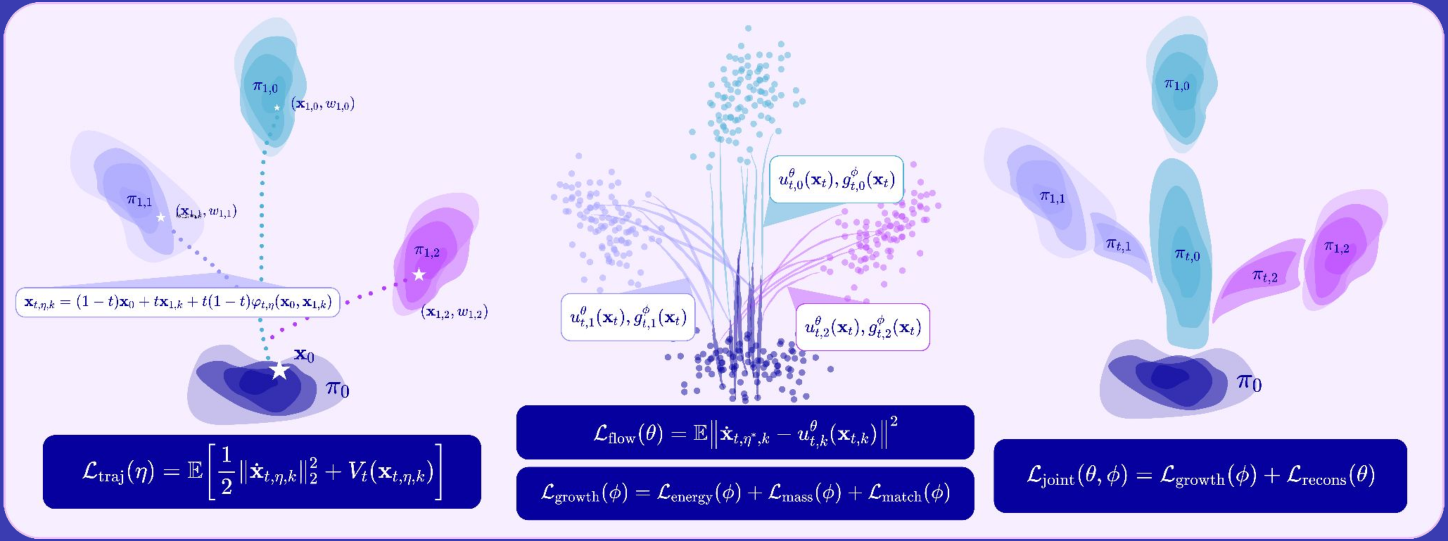
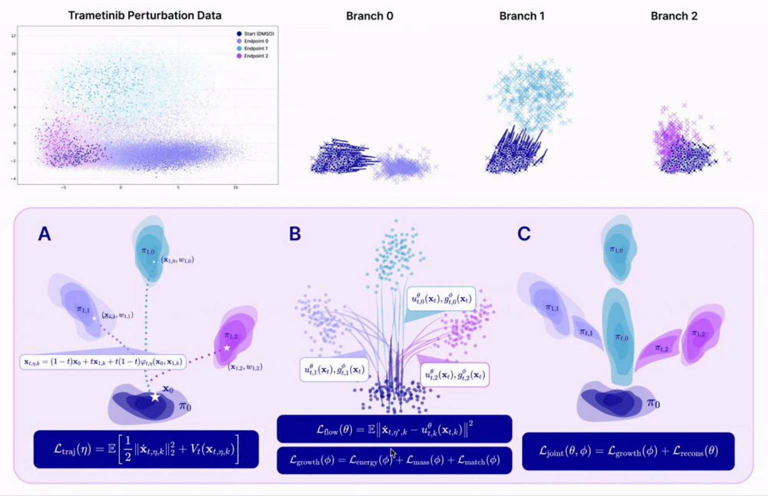
Branched Schrödinger Bridge Matching

Sophia Tang • Yinuo Zhang • Alexander Tong • Pranam Chatterjee



SS Reading Group | Jul 28, 2025

Agenda



1. Schrödinger Bridge Theory
2. Motivation
3. Branched Schrödinger Bridge Problem
4. Multi-Stage Training of BranchSBM
5. Conclusion + Q&A

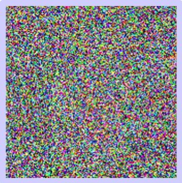
Schrödinger Bridge Theory

Flow Matching $\rightarrow$ Bridge Matching

Mapping between two empirically defined data distributions

Flow Matching

$$dX_t = f_t^{\text{base}}(X_t)dt + \sigma dB_t$$

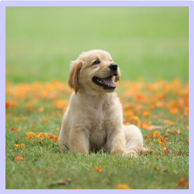


π_1

π_0

Bridge Matching

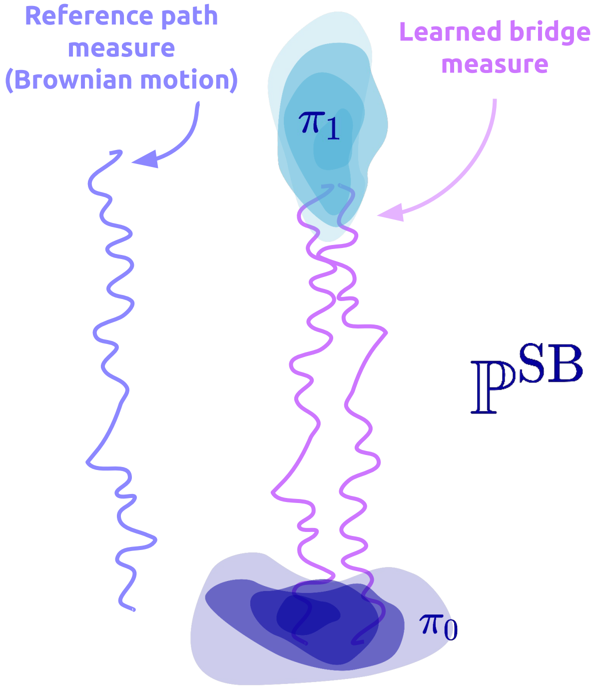
$$dX_t = [f_t^{\text{base}}(X_t) + u_t(X_t)]dt + \sigma dB_t$$



Nonlinear forward drift

Schrödinger Bridge Problem

Generating an optimal path measure between distinct initial and final distributions



$$dX_t = f_t^{\mathrm{base}}(X_t)dt + \sigma dB_t$$

Reference path measure (Brownian motion)

Match the defined boundary distributions

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

Learned bridge measure

$$dX_t = [f_t^{\mathrm{base}}(X_t) + u_t(X_t)]dt + \sigma dB_t$$

Nonlinear forward drift

Dynamic Schrödinger Bridge Problem

Defining the optimal drift field of the Schrodinger bridge path

Minimize kinetic energy from x0 to x1

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

?

