ± 0.008**	0.284 ± 0.000	**1.552 ± 0.002**	0.746 ± 0.008

Here, however, our goal is not to tailor λ to a specific biological use case, but rather to quantitatively assess which setting yields the best performance under our chosen evaluation metrics. To this end, we systematically evaluate NicheFlow using the GLVFM objective across all datasets (see Table 2).

We observe that reducing λ from 1.0 generally improves performance, with optimal results typically achieved around $\lambda = 0.1$. This value offers a balance between preserving spatial coherence and capturing meaningful gene expression variation. Notably, setting $\lambda = 0.0$ corresponds to pure spatial matching. While this minimizes spatial distortion and yields low PSD, it neglects transcriptional identity, potentially leading to biologically implausible alignments, particularly in developmental contexts where spatially adjacent cells may follow divergent fates. Conversely, $\lambda = 1.0$ disregards spatial structure, often resulting in unrealistic, long-range assignments that obscure local tissue architecture.

Figure 7 visualizes the impact of varying λ on the learned OT plans. Intermediate values yield more coherent and biologically plausible couplings. Based on these observations, we adopt $\lambda = 0.1$ as the default setting for the comparison with competing training strategies, as it offers a compromise suitable across multiple datasets and biological contexts.

D.4. K-Means regions ablation study

To ensure diverse and spatially distributed sampling during training, we partition each tissue section into K spatial regions using K-Means clustering over the 2D cell coordinates (see Section 5.1.1). At each training step, microenvironments are sampled uniformly from within these regions, encouraging broad spatial coverage and preventing oversampling of densely populated areas. In this ablation, we investigate how varying the number of spatial regions K affects model performance.

As shown in Figure 8, increasing K leads to increasingly fine-grained spatial partitions. While moderate values of K help improve spatial resolution, excessively high values (e.g., $K = 128$ or 256) result in overly small and fragmented regions. This can cause significant overlap between sampled microenvironments and reduce sampling diversity. Moreover, in sparsely populated tissue sections, high K values may yield regions with insufficient cells, degrading both representativeness and stability.

Table 3. Ablation study of the number of spatial regions K defined over the datasets. We evaluate NicheFlow with the GLVFM objective across three datasets: mouse embryonic development (MED), axolotl brain development (ABD), and mouse brain aging (MBA). Results are reported as mean ± standard deviation over five evaluation runs.

K	MED			ABD			MBA		
	1NN-F1 ↑	PSD ↓ (10^2)	SPD ↓ (10^2)	1NN-F1 ↑	PSD ↓ (10^2)	SPD ↓ (10^2)	1NN-F1 ↑	PSD ↓ (10^2)	SPD ↓ (10^2)
8	0.640 ± 0.005	**0.870 ± 0.003**	0.980 ± 0.036	0.619 ± 0.003	1.953 ± 0.008	1.369 ± 0.008	0.283 ± 0.001	1.561 ± 0.003	0.766 ± 0.004
16	0.660 ± 0.003	0.879 ± 0.005	0.862 ± 0.018	**0.633 ± 0.001**	**1.936 ± 0.005**	1.378 ± 0.031	0.282 ± 0.001	1.566 ± 0.003	**0.703 ± 0.003**
32	0.658 ± 0.004	0.901 ± 0.009	**0.803 ± 0.018**	0.623 ± 0.001	1.968 ± 0.007	1.339 ± 0.012	0.279 ± 0.001	1.599 ± 0.002	0.703 ± 0.004
64	**0.663 ± 0.002**	0.897 ± 0.021	0.824 ± 0.013	0.629 ± 0.001	2.074 ± 0.006	**1.235 ± 0.006**	**0.285 ± 0.001**	**1.556 ± 0.001**	**0.694 ± 0.002**

Conversely, using too few regions (e.g., $K = 4$) results in broad spatial partitions that may encompass multiple heterogeneous tissue compartments. This undermines the locality assumptions of our model and increases intra-region variability, which can impair the model's ability to learn.

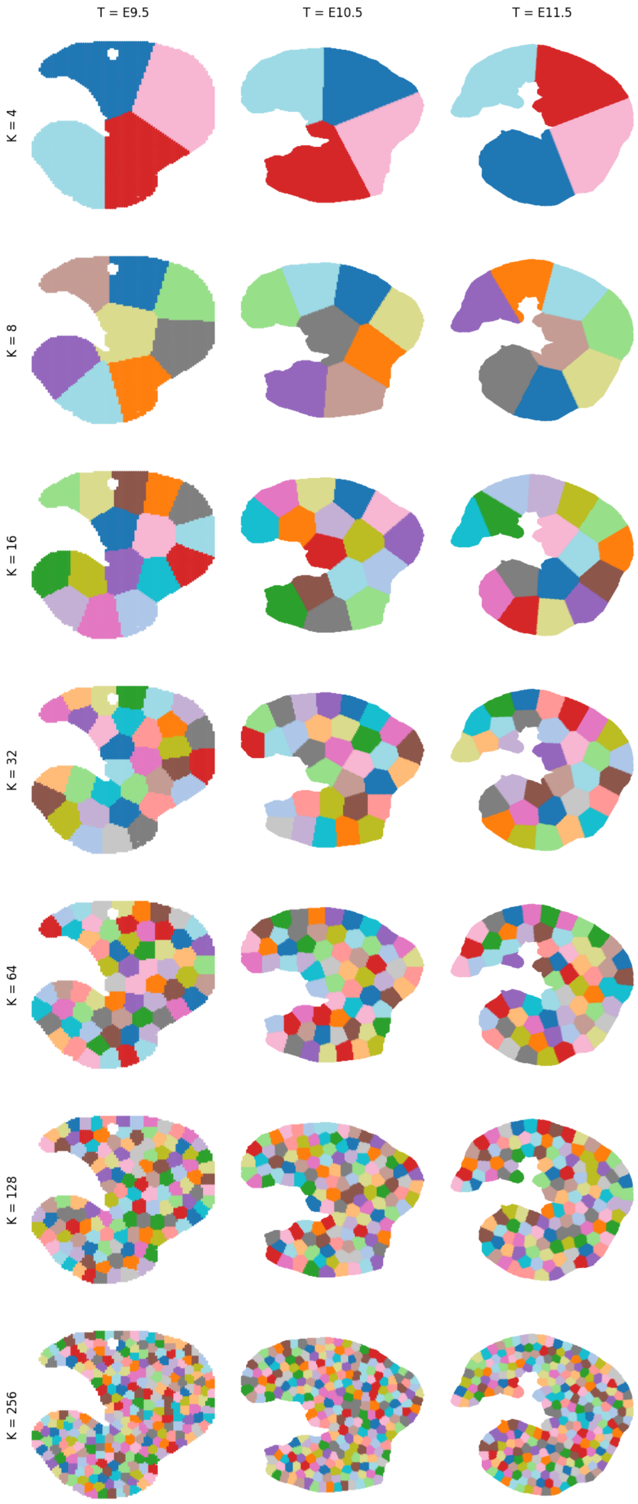


Figure 8. Visualization of spatial partitions obtained via KMeans clustering with $K = \{4, 8, 16, 32, 64, 128, 256\}$ on the embryonic development dataset. For low K, each region covers large heterogeneous areas; for high K, regions become small, dense, and highly overlapping, potentially degrading the diversity and utility of sampled microenvironments.

Given the trade-offs outlined above, we focus our evaluation on $K \in \{8, 16, 32, 64\}$, which spans a range of granularities that preserve both spatial interpretability and sampling robustness. Table 2 summarizes the quantitative results for this ablation across three datasets: Mouse embryonic development (MED), axolotl brain development (ABD), and mouse brain aging (MBA). We observe that using $K = 64$ consistently yields strong performance, achieving the highest 1NN-F1 scores on both the MED and MBA datasets, while also performing competitively on ABD. These findings indicate that $K = 64$ offers an effective balance between spatial resolution and stability, and we adopt it as the default configuration in our main experiments.

