Flow Matching: Principles and Virtual Cell Applications

MA Xiaoheng

HKUST YangLab

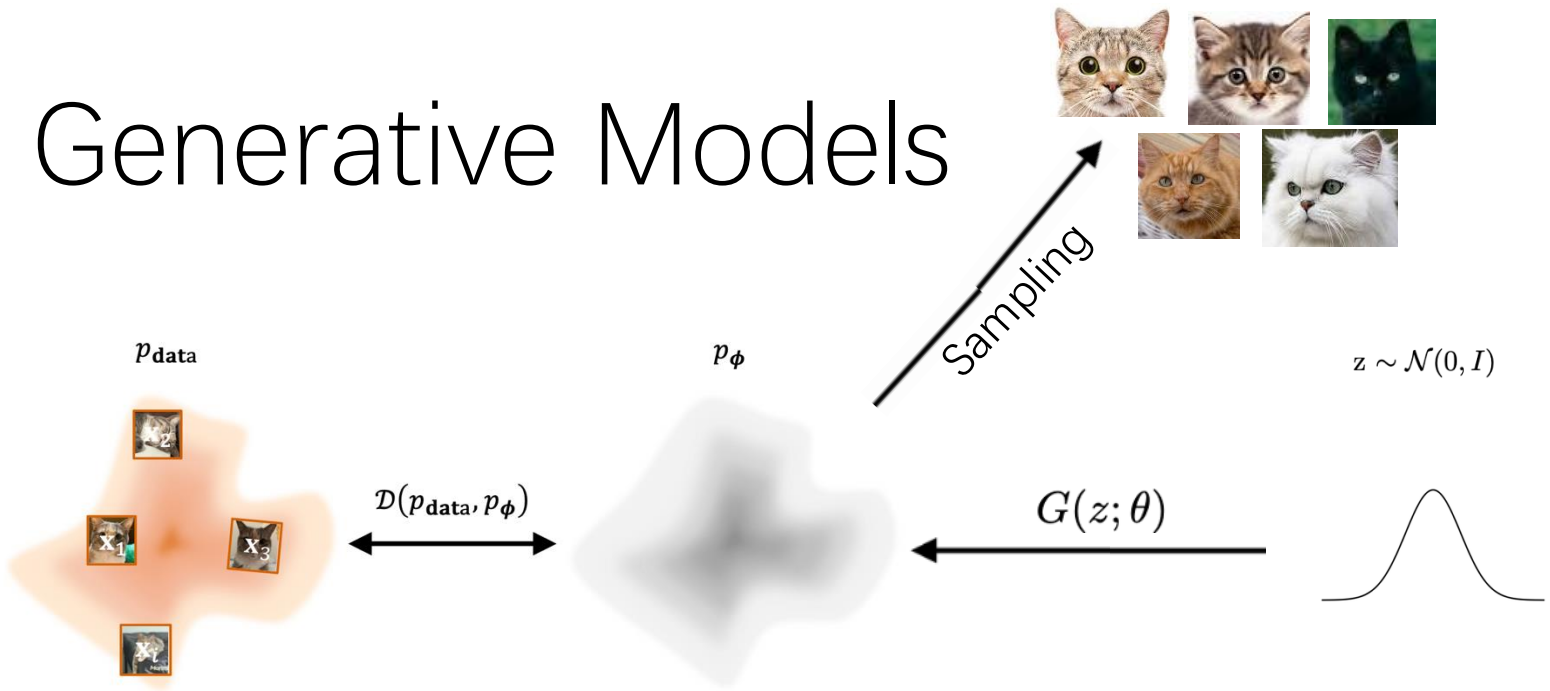
2025.11.5

Content

- 1.Deep Generative Models
- 2.Flow Matching
- 3.Flow Matching in Virtual Cell

1.Deep Generative Models

Deep Generative Models



$$\mathrm{p}_\phi(x) \approx p_{data}(x) \longrightarrow \mathrm{z} \sim \mathcal{N}(0, I) \longrightarrow \hat{x} = G(z;\theta) \longrightarrow \hat{x} \sim p_\phi(x)$$

Creating noise from data is easy; creating data from noise is generative modeling.

---Yang Song, 2020

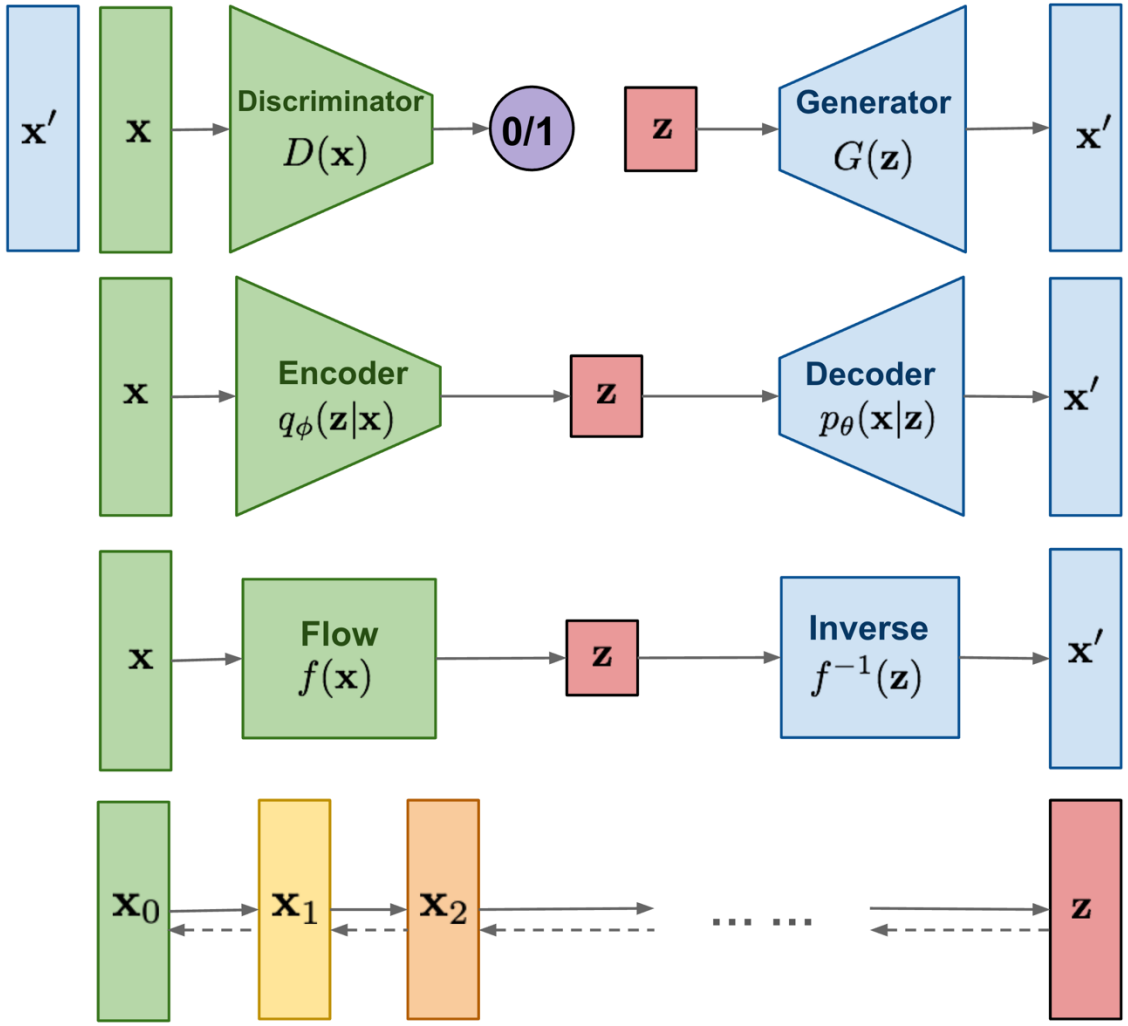
Deep Generative Models

GAN: Adversarial training

VAE: maximize variational lower bound

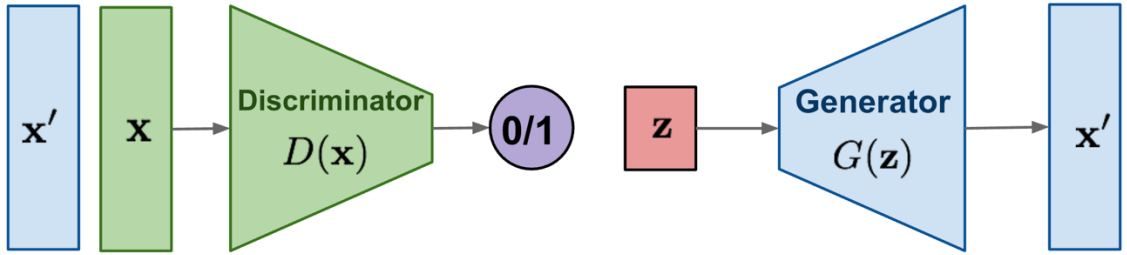
Flow-based models: Invertible transform of distributions

Diffusion models: Gradually add Gaussian noise and then reverse



GAN&VAE

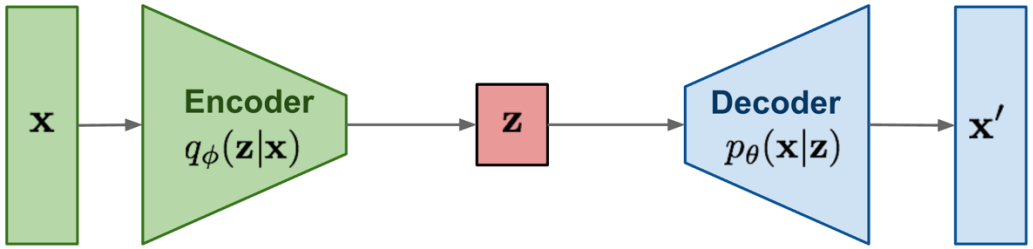
GAN: Adversarial training



$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$$

- Non-convergence
- Mode Collapse
- Vanishing Gradients

VAE: maximize variational lower bound

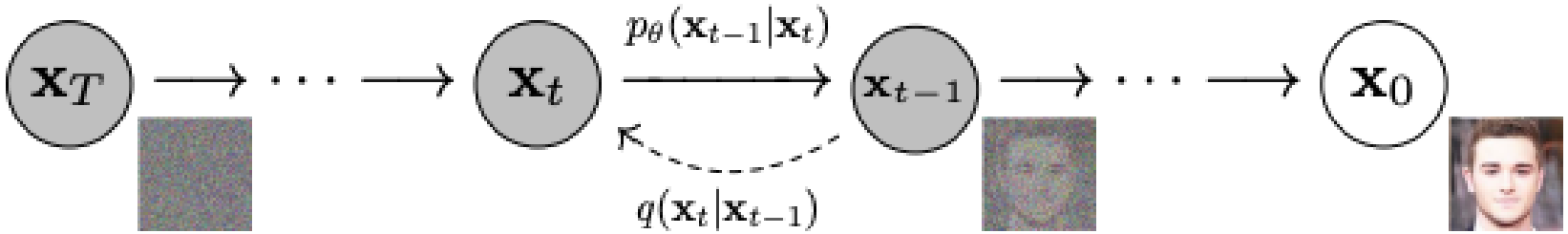


$$\mathcal{L}(\theta,\phi;x) = \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z))$$

- Blurry generation results
- One-shot generation pressure
- Strict distribution assumptions
- Likelihood via ELBO, not exact

Diffusion Models

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\varepsilon_\theta(x_t, t)\right) + \sigma_t z$$