What if you want to learn the lowest energy path on a **non-Euclidean manifold**?

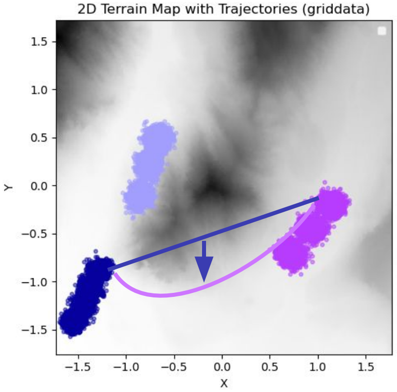
Generalized Schrödinger Bridge Problem

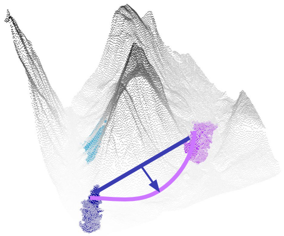
Learning to minimize the task-dependent potential energy of the path

Minimize kinetic energy or magnitude of velocity from x0 to x1

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

Minimize task-dependent potential energy of the path by defining non-linear path

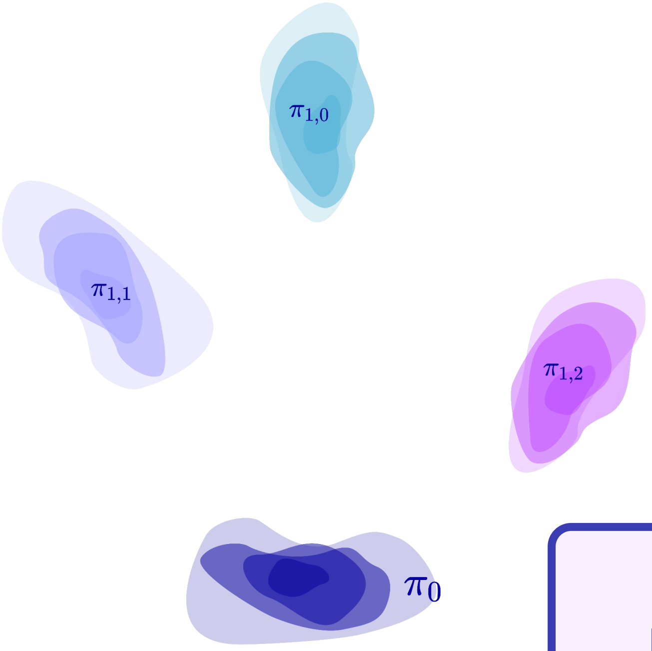




Liu, Guan-Horng, et al. "Deep generalized schrödinger bridge."

Motivation: Modeling Branched Bridges?

How do we model dynamical systems with diverging trajectories and endpoints?



1. **Single-path SBM frameworks require simulating a large batch of particles to recover a multi-modal terminal distribution.** This requires expensive batched simulation and may not capture the full population dynamics.

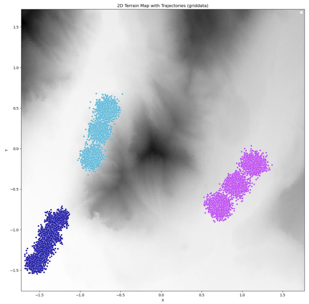
2. **Branching dynamics is governed by energy-minimizing trajectories.** To define more complex systems where the optimal dynamics cannot be accurately captured by minimizing the standard squared Euclidean cost in entropic OT, the Generalized Schrödinger Bridge (GSB) problem introduces an additional nonlinear state-cost.

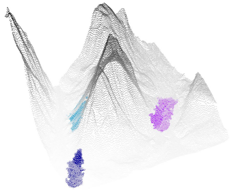
What if we could simulate **energy-minimizing branching population dynamics** by simulating **only one sample** from the initial distribution?

Motivation: Modeling Branched Bridges

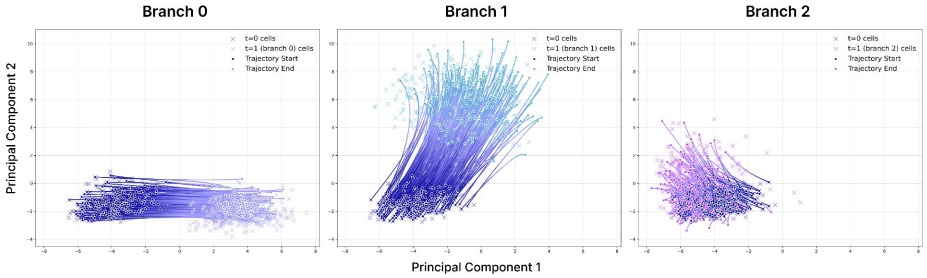
How do we model dynamical systems with diverging trajectories and endpoints?

Multi-Path Surface Navigation

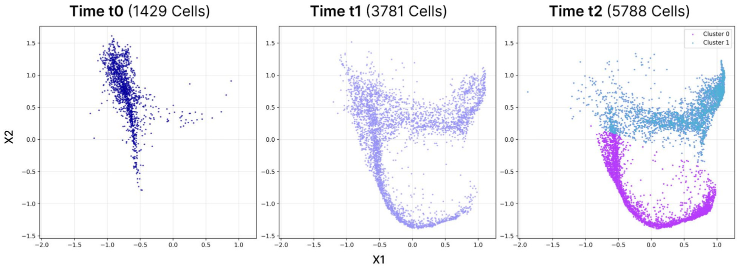




Diverging Cellular Responses to Drug Perturbation



Cell Fate Differentiation



Branched Generalized Schrödinger Bridges

How do we model dynamical systems with diverging trajectories and endpoints?

