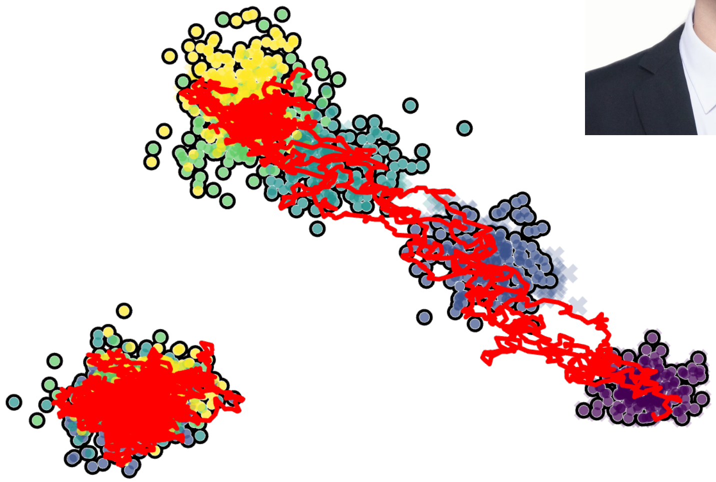
International Conference On Learning Representations

Learning stochastic dynamics from snapshots through regularized unbalanced optimal transport

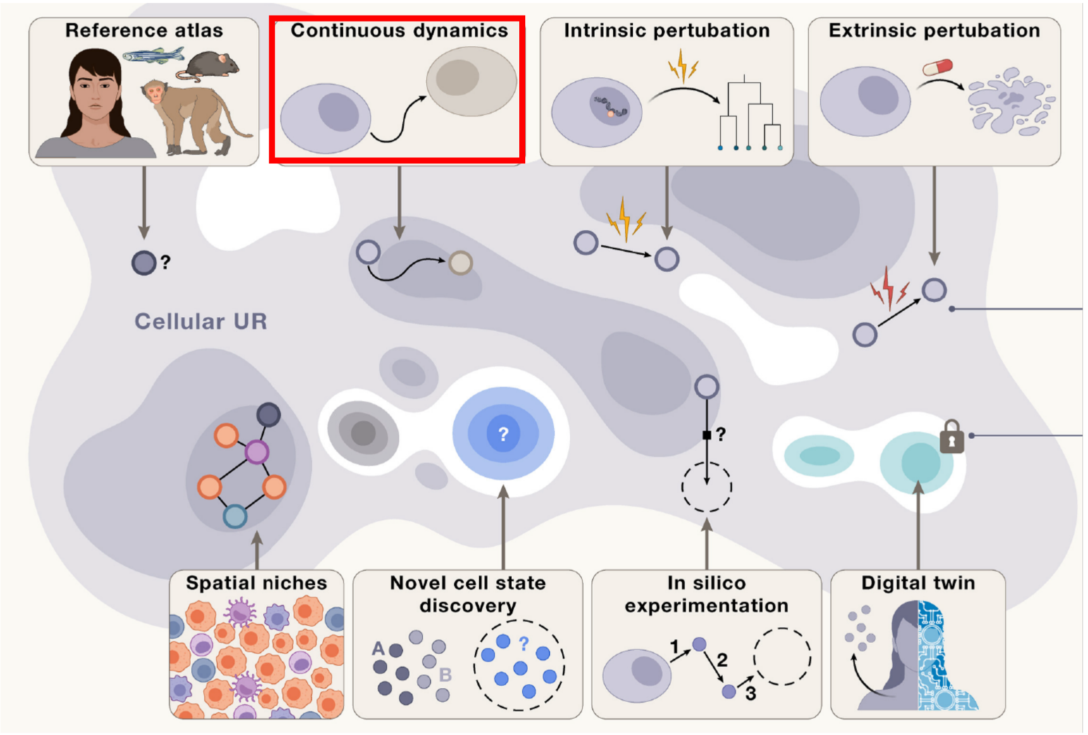
Zhenyi Zhang   Tiejun Li*   Peijie Zhou*

Peking University   *Joint corresponding authors



International Conference on Learning Representations, 2025

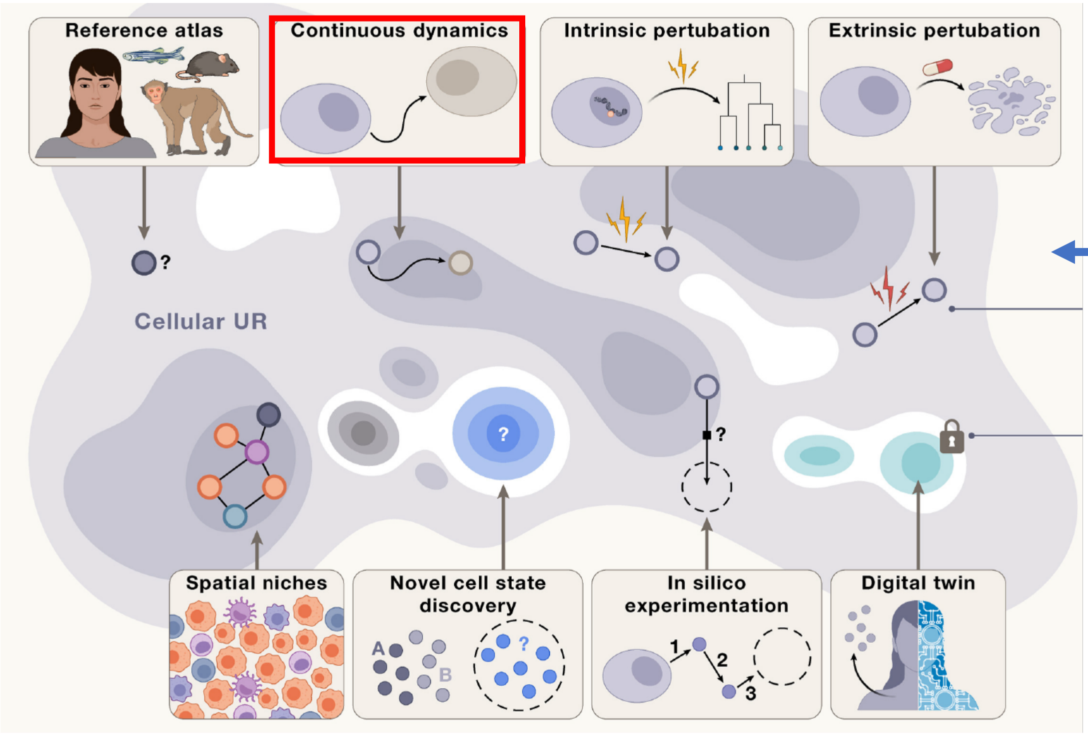
Motivation: AI Virtual Cell (AIVC)



Virtual Instrument (VI): Predict function, behavior, and dynamics

Charlotte Bunne et al.
Cell, Volume 187, Issue 25, 7045 - 7063

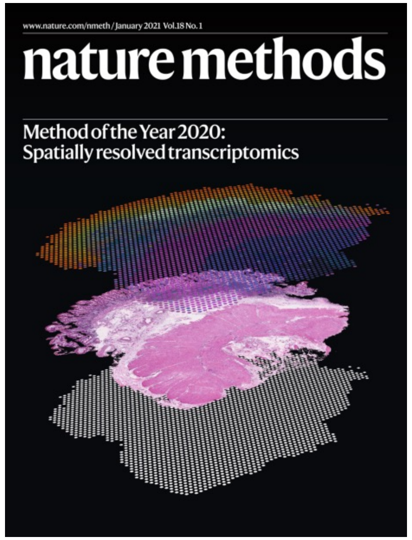
Motivation: AI Virtual Cell (AIVC)



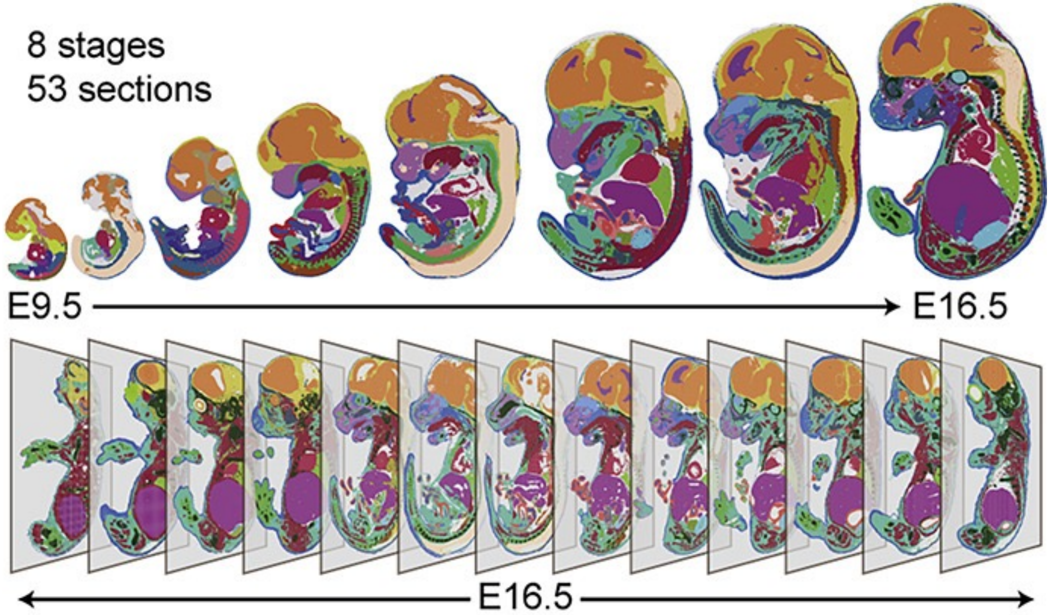
Virtual Instrument (VI): Predict function, behavior, and dynamics

Data

scRNA-seq (2018)