D.5. Justifying the choice of NicheFlow over RPCFlow

Although RPCFlow achieves competitive performance on spatial metrics such as PSD and SPD, it lacks the essential capability of meaningful conditional generation. In RPCFlow, conditioning is performed using randomly sampled point clouds from the spatial regions without explicit microenvironment structure. As shown in Figure 9, when conditioned on such random sources, RPCFlow tends to reconstruct the entire target tissue, rather than capturing local dynamics driven by the source input. This undermines its ability to model spatiotemporal evolution in a biologically grounded manner.

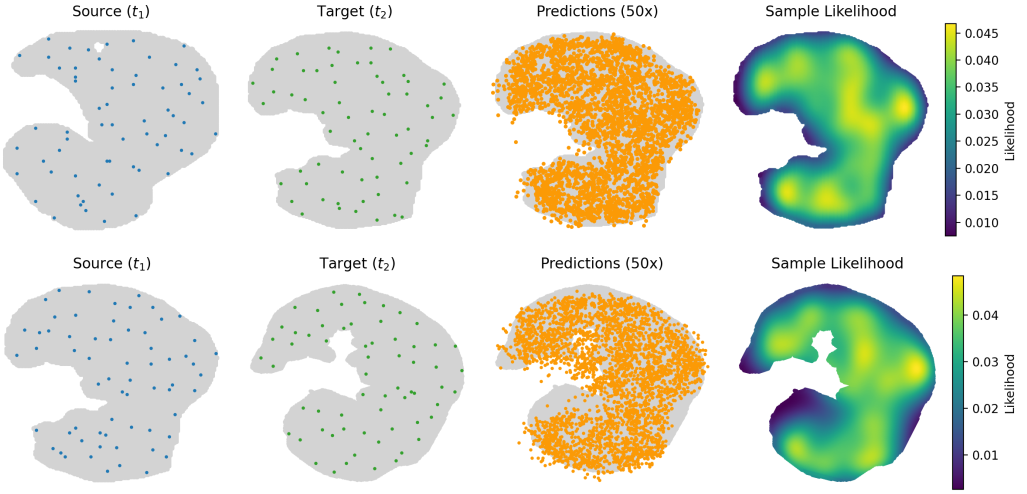
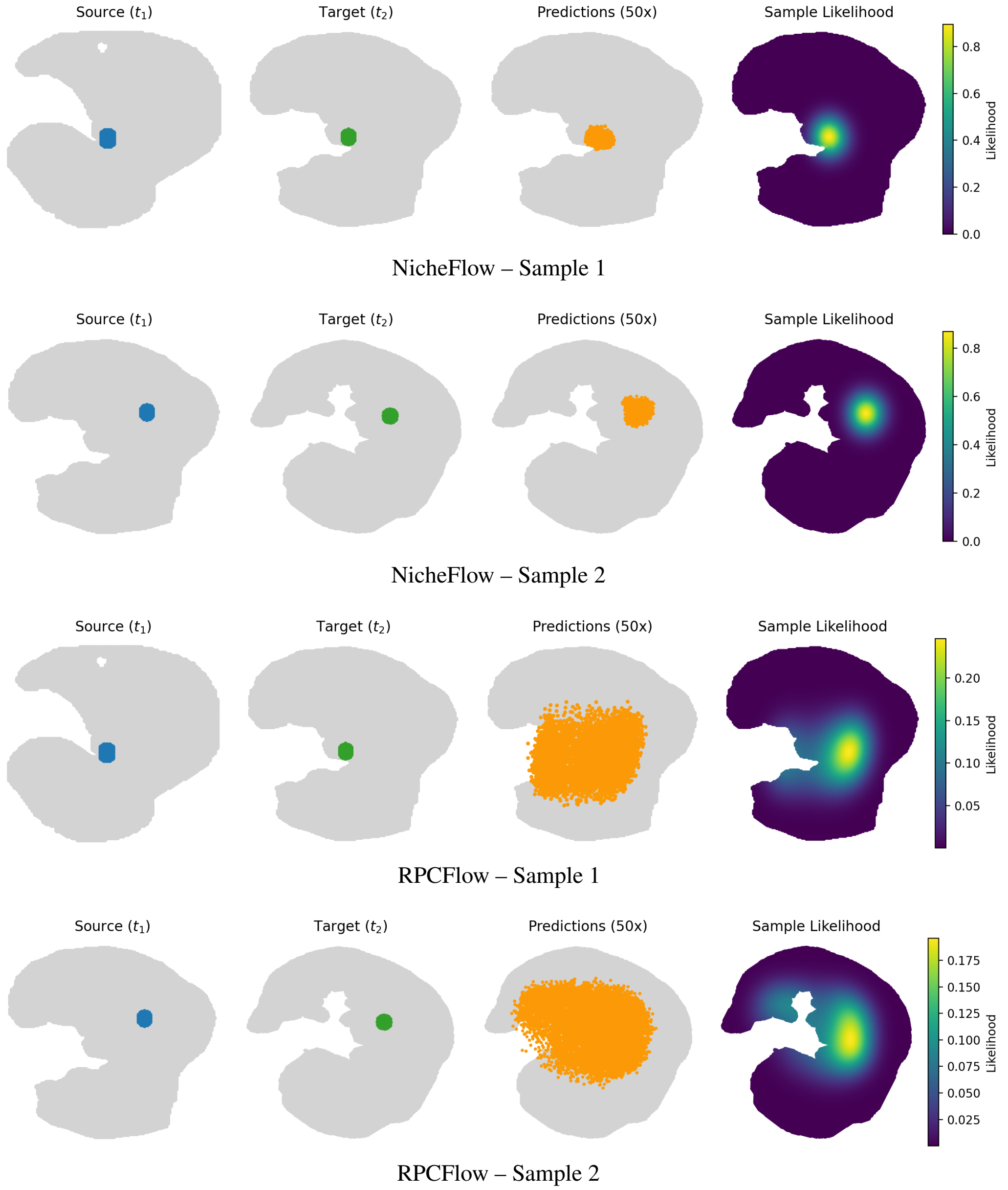


Figure 9. RPCFlow fails to perform meaningful conditional generation: the model generates a diffuse reconstruction resembling the entire target tissue rather than conditioning on the provided source points. Each row shows a different source-target pair.

In contrast, NicheFlow is explicitly designed to push localized microenvironments through time. By conditioning on fixed-radius neighborhoods centered around specific spatial positions, the model learns how cellular contexts evolve, preserving both spatial coherence and transcriptional identity. Figure 10 visualizes this distinction: While NicheFlow consistently generates well-localized predictions aligned with the input microenvironment, RPCFlow fails to retain spatial specificity, often diffusing the prediction across broader regions.

From a biological perspective, predicting the fate of a local tissue region over time is far more relevant than mapping random point sets. Microenvironments encode structured cellular contexts, such as signaling interactions or niche-specific cell states, that are crucial for downstream analysis (e.g., lineage fate prediction, microenvironment-based intervention simulation). Because RPCFlow lacks this interpretability and fails to enable such downstream tasks, it cannot serve as a practical generative model in biological settings.

In summary, while RPCFlow may appear performant under some aggregate spatial metrics, only NicheFlow enables localized, conditionally consistent generation of evolving tissue regions. This makes it not only superior for evaluation but also for practical use in biological modeling.



NicheFlow – Sample 1

NicheFlow – Sample 2

RPCFlow – Sample 1

RPCFlow – Sample 2

Figure 10. Comparison between NicheFlow and RPCFlow in a fixed-source microenvironment setting. Each row shows input (source), ground truth target, model prediction, and KDE-estimated likelihood. While NicheFlow (top) produces well-localized samples consistent with the input, RPCFlow (bottom) generates less structured and spatially inaccurate predictions.