Generalized Schrödinger Bridge Problem

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

Branched Generalized Schrödinger Bridges

How do we model dynamical systems with diverging trajectories and endpoints?

Generalized Schrödinger Bridge Problem

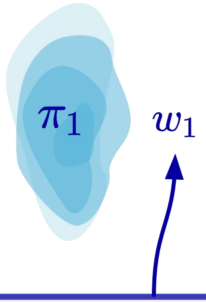
$$\min_{u_t} \int_0^1 \mathbb{E}_{p_t}\left[\frac{1}{2}\|u_t(X_t)\|^2 + V_t(X_t)\right]dt \text{ s.t. } \begin{cases} dX_t = u_t(X_t)dt + \sigma dB_t \\ X_0 \sim \pi_0, \quad X_1 \sim \pi_1 \end{cases}$$

Extend to multiple trajectories and endpoints?

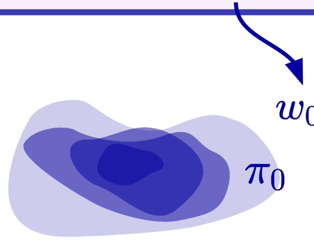
Model the **branched Schrödinger bridges** as a **set of unbalanced Schrödinger bridges**

Branched Schrödinger Bridge Problem

Unbalanced Generalized Schrödinger Bridge Problem



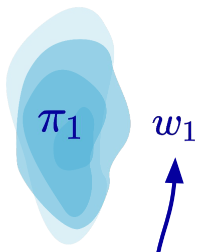
Now the initial and final distributions have **weights** that determine population size!



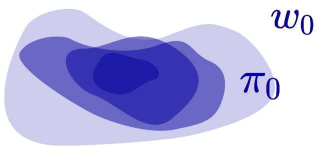
Unbalanced Conditional Stochastic Optimal Control (CondSOC)

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

Unbalanced Generalized Schrödinger Bridge Problem



Now the initial and final distributions have **weights** that determine population size!



Unbalanced Conditional Stochastic Optimal Control (CondSOC)

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

Time-varying weight

$$w_t = w_0 + \int_0^t \underbrace{g_s(X_s)}_{\text{growth rate}} ds$$

Endpoint constraints

$$w_0(X_0) = w_0^\star, \quad w_1(X_1) = w_1^\star$$

Branched Generalized Schrödinger Bridge Problem

$$\min_{\{u_{t,k},g_{t,k}\}_{k=0}^{K}} \int_0^1 \Bigg\{ \underbrace{\mathbb{E}_{p_{t,0}}\left[\frac{1}{2}\|u_{t,0}(X_{t,0})\|^2 + V_t(X_{t,0})\right] w_{t,0}}_{\text{Primary Trajectory}} + \underbrace{\sum_{k=1}^{K} \mathbb{E}_{p_{t,k}}\left[\frac{1}{2}\|u_{t,k}(X_{t,k})\|^2 + V_t(X_{t,k})\right] w_{t,k}}_{\text{Secondary Branched Trajectories}} \Bigg\} dt$$

$$w_{t,0} = 1 + \int_0^t g_{s,0}(X_{s,0})ds$$

$$w_{t,k} = \int_0^t g_{s,k}(X_{s,k})ds$$

$$\text{s.t.} \quad dX_{t,k} = u_{t,k}(X_{t,k})dt + \sigma dB_t, \;\; X_0 \sim \pi_0, \;\; X_{1,k} \sim \pi_{1,k}, \;\; w_{0,k} = \delta_{k=0}, \;\; w_{1,k} = w_{1,k}$$

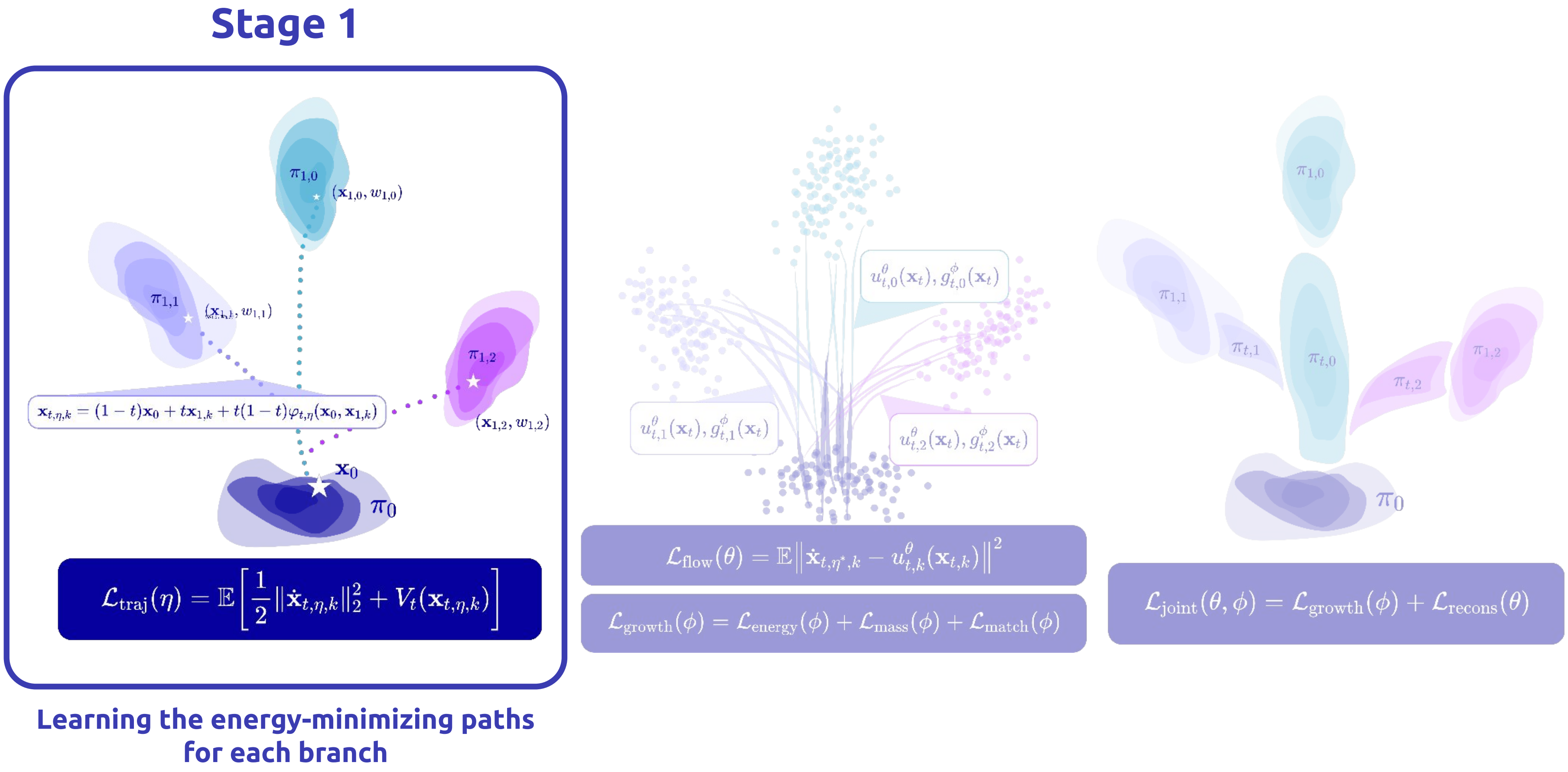
Branched Conditional Stochastic Optimal Control