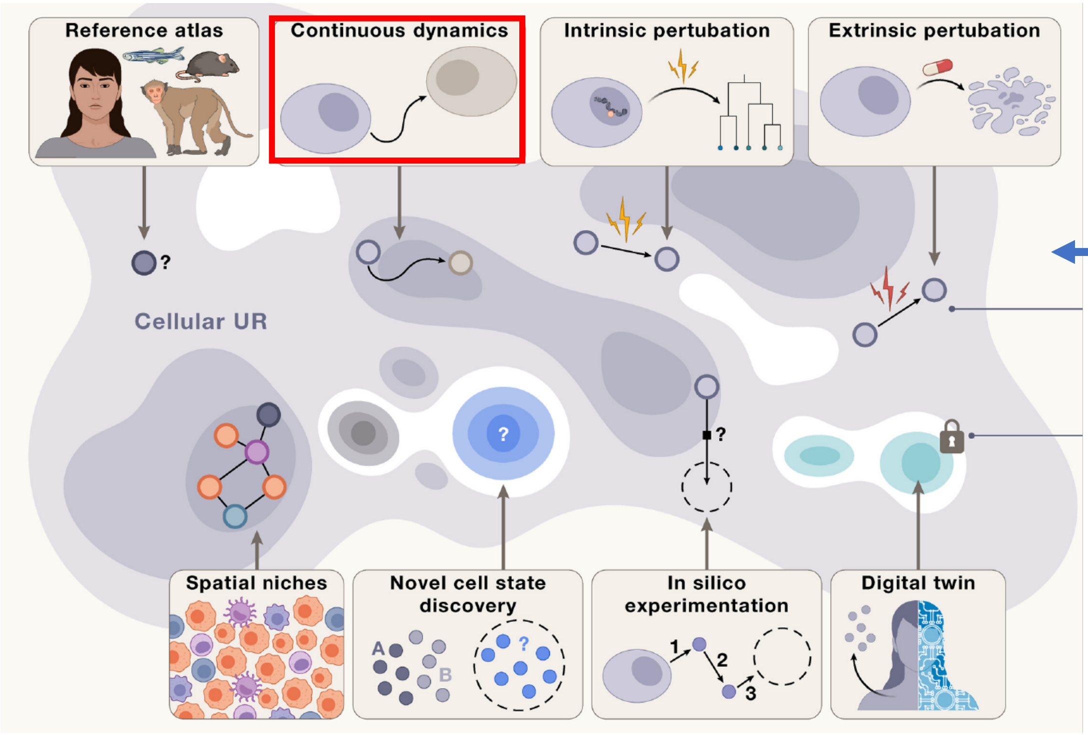
Spatial transcriptomics (2020)



Charlotte Bunne et al.
Cell, Volume 187, Issue 25, 7045 - 7063

Ao Chen et al.
Cell, Volume 185, Issue 10, 1777 - 1792.e21

Motivation: AI Virtual Cell (AIVC)

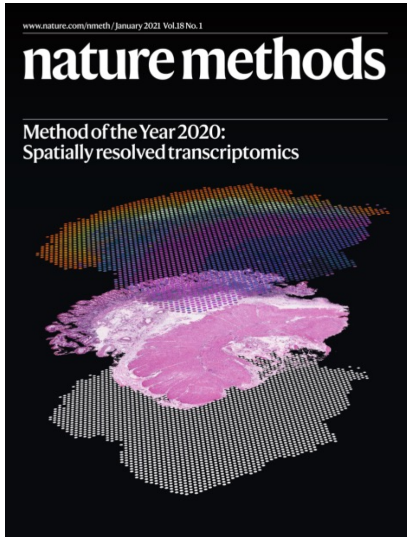


Virtual Instrument (VI): Predict function, behavior, and dynamics

Make use of dynamical systems models
(a natural "generative model")

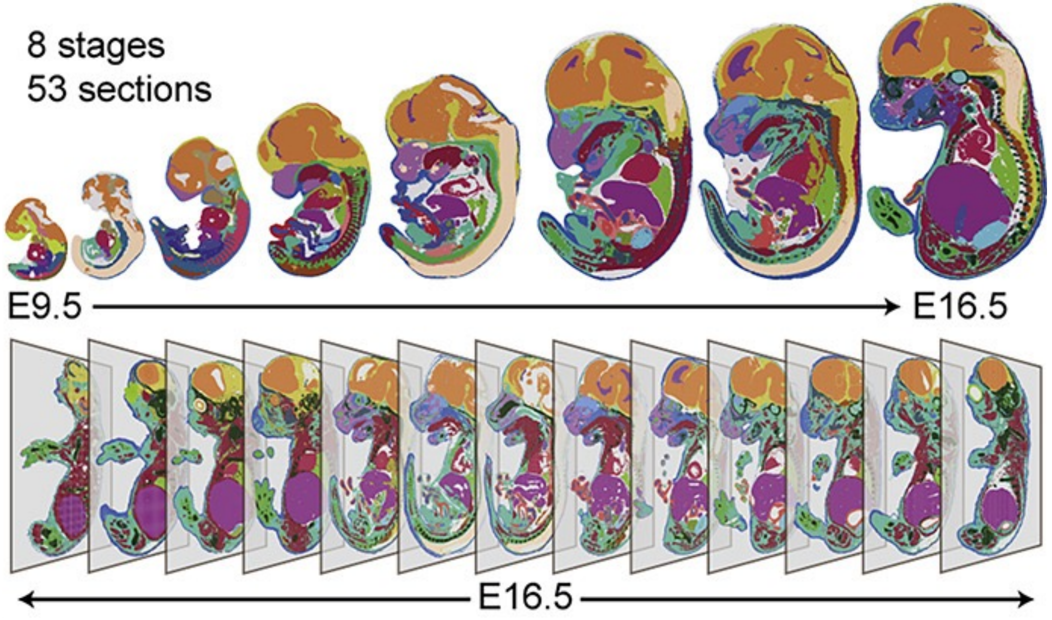
Charlotte Bunne et al.
Cell, Volume 187, Issue 25, 7045 - 7063

Data



scRNA-seq (2018)

Spatial transcriptomics (2020)



Ao Chen et al.
Cell, Volume 185, Issue 10, 1777 - 1792.e21

Goal: Inferring Dynamics from Transcriptome Data

(1) Discover

Transcriptome Data ⇄ Biological Dynamics

(2) Guide/inspire

Goal: Inferring Dynamics from Transcriptome Data

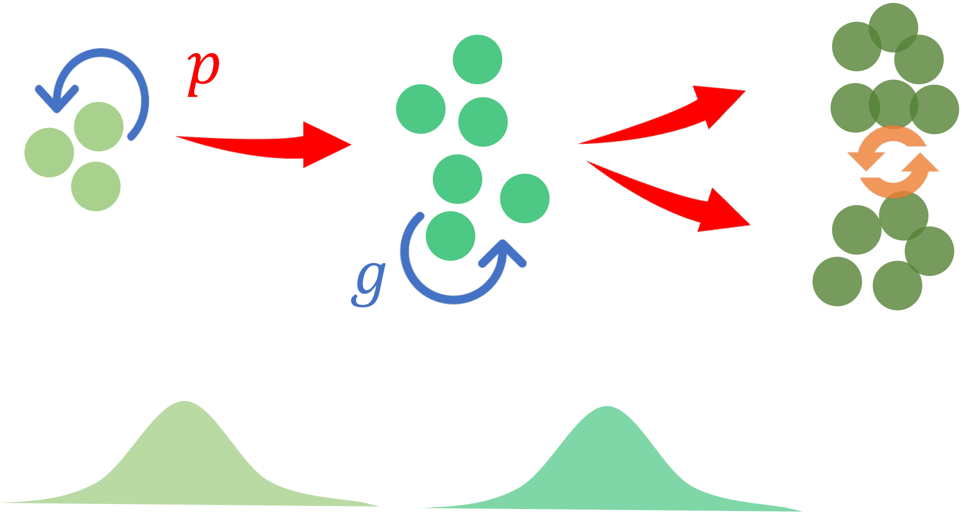
(1) Discover

Transcriptome Data ⇄ Biological Dynamics

(2) Guide/inspire

Major Gaps

- **Snapshot** nature: cells are killed during sequencing
  - ➢ Cross-sectional, instead of panel data
  - ➢ Population distribution instead of trajectory
  - ➢ Direct dynamics inference not strictly well-posed
- **High-dimensionality:** genome scale, ~20K genes
- **Sparsity and noisy**: dropout event, batch effect, multi-modality, few time points…



Goal: Inferring Dynamics from Transcriptome Data

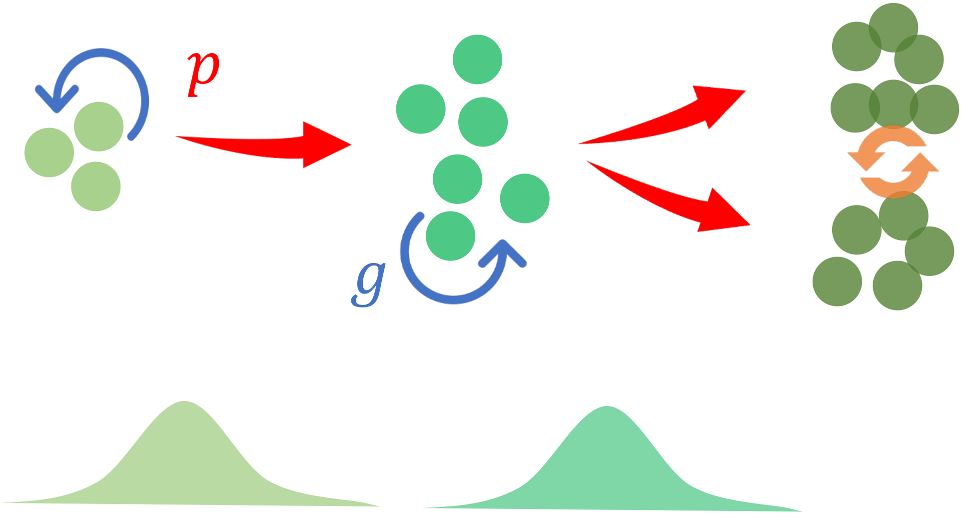
(1) Discover

Transcriptome Data ⇄ Biological Dynamics

(2) Guide/inspire

Major Gaps

- **Snapshot** nature: cells are killed during sequencing
  - ➢ Cross-sectional, instead of panel data
  - ➢ Population distribution instead of trajectory
  - ➢ Direct dynamics inference not strictly well-posed
- **High-dimensionality:** genome scale, ~20K genes
- **Sparsity and noisy**: dropout event, batch effect, multi-modality, few time points…



