BRANCHED SCHRÖDINGER BRIDGE MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting the intermediate trajectories between an initial and target distribution is a central problem in generative modeling. Existing approaches, such as flow matching and Schrödinger Bridge Matching, effectively learn mappings between two distributions by modeling a single stochastic path. However, these methods are inherently limited to unimodal transitions and cannot capture *branched* or *divergent* evolution from a common origin to multiple distinct outcomes. To address this, we introduce **Branched Schrödinger Bridge Matching (BranchSBM)**, a novel framework that learns branched Schrödinger bridges. BranchSBM parameterizes multiple time-dependent velocity fields and growth processes, enabling the representation of population-level divergence into multiple terminal distributions. We show that BranchSBM is not only more expressive but also essential for tasks involving multi-path surface navigation, modeling cell fate bifurcations from homogeneous progenitor states, and simulating diverging cellular responses to perturbations.

1 INTRODUCTION

Tasks like crowd navigation and modeling cell-state transitions under perturbation involve learning a transport map between two empirically observed endpoint distributions, rather than a noisy prior required for denoising diffusion (Austin et al., 2021) and flow matching (Lipman et al., 2023). The Schrödinger Bridge (SB) (Schrödinger, 1931) problem seeks to identify an optimal stochastic map between a pair of endpoint distributions that minimizes the Kullback-Leibler (KL) divergence to an underlying reference process.

Schrödinger Bridge Matching (SBM) solves the SB problem by parameterizing a drift field that matches a mixture of conditional stochastic bridges between endpoint pairs that each minimize the KL divergence from a known reference process. Typically, SBM assumes conservation of mass from the initial to the target distribution, which fails to capture dynamical population behaviors such as growth and destruction of mass, commonly seen in single-cell population data. Furthermore, prior works focus on transporting samples from a pair of unimodal initial and target distributions via a single, continuous trajectory, without accounting for *branching* dynamics (Liu et al., 2023a; Tong et al., 2024b; Theodoropoulos et al., 2024; Liu et al., 2022; De Bortoli et al., 2021b), where a uniform population follows a branched trajectory that diverges toward multiple distinct target distributions.

The notion of branching is central to many real-world systems. For example, when a homogeneous cell population undergoes a perturbation such as gene knockouts or drug treatments, it frequently induces fate bifurcation as the cell population splits into multiple phenotypically distinct outcomes or commits to divergent cell fates (Shalem et al., 2014; Zhang et al., 2025a). These trajectories are observable in single-cell RNA sequencing (scRNA-seq) data, where each subpopulation independently evolves and undergoes growth or contraction along its trajectory toward a distinct terminal state (Dixit et al., 2016).

In this work, we introduce **Branched Schrödinger Bridge Matching (BranchSBM)**, a novel framework for learning stochastic transport maps from an unimodal initial distribution to multiple target distributions via branched trajectories.

Our main contributions can be summarized as follows:

1. We define the Branched Generalized Schrödinger Bridge problem and introduce **BranchSBM**, a novel matching framework that learns optimal branched trajectories from an initial distribution π_0 to multiple target distributions $\{\pi_{t,k}\}$.

2. We derive the Branched Conditional Stochastic Optimal Control (CondSOC) problem as the sum of Unbalanced CondSOC objectives and leverage a multi-stage training algorithm to learn the optimal branching drift and growth fields that transport mass along a branched trajectory.
3. We demonstrate the unique capability of BranchSBM to model dynamic branching trajectories while matching multiple target distributions across various problems, including 3D navigation over LiDAR manifolds (Section 5.1), modeling differentiating single-cell population dynamics (Section 5.2), and predicting heterogeneous cell states after perturbation (Section 5.3).

2 PRELIMINARIES

Schrödinger Bridge Given a reference probability path measure $\mathbb{Q}$, the Schrödinger Bridge (SB) problem aims to find an optimal path measure $\mathbb{P}^{\text{SB}}$ that minimizes the Kullback-Leibler (KL) divergence with $\mathbb{Q}$ while satisfying the boundary distributions $\mathbb{P}_0 = \pi_0$ and $\mathbb{P}_1 = \pi_1$.

$$\mathbb{P}^{\text{SB}} = \min_{\mathbb{P}} \{\text{KL}(\mathbb{P} \| \mathbb{Q}) : \mathbb{P}_0 = \pi_0, \mathbb{P}_1 = \pi_1\} \tag{1}$$

where $\mathbb{Q}$ is commonly defined as standard Brownian motion. For an extended background and formal definition of Schrödinger Bridges, refer to Definition 3 and Appendix A.1.

Generalized Schrödinger Bridge Problem The solution to the standard SB problem minimizes the kinetic energy of the conditional drift term $u_t(X_t)$ that preserves the endpoints drawn from the coupling $(\mathbf{x}_0, \mathbf{x}_1) \sim \pi_{0,1}$ defined as

$$\min_{u_t} \int_0^1 \mathbb{E}_{p_t} \|u_t(X_t)\|^2 dt \quad \text{s.t.} \quad \begin{cases} dX_t = u_t(X_t)dt + \sigma dB_t \\ X_0 \sim \pi_0, \quad X_1 \sim \pi_1 \end{cases} \tag{2}$$

where dB_t is standard d-dimensional Brownian motion. To define more complex systems where the *optimal* dynamics cannot be accurately captured by minimizing the standard squared-Euclidean cost in entropic OT (Vargas et al., 2021), the Generalized Schrödinger Bridge (GSB) problem introduces an additional non-linear state-cost $V_t(X_t)$ (Chen et al., 2021b; Chen & Georgiou, 2016; Liu et al., 2022). The minimization objective becomes

$$\min_{u_t} \int_0^1 \mathbb{E}_{p_t} \left[\frac{1}{2} \|u_t(X_t)\|^2 + V_t(X_t) \right] dt \tag{3}$$

The state cost can also be interpreted as the *potential energy* of the system at state X_t.

3 BRANCHED SCHRÖDINGER BRIDGE MATCHING

A **key challenge** in trajectory matching is simulating trajectories that reconstruct multi-modal marginals. Existing Schrödinger bridge and flow matching techniques have approached this problem by reconstructing the weighted multi-modal distribution by simulating many independent particle trajectories. To ensure that the intermediate trajectories follow a path distribution that approximates that observed in the data, **multimarginal Schrödinger bridge matching** interpolates between intermediate distributions along the path at specified time steps (Shen et al., 2024; Chen et al., 2023; Theodoropoulos et al., 2025).

However, these methods remain limited in their ability to dynamically model *branched trajectories*, where the marginal distributions have distinct modes that diverge in the state space following some underlying energy landscape. When there is limited ground truth snapshot data of the intermediate marginals, existing methods fail to accurately infer intermediate dynamics that follow some potential energy manifold. Furthermore, simulating a branched path distribution containing multi-modal marginals with independent particle simulations can suffer from mode collapse, where particles are biased towards a high-density mode in the distribution or only traverse low-energy intermediate paths.

To address this challenge, we introduce **Branched Schrödinger Bridge Matching (BranchSBM)**, a framework that infers intermediate branched dynamics by learning a *set* of diverging velocity networks that accurately reconstruct a multi-modal target distribution following a potential energy landscape V_t and growth networks that optimally distribute mass across the set of branches while recovering

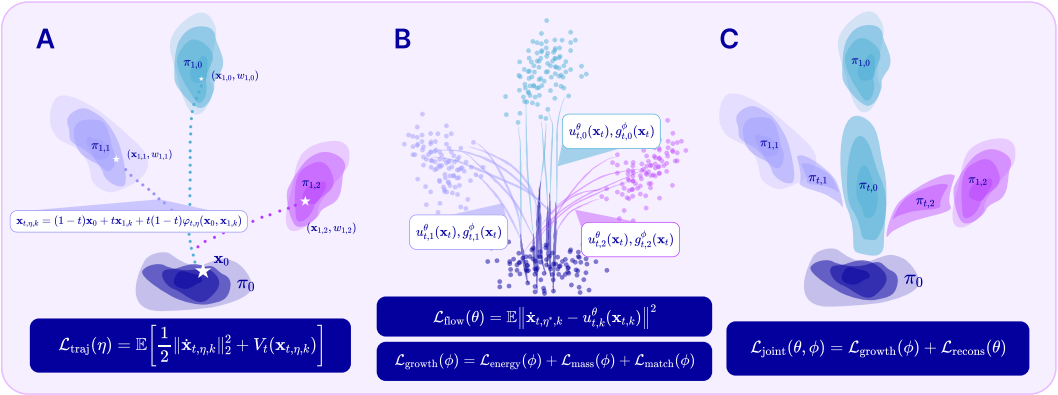

Figure 1: **Branched Schrödinger Bridge Matching (A)** Stage 1 trains a correction term that learns the optimal interpolant conditioned on endpoints **(B)** Stage 2 and 3 trains a separate flow and growth network for each branch independently **(C)** Stage 4 jointly optimizes the flow and growth networks to minimize the energy, mass, and matching loss.

the target density. Crucially, our approach does not require explicit samples from intermediate time points and can model branched, multi-modal dynamics from simulating a single sample from the initial distribution.

3.1 Unbalanced Conditional Stochastic Optimal Control

Unbalanced Generalized Schrödinger Bridge Problem Extending the definition of the Generalized Schrödinger Bridge (GSB) problem in Equation 3, we define the Unbalanced GSB problem by scaling the minimization objective by a time-dependent weight $w_t(X_t) = w_0 + \int_0^t g_s(X_s)ds$ that evolves according to a time-varying growth rate $g_t(X_t) : \mathbb{R}^d \times [0,1] \to \mathbb{R}$.

$$\min_{u_t, g_t} \int_0^1 \mathbb{E}_{p_t}\left[\frac{1}{2}\|u_t(X_t)\|^2 + V_t(X_t)\right] w_t(X_t)dt \quad \text{s.t.} \quad \begin{cases} dX_t = u_t(X_t)dt + \sigma dB_t \\ X_0 \sim \pi_0, \quad X_1 \sim \pi_1 \\ w_0(X_0) = w_0^\star, \quad w_1(X_1) = w_1^\star \end{cases} \tag{4}$$

Unbalanced Conditional Stochastic Optimal Control (CondSOC) Now, we show that we can solve the Unbalanced GSB problem as an Unbalanced CondSOC problem where the optimal drift u_t and growth g_t minimize the expectation of the objective in (4) conditioned on pairs of endpoints.

Proposition 1 (Unbalanced Conditional Stochastic Optimal Control)**.** *Suppose the marginal density can be decomposed as $p_t(X_t) = \int_{\pi_{0,1}} p_t(X_t|\mathbf{x}_0, \mathbf{x}_1)p_{0,1}(\mathbf{x}_0, \mathbf{x}_1)d\pi_{0,1}$, where $\pi_{0,1}$ is a fixed joint coupling of the data. Then, we can identify the optimal drift $u_t^\star$ and growth $g_t^\star$ that solves the Unbalanced GSB problem in (4) by minimizing the Unbalanced Conditional Stochastic Optimal Control objective given by*

$$\min_{u_t, g_t} \mathbb{E}_{(\mathbf{x}_0, \mathbf{x}_1)\sim\pi_{0,1}}\left[\int_0^1 \mathbb{E}_{p_{t|0,1}}\left[\frac{1}{2}\|u_t(X_t|\mathbf{x}_0, \mathbf{x}_1)\|^2 + V_t(X_t)\right] w_t(X_t)dt\right] \tag{5}$$

$$\text{s.t.} \quad dX_t = u_t(X_t|\mathbf{x}_0, \mathbf{x}_1)dt + \sigma dB_t, \quad X_0 = \mathbf{x}_0, \quad X_1 = \mathbf{x}_1 \quad w_0(X_0) = w_0^\star, \quad w_1(X_1) = w_1^\star \tag{6}$$

where $w_t = w_0 + \int_0^t g_s(X_s)ds$ is the time-dependent weight initialized at $w_0^\star$, u_t is the drift, g_t is the growth rate, and $\pi_{0,1}$ is the weighted coupling of paired endpoints $(\mathbf{x}_0, w_0^\star, \mathbf{x}_1, w_1^\star) \sim \pi_{0,1}$.

The proof is provided in Appendix C.1. This defines the objective for us to tractably solve the Unbalanced GSB problem by conditioning on a finite set of endpoint pairs in the dataset.

3.2 BranchSBM: Sum of Unbalanced CondSOC Problems

Branched Generalized Schrödinger Bridge Problem Given the Unbalanced GSB problem, we define the Branched GSB problem as minimizing the sum of Unbalanced GSB problems across all branches. All mass begins along a primary path indexed $k = 0$ with initial weight 1. Over $t \in [0,1]$,

mass is transferred across K secondary branches with initial weight 0 and target weight $w_{1,k}$ such that it minimizes the objective defined as

$$\min_{\{u_{t,k},g_{t,k}\}_{k=0}^{K}} \int_0^1 \left\{ \mathbb{E}_{p_{t,0}} \left[\frac{1}{2}\|u_{t,0}(X_{t,0})\|^2 + V_t(X_{t,0}) \right] w_{t,0} + \sum_{k=1}^{K} \mathbb{E}_{p_{t,k}} \left[\frac{1}{2}\|u_{t,k}(X_{t,k})\|^2 + V_t(X_{t,k}) \right] w_{t,k} \right\} dt$$
$$\text{s.t. } dX_{t,k} = u_{t,k}(X_{t,k})dt + \sigma dB_t,\ X_0 \sim \pi_0,\ X_{1,k} \sim \pi_{1,k},\ w_{0,k} = \delta_{k=0},\ w_{1,k} = w_{1,k} \tag{7}$$

When total mass across branches is conserved, we enforce $\sum_{k=0}^{K} w_{t,k} = 1$ for all $t \in [0,1]$, which constrains the growth rates such that $g_{t,0}(X_{t,0}) + \sum_{k=1}^{K} g_{t,k}(X_{t,k}) = 0$. This ensures that mass lost from the primary branch (when $g_{t,0} < 0$) is redistributed among the secondary branches (where $g_{t,k} > 0$). The primary branch evolves from an initial weight of 1 according to $w_{t,0} = 1 + \int_0^t g_s(X_{s,0})ds$ and the K secondary branches grow from the primary branch from weight 0 according to $w_{t,k} = \int_0^t g_s(X_{s,k})ds$.

Branched Conditional Stochastic Optimal Control Following a similar procedure as shown for the Unbalanced GSB problem, we can reformulate the Branched GSB problem as solving the Branched CondSOC problem where we optimize a set of parameterized drift $\{u_{t,k}\}_{k=0}^{K}$ and growth $\{g_{t,k}\}_{k=0}^{K}$ networks by minimizing the energy of the conditional trajectories between paired samples $(\mathbf{x}_0, \{\mathbf{x}_{1,k}\}_{k=0}^{K}) \sim \{p_{0,1,k}\}_{k=0}^{K}$.

Proposition 2 (Branched Conditional Stochastic Optimal Control)**.** *For each branch, let* $p_{t,k}(X_{t,k}) = \mathbb{E}_{p_{0,1k}}[p_{t,k}(X_{t,k}|\mathbf{x}_0, \mathbf{x}_{1,k})]$*, where* $\pi_{0,1,k}$ *is the joint coupling distribution of samples* $\mathbf{x}_0 \sim \pi_0$ *from the initial distribution and* $\mathbf{x}_{1,k} \sim \pi_{1,k}$ *from the* k*th target distribution. Then, we can identify the set of optimal drift and growth functions* $\{u^\star_{t,k}, g^\star_{t,k}\}_{k=0}^{K}$ *that solve the Branched GSB problem in (3.2) by minimizing the sum of Unbalanced CondSOC objectives given by*

$$\min_{\{u_{t,k},g_{t,k}\}_{k=0}^{K}} \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_{1,0})\sim\pi_{0,1,0}} \int_0^1 \left\{ \mathbb{E}_{p_{t|0,1,0}} \left[\frac{1}{2}\|u_{t,0}(X_{t,0})\|^2 + V_t(X_{t,0}) \right] w_{t,0} \right.$$
$$\left. + \sum_{k=1}^{K} \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_{1,k})\sim\pi_{0,1,k}} \int_0^1 \mathbb{E}_{p_{t|0,1,k}} \left[\frac{1}{2}\|u_{t,k}(X_{t,k})\|^2 + V_t(X_{t,k}) \right] w_{t,k} \right\} dt \tag{8}$$
$$\textit{s.t. } dX_{t,k} = u_{t,k}(X_{t,k})dt + \sigma dB_t,\ X_0 = \mathbf{x}_0,\ X_{1,k} = \mathbf{x}_{1,k},\ w_{0,k} = \delta_{k=0},\ w_{1,k} = w_{1,k} \tag{9}$$

where $w_{t,0} = 1 + \int_0^t g_{s,1}(\mathbf{x}_{s,1})ds$ *is the weight of the primary paths initialized at 1 and* $w_{t,k} = \int_0^t g_{s,k}(\mathbf{x}_{s,k})ds$ *are the weights of the* K *secondary branches initialized at 0.*

The proof is given in Appendix C.2. This defines the objective for us to tractably solve the Branched GSB problem in Section 4 by conditioning on a discrete set of branched endpoint pairs in the dataset.

Remark 1. *When* $g_{t,0}(X_{t,0}) = 0$ *and* $g_{t,k}(X_{t,k}) = 0$ *for all* $t \in [0,1]$ *and* $k \in \{1, \dots, K\}$*, then the Branched CondSOC problem is the solution to the single path GSB problem.*

4 Learning BranchSBM Using Neural Networks

Given an initial data distribution π_0 and $K+1$ target distributions $\{\pi_{1,k}\}_{k=0}^{K}$, we aim to parameterize the optimal drift and growth fields that solve the Branched CondSOC problem in Proposition 2.

4.1 Branched Neural Interpolant Optimization

Since the optimal trajectory under the state cost $V_t(X_t)$ follows a non-linear cost manifold, given a pair of endpoints $(\mathbf{x}_0, \mathbf{x}_{1,k})$, we train a neural path interpolant $\varphi_{t,\eta}(\mathbf{x}_0, \mathbf{x}_{1,k}) : \mathbb{R}^d \times \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ that defines the intermediate state $\mathbf{x}_{t,\eta,k}$ and velocity $\dot{\mathbf{x}}_{t,\eta,k} = \partial_t \mathbf{x}_{t,\eta,k}$ at time t, which minimizes (2). We note that the neural interpolant has been introduced previously for single-path interpolants in Kapuśniak et al. (2024); Neklyudov et al. (2023). We define $\mathbf{x}_{t,\eta,k}$ to be bounded at the endpoints as given by

$$\mathbf{x}_{t,\eta,k} = (1-t)\mathbf{x}_0 + t\mathbf{x}_{1,k} + t(1-t)\varphi_{t,\eta}(\mathbf{x}_0, \mathbf{x}_{1,k}) \tag{10}$$
$$\dot{\mathbf{x}}_{t,\eta,k} = \mathbf{x}_{1,k} - \mathbf{x}_0 + t(1-t)\dot{\varphi}_{t,\eta}(\mathbf{x}_0, \mathbf{x}_{1,k}) + (1-2t)\varphi_{t,\eta}(\mathbf{x}_0, \mathbf{x}_{1,k}) \tag{11}$$

To optimize $\varphi_{t,\eta}(\mathbf{x}_0, \mathbf{x}_{1,k})$ such that it predicts the energy-minimizing trajectory, we minimize the trajectory loss $\mathcal{L}_{\text{traj}}$ defined as

$$\mathcal{L}_{\text{traj}}(\eta) = \sum_{k=0}^{K} \int_0^1 \mathbb{E}_{(\mathbf{x}_0, \mathbf{x}_{1,k}) \sim \pi_{0,1,k}} \left[\frac{1}{2} \|\dot{\mathbf{x}}_{t,\eta,k}\|_2^2 + V_t(\mathbf{x}_{t,\eta,k}) \right] dt \tag{12}$$

After convergence, Stage 1 returns the network $\varphi^\star_{t,\eta}(\mathbf{x}_0, \mathbf{x}_{1,k})$ that generates the optimal conditional velocity $\dot{\mathbf{x}}^\star_{t,\eta,k}$ which defines the matching objective in Stage 2. In Stage 2, we parameterize a set of neural drift fields $u^\theta_{t,k}(\mathbf{x}_{t,k}) : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ that *generates* the *mixture of bridges* defined in Stage 1 by minimizing the conditional flow matching loss (Lipman et al., 2023; Tong et al., 2024a).

$$\mathcal{L}_{\text{flow}}(\theta) = \sum_{k=0}^{K} \int_0^1 \mathbb{E}_{(\mathbf{x}_0, \mathbf{x}_{1,k}) \sim \pi_{0,1,k}} \left\| \dot{\mathbf{x}}^\star_{t,\eta,k} - u^\theta_{t,k}(\mathbf{x}_{t,k}) \right\|_2^2 dt \tag{13}$$

Proposition 3 (Solving the GSB Problem with Stage 1 and 2 Training)**.** *Stage 1 and Stage 2 training yield the optimal drift $u^\star_t(X_t)$ that generates the optimal marginal probability distribution $p^\star_t(X_t)$ that solves the GSB problem in (3).*

The proof is provided in Appendix C.3. Since the drift for each branch $u^\theta_{t,k}(X_t)$ are trained independently in Stage 2, we can extend this result across all $K+1$ branches and conclude that the sequential Stage 1 and Stage 2 training procedures yields the optimal set of drifts $\{u^\star_{t,k}\}_{k=0}^K$ that *generate* the optimal probability paths $\{p^\star_{t,k}\}_{k=0}^K$ that solves the GSB problem for each branch.

4.2 Learning the Energy-Minimizing Branching Dynamics

Branched Energy Loss To solve the Branched CondSOC problem defined in Proposition 2, we minimize a branched energy loss $\mathcal{L}_{\text{energy}}$ defined as

$$\mathcal{L}_{\text{energy}}(\theta, \phi) = \int_0^1 \mathbb{E}_{\{p_{t,k}\}_{k=0}^K} \Bigg\{ \underbrace{\left[\frac{1}{2} \|u^\theta_{t,0}(\mathbf{x}_{t,0})\|^2 + V_t(\mathbf{x}_{t,0}) \right] w^\phi_{t,0}}_{\text{primary trajectory}} + \underbrace{\sum_{k=1}^{K} \left[\frac{1}{2} \|u^\theta_{t,k}(\mathbf{x}_{t,k})\|^2 + V_t(\mathbf{x}_{t,k}) \right] w^\phi_{t,k}}_{K \text{ branches}} \Bigg\} dt$$

$$\text{s.t. } w^\phi_{t,0} = 1 + \int_0^t g^\phi_{s,1}(\mathbf{x}_{s,1}) ds, \quad w^\phi_{t,k} = \int_0^t g^\phi_{s,k}(\mathbf{x}_{s,k}) ds \tag{14}$$

where $(\mathbf{x}_0, \mathbf{x}_{1,0})$ are the endpoints of the *primary path*. At time $t = 0$, the primary path has weight 1 and the K branches have weights 0. Over $t \in [0,1]$, the weight of the primary path changes according to $g^\phi_{t,0}(\mathbf{x}_{t,0})$ and supplies mass to the K branches, which grow at rates $g^\phi_{t,k}(\mathbf{x}_{t,k}) \geq 0$ (Lemma 2). Intuitively, the branched energy loss optimizes the branching growth rates such that they are non-zero when branching is favored over the primary path.

Weight Matching Loss We define a weight matching loss $\mathcal{L}_{\text{match}}$ that aims to minimize the difference between the predicted weights of each branch at $t = 1$, obtained by integrating the growth function $g^\phi_{t,k}(X_t)$ over $t \in [0,1]$, and the true weights of each terminal distribution $\{w^\star_{1,k}\}_{k=0}^K$.

$$\mathcal{L}_{\text{match}}(\phi) = \sum_{k=0}^{K} \mathbb{E}_{p_{1,k}} \left(w^\phi_{1,k}(\mathbf{x}_{1,k}) - w^\star_{1,k} \right)^2, \text{ s.t. } w^\phi_{1,k}(\mathbf{x}_{1,k}) = w_{0,k} + \int_0^1 g^\phi_{t,k}(\mathbf{x}_{t,k}) dt \tag{15}$$

where $w^\star_{1,k} = N_k / N_{\text{total}}$ is the fraction of the population in the kth target distribution.

Mass Conservation Loss To ensure that the growth rate satisfies conservation of total mass at all times $t \in [0,1]$, we define a mass loss $\mathcal{L}_{\text{mass}}$ that enforces the sum of the weights of all $K+1$ branches matches the true total weight at time t denoted as w_t^{total}.

$$\mathcal{L}_{\text{mass}}(\phi) = \int_0^1 \mathbb{E}_{\{p_{t,k}\}_{k=0}^K} \left[\left(\sum_{k=0}^{K} w^\phi_{t,k}(\mathbf{x}_{t,k}) - w_t^{\text{total}} \right)^2 + \sum_{k=0}^{K} \max\left(0, -w^\phi_{t,k}(\mathbf{x}_{t,k})\right) \right] dt \tag{16}$$

where $\max(0, -w^\phi_{t,k})$ assigns an additional linear penalty for negative weight predictions. For the balanced branched SBM problem where the total mass is conserved, we have $w_t^{\text{total}} = 1$.