Solving the Branched GSB problem with a tractable objective

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

Conditioning on pairs of samples from the dataset

Learning BranchSBM Using Neural Networks



Stage 1
$\pi_{1,0}$
$(\mathbf{x}_{1,0}, w_{1,0})$
$\pi_{1,1}$
$(\mathbf{x}_{1,1}, w_{1,1})$
$\pi_{1,2}$
$(\mathbf{x}_{1,2}, w_{1,2})$
$\mathbf{x}_{t,\eta,k} = (1-t)\mathbf{x}_0 + t\mathbf{x}_{1,k} + t(1-t)\varphi_{t,\eta}(\mathbf{x}_0, \mathbf{x}_{1,k})$
$\mathbf{x}_0$
π_0
$\mathcal{L}_{\text{traj}}(\eta) = \mathbb{E}\left[\frac{1}{2}\|\dot{\mathbf{x}}_{t,\eta,k}\|_2^2 + V_t(\mathbf{x}_{t,\eta,k})\right]$
$u^\theta_{t,0}(\mathbf{x}_t), g^\phi_{t,0}(\mathbf{x}_t)$
$u^\theta_{t,1}(\mathbf{x}_t), g^\phi_{t,1}(\mathbf{x}_t)$
$u^\theta_{t,2}(\mathbf{x}_t), g^\phi_{t,2}(\mathbf{x}_t)$
$\mathcal{L}_{\text{flow}}(\theta) = \mathbb{E}\left\|\dot{\mathbf{x}}_{t,\eta^*,k} - u^\theta_{t,k}(\mathbf{x}_{t,k})\right\|^2$
$\mathcal{L}_{\text{growth}}(\phi) = \mathcal{L}_{\text{energy}}(\phi) + \mathcal{L}_{\text{mass}}(\phi) + \mathcal{L}_{\text{match}}(\phi)$
$\pi_{1,0}$
$\pi_{1,1}$
$\pi_{t,1}$
$\pi_{t,0}$
$\pi_{t,2}$
$\pi_{1,2}$
π_0
$\mathcal{L}_{\text{joint}}(\theta,\phi) = \mathcal{L}_{\text{growth}}(\phi) + \mathcal{L}_{\text{recons}}(\theta)$
Learning the energy-minimizing paths
for each branch

Stage 1: Learning Energy-Minimizing Neural Interpolant

Energy-corrected interpolating state at time *t*

$$\mathbf{x}_{t,\eta,k} = \overbrace{(1-t)\mathbf{x}_0 + t\mathbf{x}_{1,k}}^{\text{Linear interpolant}} + t(1-t)\overbrace{\varphi_{t,\eta}(\mathbf{x}_0, \mathbf{x}_{1,k})}^{\text{Parameterized correction network}}$$

$\frac{\partial}{\partial t}$ **Time derivative** $\frac{\partial}{\partial t}$

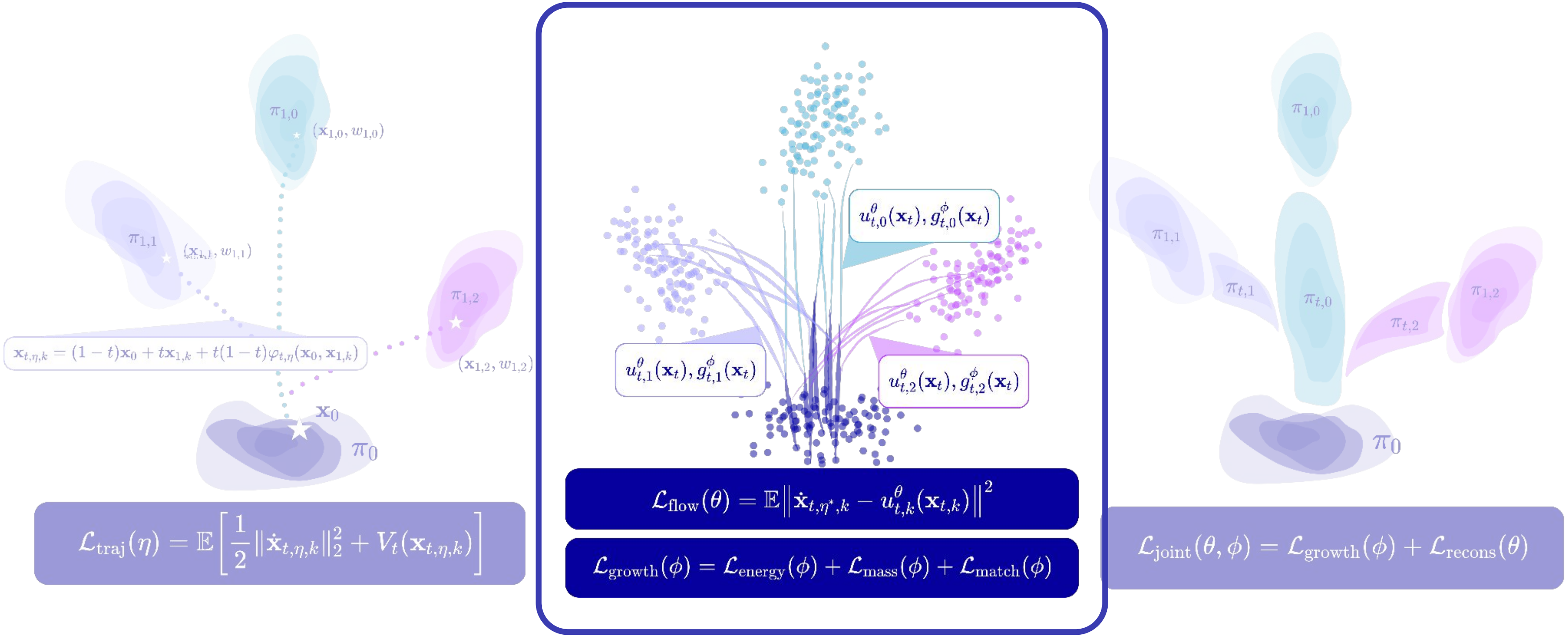
$$\dot{\mathbf{x}}_{t,\eta,k} = \mathbf{x}_1 - \mathbf{x}_0 + t(1-t)\dot{\varphi}_{t,\eta}(\mathbf{x}_0, \mathbf{x}_{1,k}) + (1-2t)\varphi_{t,\eta}(\mathbf{x}_0, \mathbf{x}_{1,k})$$