D.6. Description of the biological comparison with moscot

Here, we describe how we conduct the comparisons with moscot, as shown in Figure 3, Figure 12, and Figure 11. For both NicheFlow and moscot, we select a source microenvironment that we want to track over time. In the case of NicheFlow, this corresponds to an aggregate of point clouds. For moscot, it refers to a group of single cells spatially located within the region of interest. The same set of cells is used for both methods.

NicheFlow. To generate contour plots over the spatial slide, we push forward the selected region and assign the generated points to their nearest real neighbors based on spatial coordinates. We then compute a probability value for each real position by normalizing the number of assigned generated points. In other words, the more generated points are close to a given real point, the higher the probability assigned to that location in the contour plot. Cell type proportions are computed as the aggregated frequencies of the generated cell types across the slide. Each plot considers 10 independent generation runs from the same niche.

moscot. This baseline is not a generative model, but rather a standard discrete OT framework using a Fused Gromov-Wasserstein cost. As such, it does not generate new features or coordinates. Instead, it outputs a transition matrix that assigns matching probabilities between each source slide cell and each target slide cell. We use these transition probabilities to compute contour plots over the target slide and to aggregate cell type probabilities for the bar plots. For the latter, moscot provides a custom method called `cell_transition()`.

For the Appendix figures Figure 12, Figure 13, Figure 14, and Figure 15, we propagate the initial source region across multiple time steps. In NicheFlow, this is achieved by using the simulated point cloud from step t to predict the next state at $t + 1$, and then feeding this output as the input for the following trajectory step from $t + 1$ to $t + 2$, and so on. Ground truth points are not used as intermediate sources during this process. For moscot, pushing the source across time points can be automatically done by setting non-subsequent time points in the `cell_transition()` function.

D.6.1. ADDITIONAL COMPARISONS WITH MOSCOT ON EMBRYONIC DEVELOPMENT

We propose a similar analysis as presented in Section 5.2. In Figure 12, we compare the ability of NicheFlow to predict an entire spatial structure trajectory by pushing an initial source cloud through all the developmental stages. To this end, the trajectory of an initial point cloud is first pushed to the next time point, and the model's prediction is used as a source for predicting the subsequent time point. An accurate niche trajectory reconstruction signifies that our model can be used to sequentially predict microenvironment evolution by treating its intermediate predictions as inputs, corroborating their accurate reflection of real point clouds.

Specifically, in Figure 12 we show that pushing anterior neural crest cells twice from E9.5 to E11.5 through the flow generates realistic target point clouds with a cell composition reflecting the expected cranial structure mostly made of cavity cells, jaw and teeth (arising at E11.5 for the first time) and surface ectoderm. Doing the same with moscot oversamples regions outside of the cranial structure, thereby incorrectly mapping most of the neural crest density to brain cells.

In Figure 11, we also show that NicheFlow is more accurate than moscot at transporting mass from small organs like the liver across development. More specifically, while density leaks from the liver to the GI tract in the mapping produced by moscot, the prediction computed by NicheFlow more accurately retrieves the liver structure at the later time point. Together with previous evidence, our results underscore the importance of accounting for spatial correlations between cells during OT-based trajectory inference to buffer out the noise resulting from single-cell-based predictions.

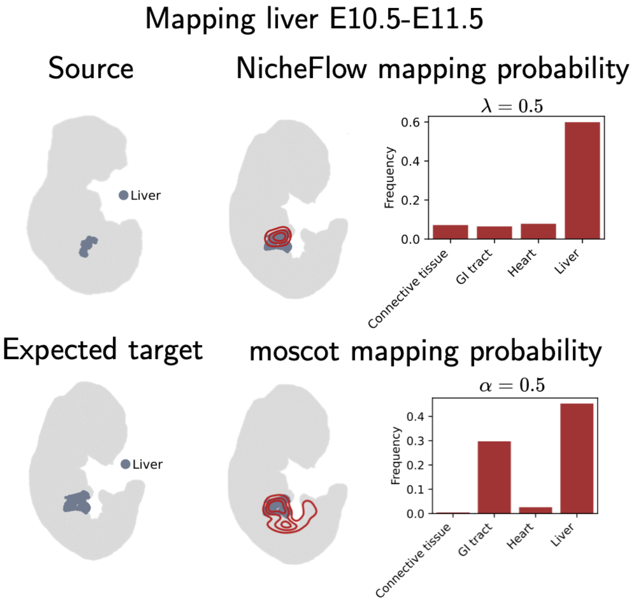


Figure 11. Comparison between moscot and NicheFlow on mapping the liver structure from E10.5 to E11.5. The liver at time E10.5 is used as a source for trajectory prediction using the different models. The left column shows the source and expected target regions highlighted on the respective E10.5 and E11.5 embryos. The middle column displays the density of the prediction obtained by transporting niches from the source to the target slide. On the right, the aggregated cell type proportions according to the density in the middle column (see Appendix D.6).

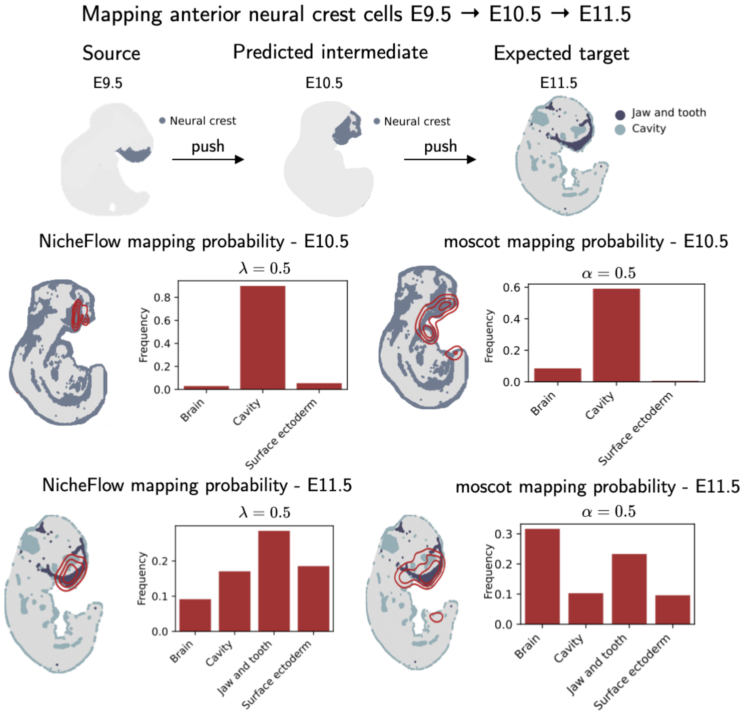


Figure 12. Comparison of NicheFlow and moscot on the prediction of the anterior neural crest cells' fate. For both models, we take source facial neural crest cells at E9.5, push them to time point E10.5, and show the compositional and density predictions in the middle panel. Then, the predictions at 10.5 are used as a source for a second trajectory prediction operation from 10.5 to 11.5, for which we inspect again the cell density over the target slide and the cell type probabilities.

D.6.2. Experiments on the axolotl brain development

To further support our result, we propose a more in-depth qualitative analysis of the NicheFlow application on the axolotl brain development dataset. More specifically, in Figure 13 we demonstrate that our model predicts the formation of crucial anatomical structures like the left and right lobes both spatially and compositionally. This vouches for flexibility in NicheFlow's performance, which extends to non-trivial topology changes and simultaneously accounts for accurate cell state and coordinate generation in time. Similar results can be observed in Figure 14, where we showcase the correct prediction of the formation of a left lobe cavity, predicting trajectories from an immature brain region.