Key: Matching samples from high-dimensional distribution

How to learn continuous and stochastic dynamics from multiple snapshot data? 🌟

Theoretical Principles: Static Optimal Transport

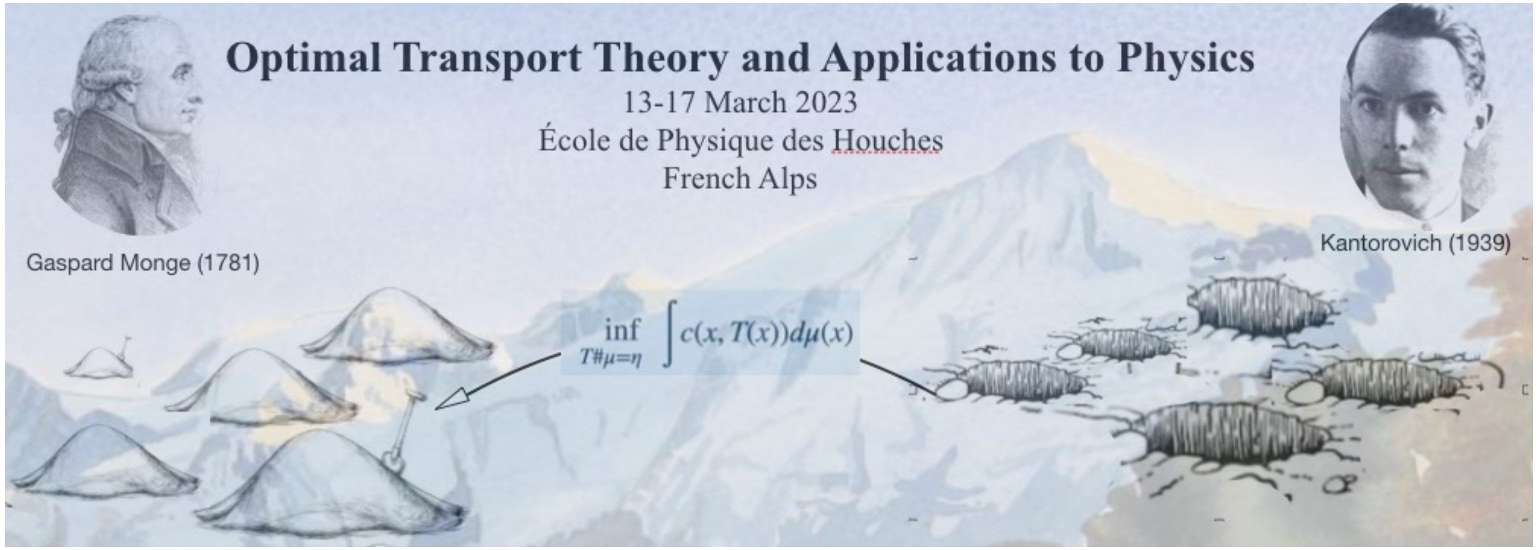
Static Optimal Transport

Static OT(1942)

$$c(x,y) = |x-y|^{2}$$

$$W(\rho_0,\rho_1) = \inf_{\pi} \sum c(x,y)\pi(x,y)dxdy$$

$$\sum_y \pi(x,y) = \rho_0 \quad \sum_x \pi(x,y) = \rho_1$$



Theoretical Principles: Static Optimal Transport

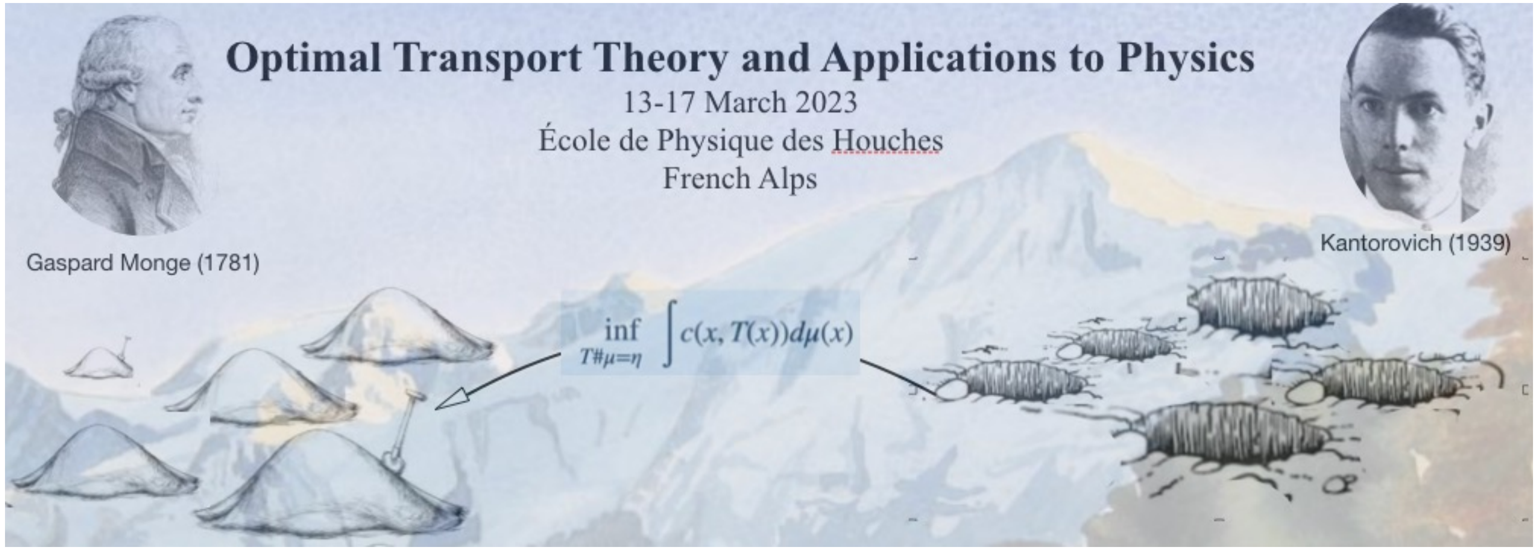
Static Optimal Transport

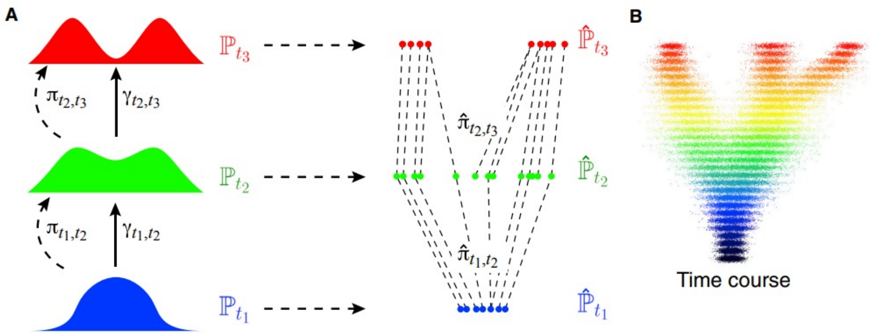
Static OT(1942)

$$c(x,y) = |x-y|^2$$

$$W(\rho_0,\rho_1) = \inf_{\pi} \sum c(x,y)\pi(x,y)dxdy$$

$$\sum_y \pi(x,y) = \rho_0 \quad \sum_x \pi(x,y) = \rho_1$$





WaddingtonOT (Geoffrey Schiebinger et al. *Cell* 2019)
Modified Kantorovich Form

Moscot (Dominik Klein et al. *Nature* 2025)

Theoretical Principles: Dynamic Optimal Transport

a) Dynamic Optimal Transport (continuous)

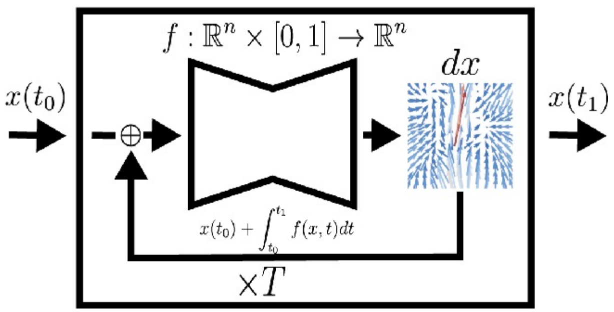
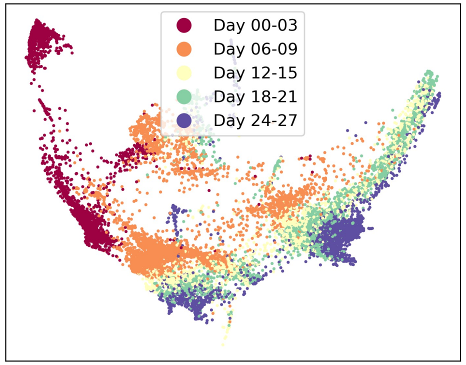


Figure 1. TrajectoryNet uses a Neural ODE to learn the derivative



TrajectoryNet (Alexander Tong et al. *ICML* 2020)
Continuous Normalizing Flow

Dynamical OT (Benamou & Brenier, 2000)

$$W(p_0, p_1) = \inf_{v,p} \int \|\boldsymbol{v}(\boldsymbol{x}, t)\|^2 p(\boldsymbol{x}, t) d\boldsymbol{x} dt$$

$$\text{s.t. } \partial_t p(\boldsymbol{x}, t) + \nabla \cdot \big(\boldsymbol{v}(\boldsymbol{x}, t) p(\boldsymbol{x}, t)\big) = 0$$

$$p(\cdot, t_0) = p_0, \; p(\cdot, t_1) = p_1$$

Theoretical Principles: Dynamic Optimal Transport

a) Dynamic Optimal Transport (continuous)

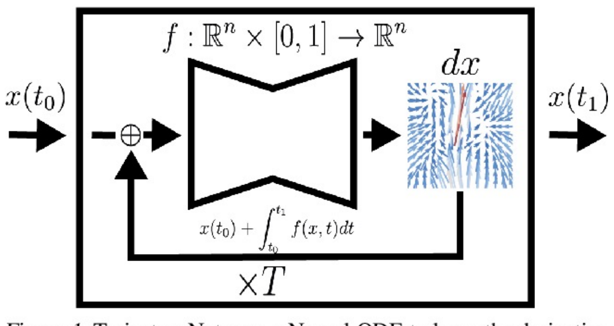
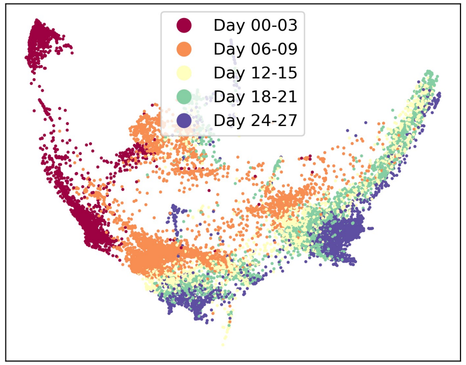


