Flow Matching

Simulation-Free Continuous Normalizing Flows for Generative Modeling

Heli Ben-Hamu

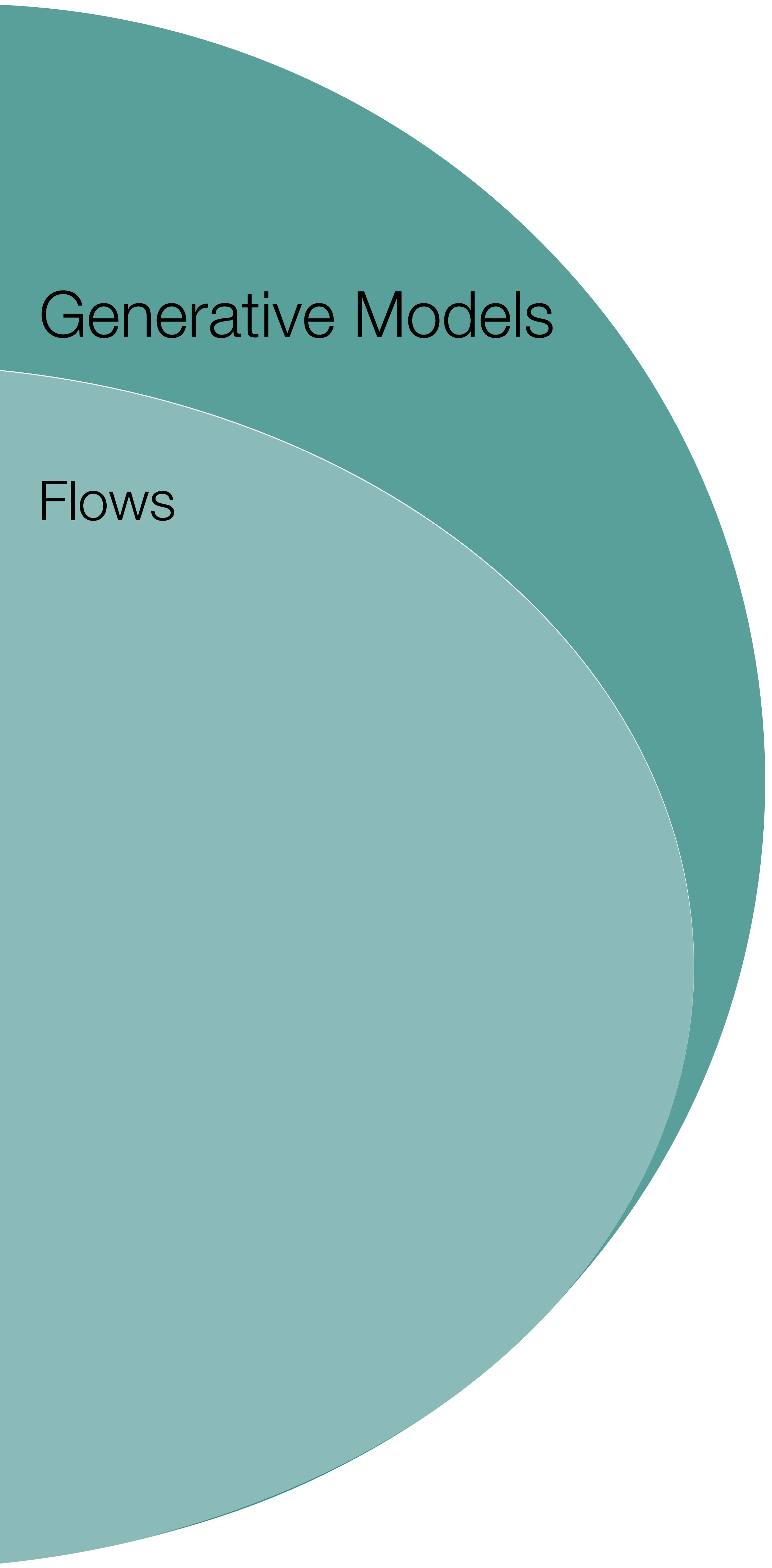
מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

June 2024

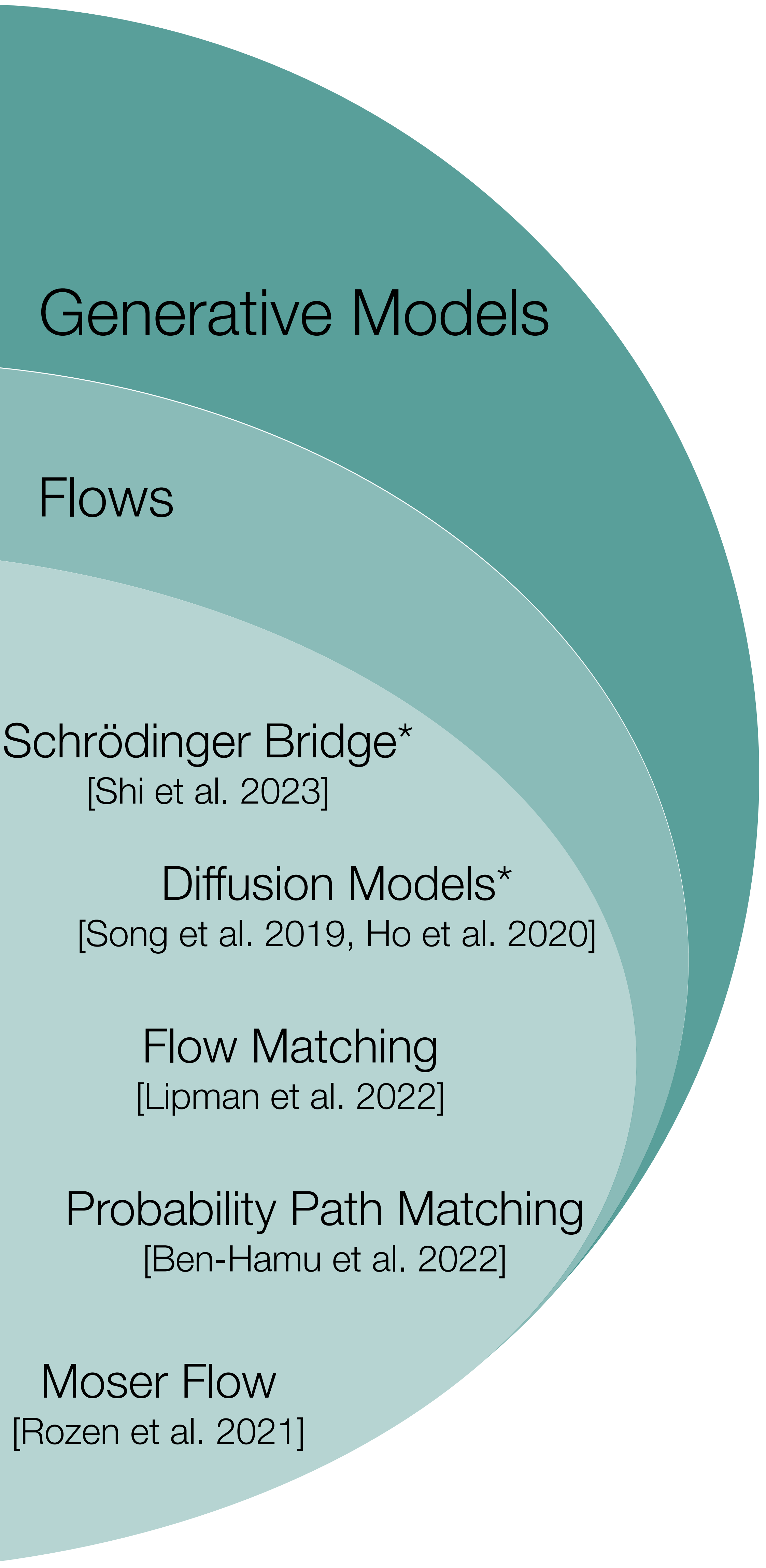
Simulation-Free Continuous Normalizing Flows for Generative Modeling

Simulation-Free Continuous Normalizing Flows for Generative Modeling

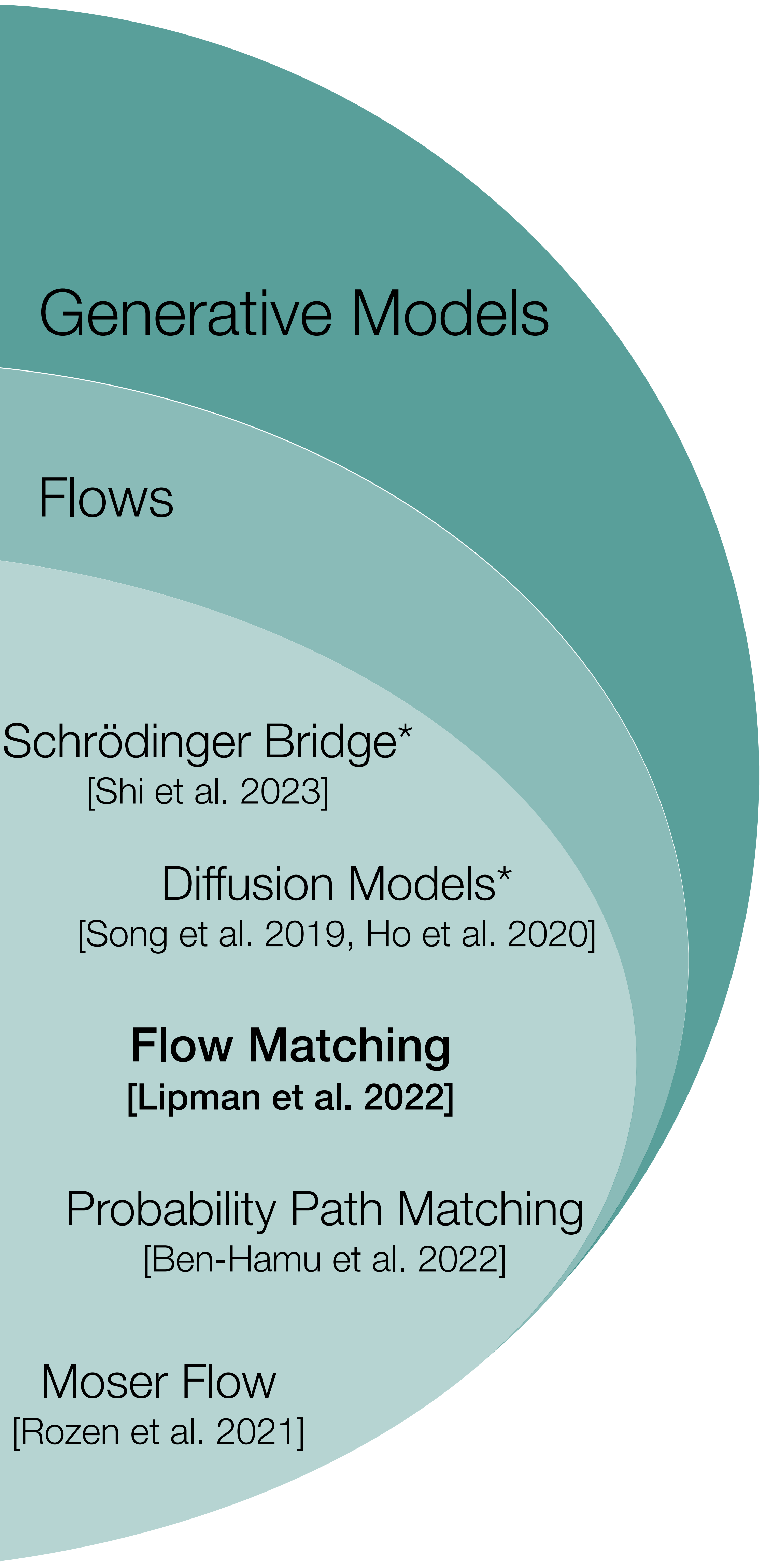
Simulation-Free Continuous Normalizing Flows for Generative Modeling