$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\,\varepsilon,\ \varepsilon \sim \mathcal{N}(0, I)$$

- Multi-step refinement
- Simplified loss

$$\mathcal{L}_{\text{simple}} = \mathbb{E}\left[\|\varepsilon - \varepsilon_\theta(x_t, t)\|^2\right]$$

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \ldots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2$$
6: **until** converged

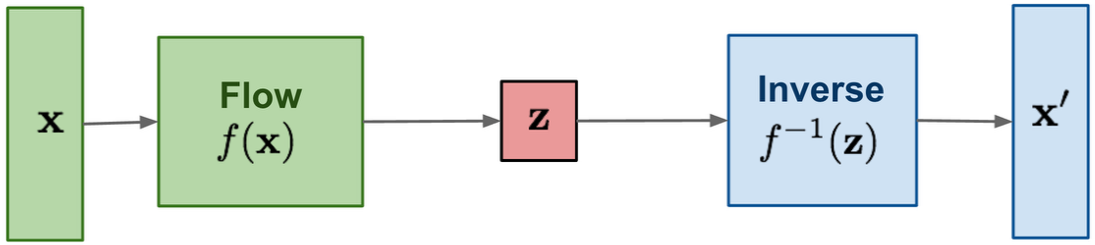
Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

- Strict distribution assumptions
- Likelihood via ELBO, not exact(PF-ODE can do)

Normalizing Flows(NF)

$$f^* = \arg\max_f \sum_{i=1}^{m} \log p_X(x^i)$$

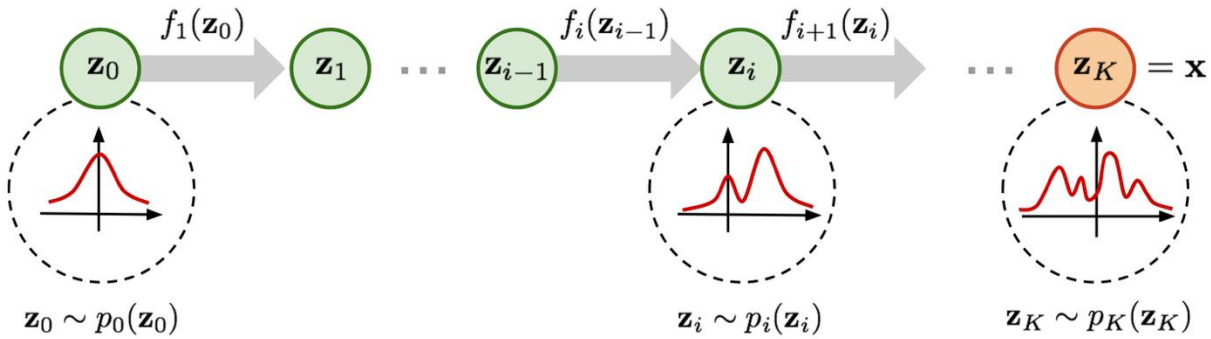


Change of Variables Formula:

$$p(x) = \pi(f^{-1}(x)) \left| \det \frac{\mathrm{d}f^{-1}}{\mathrm{d}x} \right|$$

$$\pi(z) \rightleftarrows x = f(z)$$

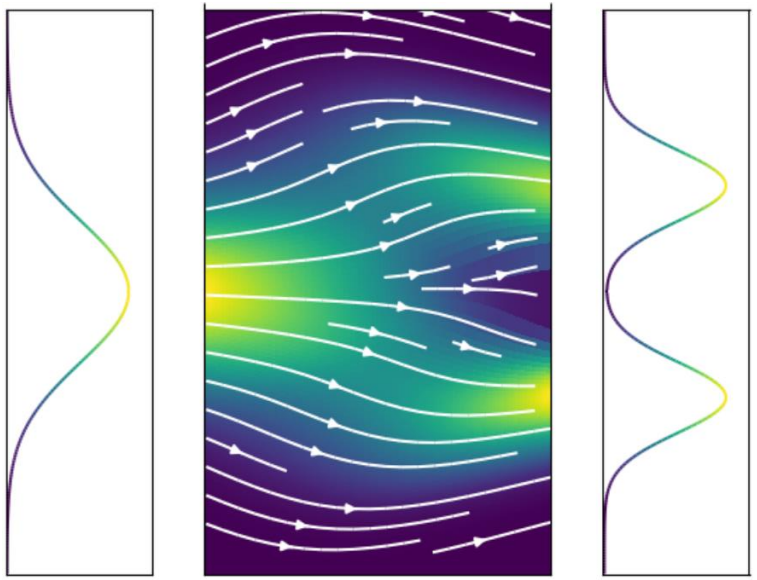
$$\pi(z) = p(f(z)) \left| \det \frac{df}{dz} \right|$$



$$\log p_i(z_i) = \log p_{i-1}(z_{i-1}) - \log \left| \det \frac{\mathrm{d}f_i}{\mathrm{d}z_{i-1}} \right|$$

$$\log p(x) = \log p_K(z_K) = \log \pi_0(z_0) - \sum_{i=1}^{K} \log \left| \det \frac{\mathrm{d}f_i}{\mathrm{d}z_{i-1}} \right|$$

Continuous Normalizing Flows(CNF)

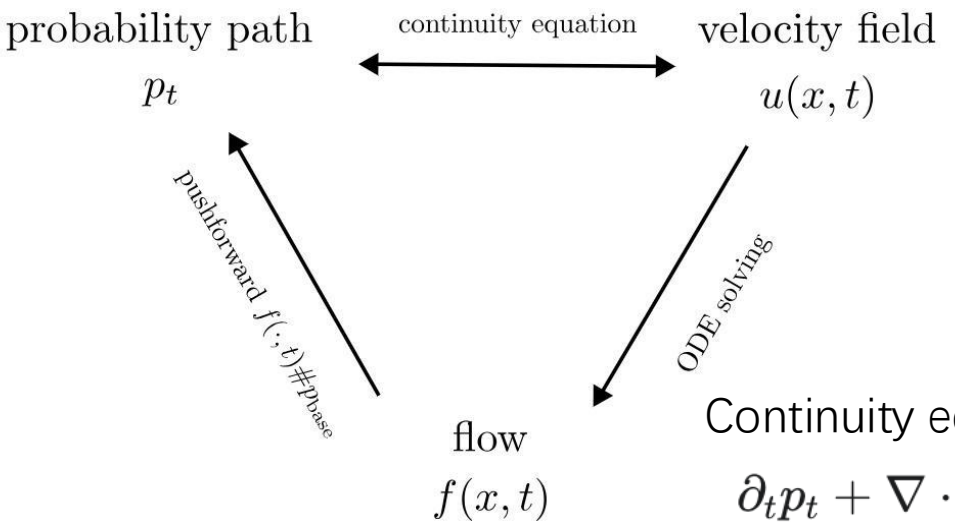


$$\begin{cases} x(0) = x_0 \\ \frac{\mathrm{d}x(t)}{\mathrm{d}t} = u(x(t), t) \quad \forall t \in [0, 1] \end{cases}$$

Flow $f(x, t)$: The solution of the ODE at time t with initial condition $x \sim p_0$.

Probability Path p_t: The probability density obtained by pushing forward the initial density p_0 by the flow $f(\cdot, t)$ at time t, i.e., $p_t = f(\cdot, t)_{\#} p_0$.

Velocity Field $u(x, t)$: The function u in the ODE.



$$\log p_1(x^{(i)}) = \log p_0(x_0) - \int_0^1 \nabla \cdot u_\theta(\cdot, t)(x(t))\mathrm{d}t$$

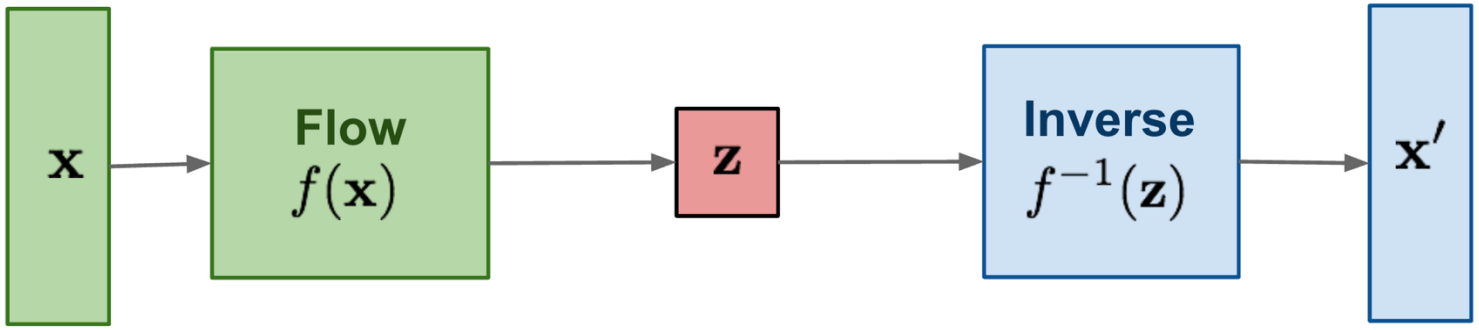
Each MLE step requires an integration, which is solved via simulation.

Continuity equation:

$$\partial_t p_t + \nabla \cdot u_t p_t = 0$$

Flow-based Model

- Exact log-likelihood
- Flexible distributions



- Simulation-based: Normalizing Flows(NF), Continuous Normalizing Flows(CNF). **Costly**
- Simulation-free: Flow Matching. **Efficient**

2.Flow Matching

Training a CNF without Simulation

$$\boldsymbol{d}x = u_t(x)\,\boldsymbol{d}t$$

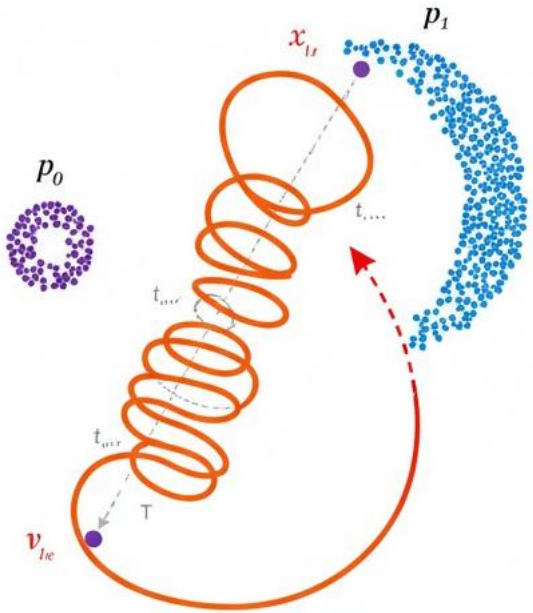
$$\frac{\partial p}{\partial t} = -\nabla \cdot (p_t u_t)$$

$$\longrightarrow \quad \mathcal{L}_{\mathbf{FM}}(\theta) := \mathbb{E}_{t\sim\mathcal{U}(0,1),x\sim p_t(x)}\|v_\theta(\boldsymbol{t},x) - u_t(x)\|^2$$

Traditional CNF:

Simulation-Based Training

Modeling Process Empirically

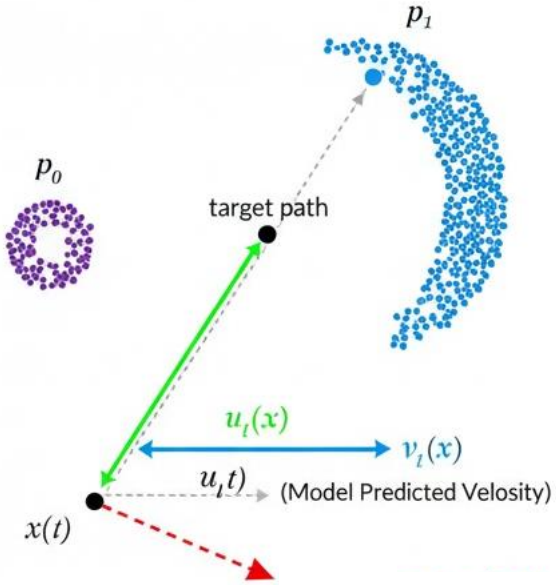


Backpragation(BP) through ODE Solver

Flow Matching:

Simulation-Free Training

Equivalent training objective, regress out vector field directly



Backpragation(BP) through v_t

Flow Matching: Conditional FM

$$q(z) = q(x_0, x_1) \qquad p_t(\mathbf{x}) = \int p_t(\mathbf{x}|\mathbf{z})q(\mathbf{z})d\mathbf{z}$$

$$\mathcal{L}_{\mathrm{FM}}(\theta) := \mathbb{E}_{t\sim\mathcal{U}(0,1),\mathbf{x}\sim p_t(\mathbf{x})}\|v_\theta(t,\mathbf{x}) - \boxed{u_t(\mathbf{x})}\|^2$$

$$u_t(\mathbf{x}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x},t)}[u_t(\mathbf{x}|\mathbf{z})] = \boxed{\int u_t(\mathbf{x}|\mathbf{z})\frac{p_t(\mathbf{x}|\mathbf{z})q(\mathbf{z})}{p_t(\mathbf{x})}d\mathbf{z}}$$

Intractable

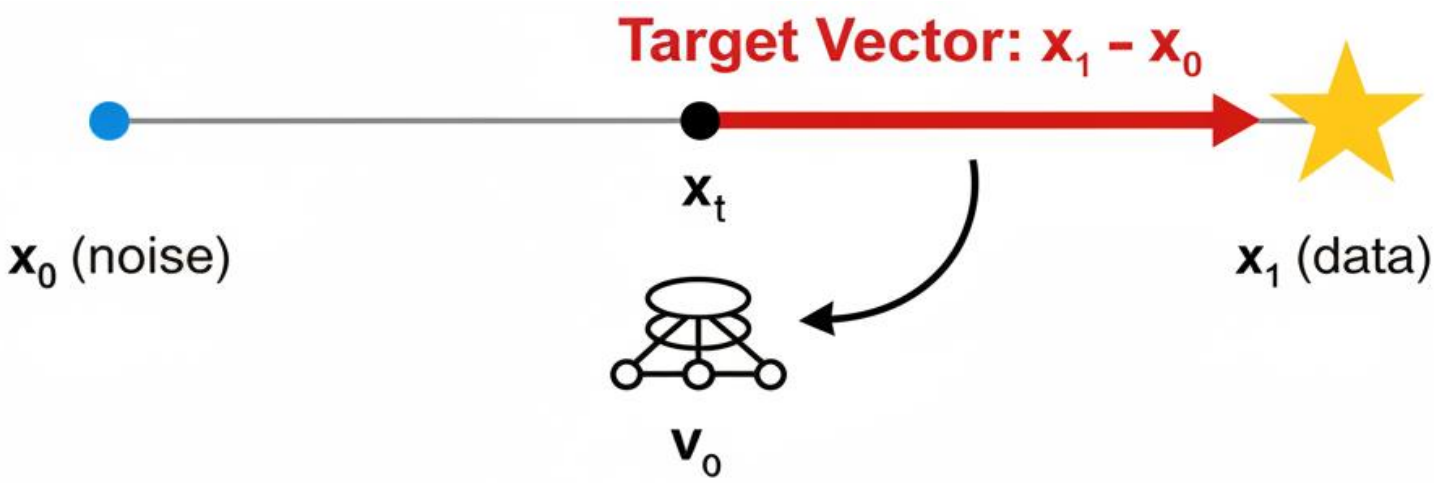
$$\mathcal{L}_{\mathrm{CFM}}(\theta) := \mathbb{E}_{t,q(\mathbf{z}),p_t(\mathbf{x}|\mathbf{z})}\|v_\theta(t,\mathbf{x}) - \boxed{u_t(\mathbf{x}|\mathbf{z})}\|^2$$

Tractable

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

$$\nabla_\theta \mathcal{L}_{\mathrm{FM}}(\theta) = \nabla_\theta \mathcal{L}_{\mathrm{CFM}}(\theta)$$

Flow Matching: I-CFM



$$x_0 \sim p_0 \qquad x_1 \sim p_1$$

$$q(z) = q(x_0, x_1) = q_0(x_0)q_1(x_1)$$

$$\longrightarrow$$

$$\mu_t = tx_1 + (1-t)x_0$$

$$p_t(x|z) = \mathcal{N}(x|tx_1 + (1-t)x_0, \sigma^2 I)$$

$$u_t(x|z) = x_1 - x_0$$

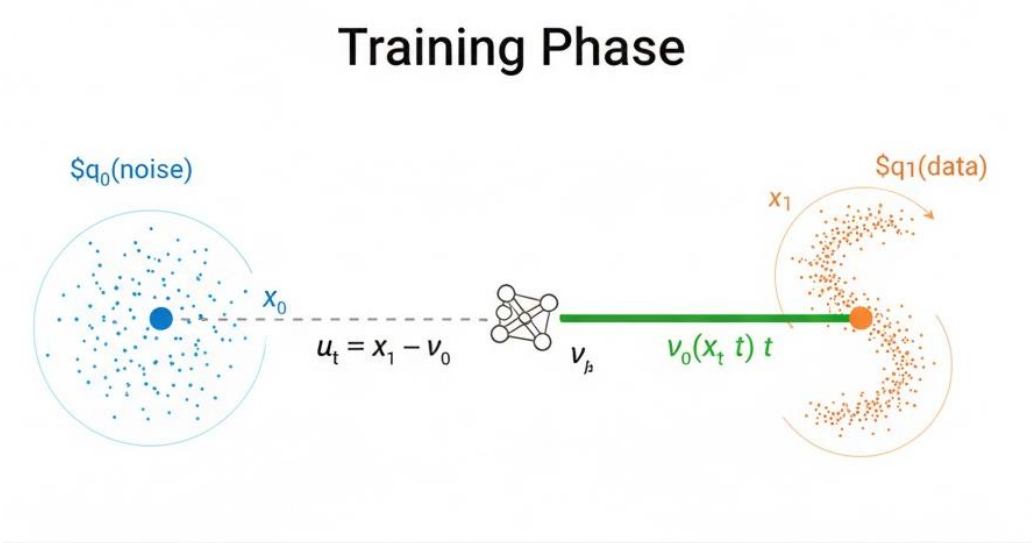
Flow Matching: I-CFM

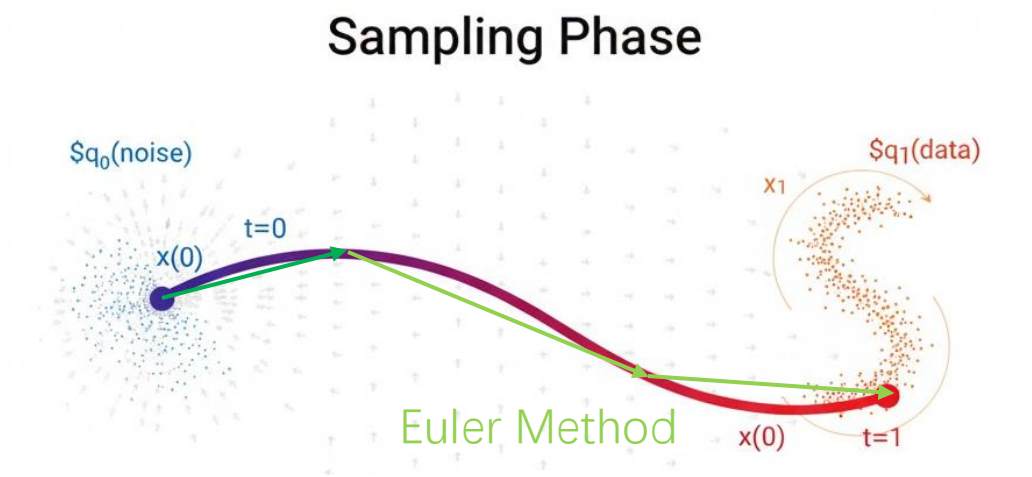
Algorithm 1 Flow Matching: Training (I-CFM)

Input: Empirical distributions q_0 (noise), q_1 (data), bandwidth σ, batchsize b, initial network v_θ.
while not converged **do**
  Sample batches of size b i.i.d. from the datasets:
  $x_0 \sim q_0(x_0); \quad x_1 \sim q_1(x_1)$
  $t \sim \mathcal{U}(0,1)$
  $\mu_t \leftarrow t x_1 + (1-t) x_0$
  $x \sim \mathcal{N}(\mu_t, \sigma^2 I)$
  $\mathcal{L}_{\mathrm{CFM}}(\theta) \leftarrow \|v_\theta(t,x) - (x_1 - x_0)\|^2$
  $\theta \leftarrow \mathrm{Update}(\theta, \nabla_\theta \mathcal{L}_{\mathrm{CFM}}(\theta))$
end while
return trained network v_θ

Algorithm 2 Flow Matching: Sampling (Inference)

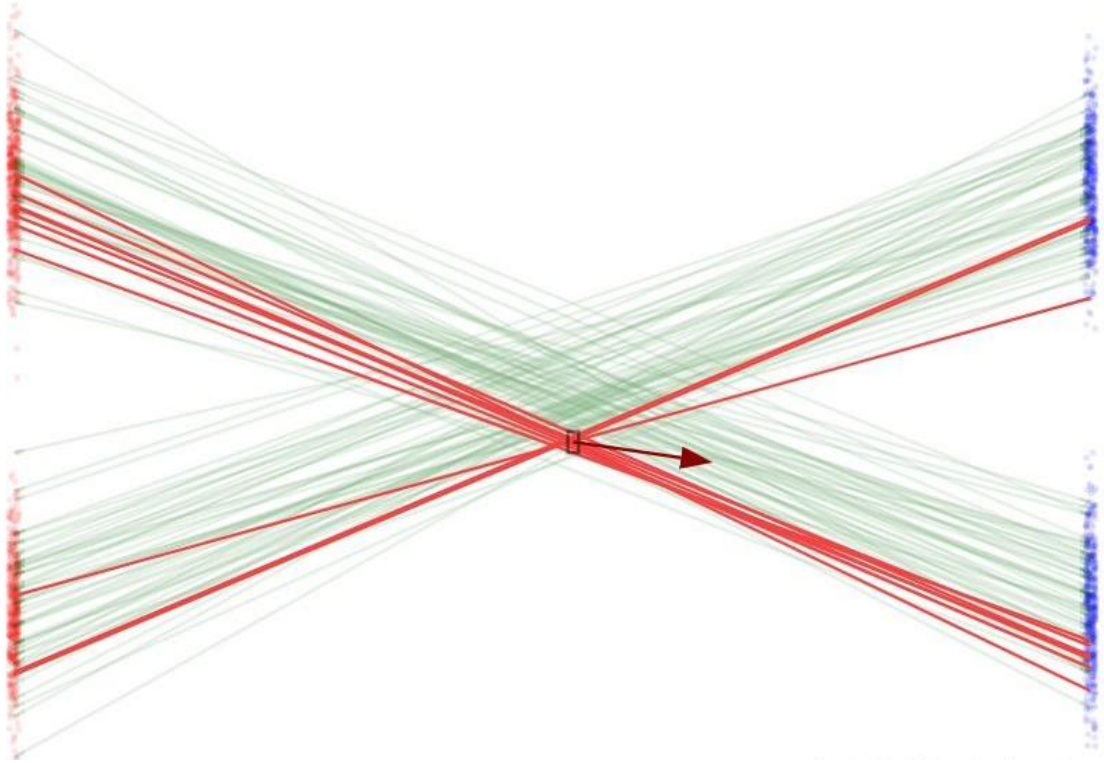
Input: Trained vector field network v_θ, noise distribution q_0, a numerical ODE solver (e.g., Euler, RK45).
/* To generate a new sample */
Sample initial noise: $x(0) \sim q_0(x)$
Solve the initial value problem (IVP) from $t = 0$ to $t = 1$:
  $\frac{dx}{dt} = v_\theta(t, x(t))$
Let $x(1)$ be the solution at $t = 1$.
return generated sample $x(1)$