Figure 1. TrajectoryNet uses a Neural ODE to learn the derivative



TrajectoryNet (Alexander Tong et al. *ICML* 2020)
Continuous Normalizing Flow

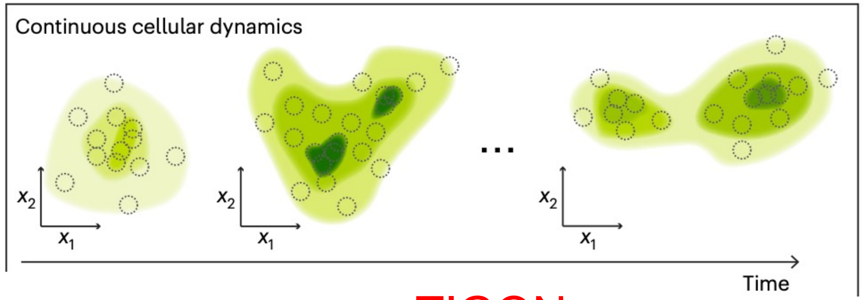
Dynamical OT (Benamou & Brenier, 2000)

$$W(p_0, p_1) = \inf_{v,p} \int ||\boldsymbol{v}(\boldsymbol{x}, t)||^2 p(\boldsymbol{x}, t) d\boldsymbol{x} dt$$

$$\text{s.t. } \partial_t p(\boldsymbol{x}, t) + \nabla \cdot \big(\boldsymbol{v}(\boldsymbol{x}, t) p(\boldsymbol{x}, t)\big) = 0$$

$$p(\cdot, t_0) = p_0, \; p(\cdot, t_1) = p_1$$

b) Unbalanced Dynamic Optimal Transport (continuous, unbalanced)



TIGON
(Yutong Sha et al. *Nature Machine Intelligence* 2024)

Wasserstein-Fisher-Rao (Benamou (2003)
Lenaic Chizat et al. (2018))

$$\inf_{v,p,g} \int \left(||\boldsymbol{v}(\boldsymbol{x}, t)||^2 + +\tau ||g||^2\right) p(\boldsymbol{x}, t) d\boldsymbol{x} dt$$

$$\text{s.t. } \partial_t p(\boldsymbol{x}, t) + \nabla \cdot \big(\boldsymbol{v}(\boldsymbol{x}, t) p(\boldsymbol{x}, t)\big) = g(\boldsymbol{x}, t) p(\boldsymbol{x}, t)$$

$$p(\cdot, t_0) = p_0, \; p(\cdot, t_1) = p_1$$

Theoretical Principles: Dynamic Optimal Transport

c) Schrödinger Bridge Problem (continuous, stochastic)

"Imagine that you observe a system of diffusing particles which is in thermal equilibrium. Suppose that at a given time t0 you see that their repartition is almost uniform and that at t1 > t0 you find a spontaneous and significant deviation from this uniformity. You are asked to explain how this deviation occurred. What is its most likely behavior?"
---- Erwin Schrodinger 1932

Theoretical Principles: Dynamic Optimal Transport

c) Schrödinger Bridge Problem (continuous, stochastic)

"Imagine that you observe a system of diffusing particles which is in thermal equilibrium. Suppose that at a given time t0 you see that their repartition is almost uniform and that at t1 > t0 you find a spontaneous and significant deviation from this uniformity. You are asked to explain how this deviation occurred. What is its most likely behavior?"
---- Erwin Schrodinger 1932

$$\min_{\mu_0^{\boldsymbol{X}} = \nu_0, \mu_1^{\boldsymbol{X}} = \nu_1} \mathcal{D}_{\mathrm{KL}}\left(\mu_{[0,1]}^{\boldsymbol{X}} \parallel \mu_{[0,1]}^{\boldsymbol{Y}}\right),$$

$$\mathrm{d}\boldsymbol{X}_t = \boldsymbol{b}(\boldsymbol{X}_t, t)\mathrm{d}t + \boldsymbol{\sigma}(\boldsymbol{X}_t, t)\mathrm{d}\boldsymbol{W}_t$$

$$\mathrm{d}\boldsymbol{Y}_t = \boldsymbol{\sigma}(\boldsymbol{Y}_t, t)\mathrm{d}\boldsymbol{W}_t$$

equivalent ⟷

Dynamic Formulation

$$\inf_{(p,\boldsymbol{b})} \int_0^1 \int_{\mathbb{R}^d} \left[\frac{1}{2}\boldsymbol{b}^T(\boldsymbol{x}, t)\boldsymbol{a}^{-1}(\boldsymbol{x}, t)\boldsymbol{b}(\boldsymbol{x}, t)\right] p(\boldsymbol{x}, t)\mathrm{d}\boldsymbol{x}\, \mathrm{d}t$$

$$\partial_t p(\boldsymbol{x}, t) = -\nabla_{\boldsymbol{x}} \cdot \left(p(\boldsymbol{x}, t)\boldsymbol{b}(\boldsymbol{x}, t)\right) + \frac{1}{2}\nabla_{\boldsymbol{x}}^2 : \left(\boldsymbol{a}(\boldsymbol{x}, t)p(\boldsymbol{x}, t)\right)$$

velocity diffusion

$$p(\cdot, t_0) = p_0, \ p(\cdot, t_1) = p_1$$

(Paolo Dai Pra 1991; Ivan Genti 2017)

How to generalize the dynamic Schrödinger Bridge problem to the unbalanced case?🔥

Formulation: Regularized Unbalanced Optimal Transport

$$\inf_{(p,\boldsymbol{b},g)} \int_0^1 \int_{\mathbb{R}^d} \frac{1}{2} \| \boldsymbol{b}(\boldsymbol{x},t) \|_2^2 \, p(\boldsymbol{x},t) \mathrm{d}\boldsymbol{x} \, \mathrm{d}t + \int_0^1 \int_{\mathbb{R}^d} \alpha \Psi(g(\boldsymbol{x},t)) p(\boldsymbol{x},t) \mathrm{d}\boldsymbol{x} \, \mathrm{d}t$$

$$\partial_t p = -\nabla_{\boldsymbol{x}} \cdot (p\boldsymbol{b}) + \frac{1}{2} \nabla_{\boldsymbol{x}}^2 : (\sigma^2(t) \boldsymbol{I} p) + g p. \qquad p(\cdot, t_0) = p_0, \; p(\cdot, t_1) = p_1$$

velocity diffusion growth

(Baradat & Lavenant, 2021)

Formulation: Regularized Unbalanced Optimal Transport

$$\inf_{(p,\boldsymbol{b},g)} \int_0^1 \int_{\mathbb{R}^d} \frac{1}{2} \| \boldsymbol{b}(\boldsymbol{x},t) \|_2^2 \, p(\boldsymbol{x},t) \mathrm{d}\boldsymbol{x} \, \mathrm{d}t + \int_0^1 \int_{\mathbb{R}^d} \alpha \Psi(g(\boldsymbol{x},t)) p(\boldsymbol{x},t) \mathrm{d}\boldsymbol{x} \, \mathrm{d}t$$

$$\partial_t p = -\nabla_{\boldsymbol{x}} \cdot (p\boldsymbol{b}) + \frac{1}{2} \nabla_{\boldsymbol{x}}^2 : (\sigma^2(t) \boldsymbol{I} p) + gp. \qquad p(\cdot, t_0) = p_0, \; p(\cdot, t_1) = p_1$$

velocity diffusion growth

(Baradat & Lavenant, 2021)

- If $g(x,t) = 0$, $\Psi(g) = +\infty$ unless $g = 0$ and $\Psi(0) = 0$ it degenerates to the **regularized optimal transport**.
- If $\sigma(t) \to 0$ and $\Psi(g(x,t)) = |g(x,t)|^2$ this degenerates to the **unbalanced dynamic optimal transport** with Wasserstein-Fisher-Rao (WFR) metric.
- If $g(x,t) = 0$ **and** $\sigma(t) \to 0$, this degenerates to **dynamic optimal transport** problem.

Formulation: Regularized Unbalanced Optimal Transport

$$\inf_{(p,\boldsymbol{b},g)} \int_0^1 \int_{\mathbb{R}^d} \frac{1}{2} \| \boldsymbol{b}(\boldsymbol{x},t) \|_2^2 \, p(\boldsymbol{x},t) \mathrm{d}\boldsymbol{x} \, \mathrm{d}t + \int_0^1 \int_{\mathbb{R}^d} \alpha \Psi(g(\boldsymbol{x},t)) p(\boldsymbol{x},t) \mathrm{d}\boldsymbol{x} \, \mathrm{d}t$$

$$\partial_t p = -\nabla_{\boldsymbol{x}} \cdot (p\boldsymbol{b}) + \frac{1}{2} \nabla_{\boldsymbol{x}}^2 : (\sigma^2(t) \boldsymbol{I} p) + gp. \qquad p(\cdot, t_0) = p_0, \; p(\cdot, t_1) = p_1$$

velocity diffusion growth

(Baradat & Lavenant, 2021)

- If $g(x,t) = 0$, $\Psi(g) = +\infty$ unless $g = 0$ and $\Psi(0) = 0$ it degenerates to the **regularized optimal transport**.
- If $\sigma(t) \to 0$ and $\Psi(g(x,t)) = |g(x,t)|^2$ this degenerates to the **unbalanced dynamic optimal transport** with Wasserstein-Fisher-Rao (WFR) metric.
- If $g(x,t) = 0$ **and** $\sigma(t) \to 0$, this degenerates to **dynamic optimal transport** problem.