Training the Growth Networks In Stage 3, we train the growth networks $\{g^{\phi}_{t,k}\}_{k=0}^{K}$ by fixing the weights of the flow networks $\{u^{\theta}_{t,k}\}_{k=0}^{K}$ and minimizing the weighted combined loss $\mathcal{L}_{\text{growth}}$ with an additional growth penalty term $\|g^{\phi}_{t,k}\|_2^2$ to ensure coercivity of $\mathcal{L}_{\text{growth}}$

$$\mathcal{L}_{\text{growth}}(\phi) = \lambda_{\text{energy}}\mathcal{L}_{\text{energy}}(\theta,\phi) + \lambda_{\text{match}}\mathcal{L}_{\text{match}}(\phi) + \lambda_{\text{mass}}\mathcal{L}_{\text{mass}}(\phi) + \lambda_{\text{growth}}\sum_{k=0}^{K}\|g^{\phi}_{t,k}\|_2^2 \tag{17}$$

We show in Lemma 2 that the optimal growth rates across the K secondary branches are non-decreasing; however, mass destruction can still be modeled by defining an additional branch with target weight equal to the ratio of mass lost over $t \in [0,1]$. To ensure that the set of optimal growth functions $g^{\star}$ exists, we establish Proposition 4 (see proof in Appendix C.4).

Proposition 4 (Existence of Optimal Growth Functions)**.** *Assume the state space $\mathcal{X} \subseteq \mathbb{R}^d$ is a bounded domain within $\mathbb{R}^d$. Let the optimal probability density of branch k be a known non-negative function bounded in $[0,1]$, denoted as $p^{\star}_{t,k} : \mathcal{X} \times [0,1] \to [0,1] \in L^{\infty}(\mathcal{X} \times [0,1])$. By Lemma 2, we can define the set of feasible growth functions in the set of square-integrable functions L^2 as*

$$\mathcal{G} := \{g = (g_{t,0}, \ldots, g_{t,K}) \in L^2(\mathcal{X} \times [0,1]) \mid g_{t,k}(\mathbf{x}) : \mathcal{X} \times [0,1] \to \mathbb{R}\ ,\ g_{t,k}(\mathbf{x}) \geq 0\} \tag{18}$$

Let the growth loss be the functional $\mathcal{L}(g) : L^2(\mathcal{X} \times [0,1]) \to \mathbb{R}$. Then, there exists an optimal function $g^{\star} = (g^{\star}_{t,0}, \ldots, g^{\star}_{t,K}) \in L^2$ where $g^{\star}_{t,k} \in \mathcal{G}$ such that $\mathcal{L}(g^{\star}) = \inf_{g\in\mathcal{G}} \mathcal{L}(g)$ which can be obtained by minimizing $\mathcal{L}(g)$ over $\mathcal{G}$.

Final Joint Training In the final Stage 4, we train the weights for both the flow and growth networks $\{u^{\theta}_{t,k}, g^{\phi}_{t,k}\}_{k=0}^{K}$ by minimizing $\mathcal{L}_{\text{growth}}$ from Stage 3 in addition to a reconstruction loss $\mathcal{L}_{\text{recons}}$ that ensures the endpoint distribution at time $t = 1$ is maintained.

$$\mathcal{L}_{\text{recons}}(\theta) = \sum_{k=0}^{K} \mathbb{E}_{p_{1,k}} \sum_{\mathbf{x}_1 \in \mathcal{N}_n(\mathbf{x}_{1,k})} \max\left(0, \|\tilde{\mathbf{x}}_{1,k} - \mathbf{x}_{1,k}\|_2 - \epsilon\right) \tag{19}$$

where $\mathcal{N}_n(\mathbf{x}_{1,k})$ is the set of n-nearest neighbors to the reconstructed state $\tilde{\mathbf{x}}_{1,k} \sim p_{1,k}$ at time $t = 1$ from the data points $\mathbf{x}_{1,k} \sim \pi_{1,k}$ at time $t = 1$.

Our multi-stage training scheme decomposes the Branched CondSOC problem into two parts. We first independently learn an optimal drift field for each branch, which is a vector field over the state space that propagates mass flow in the direction of each target distribution. Then, we fix the drift fields and learn the growth dynamics that determine the optimal distribution of mass over the branches.

5 Experiments

We evaluate BranchSBM on a variety of branched matching tasks with different state costs $V_t(X_t)$, including multi-path LiDAR navigation (Section 5.1), modeling differentiating single-cell population dynamics (Section 5.2), and predicting heterogeneous cell-states after perturbation (Section 5.3).

5.1 Branched LiDAR Surface Navigation

First, we evaluate BranchSBM for navigating branched paths along the surface of a 3-dimensional LiDAR manifold, from an initial distribution to two distinct target distributions (Figure 3).

Setup We define a single initial Gaussian mixture π_0 and two target Gaussian mixtures $\pi_{1,0}, \pi_{1,1}$ on either side of the mountain (Figure 3). We sample 5000 points i.i.d. from each of the Gaussian mixtures and assign all endpoints a target weight of $w_{1,0} = w_{1,1} = 0.5$. To ensure trajectories follow the LiDAR manifold, we define the state cost $V_t^{\text{LAND}}(X_t)$ as the data-dependent LAND metric (Kapuśniak et al., 2024; Arvanitidis et al., 2016), which assigns lower costs in regions near coordinates in the LiDAR dataset. Further experimental details are provided in Appendix E.3.

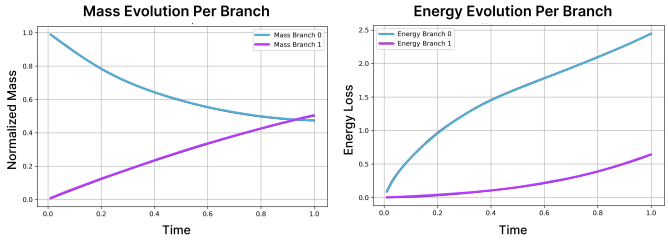

Figure 2: Plot of weight (left) and energy (right) calculated with (14) of each branch over time $t \in [0, 1]$. Mass is transferred from the primary branch to branch 1, and both converge to the target weight of 0.5 at $t = 1$. Both plots represent the average over trajectories from samples in the validation set.

Table 1: **Benchmark of BranchSBM against single-branch SBM on multi-path surface navigation.** Wasserstein distances ($\mathcal{W}_1$ and $\mathcal{W}_2$) between the reconstructed and ground-truth distributions with $N_{\text{steps}} = 100$ Euler steps at time $t = 1$ from validation samples in the initial distribution. Results are averaged over 5 independent runs.

Model	$\mathcal{W}_1$ (↓)	$\mathcal{W}_2$ (↓)
Single Branch SBM	$0.975_{\pm 0.009}$	$1.285_{\pm 0.007}$
BranchSBM	$\mathbf{0.239}_{\pm 0.001}$	$\mathbf{0.309}_{\pm 0.003}$

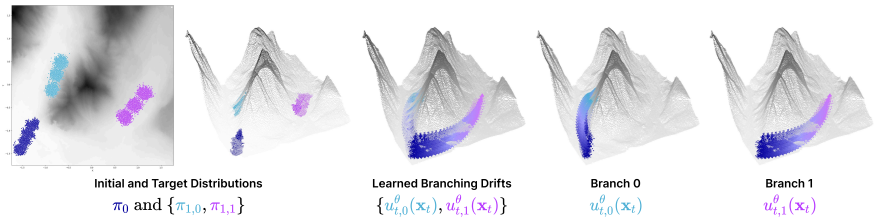

Figure 3: **Application of BranchSBM on Learning Branched Paths on a LiDAR Manifold.** Plots of the initial and target distributions, learned interpolants, and learned branched trajectories on the LiDAR manifold.

Results We show that BranchSBM can learn distinct, non-linear branched paths that curve along the 3-dimensional LiDAR manifold while minimizing the kinetic energy and state-cost. From the mass and energy curves in Figure 2, we see that mass begins in the primary branch (branch 0) and is gradually transferred to the secondary branch (branch 1) over $t \in [0, 1]$, with both curves converging to the target weight of 0.1 at $t = 1$. As mass is transferred, the slope of the cumulative energy curve decreases in branch 0 and increases in branch 1, reflecting the true energy dynamics. In Figure 3, we observe that the branching occurs at the edge of the inclined mountain, indicating that the model can determine the optimal branching time based on the paths of lowest potential energy. As shown in Table 1, BranchSBM reconstructs the endpoint distributions with significantly higher accuracy in comparison to single-branch SBM. In total, we demonstrate the capability of BranchSBM to learn branched trajectories on complex 3D manifolds.

5.2 Differentiating Single-Cell Population Dynamics

BranchSBM is uniquely positioned to model single-cell population dynamics where a homogeneous cell population (e.g., progenitor cells) differentiates into several distinct subpopulation branches, each of which independently undergoes growth dynamics. Here, we demonstrate this capability on mouse hematopoiesis data (Sha et al., 2023; Weinreb et al., 2020) and pancreatic β-cell differentiation data (Veres et al., 2019).

Mouse-Hematopoesis Results We use a dataset consisting of mouse hematopoiesis scRNA-seq data analyzed by a lineage tracing technique from (Sha et al., 2023; Weinreb et al., 2020). This data contains three time points t_i for $i \in \{0, 1, 2\}$ that are projected to two-dimensional representations $\mathbf{x} \in \mathbb{R}^2$ referred to as force-directed layouts or SPRING plots. From the plotted data, we can

Table 2: **Results for Modeling Single-Cell Differentiation.** Wasserstein distances ($\mathcal{W}_1$ and $\mathcal{W}_2$) between simulated and ground-truth cell distributions at time t_1 and t_2 on the validation dataset. BranchSBM reconstructs both intermediate and terminal states significantly better than single-branch SBM. Results are averaged over 5 independent runs.

Model	Single Branch SBM		**BranchSBM**	
Time ↓	$\mathcal{W}_1$ (↓)	$\mathcal{W}_2$ (↓)	$\mathcal{W}_1$ (↓)	$\mathcal{W}_2$ (↓)
t_1	$0.582_{\pm 0.020}$	$0.703_{\pm 0.008}$	$\mathbf{0.366}_{\pm 0.034}$	$\mathbf{0.479}_{\pm 0.044}$
t_2	$0.940_{\pm 0.075}$	$1.037_{\pm 0.074}$	$\mathbf{0.210}_{\pm 0.042}$	$\mathbf{0.265}_{\pm 0.046}$

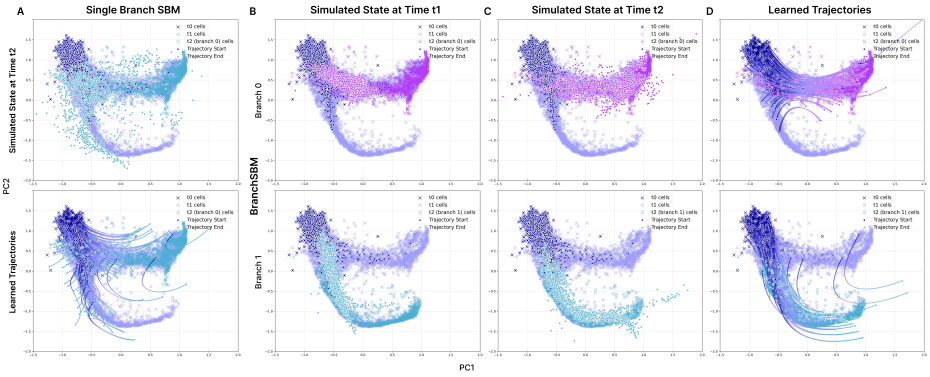

Figure 4: **Application of BranchSBM on Modeling Differentiating Single-Cell Population Dynamics.** Mouse hematopoiesis scRNA-seq data is provided for three time points t_0, t_1, t_2. **(A)** Simulated states (top) and trajectories (bottom) at time t_1 using single-branch SBM. **(B)** Simulated states with BranchSBM at t_1 ($t = 0.5$) and **(C)** t_2 ($t = 1$). **(D)** Learned trajectories over the interval $t \in [t_0, t_2]$ on validation samples.

Table 3: **Results for pancreatic β-cell differentiation experiment.** 1-Wasserstein distance ($\mathcal{W}_1$) of intermediate distributions generated by DeepRUOT (Zhang et al., 2025c), CytoBridge (Zhang et al., 2025d), and BranchSBM (Ours) at eight different time points. † denotes values taken from (Zhang et al., 2025d).

Model	$t = 1$ $\mathcal{W}_1$	$t = 2$ $\mathcal{W}_1$	$t = 3$ $\mathcal{W}_1$	$t = 4$ $\mathcal{W}_1$	$t = 5$ $\mathcal{W}_1$	$t = 6$ $\mathcal{W}_1$	$t = 7$ $\mathcal{W}_1$
DeepRUOT †	**8.0447**$_{\pm 0.0005}$	8.0773$_{\pm 0.0021}$	7.6301$_{\pm 0.0032}$	**8.0064**$_{\pm 0.0042}$	**7.9018**$_{\pm 0.0117}$	8.3977$_{\pm 0.0102}$	7.8346$_{\pm 0.0109}$
CytoBridge †	8.0448$_{\pm 0.0005}$	8.0771$_{\pm 0.0021}$	**7.6299**$_{\pm 0.0032}$	8.0066$_{\pm 0.0043}$	**7.9018**$_{\pm 0.0117}$	**8.3974**$_{\pm 0.0102}$	7.8343$_{\pm 0.0109}$
BranchSBM (Ours)	11.9774$_{\pm 0.0000}$	**7.4643**$_{\pm 2.4031}$	11.5204$_{\pm 0.0000}$	11.2593$_{\pm 0.0000}$	10.2888$_{\pm 0.0000}$	8.7301$_{\pm 0.0000}$	**6.8702**$_{\pm 0.0000}$

observe two clear branches that
indicate the differentiation of progenitor cells into two distinct cell fates (Figure 4). We use k-means clustering to define two distinct target distributions $\pi_{1,0}$ and $\pi_{1,1}$ of samples at time t_2 and set their target weights equal to $w_{1,0} = w_{1,1} = 0.5$ due to the equal ratio of cells (Figure 9). We used samples across all time steps t_i for $i \in \{0, 1, 2\}$ to define the data manifold via the LAND metric $V_{t,\eta}^{\text{LAND}}$. BranchSBM was trained on pairs sampled only from t_0 and t_2, and samples from t_1 were held out for evaluation. For comparison, we trained a single-branch SBM model with both clusters at t_2 as the target distribution.

After evaluating the reconstructed distributions at the intermediate held-out time point t_1 and final time point t_2 ($t = 1$) from simulating validation samples from the initial distribution $\mathbf{x}_0 \sim \pi_0$. In Figure 4A, we observe that single-branch SBM trained with a single target distribution p_1 containing both terminal fates fails to learn distinct branched trajectories, and the simulated cell states at time t_2 do not reach either of the terminal distributions. In contrast, we show that BranchSBM simulates branched states at intermediate time steps not included in the training data while accurately reconstructing both target distributions with significantly lower 1-Wasserstein and 2-Wasserstein distances compared to the single-branch SBM model (Figure 4B-D; Table 2).

Pancreatic β-Cell Differentiation Results We use a dataset consisting of pancreatic β-cell differentiation data (Veres et al., 2019) containing 51, 274 cells collected over eight time points as they evolve from human pluripotent stem cells to pancreatic β-like cells. This data contains eight time points t_i for $i \in \{0, \ldots, 7\}$ that are projected to 30-dimensional PC representations $\mathbf{x} \in \mathbb{R}^{30}$. We use Leiden clustering to define $K = 11$ distinct target distributions of samples at time t_7 and set their target weights relative to the total weight at t_0 (App E.4). We used samples across all time steps to define the data manifold via the RBF metric $V_{t,\eta}^{\text{RBF}}$. BranchSBM was trained on pairs sampled only from t_0 and t_7, and the distributions at all intermediate time points were inferred from learning to minimize the distance from the data manifold over time. For baselines, we compared two SOTA methods for single-cell trajectory modeling: DeepRUOT (Zhang et al., 2025c) and CytoBridge (Zhang et al., 2025d), which are trained explicitly on intermediate snapshots to reconstruct the stochastic dynamics of cells that evolve independently.

Notably, BranchSBM not only reconstructs the multi-modal terminal distribution at t_7 with superior accuracy against all baselines, but also produces intermediate trajectories that are competitive with models trained directly on intermediate snapshots using explicit reconstruction losses (Table 3). Leveraging the RBF state cost $V_{t,\eta}^{\text{RBF}}$, which encourages trajectories to remain on the underlying data manifold, BranchSBM effectively captures the true differentiation dynamics through the combined influence of the neural interpolant and the path-energy objective. These results demonstrate that BranchSBM scales reliably to a large number of branches and is particularly advantageous in settings where intermediate timepoints are sparse or unavailable, allowing the model to infer biologically meaningful trajectories even with limited temporal supervision.

5.3 Cell-State Perturbation Modeling

Predicting the effects of perturbation on cell state dynamics is a crucial problem for therapeutic design. In this experiment, we leverage BranchSBM to model the trajectories of a single cell line from a single homogeneous state to multiple heterogeneous states after a drug-induced perturbation. We demonstrate that BranchSBM is capable of modeling high-dimensional gene expression data and learning branched trajectories that accurately reconstruct diverging perturbed cell populations.

Setup For this experiment, we extract the data for a single cell line (A-549) under perturbation with Clonidine and Trametinib at 5 μL, selected based on cell abundance and response diversity from the Tahoe-100M dataset (Zhang et al., 2025a). Since both drugs had over 60K genes, we selected the top 2000 highly variable genes (HVGs) based on normalized expression and performed principal component analysis (PCA) to find the top PCs that capture the variance in the data. We set the initial distribution at $t = 0$ to be a control DMSO-treated cell population and the target distributions at $t = 1$ to be distinct clusters in the drug-treated cell population. After clustering, we identified two divergent clusters in the Clonidine-perturbed population and three in the Trametinib-perturbed population (Appendix Figure 10). To determine the weights of each branch, we take the ratio of each cluster with respect to the total perturbed cell population (Appendix Table 9). For both experiments, we simulated the top 50 PCs, which capture approximately 38% of the variance in the dataset. To further evaluate the scalability of BranchSBM on simulating trajectories in high-dimensional state spaces, we simulated the top 100 and 150 PCs for Clonidine and compared the performance across dimensions.

Table 4: **Results for Clonidine Perturbation Modeling for Increasing Principal Component Dimensions.** Maximum-mean discrepancy (MMD) across all PCs and Wasserstein distances ($\mathcal{W}_1$ and $\mathcal{W}_2$) of top 2 PCs between ground truth and reconstructed distributions at $t = 1$ simulated from the validation data at $t = 0$. Results for single-branch SBM (50 PCs) and BranchSBM (2 branches) were averaged over 5 independent runs.

Model	**RBF-MMD** (↓)	$\mathcal{W}_1$ (↓)	$\mathcal{W}_2$ (↓)
Single Branch SBM (50 PCs)	$0.279_{\pm 0.024}$	$5.124_{\pm 0.509}$	$6.149_{\pm 0.463}$
BranchSBM			
50 PCs	$0.065_{\pm 0.001}$	$\mathbf{1.076}_{\pm 0.085}$	$\mathbf{1.224}_{\pm 0.097}$
100 PCs	$\mathbf{0.053}_{\pm 0.002}$	$1.832_{\pm 0.174}$	$2.037_{\pm 0.174}$
150 PCs	$0.083_{\pm 0.001}$	$1.722_{\pm 0.064}$	$1.931_{\pm 0.035}$

Given that the intermediate trajectory between the control and perturbed state is unknown, we assume that the optimal trajectory both minimizes the kinetic energy of the drift field while minimizing the distance from the space of *feasible* cell states. We define the state cost $V_t(X_t)$ with the RBF metric (Kapuśniak et al., 2024; Arvanitidis et al., 2016), which pushes the intermediate trajectory to lie near states represented in the dataset. Further details are provided in Appendix E.5.

Clonidine Perturbation Results After multi-stage training of BranchSBM with $d \in \{50, 100, 150\}$ PCs and two branched endpoints (Appendix Figure 6A), we simulated the final perturbed state of each branch at time $t = 1$ from the samples in the initial validation data distribution $\mathbf{x}_0 \sim \pi_0$ corresponding to the control DMSO condition. In Appendix Figure 6C, we demonstrate that BranchSBM accurately reconstructs the ground-truth distributions of endpoint 0 (top row) and endpoint 1 (bottom row) across increasing PC dimensions, capturing the location and spread of the dataset. To prove the necessity of our branched framework, we simulate the target distribution with only a single endpoint distribution p_1 containing both clusters with single-branch SBM and show that it only reconstructs the population of cells in endpoint 0, which represent cells closest to the control cells along PC2, and

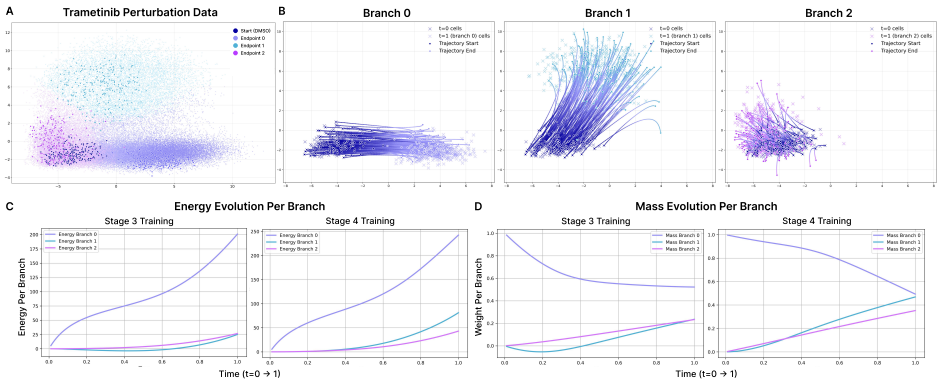

Figure 5: **Results for Trametinib Perturbation Modeling with BranchSBM. (A)** Gene expression data of DMSO control ($t = 0$) and cells after treatment with $5\mu M$ Trametinib ($t = 0$) with three distinct endpoints (purple, turquoise, and pink). **(B)** The simulated endpoints of the top 50 PCs at $t = 1$ on the validation data for each branch. **(C)** The evolution of cumulative energy across $t \in [0, 1]$ calculated as (14) along each branched trajectory after Stage 3 (growth with fixed drift) and Stage 4 (joint) training. **(D)** The evolution of mass across $t \in [0, 1]$ along each branched trajectory with target weights of $w_{1,0} = 0.603$, $w_{1,1} = 0.255$ and $w_{1,2} = 0.142$.

fails to differentiate cells in cluster 1 that differ from cluster 0 in higher-dimensional PCs (Appendix Figure 6B). Concretely, BranchSBM used across all PC dimensions outperforms single-branch SBM on only 50 PCs (Table 4), indicating that BranchSBM is required to model complex perturbation effects in high-dimensional gene expression spaces.

Trametinib Perturbation Results We further show that BranchSBM can scale beyond two branches, by modeling the perturbed cell population of Trametinib-treated cells, which diverge into three distinct clusters (Figure 5A). We trained BranchSBM with three endpoints and single-branch SBM with one endpoint containing all three clusters on the top 50 PCs. After simulating the trajectories over time $t \in [0, 1]$ on the validation cells in the control population, we show that BranchSBM generates clear trajectories to all three branched endpoints (Figure 5B) and reconstructs the overall target distribution with lower error compared to single-branch SBM (Table 5). In Figure 5C and D, we plot the evolution of cumulative energy calculated in $\mathcal{L}_{\text{energy}}(\theta, \phi)$ (14) and weight of each branch over $t \in [0, 1]$, demonstrating that BranchSBM's multi-stage training scheme effectively learns the optimal trade-off between minimizing the energy across trajectories and matching the target weights of each branch.

Table 5: **Results for Trametinib Perturbation Modeling.** Maximum-mean discrepancy (MMD) across all 50 PCs and Wasserstein distances ($\mathcal{W}_1$ and $\mathcal{W}_2$) of top 2 PCs between ground truth and reconstructed distributions at $t = 1$ simulated from the validation data at $t = 0$. Results were averaged over 5 independent runs.

Model	**RBF-MMD** (↓)	$\mathcal{W}_1$ (↓)	$\mathcal{W}_2$ (↓)
Single Branch SBM	$0.246_{\pm 0.013}$	$5.428_{\pm 0.234}$	$6.426_{\pm 0.186}$
BranchSBM	$\mathbf{0.053}_{\pm 0.001}$	$\mathbf{0.838}_{\pm 0.061}$	$\mathbf{0.973}_{\pm 0.050}$

6 Conclusion

In this work, we introduce **Branched Schrödinger Bridge Matching (BranchSBM)**, a novel matching framework that solves the Generalized Schrödinger Bridge (GSB) problem from an initial distribution to multiple weighted target distributions through the division of mass across learned branched trajectories. BranchSBM solves the branched SBM problem as a sum of Unbalanced Conditional Stochastic Optimal Control tasks, parameterizing branch-specific velocity and growth rates with neural networks to predict system trajectories with a single inference simulation. Through applications to nonlinear 3D navigation, cell differentiation, and perturbation-induced gene expression, we demonstrate that BranchSBM provides a unified and flexible framework for modeling complex branched dynamics across biological and physical systems.