Simulation-Free Continuous Normalizing Flows for Generative Modeling

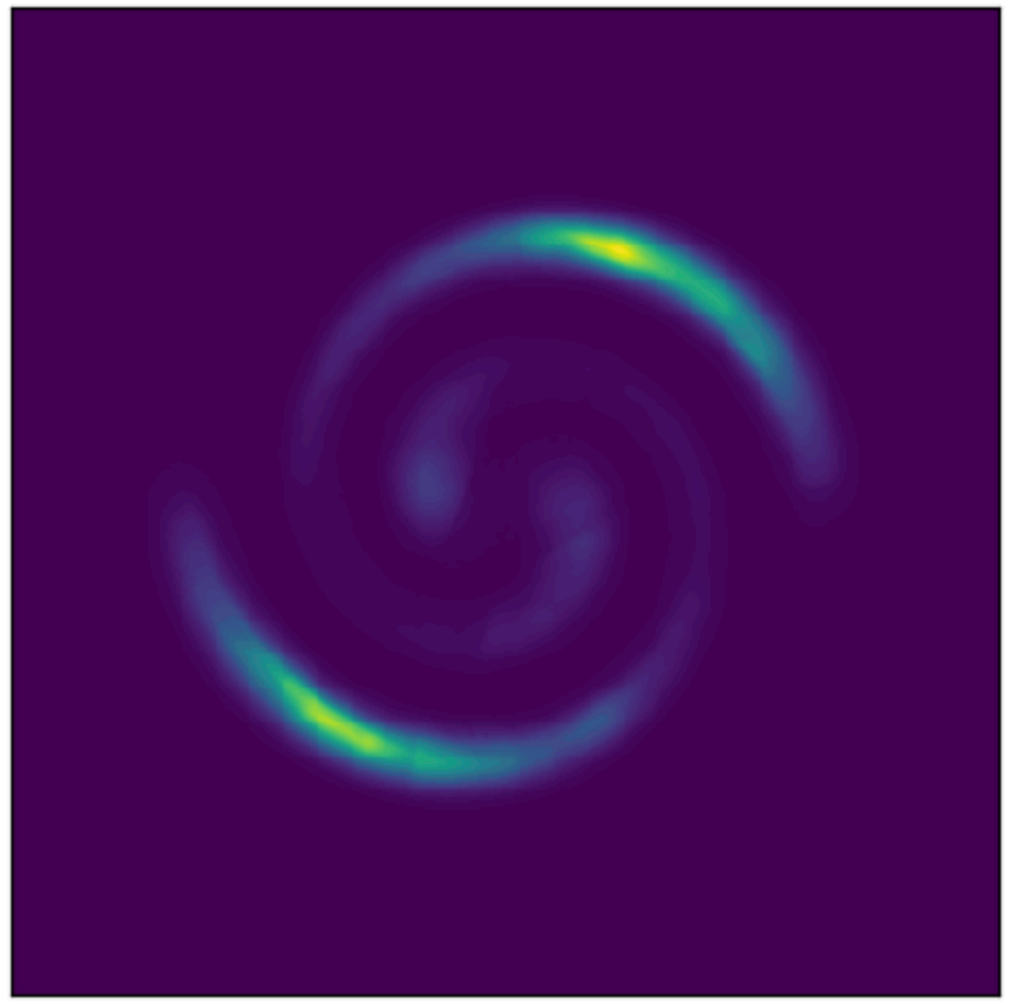


Simulation-Free Continuous Normalizing Flows for Generative Modeling



Generative Modeling

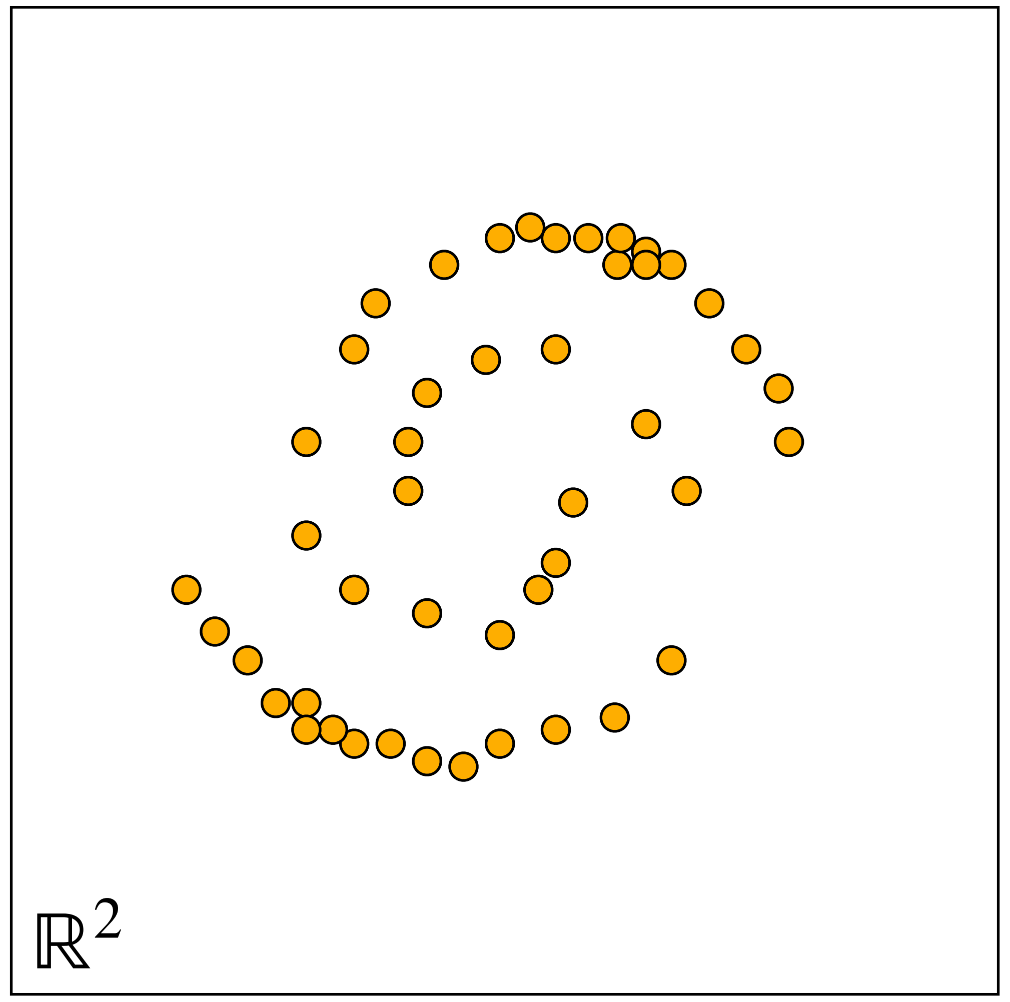
- Unknown: data distribution q



Generative Modeling

- Unknown: data distribution q
- Given: samples $x_1 \sim q$

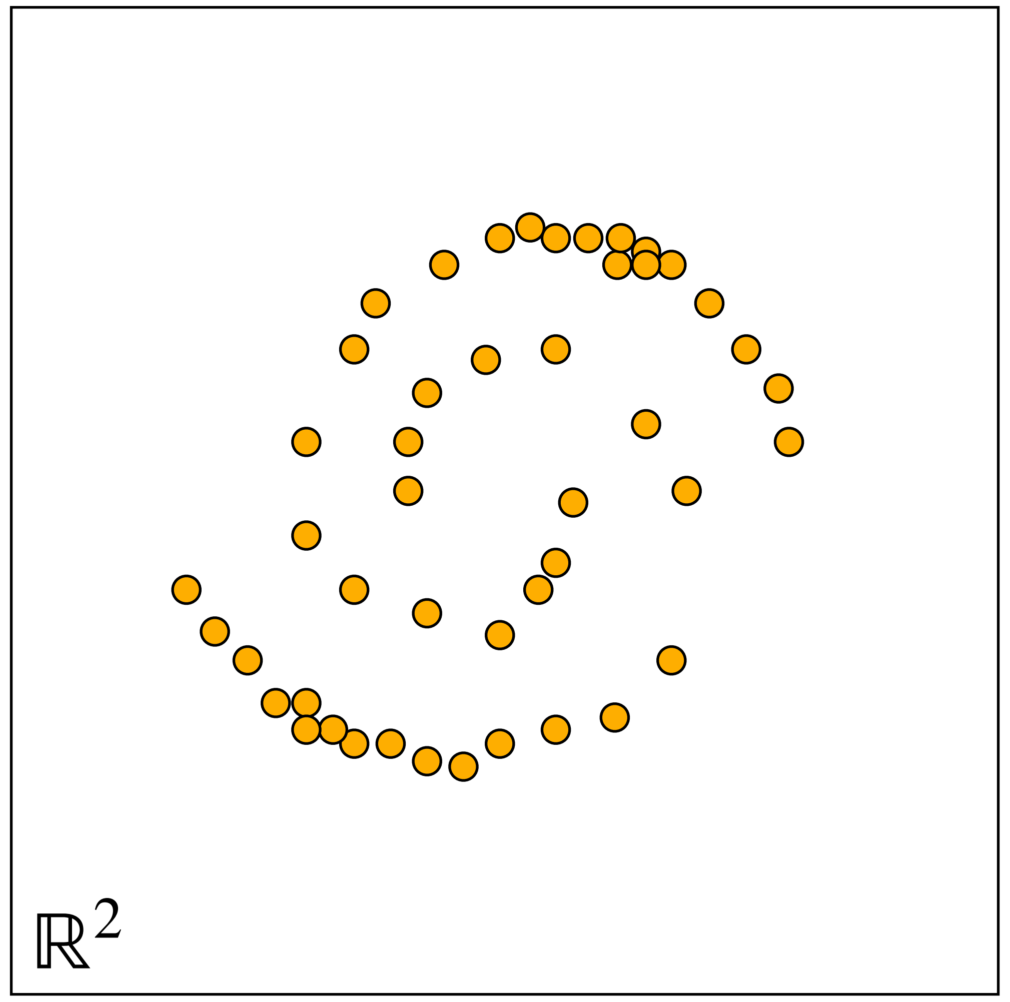
$x_1 \sim q$



Deep Generative Modeling

- Unknown: data distribution q
- Given: samples $x_1 \sim q$

$x_1 \sim q$



Sampling
$x_0 \sim p$
$\psi_\theta(x_0) \sim q$

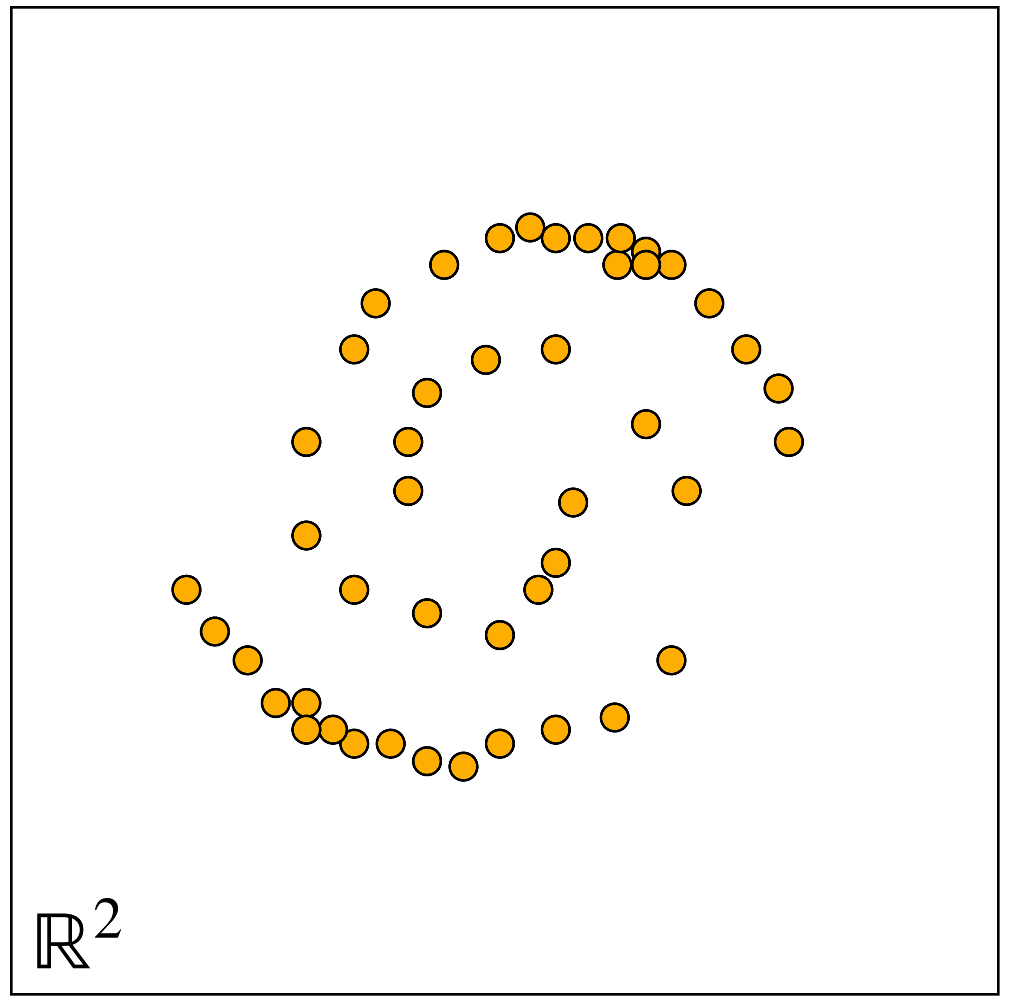
Density Estimation
$p_\theta \approx q$

Deep Generative Modeling

- Unknown: data distribution q

- Given: samples $x_1 \sim q$

$x_1 \sim q$



Generative Model

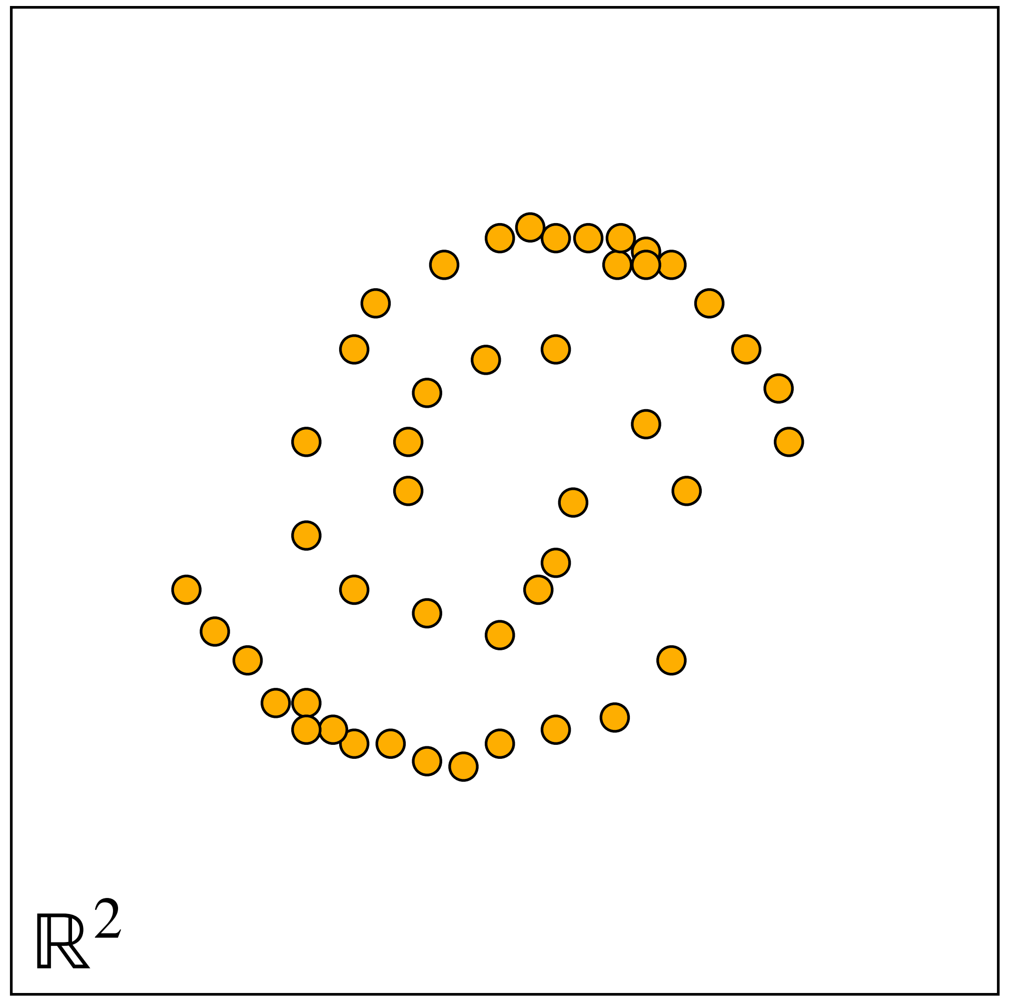
(ψ_θ , p_θ)