Remain Question:

- **How to learn RUOT from samples? (Neural SDE?)**

→ Neural ODE (Fisher information regularization)

Efficient solver: Fisher Information Regularization

$$\inf_{(p,v,g)} \int_0^1 \int_{\mathbb{R}^d} \left[\frac{1}{2}\| \boldsymbol{v}(\boldsymbol{x},t) \|_2^2 + \boxed{\frac{\sigma^4(t)}{8}\|\nabla_{\boldsymbol{x}} \log p\|_2^2} - \frac{\sigma^2(t)}{2}(1+\log p)g - \frac{1}{2}\frac{\mathrm{d}\sigma^2(t)}{\mathrm{d}t}\log p + \alpha\Psi(g)\right] p(\boldsymbol{x},t)\mathrm{d}\boldsymbol{x}\,\mathrm{d}t$$

Fisher Information Regularization

$$\partial_t p = -\nabla_{\boldsymbol{x}} \cdot \big(p\boldsymbol{v}(\boldsymbol{x},t)\big) + g(\boldsymbol{x},t)p$$

$$p(\cdot,t_0) = p_0, \qquad p(\cdot,t_1) = p_1$$

Transforming SDE (Fokker-Planck PDE) constraints into ODE (Liouville PDE) constraints

Efficient solver: Fisher Information Regularization

$$\inf_{(p,v,g)} \int_0^1 \int_{\mathbb{R}^d} \left[\frac{1}{2} \| \boldsymbol{v}(\boldsymbol{x},t) \|_2^2 + \boxed{\frac{\sigma^4(t)}{8} \|\nabla_{\boldsymbol{x}} \log p\|_2^2} - \frac{\sigma^2(t)}{2}(1+\log p) g - \frac{1}{2}\frac{\mathrm{d}\sigma^2(t)}{\mathrm{d}t} \log p + \alpha \Psi(g)\right] p(\boldsymbol{x},t) \mathrm{d}\boldsymbol{x}\, \mathrm{d}t$$

Fisher Information Regularization

$$\partial_t p = -\nabla_{\boldsymbol{x}} \cdot \big(p \boldsymbol{v}(\boldsymbol{x},t)\big) + g(\boldsymbol{x},t) p$$

$$p(\cdot, t_0) = p_0, \qquad p(\cdot, t_1) = p_1$$

Transforming SDE (Fokker-Planck PDE) constraints into ODE (Liouville PDE) constraints

$$\boldsymbol{b}(\boldsymbol{x},t) = \boldsymbol{v}(\boldsymbol{x},t) + \frac{1}{2}\sigma^2(t) \nabla_{\boldsymbol{x}} \log p\, (\boldsymbol{x},t)$$

original velocity — new velocity — score function

So to specify the SDE is equivalent to specify the **probability flow ODE** and **score function.**

Efficient solver: Fisher Information Regularization

$$\inf_{(p,v,g)} \int_0^1 \int_{\mathbb{R}^d} \left[\frac{1}{2} \| \boldsymbol{v}(\boldsymbol{x},t) \|_2^2 + \boxed{\frac{\sigma^4(t)}{8} \|\nabla_{\boldsymbol{x}} \log p\|_2^2} - \frac{\sigma^2(t)}{2}(1+\log p) g - \frac{1}{2}\frac{\mathrm{d}\sigma^2(t)}{\mathrm{d}t} \log p + \alpha \Psi(g) \right] p(\boldsymbol{x},t) \mathrm{d}\boldsymbol{x}\, \mathrm{d}t$$

Fisher Information Regularization

$$\partial_t p = -\nabla_{\boldsymbol{x}} \cdot \left(p \boldsymbol{v}(\boldsymbol{x},t) \right) + g(\boldsymbol{x},t) p$$

$$p(\cdot, t_0) = p_0, \qquad p(\cdot, t_1) = p_1$$

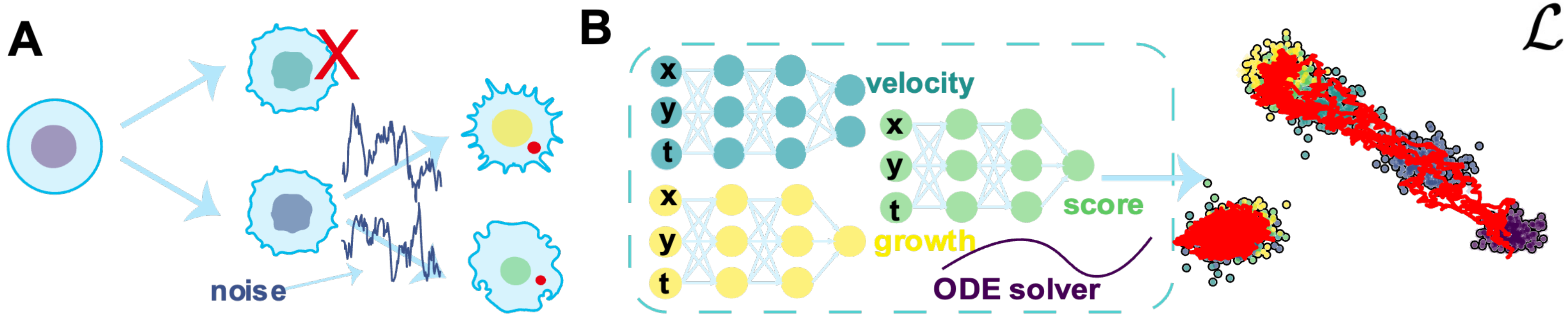
Transforming SDE (Fokker-Planck PDE) constraints into ODE (Liouville PDE) constraints

$$\boldsymbol{b}(\boldsymbol{x},t) = \boldsymbol{v}(\boldsymbol{x},t) + \frac{1}{2}\sigma^2(t) \nabla_{\boldsymbol{x}} \log p\,(\boldsymbol{x},t)$$

original velocity — new velocity — score function

So to specify the SDE is equivalent to specify the **probability flow ODE** and **score function.**

Parameterize $v(x,t)$, $g(x,t)$, and $\frac{1}{2}\sigma^2 \log p(x,t)$ using neural networks respectively.



$$\mathcal{L} = \mathcal{L}_{\text{Energy}} + \lambda_r \mathcal{L}_{\text{Recons}} + \lambda_f \mathcal{L}_{\text{FP}}.$$

Energy loss +
Reconstruction loss +
Fokker-Planck constraint

Efficient solver: Designing the Loss Function

$$\mathcal{L} = \mathcal{L}_{\text{Energy}} + \lambda_r \mathcal{L}_{\text{Recons}} + \lambda_f \mathcal{L}_{\text{FP}}.$$

Transport energy Mass Matching PINN for score matching

Zhang, Li and Zhou, ICLR 2025; Raissi et al.

Efficient solver: Designing the Loss Function

$$\mathcal{L} = \mathcal{L}_{\text{Energy}} + \lambda_r \mathcal{L}_{\text{Recons}} + \lambda_f \mathcal{L}_{\text{FP}}.$$

Transport energy   Mass Matching   PINN for score matching

Transport Energy loss

$$\mathcal{L}_{\text{Energy}} = \mathbb{E}_{\boldsymbol{x}_0 \sim p_0} \int_0^T \left[\frac{1}{2} \|\boldsymbol{v}_\theta\|_2^2 + \frac{1}{2} \|\nabla_{\boldsymbol{x}} s_\theta\|_2^2 - \left(\frac{\sigma^2(t)}{2} + s_\theta \right) g_\theta - \frac{(\sigma^2(t))'}{\sigma^2(t)} s_\theta + \alpha \Psi(g_\theta) \right] w_\theta(t) \mathrm{d}t,$$

Zhang, Li and Zhou, ICLR 2025;   Raissi et al.

Efficient solver: Designing the Loss Function

$$\mathcal{L} = \mathcal{L}_{\text{Energy}} + \lambda_r \mathcal{L}_{\text{Recons}} + \lambda_f \mathcal{L}_{\text{FP}}.$$

Transport energy   Mass Matching   PINN for score matching

Transport Energy loss

$$\mathcal{L}_{\text{Energy}} = \mathbb{E}_{\boldsymbol{x}_0 \sim p_0} \int_0^T \left[\frac{1}{2} \|\boldsymbol{v}_\theta\|_2^2 + \frac{1}{2} \|\nabla_{\boldsymbol{x}} s_\theta\|_2^2 - \left(\frac{\sigma^2(t)}{2} + s_\theta \right) g_\theta - \frac{(\sigma^2(t))'}{\sigma^2(t)} s_\theta + \alpha \Psi(g_\theta) \right] w_\theta(t) \mathrm{d}t,$$

Mass Matching loss

$$\frac{\mathrm{d} \log w_i(t)}{\mathrm{d}t} = g(\boldsymbol{x}_i(t), t)$$

Particle weights

"local mass"

$$M_k = \sum_{i=1}^{N_0} \left\| w_i(t_k) - \operatorname{card}\left(h_k^{-1}(\boldsymbol{x}_i(t_k)) \right) \frac{1}{N_0} \right\|_2^2$$

$$R_d = \lambda_m \sum_{k=1}^{T-1} M_k + \lambda_d D_{0,T},$$

$$D_{0,T} := \sum_{k=1}^{T-1} \mathcal{W}_2\left(\hat{\boldsymbol{w}}^k, \boldsymbol{w}(t_k)\right)$$

"global shape"

Zhang, Li and Zhou, ICLR 2025;   Raissi et al.

Efficient solver: Designing the Loss Function

$$\mathcal{L} = \mathcal{L}_{\text{Energy}} + \lambda_r \mathcal{L}_{\text{Recons}} + \lambda_f \mathcal{L}_{\text{FP}}.$$

Transport energy   Mass Matching   PINN for score matching

Transport Energy loss

$$\mathcal{L}_{\text{Energy}} = \mathbb{E}_{\boldsymbol{x_0} \sim p_0} \int_0^T \left[\frac{1}{2} \|\boldsymbol{v}_\theta\|_2^2 + \frac{1}{2} \|\nabla_{\boldsymbol{x}} s_\theta\|_2^2 - \left(\frac{\sigma^2(t)}{2} + s_\theta \right) g_\theta - \frac{(\sigma^2(t))'}{\sigma^2(t)} s_\theta + \alpha \Psi(g_\theta) \right] w_\theta(t) \mathrm{d}t,$$

Mass Matching loss

$$\frac{\mathrm{d} \log w_i(t)}{\mathrm{d}t} = g(\boldsymbol{x}_i(t), t)$$ Particle weights

"local mass"

$$M_k = \sum_{i=1}^{N_0} \left\| w_i(t_k) - \text{card}\left(h_k^{-1}(\boldsymbol{x}_i(t_k))\right) \frac{1}{N_0} \right\|_2^2$$

$$R_d = \lambda_m \sum_{k=1}^{T-1} M_k + \lambda_d D_{0,T},$$

$$D_{0,T} := \sum_{k=1}^{T-1} \mathcal{W}_2\left(\hat{\boldsymbol{w}}^k, \boldsymbol{w}(t_k)\right)$$ "global shape"

PINN loss for "score matching"

$$W_{\text{PINN}} = \|\partial_t p + \nabla_{\boldsymbol{x}} \cdot (p \boldsymbol{v}(\boldsymbol{x}, t)) - g(\boldsymbol{x}, t) p\|_2 + \lambda_w \|p(\boldsymbol{x}, 0) - p_0\|_2 .$$

$s_\theta \approx \log p(x, t)$

negative "landscape"

Zhang, Li and Zhou, ICLR 2025;   Raissi et al.