Minimize trajectory loss with respect to correction network

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

Kapusniak, Kacper, et al. "Metric flow matching for smooth interpolations on the data manifold."

Stage 2 + 3



Learning the Markovian drift and growth networks

Stage 2: Initial Training of Velocity Networks

Time derivative

$$\dot{\mathbf{x}}^{\star}_{t,\eta,k} = \mathbf{x}_1 - \mathbf{x}_0 + t(1-t)\dot{\varphi}^{\star}_{t,\eta}(\mathbf{x}_0, \mathbf{x}_{1,k}) + (1-2t)\varphi^{\star}_{t,\eta}(\mathbf{x}_0, \mathbf{x}_{1,k})$$

We don't know this at inference time
(simulation-free)

$$\{u^{\theta}_{t,k}\}_{k=0}^{K}$$

Parameterize a set of flow networks
for each branched trajectory

Minimize conditional flow matching loss with respect to flow network parameters

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

Stage 3: Initial Training of Growth Networks

Freeze flow network parameters and minimize energy loss

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

Freeze flow network parameters

$$w^\phi_{t,0} = 1 + \int_0^t g^\phi_{s,1}(\mathbf{x}_{s,1}) ds$$

$$w^\phi_{t,k} = \int_0^t g^\phi_{s,k}(\mathbf{x}_{s,k}) ds$$

We also need to ensure that boundary weights are matched and total population weight is conserved

Stage 3: Initial Training of Growth Networks

Match loss

$$\mathcal{L}_{\text{match}}(\phi) = \sum_{k=0}^{K} \mathbb{E}_{p_{1,k}} \left(w_{1,k}^{\phi}(\mathbf{x}_{1,k}) - w_{1,k}^{\star} \right)^2$$

Weight at time $t = 1$

$$w_{1,k}^{\phi}(\mathbf{x}_{1,k}) = w_{0,k} + \int_0^1 g_{t,k}^{\phi}(\mathbf{x}_{t,k})dt$$

Mass loss

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

Ensures total weight at all times equals the true weight of the total population (mass conservation = 1)

Ensures no negative weights

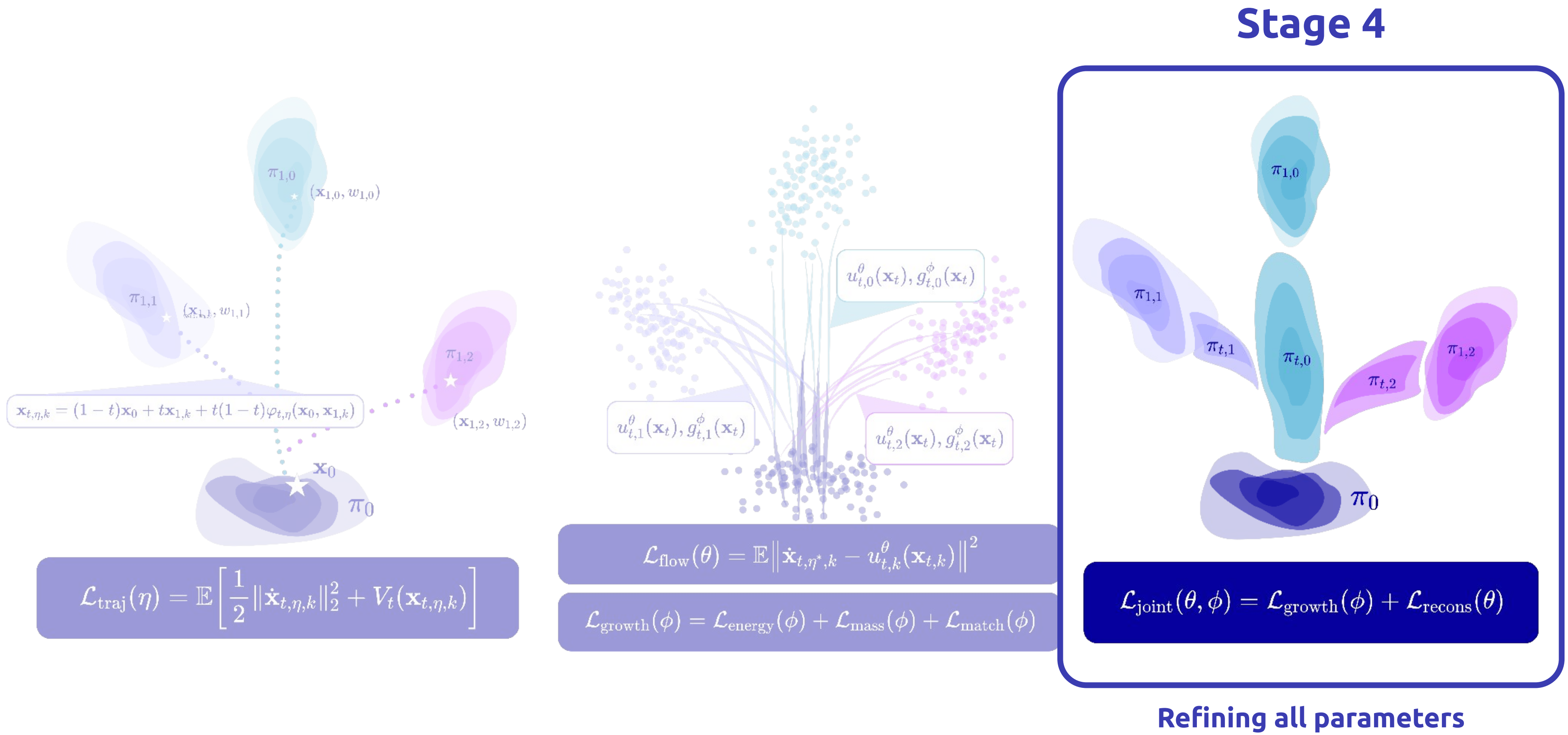
Stage 3: Initial Training of Growth Networks