Furthermore, in Figure 15 we predict the compositional and structural time evolution of the left dorsal pallium in the axolotl brain development. Following Wei et al. (Wei et al., 2022), we know that early time points populate the dorsal pallium of immature cell types like ependymoglial cells (EGC), neuroblasts (NBL), and immature neurons. In the left dorsal pallium, these disappear at the juvenile stages (the 3rd time point) and lead to differentiation into mature neurons (nptxEX) and later EGC (WntEGC, sfrpEGC). This fixed structural development is accurately predicted by NicheFlow when pushing left dorsal pallium cells forward across the trajectory.

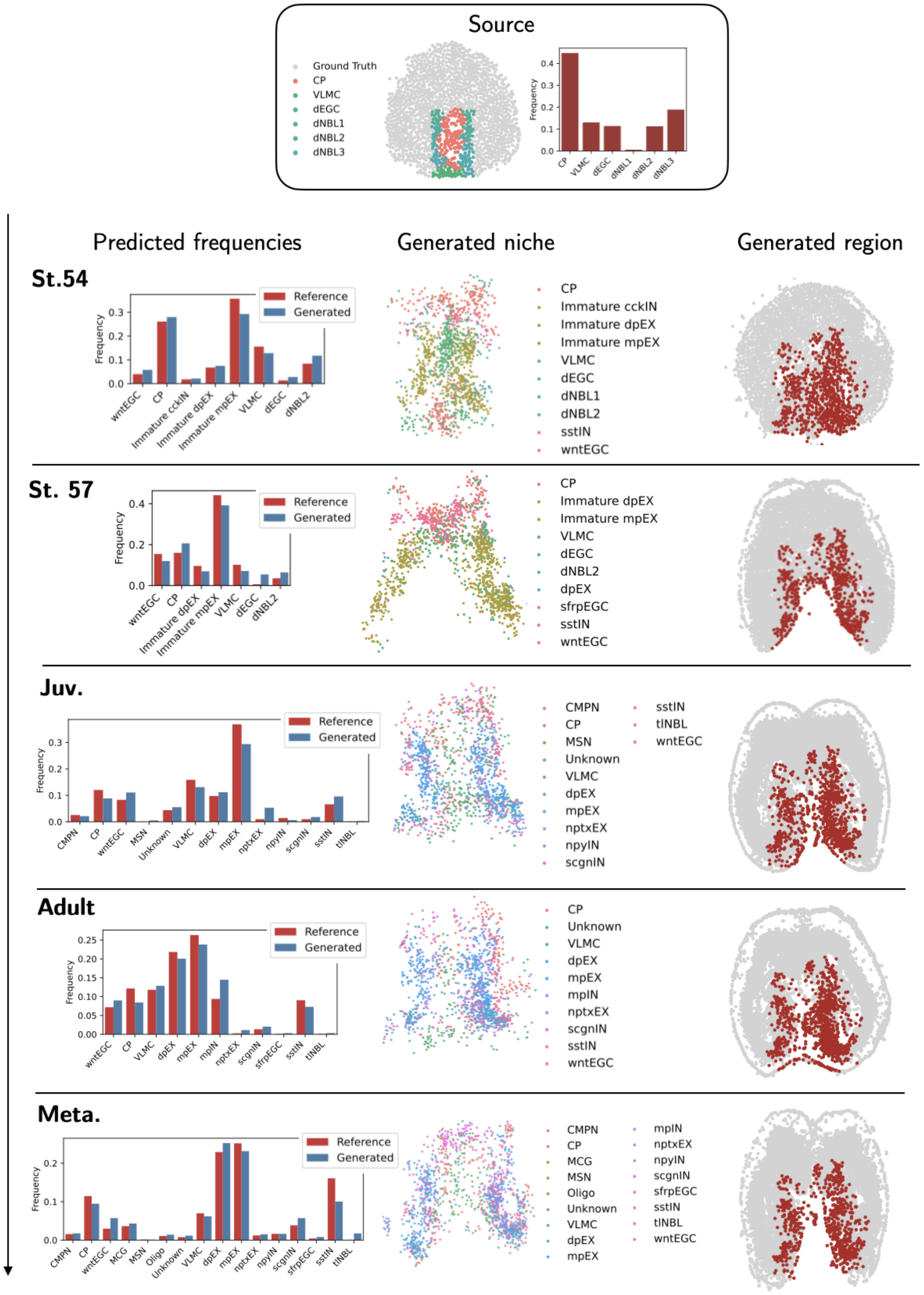


Figure 13. Prediction of hemisphere formation on the axolotl brain development. We evaluate the ability of NicheFlow to generate the anatomical splitting of central brain structure upon the formation of the right and left brain regions. The top panel shows the reference source region and cell type composition. For each stage (St.54, St.57, Juvenile, Adult, and Meta), we show: (1) predicted vs. reference cell type frequencies, (2) the generated niche visualized via 2D embedding colored by cell type, and (3) spatial projection of the generated region onto the anatomical reference (right). We set $\lambda = 0.5$.