Generator   Underlying Density

Deep Generative Modeling

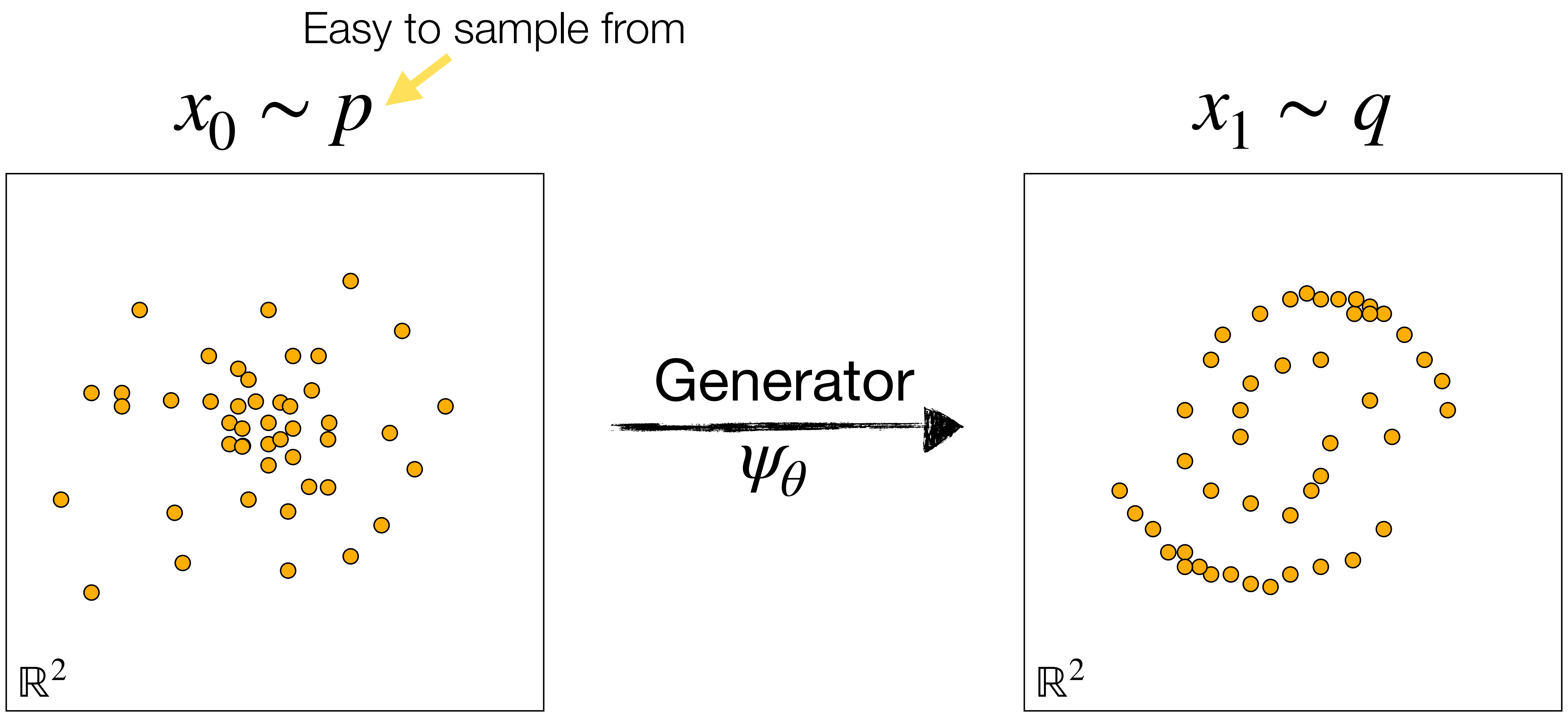
- Unknown: data distribution q
- Given: samples $x_1 \sim q$