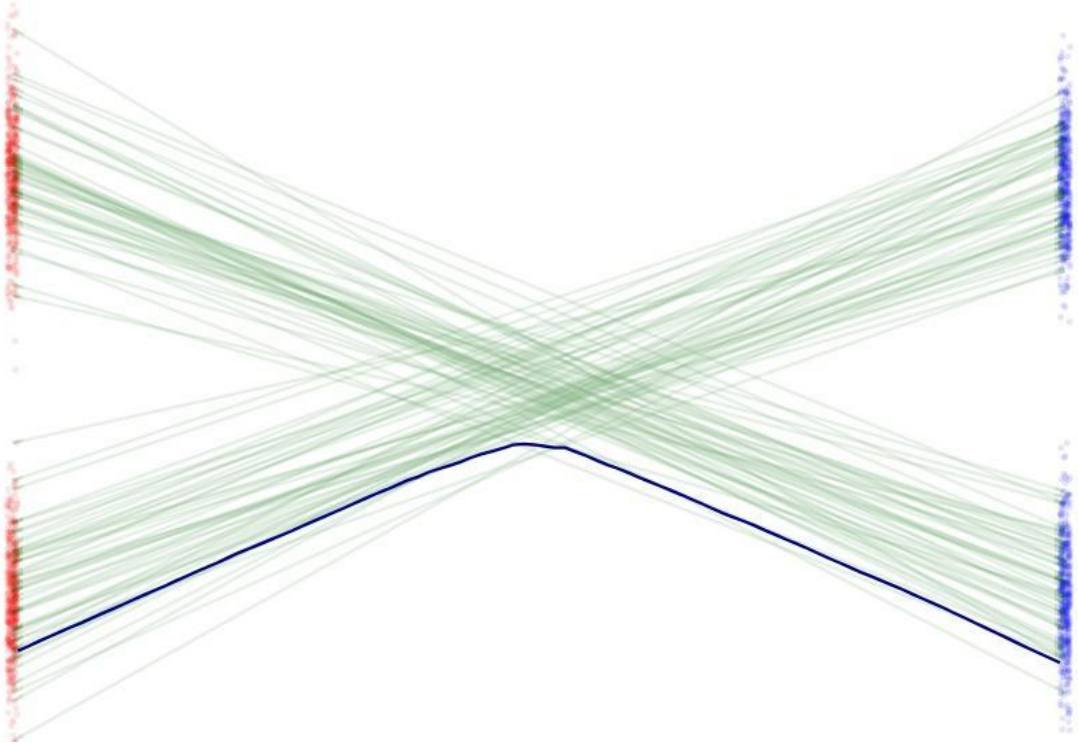




$$x(t + \Delta t) \approx x(t) + v_\theta(t, x(t)) \cdot \Delta t$$

Flow Matching: Path Straight?

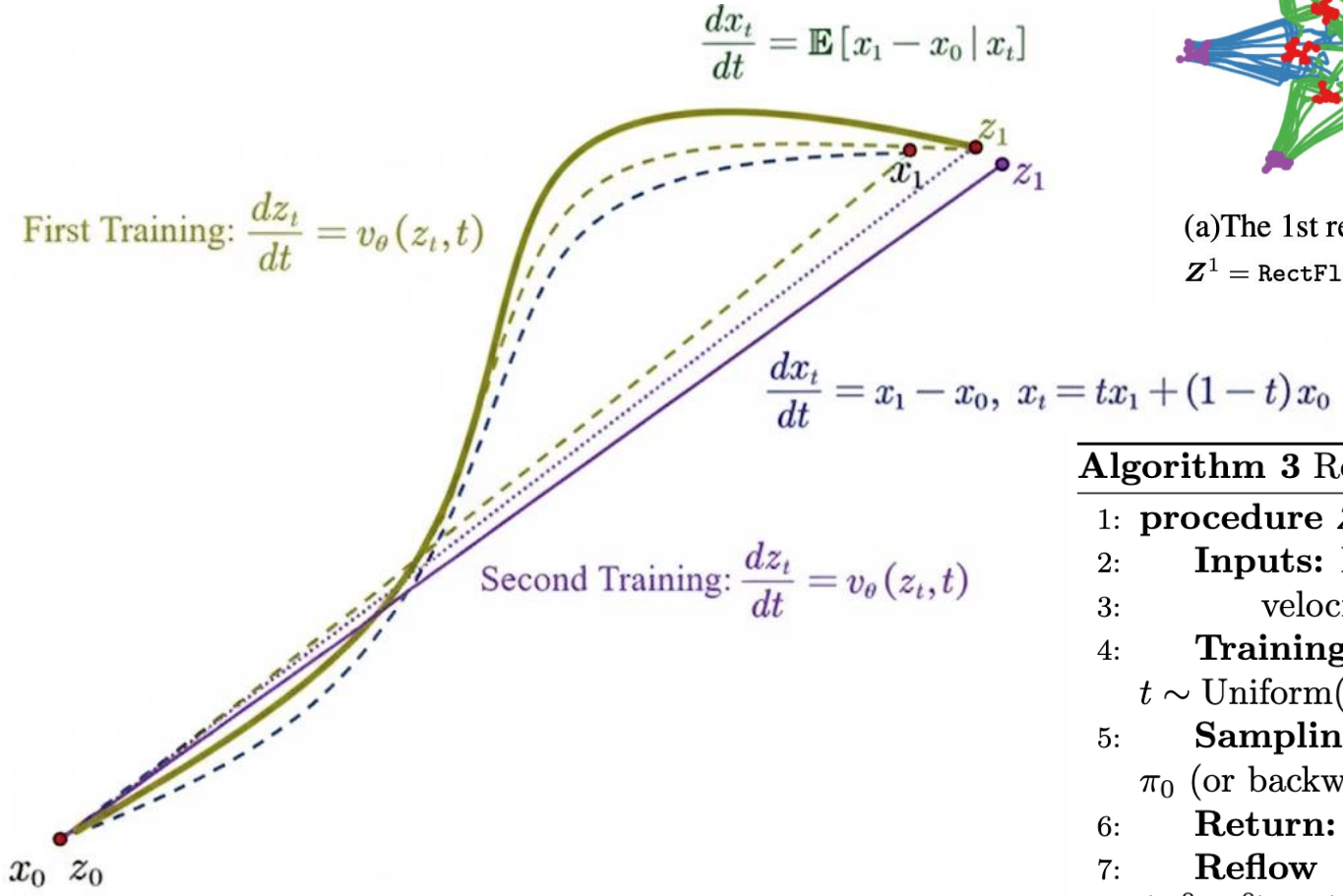


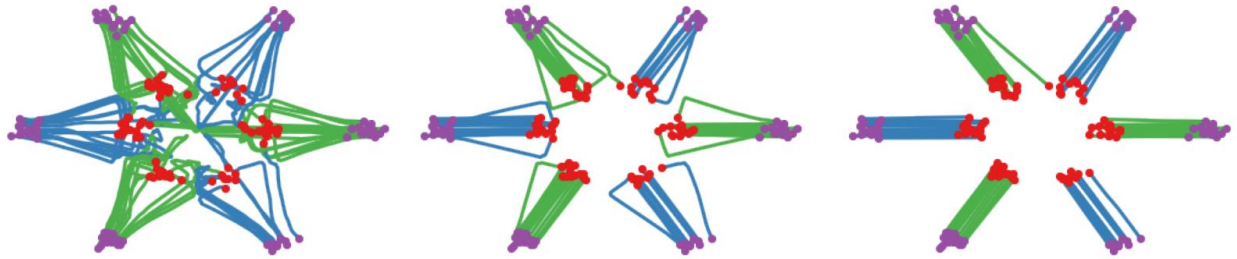


$$u_t(x|z) = x_1 - x_0$$

$$u_t(x) = E_{p_t(z|x)}[u_t(x|z)]$$

Rectified FM(ICLR 2023)





(a)The 1st rectified flow $\boldsymbol{Z}^1$
$\boldsymbol{Z}^1 = \texttt{RectFlow}((X_0, X_1))$

(b) Reflow $\boldsymbol{Z}^2$
$\boldsymbol{Z}^2 = \texttt{RectFlow}((Z_0^1, Z_1^1))$

(c) Reflow $\boldsymbol{Z}^3$
$\boldsymbol{Z}^3 = \texttt{RectFlow}((Z_0^2, Z_1^2))$

Algorithm 3 Rectified Flow

procedure $Z = \textsc{RectFlow}((X_0, X_1))$
  Inputs: Draws from a coupling (X_0, X_1) of π_0 and π_1;
    velocity model $v_\theta : R^d \to R^d$ with parameter θ.
  Training: $\hat{\theta} = \arg\min_\theta E[||X_1 - X_0 - v_\theta(tX_1 + (1-t)X_0, t)||^2]$, with $t \sim \text{Uniform}([0,1])$.
  Sampling: Draw (Z_0, Z_1) following $dZ_t = v_{\hat{\theta}}(Z_t, t)dt$ starting from $Z_0 \sim \pi_0$ (or backwardly $Z_1 \sim \pi_1$).
  Return: $Z = \{Z_t : t \in [0,1]\}$.
  Reflow (optional): $Z^{k+1} = \text{RectFlow}((Z_0^k, Z_1^k))$, starting from $(Z_0^0, Z_1^0) = (X_0, X_1)$.
end procedure

OT-CFM(TMLR 2024)

$$q(z) := \pi(x_0, x_1) \qquad \pi^* = \underset{\pi \in \Pi(p_0, p_1)}{\operatorname{argmin}} \int_{x_0, x_1} ||x_0 - x_1||^2 \pi(x_0, x_1) dx_0 dx_1$$

Algorithm 4 Minibatch OT Conditional Flow Matching (OT-CFM)

Input: Empirical or samplable distributions q_0, q_1, bandwidth σ, batch size b, initial network v_θ.