Total growth loss

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

💡

Ensures that the loss function is **coercive** such that the growth rate does not diverge to infinity in norm without incurring a penalty from the loss functional (Proof in Appendix C.4)



Stage 4
$\pi_{1,0}$
$(\mathbf{x}_{1,0}, w_{1,0})$
$\pi_{1,1}$
$(\mathbf{x}_{1,1}, w_{1,1})$
$\pi_{1,2}$
$(\mathbf{x}_{1,2}, w_{1,2})$
$\mathbf{x}_{t,\eta,k} = (1-t)\mathbf{x}_0 + t\mathbf{x}_{1,k} + t(1-t)\varphi_{t,\eta}(\mathbf{x}_0, \mathbf{x}_{1,k})$
$\mathbf{x}_0$
π_0
$\mathcal{L}_{\text{traj}}(\eta) = \mathbb{E}\left[\frac{1}{2}\|\dot{\mathbf{x}}_{t,\eta,k}\|_2^2 + V_t(\mathbf{x}_{t,\eta,k})\right]$
$u^\theta_{t,0}(\mathbf{x}_t), g^\phi_{t,0}(\mathbf{x}_t)$
$u^\theta_{t,1}(\mathbf{x}_t), g^\phi_{t,1}(\mathbf{x}_t)$
$u^\theta_{t,2}(\mathbf{x}_t), g^\phi_{t,2}(\mathbf{x}_t)$
$\mathcal{L}_{\text{flow}}(\theta) = \mathbb{E}\left\|\dot{\mathbf{x}}_{t,\eta^*,k} - u^\theta_{t,k}(\mathbf{x}_{t,k})\right\|^2$
$\mathcal{L}_{\text{growth}}(\phi) = \mathcal{L}_{\text{energy}}(\phi) + \mathcal{L}_{\text{mass}}(\phi) + \mathcal{L}_{\text{match}}(\phi)$
$\pi_{1,0}$
$\pi_{1,1}$
$\pi_{t,1}$
$\pi_{t,0}$
$\pi_{t,2}$
$\pi_{1,2}$
π_0
$\mathcal{L}_{\text{joint}}(\theta, \phi) = \mathcal{L}_{\text{growth}}(\phi) + \mathcal{L}_{\text{recons}}(\theta)$
Refining all parameters

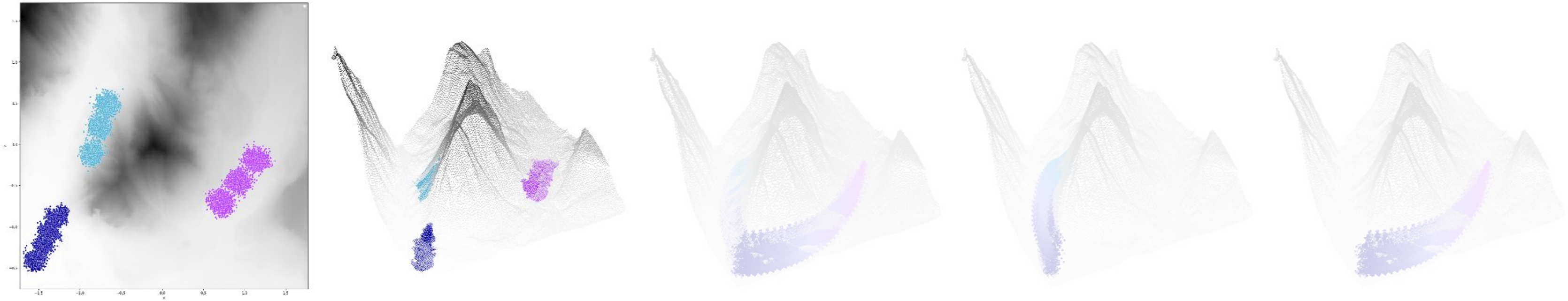
Stage 4: Joint Training of Flow and Growth Networks

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

Experimental Results

Multi-Path Navigation of 3D LiDAR Manifolds

How do we navigate diverging paths that minimize energy on 3D landscapes?