Reproducibility Statement

We have taken multiple steps to ensure reproducibility of BranchSBM. All theoretical derivations are fully detailed in the main text and Appendix, with proofs of Propositions 1-4 provided in Sections C.1-C.4. Our multi-stage training procedure is described explicitly, and pseudocode is given in Algorithm 1. Experimental setups for each task are described in detail, including synthetic LiDAR navigation (Appendix E.3), differentiating single-cell dynamics (Appendix E.4), and cell-state perturbation modeling (Appendix E.5). We specify all hyperparameters and architectures in Table 11, evaluation metrics such as Wasserstein distances and MMD are defined in Equations 95, 96, and 97, and additional loss functions are given in Equations 12-17. Figures 1-5 and Tables 1-5 provide full benchmarking comparisons to baselines. We will release all code, training data splits, and pretrained BranchSBM models upon publication to allow complete reproducibility.

Ethics Statement

This work focuses on the development of a theoretical and computational framework for learning branched Schrödinger bridges in generative modeling. All experiments are conducted on publicly available or synthetic datasets, including 3D LiDAR manifolds, mouse hematopoiesis scRNA-seq data, and the Tahoe-100M perturbation dataset, and do not involve any personally identifiable, sensitive human subject data, or animal experiments. BranchSBM provides a general-purpose algorithm for modeling stochastic processes with branching dynamics, with potential applications in biological modeling, navigation, and physics. Possible risks, such as misuse in simulating sensitive biomedical data, are mitigated by the fact that BranchSBM is a general generative framework that requires curated training datasets and does not directly generate actionable biological interventions. We believe the potential benefits of this work, in advancing generative modeling theory and enabling more accurate modeling of branching systems in biology and beyond, outweigh possible risks. Code and models will be released under a research-only license to encourage responsible use.

REFERENCES

Sayali Anil Alatkar and Daifeng Wang. Artemis integrates autoencoders and schrodinger bridges to predict continuous dynamics of gene expression, cell population and perturbation from time-series single-cell data. *bioRxiv*, pp. 2025–01, 2025.

Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. A locally adaptive normal distribution. *Advances in Neural Information Processing Systems*, 2016.

Lazar Atanackovic, Xi Zhang, Brandon Amos, Mathieu Blanchette, Leo J Lee, Yoshua Bengio, Alexander Tong, and Kirill Neklyudov. Meta flow matching: Integrating vector fields on the wasserstein manifold. *arXiv preprint arXiv:2408.14608*, 2024.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 2021.

Aymeric Baradat and Hugo Lavenant. Regularized unbalanced optimal transport as entropy minimization with respect to branching brownian motion. *arXiv preprint arXiv:2111.01666*, 2021.

Bernard Beauzamy. *Introduction to Banach spaces and their geometry*, volume 68. Elsevier, 2011.

Barbora Benešová and Martin Kružík. Weak lower semicontinuity of integral functionals and applications. *arXiv*, 2016.

Charlotte Bunne, Ya-Ping Hsieh, Marco Cuturi, and Andreas Krause. The schrödinger bridge between gaussian measures has a closed form. *International Conference on Artificial Intelligence and Statistics*, 2022a.

Charlotte Bunne, Laetitia Papaxanthos, Andreas Krause, and Marco Cuturi. Proximal optimal transport modeling of population dynamics. In *International Conference on Artificial Intelligence and Statistics*, pp. 6511–6528. PMLR, 2022b.

Charlotte Bunne, Stefan G Stark, Gabriele Gut, Jacobo Sarabia Del Castillo, Mitch Levesque, Kjong-Van Lehmann, Lucas Pelkmans, Andreas Krause, and Gunnar Rätsch. Learning single-cell perturbation responses using neural optimal transport. *Nature methods*, 20(11):1759–1768, 2023.

Tianrong Chen, Guan-Horng Liu, and Evangelos A. Theodorou. Likelihood training of schrödinger bridge using forward-backward sdes theory. *International Conference on Learning Representations*, 2021a.

Tianrong Chen, Guan-Horng Liu, Molei Tao, and Evangelos A Theodorou. Deep momentum multi-marginal schrödinger bridge. *arXiv preprint arXiv:2303.01751*, 2023.

Yongxin Chen and Tryphon Georgiou. Stochastic bridges of linear systems. *IEEE Transactions on Automatic Control*, 61(2):526–531, 2016. doi: 10.1109/TAC.2015.2440567.

Yongxin Chen, Tryphon Georgiou, and Michele Pavon. On the relation between optimal transport and schrödinger bridges: A stochastic control viewpoint. *Journal of Optimization Theory and Applications*, 2016.

Yongxin Chen, Tryphon T. Georgiou, and Michele Pavon. The most likely evolution of diffusing and vanishing particles: Schrodinger bridges with unbalanced marginals. *arXiv preprint arXiv:2108.02879*, 2021b.

Lénaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Unbalanced optimal transport: Dynamic and kantorovich formulations. *Journal of Functional Analysis*, 274(11): 3090–3123, 2018. ISSN 0022-1236.

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

Bernard Dacorogna. *Direct Methods in the Calculus of Variations*. Springer New York, 1989. ISBN 9780387357799.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 2021a.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 2021b.

Valentin De Bortoli, Iryna Korshunova, Andriy Mnih, and Arnaud Doucet. Schrodinger bridge flow for unpaired data translation. *Advances in Neural Information Processing Systems*, 37: 103384–103441, 2024.

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

Alice Driessen, Benedek Harsanyi, Marianna Rapsomaniki, and Jannis Born. Towards generalizable single-cell perturbation modeling via the conditional monge gap. *arXiv preprint arXiv:2504.08328*, 2025.

Ivar Ekeland and Roger Temam. *Convex analysis and variational problems*. SIAM, 1999.

Jhanvi Garg, Xianyang Zhang, and Quan Zhou. Soft-constrained schrödinger bridge: a stochastic control approach. In *International Conference on Artificial Intelligence and Statistics*, pp. 4429–4437. PMLR, 2024.

George I Gavriilidis, Vasileios Vasileiou, Aspasia Orfanou, Naveed Ishaque, and Fotis Psomopoulos. A mini-review on perturbation modelling across single-cell omic modalities. *Computational and Structural Biotechnology Journal*, 23:1886, 2024.

Nikita Gushchin, Sergei Kholkin, Evgeny Burnaev, and Alexander Korotin. Light and optimal schrödinger bridge matching. In *Forty-first International Conference on Machine Learning*, 2024.

Guillaume Huguet, Daniel Sumner Magruder, Alexander Tong, Oluwadamilola Fasina, Manik Kuchroo, Guy Wolf, and Smita Krishnaswamy. Manifold interpolating optimal-transport flows for trajectory inference. *Advances in neural information processing systems*, 35:29705–29718, 2022.

Kacper Kapuśniak, Peter Potaptchik, Teodora Reu, Leo Zhang, Alexander Tong, Michael Bronstein, Avishek Joey Bose, and Francesco Di Giovanni. Metric flow matching for smooth interpolations on the data manifold. *Advances in Neural Information Processing Systems*, 2024.

Jun Hyeong Kim, Seonghwan Kim, Seokhyun Moon, Hyeongwoo Kim, Jeheon Woo, and Woo Youn Kim. Discrete diffusion schrödinger bridge matching for graph transformation. *International Conference on Learning Representations*, 2024.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference for Learning Representations*, 2014.

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

Mutsumi Kobayashi, Misato Kobayashi, Junko Odajima, Keiko Shioda, Young Sun Hwang, Kotaro Sasaki, Pranam Chatterjee, Christian Kramme, Richie E Kohman, George M Church, et al. Expanding homogeneous culture of human primordial germ cell-like cells maintaining germline features without serum or feeder layers. *Stem Cell Reports*, 17(3):507–521, 2022.

Alexander Korotin, Nikita Gushchin, and Evgeny Burnaev. Light schrödinger bridge. *arXiv preprint arXiv:2310.01174*, 2023.

Christian Kramme, Alexandru M. Plesa, Helen H. Wang, Bennett Wolf, Merrick Pierson Smela, Xiaoge Guo, Richie E. Kohman, Pranam Chatterjee, and George M. Church. An integrated pipeline for mammalian genetic screening. *Cell Reports Methods*, 1(6):100082, October 2021.

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

Peter Lippmann, Enrique Fita Sanmartín, and Fred A Hamprecht. Theory and approximate solvers for branched optimal transport with multiple sources. *Advances in Neural Information Processing Systems*, 35:267–279, 2022.

Guan-Horng Liu, Tianrong Chen, Oswin So, and Evangelos A. Theodorou. Deep generalized schrödinger bridge. *Advances in Neural Information Processing Systems*, 2022.

Guan-Horng Liu, Yaron Lipman, Maximilian Nickel, Brian Karrer, Evangelos A. Theodorou, and Ricky T. Q. Chen. Generalized schrödinger bridge matching. *International Conference on Learning Representations*, 2023a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *International Conference on Learning Representations*, 2023b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *International Conference for Learning Representations*, 2017.

Frederike Lübeck, Charlotte Bunne, Gabriele Gut, Jacobo Sarabia del Castillo, Lucas Pelkmans, and David Alvarez-Melis. Neural unbalanced optimal transport via cycle-consistent semi-couplings. *arXiv preprint arXiv:2209.15621*, 2022.

Christian Léonard. A survey of the schrödinger problem and some of its connections with optimal transport. *Discrete & Continuous Dynamical Systems-A*, 2014.

Stathis Megas, Daniel G Chen, Krzysztof Polanski, Moshe Eliasof, Carola-Bibiane Schönlieb, and Sarah A Teichmann. Estimation of single-cell and tissue perturbation effect in spatial transcriptomics via spatial causal disentanglement. In *The Thirteenth International Conference on Learning Representations*, 2025.

Kirill Neklyudov, Rob Brekelmans, Alexander Tong, Lazar Atanackovic, Qiang Liu, and Alireza Makhzani. A computational framework for solving wasserstein lagrangian flows. *arXiv preprint arXiv:2310.10649*, 2023.

Maxence Noble, Valentin De Bortoli, Arnaud Doucet, and Alain Durmus. Tree-based diffusion schrödinger bridge with applications to wasserstein barycenters. *Advances in Neural Information Processing Systems*, 36:55193–55236, 2023.

Matteo Pariset, Ya-Ping Hsieh, Charlotte Bunne, Andreas Krause, and Valentin De Bortoli. Unbalanced diffusion schr\" odinger bridge. *arXiv preprint arXiv:2306.09099*, 2023.

Michele Pavon, Esteban G Tabak, and Giulio Trigila. The data-driven schroedinger bridge. *arXiv*, 2018.

Michele Pavon, Giulio Trigila, and Esteban G Tabak. The data-driven schrödinger bridge. *Communications on Pure and Applied Mathematics*, 74(7):1545–1573, 2021.

Stefano Peluchetti. Bm^2: Coupled schrödinger bridge matching. *arXiv preprint arXiv:2409.09376*, 2024.

Merrick Pierson Smela, Christian C Kramme, Patrick RJ Fortuna, Bennett Wolf, Shrey Goel, Jessica Adams, Carl Ma, Sergiy Velychko, Ursula Widocki, Venkata Srikar Kavirayuni, et al. Rapid human oogonia-like cell specification via transcription factor-directed differentiation. *EMBO Reports*, pp. 1–30, 2025.

L. C. G. Rogers and David Williams. *Diffusions, Markov Processes and Martingales*. Cambridge University Press, 2000. ISBN 9780511805141.

Martin Rohbeck, Charlotte Bunne, Edward De Brouwer, Jan-Christian Huetter, Anne Biton, Kelvin Y. Chen, Aviv Regev, and Romain Lopez. Modeling complex system dynamics with flow matching across time and conditions. *The Thirteenth International Conference on Learning Representations*, 2025.

Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with gears. *Nature Biotechnology*, 42(6):927–935, 2023.

Jayoung Ryu, Charlotte Bunne, Luca Pinello, Aviv Regev, and Romain Lopez. Cross-modality matching and prediction of perturbation responses with labeled gromov-wasserstein optimal transport. *arXiv preprint arXiv:2405.00838*, 2024.

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

Erwin Schrödinger. *Über die umkehrung der naturgesetze*. Verlag der Akademie der Wissenschaften in Kommission bei Walter De Gruyter u . . . , 1931.

Yutong Sha, Yuchi Qiu, Peijie Zhou, and Qing Nie. Reconstructing growth and dynamic trajectories from single-cell transcriptomics data. *Nature Machine Intelligence*, 6(1):25–39, 2023.

Ophir Shalem, Neville E Sanjana, Ella Hartenian, Xi Shi, David A Scott, Tarjei S Mikkelsen, Dirk Heckl, Benjamin L Ebert, David E Root, John G Doench, et al. Genome-scale crispr-cas9 knockout screening in human cells. *Science*, 343(6166):84–87, 2014.

Yunyi Shen, Renato Berlinghieri, and Tamara Broderick. Multi-marginal schr\" odinger bridges with iterative reference refinement. *arXiv preprint arXiv:2408.06277*, 2024.

Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. *Advances in Neural Information Processing Systems*, 2023.

Merrick D Pierson Smela, Christian C Kramme, Patrick RJ Fortuna, Jessica L Adams, Rui Su, Edward Dong, Mutsumi Kobayashi, Garyk Brixi, Venkata Srikar Kavirayuni, Emma Tysinger, et al. Directed differentiation of human ipscs to functional ovarian granulosa-like cells via transcription factor overexpression. *Elife*, 12:e83291, 2023.

Vignesh Ram Somnath, Matteo Pariset, Ya-Ping Hsieh, Maria Rodriguez Martinez, Andreas Krause, and Charlotte Bunne. Aligned diffusion schrödinger bridges. *Uncertainty in Artificial Intelligence*, 2023.

J Michael Steele. *The Cauchy-Schwarz master class: an introduction to the art of mathematical inequalities*. Cambridge University Press, 2004.

Panagiotis Theodoropoulos, Nikolaos Komianos, Vincent Pacelli, Guan-Horng Liu, and Evangelos A. Theodorou. Feedback schrödinger bridge matching. *The Thirteenth International Conference on Learning Representations*, 2024.

Panagiotis Theodoropoulos, Augustinos D Saravanos, Evangelos A Theodorou, and Guan-Horng Liu. Momentum multi-marginal schrödinger bridge matching. *arXiv preprint arXiv:2506.10168*, 2025.

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

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024a.

Alexander Y. Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-free Schrödinger bridges via score and flow matching. *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, 238:1279–1287, 2024b.

Francisco Vargas, Pierre Thodoroff, Neil D. Lawrence, and Austen Lamacraft. Solving schrödinger bridges via maximum likelihood. *arXiv preprint arXiv:2106.02081*, 2021.

Adrian Veres, Aubrey L Faust, Henry L Bushnell, Elise N Engquist, Jennifer Hyoje-Ryu Kenty, George Harb, Yeh-Chuin Poh, Elad Sintov, Mads Gürtler, Felicia W Pagliuca, et al. Charting cellular identity during human in vitro β-cell differentiation. *Nature*, 569(7756):368–373, 2019.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2008.

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

Gefei Wang, Yuling Jiao, Qian Xu, Yang Wang, and Can Yang. Deep generative learning via schrödinger bridge. *International Conference on Machine Learning*, 2021.

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

Toshiaki Yachimura, Hanbo Wang, Yusuke Imoto, Momoko Yoshida, Sohei Tasaki, Yoji Kojima, Yukihiro Yabuta, Mitinori Saitou, and Yasuaki Hiraoka. scegot: single-cell trajectory inference framework based on entropic gaussian mixture optimal transport. *BMC bioinformatics*, 25(1):388, 2024.

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

Jesse Zhang, Airol A Ubas, Richard de Borja, Valentine Svensson, Nicole Thomas, Neha Thakar, Ian Lai, Aidan Winters, Umair Khan, and Matthew G. et al. Jones. Tahoe-100m: A giga-scale single-cell perturbation atlas for context-dependent gene function and cellular modeling. *bioRxiv*, 2025a.

Stephen Zhang, Anton Afanassiev, Laura Greenstreet, Tetsuya Matsumoto, and Geoffrey Schiebinger. Optimal transport analysis reveals trajectories in steady-state systems. *PLoS computational biology*, 17(12):e1009466, 2021.

Yuhui Zhang, Yuchang Su, Chenyu Wang, Tianhong Li, Zoe Wefers, Jeffrey Nirschl, James Burgess, Daisy Ding, Alejandro Lozano, Emma Lundberg, et al. Cellflow: Simulating cellular morphology changes via flow matching. *arXiv preprint arXiv:2502.09775*, 2025b.

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

Zhenyi Zhang, Zihan Wang, Yuhao Sun, Tiejun Li, and Peijie Zhou. Modeling cell dynamics and interactions with unbalanced mean field schrödinger bridge. *arXiv preprint arXiv:2505.11197*, 2025d.

OUTLINE OF APPENDIX

In Appendix A, we provide an extended background on the relevant theory for learning optimal stochastic bridges (A.1) and simulating trajectories on the data manifold (A.2). In Appendix B, we discuss the relationship between our proposed formulation for BranchSBM and previous related works. Appendix C provides the theoretical basis for Sections 3 and 4, including formal proofs for Proposition 1 (C.1), Proposition 2 (C.2), Proposition 3 (C.3), and Proposition 4 (C.4). In Appendix E, we describe further details on experiments and hyperparameters used including specific details for each experiment, including multi-path LiDAR navigation (E.3), modeling differentiating single-cells (E.4), and modeling cell-state perturbations (E.4). Finally, we provide the pseudo code for the multi-stage training algorithm in Appendix F.

Notation We denote the state space as $\mathcal{X} \subseteq \mathbb{R}^d$ and time interval as $t \in [0, 1]$. The branches are indexed with $k \in \{0, \dots, K\}$. We denote initial data distribution at time $t = 0$ as π_0 and the terminal data distributions at $t = 1$ as $\{\pi_{1,k}\}_{k=0}^K$. The joint data distribution is denoted $\pi_{0,1,k}$ and a pair of samples is given by $(\mathbf{x}_0, \mathbf{x}_{1,k}) \sim \pi_{0,1,k}$. For simplicity, we denote $d(\mathbf{x}_0, \mathbf{x}_{1,k}) = d\pi_{0,1,k}$. Let $u_{t,k}(X_t)$ denote the marginal velocity field, $g_{t,k}(X_t)$ denote the growth rate, and $p_{t,k}(X_t)$ denote the marginal probability density, where we sometimes drop the input X_t for simplicity. In addition, we denote the conditional velocity field and probability density as $u_{t|0,1,k} \equiv u_{t|0,1,k}(X_t|\mathbf{x}_0, \mathbf{x}_{1,k})$ and $p_{t|0,1,k} \equiv p_{t|0,1,k}(X_t|\mathbf{x}_0, \mathbf{x}_{1,k})$ respectively. The optimal values for any quantity are superscripted with a $\star$ symbol. We denote the parameterized flow neural networks as $u^\theta_{t,k}$ with parameters θ and the growth neural networks with $g^\phi_{t,k}$ with parameters ϕ. In the context of unbalanced endpoint distributions, we denote the true initial weight of a sample as $w^\star_0$ and the final weight of a sample from the kth target distribution as $w^\star_{1,k}$. The predicted weights generated from the growth dynamics are given by $w_t(X_t)$, and we seek to match the predicted weight at time $t = 1$ given by $w_{1,k}(X_{1,k})$ to the true weight $w^\star_{1,k}$. L^2 denotes the space of square integrable functions and $\|\cdot\|_{L^2}$ be the L^2-norm in function space. L^∞ denotes the space of essentially bounded functions such that $\|f\|_\infty < \infty$.

A EXTENDED THEORETICAL BACKGROUND

A.1 LEARNING OPTIMAL STOCHASTIC BRIDGES

Pinned-Down Stochastic Bridges Let $\mathbb{Q} \in \mathcal{M}$ be a Markovian reference path measure that evolves over $t \in [0, 1]$ according to a drift field $f_t(X_t) : \mathbb{R}^d \to \mathbb{R}^d$ and stochastic d-dimensional Brownian motion $B_t \in \mathbb{R}^d$, via the SDE

$$dX_t = f_t(X_t)dt + \sigma_t dB_t, \quad X_0 \sim \pi_0 \tag{20}$$

Given $\mathbb{Q}$, consider a stochastic process $(X_t)_{t\in[0,1]}$ over the time interval $t \in [0, 1]$ *pinned-down* at the initial point $X_0 = \mathbf{x}_0$ and final point $X_1 = \mathbf{x}_1$ denoted as $\mathbb{Q}_{\cdot|0,1}(\cdot|\mathbf{x}_0, \mathbf{x}_1)$. Due to the endpoint conditions, $\mathbb{Q}_{\cdot|0,1}$ is not necessarily Markov, and evolves via the SDE

$$dX_t = \{f_t(X_t) + \sigma_t^2 \nabla_{\mathbf{x}} \log \mathbb{Q}_{1|t}(\mathbf{x}_1|X_t)\}dt + \sigma_t dB_t, \quad X_0 = \mathbf{x}_0 \tag{21}$$

where $\nabla_{\mathbf{x}} \log \mathbb{Q}_{1|t}(\mathbf{x}_1|X_t)$ is a non-Markovian score function that *corrects* the drift field $f_t(X_t)$ of the reference process such that it points toward the target endpoint $\mathbf{x}_1$. Since $\log \mathbb{Q}_{1|t}(\mathbf{x}_1|X_t)$ is the log-likelihood that the final state satisfies the condition $X_1 = \mathbf{x}_1$, the gradient defines how the log-likelihood changes with respect to the changes in the state $\mathbf{x}$ at time t. The drift moves $\mathbf{x}$ in the direction of the largest increase in log-likelihood given by the score function, which ensures that the process satisfies $X_1 = \mathbf{x}_1$ following the theory of Doob's h-transform (Rogers & Williams, 2000). Now, we can define the conditional probability distribution p_t as a *mixture* of pinned-down stochastic bridges over pairs of endpoints in the data coupling $\pi_{0,1} = \pi_0 \otimes \pi_1$ given by

$$p_t(\mathbf{x}) = \pi_{0,1}\mathbb{Q}_{t|0,1}(\mathbf{x}|\mathbf{x}_0, \mathbf{x}_1) = \int \mathbb{Q}_{t|0,1}(\mathbf{x}|\mathbf{x}_0, \mathbf{x}_1)d\pi_{0,1} \tag{22}$$

To simplify notation, we denote each conditional bridge as $\mathbb{Q}_{t|0,1}(X_t|\mathbf{x}_0, \mathbf{x}_1) = p_{t|0,1}(X_t|\mathbf{x}_0, \mathbf{x}_1)$ and the joint distribution $\pi_{0,1} = p_{0,1}(\mathbf{x}_0, \mathbf{x}_1)$. Now, we can rewrite the marginal p_t as

$$p_t(X_t) = \int p_{t|0,1}(X_t|\mathbf{x}_0, \mathbf{x}_1)p_{0,1}d(\mathbf{x}_0, \mathbf{x}_1) = \mathbb{E}_{p_{0,1}}\left[p_{t|0,1}(X_t|\mathbf{x}_0, \mathbf{x}_1)\right] \tag{23}$$

Furthermore, we denote $u_{t|0,1} \equiv u_{t|0,1}(X_t|\mathbf{x}_0, \mathbf{x}_1) = \{f_t(X_t) + \sigma_t^2 \nabla_{\mathbf{x}} \log \mathbb{Q}_{1|t}(\mathbf{x}_1|X_t)\}$ as the conditional drift that *generates* $p_{t|0,1} \equiv p_{t|0,1}(X_t|\mathbf{x}_0, \mathbf{x}_1)$, that satisfies the conditional Fokker-Planck equation

$$\frac{\partial}{\partial t} p_{t|0,1} = -\nabla \cdot (u_{t|0,1} p_{t|0,1}) + \frac{1}{2}\sigma^2 \Delta p_{t|0,1} \tag{24}$$

Definition 1 (Reciprocal Class)**.** *Given our definition of the conditional bridge $\mathbb{Q}_{\cdot|0,1}$, we can define the reciprocal class, denoted $\mathcal{R}(\mathbb{Q})$, of the reference measure $\mathbb{Q}$ as the class of path measures that share the same bridge as $\mathbb{Q}$, defined as $\mathcal{R}(\mathbb{Q}) = \{\Pi \mid \Pi(X_t|\mathbf{x}_0, \mathbf{x}_1) = \mathbb{Q}(X_t|\mathbf{x}_0, \mathbf{x}_1)\}$.*

Markovian Projections Given a mixture of conditional stochastic bridges $\Pi = \Pi_{0,1}\mathbb{Q}_{\cdot|0,1}$ under the reference measure $\mathbb{Q}$ that require knowledge of the joint distribution $p_{0,1}$, we aim to project Π to the space of Markovian measures $\mathcal{M}$, where the drift dynamics $u_t(X_t)$ is only dependent on the current state X_t and require no knowledge on the endpoints. This allows us to parameterize a Markovian drift $u_t^\theta(X_t)$ that can transport samples from the initial distribution $\mathbf{x}_0 \sim \pi_0$ to samples from the target distribution $\mathbf{x}_1 \sim \pi_1$. To do this, we define the *Markovian projection* of Π (Shi et al., 2023; Liu et al., 2023b).