while Training **do do**

  / Sample batches of size b i.i.d. from the datasets */*

  $x_0 \sim q_0(x_0);\ x_1 \sim q_1(x_1)$

  $\pi \leftarrow \mathrm{OT}(x_1, x_0)$

  $(x_0, x_1) \sim \pi$

  $t \sim \mathcal{U}(0, 1)$

  $\mu_t \leftarrow t x_1 + (1 - t) x_0$

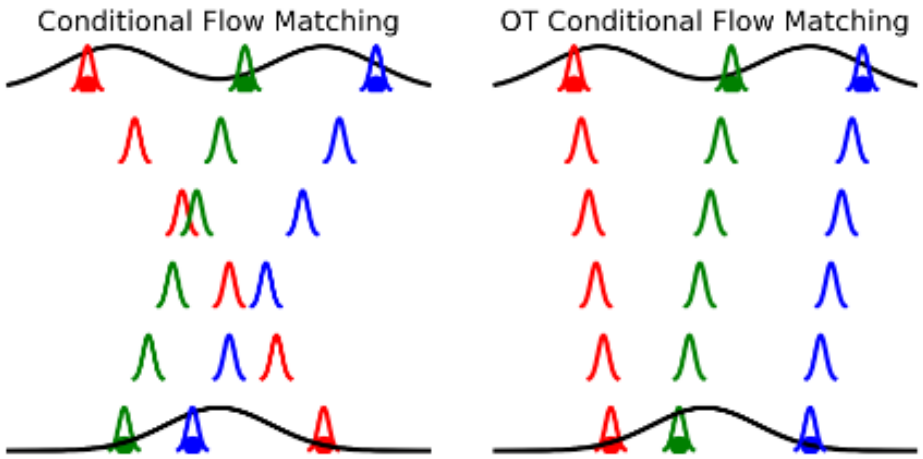
  $\boldsymbol{x} \sim \mathcal{N}(\mu_t, \sigma^2 I)$

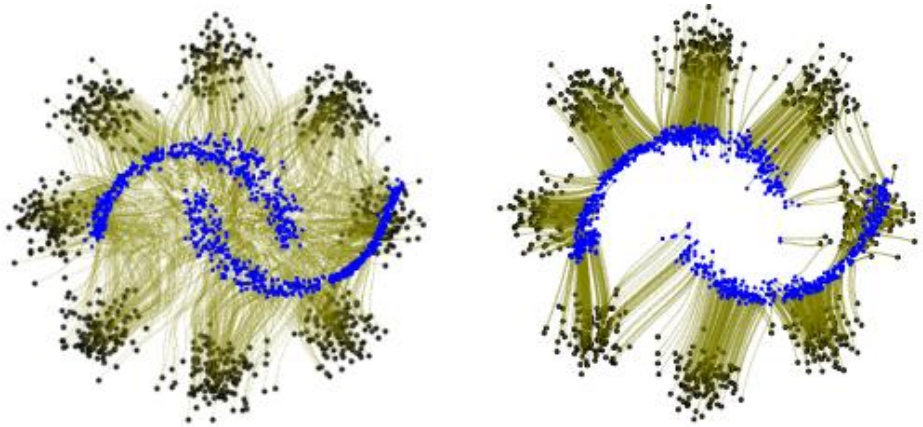
  $\mathcal{L}_{\mathrm{CFM}}(\theta) \leftarrow ||v_\theta(t, \boldsymbol{x}) - (x_1 - x_0)||^2$

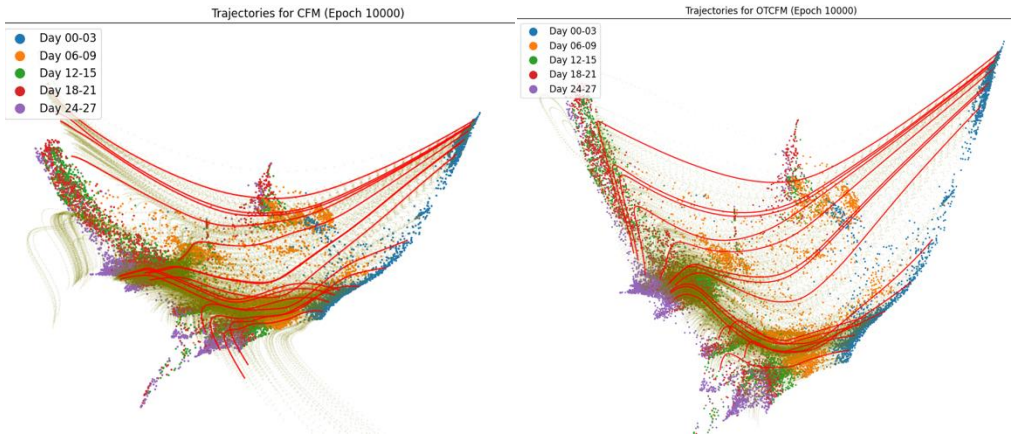
  $\theta \leftarrow \mathrm{Update}(\theta, \nabla_\theta \mathcal{L}_{\mathrm{CFM}}(\theta))$

end while

return v_θ







Diffusion & Flow Matching

Diffusion:

Flow Matching:

$$v_t(x) = u_t(x) - \frac{1}{2}\sigma_t^2 \nabla_x \log p(x,t)$$

- SDE $dX_t = u_t(X_t)dt + \sigma_t dW_t$

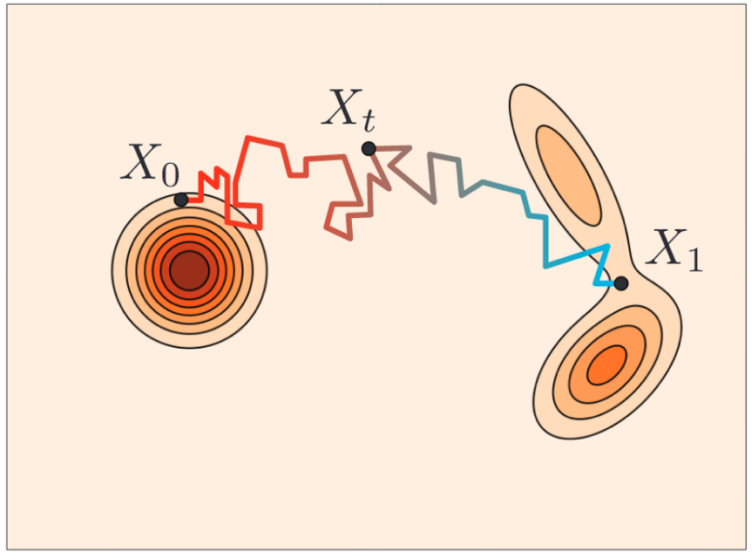
$$\partial_t p(x,t) = -\nabla_x \cdot \left(\left[u_t(x) - \frac{1}{2}\sigma_t^2 \nabla_x \log p(x,t)\right] p(x,t)\right)$$

$$\longleftrightarrow$$

- ODE $dX_t = u_t(X_t)dt$

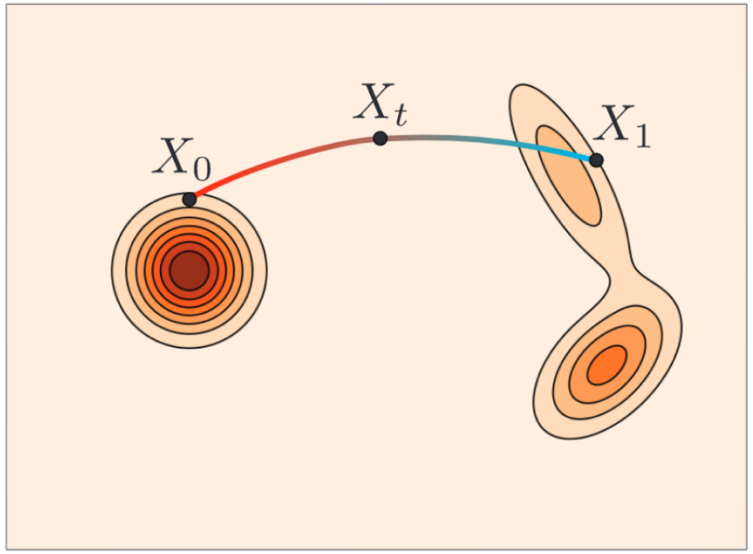
$$\partial_t p(x,t) = -\nabla_x \cdot \left(u_t(x)p(x,t)\right) + \tfrac{1}{2}\sigma_t^2 \Delta_x p(x,t)$$

$$\partial_t p(x,t) = -\nabla_x \cdot \left(u_t(x)p(x,t)\right)$$



PF-ODE:

$$\boxed{dX_t = \left(u_t(X_t) - \frac{1}{2}\sigma_t^2 \nabla_x \log p(x,t)\right) dt}$$



Diffusion & Flow Matching

- Training: Diffusion = FM with Gaussian source + Diffusion Path

$$p_t(x|x_1) = \mathcal{N}(x|\mu_t(x_1), \sigma_t(x_1)^2 I)$$

$$\mu_t(x_1) = \alpha_{1-t} x_1$$

$$\sigma_t(x_1) = \sqrt{1 - \alpha_{1-t}^2}$$

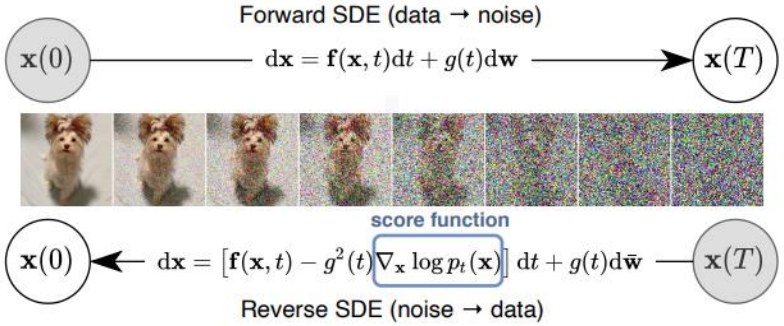
$$p_t(x|x_1) = \mathcal{N}(x|\alpha_{1-t} x_1, (1 - \alpha_{1-t}^2) I)$$

- Sampling: Forward SDE, Reverse SDE, and PF-ODE share same marginal distributions $p_t(x)$.

PF-ODE: $$\boxed{dX_t = \left(u_t(X_t) - \frac{1}{2}\sigma_t^2 \nabla \log p_t(X_t)\right) dt}$$

Different Path

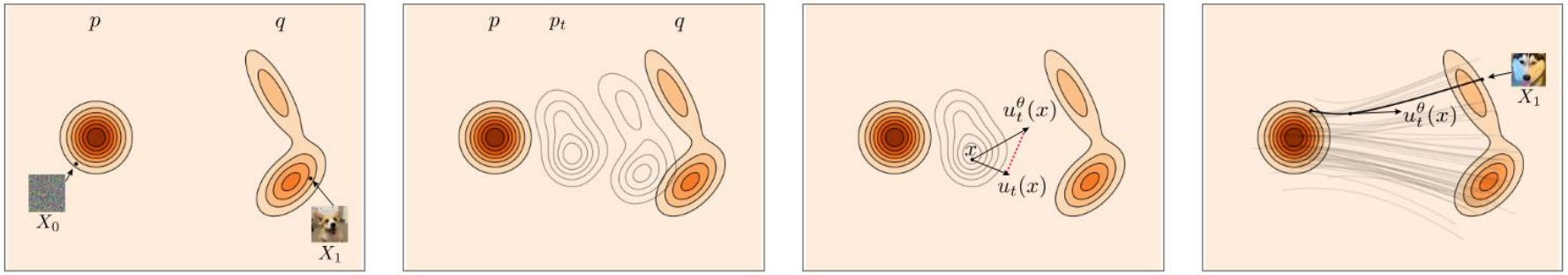
Same Marginal Distribution



Song Y. et al ICLR(2021)

Flow Matching

- Straighter deterministic path → few-step ODE sampling; diffusion also supports fast samplers (DDIM, DPM-Solver).
- Exact likelihood via CNF formulation; diffusion supports exact likelihood via PF-ODE (costly).
- Flexible Probability Paths.
- Versatile Data Priors.



(a) Data. **(b)** Path design. **(c)** Training. **(d)** Sampling.

Flow Matching Goes Mainstream

From Winding Diffusion to a Straight, Solvable Flow—Fewer Steps, Lower Latency, Stronger Alignment.