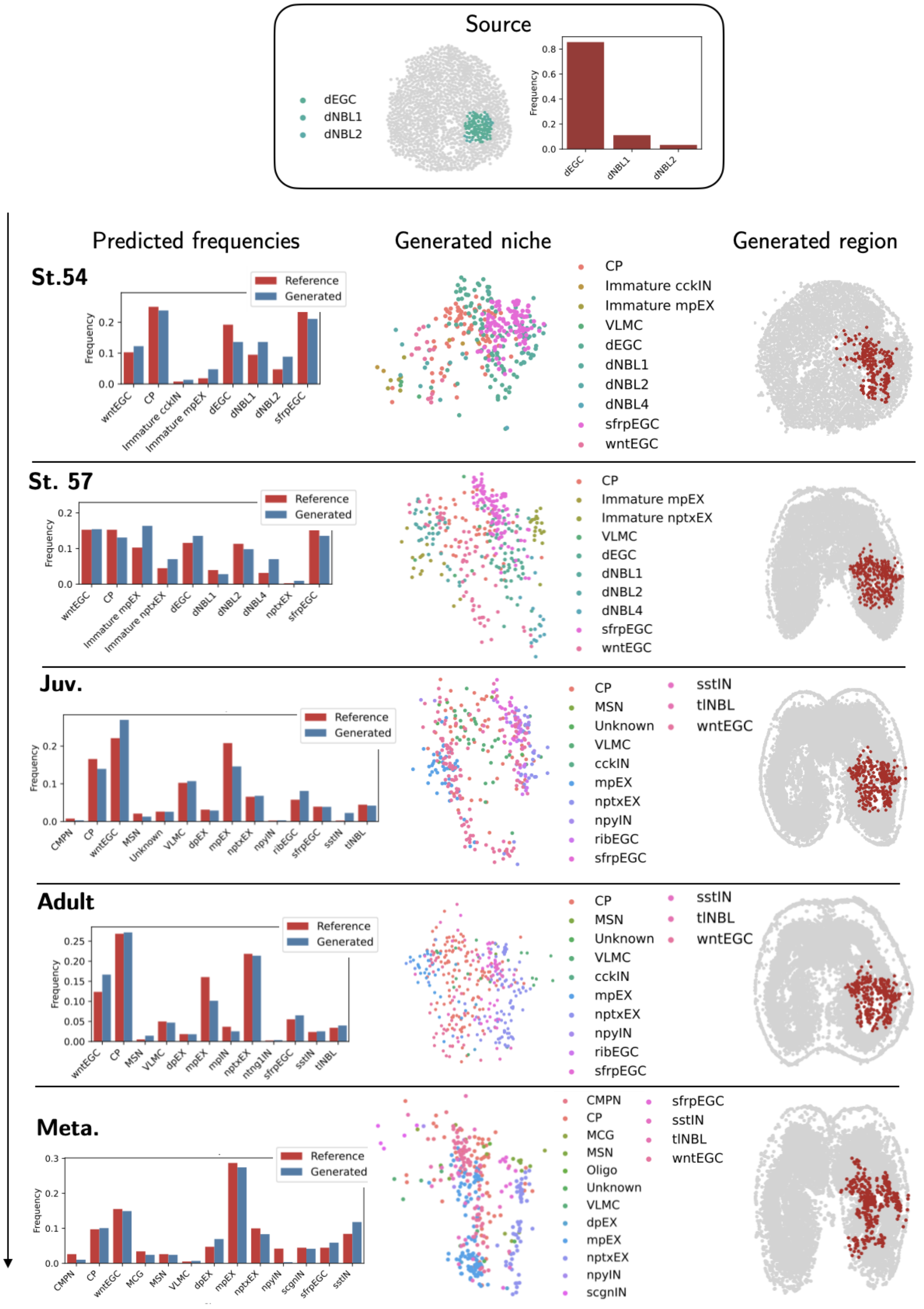


Figure 14. Prediction of cavity formation in the left brain lobe during axolotl development. We assess the ability of NicheFlow to model the emergence of a cavity structure within the left lobe of the axolotl brain. The top panel shows the reference source region and its cell type composition. For each developmental stage (St.54, St.57, Juvenile, Adult, and Meta), we display: (1) predicted versus reference cell type frequencies, (2) the generated niche visualized in 2D embedding space, colored by cell type, and (3) spatial projection of the generated region onto the anatomical brain reference (right). The progression illustrates the model's ability to recapitulate the asymmetric cavity formation localized to the left lobe. We set $\lambda = 0.5$.

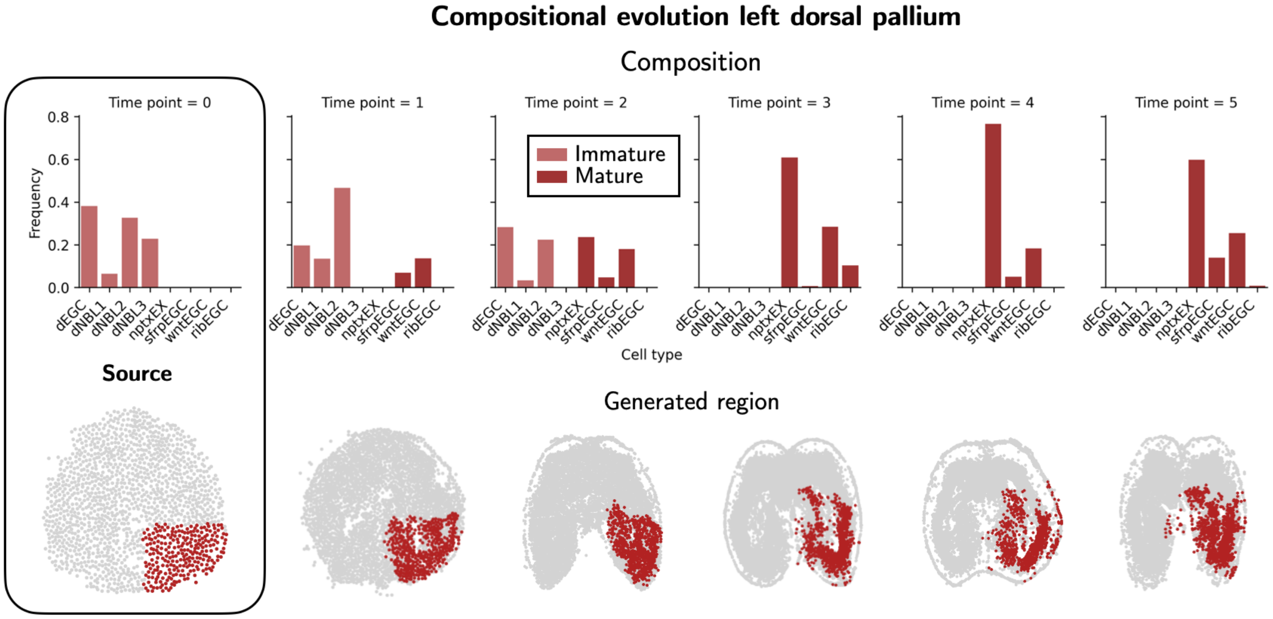


Figure 15. Structural and compositional prediction of the left dorsal pallium during the axolotl brain development. We use the left dorsal pallium region at the St.44 developmental stadium (highlighted on the left) and predict its trajectory over time. In the top row, we show the proportion of different cell types in the predicted region, colored as immature (light red) and mature (dark red). At the bottom, we show the structural prediction for the left dorsal pallium overlaid on the true slide. We set $\lambda = 0.5$.

E. Algorithms

Here, we present the algorithmic procedures underlying NicheFlow. To streamline the exposition, we first define compact notation for representing noisy microenvironments and their interpolations.

We denote a single noisy microenvironment (simplifying Equation (11)) as:

$$\mathcal{M}^z \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{D+2})^{1\times k} := \{(\boldsymbol{c}_i^z, \boldsymbol{z}_i) \mid \boldsymbol{c}_i^z \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_2),\ \boldsymbol{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D),\ \forall i = 1, \dots, k\}, \tag{33}$$

where k denotes the number of spatial points per microenvironment, $\boldsymbol{c}_i^z$ are 2D cell coordinates, and $\boldsymbol{z}_i$ are associated feature vectors in $\mathbb{R}^D$. For a collection of N microenvironments, we define the corresponding set of noisy microenvironments

$$\boldsymbol{\mathcal{M}}^{\boldsymbol{z}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{D+2})^{N\times k} := \left\{\mathcal{M}_i^z \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{D+2})^{1\times k} \mid \forall i = 1, \dots, N\right\}. \tag{34}$$

Given a noisy sample $\mathcal{M}^z$ and a corresponding ground-truth target microenvironment $\mathcal{M}^1$, we define their linear interpolation at time $t \in [0,1]$ as: $\mathcal{M}^t = (1-t)\mathcal{M}^z + t\mathcal{M}^1$, where the interpolation is performed element-wise across the matrix rows. For batched data, this generalizes to:

$$\boldsymbol{\mathcal{M}}^t = (1-t)\boldsymbol{\mathcal{M}}^z + t\boldsymbol{\mathcal{M}}^1. \tag{35}$$

Algorithm 1 SAMPLEANDINTERPOLATE

```
Require: Number of samples $N$, feature dimension $D$, microenvironment size $k$, OT plan $\pi^\star_{\epsilon,\lambda}$
Ensure: Source ($\boldsymbol{\mathcal{M}}^0$), target ($\boldsymbol{\mathcal{M}}^1$), interpolated ($\boldsymbol{\mathcal{M}}^t$), and noisy ($\boldsymbol{\mathcal{M}}^z$) microenvironments
$(\boldsymbol{\mathcal{M}}^0, \boldsymbol{\mathcal{M}}^1) \leftarrow$ Sample from KMeans regions          ▷ Initial microenvironment pairs
$(\boldsymbol{\mathcal{M}}^0, \boldsymbol{\mathcal{M}}^1) \leftarrow \pi^\star_{\epsilon,\lambda}(\boldsymbol{\mathcal{M}}^0, \boldsymbol{\mathcal{M}}^1)$          ▷ Resample with OT plan
$\boldsymbol{\mathcal{M}}^z \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{D+2})^{N\times k}$          ▷ Noisy initial states
$t \sim \mathcal{U}(0,1)$          ▷ Random interpolation time
$\boldsymbol{\mathcal{M}}^t \leftarrow (1-t)\boldsymbol{\mathcal{M}}^z + t\boldsymbol{\mathcal{M}}^1$          ▷ Linearly interpolated states
```

E.1. OT Conditional Flow Matching

The OT Conditional Flow Matching (OT-CFM) algorithm consists of a training and an inference phase. During training, the model learns a time-dependent vector field u_t^θ conditioned on a source microenvironment $\mathcal{M}^0$, which maps a noisy microenvironment $\mathcal{M}^z$, sampled from Gaussian noise, to its corresponding target microenvironment $\mathcal{M}^1$. The supervision is provided via source-target pairs $(\mathcal{M}^0, \mathcal{M}^1)$ obtained through entropic OT over pooled microenvironment representations. At inference time, the learned vector field is integrated starting from $\mathcal{M}^z$, conditioned on a given source $\mathcal{M}^0$, to generate the predicted target microenvironment $\mathcal{M}^1$.