Initial and Target Distributions

p_0 and $\{p_{1,0}, p_{1,1}\}$

Learned Branching Drifts

$\{u^{\theta}_{t,0}(\mathbf{x}_t), u^{\theta}_{t,1}(\mathbf{x}_t)\}$

Branch 0

$u^{\theta}_{t,0}(\mathbf{x}_t)$

Branch 1

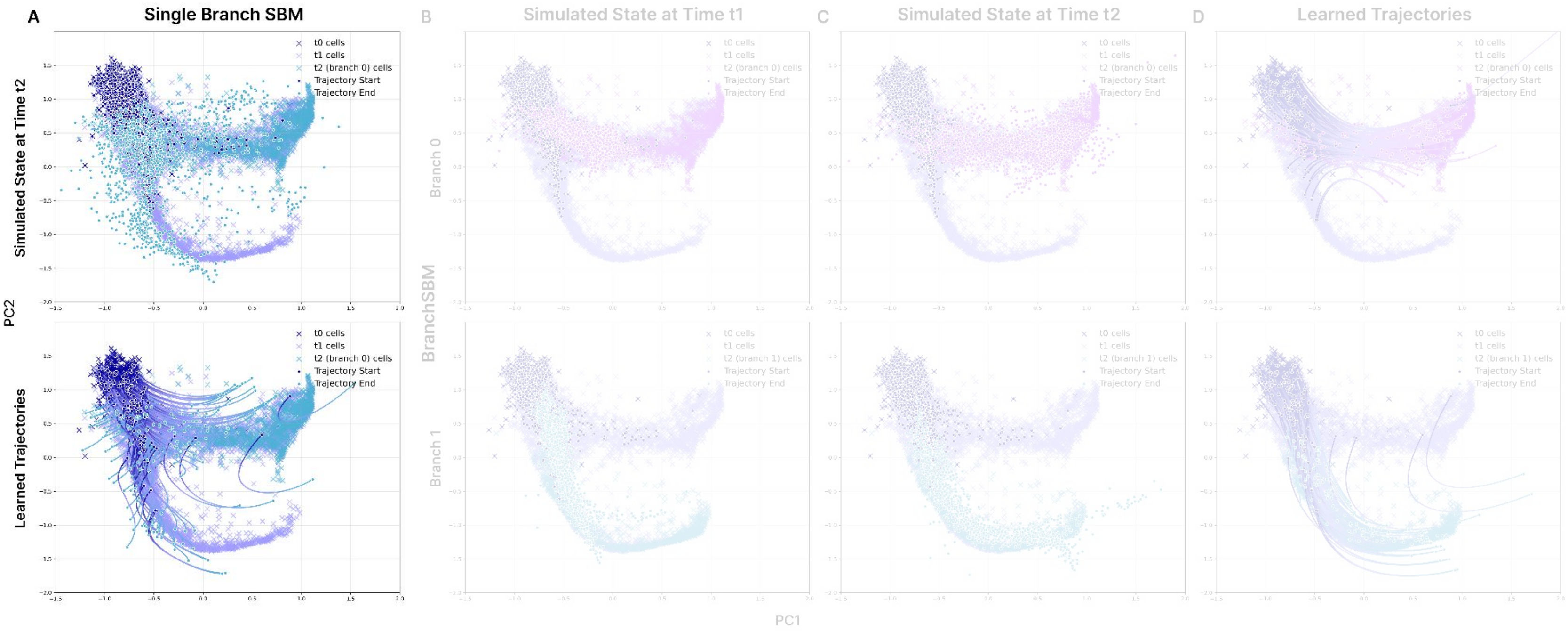
$u^{\theta}_{t,1}(\mathbf{x}_t)$

Table 1: Benchmark of BranchSBM against single-branch SBM on multi-path surface navigation. Wasserstein distances (W1 and W2) between the reconstructed and ground-truth distributions with 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	$\mathbf{0.239}_{\pm 0.001}$	$\mathbf{0.309}_{\pm 0.003}$

Modeling Cell Fate Differentiation

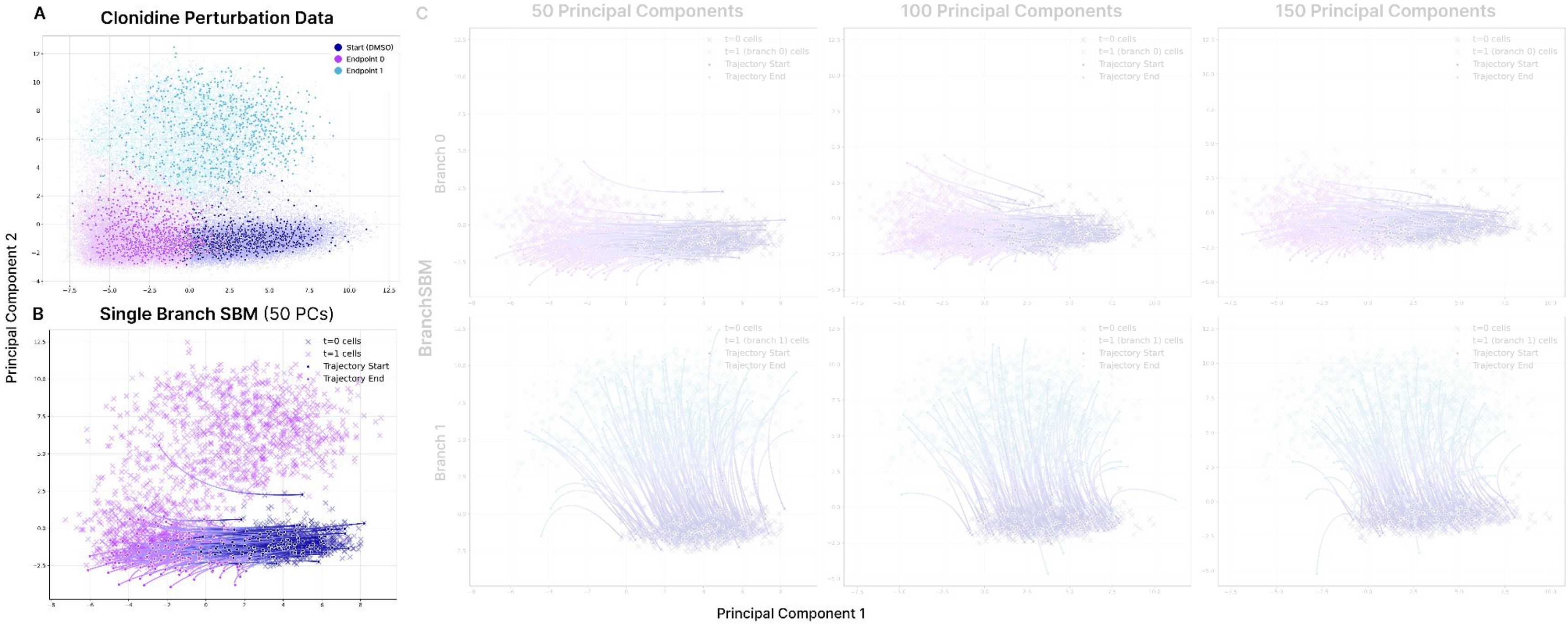
How do we model cell differentiation in the gene expression space?