$x_1 \sim q$



Goal: find parameters θ s.t. $p_\theta \approx q$

Deep Generative Modeling



Sampling

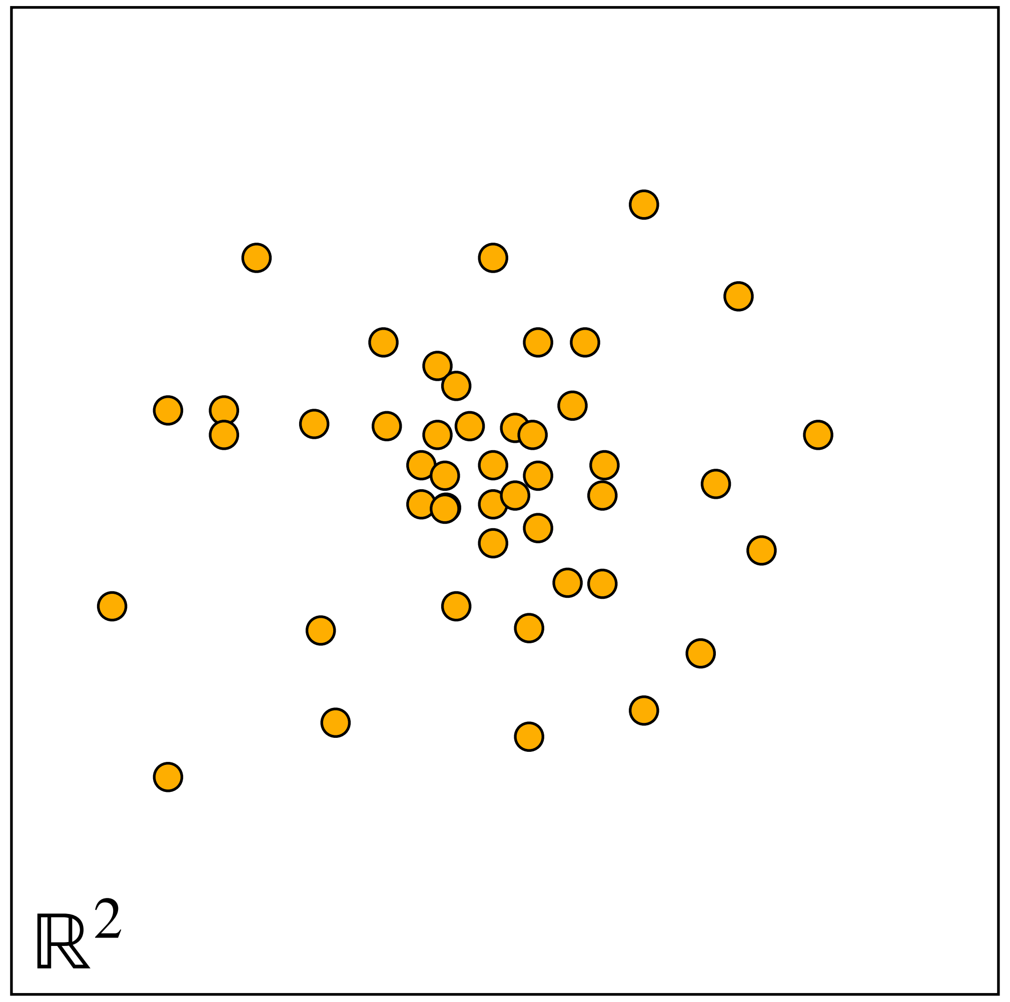
$x_0 \sim p$

$\psi_\theta(x_0) \sim q$

Deep Generative Modeling

Easy to sample from

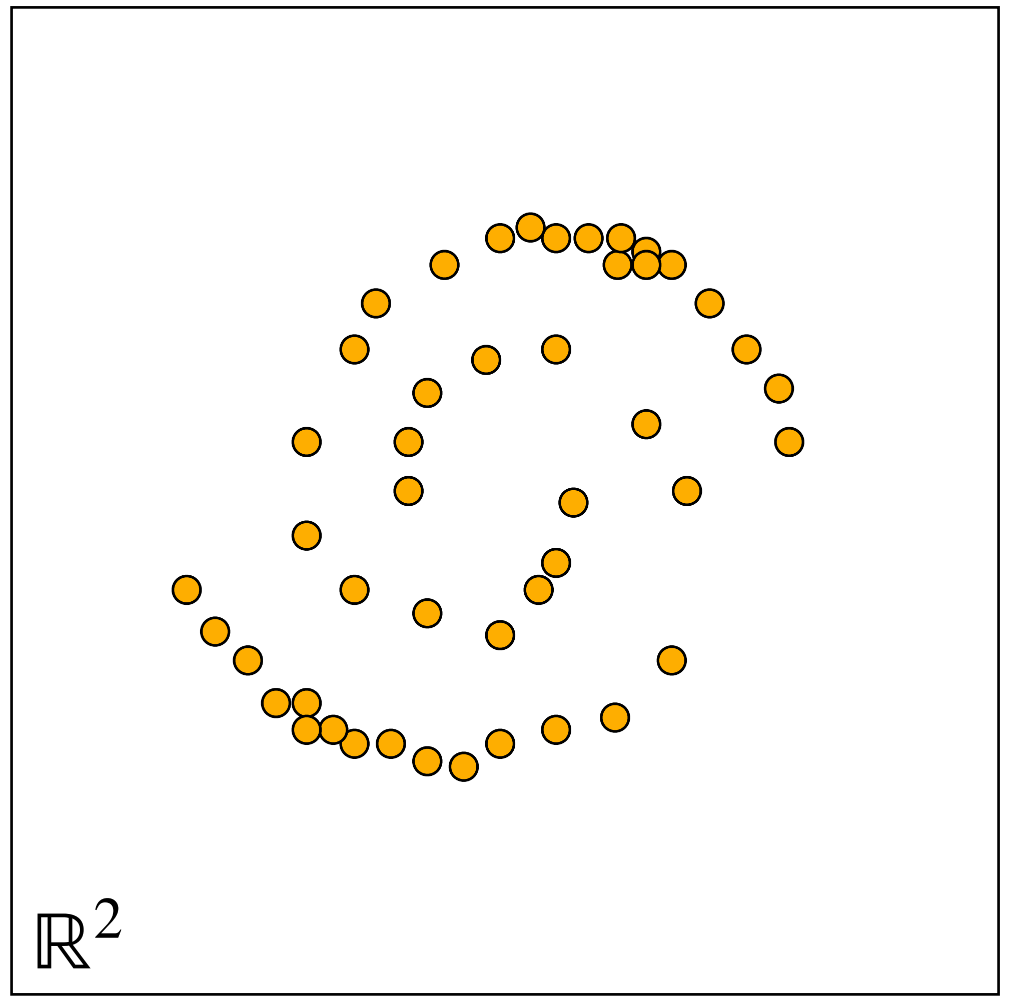
$x_0 \sim p$



Generator

ψ_θ

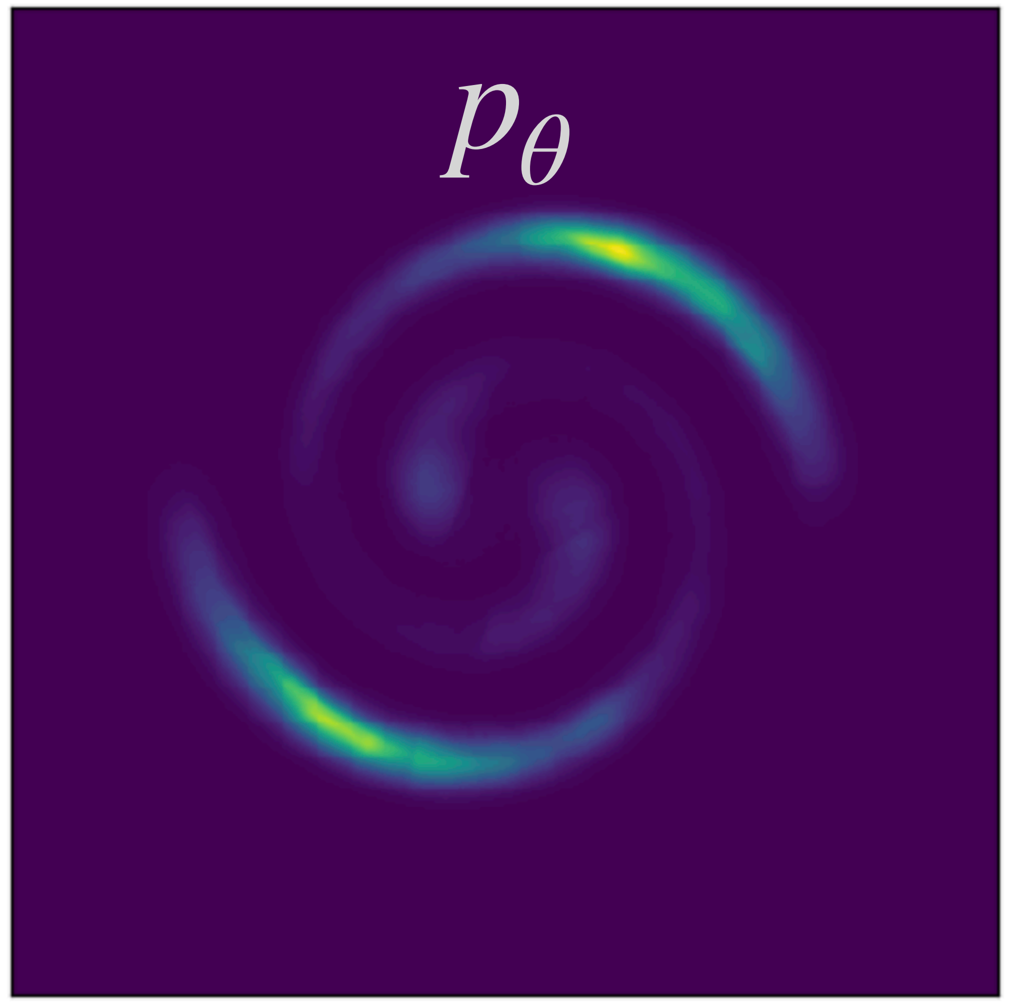
$x_1 \sim q$



Sampling

$x_0 \sim p$

$\psi_\theta(x_0) \sim q$



Density Estimation

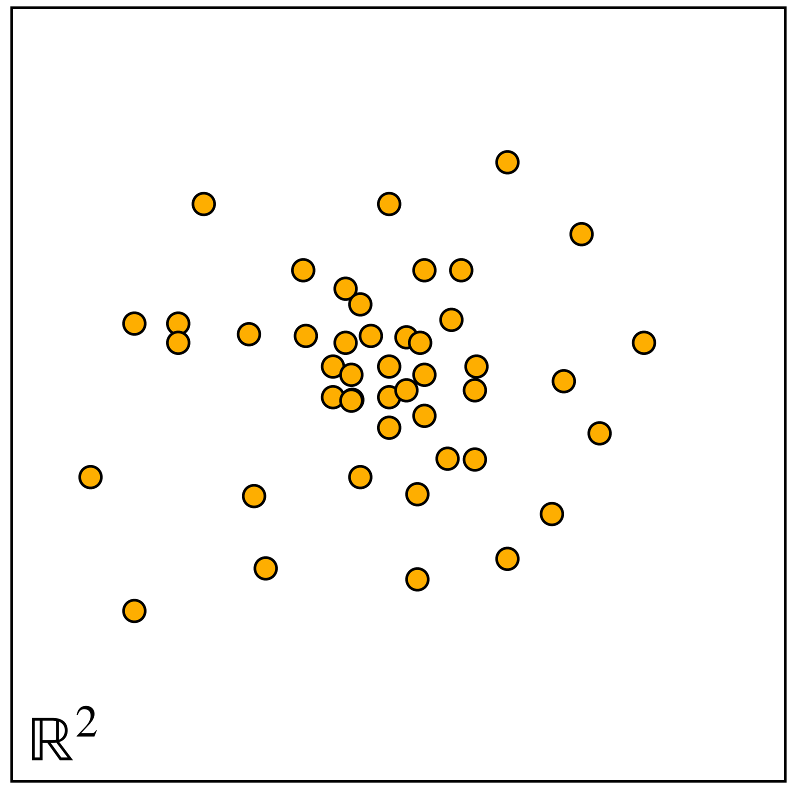
$p_\theta \approx q$

Simulation-Free Continuous Normalizing Flows for Generative Modeling

Generative Models

Easy to sample

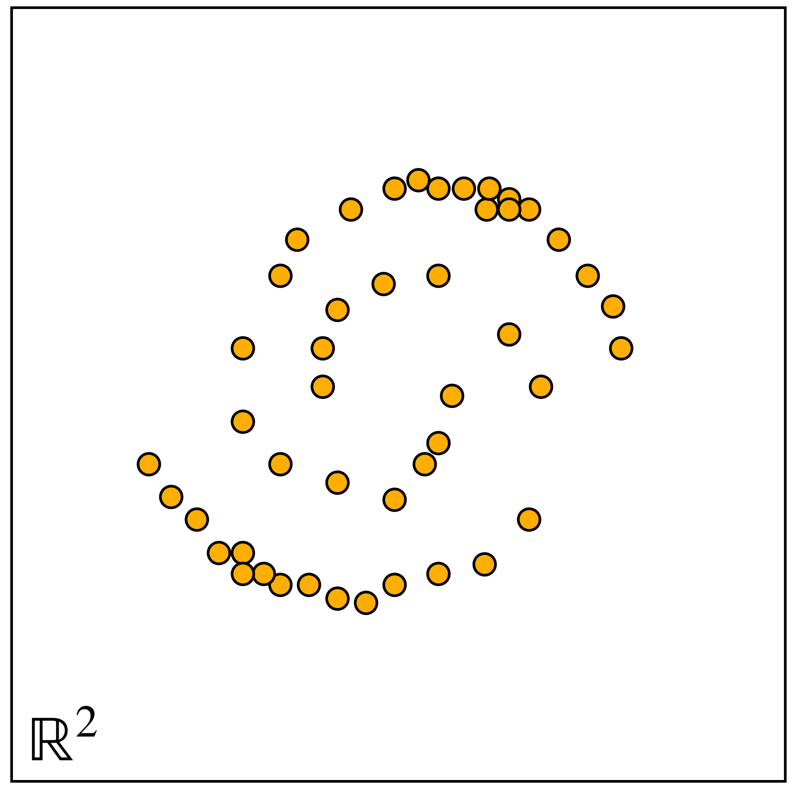
$x_0 \sim p$



Generator

ψ_θ

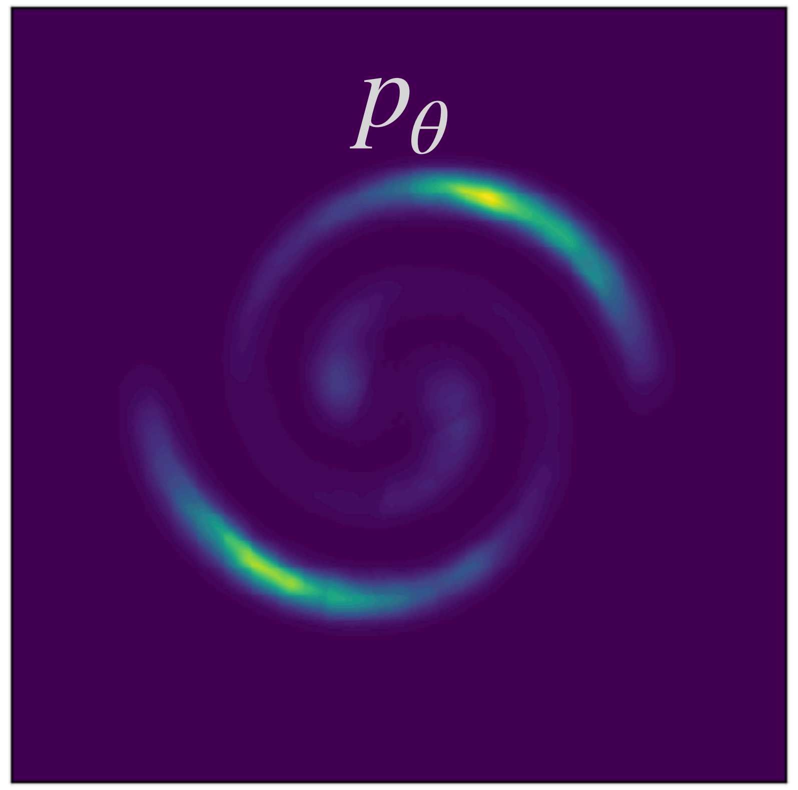
$x_1 \sim q$



Sampling

$x_0 \sim p$

$\psi_\theta(x_0) \sim q$

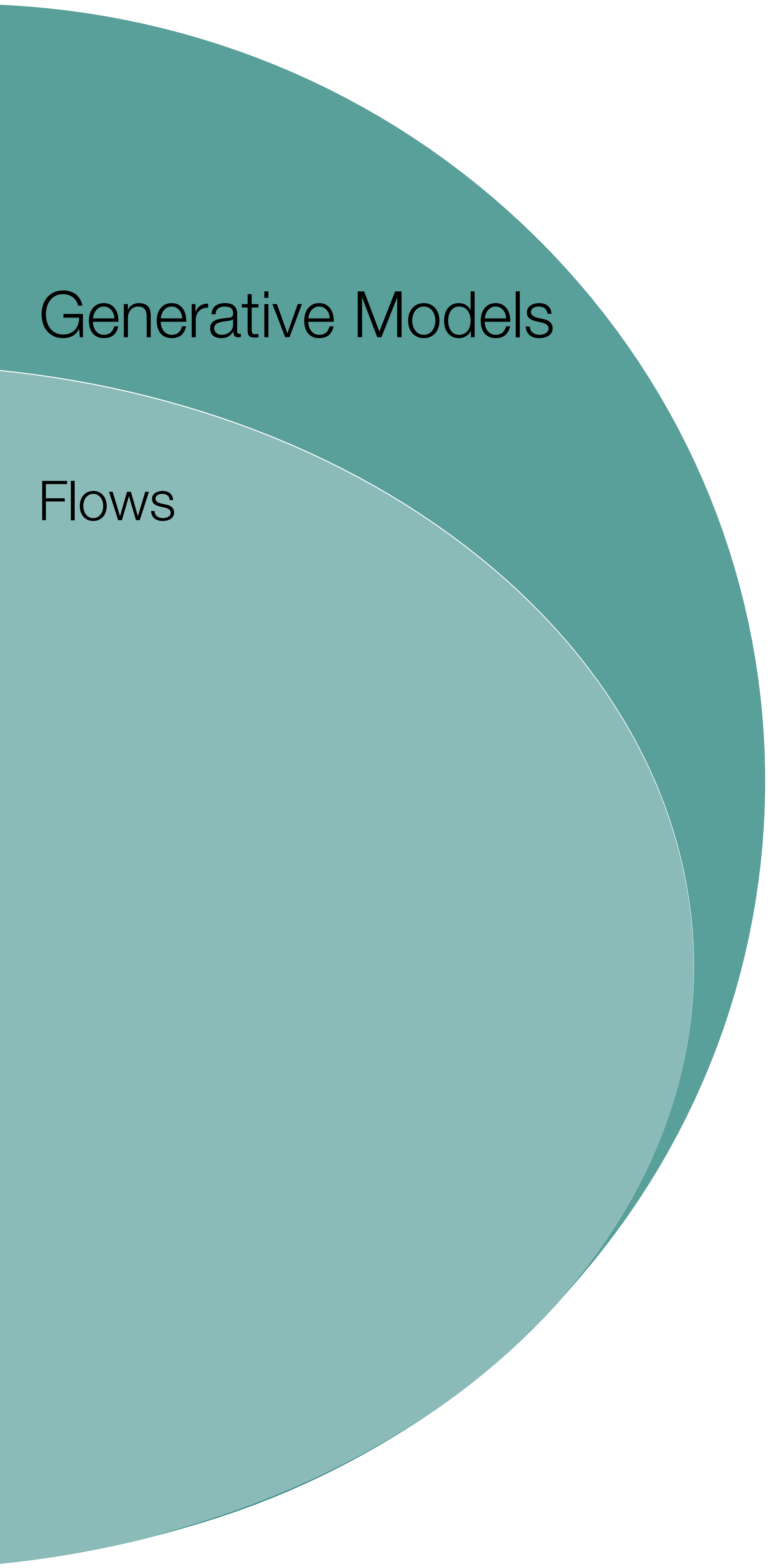


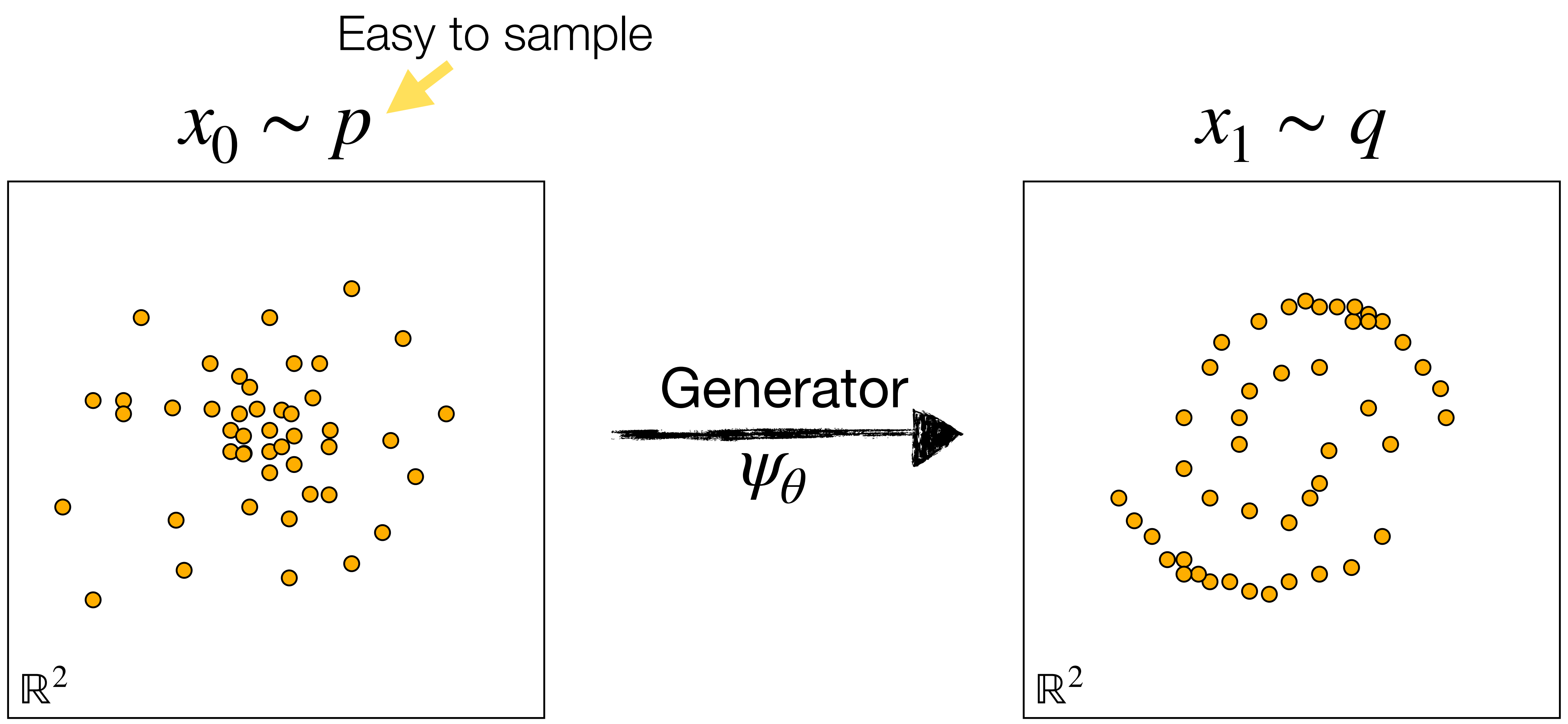
Density Estimation

$p_\theta \approx q$

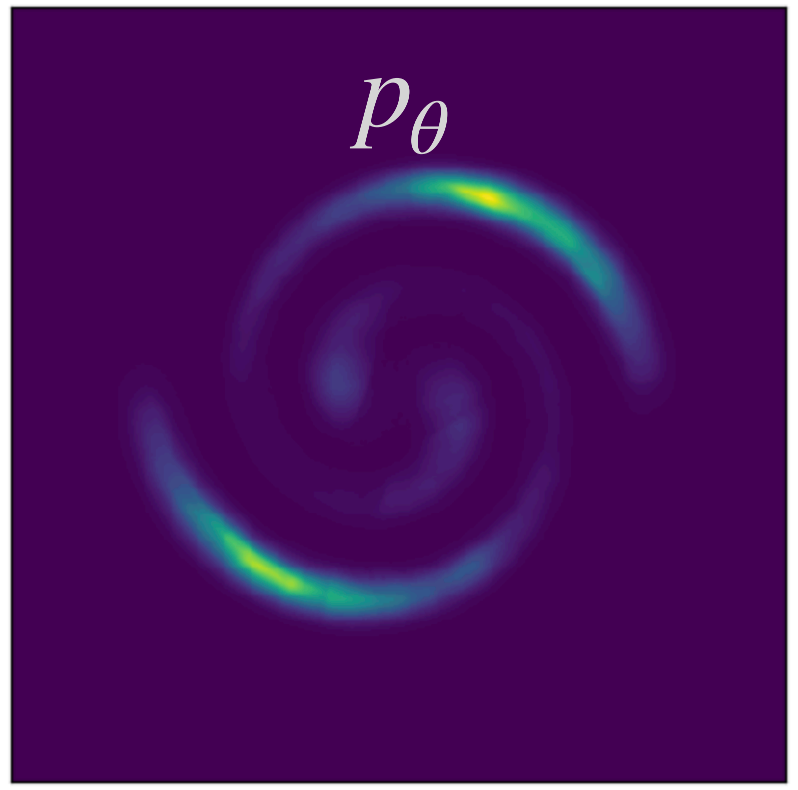
How to model ψ_θ?