Definition 2 (Markovian Projection)**.** *Given a conditional bridge $\mathbb{Q}_{\cdot|0,1}$ that evolves via the SDE in (21), we define a Markovian projection of the mixture of bridges $\Pi = \Pi_{0,1}\mathbb{Q}_{\cdot|0,1}$ as a Markov process $\mathbb{M}^\star = proj_{\mathcal{M}}(\Pi) \in \mathcal{M}$ with the same marginals as Π such that $X_t \sim \Pi_t$ for all $t \in [0, 1]$, $X_1 \sim \pi_1$ and evolves via the SDE*

$$dX_t = \{f_t(X_t) + v_t^\star(X_t)\}dt + \sigma_t dB_t \tag{25}$$

$$v_t^\star(X_t) = \sigma^2 \mathbb{E}_{\Pi_{1|t}}\left[\nabla_{\mathbf{x}_t} \log \mathbb{Q}_{1|t}(X_1|X_t)|X_t = \mathbf{x}_t\right] \tag{26}$$

where $\Pi_{1|t}$ is the conditional distribution of X_1 under the mixture of bridges Π and $\nabla_{\mathbf{x}_t} \log \mathbb{Q}_{1|t}(X_1|X_t)$ points in the direction of greatest increase in the log-likelihood of the target endpoint $X_1 \sim \pi_1$ under the reference process $\mathbb{Q}$. In addition, the Markov projection $\mathbb{M}^\star$ minimizes the KL-divergence with the mixture of bridges $\mathbb{M}^\star = \arg\min_{\mathbb{M}\in\mathcal{M}} KL(\Pi\|\mathbb{M})$ and can be obtained by parameterizing $v_t^\theta(X_t)$ and minimizing the dynamic formulation given by

$$KL(\Pi\|\mathbb{M}) = \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_1)\sim\Pi_{0,1}} \mathbb{E}_{\mathbf{x}_t\sim\Pi_{t|0,1}} \int_0^1 \frac{1}{2\sigma_t^2}\left[\left\|\sigma_t^2 \nabla_{\mathbf{x}_t} \log \mathbb{Q}_{1|t}(\mathbf{x}_1|\mathbf{x}_t) - v_t^\theta(\mathbf{x}_t)\right\|^2\right] dt \tag{27}$$

In general, the Markovian projection of a reference measure $\mathbb{Q}$ does not preserve the conditional bridge and is not in the reciprocal class $\mathcal{R}(\mathbb{Q})$. The unique path measure $\mathbb{P}$ that is the Markovian projection of $\mathbb{Q}$, is in the reciprocal class $\mathcal{R}(\mathbb{Q})$, *and* preserves the endpoint distributions is called the *Schrödinger Bridge*.

Definition 3 (Schrödinger Bridge)**.** *Given a reference measure $\mathbb{Q}$, a initial distribution π_0, and final distribution π_1. A path measure $\mathbb{P}$ is the unique Schrödinger bridge if it satisfies*

1. *$\mathbb{P}$ is the Markovian projection of $\mathbb{Q}$, such that $\mathbb{P} = proj_{\mathcal{M}}(\mathbb{Q})$.*
2. *$\mathbb{P}$ is in the reciprocal class of $\mathbb{Q}$, i.e. $\mathbb{P} \in \mathcal{R}(\mathbb{Q})$, such that it preserves the conditional bridge $\mathbb{P}(X_t|\mathbf{x}_0, \mathbf{x}_1) = \mathbb{Q}(X_t|\mathbf{x}_0, \mathbf{x}_1)$.*
3. *$\mathbb{P}$ preserves the endpoint distributions $\mathbb{P}_0 = \pi_0$ and $\mathbb{P}_1 = \pi_1$.*

The goal of Schrödinger Bridge Matching (SBM) is to estimate the Schrödinger Bridge that transports samples from an initial distribution π_0 to a final distribution π_1 given the optimal reference dynamics. We further discuss previous approaches to solving the SB problem in Appendix B.

A.2 Simulating Trajectories on the Data Manifold

Riemannian Manifolds and Metrics Since the interpolant $\mathbf{x}_{t,\eta}$ learned in Stage 1 is defined by minimizing a non-linear state cost $V_t(X_t)$, the resulting velocity field $u_t^\theta(X_t)$ lies on the tangent bundle $\mathcal{T}_{\mathbf{x}}\Omega$ of a smooth d-dimensional manifold $\Omega \in \mathbb{R}^d$ called a *Riemannian manifold*. Intuitively, a Riemannian manifold can be thought of as a smooth surface where the local curvature around a point $\mathbf{x} \in \Omega$ can be approximated by a tangent space $\mathcal{T}_{\mathbf{x}}\Omega$ that defines the set of directions in which

$\mathbf{x}$ can move along the manifold. These *directions* are defined by a set of tangent vectors $u \in \mathcal{T}_{\mathbf{x}}\Omega$, which *pushes* the point $\mathbf{x}$ along the manifold.

In Stage 2, we seek to parameterize a vector field $u_t^\theta(X_t)$ that minimizes the *angle* from the tangent vector $\dot{\mathbf{x}}_{t,\eta} = \partial_t \mathbf{x}_{t,\eta}$ at each point $\mathbf{x}$ on the manifold optimized in Stage 1. To do this, we must define the concepts of *length* and *angles* on Riemannian manifolds. First, we define a *location-dependent inner product* in Riemannian manifolds known as the *Riemannian metric* $g_{\mathbf{x}} : \mathcal{T}_{\mathbf{x}}\Omega \times \mathcal{T}_{\mathbf{x}}\Omega \to \mathbb{R}$ that maps two vectors $u, v \in \mathcal{T}_{\mathbf{x}}\Omega$ to a scalar that describes the relative direction and length of the two vectors. Formally, the Riemannian metric can be written as the *billinear* and *positive definite* function

$$g_{\mathbf{x}}(u, v) = u^\top \mathbf{G}(\mathbf{x}) v = \langle u, \mathbf{G}v \rangle \quad \text{s.t.} \quad \begin{cases} \forall u \neq 0 \;\; g_{\mathbf{x}}(u, u) > 0 \\ \mathbf{G} \succ 0 \end{cases} \tag{28}$$

which defines the norm of a tangent vector as $\|u\|_{g_{\mathbf{x}}} = \sqrt{g_{\mathbf{x}}(u, u)}$. Now, we can decompose the Riemannian norm of the tangent vector $\|\dot{\mathbf{x}}_{t,\eta}\|_{g_{\mathbf{x}}}$ to get the loss defined in (12) as follows

$$\mathcal{L}_{\text{traj}}(\eta) = \mathbb{E}_{t,(\mathbf{x}_0,\mathbf{x}_1)\sim p_{0,1}} \left[\|\dot{\mathbf{x}}_{t,\eta}\|_{g_{\mathbf{x}}}^2 \right] \tag{29}$$

$$= \mathbb{E}_{t,(\mathbf{x}_0,\mathbf{x}_1)\sim p_{0,1}} \langle \dot{\mathbf{x}}_{t,\eta}, \mathbf{G}(\mathbf{x}_{t,\eta}) \dot{\mathbf{x}}_{t,\eta} \rangle \tag{30}$$

$$= \mathbb{E}_{t,(\mathbf{x}_0,\mathbf{x}_1)\sim p_{0,1}} \left[\|\dot{\mathbf{x}}_{t,\eta}\|_2^2 + \langle \dot{\mathbf{x}}_{t,\eta}, (\mathbf{G}(\mathbf{x}_{t,\eta}) - \mathbf{I}) \dot{\mathbf{x}}_{t,\eta} \rangle \right] \tag{31}$$

$$= \mathbb{E}_{t,(\mathbf{x}_0,\mathbf{x}_1)\sim p_{0,1}} \left[\|\dot{\mathbf{x}}_{t,\eta}\|_2^2 + V_t(\mathbf{x}_{t,\eta}) \right] \tag{32}$$

where the state cost is defined as $V_t(\mathbf{x}_{t,\eta}) = \langle \dot{\mathbf{x}}_{t,\eta}, (\mathbf{G}(\mathbf{x}_{t,\eta}) - \mathbf{I}) \dot{\mathbf{x}}_{t,\eta} \rangle$.

Data-Dependent State Cost Following Kapuśniak et al. (2024), we define the metric matrix $\mathbf{G}(\mathbf{x}_{t,\eta})$ described previously as the data-dependent LAND and RBF metrics of the form $\mathbf{G}_{\text{LAND}}(\mathbf{x}, \mathcal{D}) = \mathbf{G}_{\text{RBF}}(\mathbf{x}, \mathcal{D}) = (\text{diag}(\mathbf{h}(\mathbf{x})) - \varepsilon \mathbf{I})^{-1}$ which assigns higher cost (i.e. $\|\mathbf{G}(\mathbf{x})\|$ is larger) when $\mathbf{x}$ moves away from the support of the dataset $\mathcal{D}$. Specifically, given a dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$, we define the elements $\mathbf{h}^{\text{LAND}}(\mathbf{x}) \in \mathbb{R}^d$ that scales down each dimension $j \in \{1, \ldots, d\}$ in the LAND metric as

$$h_j^{\text{LAND}}(\mathbf{x}) = \sum_{i=1}^{N} (x_i^j - x^j)^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2\sigma^2} \right) \tag{33}$$

where the exp term is positive when there is a high concentration of data points around the point $\mathbf{x}$ (i.e. $\|\mathbf{x} - \mathbf{x}_i\|$ is small) and approaches 0 as the concentration of data around $\mathbf{x}$ decreases (i.e. $\|\mathbf{x} - \mathbf{x}_i\|$ is large). Writing $\langle \dot{\mathbf{x}}_{t,\eta}, \mathbf{G}(\mathbf{x}_{t,\eta}) \dot{\mathbf{x}}_{t,\eta} \rangle$ in terms of $h_j(\mathbf{x})$, we get

$$\mathcal{L}_{\text{traj}}(\eta) = \mathbb{E}_{t,(\mathbf{x}_0,\mathbf{x}_1)\sim p_{0,1}} \langle \dot{\mathbf{x}}_{t,\eta}, \mathbf{G}(\mathbf{x}_{t,\eta}) \dot{\mathbf{x}}_{t,\eta} \rangle \tag{34}$$

$$= \mathbb{E}_{t,(\mathbf{x}_0,\mathbf{x}_1)\sim p_{0,1}} \left[\sum_{j=1}^{d} \frac{(\dot{\mathbf{x}}_{t,\eta})_j^2}{h_j(\mathbf{x}_{t,\eta}) + \varepsilon} \right] \tag{35}$$

When $h_j(\mathbf{x})$ is large, the loss is minimized, and when $h_j(\mathbf{x})$ is small, the loss is scaled up. While the LAND metric effectively defines the data manifold in low dimensions, in high-dimensional state spaces, setting a suitable variance σ in $h_j^{\text{LAND}}(\mathbf{x}_t)$ to ensure that the path does not deviate far from the data manifold without overfitting is challenging. To overcome this limitation, the RBF metric clusters the dataset into N_c clusters with centroids denoted as $\hat{\mathbf{x}}_n \in \mathbb{R}^d$ and trains a set of parameters $\{\omega_{\alpha,n}\}_{n=1}^{N_c}$ to enforce $h_j(\mathbf{x}_i) \approx 1$ for all points in the dataset such that points $\mathbf{x}$ within the data manifold are also assigned $h_j(\mathbf{x}) \approx 1$. Specifically, h_j^{RBF} is defined as

$$h_j^{\text{RBF}}(\mathbf{x}) = \sum_{n=1}^{N_c} \omega_{n,j}(\mathbf{x}) \exp\left(-\frac{\lambda_{n,j}}{2} \|\mathbf{x} - \hat{\mathbf{x}}_n\|^2 \right) \tag{36}$$

$$\lambda_n = \frac{1}{2} \left(\frac{\kappa}{|C_n|} \sum_{\mathbf{x} \in C_n} \|\mathbf{x} - \hat{\mathbf{x}}_n\|^2 \right)^{-2} \tag{37}$$

where C_n denotes the nth cluster, κ is a tunable hyperparameter, and $\lambda_{n,j}$ is the bandwidth of cluster n the jth dimension. To train the parameters, we minimize the following loss

$$\mathcal{L}_{\text{RBF}}(\{\omega_{\alpha,n}\}) = \sum_{\mathbf{x}_i \in \mathcal{D}} \left(1 - h_n^{\text{RBF}}(\mathbf{x}_i) \right)^2 \tag{38}$$

In our experiments, we use the LAND metric for the LiDAR and mouse hematopoiesis datasets with dimensions $d = 2$ and $d = 3$ respectively, and the RBF metric for the perturbation modeling experiment with gene expression data of dimensions $d \in \{50, 100, 150\}$.

Sampling on the Data Manifold In Riemannian geometry, each Euclidean step $\Delta t \cdot u_t^\theta(\mathbf{x})$ following the tangent vector $u_t^\theta(\mathbf{x})$ along the manifold requires mapping back to the manifold via an exponential map $X_{t+\Delta t} = \exp_{\mathbf{x}}(\Delta t \cdot u_t^\theta(\mathbf{x}))$. In general, computing the exponential map $\exp_{\mathbf{x}}$ under the Riemannian metric $\mathbf{G}(\mathbf{x})$ requires simulation of the geodesic flow (i.e., approximating the final state at $t = 1$ under the initial conditions $\mathbf{x}_0 = \mathbf{x}$ and $\dot{\mathbf{x}}_0 = u_t^\theta(\mathbf{x})$).

While the manifold defined by the data-dependent metric $\mathbf{G}(\mathbf{x}, \mathcal{D})$ induces a Riemannian geometry in Euclidean space that follows an optimal cost structure, its underlying space is still Euclidean. $\mathbf{G}(\mathbf{x}, \mathcal{D})$ just assigns varying costs of moving in the Euclidean space $\mathbb{R}^d$. Therefore, we can avoid computing the exponential map and generate trajectories with simple Euclidean Euler integration following

$$\mathbf{x}_{t+\Delta t} = \mathbf{x}_t + \Delta t \cdot u_t^\theta(\mathbf{x}_t) \tag{39}$$

where $\Delta t = 1/N_{\text{steps}}$ is the discretized step size.

B Comparison to Existing Works

In this section, we discuss the relationship between our proposed formulation for Branched Schrödinger Bridge Matching and related previous works. We establish reasons why BranchSBM is the theoretically optimal formulation to solve the problem of modeling stochastic dynamical systems with diverging trajectories over time by modeling branched Schrödinger bridges. We conclude by introducing an alternative perspective of the Branched GSB problem defined in Section 3.2 as the problem of modeling probabilistic trajectories of dynamic systems with nondeterministic states, which BranchSBM is well-suited to solve.

B.1 Modeling Branched Schrödinger Bridges

Schrödinger Bridge Matching Computational methods for solving the Schrödinger Bridge (SB) problem for predicting trajectories between initial and target distributions have been extensively studied in existing literature (De Bortoli et al., 2021a; Chen et al., 2021b; Korotin et al., 2023; Bunne et al., 2022a; Chizat et al., 2018; Liu et al., 2023a; 2022; Shi et al., 2023; Kim et al., 2024; Wang et al., 2021; Tong et al., 2024b; Peluchetti, 2024; Bunne et al., 2022a; Somnath et al., 2023; Gushchin et al., 2024; Pavon et al., 2021; Garg et al., 2024; De Bortoli et al., 2024; Shen et al., 2024; Chen et al., 2023; Noble et al., 2023). Previous work has framed the SB problem as an entropy-regularized Optimal Control (EOT) problem (Cuturi, 2013; Léonard, 2014; Pavon et al., 2018) or a stochastic optimal control (SOC) problem (Chen et al., 2016; 2021a; Liu et al., 2023a), which we build on in this work.

Conditional Stochastic Optimal Control Several works (Chen et al., 2016; 2021a; Liu et al., 2023a) have reframed the SB problem as a Conditional Stochastic Optimal Control (CondSOC) problem, which takes the canonical form

$$\min_{u_t} \int_0^1 \mathbb{E}_{p_t} \left[\frac{1}{2} \|u_t(X_t)\|^2 + V_t(X_t) \right] dt + \mathbb{E}_{p_1} \left[\phi(X_1) \right] \tag{40}$$

$$\text{s.t. } dX_t = u_t(X_t)dt + dB_t, \quad X_0 \sim \pi_0 \tag{41}$$

where $\phi(X_1) : \mathcal{X} \to \mathbb{R}$ acts as a reconstruction loss that enforces that the distribution of $X_1 \sim p_1$ matches the true distribution π_1. Due to the intractability of π_0, π_1, GSBM (Liu et al., 2023a) uses spline optimization to learn an optimal Gaussian probability path $p^\star_{t|0,1} = \mathcal{N}(\mu_t, \gamma_t^2 \mathbf{I}_d)$ using only samples $\mathbf{x}_0 \sim \pi_0$ and $\mathbf{x}_1 \sim \pi_1$ from the initial and terminal distributions. Although GSBM does not require knowledge of the densities, it follows an iterative optimization scheme that alternates between matching the drift u_t given a fixed marginal p_t, and updating the marginal given the drift. This strategy can get stuck in suboptimal solutions and is sensitive to the initialization of u_t^θ. GSBM is also limited to learning unimodal Gaussian paths between one source and one target distribution with balanced mass, making it unsuitable for modeling tasks with multimodal terminal distributions and splitting of mass over multiple distinct paths.

Regularized Unbalanced Optimal Transport Several previous works have studied the unbalanced optimal transport problem (Zhang et al., 2025c; Chen et al., 2021b; Lübeck et al., 2022; Pariset et al., 2023); however, these approaches address a fundamentally different setting from the unbalanced Generalized Schrödinger Bridge (GSB) problem considered in this work. Specifically, DeepRUOT (Zhang et al., 2025c) solves the Regularized Unbalanced Optimal Transport (RUOT) problem by parameterizing the canonical stochastic bridge SDE $dX_t = f_t(X_t)dt + \sigma_t dB_t$ with a probability flow ODE given by

$$dX_t = \underbrace{\left\{ f_t(X_t) + \frac{1}{2}\sigma_t^2 \nabla_{\mathbf{x}} \log p_t^\theta(X_t) \right\}}_{u_t^\theta(X_t)} dt \tag{42}$$

where $u_t^\theta(X_t)$ is the drift of the probability flow ODE that is learned along with the probability density $p_t^\theta(X_t)$ to derive the drift of the SDE $f_t(X_t)$. Unlike GSBM (Liu et al., 2023a), which enforces a hard constraint on the endpoints, DeepRUOT models a probability density flow by minimizing a reconstruction loss that encourages alignment with both intermediate and terminal distributions. While effective when intermediate distributions are observed, the method fails to learn meaningful trajectories in settings where these intermediate snapshots are unavailable, limiting its applicability in many real-world scenarios.

Learning Diverging Trajectories with Single Target SBM To model diverging trajectories with single-branch SBM where the target distribution is multi-modal, we can follow a set of N_{samples} samples from the initial distribution π_0 to determine the distribution of samples that end at each of the modes of the target distribution π_1. While this can estimate the splitting of mass across different trajectories, it does not explicitly learn the optimal distribution of mass in the latent space over time.

Furthermore, if the path toward a specific cluster in the target distribution has lower potential than that of the other clusters, mode collapse could occur, where all samples follow the same trajectory without reaching the other clusters. BranchSBM learns to generate the correct mass distribution over each of the target states by optimizing the growth term with respect to the matching loss $\mathcal{L}_{\text{match}}$ defined in Equation 15.

With the standard SBM formulation, it is also challenging to determine the time at which branching occurs and the population diverges toward different targets, as all samples follow stochastic trajectories. With BranchSBM, we model population-wide branching dynamics with growth networks that are trained to predict the origin of a branch from having zero mass ($w_{t,k} = 0$) and the rate at which it grows/shrinks over time $\partial_t w_{t,k} = g_{t,k}(X_{t,k})$. The mass of a branch at any given time step can be simulated with $w_{t,k}(X_{t,k}) = w_{0,k} + \int_0^t g_{s,k}^\phi(X_{s,k})ds$.

Branching Dynamics While branched dynamics have been explored in the context of optimal transport (Lippmann et al., 2022) and Brownian motion (Baradat & Lavenant, 2021), no previous work has explicitly formulated or solved the branched Schrödinger bridge problem that seeks to match an initial distribution to a multiple terminal distributions via stochastic bridges. For instance, the concept of Branching Brownian Motion (BBM) introduced in Baradat & Lavenant (2021) models a population of particles that each independently follow stochastic trajectories according to standard Brownian motion with positive diffusivity $\nu > 0$. To model branching dynamics, each particle has an independent branching rate $\lambda > 0$ that determines the probability that the particle undergoes a *branching event*, defined as the particle dying and generating k new particles that then evolve independently. The number of generated particles is sampled from a probability distribution over non-negative integers $k \sim p = (p_k)_{k\in\mathbb{N}}$, where p_k is the probability of generating k particles at the branching event and $\sum_{k\in\mathbb{N}} p_k = 1$. Given p_k, $q_k = \lambda p_k$ defines the rate of branching events that generate k new particles and define a new probability measure $q = (q_k)_{k\in\mathbb{N}}$ called the *branching mechanism*.

While BBM defines branching as the generation of additional particles from a single particle following an independent, temporal probability measure q, this model fails to model the *division* of mass across multiple trajectories, where total mass remains constant but the mass of each branch changes. In addition, BBM assumes that each branched particle undergoes independent Brownian motion, without explicitly defining a terminal state or branch-specific drift. For these reasons, the BBM model is unsuitable for modeling branching in the majority of real-world contexts, such as cell state transitions,

where undifferentiated cells split probabilistically into distinct fates rather than proliferating in number. In such systems, branching reflects a redistribution of probability mass over developmental trajectories, governed by underlying regulatory patterns rather than purely stochastic reproduction.

Where existing frameworks fall short is in modeling **meaningful energy-aware, conditional stochastic trajectories** with **unbalanced and multi-modal dynamics**, which we address in this work. Specifically, we formulate the Unbalanced CondSOC problem followed by the Branched CondSOC problem that defines a *set* of optimal drifts and growth fields that define a set of branched trajectories following optimal energy-minimizing trajectories defined by the state cost $V_t(X_t)$. Instead of spline optimization, we leverage a parameterized network $\varphi^\star_{t,\eta}(\mathbf{x}_0, \mathbf{x}_1)$ that learns to predict an optimal interpolating path given a pair of endpoints.

To model dynamic growth of mass along branched trajectories, we initialize a normalized population weight of 1 at a *primary branch* ($k = 0$), that can split across K branches and generate weights $\{w_{t,k}\}_{k=0}^K$ that evolve co-currently via learned growth rates $\{g^\phi_{t,k}\}_{k=0}^K$. This approach relies on learning the growth rate over an entire population of samples across all branches rather than learning independent growth rates of individual samples, which enforces stronger constraints during training to ensure that the model captures true population growth dynamics.

B.2 Modeling Perturbation Responses

In single-cell transcriptomics, perturbations such as gene knockouts, transcription factor induction, or drug treatments frequently induce cell state transitions that diverge into distinct fates, reflecting differentiation, resistance, and off-target effects (Shalem et al., 2014; Kramme et al., 2021; Dixit et al., 2016; Gavriilidis et al., 2024; Zhang et al., 2025a; Kobayashi et al., 2022; Smela et al., 2023; Pierson Smela et al., 2025; Yeo et al., 2021). Trajectory inference methods for modeling cell-state dynamics with single-cell RNA sequencing data, including flow matching (Zhang et al., 2025b; Rohbeck et al., 2025; Atanackovic et al., 2024; Tong et al., 2024b; Wang et al., 2025), Schrödinger Bridge Matching (Alatkar & Wang, 2025; Tong et al., 2024b), and optimal transport (Zhang et al., 2025c; Tong et al., 2020; Bunne et al., 2023; Driessen et al., 2025; Huguet et al., 2022; Klein et al., 2024; Yachimura et al., 2024; Schiebinger et al., 2019; Zhang et al., 2021; Bunne et al., 2022b), have been widely explored. While many algorithms have been developed to predict perturbation responses (Bunne et al., 2023; Megas et al., 2025; Rohbeck et al., 2025; Roohani et al., 2023; Ryu et al., 2024), they either predict only the terminal perturbed state without modeling the intermediate cell-state transitions or model single unimodal perturbation trajectories. Schrödinger Bridge Matching frameworks like [SF]^{2}M (Tong et al., 2024b) and DeepGSB (Liu et al., 2022) have been shown to effectively model stochastic transitions in biological systems, offering better scalability and sampling efficiency than classical OT.