Efficient solver: DeepRUOT

Algorithm 1 Training Regularized Unbalanced Optimal Transport

Require: Datasets $A_0, \ldots, A_{T-1}$, batch size N, maximum ode iteration n_{ode}, maximum log density iteration $n_{\text{log-density}}$, initialized ODE $\boldsymbol{v_\theta}$, growth g_θ and log density s_θ

Ensure: Trained neural ODE $\boldsymbol{v_\theta}$, growth function g_θ and log density function s_θ.

```
Pre-Training Stage:
for i = 1 to n_ode do                                          ▷ Distribution Reconstruction training
    for t = 0 to T − 2 do
        Â_{t+1} ← φ_θ^v (Â_t, t + 1), w(Â_{t+1}) ← φ_θ^g (w(Â_t), t + 1).
        L_Recons ← L_Recons + λ_m M_t + λ_d W_2(ŵ^t, w(t)) (11), update v_θ and g_θ w.r.t. L_Recons
        with hyperparameter scheduling (Appendix C.2).
for t = 0 to T − 2 do
    Â_{t+1} ← φ_θ^v (Â_t, t + 1)                               ▷ Generating samples from learned v_θ.
for i = 1 to n_log-density do                                  ▷ CFM Score matching (Tong et al., 2024b)
    (x_0, x_1) ∼ q(x_0, x_1);   t ∼ U(0, 1);   x ∼ p(x, t | (x_0, x_1)) at generated datasets
    Â_0, ··· , Â_{T−1}, L_score ← ‖λ_s(t)∇_x s_θ(x, t) + ε_1‖_2^2 (13) , update s_θ w.r.t. the loss L_score
Training Stage:
for i = 1 to n_ode do
    Estimate the initial distribution through Gaussian Mixture Model (Appendix A.3).
    for t = 0 to T − 2 do
        Â_{t+1} ← φ_θ^v (Â_t, t + 1), w(Â_{t+1}) ← φ_θ^g (w(Â_t), t + 1)
        L_Energy ← E_{x_t∼p_t} ∫_t^{t+1} [ ½‖v_θ‖_2^2 + ½‖∇_x s_θ‖_2^2 − (σ²/2 + s_θ) g_θ − ((σ²(t))′/σ²(t)) s_θ + αΨ(g_θ) ] w(z)dz
        L_Recons ← L_Recons + λ_m M_t + λ_d W_2(ŵ^t, w(t)) (11)
        L_FP ← ‖∂_t p_θ + ∇_x · (p_θ v_θ(x, t)) − g_θ(x, t)p_θ‖ + λ_w ‖p_θ(x, 0) − p_0‖ (12)
        L ← L_Energy + λ_r L_Recons + λ_f L_FP (9) , update v_θ, g_θ and s_θ w.r.t. L
```

Efficient solver: DeepRUOT

Algorithm 1 Training Regularized Unbalanced Optimal Transport

Require: Datasets $A_0, \ldots, A_{T-1}$, batch size N, maximum ode iteration n_{ode}, maximum log density iteration $n_{\text{log-density}}$, initialized ODE $\boldsymbol{v_\theta}$, growth g_θ and log density s_θ

Ensure: Trained neural ODE $\boldsymbol{v_\theta}$, growth function g_θ and log density function s_θ.

Pre-Training Stage:

for $i = 1$ to n_{ode} **do** ▷ *Distribution Reconstruction training*

  for $t = 0$ to $T-2$ **do**

    $\widehat{A}_{t+1} \leftarrow \phi_\theta^{\boldsymbol{v}}\left(\widehat{A}_t, t+1\right), w(\widehat{A}_{t+1}) \leftarrow \phi_\theta^{g}\left(w(\widehat{A}_t), t+1\right)$.

    $\mathcal{L}_{\text{Recons}} \leftarrow \mathcal{L}_{\text{Recons}} + \lambda_m M_t + \lambda_d \mathcal{W}_2(\hat{\boldsymbol{w}}^t, \boldsymbol{w}(t))$ (11), update $\boldsymbol{v_\theta}$ and g_θ w.r.t. $\mathcal{L}_{\text{Recons}}$ with hyperparameter scheduling (Appendix C.2).

for $t = 0$ to $T-2$ **do**

  $\widehat{A}_{t+1} \leftarrow \phi_\theta^{\boldsymbol{v}}\left(\widehat{A}_t, t+1\right)$ ▷ *Generating samples from learned $\boldsymbol{v_\theta}$.*

for $i = 1$ to $n_{\text{log-density}}$ **do** ▷ *CFM Score matching (Tong et al., 2024b)*

  $(\boldsymbol{x}_0, \boldsymbol{x}_1) \sim q(\boldsymbol{x}_0, \boldsymbol{x}_1); \quad t \sim \mathcal{U}(0,1); \quad \boldsymbol{x} \sim p(\boldsymbol{x}, t \mid (\boldsymbol{x}_0, \boldsymbol{x}_1))$ at generated datasets $\widehat{A}_0, \cdots, \widehat{A}_{T-1}$, $\mathcal{L}_{\text{score}} \leftarrow \|\lambda_s(t)\nabla_{\boldsymbol{x}} s_\theta(\boldsymbol{x}, t) + \boldsymbol{\epsilon_1}\|_2^2$ (13), update s_θ w.r.t. the loss $\mathcal{L}_{\text{score}}$

Training Stage:

for $i = 1$ to n_{ode} **do**

  Estimate the initial distribution through Gaussian Mixture Model (Appendix A.3).

  for $t = 0$ to $T-2$ **do**

    $\widehat{A}_{t+1} \leftarrow \phi_\theta^{\boldsymbol{v}}\left(\widehat{A}_t, t+1\right), w(\widehat{A}_{t+1}) \leftarrow \phi_\theta^{g}\left(w(\widehat{A}_t), t+1\right)$

    $\mathcal{L}_{\text{Energy}} \leftarrow \mathbb{E}_{\boldsymbol{x}_t \sim p_t} \int_t^{t+1} \left[\frac{1}{2}\|\boldsymbol{v}_\theta\|_2^2 + \frac{1}{2}\|\nabla_{\boldsymbol{x}} s_\theta\|_2^2 - \left(\frac{\sigma^2}{2} + s_\theta\right) g_\theta - \frac{(\sigma^2(t))'}{\sigma^2(t)} s_\theta + \alpha \Psi(g_\theta)\right] w(z)\mathrm{d}z$

    $\mathcal{L}_{\text{Recons}} \leftarrow \mathcal{L}_{\text{Recons}} + \lambda_m M_t + \lambda_d \mathcal{W}_2(\hat{\boldsymbol{w}}^t, \boldsymbol{w}(t))$ (11)

    $\mathcal{L}_{\text{FP}} \leftarrow \|\partial_t p_\theta + \nabla_{\boldsymbol{x}} \cdot (p_\theta \boldsymbol{v}_\theta(\boldsymbol{x}, t)) - g_\theta(\boldsymbol{x}, t) p_\theta\| + \lambda_w \|p_\theta(\boldsymbol{x}, 0) - p_0\|$ (12)

    $\mathcal{L} \leftarrow \mathcal{L}_{\text{Energy}} + \lambda_r \mathcal{L}_{\text{Recons}} + \lambda_f \mathcal{L}_{\text{FP}}$ (9), update $\boldsymbol{v_\theta}$, g_θ and s_θ w.r.t. $\mathcal{L}$

Pretraining Stage

Learn initial velocity, growth and score function

Recons loss

Flow matching

Efficient solver: DeepRUOT

Algorithm 1 Training Regularized Unbalanced Optimal Transport

Require: Datasets $A_0, \dots, A_{T-1}$, batch size N, maximum ode iteration n_{ode}, maximum log density iteration $n_{\text{log-density}}$, initialized ODE $\boldsymbol{v_\theta}$, growth g_θ and log density s_θ

Ensure: Trained neural ODE $\boldsymbol{v_\theta}$, growth function g_θ and log density function s_θ.

Pre-Training Stage:

for $i = 1$ to n_{ode} **do** ▷ *Distribution Reconstruction training*

  for $t = 0$ to $T-2$ **do**

    $\widehat{A}_{t+1} \leftarrow \phi_\theta^{\boldsymbol{v}}\left(\widehat{A}_t, t+1\right), w(\widehat{A}_{t+1}) \leftarrow \phi_\theta^{g}\left(w(\widehat{A}_t), t+1\right)$.

    $\mathcal{L}_{\text{Recons}} \leftarrow \mathcal{L}_{\text{Recons}} + \lambda_m M_t + \lambda_d \mathcal{W}_2(\hat{\boldsymbol{w}}^t, \boldsymbol{w}(t))$ (11), update $\boldsymbol{v_\theta}$ and g_θ w.r.t. $\mathcal{L}_{\text{Recons}}$ with hyperparameter scheduling (Appendix C.2).

for $t = 0$ to $T-2$ **do**

  $\widehat{A}_{t+1} \leftarrow \phi_\theta^{\boldsymbol{v}}\left(\widehat{A}_t, t+1\right)$ ▷ *Generating samples from learned $\boldsymbol{v_\theta}$.*

for $i = 1$ to $n_{\text{log-density}}$ **do** ▷ *CFM Score matching (Tong et al., 2024b)*

  $(\boldsymbol{x}_0, \boldsymbol{x}_1) \sim q(\boldsymbol{x}_0, \boldsymbol{x}_1); \quad t \sim \mathcal{U}(0,1); \quad \boldsymbol{x} \sim p(\boldsymbol{x}, t \mid (\boldsymbol{x}_0, \boldsymbol{x}_1))$ at generated datasets

  $\widehat{A}_0, \cdots, \widehat{A}_{T-1}, \mathcal{L}_{\text{score}} \leftarrow \|\lambda_s(t)\nabla_{\boldsymbol{x}} s_\theta(\boldsymbol{x}, t) + \boldsymbol{\epsilon_1}\|_2^2$ (13), update s_θ w.r.t. the loss $\mathcal{L}_{\text{score}}$

Training Stage:

for $i = 1$ to n_{ode} **do**

  Estimate the initial distribution through Gaussian Mixture Model (Appendix A.3).

  for $t = 0$ to $T-2$ **do**

    $\widehat{A}_{t+1} \leftarrow \phi_\theta^{\boldsymbol{v}}\left(\widehat{A}_t, t+1\right), w(\widehat{A}_{t+1}) \leftarrow \phi_\theta^{g}\left(w(\widehat{A}_t), t+1\right)$