We optimize the following loss:

$$\mathcal{L}_{\mathrm{OT-CFM}}(\boldsymbol{\mathcal{M}}^0, \boldsymbol{\mathcal{M}}^1, \boldsymbol{\mathcal{M}}^t, \boldsymbol{\mathcal{M}}^z; \theta) = \frac{1}{2} \sum_{\substack{\mathcal{M}^0 \in \boldsymbol{\mathcal{M}}^0 \\ \mathcal{M}^z \in \boldsymbol{\mathcal{M}}^z \\ \mathcal{M}^t \in \boldsymbol{\mathcal{M}}^t \\ \mathcal{M}^1 \in \boldsymbol{\mathcal{M}}^1}} \left\| u_t^\theta(\mathcal{M}^t, \mathcal{M}^0) - (\mathcal{M}^1 - \mathcal{M}^z) \right\|_2^2 \tag{36}$$

The full pseudocode for both phases is provided in Algorithms 2 and 3.

Algorithm 2 OT CFM — Training

```
Require: Number of samples $N$, feature dimension $D$, microenvironment size $k$, OT plan $\pi^\star_{\epsilon,\lambda}$, conditional velocity field $u_t^\theta$
Ensure: Trained parameters $\theta$ of $u_t^\theta$
$(\boldsymbol{\mathcal{M}}^0, \boldsymbol{\mathcal{M}}^1, \boldsymbol{\mathcal{M}}^t, \boldsymbol{\mathcal{M}}^z) \leftarrow$ SAMPLEANDINTERPOLATE$(N, D, k, \pi^\star_{\epsilon,\lambda})$          ▷ Microenvironments (Algorithm 1)
$\theta \leftarrow \nabla_\theta \mathcal{L}_{\mathrm{OT-CFM}}(\boldsymbol{\mathcal{M}}^0, \boldsymbol{\mathcal{M}}^1, \boldsymbol{\mathcal{M}}^t, \boldsymbol{\mathcal{M}}^z; \theta)$          ▷ Compute loss (Equation (36)) & update parameters $\theta$
```

Algorithm 3 OT CFM — Inference

```
Require: Source microenvironment $\mathcal{M}^0$, learned conditional velocity field $u_t^\theta$
Ensure: Generated microenvironment $\mathcal{M}^1$
$\mathcal{M}^z \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{D+2})^{1\times k}$          ▷ Sample noisy sample
$\mathcal{M}^1 \leftarrow \mathcal{M}^z + \int_0^1 u_t^\theta(\mathcal{M}^t, \mathcal{M}^0)\, dt$          ▷ Solve ODE
```

E.2. OT Gaussian Variational Flow Matching

The OT Gaussian Variational Flow Matching (OT-GVFM) algorithm adopts a variational perspective on Flow Matching. Instead of directly learning a time-dependent conditional velocity field, the model learns a factorized variational posterior $q_t^\theta(\mathcal{M}^1 \mid \mathcal{M}^t, \mathcal{M}^0)$ over target microenvironments $\mathcal{M}^1$, conditioned on an interpolated microenvironment $\mathcal{M}^t$ and a source microenvironment $\mathcal{M}^0$. The predicted velocity field is then computed as the difference between the posterior mean $\mu_t^\theta(\mathcal{M}^t, \mathcal{M}^0)$ and the current state $\mathcal{M}^t$.

The training objective minimizes the discrepancy between ground-truth targets and the predicted posterior means $(\bar{\boldsymbol{f}}_t^\theta, \bar{\boldsymbol{r}}_t^\theta)$:

$$\mathcal{L}_{\mathrm{OT-GVFM}}(\boldsymbol{\mathcal{M}}^0, \boldsymbol{\mathcal{M}}^1, \boldsymbol{\mathcal{M}}^t; \theta) = \frac{1}{2} \sum_{\substack{\mathcal{M}^0 \in \boldsymbol{\mathcal{M}}^0 \\ \mathcal{M}^t \in \boldsymbol{\mathcal{M}}^t \\ \mathcal{M}^1 \in \boldsymbol{\mathcal{M}}^1}} \sum_{\substack{(\boldsymbol{c}_1, \boldsymbol{x}_1) \in \mathcal{M}^1 \\ (\bar{\boldsymbol{f}}_t^\theta, \bar{\boldsymbol{r}}_t^\theta) \in \mu_t^\theta(\mathcal{M}^t, \mathcal{M}^0)}} \left(\|\boldsymbol{c}_1 - \bar{\boldsymbol{f}}_t^\theta\|_2^2 + \|\boldsymbol{x}_1 - \bar{\boldsymbol{r}}_t^\theta\|_2^2 \right). \tag{37}$$

At inference time, trajectories are generated by integrating the vector field $\mu_t^\theta(\mathcal{M}^t, \mathcal{M}^0) - \mathcal{M}^t$, starting from a noise-sampled microenvironment $\mathcal{M}^z$ and conditioned on a given source $\mathcal{M}^0$. To ensure stability near $t = 1$, the vector field is scaled by a time-dependent denominator, as in prior VFM formulations.

The pseudocode for both phases is provided in Algorithms 4 and 5.

```
Algorithm 4 OT-GVFM — Training
Require: Number of samples $N$, feature dimension $D$, microenvironment size $k$, OT plan $\pi^\star_{\epsilon,\lambda}$, source-conditioned posterior mean
    predictor $\mu_t^\theta$
Ensure: Trained parameters $\theta$ of $\mu_t^\theta$
  $(\boldsymbol{\mathcal{M}}^0, \boldsymbol{\mathcal{M}}^1, \boldsymbol{\mathcal{M}}^t, \boldsymbol{\mathcal{M}}^z) \leftarrow$ SampleAndInterpolate$(N, D, k, \pi^\star_{\epsilon,\lambda})$    ▷ Microenvironments (Algorithm 1)
  $\theta \leftarrow \nabla_\theta \mathcal{L}_{\mathrm{OT-GVFM}}(\boldsymbol{\mathcal{M}}^0, \boldsymbol{\mathcal{M}}^1, \boldsymbol{\mathcal{M}}^t; \theta)$    ▷ Compute loss (Equation (37)) & update parameters $\theta$
```

```
Algorithm 5 OT-GVFM — Inference
Require: Source microenvironment $\mathcal{M}^0$, learned source-conditioned posterior mean predictor $\mu_t^\theta$
Ensure: Generated microenvironment $\mathcal{M}^1$
  $\mathcal{M}^z \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{D+2})^{1 \times k}$    ▷ Sample noisy sample
  $\mathcal{M}^1 \leftarrow \mathcal{M}^z + \int_0^1 \frac{\mu_t^\theta(\mathcal{M}^t, \mathcal{M}^0) - \mathcal{M}^t}{1 - t + \epsilon} dt$    ▷ Solve ODE
```