However, these models are still limited to modeling trajectories between a single pair of boundary distributions, limiting their ability to represent divergent trajectories arising from the same perturbed initial state. This is a key limitation when modeling processes like fate bifurcation post-perturbation, where a cell population exposed to the same stimulus may split into multiple phenotypically distinct outcomes. BranchSBM extends the SBM framework to support multiple terminal marginals, enabling modeling of stochastic bifurcations in a mathematically principled way. By learning a mixture of conditional stochastic processes from a common source to multiple target distributions, BranchSBM can capture the heterogeneity and uncertainty of cell fate decisions under perturbation. Moreover, it retains the empirical tractability of previous SB-based models, requiring only samples from distributions, and ensures that intermediate trajectories lie on the manifold of feasible cell states.

B.3 Modeling Probabilistic Trajectories with BranchSBM

We conclude the discussion with an alternative interpretation of the Branched Schrödinger Bridge problem that deviates from the branching population dynamics problem. We instead consider Branched SB as a probabilistic trajectory matching problem, where each branch is one of multiple possible trajectories that a sample X_t could follow. Since single-path SBM learns only a single deterministic drift field $u_t(X_t)$ that determines the direction and flow of the SDE $dX_t = u^\theta_t(X_t)dt + \sigma dB_t$, the probabilistic aspect of the trajectory remains restricted to Brownian motion via σdB_t. This fails to capture probabilistic dynamics where probability densities begin concentrated at a single state but evolve into multi-modal probability densities $\{p_{t,k}(X_t)\}_{k=0}^K$ over multiple different states, each

of which evolve according to an SDE $dX_t = u_{t,k}(X_t)dt + \sigma_k dB_t$ with an independent drift term $u_{t,k}(X_t)$ and noise scaling σ_k.

Where other SBM frameworks fall short, BranchSBM is capable of modeling multiple probabilistic trajectories, where a system begins at a single deterministic state $X_0 = \mathbf{x}_0$ with probability $w_{1,0} = 1$ and evolves via multiple probabilistic trajectories that diverge in the state space governed by the SDEs $dX_t = u_{t,k}(X_t)dt + \sigma_k dB_t$. At time t, the system exists in a non-deterministic superposition of states $\{X_{t,k}, w_{t,k}\}_{k=0}^K$, each with a probability $w_{t,k}$ such that $\sum_{k=0}^K w_{t,k}^\phi = 1$. In addition, BranchSBM can model the evolution of the probability weights $w_{t,k}^\phi$ by parameterizing the *probabilistic growth rates* $\{g_{t,k}^\phi\}_{k=0}^K$ that preserve conservation of probability mass $\sum_{k=0}^K w_{t,k}^\phi = 1$ at all times $t \in [0, 1]$ by minimizing the mass loss $\mathcal{L}_{\text{mass}}(\phi)$ (16). This problem is prevalent in many biological and physical systems, where a system does not exist in a single deterministic state, but rather a *superposition* over a distribution of states.

Table 6: **BranchSBM enables robust modeling of branched Schroödinger bridges compared to existing frameworks.** BranchSBM can model both branching and unbalanced trajectories, follows intermediate trajectories governed by a task-specific state cost, requires only endpoint samples for training, and samples trajectories from only a single sample from the starting distribution.

Method	Models Branching	Models Unbalanced	Intermediate Dynamics	Requirements for Training	Requirements for Sampling
Generalized SBM Liu et al. (2023a)	No	No	Entropic OT with learned drift u_t^θ	Samples from π_0, π_1	Endpoints $(\mathbf{x}_0, \mathbf{x}_1) \sim p_{0,1}^\theta$
DeepRUOT (Zhang et al., 2025c)	Requires simulating multiple samples	Growth rate $g_t^\phi(X_t)$	Regularized OT with learned drift u_t^θ and density p_t^θ	Samples from π_0, π_1	Sample $\mathbf{x}_0 \sim \pi_0$
BranchSBM (Ours)	Simulates divergent trajectories and terminal states from single sample $\mathbf{x}_0$	Branched growth rates $\{g_{t,k}^\phi(X_t)\}_{k=0}^K$	Branched drifts $\{u_{t,k}^\theta\}_{k=0}^K$ that minimize state-cost V_t	Samples from π_0 and clusters $\{\pi_{1,k}\}_{k=0}^K$	Sample $\mathbf{x}_0 \sim \pi_0$

C Theoretical Proofs

C.1 Proof of Proposition 1

Proposition 1 (Unbalanced Conditional Stochastic Optimal Control) *Suppose the marginal density can be decomposed as $p_t(X_t) = \int_{\pi_{0,1}} p_t(X_t|\mathbf{x}_0, \mathbf{x}_1) p_{0,1}(\mathbf{x}_0, \mathbf{x}_1) d\pi_{0,1}$, where $\pi_{0,1}$ is a fixed joint coupling of the data. Then, we can identify the optimal drift $u_t^\star$ and growth $g_t^\star$ that solves the Unbalanced GSB problem in (4) by minimizing the Unbalanced Conditional Stochastic Optimal Control objective given by*

$$\min_{u_t, g_t} \mathbb{E}_{(\mathbf{x}_0, \mathbf{x}_1) \sim \pi_{0,1}} \left[\int_0^1 \mathbb{E}_{p_{t|0,1}} \left[\frac{1}{2} \|u_t(X_t|\mathbf{x}_0, \mathbf{x}_1)\|^2 + V_t(X_t) \right] w_t(X_t) dt \right] \tag{43}$$

$$\text{s.t. } dX_t = u_t(X_t|\mathbf{x}_0, \mathbf{x}_1)dt + \sigma dB_t, \quad X_0 = \mathbf{x}_0, \quad X_1 = \mathbf{x}_1 \quad w_0(X_0) = w_0^\star, \quad w_1(X_1) = w_1^\star \tag{44}$$

where $w_t = w_0 + \int_0^t g_s(X_s)ds$ is the time-dependent weight initialized at $w_0^\star$, u_t is the drift, g_t is the growth rate, and $\pi_{0,1}$ is the weighted coupling of paired endpoints $(\mathbf{x}_0, w_0, \mathbf{x}_1, w_1) \sim \pi_{0,1}$.

Proof. We define the Unbalanced Generalized Schrödinger Bridge problem as the solution (u_t, p_t, g_t) to the energy minimization problem such that the unbalanced Fokker-Planck equation is satisfied.

$$\min_{u_t, g_t} \int_0^1 \mathbb{E}_{p_t} \left[\frac{1}{2} \|u_t(X_t)\|^2 + V_t(X_t) \right] \left(w_0 + \int_0^t g_s(X_s)ds \right) dt \tag{45}$$

$$\text{s.t. } \begin{cases} \frac{\partial}{\partial t} p_t(X_t) = -\nabla \cdot (u_t(X_t) p_t(X_t)) + \frac{\sigma^2}{2} \Delta p_t(X_t) + g_t(X_t) p_t(X_t) \\ p_0 = \pi_0, \; p_1 = \pi_1 \end{cases} \tag{46}$$

Under the assumption that the joint probability $p_{0,1}(\mathbf{x}_0, \mathbf{x}_1)$ is fixed over all times $t \in [0, 1]$ and the marginal probability can be factorized as $p_t(X_t) = \mathbb{E}_{p_{0,1}}[p_t(X_t|\mathbf{x}_0, \mathbf{x}_1)]$, we can decompose the minimization objective into

$$\begin{aligned}
&\int_0^1 \mathbb{E}_{p_t}\left[\frac{1}{2}\|u_t(X_t)\|^2 + V_t(X_t)\right]\left(w_0 + \int_0^t g_s(X_s)ds\right)dt \qquad (47)\\
&= \int_0^1 \mathbb{E}_{p_{0,1}}\mathbb{E}_{p_{t|0,1}}\left[\frac{1}{2}\|u_t(X_t)\|^2 + V_t(X_t)\right]\left(w_0 + \int_0^t g_s(X_s)ds\right)dt \quad \text{(law of total expectation)}\\
&= \mathbb{E}_{p_{0,1}}\int_0^1 \mathbb{E}_{p_{t|0,1}}\left[\frac{1}{2}\|u_t(X_t)\|^2 + V_t(X_t)\right]\left(w_0 + \int_0^t g_s(X_s)ds\right)dt \quad \text{(Fubini's theorem)}
\end{aligned}$$

which can be solved by identifying the conditional drift u_t that minimizes the expected objective value over all endpoint samples $(\mathbf{x}_0, \mathbf{x}_1) \sim p_{0,1}$. Under similar assumptions, we can decompose all terms in the Fokker-Planck equation. For the left-hand side, we have:

$$\frac{\partial}{\partial t}p_t(X_t) = \frac{\partial}{\partial t}\int_{\pi_{0,1}} p_t(X_t|\mathbf{x}_0, \mathbf{x}_1)p_{0,1}(\mathbf{x}_0, \mathbf{x}_1)d\pi_{0,1} = \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_1)\sim\pi_{0,1}}\left[\frac{\partial}{\partial t}p_t(X_t|\mathbf{x}_0, \mathbf{x}_1)\right] \qquad (48)$$

For the divergence term, we have:

$$\begin{aligned}
\nabla \cdot (u_t(X_t)p_t(X_t)) &= \nabla \cdot \left(u_t(X_t)\int_{\pi_{0,1}} p_t(X_t|\mathbf{x}_0, \mathbf{x}_1)p_{0,1}(\mathbf{x}_0, \mathbf{x}_1)d\pi_{0,1}\right)\\
&= \nabla \cdot \left(\int_{\pi_{0,1}} u_t(X_t)p_t(X_t|\mathbf{x}_0, \mathbf{x}_1)p_{0,1}(\mathbf{x}_0, \mathbf{x}_1)d\pi_{0,1}\right) \qquad (49)
\end{aligned}$$

By Fubini's Theorem and the linearity of the divergence operator, we can switch the order of integration to get

$$\begin{aligned}
\nabla \cdot (u_t(X_t)p_t(X_t)) &= \int_{\pi_{0,1}} \nabla \cdot \Big(u_t(X_t)p_t(X_t|\mathbf{x}_0, \mathbf{x}_1)\Big)p_{0,1}(\mathbf{x}_0, \mathbf{x}_1)d\pi_{0,1}\\
&= \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_1)\sim\pi_{0,1}}\Big[\nabla \cdot \big(u_t(X_t)p_t(X_t|\mathbf{x}_0, \mathbf{x}_1)\big)\Big] \qquad (50)
\end{aligned}$$

For the Laplacian term, we have:

$$\begin{aligned}
\frac{\sigma^2}{2}\Delta p_t(X_t) &= \frac{\sigma^2}{2}\nabla \cdot (\nabla p_t(X_t))\\
&= \frac{\sigma^2}{2}\nabla \cdot \left(\nabla \int_{\pi_{0,1}} p_t(X_t|\mathbf{x}_0, \mathbf{x}_1)p_{0,1}(\mathbf{x}_0, \mathbf{x}_1)d\pi_{0,1}\right)\\
&= \frac{\sigma^2}{2}\int_{\pi_{0,1}} \Big(\nabla \cdot \nabla p_t(X_t|\mathbf{x}_0, \mathbf{x}_1)\Big)p_{0,1}(\mathbf{x}_0, \mathbf{x}_1)d\pi_{0,1}\\
&= \frac{\sigma^2}{2}\mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_1)\sim\pi_{0,1}}\left[\Delta p_{t|0,1}(X_t|\mathbf{x}_0, \mathbf{x}_1)\right] \qquad (51)
\end{aligned}$$

For the growth term, we have:

$$\begin{aligned}
g_t(X_t)p_t(X_t) &= g_t(X_t)\int_{\pi_{0,1}} p_t(X_t|\mathbf{x}_0, \mathbf{x}_1)p_{0,1}(\mathbf{x}_0, \mathbf{x}_1)d\pi_{0,1}\\
&= \int_{\pi_{0,1}} g_t(X_t)p_t(X_t|\mathbf{x}_0, \mathbf{x}_1)p_{0,1}(\mathbf{x}_0, \mathbf{x}_1)d\pi_{0,1}\\
&= \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_1)\sim\pi_{0,1}}\Big[g_t(X_t)p_t(X_t|\mathbf{x}_0, \mathbf{x}_1)\Big] \qquad (52)
\end{aligned}$$

Combining all the terms of the Fokker-Planck equation, we have shown that (46) can be rewritten as

$$\frac{\partial}{\partial t}p_t(X_t|\mathbf{x}_0, \mathbf{x}_1) = -\nabla \cdot \big(u_t p_t(X_t|\mathbf{x}_0, \mathbf{x}_1)\big) + \Delta p_{t|0,1}(X_t|\mathbf{x}_0, \mathbf{x}_1) + g_t p_t(X_t|\mathbf{x}_0, \mathbf{x}_1) \qquad (53)$$

Therefore, the Unbalanced GSB problem is equivalent to solving the Unbalanced CondSOC problem, and we conclude our proof of Proposition 1.

C.2 Proof of Proposition 2

Proposition 2 (Branched Conditional Stochastic Optimal Control) *For each branch, let $p_{t,k}(X_{t,k}) = \mathbb{E}_{p_{0,1k}}[p_{t,k}(X_{t,k}|\mathbf{x}_0, \mathbf{x}_{1,k})]$, where $\pi_{0,1,k}$ is the joint coupling distribution of samples $\mathbf{x}_0 \sim \pi_0$ from the initial distribution and $\mathbf{x}_{1,k} \sim \pi_{1,k}$ from the kth target distribution. Then, we can identify the set of optimal drift and growth functions $\{u^\star_{t,k}, g^\star_{t,k}\}_{k=0}^K$ that solve the Branched GSB problem in (3.2) by minimizing the sum of Unbalanced CondSOC objectives given by*

$$\min_{\{u_{t,k},g_{t,k}\}_{k=0}^K} \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_{1,0})\sim\pi_{0,1,0}} \int_0^1 \left\{ \mathbb{E}_{p_{t|0,1,0}} \left[\frac{1}{2}\|u_{t,0}(X_{t,0})\|^2 + V_t(X_{t,0}) \right] w_{t,0} \right.$$
$$\left. + \sum_{k=1}^K \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_{1,k})\sim\pi_{0,1,k}} \int_0^1 \mathbb{E}_{p_{t|0,1,k}} \left[\frac{1}{2}\|u_{t,k}(X_{t,k})\|^2 + V_t(X_{t,k}) \right] w_{t,k} \right\} dt \tag{54}$$

$$\text{s.t. } dX_{t,k} = u_{t,k}(X_{t,k})dt + \sigma dB_t,\ X_0 = \mathbf{x}_0,\ X_{1,k} = \mathbf{x}_{1,k},\ w_{0,k} = \delta_{k=0},\ w_{1,k} = w_{1,k} \tag{55}$$

where $w_{t,0} = 1 + \int_0^t g_{s,0}(\mathbf{x}_{s,0})ds$ is the weight of the primary paths initialized at 1 and $w_{t,k} = \int_0^t g_{s,k}(\mathbf{x}_{s,k})ds$ are the weights of the K secondary branches initialized at 0.

Proof. We extend the proof of Proposition 1 to the branching case by defining each branch k as an independent Unbalanced Generalized Schrödinger Bridge problem in (4) given by

$$\min_{u_{t,k},g_{t,k}} \int_0^1 \mathbb{E}_{p_{t,k}} \left[\frac{1}{2}\|u_{t,k}(X_{t,k})\|^2 + V_t(X_{t,k}) \right] \left(w_{0,k} + \int_0^t g_{s,k}(X_{s,k})ds \right) dt \tag{56}$$

$$\text{s.t. } \begin{cases} \frac{\partial}{\partial t}p_{t,k}(X_{t,k}) = -\nabla\cdot(u_{t,k}(X_t)p_{t,k}(X_{t,k})) + \frac{\sigma^2}{2}\Delta p_{t,k}(X_{t,k}) + g_{t,k}(X_{t,k})p_{t,k}(X_{t,k}) \\ p_0 = \pi_0,\ p_{t,k} = \pi_{t,k} \end{cases} \tag{57}$$

such that each branch independently solves the Fokker-Planck equation defined as $\frac{\partial}{\partial t}p_{t,k} = -\nabla\cdot(u_{t,k}p_{t,k}) + \frac{\sigma^2}{2}\Delta p_{t,k}$. Now, we show that the sum of unbalanced CondSOC problems still satisfies the global Fokker-Planck equation

$$\frac{\partial}{\partial t}p_t = -\nabla\cdot(u_t p_t) + \frac{\sigma^2}{2}\Delta p_t \tag{58}$$

where we define p_t as the weighted sum of the branched distributions given by

$$p_t(X_t) = \sum_{k=0}^K w_{t,k}p_{t,k}(X_t) \tag{59}$$

To obtain an expression for the global Fokker-Planck equation, we differentiate both sides and substitute the branched FPE as follows

$$\frac{\partial}{\partial t}p_t = \frac{\partial}{\partial t}\left[\sum_{k=0}^K w_{t,k}p_{t,k}\right]$$
$$\frac{\partial}{\partial t}p_t = \sum_{k=0}^K \left[w_{t,k}\left(\frac{\partial}{\partial t}p_{t,k}\right) + \left(\frac{\partial}{\partial t}w_{t,k}\right)p_{t,k} \right]$$
$$\frac{\partial}{\partial t}p_t = \sum_{k=0}^K \left[w_{t,k}\left(-\nabla\cdot(u_{t,k}p_{t,k}) + \frac{\sigma^2}{2}\Delta p_{t,k}\right) + g_{t,k}p_{t,k} \right]$$
(substitute branched FPE and $\partial_t w_{t,k} = g_{t,k}$)
$$\frac{\partial}{\partial t}p_t = \sum_{k=0}^K \left[-w_{t,k}\nabla\cdot(u_{t,k}p_{t,k}) + w_{t,k}\frac{\sigma^2}{2}\Delta p_{t,k} + g_{t,k}p_{t,k} \right]$$
$$\frac{\partial}{\partial t}p_t = \sum_{k=0}^K (-w_{t,k}\nabla\cdot(u_{t,k}p_{t,k})) + \sum_{k=0}^K w_{t,k}\frac{\sigma^2}{2}\Delta p_{t,k} + \sum_{k=0}^K g_{t,k}p_{t,k} \tag{60}$$

By linearity of the Laplacian, the diffusion term can be rewritten as

$$\sum_{k=0}^K w_{t,k}\frac{\sigma^2}{2}\Delta p_{t,k} = \frac{\sigma^2}{2}\Delta \underbrace{\left(\sum_{k=0}^K w_{t,k}p_{t,k}\right)}_{p_t} = \frac{\sigma^2}{2}\Delta p_t \tag{61}$$

Since the global growth term $\sum_{k=0}^{K} g_{t,k} p_{t,k}$ is the sum of the growth dynamics across all branches and is separate from the drift and diffusion dynamics, it doesn't alter the direction or motion of particles along the branched fields. Thus, both the diffusion and growth terms satisfy the global FPE.

Now, we set the divergence term in (60) equal to the global divergence $\nabla \cdot (u_t p_t)$ and derive the expression for the total drift field $u_t(X_t)$ that satisfies the global FPE. By the linearity of the divergence operator, we get

$$
\begin{aligned}
-\nabla \cdot (u_t p_t) &= \sum_{k=0}^{K} \left(-w_{t,k} \nabla \cdot (u_{t,k} p_{t,k})\right) \\
-\nabla \cdot (u_t p_t) &= -\nabla \cdot \left(\sum_{k=0}^{K} w_{t,k} u_{t,k} p_{t,k} \right) \\
-\nabla \cdot (u_t p_t) &= -\nabla \cdot \underbrace{\left(\frac{1}{p_t} \sum_{k=0}^{K} w_{t,k} u_{t,k} p_{t,k} \right)}_{u_t} p_t
\end{aligned} \tag{62}
$$

Under the global FPE constraint, the drift u_t is defined as the mass-weighted average of the drift fields for each branch, given by $u_t(X_t) = \frac{1}{p_t(X_t)} \sum_{k=0}^{K} w_{t,k}(X_t) u_{t,k}(X_t) p_{t,k}(X_t)$. Intuitively, this means that in the global context, for any $X_t = \mathbf{x}_t$, the drift of state X_t along the dynamics of branch k is scaled by the weight of $\mathbf{x}_t$ at time t along branch k and the ratio of probability density of $\mathbf{x}_t$ under branch k over the total probability density of $\mathbf{x}_t$ across all branches. Therefore, our definition of the weighted drift decoupled over individual branches satisfies the global FPE equation in (58), and this concludes the proof of Proposition 2.

Remark 1 (Reduction to Single Path GSBM) When $g_{t,0}(X_{t,0}) = 0$ and $g_{t,k}(X_{t,k}) = 0$ for all $t \in [0,1]$ and $k \in \{1, \ldots, K\}$, then the Branched CondSOC problem is the solution to the single path GSB problem given by

$$
\min_{u_t} \int \mathbb{E}_{p_{t|0,1}} \left[\frac{1}{2} \|u_t(X_t)\|^2 + V_t(X_t) \right] dt \;\; \text{s.t.} \;\; \begin{cases} dX_t = u_t(X_t)dt + \sigma dB_t \\ X_0 \sim \pi_0, \quad X_1 \sim \pi_1 \end{cases} \tag{63}
$$

where the probability density $p_{t|0,1}(X_t | \mathbf{x}_0, \mathbf{x}_1)$ is conditioned explicitly on a pair of endpoints $(\mathbf{x}_0, \mathbf{x}_1) \sim \pi_{0,1}$ drawn from the joint distribution.

C.3 Proof of Proposition 3

Proposition 3 (Solving the GSB Problem with Stage 1 and 2 Training) *Stage 1 and Stage 2 training yield the optimal drift $u_t^\star(X_t)$ that generates the optimal marginal probability distribution $p_t^\star(X_t)$ that solves the GSB problem in (3).*

Proof. Let the marginal probability distribution $p_t^\star(X_t)$ and corresponding drift $u_t^\star(X_t)$ define the optimal solution to the GSB problem. It suffices to show that

1. Given $(\mathbf{x}_0, \mathbf{x}_1) \sim \pi_{0,1}$, Stage 1 training with the trajectory loss $\mathcal{L}_{\text{traj}}(\eta)$ (12) yields the interpolant $\mathbf{x}_{t,\eta}^\star$ and time-derivative $\dot{\mathbf{x}}_{t,\eta}^\star$ that define $p_t^\star(X_t)$.
2. Given $p_t^\star(X_t)$, Stage 2 training with the explicit flow matching loss $\mathcal{L}_{\text{flow}}(\theta)$ (13) yields the optimal drift $u_t^\star(X_t)$.

To prove Part 1, we establish the following Lemma.

Lemma 1. *Given the Markovian reference process $\mathbb{Q}$ with drift $v_t^\star(X_t)$ that is the minimizer of the unconstrained GSB problem in (3), Stage 1 training returns the velocity field $\dot{\mathbf{x}}_{t,\eta}^\star$ that generates reciprocal projection $\Pi^\star = \text{proj}_{\mathcal{R}(\mathbb{Q})}(\mathbb{P})$ of the path measure $\mathbb{P} = p_{0,1}$.*

Proof of Lemma. It suffices to prove that $\dot{\mathbf{x}}_{t,\eta}^\star$ minimizes the KL-divergence with $v_t^\star(X_t)$ under endpoint constraints $X_0 = \mathbf{x}_0$ and $X_1 = \mathbf{x}_1$.

Consider the unconstrained Markovian drift $v_t^\star$ that is the minimizer of the energy function given by

$$v_t^\star = \arg\min_{v_t} \int_0^1 \mathbb{E}_{p_t}\left[\frac{1}{2}\|v_t(X_t)\|^2 + V_t(X_t)\right] dt \tag{64}$$

The class of interpolants is given by parameters η is defined as

$$\mathbf{x}_{t,\eta} = (1-t)\mathbf{x}_0 + t\mathbf{x}_1 + t(1-t)\varphi_{t,\eta}(\mathbf{x}_0, \mathbf{x}_1) \tag{65}$$
$$\dot{\mathbf{x}}_{t,\eta} = \mathbf{x}_1 - \mathbf{x}_0 + t(1-t)\dot{\varphi}_{t,\eta}(\mathbf{x}_0, \mathbf{x}_1) + (1-2t)\varphi_{t,\eta}(\mathbf{x}_0, \mathbf{x}_1) \tag{66}$$

Stage 1 training yields the optimal interpolant $\dot{\mathbf{x}}_{t,\eta}^\star$ that minimizes the energy function across all time points $t \in [0,1]$, defined as

$$\dot{\mathbf{x}}_{t,\eta}^\star = \arg\min_{\dot{\mathbf{x}}_{t,\eta}^\star} \int_0^1 \mathbb{E}_{p_t}\left[\frac{1}{2}\|\dot{\mathbf{x}}_{t,\eta}^\star(\mathbf{x}_0, \mathbf{x}_1)\|^2 + V_t(\mathbf{x}_{t,\eta}^\star)\right] dt \tag{67}$$

Therefore, we aim to prove that $\dot{\mathbf{x}}_{t,\eta}^\star$ is the velocity field corresponding to $\Pi^\star = p_{0,1}\mathbb{Q}_{\cdot|0,1}$, which is the reciprocal projection of $p_{0,1}$ onto the class of path measures that share the same bridge marginals conditioned on a pair of endpoints, called the reciprocal class $\mathcal{R}(\mathbb{Q})$ (Léonard, 2014).