Stable Diffusion 3 — FM-based Rectified Flow + MMDiT; at-scale results show stronger text fidelity and layout stability.

FLUX.1 — Rectified Flow Transformer; unifies generation + editing for low-latency, multi-turn edits with identity consistency.

Lower cost, faster responses, better control—Flow Matching wins the cost–latency–quality triangle.

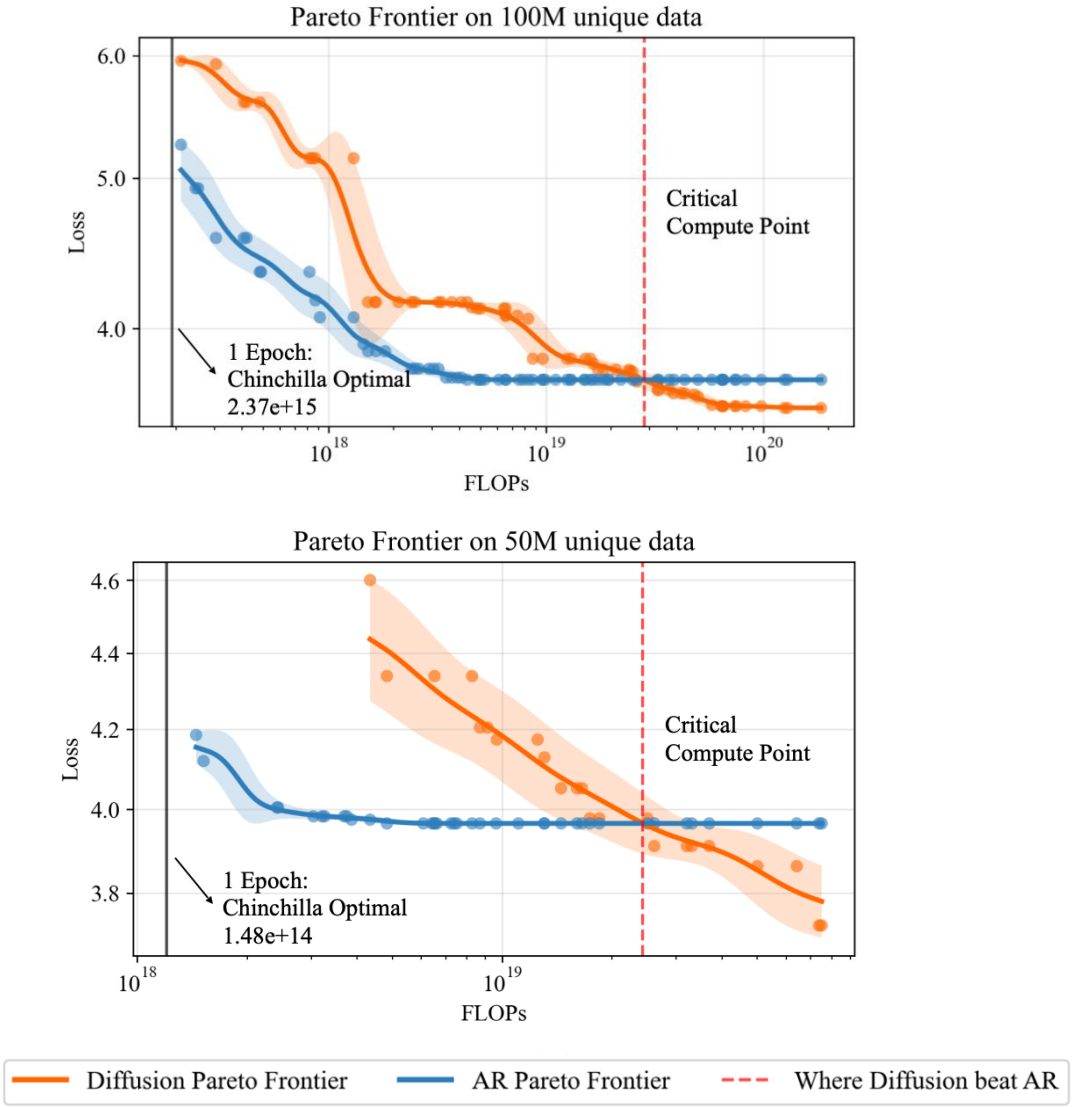
3.Flow Matching in Virtual Cell

Diffusion beats AR in Data-Constrained Setting

- Data Constrained: Diffusion
- Computation Constrained: AR

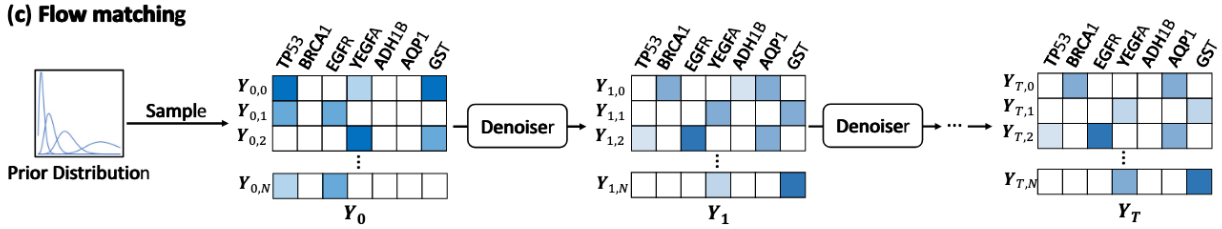
Diffusion Beats Autoregressive in Data-Constrained Settings

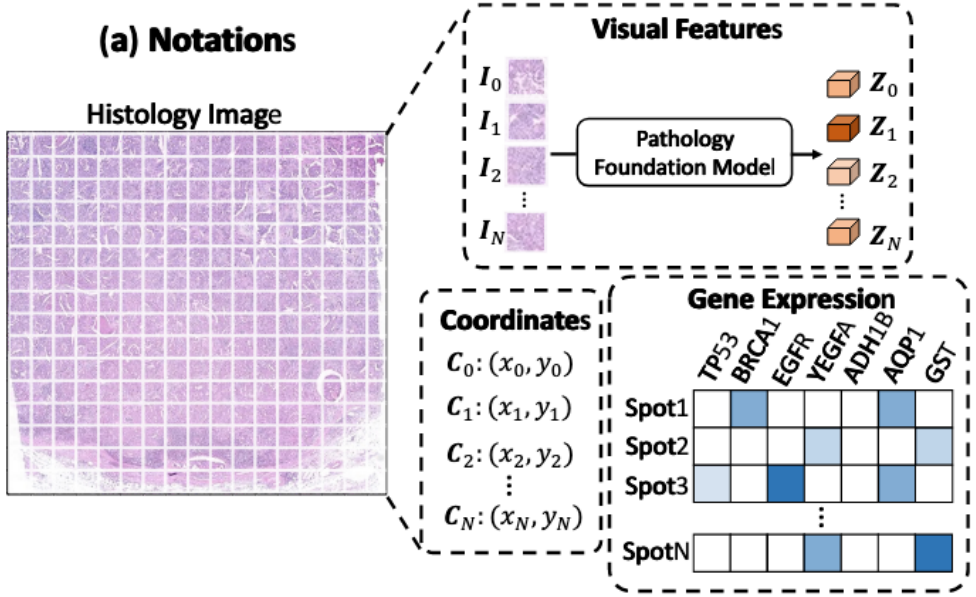
Mihir Prabhudesai* Carnegie Mellon University
Mengning Wu* Carnegie Mellon University
Amir Zadeh Lambda
Katerina Fragkiadaki Carnegie Mellon University
Deepak Pathak Carnegie Mellon University



Spatial & Morphology Analysis

- STFlow(Liu T. et al, ICML 2025)