    $\mathcal{L}_{\text{Energy}} \leftarrow \mathbb{E}_{\boldsymbol{x}_t \sim p_t} \int_t^{t+1} \left[\frac{1}{2}\|\boldsymbol{v}_\theta\|_2^2 + \frac{1}{2}\|\nabla_{\boldsymbol{x}} s_\theta\|_2^2 - \left(\frac{\sigma^2}{2} + s_\theta\right) g_\theta - \frac{(\sigma^2(t))'}{\sigma^2(t)} s_\theta + \alpha \Psi(g_\theta)\right] w(z)\mathrm{d}z$

    $\mathcal{L}_{\text{Recons}} \leftarrow \mathcal{L}_{\text{Recons}} + \lambda_m M_t + \lambda_d \mathcal{W}_2(\hat{\boldsymbol{w}}^t, \boldsymbol{w}(t))$ (11)

    $\mathcal{L}_{\text{FP}} \leftarrow \|\partial_t p_\theta + \nabla_{\boldsymbol{x}} \cdot (p_\theta \boldsymbol{v}_\theta(\boldsymbol{x}, t)) - g_\theta(\boldsymbol{x}, t) p_\theta\| + \lambda_w \|p_\theta(\boldsymbol{x}, 0) - p_0\|$ (12)

    $\mathcal{L} \leftarrow \mathcal{L}_{\text{Energy}} + \lambda_r \mathcal{L}_{\text{Recons}} + \lambda_f \mathcal{L}_{\text{FP}}$ (9), update $\boldsymbol{v_\theta}$, g_θ and s_θ w.r.t. $\mathcal{L}$

Pretraining Stage

Learn initial velocity, growth and score function

Recons loss (velocity, growth) | Flow matching (score function)

Training Stage

$$\mathcal{L} = \mathcal{L}_{\text{Energy}} + \lambda_r \mathcal{L}_{\text{Recons}} + \lambda_f \mathcal{L}_{\text{FP}}.$$

Energy loss +
Reconstruction loss +
Fokker-Planck constraint

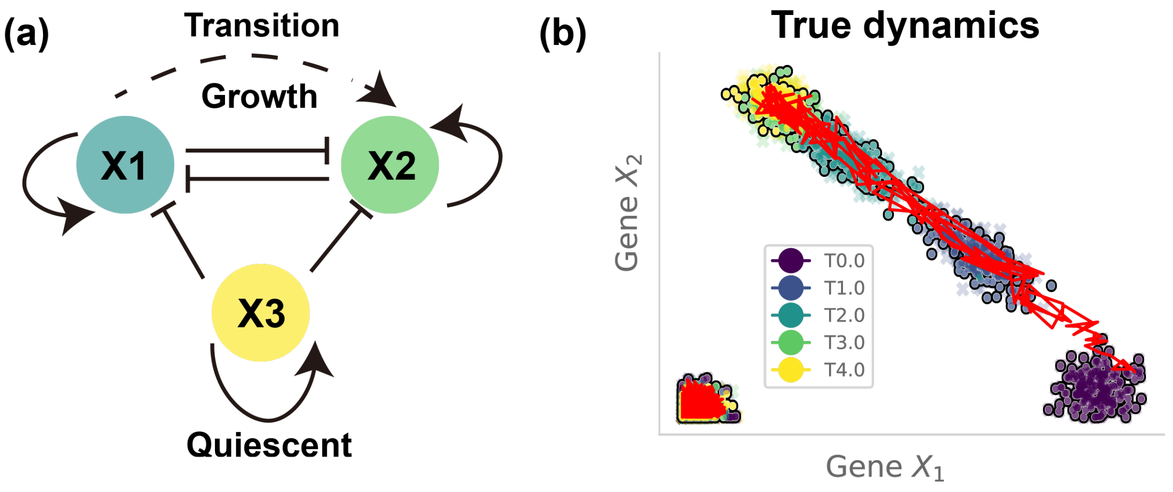
Experimental Results🛠️

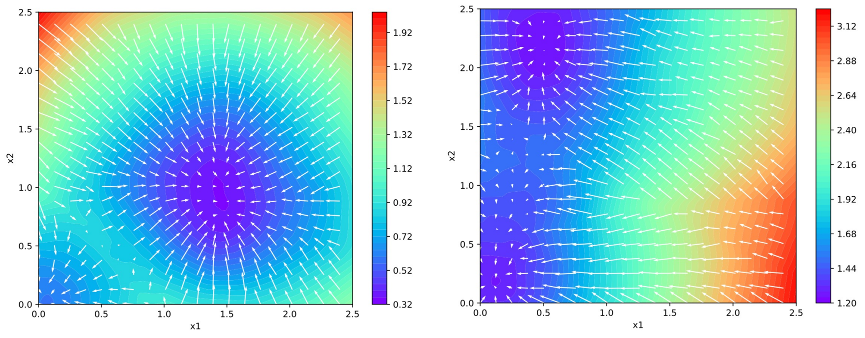
Synthetic gene regulatory network: Unbalanced Stochastic case

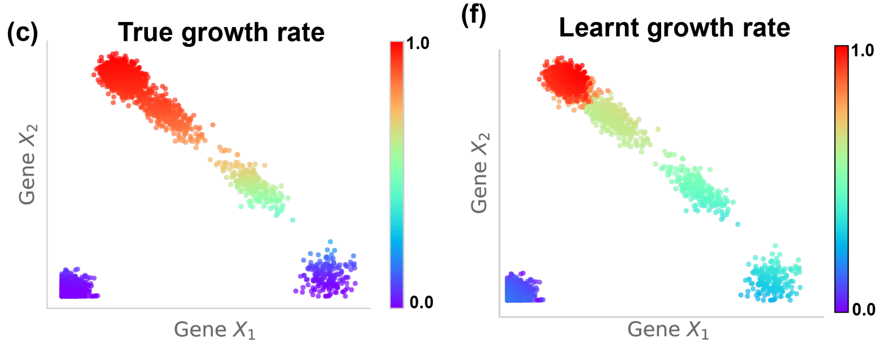
$$\frac{\mathrm{d}X_1}{\mathrm{d}t} = \frac{\alpha_1 X_1^2 + \beta}{1 + \alpha_1 X_1^2 + \gamma_2 X_2^2 + \gamma_3 X_3^2 + \beta} - \delta_1 X_1 + \eta_1 \xi_t,$$

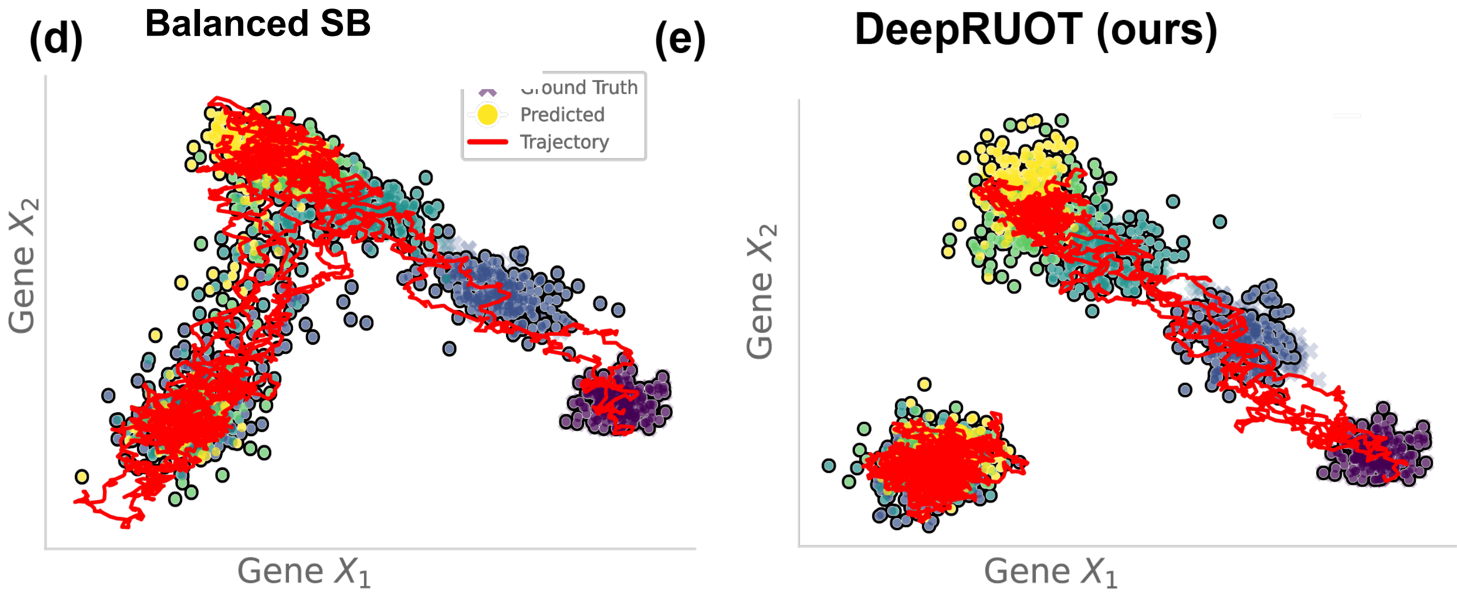
$$\frac{\mathrm{d}X_2}{\mathrm{d}t} = \frac{\alpha_2 X_2^2 + \beta}{1 + \gamma_1 X_1^2 + \alpha_2 X_2^2 + \gamma_3 X_3^2 + \beta} - \delta_2 X_2 + \eta_2 \xi_t,$$

$$\frac{\mathrm{d}X_3}{\mathrm{d}t} = \frac{\alpha_3 X_3^2}{1 + \alpha_3 X_3^2} - \delta_3 X_3 + \eta_3 \xi_t.$$