By definition, the reciprocal projection $\Pi^\star$ is defined as the minimizer of the KL-divergence with $\mathbb{P} = p_{0,1}$ that lies within the reciprocal class $\mathcal{R}(\mathbb{Q})$ of measures that match the bridge conditions of $\mathbb{Q}$.

$$\Pi^\star = \arg\min_{\Pi \in \mathcal{R}(\mathbb{Q})} \mathrm{KL}(\mathbb{P}\|\Pi) \tag{68}$$

The reciprocal projection matches the endpoint constraints of $\mathbb{P}$ while following the bridge conditionals $\mathbb{Q}_{\cdot|0,1} = \mathbb{Q}(\cdot|X_0 = \mathbf{x}_0, X_1 = \mathbf{x}_1)$ (Léonard, 2014). Therefore, we can write the generating velocity field $\dot{\mathbf{x}}_{t,\eta}^\star$ as

$$\dot{\mathbf{x}}_{t,\eta}^\star(\mathbf{x}_0, \mathbf{x}_1) = \mathbb{E}_{\mathbf{x}_t \sim \mathbb{Q}}\left[\dot{\mathbf{x}}_t | X_0 = \mathbf{x}_0, X_1 = \mathbf{x}_1\right] \tag{69}$$

Since both $v_t^\star$ generating $\mathbb{Q}$ and $\dot{\mathbf{x}}_{t,\eta}^\star$ minimize the same energy function, the objective in (68) reduces to determining $\dot{\mathbf{x}}_{t,\eta}^\star$ that is the minimizer of the dynamic formulation of KL divergence between the reference process $\mathbb{Q}$ and all path measures $\mathbb{P} = p_{0,1}$ that preserve endpoint constraints given by

$$\dot{\mathbf{x}}_{t,\eta}^\star = \arg\min_{\dot{\mathbf{x}}_{t,\eta}} \mathrm{KL}(\mathbb{P}\|\mathbb{Q}) \tag{70}$$
$$= \arg\min_{\dot{\mathbf{x}}_{t,\eta}} \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_1)\sim p_{0,1}}\left[\int_0^1 \frac{1}{2}\|\dot{\mathbf{x}}_{t,\eta}^\star - v_t^\star(\mathbf{x}_{t,\eta})\|^2 dt\right] \tag{71}$$

Therefore, we conclude that $\dot{\mathbf{x}}_{t,\eta}^\star$ generates the reciprocal projection $\Pi^\star = p_{0,1}\mathbb{Q}_{\cdot|0,1}$.

By Lemma 1, we know that $v_t^\star$ is the optimal drift energy function in the GSB problem without endpoint constraints and $\dot{\mathbf{x}}_{t,\eta}^\star$ is the reciprocal projection that follows the dynamics of $v_t^\star$ while preserving the coupling $p_{0,1}$. Therefore, we can define $u_{t|0,1}^\star(X_t|\mathbf{x}_0, \mathbf{x}_1) \equiv \dot{\mathbf{x}}_{t,\eta}^\star(\mathbf{x}_0, \mathbf{x}_1)$ as the conditional drift that generates the conditional probability distribution $p_{t|0,1}^\star(X_t|\mathbf{x}_0, \mathbf{x}_1)$ that satisfies the Fokker-Planck equation

$$\frac{\partial}{\partial t} p_{t|0,1} = -\nabla \cdot (u_{t|0,1} p_{t|0,1}) + \frac{1}{2}\sigma^2 \Delta p_{t|0,1} \tag{72}$$

Given that $p_{t|0,1}^\star(X_t|\mathbf{x}_0, \mathbf{x}_1)$ is the optimal bridge that solves the GSB problem for a pair of endpoints $(\mathbf{x}_0, \mathbf{x}_1) \sim p_{0,1}$, we can define the marginal probability distribution $p_t^\star$ as the mixture of bridges

$$p_t^\star = \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_1)\sim \pi_{0,1}}\left[p_{t|0,1}^\star(X_t|\mathbf{x}_0, \mathbf{x}_1)\right] \tag{73}$$

which concludes the proof of Part 1.

For Part 2 of the proof, we aim to show that Stage 2 training yields the optimal Markovian drift $u_t^\star(X_t)$ that *generates* $p_t^\star(X_t)$. To do this, we write the Fokker-Planck equation for $p_t^\star(X_t)$ in terms

of the conditional bridge $p_{t|0,1}$ and drift field $u_{t|0,1}$ to extract an expression for $u_t^\star(X_t)$ that satisfies it. Starting from the definition of $p_t^\star$, we have

$$\begin{aligned}
p_t^\star &= \int p_{t|0,1}^\star p_{0,1} d\mathbf{x}_{0,1} \\
\frac{\partial}{\partial t} p_t^\star &= \int \left(\frac{\partial}{\partial t} p_{t|0,1}^\star \right) p_{0,1} d\mathbf{x}_{0,1} \\
&= \int \left(-\nabla \cdot (u_{t|0,1} p_{t|0,1}^\star) + \frac{1}{2}\sigma^2 \Delta p_{t|0,1}^\star \right) p_{0,1} d\mathbf{x}_{0,1} \\
&= \int \left(-\nabla \cdot (u_{t|0,1} p_{t|0,1}^\star) \right) p_{0,1} d\mathbf{x}_{0,1} + \int \left(\frac{1}{2}\sigma^2 \Delta p_{t|0,1}^\star \right) p_{0,1} d\mathbf{x}_{0,1} \\
&= -\nabla \cdot \int (u_{t|0,1} p_{t|0,1}^\star) p_{0,1} d\mathbf{x}_{0,1} + \frac{1}{2}\sigma^2 \Delta \int p_{t|0,1}^\star p_{0,1} d\mathbf{x}_{0,1} \\
&= -\nabla \cdot \underbrace{\int (u_{t|0,1} p_{t|0,1}^\star) p_{0,1} d\mathbf{x}_{0,1}}_{(u_t^\star p_t^\star)} + \frac{1}{2}\sigma^2 \Delta \underbrace{\int p_{t|0,1}^\star p_{0,1} d\mathbf{x}_{0,1}}_{p_t^\star}
\end{aligned} \tag{74}$$

For (74) to satisfy the Fokker-Planck equation, we set the first term equal to $(u_t^\star p_t^\star)$ and solve for $u_t^\star$ to get

$$u_t^\star p_t^\star = \int (u_{t|0,1}^\star p_{t|0,1}^\star) p_{0,1} d\pi_{0,1} \tag{75}$$

$$u_t^\star = \frac{\mathbb{E}_{\pi_{0,1}} \left[u_{t|0,1}^\star p_{t|0,1}^\star \right]}{p_t^\star} \tag{76}$$

Therefore, the optimal Markovian drift (or *Markovian projection*) is the *average* of the conditional drifts defined in part 1 as $u_{t|0,1}^\star \equiv \dot{\mathbf{x}}_{t,\eta}^\star$ over the joint distribution $p_{0,1}$. This means that the minimizer of the conditional flow matching loss $\mathcal{L}_{\text{flow}}(\theta)$ in (13) defined as the expected mean-squared error between a Markovian drift field $u_t(X_t)$ and $\dot{\mathbf{x}}_{t,\eta}^\star$ over pairs $(\mathbf{x}_0, \mathbf{x}_1) \sim p_{0,1}$ in the dataset is the optimal drift $u_t^\star(X_t)$ that solves the GSB problem.

$$u_t^\star(X_t) = \arg\min_\theta \mathcal{L}_{\text{flow}}(\theta) = \arg\min_{u_t^\theta} \int_0^1 \mathbb{E}_{(\mathbf{x}_0,\mathbf{x}_1)\sim p_{0,1}} \|\dot{\mathbf{x}}_{t,\eta}^\star - u_t^\theta(X_t)\|^2 dt \tag{77}$$

which concludes the proof of Proposition 3. Since the drift for each branch $u_{t,k}^\theta(X_t)$ are trained independently in Stage 2, we can extend this result across all $K+1$ branches and conclude that the sequential Stage 1 and Stage 2 training procedures yields the optimal set of drifts $\{u_{t,k}^\star\}_{k=0}^K$ that *generate* the optimal probability paths $\{p_{t,k}^\star\}_{k=0}^K$ that solves the GSB problem for each branch.

C.4 Proof of Proposition 4

Lemma 2. *Suppose the optimal drift field $u_{t,k}^\star : \mathbb{R}^d \to \mathbb{R}^d$ and probability density $p_{t,k}^\star : \mathbb{R}^d \to \mathbb{R}$ that minimizes the GSB problem in (3) is well-defined over the state space $\mathcal{X} \subseteq \mathbb{R}^d$ for each branch. Then, the optimal weight $w_{t,k}^\star$ at any of the secondary branches $k \in \{1, \dots, K\}$ is non-decreasing over the interval $t \in [0,1]$. Equivalently, the optimal growth rates $g_{t,k}^\star(X_t) \geq 0$ for all $t \in [0,1]$.*

Proof. We will prove this lemma by contradiction. Suppose there exists a branch k that decreases in weight over the time interval $[t_1, t_2]$ for $0 \leq t_1 < t_2 \leq 1$, such that $w_{t_1,k} > w_{t_2,k}$. We know that the target weight at time $t = 1$ is non-negative $w_{1,k} \geq 0$ and the total weight across all branches is conserved (i.e. $\sum_{k=0}^K w_{t,k}(X_t) = w_t^{\text{total}}$) and non-decreasing (i.e. $\partial_t w_t^{\text{total}} > 0$ for all $t \in [0,1]$). In this proof, we let $\mathcal{E}_k(t_1, t_2) = \int_{t_1}^{t_2} \left[\frac{1}{2} \|u_{t,k}^\star(X_t)\|_2^2 + V_t(X_t) \right]$ denote the energy of following the dynamics of the kth branch over the time interval $[t_1, t_2]$.

Then, there can only be two possible reasons for the loss of mass along a branch k: **(1)** the mass is destroyed, or **(2)** the mass is transferred to a different branch $j \neq k$.

Case 1. The destruction of mass directly violates the assumption that the total mass across all branches is conserved and non-decreasing. So, we only need to consider the possibility of Case 2.

Case 2. Suppose the mass is transferred to a different branch $j \neq k$ over the interval $[t_1, t_2]$. By Proposition 3, Stage 1 and 2 training yields the optimal velocity fields $\{u^\star_{t,k}(X_t)\}_{k=0}^K$ that generate the optimal interpolating probability density $\{p^\star_{t,k}(X_t)\}_{k=0}^K$ that independently minimize the GSB problem in (3).

If mass is transferred from branch k to branch j over the interval $[t_1, t_2]$, it must be compensated for over time $[t_2, 1]$ to reach the target weight $w_{1,k} > 0$. Then, without loss of generality, we can consider two sub-cases: (1.1) the mass is compensated from the primary branch, and (1.2) the mass that diverges to the jth branch returns to the kth branch following a continuous trajectory.

Case 1.1. Since all mass along the K secondary branches originates from the primary branch, this implies that there exists a positive weight $\tilde{w} > 0$ that first follows the dynamics of branch k and is transferred to branch j, contributing to the final weight of the jth endpoint, and the total weight supplied from the primary branch to branch k is $w_{1,k} + \tilde{w}$. Given that each branch has no capacity constraints, it follows that the dynamics along branch k over $[0, t_1]$ and branch j over $[t_1, 1]$ is optimal for all mass reaching endpoint j. Formally, we express this in terms of energy as

$$\mathcal{E}_k(0, t_1) + \mathcal{E}_j(t_1, 1) < \mathcal{E}_j(0, 1) \tag{78}$$

which contradicts the assumption that the dynamics of branch j given by $(u^\star_{t,j}, p^\star_{t,j})$ are optimal over the state space $\mathcal{X}$.

Case 1.2. In this case, there exists a positive mass that follows the dynamics of branch k over the interval $[0, t_1]$, the dynamics of branch j over $[t_1, t_2]$, and back to branch k over $[t_2, 1]$. Similarly to Case 1.1, given that each branch has no capacity constraints, this implies that this concatenation of dynamics is optimal for all mass reaching endpoint k. Expressing in terms of energy, we have

$$\mathcal{E}_k(0, t_1) + \mathcal{E}_j(t_1, t_2) + \mathcal{E}_k(t_2, 1) < \mathcal{E}_k(0, 1) \tag{79}$$

which contradicts the assumption that the dynamics of branch k given by $(u^\star_{t,k}, p^\star_{t,k})$ are optimal over the state space $\mathcal{X}$.

Given that both sub-cases lead to a contradiction of the optimality assumption, we conclude that mass along each branch cannot be transferred to another branch and is non-decreasing over $t \in [0, 1]$.

Note that we do not need to consider the case where the mass is compensated from another secondary branch $\ell \neq j$, as this would imply that mass is transferred from branch ℓ to branch k, which is not possible under the same argument.

Proposition 4 (Existence of Optimal Growth Functions) *Assume the state space $\mathcal{X} \subseteq \mathbb{R}^d$ is a bounded domain within $\mathbb{R}^d$. Let the optimal probability density of branch k be a known non-negative function bounded in $[0, 1]$, denoted as $p^\star_{t,k} : \mathcal{X} \times [0, 1] \to [0, 1] \in L^\infty(\mathcal{X} \times [0, 1])$. By Lemma 2, we can define the set of feasible growth functions in the set of square-integrable functions L^2 as*

$$\mathcal{G} := \{g = (g_{t,0}, \ldots, g_{t,K}) \in L^2(\mathcal{X} \times [0, 1]) \mid g_{t,k}(\mathbf{x}) : \mathcal{X} \times [0, 1] \to \mathbb{R}\ ,\ g_{t,k}(\mathbf{x}) \geq 0\} \tag{80}$$

Let the growth loss be the functional $\mathcal{L}(g) : L^2(\mathcal{X} \times [0, 1]) \to \mathbb{R}$. Then, there exists an optimal function $g^\star = (g^\star_{t,0}, \ldots, g^\star_{t,K}) \in L^2$ where $g^\star_{t,k} \in \mathcal{G}$ such that $\mathcal{L}(g^\star) = \inf_{g \in \mathcal{G}} \mathcal{L}(g)$ which can be obtained by minimizing $\mathcal{L}(g)$ over $\mathcal{G}$.

Proof. This proof draws on several concepts from functional and convex analysis. For a comprehensive background on these concepts, see Benešová & Kružík (2016). We follow the *direct method in the calculus of variations* (Dacorogna, 1989) for proving there exists a minimizer for the functional $\mathcal{L}_{\text{growth}}(g)$ in (17) with the following steps:

1. Show that the set $\mathcal{G}$ of *feasible* growth functions is convex and closed under the weak topology of the set of square integrable functions L^2 (Lemma 3).
2. Show that the minimizing sequence $\{g^{(n)}\}$ has a weakly convergent subsequence (Lemma 4).
3. Show that the functional $\mathcal{L}_{\text{growth}}(g)$ is weakly lower semi-continuous (Lemmas 5, 6, 7, 8).

We prove each with a sequence of Lemmas.

Lemma 3. *The set of feasible growth functions* $\mathcal{G} := \{g_{t,k}(\mathbf{x}) : \mathcal{X} \times [0,1] \to \mathbb{R} \in L^2(\mathcal{X} \times [0,1]) \mid g_{t,k}(\mathbf{x}) \geq 0\}$ *is convex and closed over the weak topology of* L^2.

Proof. We first prove convexity and then closure under the weak topology of L^2.

Proof of Convexity. To prove that the set of functions $\mathcal{G}$ is convex, we first define what it means for a set of functions to be convex.

Definition 4 (Convex Set)**.** *A set of functions* $\mathcal{G}$ *is said to be convex if any convex combination of two functions in the set* $g_1, g_2 \in \mathcal{G}$ *is also in the set. Formally,* $\mathcal{G}$ *if*

$$\forall g, g' \in \mathcal{G}, \quad \forall \lambda \in [0,1], \quad g^\lambda = \lambda g + (1-\lambda) g' \in \mathcal{G} \tag{81}$$

Recall our definition of $\mathcal{G}$ as the set of functions that return strictly non-negative values:

$$\mathcal{G} := \{g_{t,k}(\mathbf{x}) : \mathcal{X} \times [0,1] \to \mathbb{R} \in L^2(\mathcal{X} \times [0,1]) \mid g_{t,k}(\mathbf{x}) \geq 0\} \tag{82}$$

Given any $\lambda \in [0,1]$, we have

$$\begin{aligned} g^\lambda_{t,k}(\mathbf{x}) &= \lambda g_{t,k}(\mathbf{x}) + (1-\lambda) g'_{t,k}(\mathbf{x}) \\ g^\lambda_{t,k}(\mathbf{x}) &\geq \lambda \cdot 0 + (1-\lambda) \cdot 0 \\ g^\lambda_{t,k}(\mathbf{x}) &\geq 0 \end{aligned} \tag{83}$$

which means $g^\lambda_{t,k} \in \mathcal{G}$ and $\mathcal{G}$ is convex.

Proof of Closure. First, we define what it means to be closed under the weak topology of L^2.

Definition 5 (Closure Under Weak Topology of L^2)**.** *A set of functions* $\mathcal{G}$ *is said to be closed in the weak topology of* L^2 *if the statement is true: if a sequence of functions* $\{g^{(n)} : g^{(n)} \in \mathcal{G}\}$ *indexed by* n*, converges to some function* $g^{(\infty)} \in L^2$ *as* $n \to \infty$*, then* $g^{(\infty)} \in \mathcal{G}$.

To show that $\mathcal{G}$ is closed under weak topology of L^2, we need to show that all sequences $\{g^{(n)}_{t,k} : g^{(n)}_{t,k} \in \mathcal{G}\}$ converge to $g \in \mathcal{G}$ as $n \to \infty$, such that $g \geq 0$. The proof follows directly from Fatou's Lemma (Ekeland & Temam, 1999) which states that given a sequence of non-negative, measurable functions $\{g^{(n)}_{t,k} : g^{(n)}_{t,k} \in \mathcal{G}\}$, the following is true

$$0 \leq \int_{\mathcal{X}} \liminf_{n\to\infty} g^{(n)}_{t,k}(\mathbf{x}) d\mathbf{x} \leq \liminf_{n\to\infty} \int_{\mathcal{X}} g^{(n)}_{t,k}(\mathbf{x}) d\mathbf{x} \qquad \text{(Fatou's Lemma)}$$

meaning that the limit of a converging sequence of non-negative functions $g^{(\infty)}_{t,k}$ is also non-negative over the state space $\mathcal{X}$, and thus is in the set of feasible growth functions $g^{(\infty)}_{t,k} \in \mathcal{G}$, concluding our proof.

Lemma 4. *Given a minimizing sequence* $\{g^{(n)} \in \mathcal{G}\}$ *under the functional* $\mathcal{L}(g) : L^2 \to \mathbb{R}$ *such that* $\mathcal{L}(g^{(n)}) \to \inf_{g\in\mathcal{G}} \mathcal{L}(g)$*, there exists a subsequence* $\{g^{(n_i)} \in \mathcal{G}\}$ *that converges weakly in* L^2 *to some limit* $g^\star \in \mathcal{G}$.

Proof. It suffices to show the following:

1. The functional $\{g^{(n)}\}$ is bounded in L^2 such that there exists a positive value M where $\|g^{(n)}\|_{L^2} \leq M$ for all n.
2. The space of feasible functions $\mathcal{G}$ is reflexive, such that all bounded sequences have a weakly convergent subsequence in L^2.

The proof of Part 1 follows from the growth penalty term in $\mathcal{L}(g)$ defined as the squared-norm $\|g\|^2_{L^2}$ of the growth functional. This term ensures that $\mathcal{L}(g)$ is *coercive*, such that

$$\|g^{(n)}\|_{L^2} \to \infty \implies \mathcal{L}(g^{(n)}) \to \infty \tag{84}$$

which ensures that the sequence does not diverge to infinity in norm without incurring a penalty from the loss functional. Given that $\mathcal{L}(g^{(n)}) \to \inf_{g\in\mathcal{G}} \mathcal{L}(g)$, where $\inf_{g\in\mathcal{G}} \mathcal{L}(g) < \infty$, it follows from coercivity that $\|g^{(n)}\|_{L^2}$ does not diverge to infinity and $\{g^{(n)}\}$ is bounded in L^2.

The proof of Part 2 follows from the well-established result that for $1 < p < \infty$, the space L^p is reflexive (Beauzamy, 2011). Therefore, by reflexivity of L^2, we have that every minimizing sequence $\{g^{(n)}\}$ has a weakly convergent subsequence $\{g^{(n_i)}\}$, such that $g^{(n_i)} \to g^\star$ in L^2, concluding our proof.

Before proving that each component of the loss functional is weakly lower semi-continuous, we establish the definitions for weakly continuous and weakly lower semi-continuous functionals.

Definition 6 (Weakly Continuous Functionals)**.** *A functional $\mathcal{L} : L^2 \to \mathbb{R}$ is said to be weakly continuous if it satisfies*

$$n \to \infty \implies g^{(n)} \to g \implies \mathcal{L}(g^{(n)}) \to \mathcal{L}(g) \tag{85}$$

such that if a sequence $\{g^{(n)} : g^{(n)} \in \mathcal{G}\}$ converges $g^{(n)} \to g$ as $n \to \infty$, then the functional also converges $\mathcal{L}(g^{(n)}) \to \mathcal{L}(g)$.

Definition 7 (Weakly Lower Semi-Continuous Functionals)**.** *A functional $\mathcal{L} : L^2 \to \mathbb{R}$ is said to be weakly lower semi-continuous (w.l.s.c.) if it satisfies*

$$n \to \infty \implies g^{(n)} \to g \implies \lim \inf_{n\to\infty} \mathcal{L}(g^{(n)}) \geq \mathcal{L}(g) \tag{86}$$

such that if a sequence $\{g^{(n)} : g^{(n)} \in \mathcal{G}\}$ converges $g^{(n)} \to g$ as $n \to \infty$, then the functional is lower bounded by $\mathcal{L}(g)$. By definition, weak continuity implies w.l.s.c.

Lemma 5. *The functional $\mathcal{L}_{match}(g) : L^2 \to \mathbb{R}$ defined as*

$$\begin{aligned}
\mathcal{L}_{match}(g) &= \sum_{k=0}^{K} \mathbb{E}_{p^\star_{t,k}} \left(w_{0,k} + \int_0^1 g_{t,k} dt - w^\star_{1,k} \right)^2 \\
&= \sum_{k=0}^{K} \left(w_{0,k} + \int_0^1 \mathbb{E}_{p^\star_{t,k}}[g_{t,k}] dt - w^\star_{1,k} \right)^2 && \text{(linearity of expectation)} \\
&= \sum_{k=0}^{K} \left(\int_0^1 \int_{\mathcal{X}} p^\star_{t,k} g_{t,k} d\mathbf{x} dt + c \right)^2 && (87)
\end{aligned}$$

where $c = w_{0,k} - w^\star_{1,k}$ is a constant, is weakly lower semi-continuous in L^2 (w.l.s.c.).

Proof of Lemma. First, we show that the map $\phi(g_{t,k}) = \int_{\mathcal{X}} p^\star_{t,k} g_{t,k} d\mathbf{x}$ is a bounded linear functional in L^2. By linearity of integration, we have that for two functions $g, g' \in \mathcal{G}$, $\phi(cg + c'g') = c\phi(g) + c'\phi(g')$, so $\phi(\cdot)$ is a linear map. To establish that $\phi(\cdot)$ is bounded in L^2, we must show that $|\phi(g)| \leq C\|g\|_{L_2}$. By the Cauchy-Schwarz Inequality (Steele, 2004), we have

$$|\phi(g_{t,k})| = |\langle p^\star_{t,k}, g_{t,k} \rangle| = \left| \int_{\mathcal{X}} p^\star_{t,k} g_{t,k} d\mathbf{x} \right| \leq \|p^\star_{t,k}\|_{L^2} \|g_{t,k}\|_{L^2} \tag{88}$$

which is valid as $p^\star_{t,k}, g_{t,k} \in L^2$. Since $\|p^\star_{t,k}\|_{L^2}$ is a fixed constant with respect to $g_{t,k}$, we have shown that $|\phi(g_{t,k})| \leq C\|g_{t,k}\|_{L_2}$ and $g_{t,k}$ is bounded. By the definition of weak topology on L^2, all bounded linear functionals are weakly continuous, such that $\lim_{n\to\infty} \phi(g^{(n)}) \to \phi(g)$. Given that $\phi(g)$ is weakly continuous, as $g^{(n)} \to g^\star$, we have $\phi(g^{(n)}) \to \phi(g^\star)$. Since the square function $\psi(\cdot) = (\cdot)^2$ is convex and continuous and bounded below by $\psi \geq 0$, the function does not contain discontinuities and $\lim_{n\to\infty}(\phi(g^{(n)}) - c)^2 \geq (\phi(g^\star) - c)^2$, which is the definition of a w.l.s.c. functional. Since the sum of w.l.s.c. functionals is also w.l.s.c., we conclude our proof.