Simulation-Free Continuous Normalizing Flows for Generative Modeling



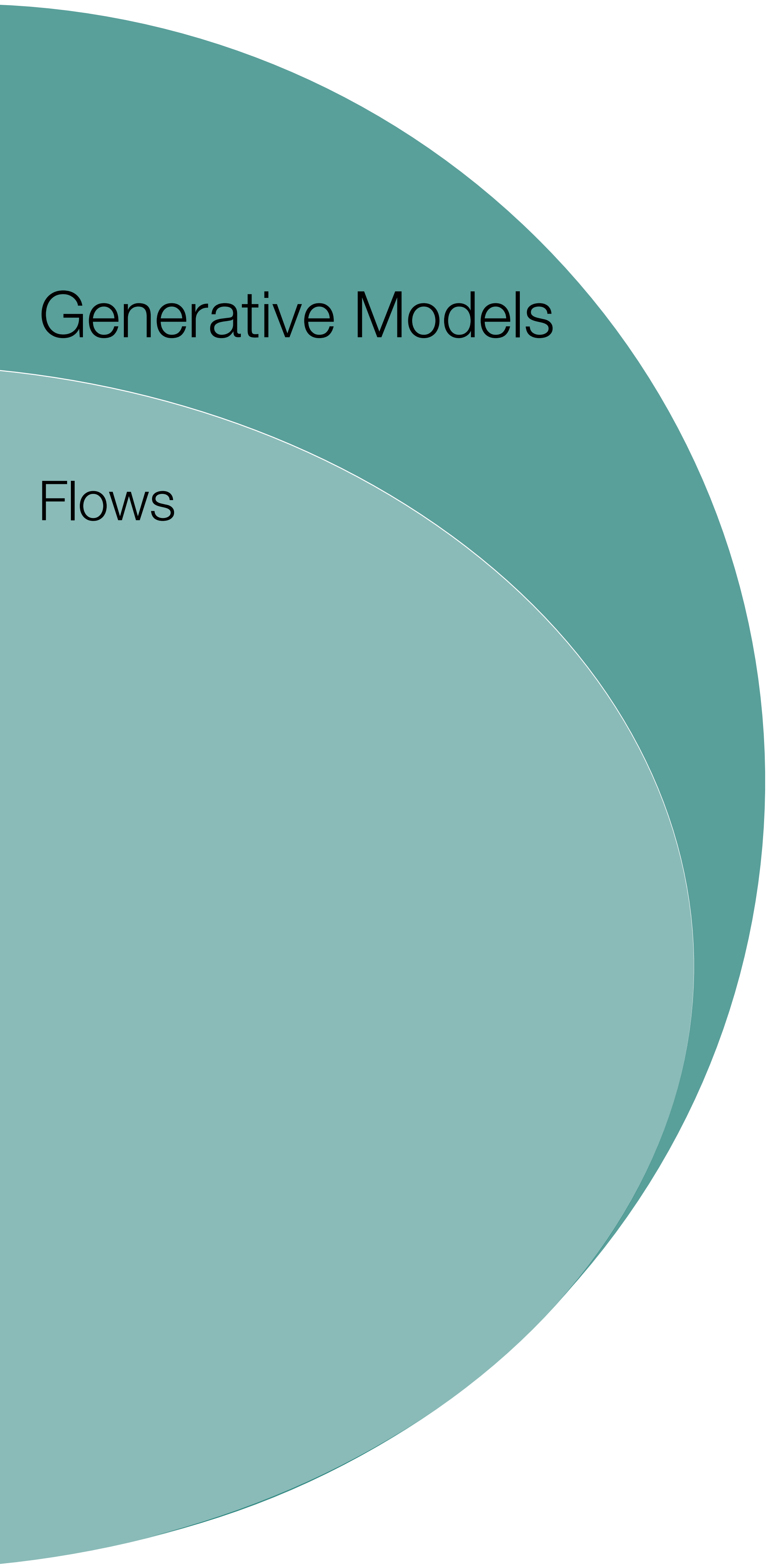


Sampling
$x_0 \sim p$
$\psi_\theta(x_0) \sim q$

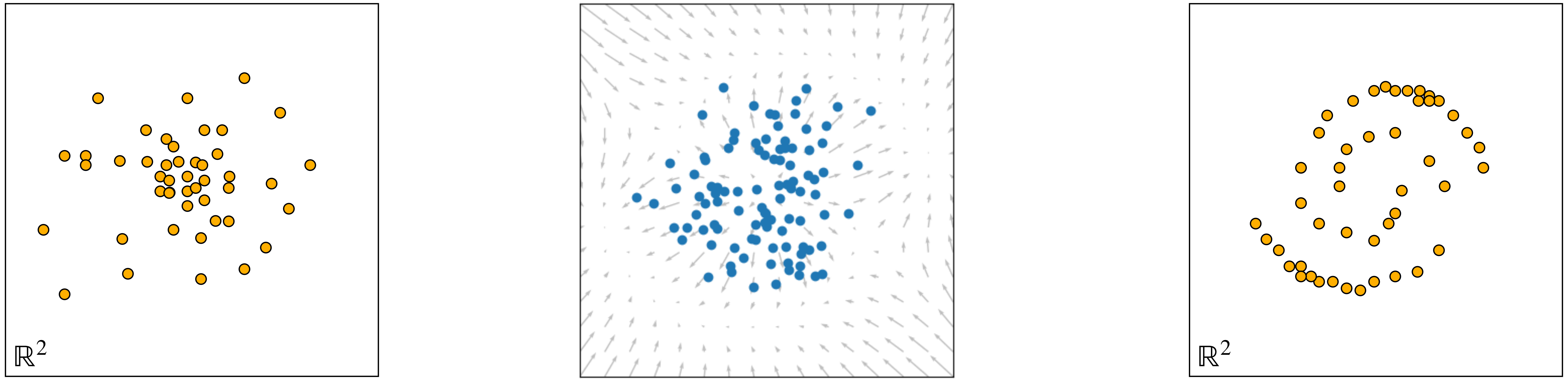


Density Estimation
$p_\theta \approx q$

Simulation-Free Continuous Normalizing Flows for Generative Modeling



$x_0 \sim p$ $\qquad$ $x_1 \sim q$



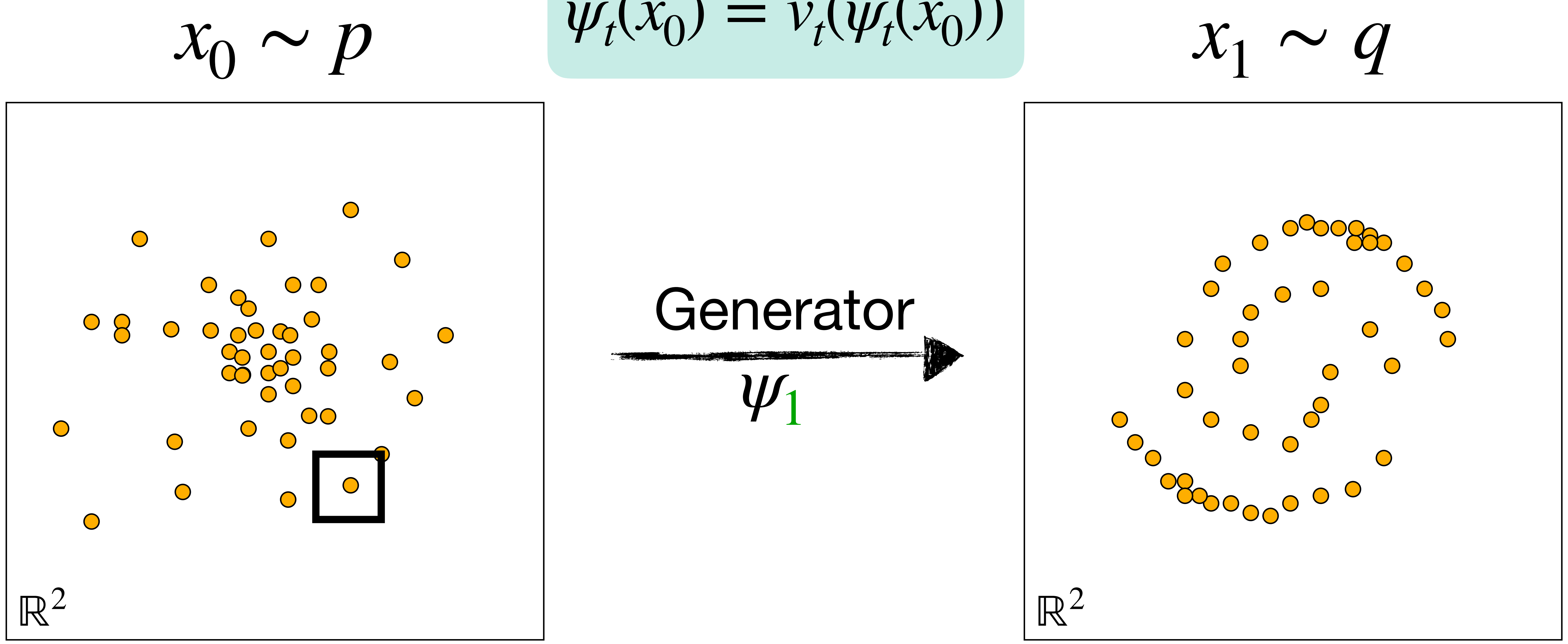
$$\psi_t : [0,1] \times \mathbb{R}^2 \to \mathbb{R}^2$$

Continuous Normalizing Flows

Dynamical perspective

Flow ODE

$$\dot{\psi}_t(x_0) = v_t(\psi_t(x_0))$$



$$\psi_t : [0,1] \times \mathbb{R}^2 \to \mathbb{R}^2$$

Velocity Field $\quad v_t : [0,1] \times \mathbb{R}^2 \to \mathbb{R}^2$

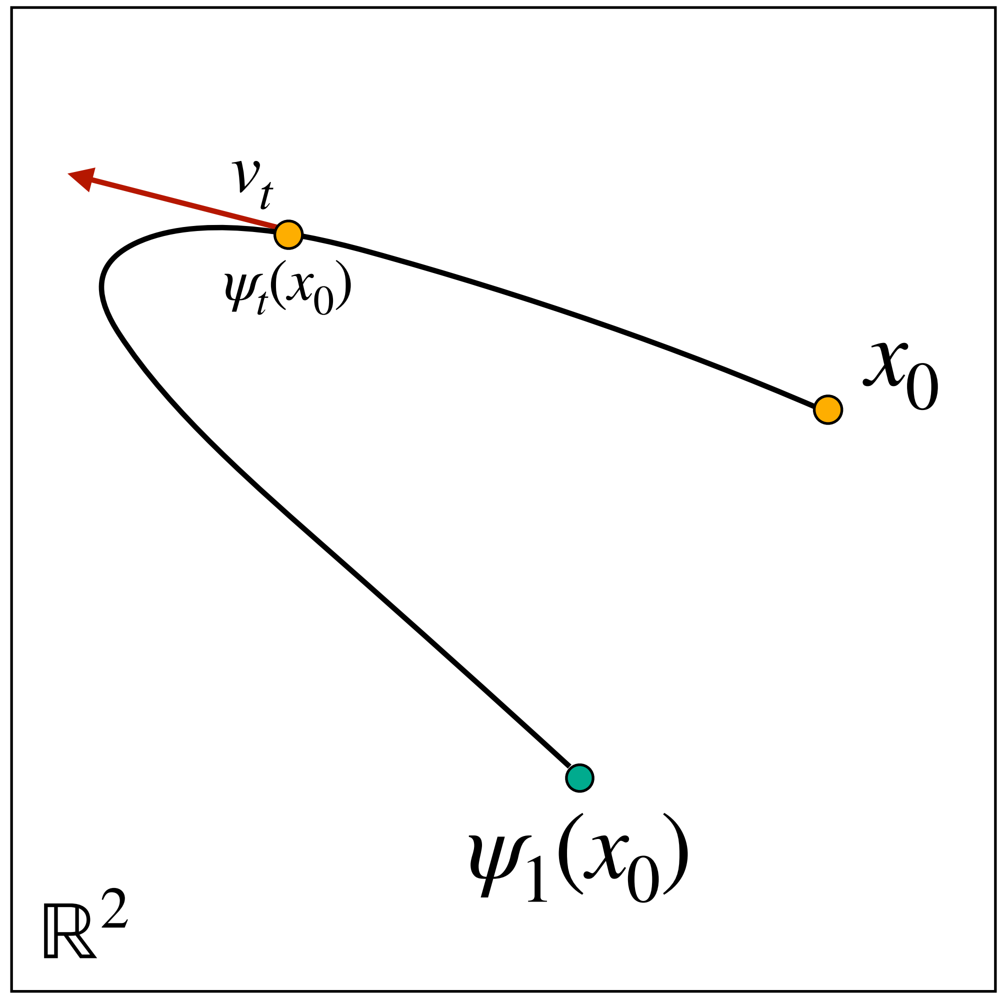
Chen et al. *Neural Ordinary Differential Equations* (NeurIPS 2018)