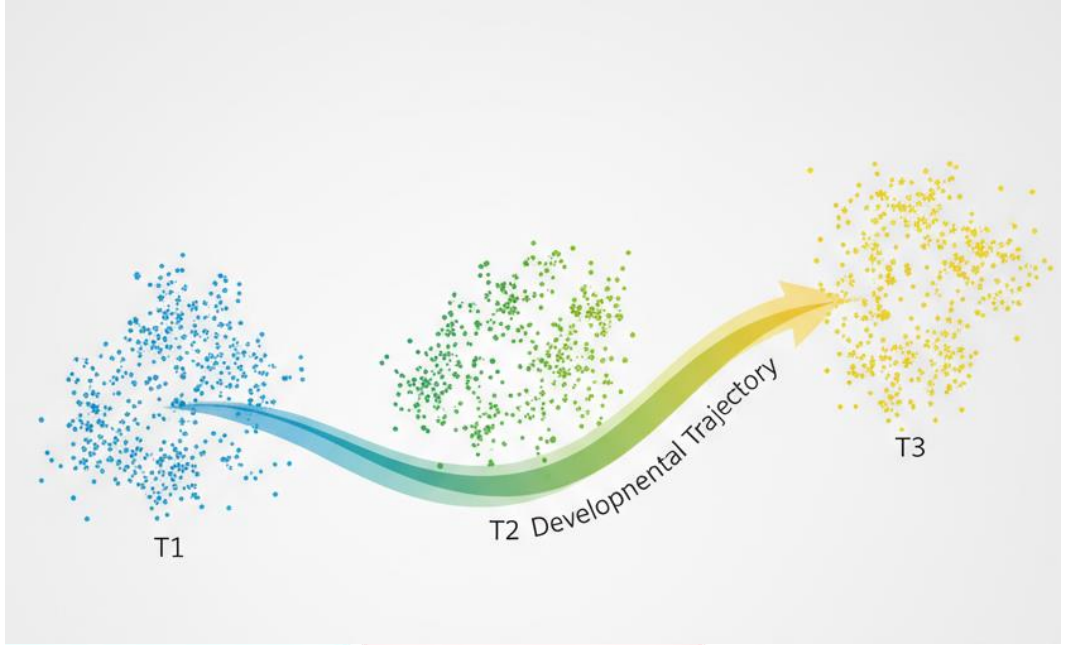


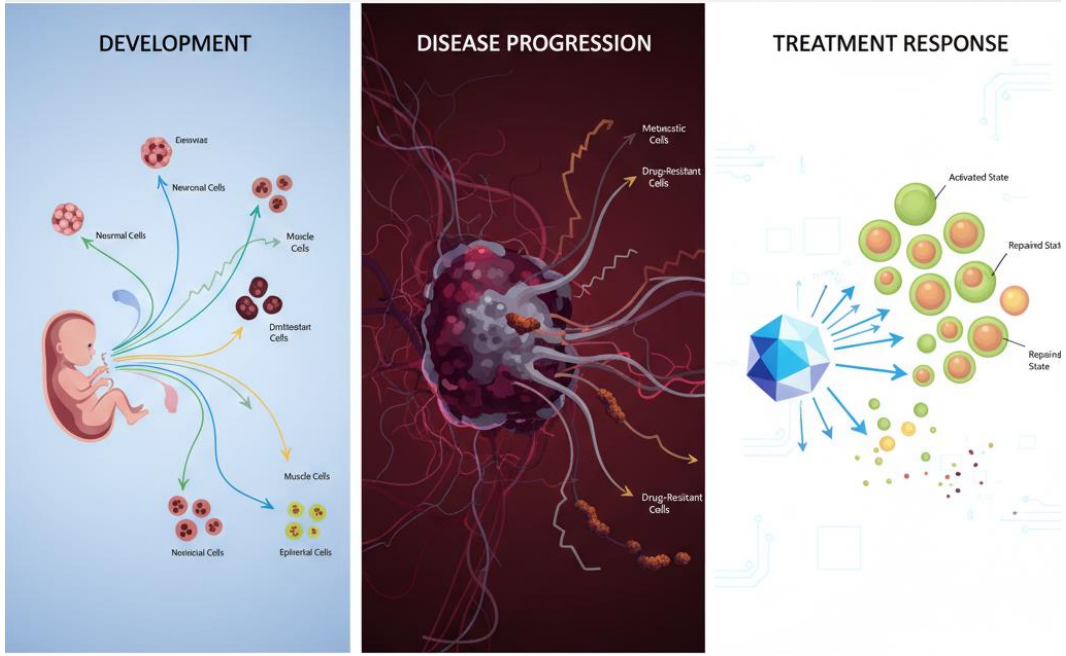
	Gaussian			Zero			ZINB		
	Ciga	UNI	Gigapath	Ciga	UNI	Gigapath	Ciga	UNI	Gigapath
IDC	$0.451_{.000}$	$0.580_{.001}$	$0.559_{.001}$	$0.450_{.006}$	$0.585_{.003}$	$0.567_{.000}$	$0.460_{.028}$	$0.587_{.003}$	$0.565_{.000}$
PRAD	$0.370_{.000}$	$0.410_{.004}$	$0.410_{.003}$	$0.370_{.005}$	$0.410_{.002}$	$0.414_{.003}$	$0.375_{.000}$	$0.421_{.002}$	$0.415_{.001}$
PAAD	$0.430_{.000}$	$0.500_{.001}$	$0.507_{.003}$	$0.424_{.009}$	$0.496_{.001}$	$0.506_{.000}$	$0.440_{.005}$	$0.507_{.004}$	$0.510_{.000}$
SKCM	$0.600_{.002}$	$0.698_{.003}$	$0.643_{.005}$	$0.600_{.001}$	$0.698_{.002}$	$0.654_{.001}$	$0.602_{.002}$	$0.704_{.005}$	$0.652_{.001}$
COAD	$0.337_{.007}$	$0.317_{.006}$	$0.320_{.006}$	$0.332_{.002}$	$0.319_{.005}$	$0.316_{.004}$	$0.334_{.006}$	$0.326_{.009}$	$0.315_{.007}$
READ	$0.043_{.003}$	$0.226_{.013}$	$0.253_{.000}$	$0.130_{.000}$	$0.230_{.006}$	$0.265_{.001}$	$0.137_{.004}$	$0.240_{.014}$	$0.257_{.006}$
CCRCC	$0.259_{.004}$	$0.326_{.003}$	$0.320_{.002}$	$0.233_{.001}$	$0.319_{.002}$	$0.324_{.002}$	$0.250_{.054}$	$0.332_{.003}$	$0.326_{.003}$
HCC	$0.103_{.000}$	$0.115_{.001}$	$0.115_{.003}$	$0.095_{.000}$	$0.115_{.002}$	$0.114_{.004}$	$0.094_{.001}$	$0.124_{.004}$	$0.117_{.001}$
LUNG	$0.575_{.000}$	$0.605_{.001}$	$0.594_{.003}$	$0.580_{.001}$	$0.604_{.002}$	$0.596_{.003}$	$0.583_{.000}$	$0.610_{.002}$	$0.602_{.000}$
LYMPH	$0.300_{.002}$	$0.302_{.000}$	$0.302_{.001}$	$0.301_{.005}$	$0.300_{.000}$	$0.301_{.002}$	$0.308_{.000}$	$0.305_{.001}$	$0.305_{.001}$
Average	0.347	0.407	0.402	0.351	0.407	0.405	0.361	0.415	0.406

Empirically, specialized priors (like **ZINB**) significantly outperform the standard **Gaussian**.

Cell Dynamics & Fate

- **scRNA-seq**: destructive capturing only static snapshots of cells at different timepoints.
- **Single-Cell Trajectory Inference**: reconstructs continuous cellular processes from these discrete, unpaired static snapshots.





Cell Dynamics & Fate

Alexander Tong

AITHYRA
Verified email at aithyra.at - Homepage

Flow Models Deep Learning Optimal Transport Single-cell Protein design

Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics 231 2020
A Tong, J Huang, G Wolf, D Van Dijk, S Krishnaswamy
International conference on machine learning, 9526-9536

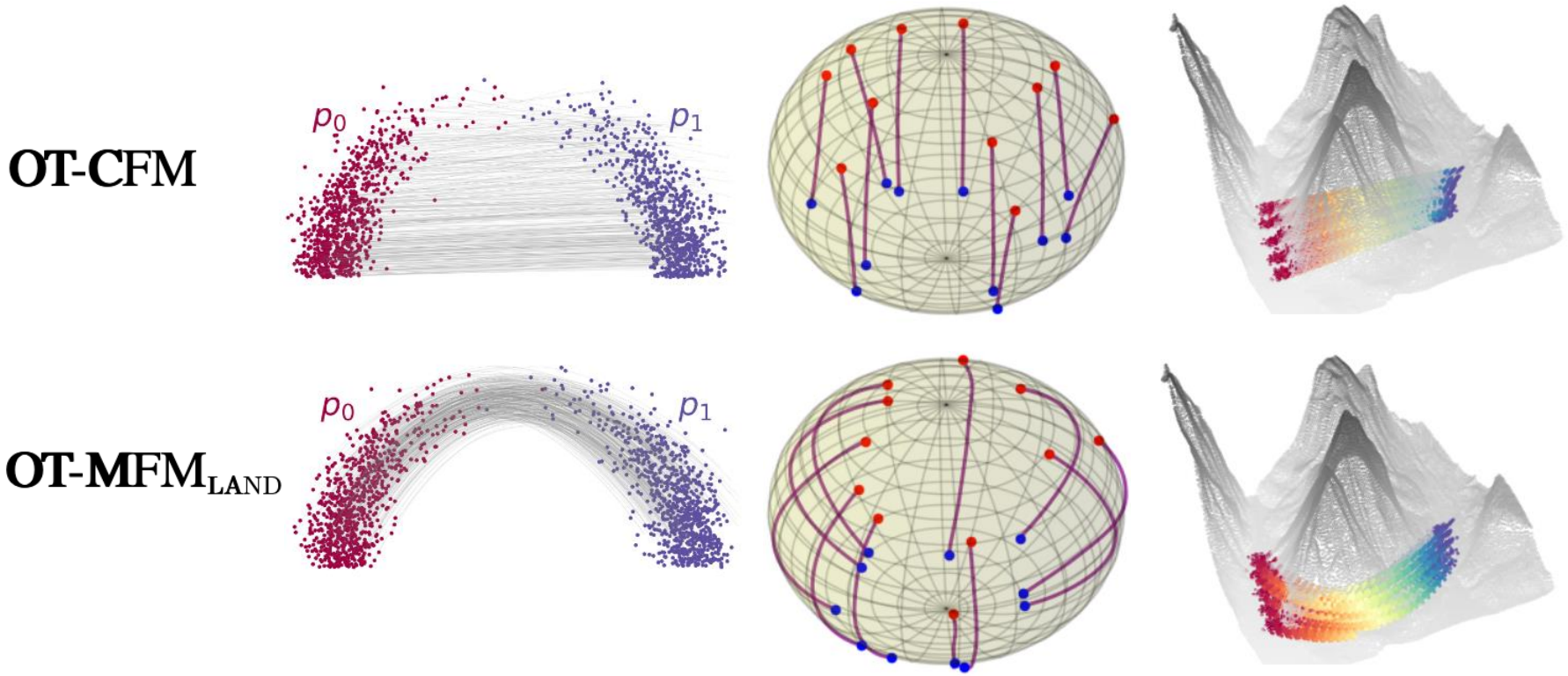
TrajectoryNet: CNF in Cell Dynamics

Improving and generalizing flow-based generative models with minibatch optimal transport 575 * 2024
A Tong, K Fatras, N Malkin, G Huguet, Y Zhang, J Rector-Brooks, G Wolf, ...
Transactions on Machine Learning Research

OT-CFM: Flow Matching in Cell Dynamics

Cell Dynamics & Fate

- Metric Flow Matching(K. Kapusniak et al, NeurIPS 2024)



Method	Cite (100D)
SF^2 M-Geo	44.498 ± 0.416
SF^2 M-Exact	46.530 ± 0.426
OT-CFM	45.393 ± 0.416
I-CFM	48.276 ± 3.281
WLF-OT	44.821 ± 0.126
WLF-UOT	43.731 ± 1.375
WLF-SB	46.131 ± 0.083
I-MFM$_{\text{RBF}}$	45.987 ± 4.014
OT-MFM$_{\text{RBF}}$	41.784 ± 1.020

Cell Dynamics & Fate

- Unbalanced Flow Matching: Simulation-based to Simulation-free

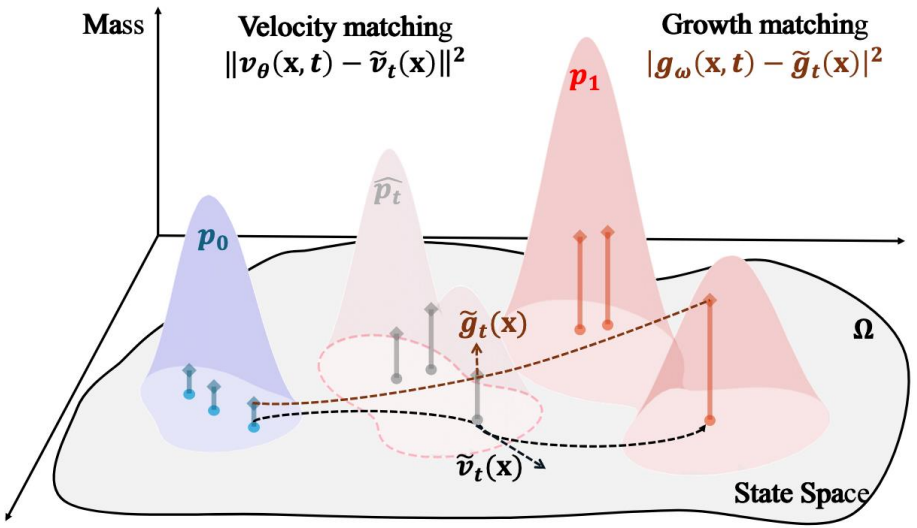
$$\partial_t p_t = -\nabla \cdot (p_t v_t)$$

$$\begin{cases} x(0) = x_0 \\ \dfrac{\mathrm{d}x(t)}{\mathrm{d}t} = u(x(t), t) \quad \forall t \in [0, 1] \end{cases}$$