Lemma 6. *The functional $\mathcal{L}_{energy}(g) : L^2 \to \mathbb{R}$ defined as*

$$\begin{aligned}
\mathcal{L}_{energy}(g) &= \int_0^1 \mathbb{E}_{p^\star_{t,k}} \left[\frac{1}{2} \|u_{t,k}(X_t)\|^2 + V_t(X_t) \right] \left(w_{0,k} + \int_0^t g_{s,k}(X_t) ds \right) dt \\
&= \int_0^1 \alpha(t) \left(w_{0,k} + \int_0^t \int_{\mathcal{X}} p^\star_{s,k} g_{s,k} d\mathbf{x} ds \right) dt && \text{(linearity of expectation)}
\end{aligned}$$

where $\alpha(t)$ is a constant not dependent on $g_{t,k}$, is weakly lower semi-continuous in L^2 (w.l.s.c.).

Proof of Lemma. Following a similar proof as Lemma 5, we have that the map $\phi(g_{s,k}) = \int_{\mathcal{X}} p^{\star}_{s,k} g_{s,k} d\mathbf{x}$ is a bounded linear functional, and thus is weakly continuous in L^2. Given that $\phi(g)$ is bounded, $t \in [0,1]$, and linearity of integration holds, the integral of $\phi(g)$ over the interval $s \in [0,t]$ remains a bounded linear functional in L^2, and thus is weakly continuous in L^2. Following similar logic, the outer integral $\int_0^1 \alpha(t) \left(w_{0,k} + \int_0^t \phi(g) ds \right) dt$ is also a bounded linear functional given that both $\alpha(t)$ and $w_{0,k}$ are constants, and we have shown that $\mathcal{L}_{\text{energy}}(g)$ is weakly continuous. Since weak continuity implies w.l.s.c. and the sum of w.l.s.c. functionals is also w.l.s.c., we conclude our proof.

Lemma 7. *The mass loss functional $\mathcal{L}_{mass}(g) : L^2 \to \mathbb{R}$ defined as*

$$\begin{aligned}
\mathcal{L}_{mass}(g) &= \int_0^1 \mathbb{E}_{\{p_{t,k}\}_{k=0}^K} \left[\sum_{k=0}^K w_{t,k}^{\phi}(\mathbf{x}_{t,k}) - w_t^{total} \right]^2 dt \\
&= \int_0^1 \left[\sum_{k=0}^K \left(w_{0,k} + \int_0^t \mathbb{E}_{p_{t,k}}[g_{t,k}] \right) - w_t^{total} \right]^2 dt \qquad \text{(linearity of expectation)} \\
&= \int_0^1 \left[1 + \sum_{k=0}^K \int_0^t \mathbb{E}_{p_{t,k}}[g_{t,k}] - w_t^{total} \right]^2 dt \qquad (89)
\end{aligned}$$

where w_t^{total} is a constant not dependent on $g_{t,k}$, is weakly lower semi-continuous in $(L^2)^{K+1}$ (w.l.s.c.). Note that we do not need to account for the negative penalty loss as we assume the growth function is strictly positive.

Proof of Lemma. Following a similar proof as Lemma 5, we have that the map $\phi(g) = \int_{\mathcal{X}} p^{\star}_{s,k} g_{s,k} d\mathbf{x}$ is a bounded linear functional, and thus is weakly continuous in L^2. Since the sum of bounded linear functionals across $K+1$ branches is also weakly continuous in L^2, following similar logic to Lemma 5, the composition of a weakly continuous functional with the convex and continuous square map is w.l.s.c.. Since the sum of w.l.s.c. functionals is also w.l.s.c., $\mathcal{L}_{\text{mass}}(g)$ is w.l.s.c., which concludes our proof.

Lemma 8. *The combined Stage 3 training loss $\mathcal{L}_{growth}(g) = \lambda_{energy}\mathcal{L}_{energy}(g) + \lambda_{match}\mathcal{L}_{match}(g) + \lambda_{mass}\mathcal{L}_{mass}(g) + \lambda_{growth}\|g\|_{L^2}$ is weak lower semi-continuous in L^2 (w.l.s.c.).*

Proof of Lemma. Given the weighted sum of w.l.s.c. functionals $\sum_{i=1}^M \lambda_i \mathcal{L}_i(g)$ that each satisfy $\lim_{n\to\infty} \mathcal{L}_i(g^{(n)}) \geq \mathcal{L}_i(g)$ for constants $\{\lambda_i\}_{i=1}^M$ for a converging sequence $g^{(n)} \to g$ as $n \to \infty$, it follows easily that

$$\lim_{n\to\infty} \sum_{i=1}^M \mathcal{L}_i(g^{(n)}) \geq \sum_{i=1}^M \mathcal{L}_i(g) \qquad (90)$$

By definition, the norm in L^2 is lower semi-continuous in the strong topology and thus the weak topology. Since $\mathcal{L}_{\text{energy}}$, $\mathcal{L}_{\text{match}}$, and $\mathcal{L}_{\text{mass}}$ are w.l.s.c. by Lemmas 5, 6, and 7, we conclude that $\mathcal{L}_{\text{growth}}(g)$ is w.l.s.c., which concludes the proof.

In total, we have shown:

1. The set $\mathcal{G}$ of *feasible* growth functions is convex and closed under the weak topology of the set of square integrable functions L^2 (Lemma 3).
2. The minimizing sequence $\{g^{(n)}\}$ has a weakly convergent subsequence (Lemma 4).
3. The functional $\mathcal{L}(g)$ is weakly lower semi-continuous (Lemmas 5, 6, 7, 8).

Thus, by the *direct method in the calculus of variations*, we have shown

$$\exists g^{\star} = (g^{\star}_{t,0}, \ldots, g^{\star}_{t,1}) \in \mathcal{G} \;\text{ s.t. }\; \mathcal{L}(g^{\star}) = \inf_{g \in \mathcal{G}} \mathcal{L}(g) \qquad (91)$$

which concludes the proof of Proposition 4.

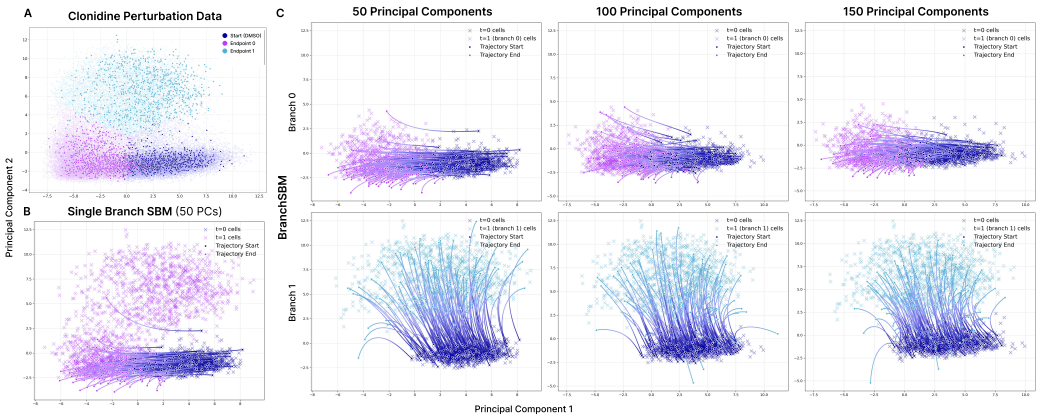

Figure 6: **Results for Clonidine Perturbation Modeling.** **(A)** Gene expression data of DMSO control (set to $t = 0$) and cell states (set to $t = 1$) after Clonidine perturbation with two distinct endpoints (pink and purple). **(B)** The simulated trajectories for single-branch SBM on the top 50 PCs with both clusters. All samples take the low-energy path without reaching the second cluster. **(C)** The simulated endpoints of the top 50, 100, and 150 PCs at $t = 1$ on the validation data for each branch.

D Additional Experiments and Discussions

D.1 Comparison to Single-Branch Schrödinger Bridge Matching

Setup For each experiment, we compared the performance of BranchSBM against single-branch SBM. Instead of clustering the data at $t = 1$ into distinct endpoint distributions, we left the unclustered data as a single target distribution p_1 and let the model learn the optimal Schrödinger Bridge from the initial distribution π_0. For the single-branch task, we assume mass conservation and set the weights for all samples from π_0 and π_1 to 1.0, while keeping the model architecture, state-cost V_t, and hyperparameters equivalent to BranchSBM. Since single-branch SBM does not require modeling the growth of separate branches, we train only Stages 1 and 2 to optimize the drift field u_t^θ of the single branch. For evaluation, the trajectories of validation samples from the initial distribution $\mathbf{x}_0 \sim \pi_0$ were simulated over $t \in [0, 1]$ and compared with the ground truth distribution at $t = 1$. For BranchSBM, we take the overall distribution generated across all branches and compare it to the overall ground truth distribution.

Modeling Mouse Hematopoiesis Differentiation In Figure 7, we provide a side-by-side comparison of the reconstructed distributions at time t_1 and t_2 using BranchSBM (top) with two branches and single-branch SBM (bottom), as well as the learned trajectories over the time interval $t \in [t_1, t_2]$. While single-branch SBM produces samples that loosely capture the two target cell fates, the resulting distributions display high variance and fail to align with the true differentiation trajectories. In contrast, BranchSBM generates intermediate distributions that are sharply concentrated along the correct developmental paths, more faithfully reflecting the underlying branching structure of the data.

Modeling Clonidine and Trametinib Perturbation For both Clonidine and Trametinib, we performed the single-branch experiment on the top 50 principal components identified by PCA. After training, we simulated all the validation samples from the initial distribution π_0 by integrating the single velocity field u_t^θ to $t = 1$. We evaluated the performance of the single-branch parameterization by computing the RBF-MMD (97) of all PCs and $\mathcal{W}_1$ (95) and $\mathcal{W}_2$ (96) distances of the top-2 PCs of the simulated samples at time $t = 1$ with the ground truth data points.

In Table 4, we show that BranchSBM with two branches trained on gene expression vectors across all dimensions $d \in \{50, 100, 150\}$ outperforms single-branch SBM on dimension $d = 50$ in reconstructing the distribution of cells perturbed with Clonidine. In Table 5, we further show improved performance of BranchSBM with three branches compared to single-branch SBM.

In Figure 8A and 8B, we see that single-branch SBM only reconstructs cluster 0, while failing to generate samples from cluster 1 for Clonidine perturbation or clusters 1 and 2 for Trametinib

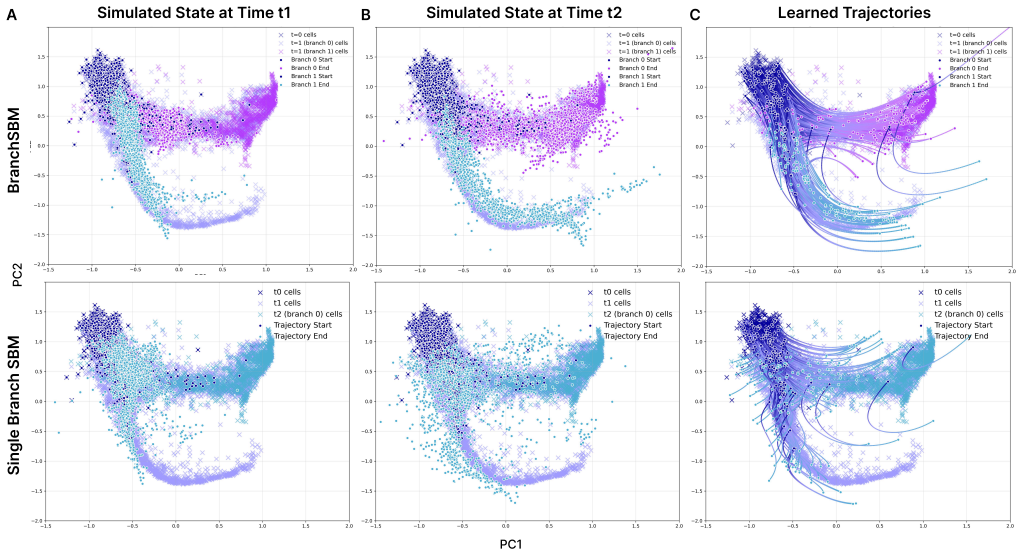

Figure 7: **Comparison of BranchSBM to Single-Branch SBM for Cell-Fate Differentiation.** Mouse hematopoiesis scRNA-seq data is provided for three time points t_0, t_1, t_2. **(A, B)** Distribution of simulated cell states at time **(A)** t_1 and **(B)** t_2 across both branches for BranchSBM (top) and single-branch SBM (bottom). **(C)** Learned trajectories of BranchSBM and single-branch SBM over the $t \in [t_1, t_2]$ on validation samples.

perturbation. The endpoints for both perturbation experiments are clustered largely on variance in the first two or three principal components (PCs), where PC1 captures the divergence from the control state to the perturbed clusters and PC2 and higher captures the divergence between clusters in the perturbed population, where cluster 0 is closest to the control state along PC2 and cluster 1 and 2 are farther from the control. From Figure 8A and B, we can conclude that single-branch SBM is not expressive enough to capture the complexities of higher-dimensional PCs and follows the most obvious trajectory from the control to cluster 0, resulting in an inaccurate representation of the perturbed cell population.

In contrast, we demonstrate that BranchSBM is capable of stimulating trajectories to both clusters in the population perturbed with Clonidine (Figure 8B) and three clusters in the population perturbed with Trametinib (Figure 8D), generating branched distributions that accurately capture the location and spread of the perturbed cell distribution in the dataset.

D.2 Effect of Final Joint Training on Losses

In Table 7, we show the final losses after convergence, summed across each branch and averaged over the batch size, for Stage 3 training of only the growth networks and Stage 4 joint training of the flow and growth networks discussed in Section 4. All losses are calculated exactly as shown in Section 4. Crucially, we find that the final joint training stage refines the parameters of both the flow and growth networks simultaneously to minimize the energy loss $\mathcal{L}_{\text{energy}}(\theta, \phi)$ (14) while ensuring the growth parameters maintain minimal losses across $\mathcal{L}_{\text{match}}(\phi)$ (15) and $\mathcal{L}_{\text{mass}}(\phi)$ (16) for all experiments. This indicates that jointly optimizing both the drift and growth dynamics leads to further refinement towards modeling the optimal branching trajectories in the data.

E Experiment Details

E.1 Multi-Stage Training

To ensure stable training while incorporating all loss functions, we introduce a multi-stage training approach (Algorithm 1).

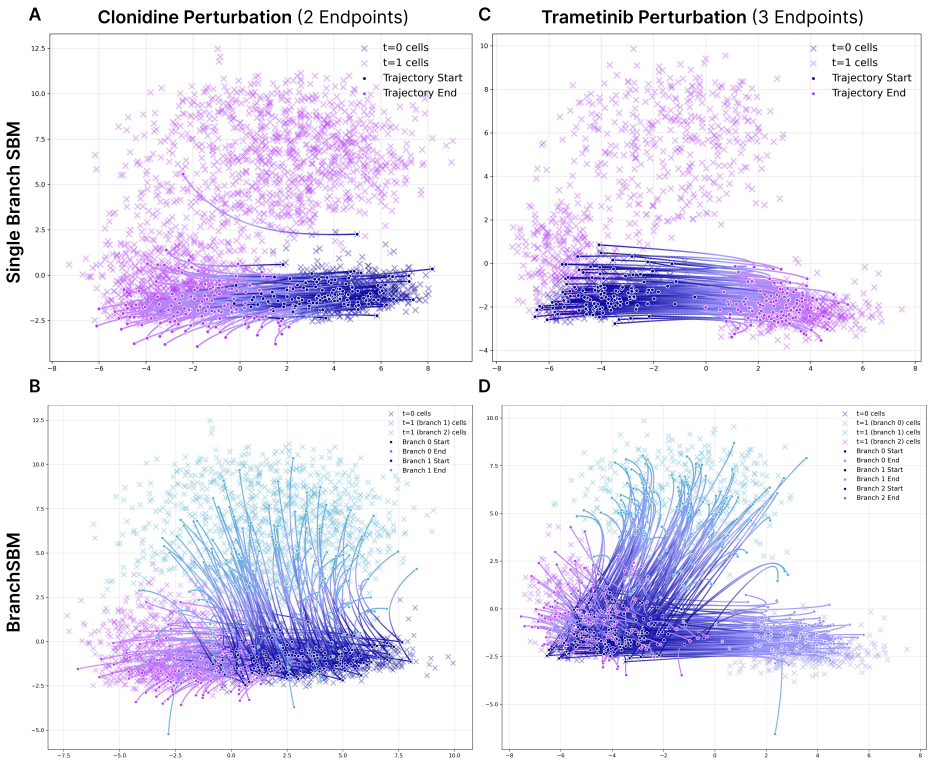

Figure 8: **Comparison of BranchSBM to Single-Branch SBM for Perturbation Modelling. (A, B)** Clonidine perturbation trajectories with two target clusters generated by **(A)** single-branch SBM and **(B)** BranchSBM from the validation data. **(C, D)** Trametinib perturbation trajectories with three target clusters generated by **(C)** single-branch SBM and **(D)** BranchSBM. In both experiments, single-branch SBM only generated states in cluster 0 and not cluster 1 or 2, whereas BranchSBM reconstructed all perturbed clusters via branched trajectories.

Table 7: **Validation Losses for Stage 3 and 4 Training Across Experiments.** Losses are summed across both branches and averaged over batch size. The final Stage 4 joint training stage that refines both the flow and growth networks simultaneously minimize the energy loss $\mathcal{L}_{\text{energy}}(\theta, \phi)$ (14) from Stage 3 while ensuring the growth parameters maintain minimal losses across $\mathcal{L}_{\text{match}}(\phi)$ (15) and $\mathcal{L}_{\text{mass}}(\phi)$ (16)

Experiment	**Stage 3**			**Stage 4**		
	$\mathcal{L}_{\text{energy}}(\theta, \phi)$	$\mathcal{L}_{\text{mass}}(\theta, \phi)$	$\mathcal{L}_{\text{match}}(\theta, \phi)$	$\mathcal{L}_{\text{energy}}(\theta, \phi)$	$\mathcal{L}_{\text{mass}}(\theta, \phi)$	$\mathcal{L}_{\text{match}}(\theta, \phi)$
LiDAR	1.276	3.0×10^{-5}	0.007	0.768	2.0×10^{-5}	0.102
Mouse Hematopoiesis	2.209	1.2×10^{-4}	0.054	1.918	5.0×10^{-5}	0.076
Chlonidine Perturbation	36.469	0.030	0.109	25.798	0.053	0.153
Trametinib Perturbation	35.834	0.023	0.078	32.843	0.017	0.056

Stage 1 First, we train a neural interpolant $\varphi_{t,\eta}(\mathbf{x}_0, \mathbf{x}_{t,k}) : \mathbb{R}^d \times \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ that takes the endpoints of a branched coupling and defines the optimal interpolating state X_t at time t by minimizing the energy function $\mathcal{L}_{\text{traj}}(\eta)$ in (12). This is used to calculate the optimal conditional velocity $\dot{\mathbf{x}}_{t,\eta,k}$ that preserves the endpoints $X_0 = \mathbf{x}_0$ and $X_{1,k} = \mathbf{x}_{1,k}$ for the flow matching objective in Stage 2.

Stage 2 Next, we train a set of flow networks $\{u^{\theta}_{t,k}\}_{k=0}^{K}$ that *generate* the optimal interpolating trajectories for each branch with the conditional flow matching objective $\mathcal{L}_{\text{flow}}$ in (13).

Stage 3 We freeze the parameters of the flow networks and only train the growth networks $\{g^{\phi}_{t,k}\}_{k=0}^{K}$ by minimizing $\mathcal{L}_{\text{growth}}$ in (17).

Stage 4 Finally, we unfreeze the parameters of both the flow and growth networks and jointly train $\{u^{\theta}_{t,k}, g^{\phi}_{t,k}\}_{k=0}^{K}$ by minimizing the growth loss $\mathcal{L}_{\text{growth}}$ in (17) from Stage 3 in addition to the distribution reconstruction loss $\mathcal{L}_{\text{recons}}$ in (19).

E.2 General Training Details

Leiden Clustering To identify branch endpoints in the dataset, we apply an automated clustering pipeline to the final time point ($t = 7$), where the differentiated cell-states are most clearly defined. First, we construct a 20-nearest-neighbor (kNN) graph on the final-timepoint embeddings and run Leiden community detection using the `ModularityVertexPartition` objective. Leiden clustering is particularly suited and the state-of-the-art method for clustering single-cell data, given that it is robust to noise, can handle heterogeneous cluster sizes and shapes, and guarantees well-connected clusters, all while being fully unsupervised and automatic. This gives us stable and biologically meaningful terminal clusters compared to alternative methods such as K-means clustering, which assumes spherical clusters and struggles with sparse high-dimensional manifolds. After initial clustering, we merge any cluster with fewer than `min_cells` cells into the nearest large cluster based on Euclidean centroid distance.

Model Architecture We parameterized the branched trajectory $\varphi_{t,\eta}(\mathbf{x}_0, \mathbf{x}_1)$ with a 3-layer MLP with Scaled Exponential Linear Unit (SELU) activations. The endpoint pair $(\mathbf{x}_0, \mathbf{x}_1)$ and the time step t are concatenated into a single $(2d + 1)$-dimensional vector and used as input to the model. Similarly, we parameterize each flow network $u^{\theta}_{t,k}(\mathbf{x}_t)$ and growth network $g^{\theta}_{t,k}(\mathbf{x}_t)$ with the same 3-layer MLP and SELU activations but takes the interpolating state $\mathbf{x}_t$ and time t concatenated into a $(d + 1)$-dimensional vector.

To ensure that the growth rates of all secondary branches are non-negative (i.e. for all $k \in \{1, \ldots, K\}$, $g_{t,k}(X_t) \geq 0$), we apply an additional softplus activation to the output of the 3-layer MLP in the growth networks, defined as $\text{softplus}(\cdot) = \log(1 + \exp(\cdot))$, which is smooth function that transforms negative values to be positive near 0. For the growth network of the primary branch ($k = 0$), we allow for both positive and negative growth, as all mass starts at the primary branch and flows into the secondary branches, but the primary branch itself can grow as well, depending on whether mass is conserved.

State Cost V_t Depending on the dimensionality of the data type, we set the state cost $V_t(X_t) : \mathbb{R}^d \to [0, +\infty]$ to be either the LAND or RBF metric discussed in Appendix A.2. For the experiments on LiDAR ($d = 3$) and Mouse Hematopoiesis scRNA-seq ($d = 2$) data, we used the LAND metric (33) with hyperparameters $\sigma = 0.125$ and $\varepsilon = 0.001$.

To avoid the task of setting a suitable variance σ for the high-dimensional gene expression space $d \in \{50, 100, 150\}$, we use the RBF metric (37) that learns parameters to ensure the regions within the data manifold have low cost and regions far from the data manifold have high cost. Using the training scheme in Kapuśniak et al. (2024), we identified N_c cluster centers with k-means clustering, and trained the parameters $\{\omega_{j,n}\}_{n=1}^{N_c}$ by minimizing $\mathcal{L}_{\text{RBF}}$ (38) on the training data. We found that setting the number of cluster centers N_c too low resulted in non-decreasing $\mathcal{L}_{\text{RBF}}$ and that increasing N_c for higher dimensions enabled more effective training. Furthermore, we found that modeling higher-dimensional PCs required setting a larger κ, which determines the *spread* of the RBF kernel around each cluster center. The specific values for N_c and κ depending on the dimension of principal components are provided in Table 11.

Optimal Transport Coupling Since our experiments consist of unpaired initial and target distributions and we seek to minimize the energy of the interpolating bridge, we define pairings $(\mathbf{x}_0, \mathbf{x}_{1,k})$ using the optimal transport plan $\pi^{\star}_{0,1,k}$ that minimizes the distance between the initial distribution π_0 and each target distribution $\pi_{1,k}$ in probability space. Specifically, we define $\pi^{\star}_{0,1,k}$ as the

2-Wasserstein transport plan (Villani et al., 2008) between π_0 and $\pi_{1,k}$ defined as

$$\pi^{\star}_{0,1,k} = \arg\min_{\pi_{0,1,k}\in\Pi} \int_{\pi_0\otimes\pi_{1,k}} \|\mathbf{x}_0 - \mathbf{x}_{1,k}\|_2^2 d\pi(\mathbf{x}_0, \mathbf{x}_{1,k}) \tag{92}$$

where $\pi_0 \otimes \pi_{1,k}$ is the set of all possible couplings between the endpoint distributions. For each of the branches, the dataset was paired such that $(\mathbf{x}_0, \mathbf{x}_{1,k}) \sim \pi^{\star}_{0,1,k}$.