Continuous Normalizing Flows

Dynamical perspective

Flow ODE

$$\dot{\psi}_t(x_0) = v_t(\psi_t(x_0))$$



$$\psi_t(x_0) = x_0 + \int_0^t v_s(\psi_s(x_0))ds$$

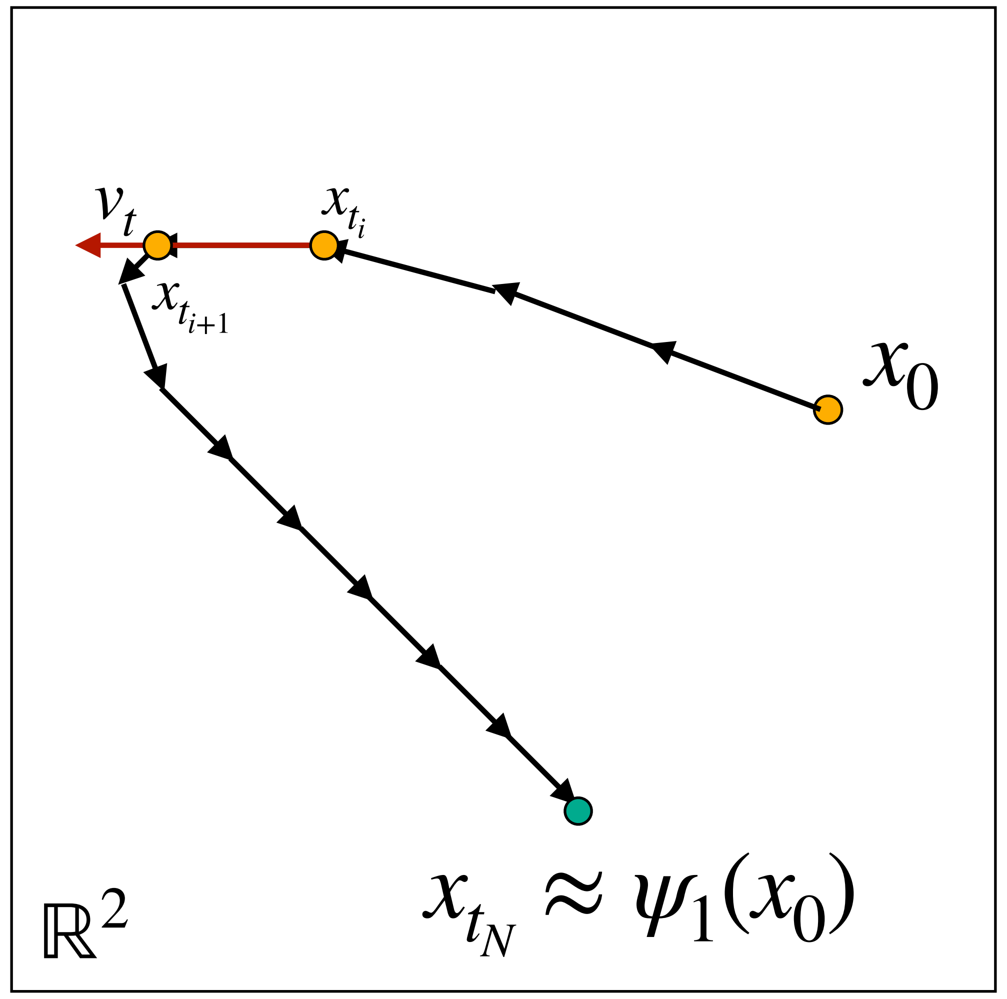
Chen et al. *Neural Ordinary Differential Equations* (NeurIPS 2018)

Continuous Normalizing Flows

Dynamical perspective

Flow ODE

$$\dot{\psi}_t(x_0) = v_t(\psi_t(x_0))$$



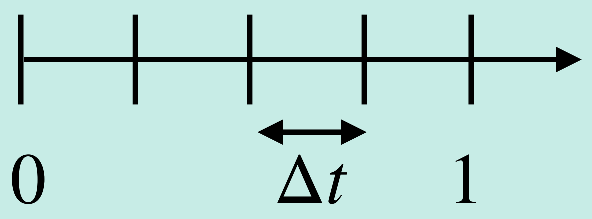
$$\psi_t(x_0) = x_0 + \int_0^t v_s(\psi_s(x_0))ds$$

Sampling

Numerical ODE solver

e.g., Euler solver:

$$x_{t_{i+1}} = x_{t_i} + \Delta t \cdot v_t(x_{t_i})$$



$$N = \frac{1}{\Delta t}$$

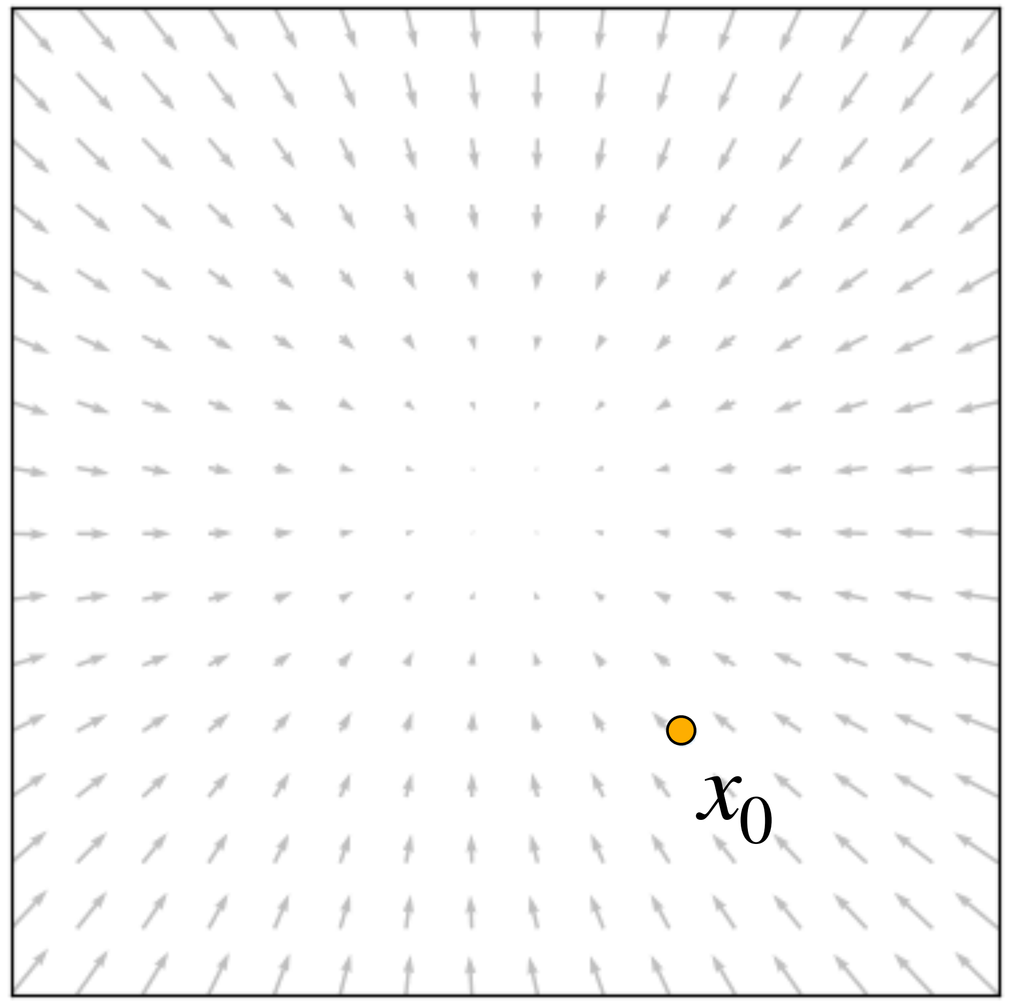
Chen et al. *Neural Ordinary Differential Equations* (NeurIPS 2018)

Continuous Normalizing Flows

Dynamical perspective

Flow ODE

$$\dot{\psi}_t(x_0) = v_t(\psi_t(x_0))$$



$$\psi_t(x_0) = x_0 + \int_0^t v_s(\psi_s(x_0))ds$$

Chen et al. *Neural Ordinary Differential Equations* (NeurIPS 2018)

Continuous Normalizing Flows

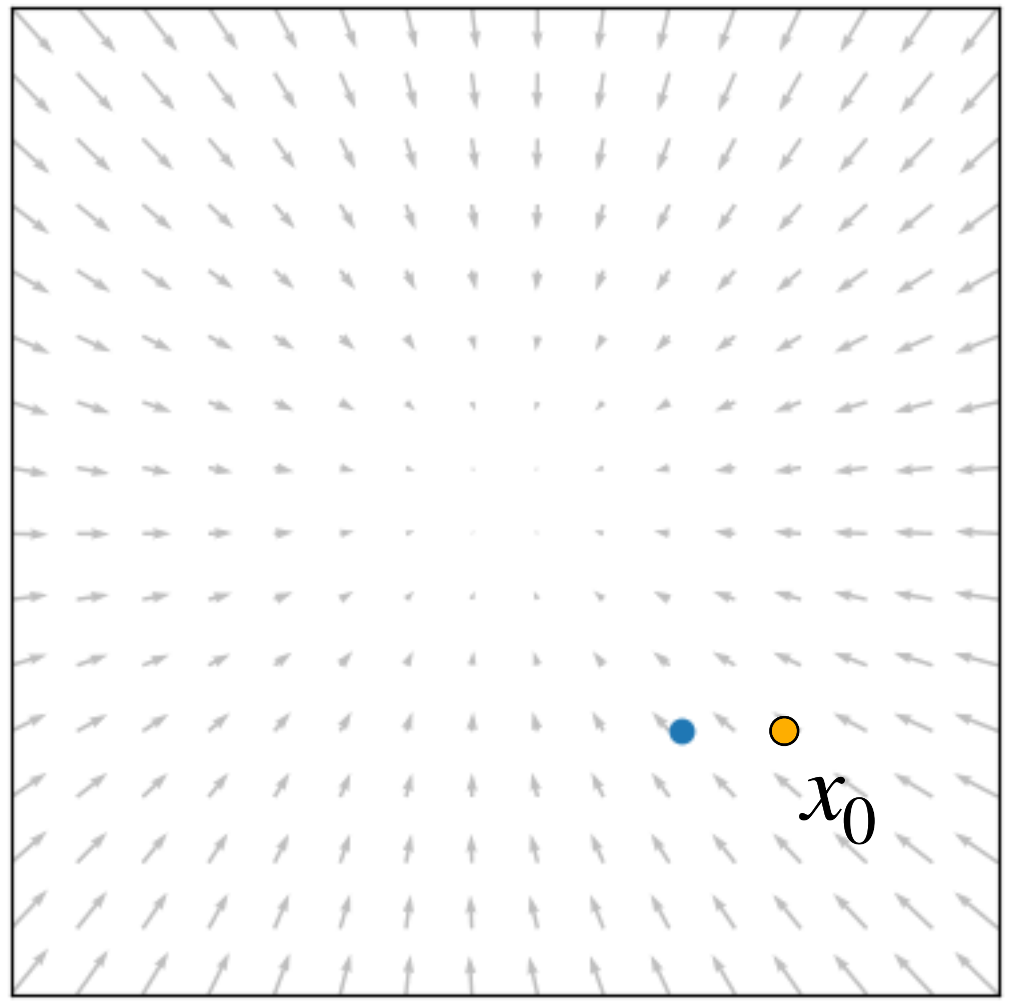
Dynamical perspective

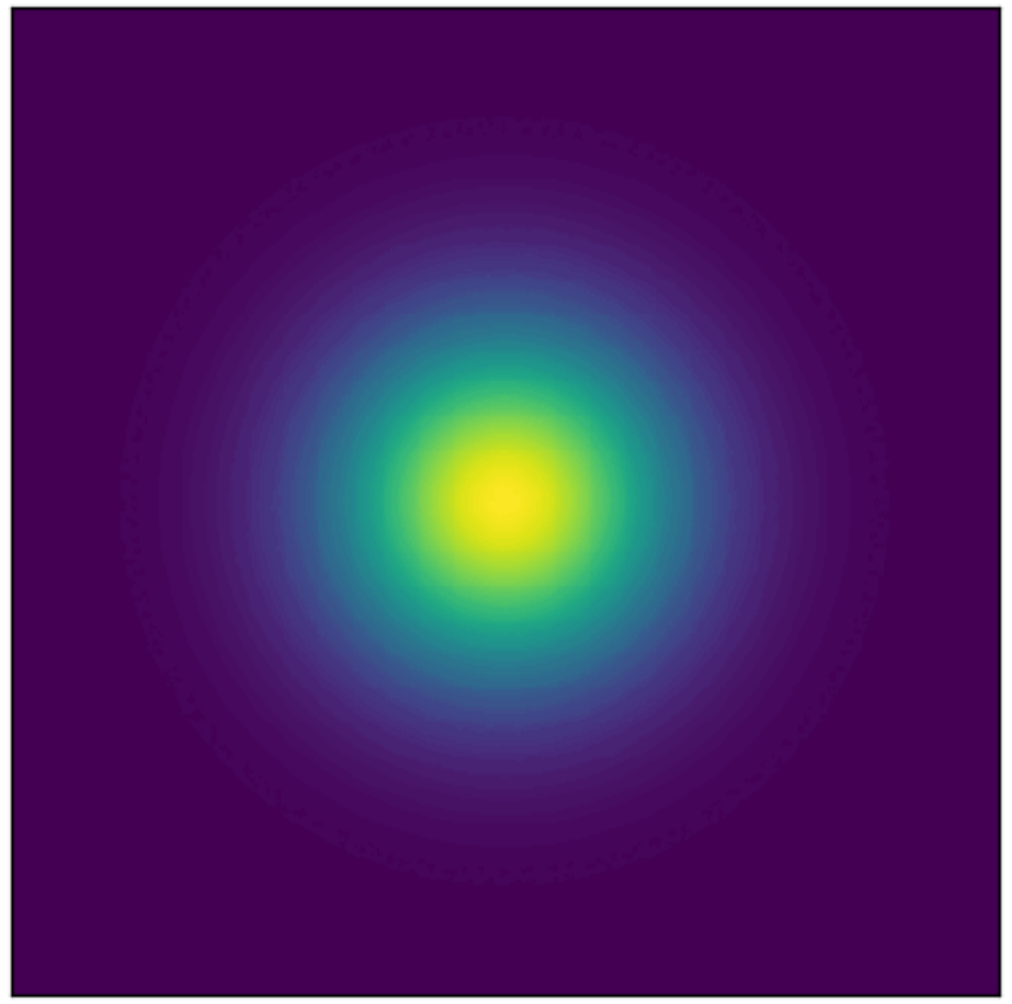
Flow ODE

$$\dot{\psi}_t(x_0) = v_t(\psi_t(x_0))$$

Continuity Equation PDE

$$\dot{p}_t = -\operatorname{div}(p_t v_t)$$





v_t generates p_t iff the continuity equation holds $\forall t \in [0,1], x \in \mathcal{X}$