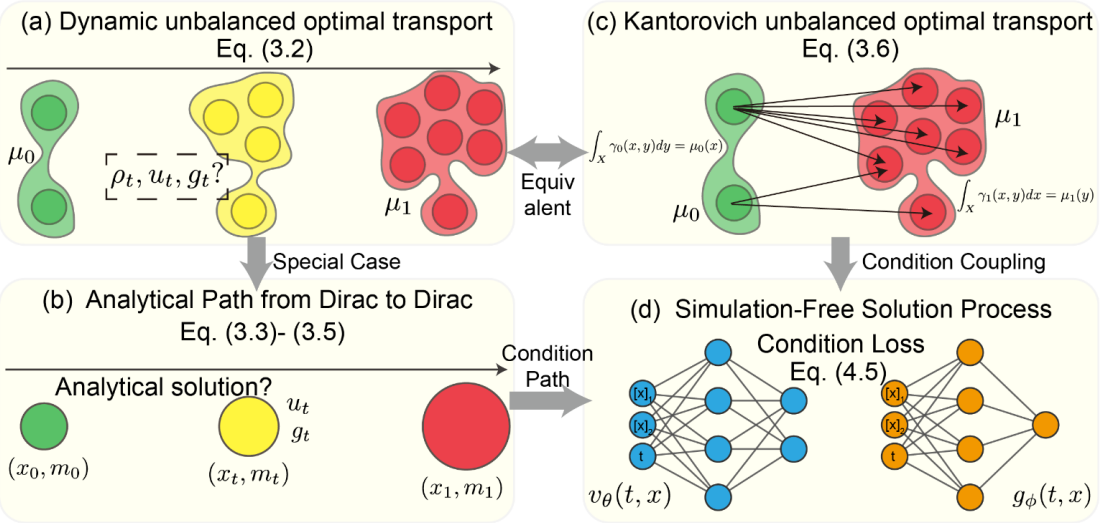
$$\longrightarrow$$

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

$$\frac{d}{dt}\begin{pmatrix} \boldsymbol{\phi}_t(\boldsymbol{x}) \\ \ln m_t(\boldsymbol{x}) \end{pmatrix} = \begin{pmatrix} \boldsymbol{u}_t(\boldsymbol{\phi}_t(\boldsymbol{x})) \\ g_t(\boldsymbol{\phi}_t(\boldsymbol{x})) \end{pmatrix}, \quad \boldsymbol{\phi}_0(\boldsymbol{x}) = \boldsymbol{x}, \quad m_0(\boldsymbol{x}) \textbf{ given.}$$

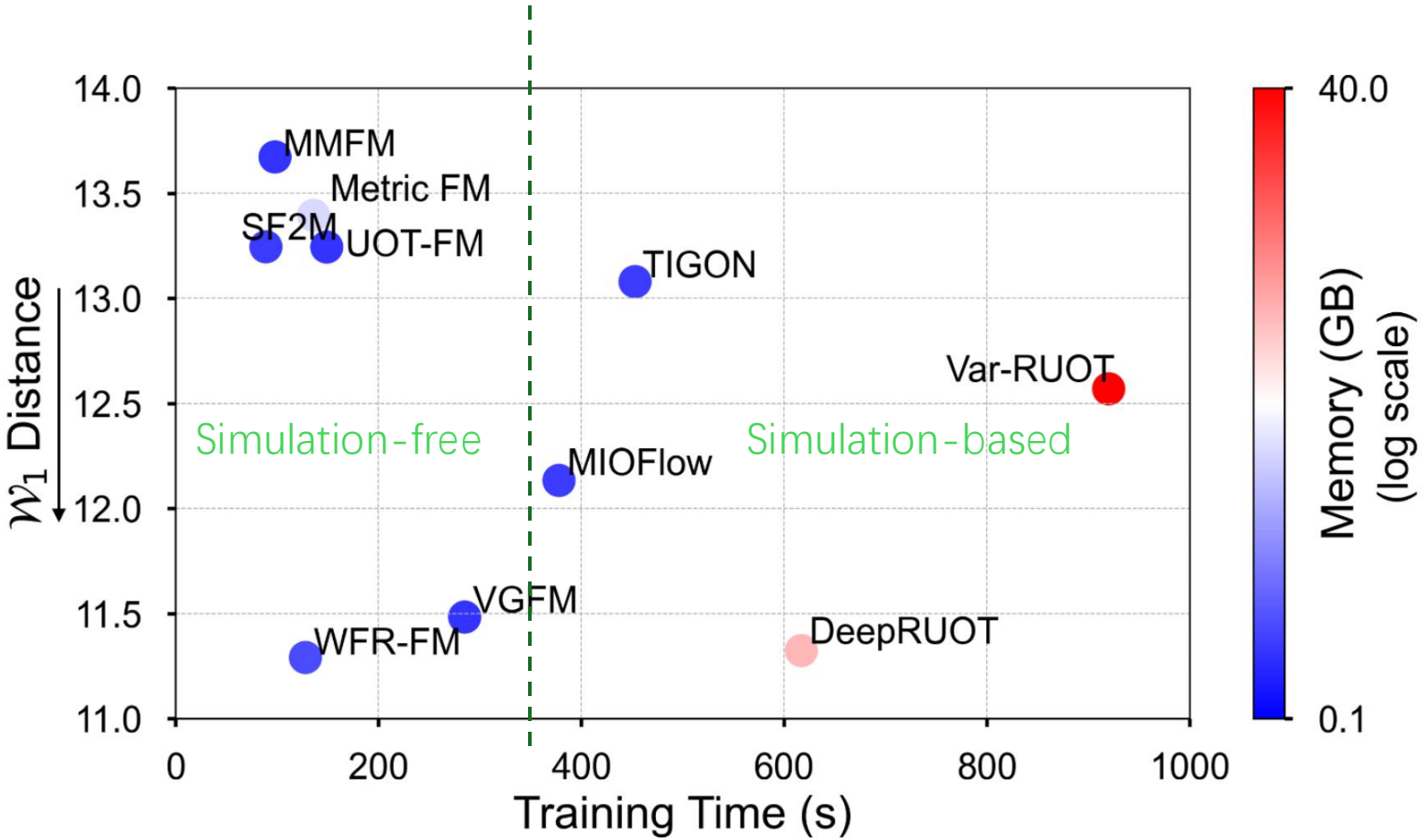


WANG D. et al NeurIPS(2025)



PENG D. et al (2025), ICLR Under Review

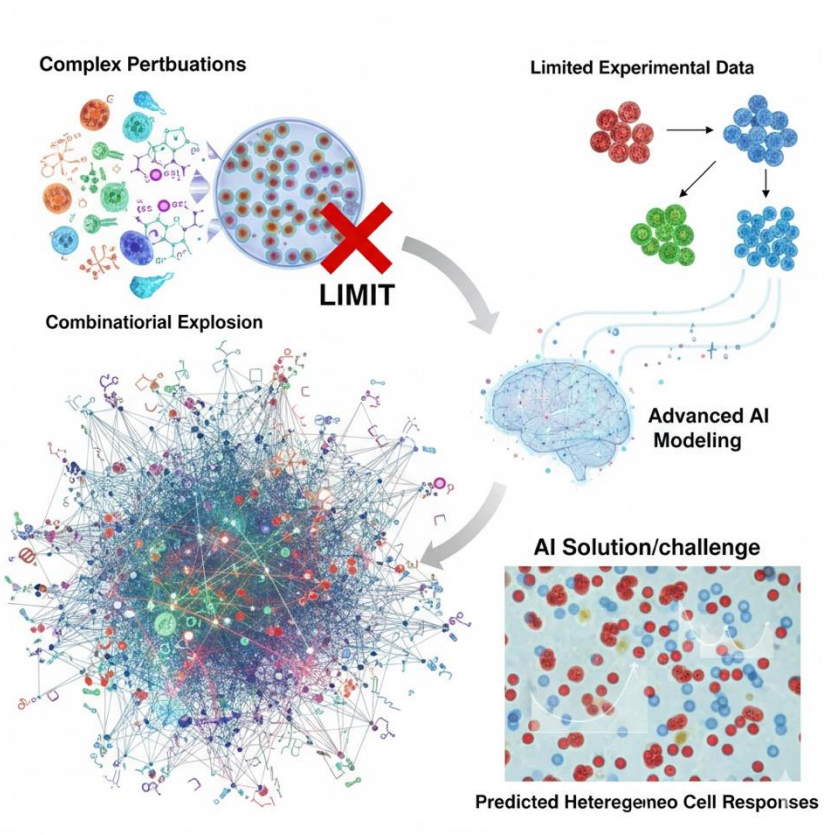
Cell Dynamics & Fate

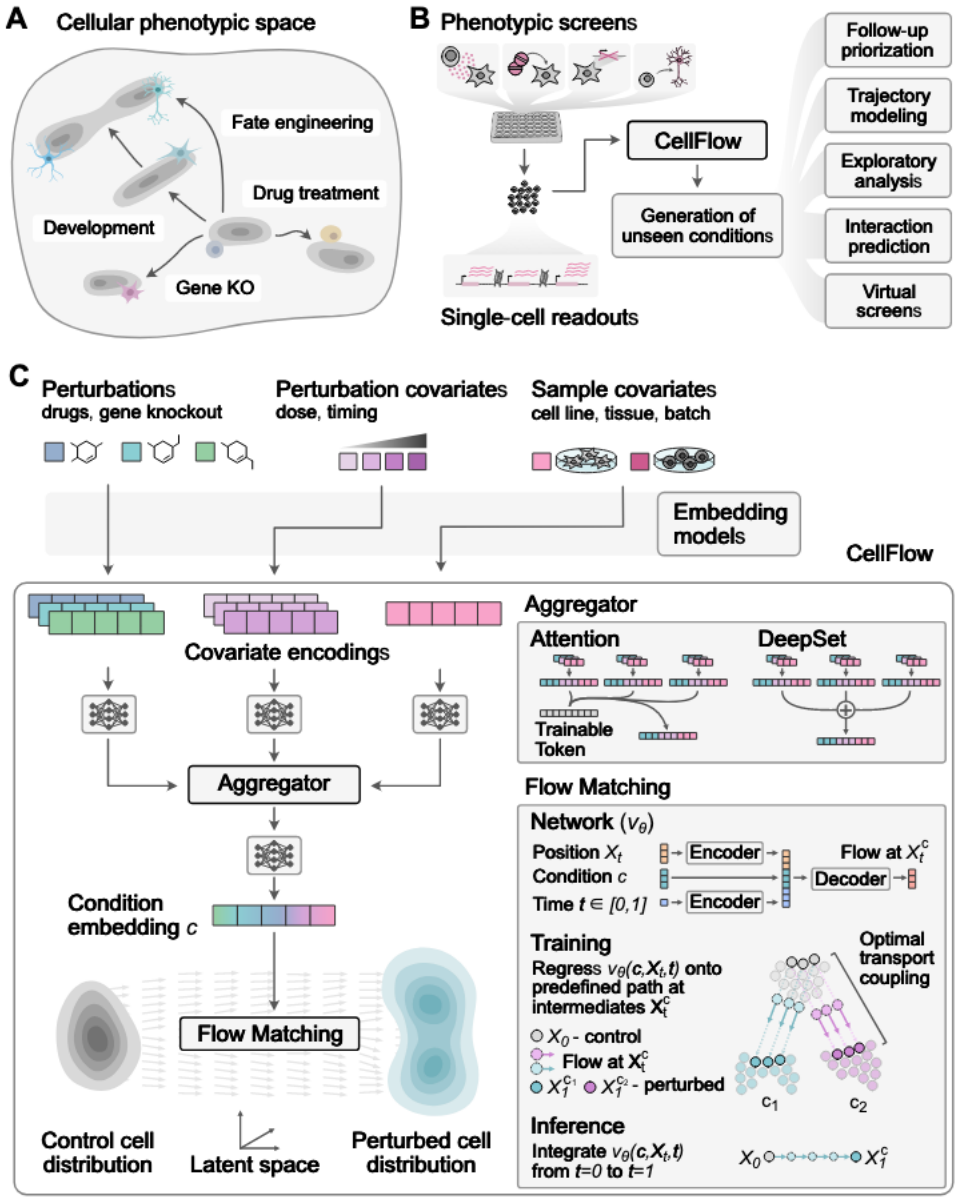


PENG D. et al (2025), ICLR Under Review

Perturbation Response

- CellFlow(Theis Lab, Biorxiv, 2025)





Why Flow Matching in Virtual Cell?

- Better Performance in Data-Constrained Settings
- Flexible Priors Matching Biology
- From Distribution to Distribution
- Interpretable Dynamic Trajectory Modeling
- Flexible Path Design
- Efficient and practical: simulation-free, exact-likelihood options

Thank You!