Training We train each stage for a maximum of 100 epochs. For Stage 1, we used the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 1.0×10^{-4} to train $\varphi_{t,\eta}(\mathbf{x}_0, \mathbf{x}_1)$. For Stage 2, 3, and 4, we used the AdamW optimizer (Loshchilov & Hutter, 2017) with weight decay 1.0×10^{-5} and learning rate 1.0×10^{-3} to train each flow network $u^{\theta}_{t,k}$ and growth network $g^{\phi}_{t,k}$. All experiments were performed on one NVIDIA A100 GPU. We trained on the LiDAR and mouse hematopoiesis data with a batch size of 128 and the Clonidine and Trametinib perturbation data with a batch size of 128, each divided with a 0.9/0.1 train/validation split. All hyperparameters across experiments are provided in Table 11.

Computational Overhead Although we train $K+1$ flow and growth networks, the overall training time remains comparable to that of single-branch SBM, since the networks for each branch is trained only on the subset of data corresponding to its respective target distribution. While the method does incur higher space complexity, we find that simple MLP architectures suffice for strong performance, suggesting that scalability is not a major concern. BranchSBM also significantly reduces inference time, as predicting branching population dynamics requires simulating only a single sample from the initial distribution, unlike other models that require simulating large batches of samples.

E.3 LiDAR Experiment Details

LiDAR Data We used the same LiDAR manifold from Liu et al. (2023a); Kapuśniak et al. (2024). The data is a collection of three-dimensional point clouds within 10 unit cubes $[-5, 5]^3 \subset \mathbb{R}^3$ that span the surface of Mount Rainier. Given any point $\mathbf{x} \in \mathbb{R}^3$ in the three-dimensional space, we project it onto the LiDAR manifold by identifying the k-nearest neighbors $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ and fitting a 2D tangent plane to the set of neighbors

$$\arg\min_{a,b,c} \frac{1}{k}\sum_{i=1}^{k} \exp\left(\frac{-\|\mathbf{x}-\mathbf{x}_i\|}{\tau}\right)(a\mathbf{x}_i^{(x)} + b\mathbf{x}_i^{(y)} + c - \mathbf{x}_i^{(z)})^2 \tag{93}$$

where $\tau = 0.001$ following Liu et al. (2023a). Then, we solve for the tangent plane $ax + by + c = z$ using the Moore-Penrose pseudoinverse from $k = 20$ neighbors. From the tangent plane, we can project any point $\mathbf{x}$ to the LiDAR manifold with the function $\pi(\mathbf{x})$ defined as

$$\pi(\mathbf{x}) = \mathbf{x} - \left(\frac{\mathbf{x}^\top\mathbf{v} + c}{\|\mathbf{v}\|^2}\right)\mathbf{v}, \quad \text{where} \quad \mathbf{v} = \begin{bmatrix} a \\ b \\ -1 \end{bmatrix} \tag{94}$$

Synthetic Distributions To reformulate the experiment in Liu et al. (2023a) as a branching problem, we define a single initial distribution and two divergent target distributions. Specifically, we define a single initial distribution $\pi_0 = \mathcal{N}(\mu_0, \sigma_0)$ as a mixture of four Gaussians and two target distributions $\pi_{1,0}, \pi_{1,1}$ on either side of the mountain, both as mixtures of three Gaussians. The exact parameters of each Gaussian are given in Table 8.

We sample a total of 5000 points i.i.d. from each of the Gaussian mixtures $\{\mathbf{x}_0^i\}_{i=1}^{5000} \sim \pi_0$, $\{\mathbf{x}_{1,0}^i\}_{i=1}^{5000} \sim \pi_{1,0}$, and $\{\mathbf{x}_{1,1}^i\}_{i=1}^{5000} \sim \pi_{1,1}$. The data points are projected to the LiDAR manifold with the projection function $\pi(\mathbf{x})$ in (94).

Evaluation Metrics To determine how closely the simulated trajectories match the ground truth trajectories, we compute the Wasserstein-1 ($\mathcal{W}_1$) and Wasserstein-2 ($\mathcal{W}_2$) distances defined as

$$\mathcal{W}_1 = \left(\min_{\pi\in\Pi(p,q)} \int \|\mathbf{x}-\mathbf{y}\|_2 d\pi(\mathbf{x},\mathbf{y})\right) \tag{95}$$

$$\mathcal{W}_2 = \left(\min_{\pi\in\Pi(p,q)} \int \|\mathbf{x}-\mathbf{y}\|_2^2 d\pi(\mathbf{x},\mathbf{y})\right)^{1/2} \tag{96}$$

Table 8: **Synthetic Gaussian mixture distribution parameters for LiDAR experiment.** 5000 datapoints are drawn i.i.d. from each of the Gaussian mixtures and paired randomly $(\mathbf{x}_0, \mathbf{x}_{1,0}, \mathbf{x}_{1,1})$ to define the training dataset. A visualization of the training data on the LiDAR manifold is provided in Figure 3.

	Distribution	μ	σ
π_0	$\mathcal{N}(\mu_0, \sigma_0)$	$(-4.5, -4.0, 0.5), (-4.2, -3.5, 0.5), (-4.0, -3.0, 0.5), (-3.75, -2.5, 0.5)$	0.02
$\pi_{1,0}$	$\mathcal{N}(\mu_{1,0}, \sigma_{1,0})$	$(-2.5, -0.25, 0.5), (-2.25, 0.675, 0.5), (-2, 1.5, 0.5)$	0.03
$\pi_{1,1}$	$\mathcal{N}(\mu_{1,1}, \sigma_{1,1})$	$(2, -2, 0.5), (2.6, -1.25, 0.5), (3.2, -0.5, 0.5)$	0.03

where p denotes the ground truth distribution and q denotes the predicted distribution. After training the velocity and growth networks on samples from the initial Gaussian mixture π_0 and target Gaussian mixtures $\pi_{1,0}$ and $\pi_{1,1}$, we evaluate $\mathcal{W}_1$ and $\mathcal{W}_2$ of the reconstructed distribution simulated from the validation points in the initial distribution π_0 against the true distribution at $t = 1$.

E.4 Differentiating Single-Cell Experiment Details

Mouse Hematopoiesis scRNA-seq Data We used the mouse hematopoiesis dataset from Zhang et al. (2025c) consisting of three timesteps t_0, t_1, t_2, with a total of 1429 cells at t_0, 3781 cells from t_1, and 5788 cells from t_2. The data points at t_0 form a homogeneous cluster, while the data points at t_2 are clearly divided into two distinct cell fates. We performed k-means clustering with $k = 2$ clusters to create branching on the cells at t_2, splitting the cells into two clusters: endpoint 0 with 2902 cells and endpoint 1 with 2886 cells. Since the two endpoints are near equal in ratio, we set the final weights of both endpoints as 0.5 (i.e. $w_{1,0} = w_{1,1} = 0.5$). To match the size of the t_0 samples, we randomly sampled 1429 samples from each of these two clusters and used them as the endpoints for branches 0 and 1, respectively. Training and validation follow a $0.9/0.1$ split ratio.

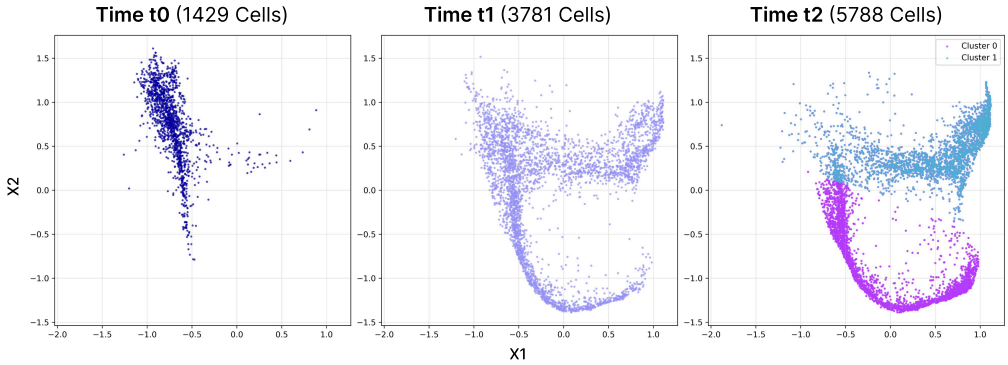

Figure 9: **Mouse Hematopoiesis Single-Cell RNA Sequencing Data Plotted by Time Point.** Real scRNA-seq data is projected to 2D force-directed SPRING plots (Sha et al., 2023; Weinreb et al., 2020; Zhang et al., 2025c). There is a clear divergence of cell fate between times t_0 (left), t_1 (middle), and t_2 (right) from the initial homogeneous progenitor cells into two distinct cell fates (shown in pink and purple in the t_2 plot). Cells at time t_2 are clustered into endpoint 0 (pink; 2902 cells) and endpoint 1 (turquoise; 2886 cells).

Pancreatic β-Cell Differentiation Data The pancreatic β-cell dataset from Veres et al. (2019) contains $51,274$ cells over eight time points that evolve from human pluripotent stem cells to pancreatic β-like cells. Following (Zhang et al., 2025d), the gene expression representations are projected to 30-dimensional PCA space. After Leiden clustering with $k = 20$ and `min_cells` $=$ 100, we obtain 11 clusters for $K = 11$ branches.

Evaluation Metrics To determine how closely the reconstructed distributions along the trajectory match the ground truth distributions, we compute the 1-Wasserstein ($\mathcal{W}_1$) (95) and 2-Wasserstein ($\mathcal{W}_2$) (96) distances similar to the LiDAR experiment. After training the velocity and growth networks

on samples from the initial cell distribution π_{t_1} and differentiated target cell distributions $\pi_{t_2,0}$ and $\pi_{t_2,1}$, we evaluate $\mathcal{W}_1$ and $\mathcal{W}_2$ between the reconstructed branched distributions at the intermediate time t_1 ($p_{t_1,0}$ and $p_{t_1,1}$) and the target distributions at the final time t_2 ($p_{t_2,0}$ and $p_{t_2,1}$) simulated from the validation points in the initial distribution π_0 and the true distributions π_{t_1} and π_{t_2}.

E.5 Cell-State Perturbation Modeling Experiment Details

Tahoe Single-Cell Perturbation Data The Tahoe-100M dataset consists of 50 cell lines and over 1000 different drug-dose conditions (Zhang et al., 2025a). For this experiment, we extract the data for a single cell line (A-549) under two drug perturbation conditions selected based on cell abundance and response diversity.

Clonidine at 5 μL was selected first due to having the largest number of cells at this dosage, while Trametinib was chosen as the second drug based on its second-highest cell count under the same condition. For both drugs, we selected the top 2000 highly variable genes (HVGs) based on normalized expression and projected the data into a 50-dimensional PCA space, which captured approximately 38% of the total variance in both cases.

We applied K-nearest neighbor ($K = 50$) and conducted Leiden clustering separately for drugged versus DMSO control conditions. The most abundant DMSO cluster was selected as the initial state ($t = 0$). For Clonidine, we identified two clusters that were most distinct from the DMSO control along PC1 and PC2, respectively. These were selected as two distinct endpoints for branches 1 and 2 at $t = 1$. We applied centroid-based sampling to obtain balanced training sets of 1033 cells per cluster (Figure 10).

For Trametinib, we extended the branching up to three endpoints. From its Leiden clustering results, we identified three clusters that were the most divergent from the DMSO control clusters along PC1, PC2, and PC3, respectively. All selected clusters contained at least 100 cells and were subsampled to 381 cells each for branch training. The remaining cells were clustered with K-means into three (Clonidine) or four (Trametinib) groups to construct the metrics dataset. The training and validation dataset split followed a 0.9/0.1 ratio. The final visualization utilized the first two principal components.

Table 9: **Training cluster cell counts for perturbation experiments.**

	Clonidine		**Trametinib**		
	Cluster 0	Cluster 1	Cluster 0	Cluster 1	Cluster 2
Original Cell Count	1675	1033	1622	686	381
Initial Weight $w_{0,k}$	1.0	0	1.0	0	0
Target Weight $w_{1,k}$	0.619	0.381	0.603	0.255	0.142

Evaluation Metrics To quantify the alignment of the reconstructed and ground-truth distributions for the cell-state perturbation experiment on principal component (PCs) dimensions $d \in \{50, 100, 150\}$, we calculate the Maximum Mean Discrepancy with the RBF kernel (RBF-MMD) on all predicted PCs and the 1-Wasserstein ($\mathcal{W}_1$) (95) and 2-Wasserstein ($\mathcal{W}_2$) (96) distances on the top-2 PCs.

This choice is made because the Wasserstein distance between empirical measures (i.e., discrete point clouds) is known to converge to the true Wasserstein distance between the underlying distributions at a rate that scales as $\varepsilon = \mathcal{O}(n^{-1/d})$ for some constant C, where n is the number of samples and d is the dimensionality of the state space. Therefore, an exponential number of samples is required to achieve the same estimation accuracy in high dimensions.

Computing the Wasserstein distance on the top-2 principal components provides a statistically stable and computationally tractable approximation that preserves the major axes of variation in the data. In addition, we evaluate the RBF-MMD metric on *all simulated PCs*, ensuring that higher-dimensional reconstruction accuracy is still quantitatively assessed.

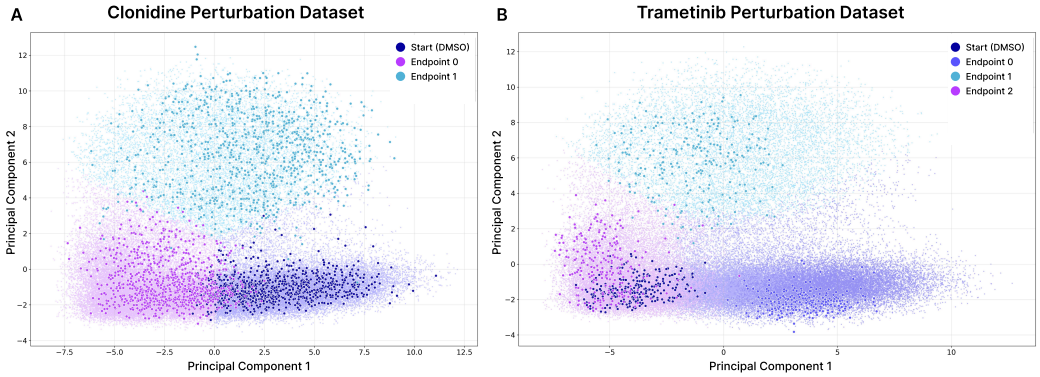

Figure 10: **Clustered Cell-State Perturbation Data from the Tahoe-100M Dataset.** PCA was conducted on all cells for the control DMSO-treated and two drug-treated populations, and clustered. Plots show divergence along the top 2 PCs. **(A)** Clonidine-treated cells ($5\mu M$) are plotted in pink (endpoint 0), and turquoise (endpoint 1), and the distribution of control cells is plotted in navy. **(B)** Trametinib-treated cells ($5\mu M$) are plotted in purple (endpoint 0), turquoise (endpoint 1), and pink (endpoint 2), and the distribution of control cells is plotted in navy.

Given the predicted distribution p and true distribution q and n samples from each distribution $\{\mathbf{x}_i \sim p\}_{i=1}^n$ and $\{\mathbf{y}_i \sim q\}_{i=1}^n$, the RBF-MMD between p and q is calculated as

$$\text{MMD}(p,q) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} k_{\text{mix}}(\mathbf{x}_i,\mathbf{x}_j) + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} k_{\text{mix}}(\mathbf{y}_i,\mathbf{y}_j) - \frac{2}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} k_{\text{mix}}(\mathbf{x}_i,\mathbf{y}_j) \tag{97}$$

where $k_{\text{mix}}(\cdot,\cdot)$ is a mixture of RBF kernel functions defined as

$$k_{\text{mix}}(\mathbf{x},y) = \frac{1}{|\Sigma|}\sum_{\sigma\in\Sigma}\exp\left(-\frac{\|\mathbf{x}-y\|^2}{2\sigma^2}\right) \tag{98}$$

where $\Sigma = \{0.01, 0.1, 1, 10, 100\}$ is the set of values that determine how much the distances between pairs of points are scaled when computing the overall discrepancy. The equations for 1-Wasserstein ($\mathcal{W}_1$) and 2-Wasserstein ($\mathcal{W}_2$) distances are provided in (95) and (96) respectively.

E.6 Computational Costs

For each experiment, we report the GPU hours on a single NVIDIA A100 GPU, demonstrating that only a minor increase in compute time is required for increasing the number of branches.

Table 10: **Total training time for each experiment on a NVIDIA A100 GPU for different dimensions and number of branches K.**

Experiment (Branches)	Total Training Time (min)
Mouse Hematopoesis ($K=1$)	13m 4s
Mouse Hematopoiesis ($K=2$)	14m 3s
Clonidine 50D ($K=1$)	17m 22s
Clonidine 50D ($K=2$)	22m 50s
Clonidine 100D ($K=2$)	23m 3s
Clonidine 150D ($K=2$)	18m 31s
Trametinib ($K=1$)	4m 52s
Trametinib ($K=3$)	6m 44s

E.7 Hyperparameter Selection and Discussion

In this section, we present the hyperparameters used in each experiment. While the model architecture remained largely the same across experiments, we increased the hidden dimension to 1024 for dimensions $d \in \{50, 100, 150\}$. For low-dimensional data, we found that increasing model complexity underperforms in comparison to lower hidden dimensions, and we established that a hidden dimension of 64 achieves relatively optimal performance for $d \in \{2, 3\}$. While beyond the scope of this study, we believe that further exploration of diverse model architectures and hyperparameter tuning could improve the performance of BranchSBM. Exploration of diverse task-dependent state costs for novel applications is another exciting extension of our work.

Table 11: **Hyperparameter settings for different datasets.** The Clonidine perturbation experiment is split into three columns for each of the three dimensions of principal components (PCs) used $d \in \{50, 100, 150\}$.

Parameter	**Dataset**					
	LiDAR	Mouse Hematopoesis scRNA	Clonidine Perturbation			Trametinib Perturbation
			50PCs	100PCs	150PCs	
branches	2	2		2		3
data dimension	3	2	50	100	150	50
batch size	128	128		32		32
λ_{energy}	1.0	1.0		1.0		1.0
λ_{mass}	100	100		100		100
λ_{match}	1.0×10^3	1.0×10^3		1.0×10^3		1.0×10^3
λ_{recons}	1.0	1.0		1.0		1.0
λ_{growth}	0.01	0.01		0.01		0.01
V_t	LAND	LAND		RBF		RBF
RBF N_c	-	-	150	300	300	150
RBF κ	-	-	1.5	2.0	3.0	1.5
hidden dimension	64	64		1024		1024
lr $\varphi_{t,\eta}$	1.0×10^{-4}	1.0×10^{-4}		1.0×10^{-4}		1.0×10^{-4}
lr u_t^θ	1.0×10^{-3}	1.0×10^{-3}		1.0×10^{-3}		1.0×10^{-3}
lr g_t^ϕ	1.0×10^{-3}	1.0×10^{-3}		1.0×10^{-3}		1.0×10^{-3}

F TRAINING ALGORITHM

Here, we provide the pseudocode for BranchSBM's multi-stage training algorithm for stable optimization of the velocity and growth networks over the K branched trajectories.

Algorithm 1 Multi-Stage Training of **BranchSBM**

Stage 1: Learning the Branched Neural Interpolants
while Training **do**
  $\forall k, (\mathbf{x}_0, \mathbf{x}_{1,k}) \sim \pi^\star_{0,1,k}$, $t \sim \mathcal{U}(0,1)$
  for $k = 0$ to K **do**
    $\mathbf{x}_{t,\eta,k} \leftarrow (1-t)\mathbf{x}_0 + t\mathbf{x}_{1,k} + t(1-t)\varphi_{t,\eta}(\mathbf{x}_0, \mathbf{x}_{1,k})$
    $\dot{\mathbf{x}}_{t,\eta,k} \leftarrow \mathbf{x}_1 - \mathbf{x}_0 + t(1-t)\dot{\varphi}_{t,\eta}(\mathbf{x}_0, \mathbf{x}_{1,k}) + (1-2t)\varphi_{t,\eta}(\mathbf{x}_0, \mathbf{x}_{1,k})$
    Compute $V_t(\mathbf{x}_{t,\eta,k})$ given the task-specific definition
    $\mathcal{L}_{\text{traj}}(\eta) \leftarrow \int_0^1 \left[\frac{1}{2}\|\dot{\mathbf{x}}_{t,\eta,k}\|^2 + V_t(\mathbf{x}_{t,\eta,k})\right] dt$
    Update $\varphi_{t,\eta}$ using gradient $\nabla_\eta \mathcal{L}_{\text{traj}}(\eta)$
  end for
end while
Stage 2: Initial Training of Velocity Networks
while Training **do**
  Initialize $K+1$ flow networks $\{u^\theta_{t,k}(X_t)\}_{k=0}^K$
  for $k = 0$ to K **do**
    Calculate $\mathbf{x}_{t,\eta,k}$ and $\dot{\mathbf{x}}_{t,\eta,k}$ with the trained network $\varphi^\star_{t,\eta}(\mathbf{x}_0, \mathbf{x}_{1,k})$
    $\mathcal{L}_{\text{flow}}(\theta) \leftarrow \|\dot{\mathbf{x}}_{t,\eta,k} - u^\theta_{t,k}(\mathbf{x}_{t,\eta,k})\|_2^2$
    Update $u^\theta_{t,k}$ using gradient $\nabla_\theta \mathcal{L}_{\text{flow}}(\theta)$
  end for
end while
Stage 3: Initial Training of Growth Networks
while Training **do**
  Freeze parameters of flow networks and initialize $K+1$ growth networks $\{g^\phi_{t,k}(X_t)\}_{k=0}^K$
  for $t = 0$ to 1 **do**
    for $k = 0$ to K **do**
      $\mathbf{x}_{t,k} \leftarrow \int_0^t u^\theta_{s,k}(\mathbf{x}_{s,k})ds$
      if $k = 0$ **then**
        $w^\phi_{t,k} \leftarrow 1 + \int_0^t g^\phi_{t,k}(\mathbf{x}_{s,k})ds$
      else
        $w^\phi_{t,k} \leftarrow \int_0^t g^\phi_{t,k}(\mathbf{x}_{s,k})ds$
      end if
      $\mathcal{L}_{\text{energy}}(\phi) \leftarrow \mathcal{L}_{\text{energy}}(\phi) + \int_t^{t+\Delta t} \left[\frac{1}{2}\|u^\theta_{t,k}\|^2 + V_t(\mathbf{x}_{t,k})\right] w^\phi_{t,k}$
    end for
    $\mathcal{L}_{\text{mass}} \leftarrow \left(\sum_{k=0}^K w^\phi_{t,k} - w^{\text{total}}\right)^2$
  end for
  $\mathcal{L}_{\text{match}} \leftarrow \sum_{k=0}^K \left(w^\phi_{1,k}(\mathbf{x}_{1,k}) - w^\star_{1,k}\right)^2$
  $\mathcal{L}_{\text{recons}}(\theta) \leftarrow \sum_{k=0}^K \sum_{\mathbf{x}_{1,k} \in \mathcal{N}_n(\mathbf{x}_{1,k})} \max(0, \|\tilde{\mathbf{x}}_{1,k} - \mathbf{x}_{1,k}\|_2 - \epsilon)$
  $\mathcal{L}_{\text{growth}}(\phi) \leftarrow \lambda_{\text{energy}}\mathcal{L}_{\text{energy}}(\theta,\phi) + \lambda_{\text{match}}\mathcal{L}_{\text{match}}(\phi) + \lambda_{\text{mass}}\mathcal{L}_{\text{mass}}(\phi) + \lambda_{\text{growth}} \sum_{k=0}^K \|g^\phi_{t,k}\|_2^2$
  Update $g^\phi_{t,k}$ using gradient $\nabla_\phi \mathcal{L}_{\text{growth}}(\phi)$
end while
Stage 4: Final Joint Training
while Training **do**
  Unfreeze parameters of flow networks $\{u^\theta_{t,k}(X_t)\}_{k=0}^K$
  Repeat steps of Stage 3 and calculate $\mathcal{L}_{\text{joint}}(\theta,\phi) \leftarrow \mathcal{L}_{\text{growth}}(\theta,\phi) + \mathcal{L}_{\text{recons}}(\theta)$
  Jointly update $u^\theta_{t,k}$ and $g^\phi_{t,k}$ for all branches using gradients $\nabla_\theta \mathcal{L}_{\text{joint}}(\theta,\phi)$ and $\nabla_\phi \mathcal{L}_{\text{joint}}(\theta,\phi)$
end while

G USE OF LARGE LANGUAGE MODELS (LLMS)

We acknowledge the use of large language models (LLMs) to assist in polishing and editing parts of this manuscript. LLMs were used to refine phrasing, improve clarity, and ensure consistency of style across sections. All technical content, experiments, analyses, and conclusions were developed by the authors, with LLM support limited to language refinement and editorial improvements.