E.3. NicheFlow: OT Gaussian-Laplace Variational Flow Matching

NicheFlow extends OT-GVFM by modifying the variational posterior: It assumes a Gaussian distribution for gene expression features and a Laplace distribution for spatial coordinates. This change leads to a hybrid loss that combines an L^2 loss on gene expression and an L^1 loss on spatial locations:

$$\mathcal{L}_{\mathrm{NicheFlow}}(\boldsymbol{\mathcal{M}}^0, \boldsymbol{\mathcal{M}}^1, \boldsymbol{\mathcal{M}}^t; \theta) = \sum_{\substack{\mathcal{M}^0 \in \boldsymbol{\mathcal{M}}^0 \\ \mathcal{M}^t \in \boldsymbol{\mathcal{M}}^t \\ \mathcal{M}^1 \in \boldsymbol{\mathcal{M}}^1}} \sum_{\substack{(\boldsymbol{c}_1, \boldsymbol{x}_1) \in \mathcal{M}^1 \\ (\bar{\boldsymbol{f}}_t^\theta, \bar{\boldsymbol{r}}_t^\theta) \in \mu_t^\theta(\mathcal{M}^t, \mathcal{M}^0)}} \left(\|\boldsymbol{c}_1 - \bar{\boldsymbol{f}}_t^\theta\|_1 + \frac{1}{2} \|\boldsymbol{x}_1 - \bar{\boldsymbol{r}}_t^\theta\|_2^2 \right) \tag{38}$$

At inference time, the model integrates the velocity field defined by the difference between the predicted mean $\mu_t^\theta(\mathcal{M}^t, \mathcal{M}^0)$ and the current state $\mathcal{M}^t$, starting from noise and conditioned on the source $\mathcal{M}^0$, identical to the OT-GVFM procedure.

The pseudocode for both phases is provided in Algorithms 6 and 7.

```
Algorithm 6 NicheFlow — Training
Require: Number of samples $N$, feature dimension $D$, microenvironment size $k$, OT plan $\pi^\star_{\epsilon,\lambda}$, source-conditioned posterior mean
    predictor $\mu_t^\theta$
Ensure: Trained parameters $\theta$ of $\mu_t^\theta$
  $(\boldsymbol{\mathcal{M}}^0, \boldsymbol{\mathcal{M}}^1, \boldsymbol{\mathcal{M}}^t, \boldsymbol{\mathcal{M}}^z) \leftarrow$ SampleAndInterpolate$(N, D, k, \pi^\star_{\epsilon,\lambda})$    ▷ Microenvironments (Algorithm 1)
  $\theta \leftarrow \nabla_\theta \mathcal{L}_{\mathrm{NicheFlow}}(\boldsymbol{\mathcal{M}}^0, \boldsymbol{\mathcal{M}}^1, \boldsymbol{\mathcal{M}}^t; \theta)$    ▷ Compute loss (Equation (38)) & update parameters $\theta$
```

Algorithm 7 NicheFlow — Inference

```
Require: Source microenvironment $\mathcal{M}^0$, learned source-conditioned posterior mean predictor $\mu_t^\theta$
Ensure: Generated microenvironment $\mathcal{M}^1$
$\mathcal{M}^z \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{D+2})^{1\times k}$    ▷ Sample noisy microenvironment
$\mathcal{M}^1 \leftarrow \mathcal{M}^z + \int_0^1 \frac{\mu_t^\theta(\mathcal{M}^t, \mathcal{M}^0) - \mathcal{M}^t}{1-t+\epsilon} dt$    ▷ Solve ODE
```

F. Experimental setup

F.1. Dataset description

We evaluate our model on three publicly available, time-resolved spatial transcriptomics datasets spanning development and aging processes. Each dataset provides single-cell resolution profiles with matched spatial coordinates and curated cell type annotations. A detailed summary of the organism, tissue, number of time points, cell types, total number of cells, and acquisition technology for each dataset is provided in Table 4.

Table 4. Overview of the time-resolved spatial transcriptomics datasets used in our experiments. MED: Mouse Embryonic Development, ABD: Axolotl Brain Development, MBA: Mouse Brain Aging. Each dataset varies in organism, tissue type, number of timepoints, cell types, total number of cells, and spatial transcriptomics technology.

Dataset	Organism	Tissue	Timepoints	Cell Types	Cells	Technology
MED	Mouse	Whole embryo	3	24	54k	Stereo-seq
ABD	Axolotl	Brain	6	33	36k	Stereo-seq
MBA	Mouse	Brain	20	18	1.5M	MERFISH

F.2. Dataset and microenvironments preprocessing

Dataset preprocessing. All datasets used in our study underwent a preprocessing procedure appropriate for spatial transcriptomic analysis, involving total count normalization, logarithmic transformation, and principal component analysis (PCA). For the mouse embryonic development (Klein et al., 2025) and axolotl brain development (Wei et al., 2022) datasets, total count normalization, and logarithmic transformation had already been applied; we, therefore, performed PCA ourselves on the transformed data. For the mouse brain aging dataset (Sun et al., 2025), we applied all three steps: We first normalized raw count matrices so that each cell had the same total expression, followed by a natural logarithm transformation of the form $\log(x + 1)$ to stabilize variance and mitigate the influence of large values. We then computed PCA on the log-transformed data. To reduce computational overhead due to high cell counts in the aging dataset, we subsampled the data by a factor of 0.2.

Finally, we standardized PCA components across all cells to ensure consistent scaling across time points. Spatial coordinates were standardized independently for each time point by subtracting the per-time-point mean and dividing by the standard deviation. This standardization preserves the relative spatial configuration while accounting for scale and position differences over developmental time.

Microenvironments preprocessing. To facilitate efficient training and enable consistent microenvironment construction, we precompute all fixed-radius neighborhoods for each dataset using a radius r chosen based on spatial resolution. To reduce variability in the number of neighbors and improve batching efficiency, we fix the number of nodes per microenvironment to the most frequent neighbor count observed within each slide. This standardization ensures structural comparability across microenvironments while significantly reducing computational overhead during training, as costly radius or nearest-neighbor queries are avoided at runtime.

F.3. PSD and SPD metrics

We assess spatial fidelity using two complementary asymmetric distance measures. The *point-to-shape distance* (PSD) captures how much predicted cell positions diverge from the actual tissue layout, computed as the average squared distance from each simulated point to its nearest neighbor in the ground truth. Conversely, the *shape-to-point distance* (SPD) quantifies how comprehensively the predicted distribution spans the target structure by averaging the squared distance from each ground truth point to its closest generated counterpart.

Let $\mathcal{G}^t$ denote the set of generated coordinates and $\mathcal{R}^t$ the set of ground truth coordinates at time t. Define $\mathrm{NN}^t_{\text{ref}}(\boldsymbol{c}_i)$ as the nearest neighbor of $\boldsymbol{c}_i \in \mathcal{G}^t$ in $\mathcal{R}^t$, and $\mathrm{NN}^t_{\text{gen}}(\boldsymbol{c}_i)$ as the nearest neighbor of $\boldsymbol{c}_i \in \mathcal{R}^t$ in $\mathcal{G}^t$. Then, the two metrics are given by:

$$\mathrm{PSD} = \frac{1}{|\mathcal{G}|} \sum_{\mathcal{G}^t \in \mathcal{G}} \sum_{\boldsymbol{c}_i \in \mathcal{G}^t} \|\boldsymbol{c}_i - \mathrm{NN}^t_{\mathrm{ref}}(\boldsymbol{c}_i)\|_2^2, \tag{39}$$

$$\mathrm{SPD} = \frac{1}{|\mathcal{R}|} \sum_{\mathcal{R}^t \in \mathcal{R}} \sum_{\boldsymbol{c}_i \in \mathcal{R}^t} \|\boldsymbol{c}_i - \mathrm{NN}^t_{\mathrm{gen}}(\boldsymbol{c}_i)\|_2^2. \tag{40}$$

where $\mathcal{G} := \cup_{t\in\mathcal{T}}\mathcal{G}^t$ and $\mathcal{R} := \cup_{t\in\mathcal{T}}\mathcal{R}^t$.