Our model infers intermediate states that lie on the data manifold

Modeling Drug-Induced Perturbation Responses

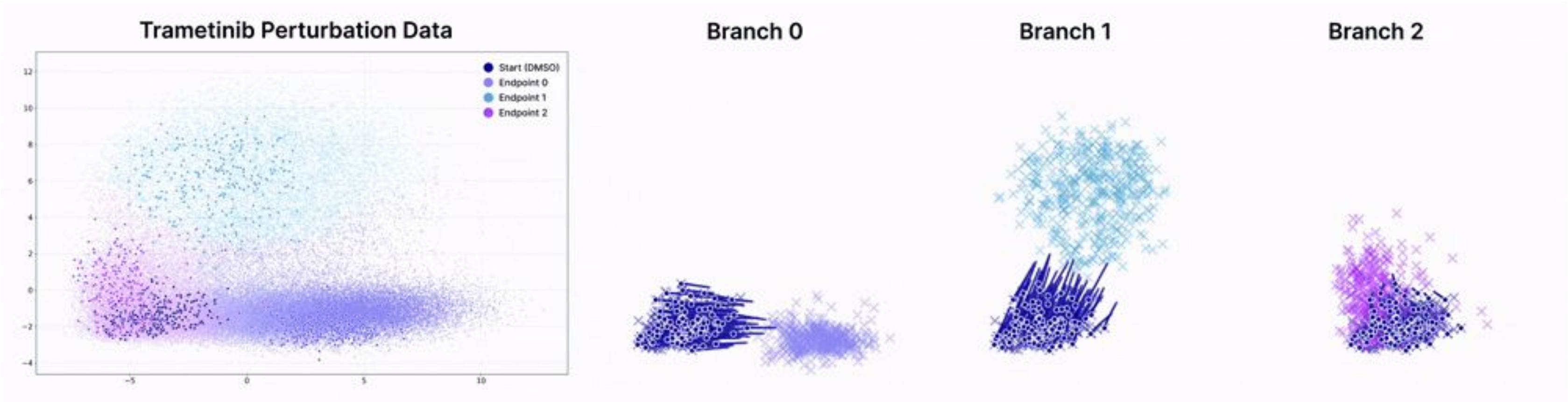
How do we model diverging cellular responses to drug-perturbation?



Our framework can scale to model trajectories in high-dimensional gene expression space

Modeling Drug-Induced Perturbation Responses

How do we model diverging cellular responses to drug-perturbation?



Modeling Drug-Induced Perturbation Responses

How do we model diverging cellular responses to drug-perturbation?

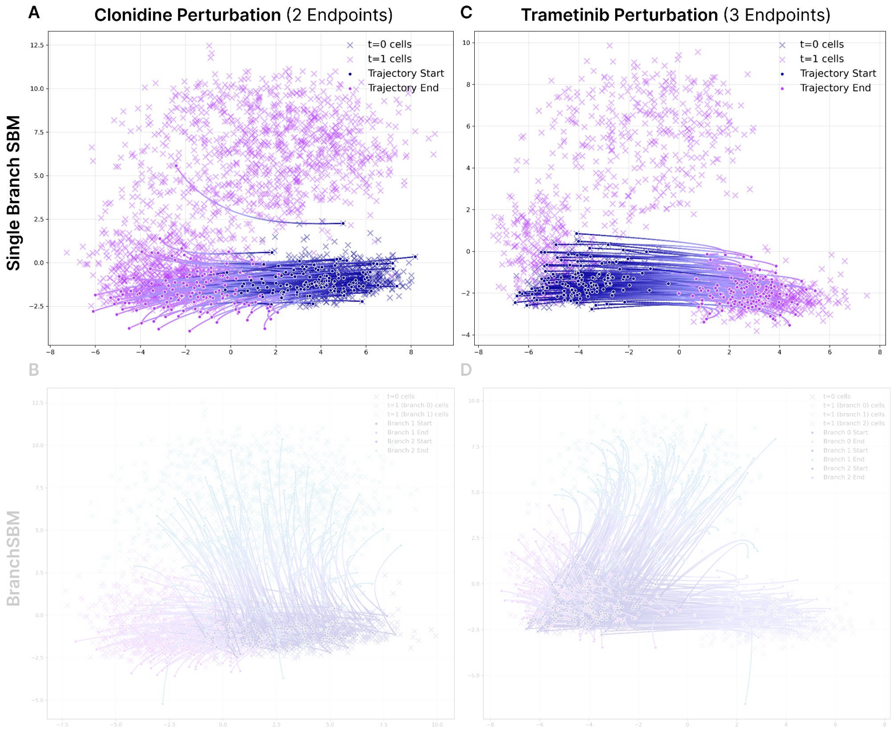


Table 2: Results for Clonidine Perturbation Modeling for Increasing Principal Component Dimensions. Maximum-mean discrepancy (MMD) across all PCs and Wasserstein distances (W1 and W2) of top 2 PCs between ground truth and reconstructed distributions at t = 1 simulated from the validation data at t = 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	$0.083_{\pm 0.001}$	$1.722_{\pm 0.064}$	$1.931_{\pm 0.035}$

Table 3: Results for Trametinib Perturbation Modeling

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	$\mathbf{0.053}_{\pm 0.001}$	$\mathbf{0.838}_{\pm 0.061}$	$\mathbf{0.973}_{\pm 0.050}$

Conclusions

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 to multiple target distributions.
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**, **modeling differentiating single-cell population dynamics**, and **predicting heterogeneous cell states after perturbation**.

Penn | DukeNUS Medical School

Thanks for Listening!

Feel free to ask any questions or contact me at sophtang@seas.upenn.edu

arXiv

Special thanks to my amazing co-authors: Yinuo Zhang, Alex Tong, and Pranam Chatterjee