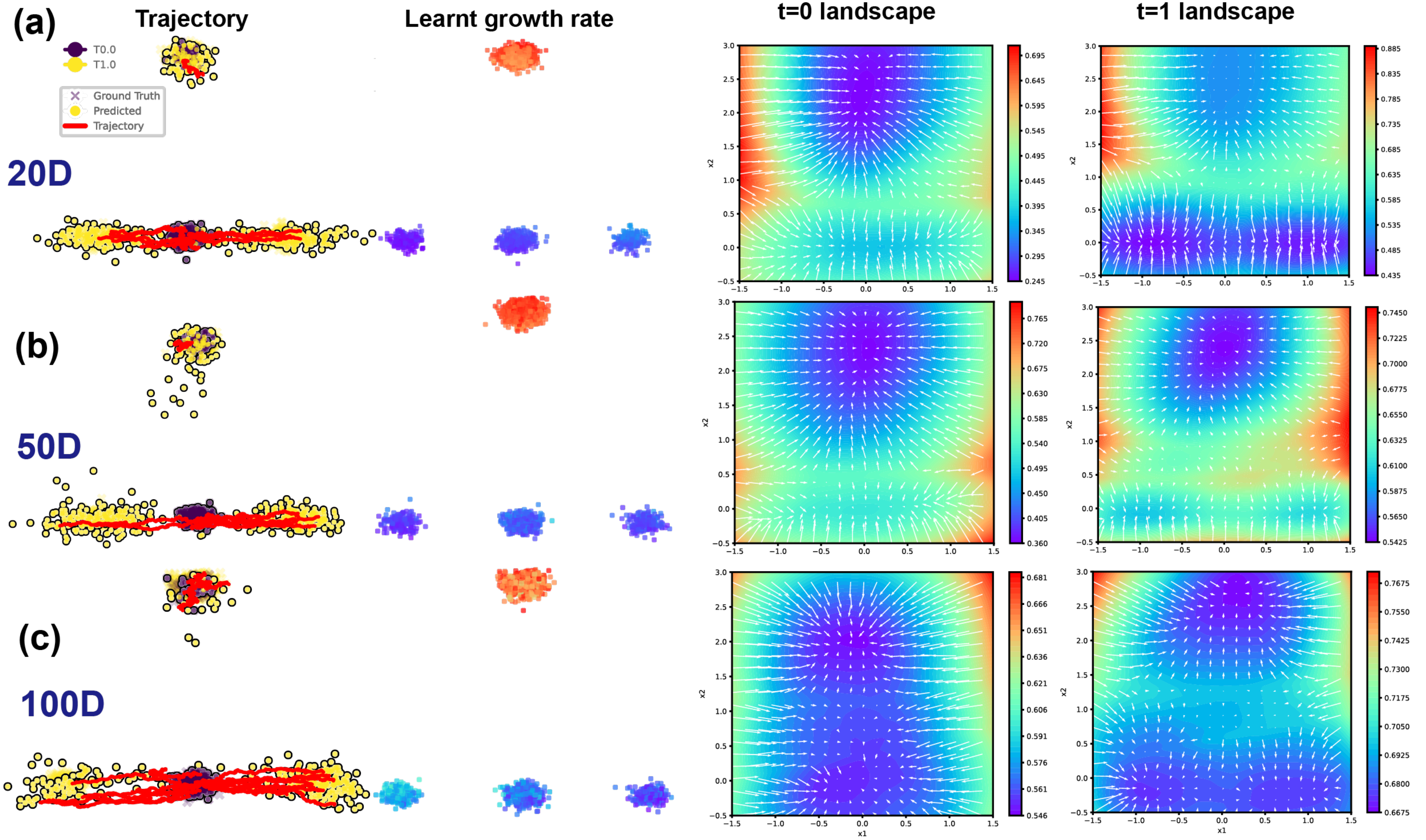






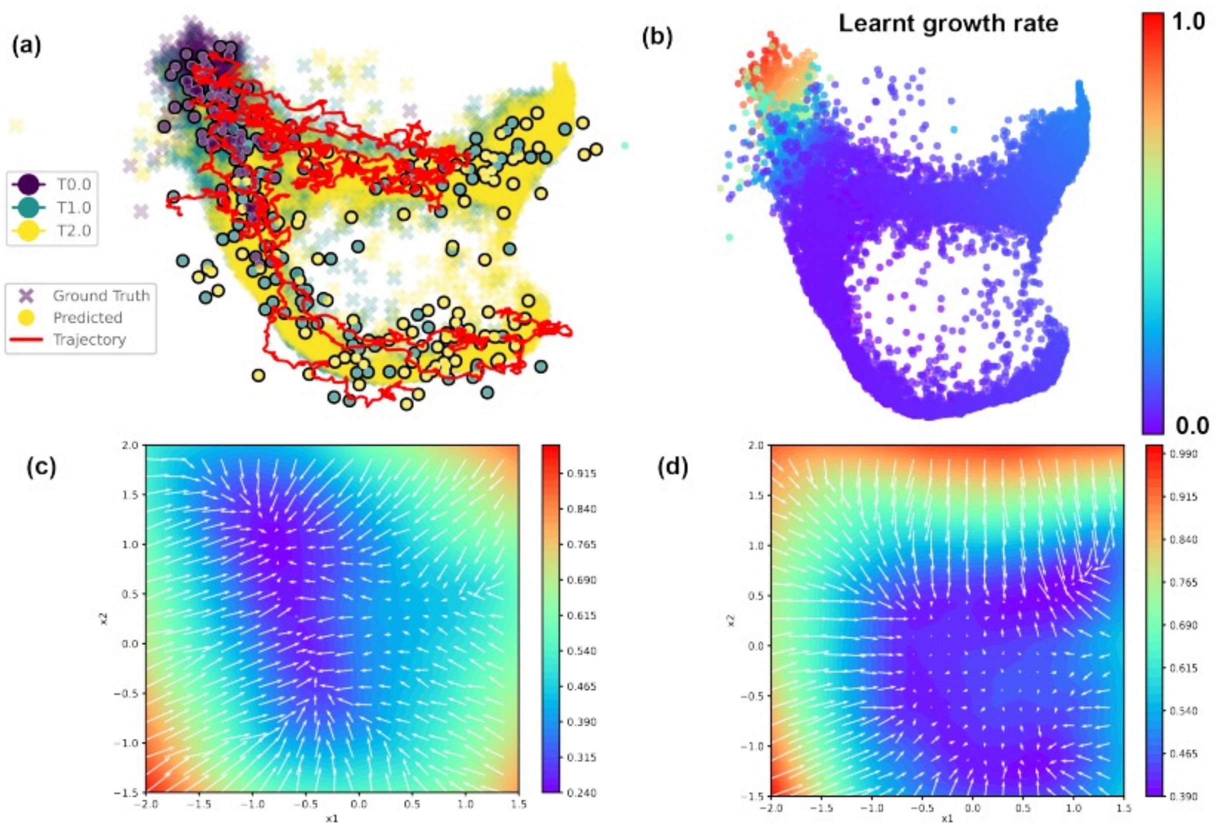


Gaussian mixture model: High dimensional case

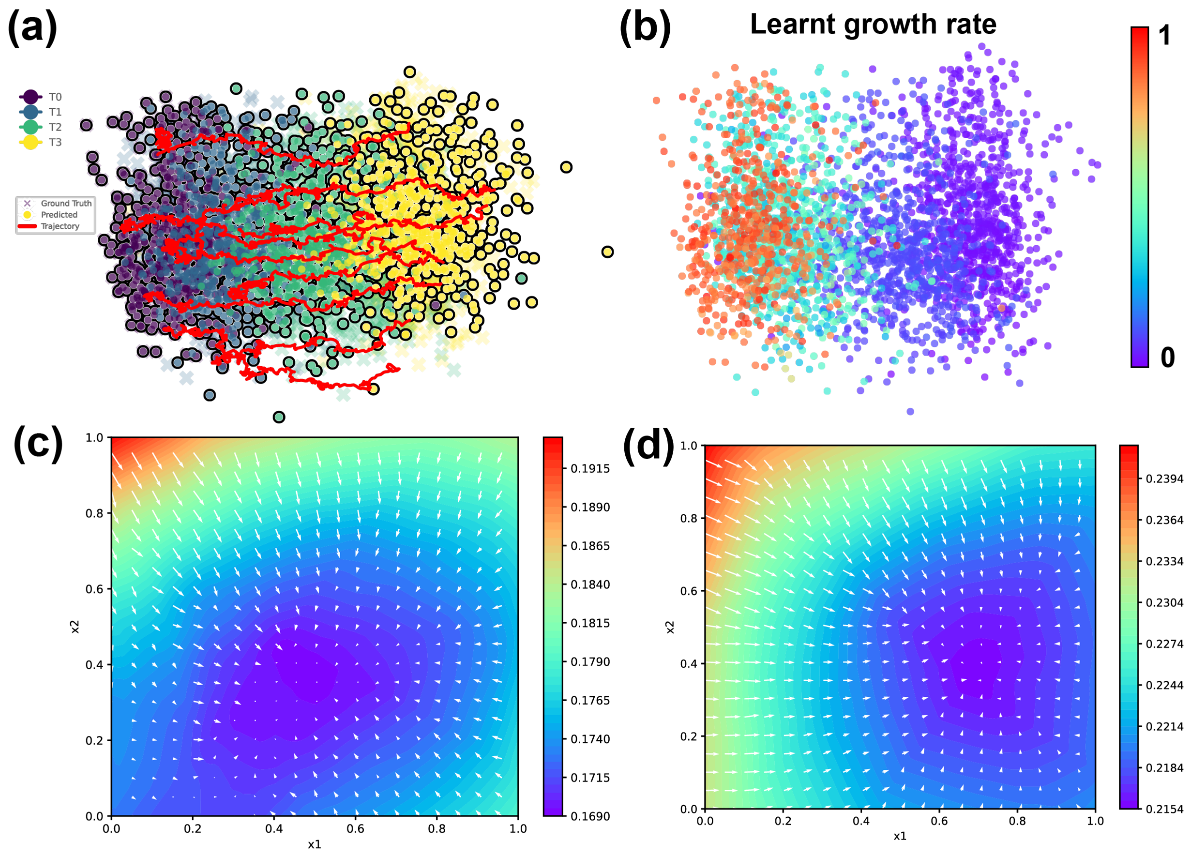


Real scRNA seq data

hematopoiesis



EMT



Summary and Future Directions

- DeepRUOT provides a promising approach to learn unbalanced stochastic dynamics from snapshots datasets
- Math: Inference of noise term in spatiotemporal dynamics?
- Physics: Mechanical effects into the model? Cellular Interactions?
- AI: Learning latent cell state space simultaneously?
- Biology: Integrating with live-imaging data or multi-modal measurements?

Resources

Visit our poster: **#13 (Hall 3 + Hall 2B)**
Today 3 p.m. CST — 5:30 p.m. CST

Project Website

Contact information:

Zhenyi Zhang: zhenyizhang@stu.pku.edu.cn

Tiejun Li: tieli@pku.edu.cn

Peijie Zhou: pjzhou@pku.edu.cn `

Center for Machine Learning Research, Peking University
北京大学国际机器学习研究中心

北京大学数学科学学院
School of Mathematical Sciences,Peking University

AISI 北京科学智能研究院
AI for Science Institute, BEIJING
Empowering Science Enabling Industry