F.4. Cell-type classification for evaluation

To evaluate cell-type fidelity of generated microenvironments, we train a supervised classifier to assign cell type labels based on the gene expression profiles. We apply the same preprocessing steps used for training our generative models: Total count normalization, log-transformation, and PCA reduction (see Appendix F.2). The resulting low-dimensional embeddings are used as input features for a simple multilayer perceptron (MLP), trained to predict discrete cell type labels.

The classifier consists of a two-layer feedforward network with ReLU activations and a final linear projection to the number of cell types. It is trained using cross-entropy loss and optimized with the AdamW optimizer. We report performance using the weighted F1-score.

We obtain strong classification results across all datasets. On the mouse embryonic development dataset, the classifier achieves a weighted F1-score of 0.85; on axolotl brain development, 0.80; and on the aging dataset, 0.97. These results correlate with the number of input genes and the variance retained in the PCA-reduced space. The aging dataset contains only 300 genes, and 50 principal components explain sufficient variance to accurately distinguish most cell types. In contrast, the embryonic development dataset contains approximately 2,000 genes, and the axolotl brain development dataset includes over 12,700 genes, making classification more challenging due to higher gene expression variability.

F.5. Discretized microenvironments

To ensure consistent and reproducible evaluation across methods and datasets, we construct a fixed set of evaluation microenvironments by discretizing the spatial domain of each tissue section. For each time point, we define a regular 2D grid over the tissue and select the closest cell to each grid point as the centroid of a microenvironment. Each centroid is then used to construct a fixed-radius neighborhood, following the microenvironment definition in Section 4.1. This procedure ensures full spatial coverage by verifying that every cell belongs to at least one microenvironment.

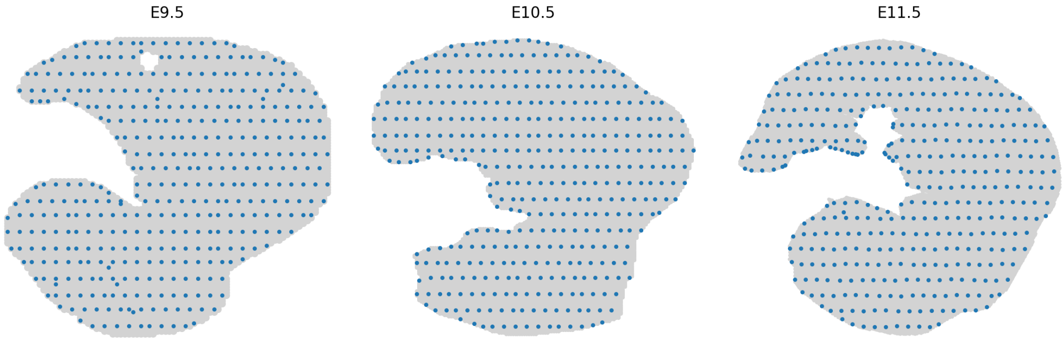


Figure 16. Discretized grid of microenvironments for the mouse embryonic development dataset. Each blue point denotes a centroid around which a microenvironment is constructed. To ensure consistent coverage across tissue sections, additional centroids are randomly sampled such that each slide contains the same number of microenvironments.

Figure 16 illustrates this discretization for the mouse embryonic development dataset. Each blue dot corresponds to a selected centroid. In cases where the number of grid-based centroids falls below a target threshold, additional centroids are randomly sampled to match a fixed total count per slide. This augmentation ensures that all slides contribute equally to the evaluation and prevents bias from sparse regions.

We apply the same discretization procedure to all three datasets used in our experiments: Mouse embryogenesis, axolotl brain development, and mouse brain aging. By standardizing the evaluation regions spatially and deterministically, we eliminate the need for stochastic region sampling during evaluation, which would otherwise lead to nondeterministic and irreproducible results.

F.6. Microenvironment Transformer

To model the spatiotemporal evolution of local cellular neighborhoods, we design a permutation invariant transformer-based architecture tailored to structured point cloud data. Our Microenvironment Transformer processes local microenvironments, sets of cells with gene expression features and spatial coordinates, and predicts time-dependent outputs such as velocity fields or future states.

The model operates on a source $\mathcal{M}^0$ and noisy $\mathcal{M}$ microenvironments with per-cell features $\boldsymbol{x}_i \in \mathbb{R}^D$ and 2D coordinates $\boldsymbol{c}_i \in \mathbb{R}^2$. The architecture consists of the following components:

1. **Input Embeddings:**
   (a) **Feature Embedding:** A linear transformation is applied to the input features $\boldsymbol{x}_i$ of each cell.
   (b) **Coordinate Embedding:** Spatial coordinates $\boldsymbol{c}_i$ are linearly projected and concatenated to the feature embedding.
   (c) **Time Embedding:** For the noisy microenvironment only, time $t \in [0, 1]$ is encoded using sinusoidal functions $\cos(\omega t)$ and $\sin(\omega t)$, followed by a linear projection and concatenation with the input embedding.
2. **Transformer Encoder (Source Microenvironment):**
   (a) **Self-Attention:** A stack of transformer encoder blocks with multi-head self-attention processes the embedded source microenvironment.
   (b) **No Time Embedding:** Time information is *not* provided to the encoder, as it encodes the source $\mathcal{M}^0$.
   (c) **Residual Feedforward:** Each block contains a feedforward subnetwork with LeakyReLU activation and residual connection.
   (d) **Layer Normalization:** Applied after both the attention and feedforward layers.
   (e) **Masking:** Binary masks are used to ignore padding in variable-length microenvironments.
3. **Transformer Decoder (Noisy Microenvironment):**
   (a) **Time Embedding:** Temporal context is injected into the decoder by embedding the time t and concatenating it to the noisy point embedding.
   (b) **Cross-Attention:** Decoder layers apply cross-attention between the noisy microenvironment and the encoded source microenvironment.
   (c) **Self-Attention and Feedforward:** Each decoder block includes standard self-attention and residual feedforward layers.
   (d) **Layer Normalization and Masking:** As with the encoder, normalization, and masking are applied throughout.
4. **Final Output Projection:**
   (a) **Prediction Head:** A linear layer maps the decoder outputs to the desired dimensionality.

This architecture allows for flexible and expressive modeling of temporal dynamics in cellular point clouds. By encoding only the source and decoding the temporally conditioned target, the model supports variational and flow-based training objectives with explicit temporal conditioning.

F.7. Hyperparameters and Computational Costs

Model hyperparameters. For all experiments, we use the same configuration for the Microenvironment Transformer architecture. The full set of hyperparameters is as follows:

- **Input feature dimension:** 50 PCA features concatenated with a one-hot encoding of the time-point, resulting in a total dimensionality of $50 + |\mathcal{S}|$, where $|\mathcal{S}|$ is the number of slides (timepoints) in the dataset.

- **Input coordinate dimension:** 2
- **Embedding dimension:** 128
- **MLP hidden dimension:** 256
- **Number of attention heads:** 4
- **Number of encoder layers:** 2
- **Number of decoder layers:** 2
- **Dropout rate:** 0.1
- **Output dimension:** 52 (gene expressions PC features + coordinates)

OT and mini-batching. To ensure spatial diversity and computational tractability, we uniformly sample 256 source–target microenvironment pairs from the K spatial clusters obtained via KMeans. We then compute the entropic OT plan between these sampled pairs and resample 64 source-target pairs from this plan to define a single training instance. During training, we process 16 such instances per batch.

Optimization. All models are trained using the AdamW optimizer with a learning rate of $2 \cdot 10^{-4}$ and a weight decay of $1 \cdot 10^{-5}$. We train each model until convergence.

Computational cost. All models were trained on a single NVIDIA GeForce GTX 1080 Ti GPU with 11GB of memory. Depending on the dataset and training objective (e.g., CFM or VFM), training takes approximately 12–16 hours per model.