Chen et al. *Neural Ordinary Differential Equations* (NeurIPS 2018)

Continuous Normalizing Flows

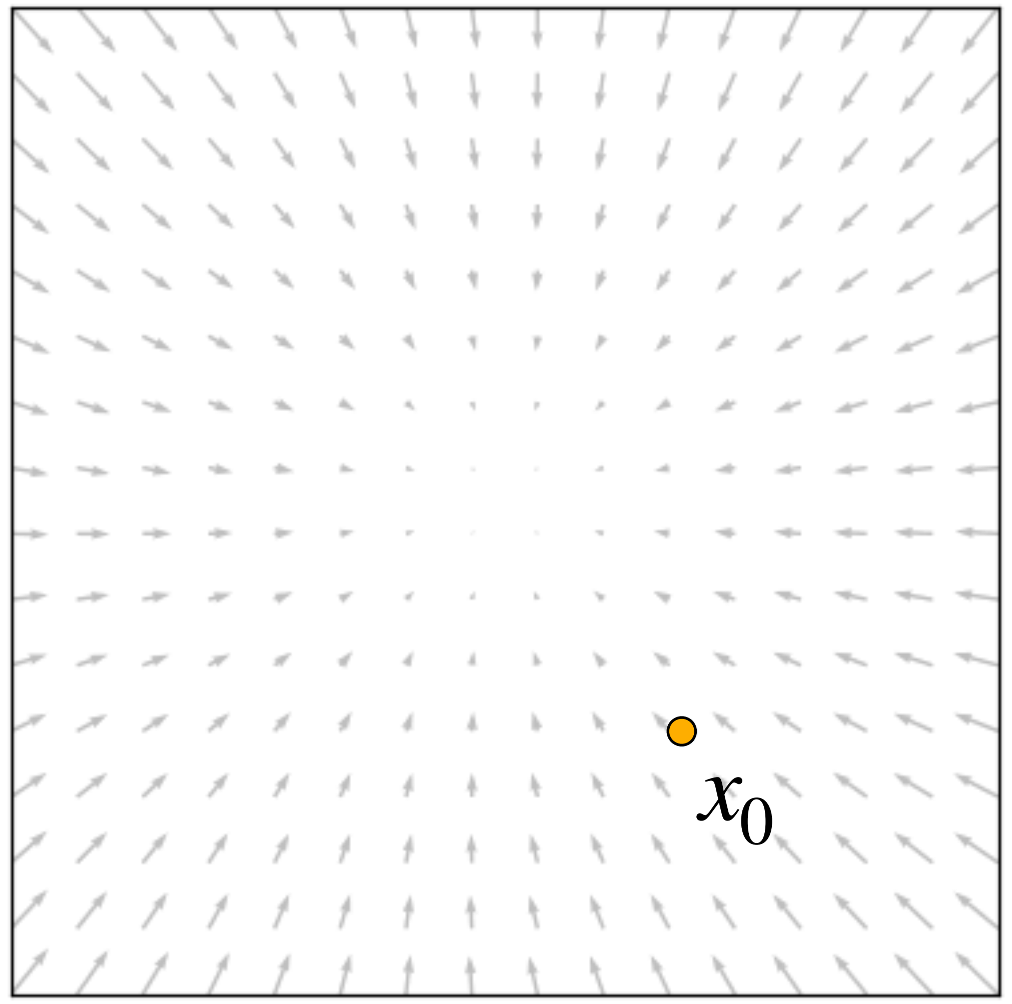
Dynamical perspective

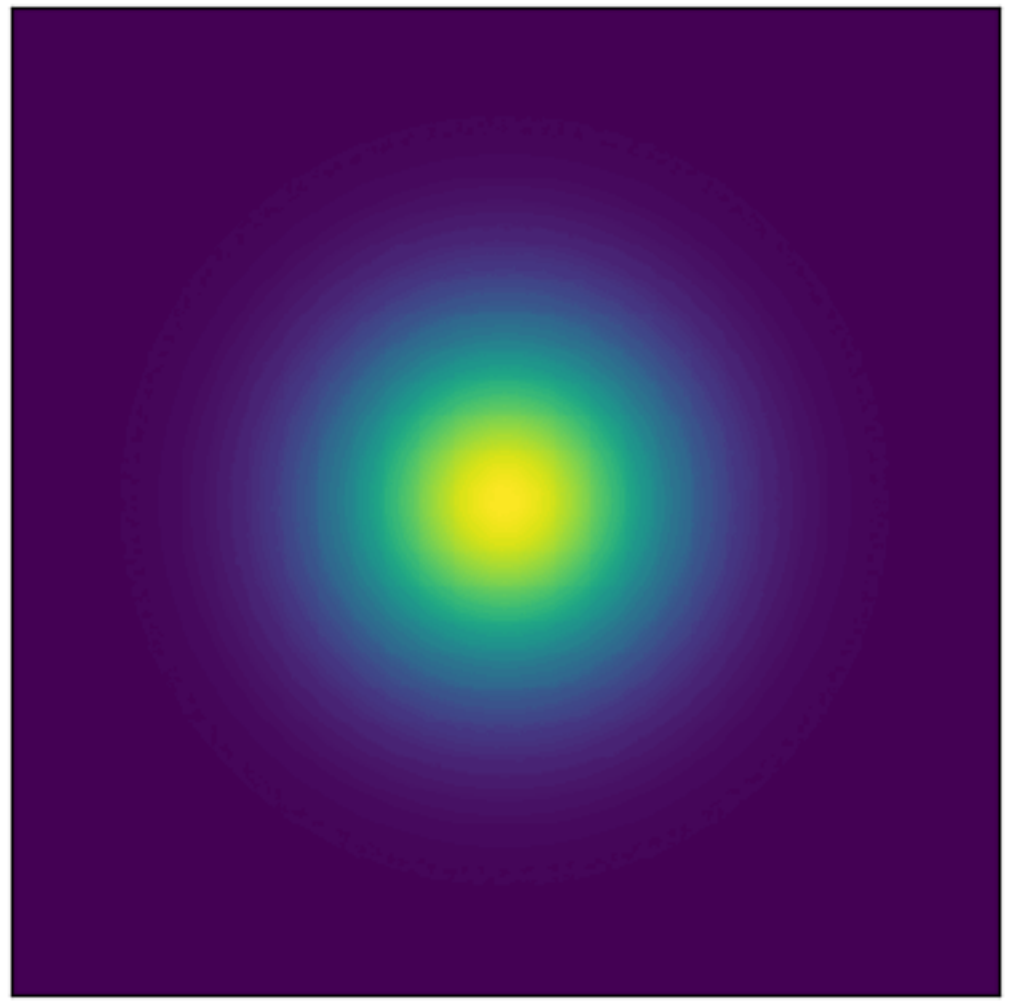
Flow ODE

$$\dot{\psi}_t(x_0) = v_t(\psi_t(x_0))$$

Continuity Equation PDE

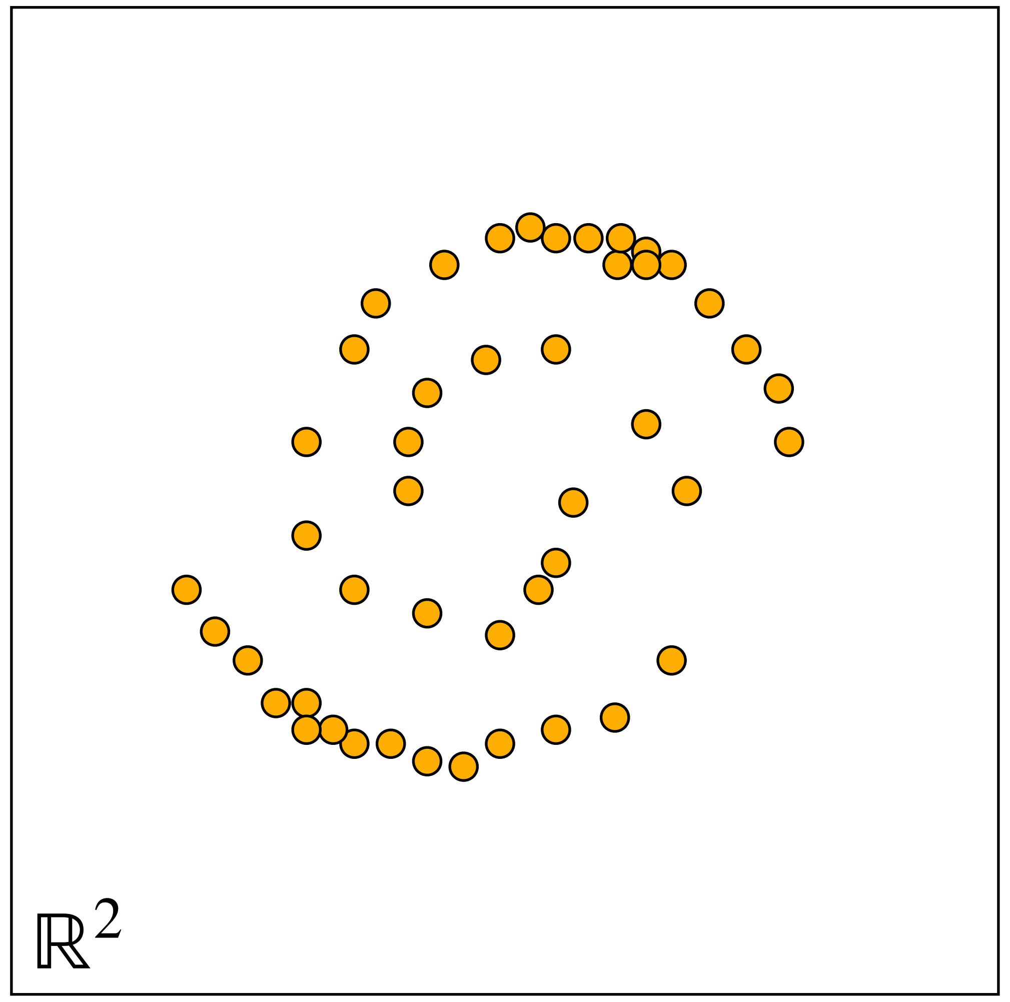
$$\dot{p}_t = -\operatorname{div}(p_t v_t)$$





Given:

Samples $x_1 \sim q$



Goal: find velocity field $v_t(x)$ s.t. $p_1 \approx q$

Chen et al. *Neural Ordinary Differential Equations* (NeurIPS 2018)

Training flows

$x_t = \psi_t(x_0)$

Continuity Equation PDE

$$\dot{p}_t = -\operatorname{div}(p_t v_t)$$

Instantaneous change of coordinates (trajectory x_t)

$$\frac{d}{dt}\log p_t(x_t) = -\operatorname{div}(v_t(x_t))$$

Log-likelihood computation

$$\log p_1(x_1) = \log p(x_0) + \int_1^0 \operatorname{div}(v_t(x_t))dt$$

$$x_t = x_1 + \int_1^t v_s(x_s)ds$$

Chen et al., *Neural Ordinary Differential Equations.* (NeurIPS 2018)

Training flows

$x_t = \psi_t(x_0)$

Log-likelihood computation

$$\log p_1(x_1) = \log p(x_0) + \int_1^0 \mathrm{div}(v_t(x_t))dt$$

$$x_t = x_1 + \int_1^t v_s(x_s)ds$$

Requires:

- **Simulating** x_t
- **Backprop through simulation**
- **(Unbiased) estimator of** $\mathrm{div}(v_t)$
- **Can compute** $\log p(x)$

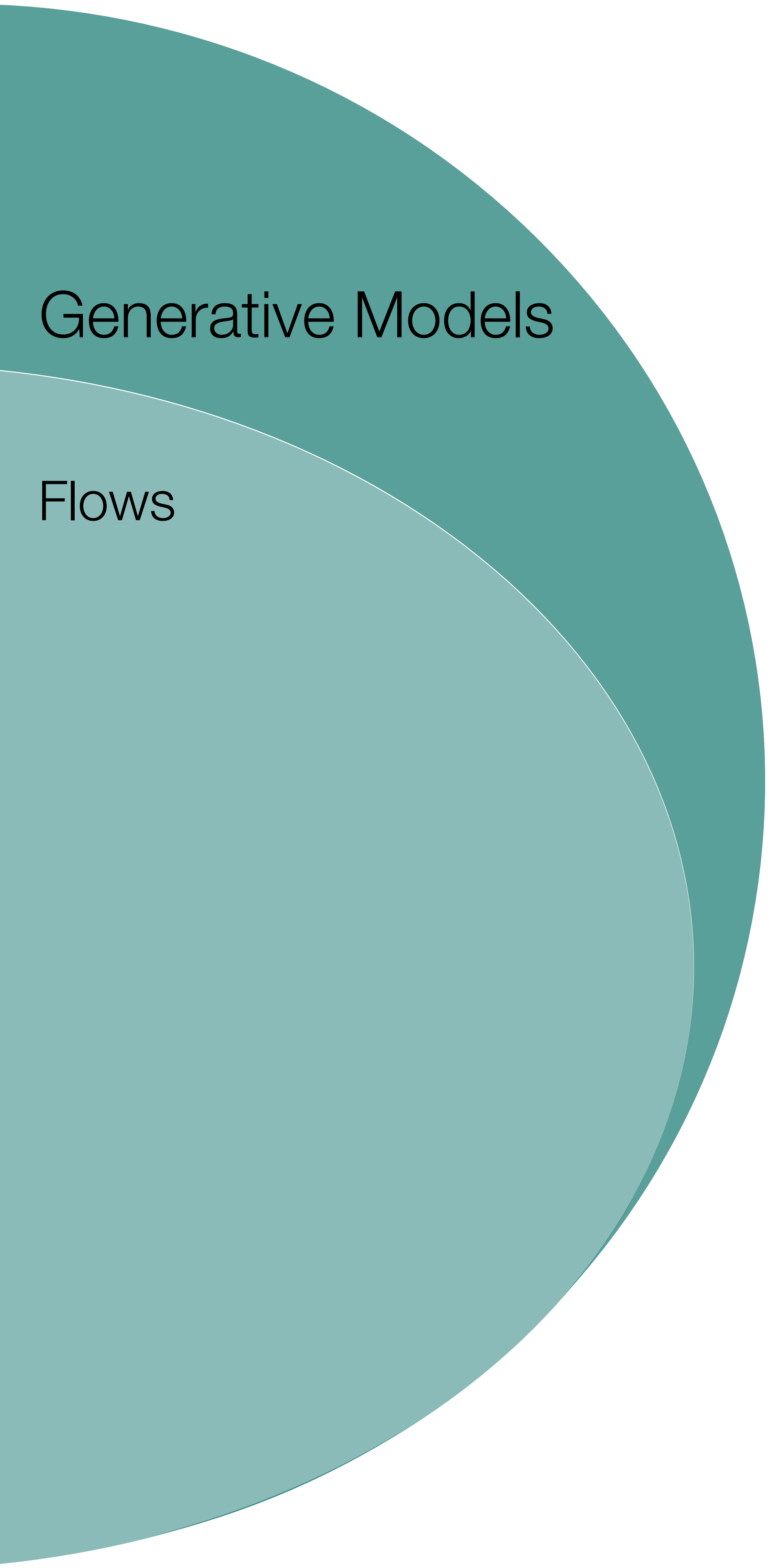
Neural Network

v_t

$$D_{\mathrm{KL}}(q \,\|\, p_1) = -\mathbb{E}_{x\sim q} \log p_1(x) + c$$

Maximum Likelihood Objective

Chen et al., *Neural Ordinary Differential Equations.* (NeurIPS 2018)



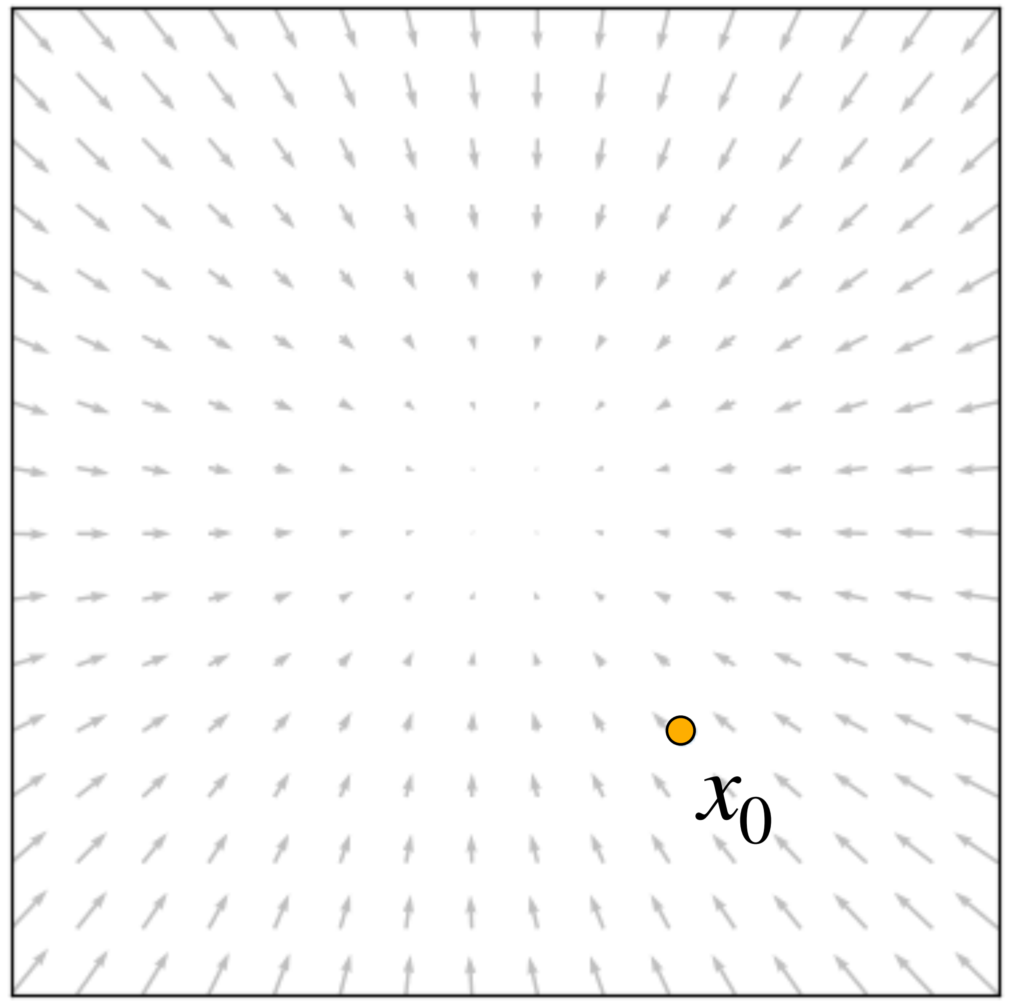
Simulation-Free Continuous Normalizing Flows for Generative Modeling

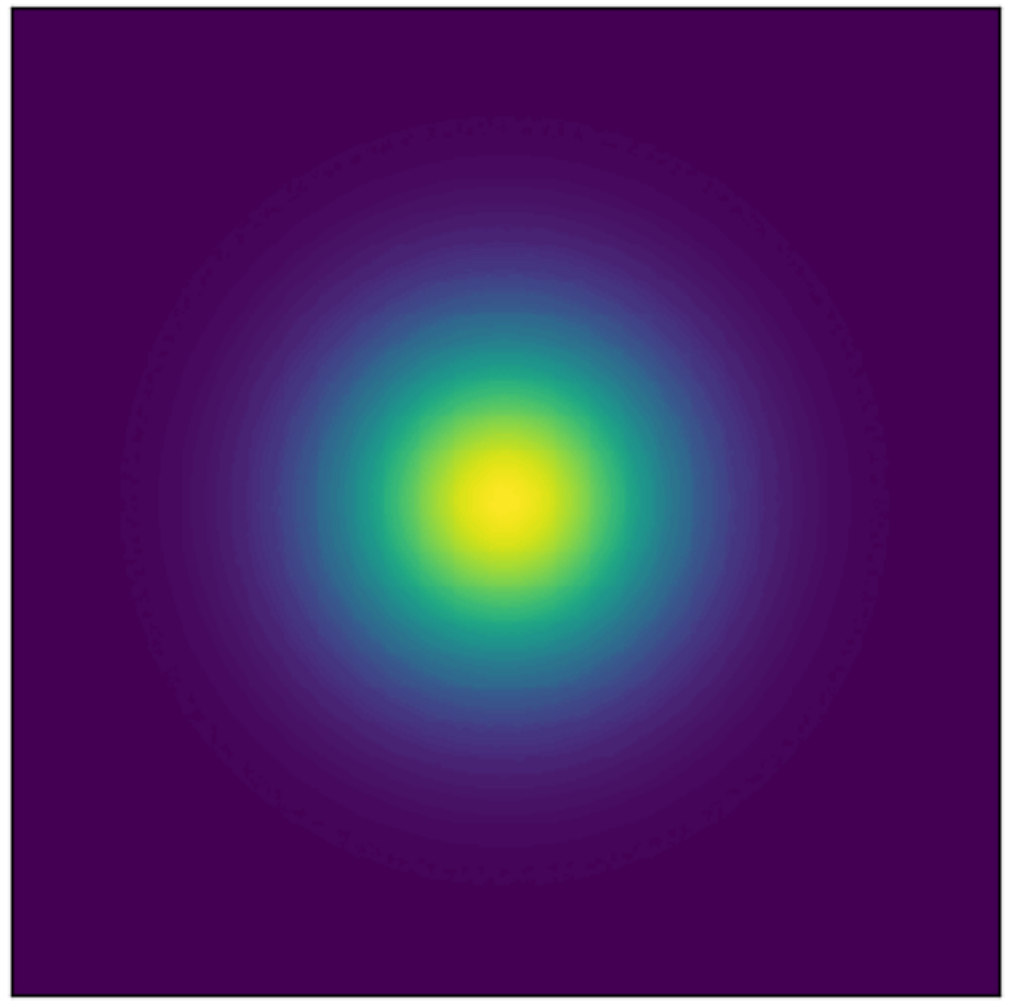
Flow ODE

$$\dot{\psi}_t(x_0) = v_t(\psi_t(x_0))$$

Continuity Equation PDE

$$\dot{p}_t = -\operatorname{div}(p_t v_t)$$





Can we find a **simulation-free** training scheme for CNFs?

Too Much Freedom?

$$D_{\mathrm{KL}}(q \,\|\, p_1) = -\,\mathbb{E}_{x \sim q} \log p_1(x) + c$$

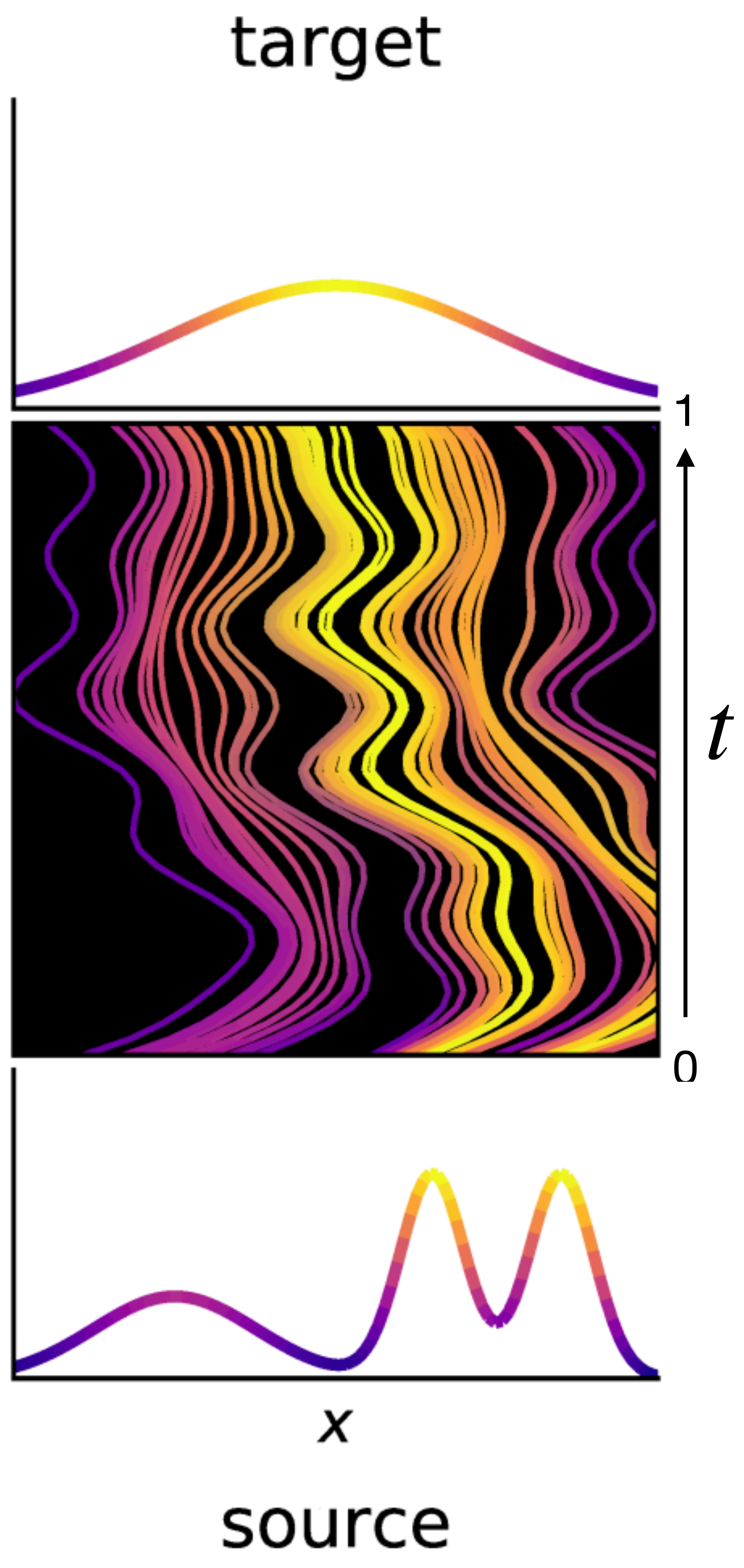


Image from: Finlay et al., *How to train your neural ODE: the world of Jacobian and kinetic regularization*. (ICML 2020)

Degrees of Freedom

$$D_{\mathrm{KL}}(q \,\|\, p_1) = -\,\mathbb{E}_{x \sim q} \log p_1(x) + c$$

Assume: $D_{\mathrm{KL}}(q \,\|\, p_1)$ is minimized

Infinitely many probability paths p_t

Infinitely many velocity fields v_t

Degrees of Freedom

How to reduce degrees of freedom?

Regularization

Regularization

$$D_{\mathrm{KL}}(q \,\|\, p_1) = -\mathbb{E}_{x\sim q} \log p_1(x) + c$$

$+$Kinetic Regularization

Still need to:

- **Simulate x_t**
- **Backprop through simulation**
- **(Unbiased) estimator of $\mathrm{div}(v_t)$**
- **Can compute $\log p(x)$**

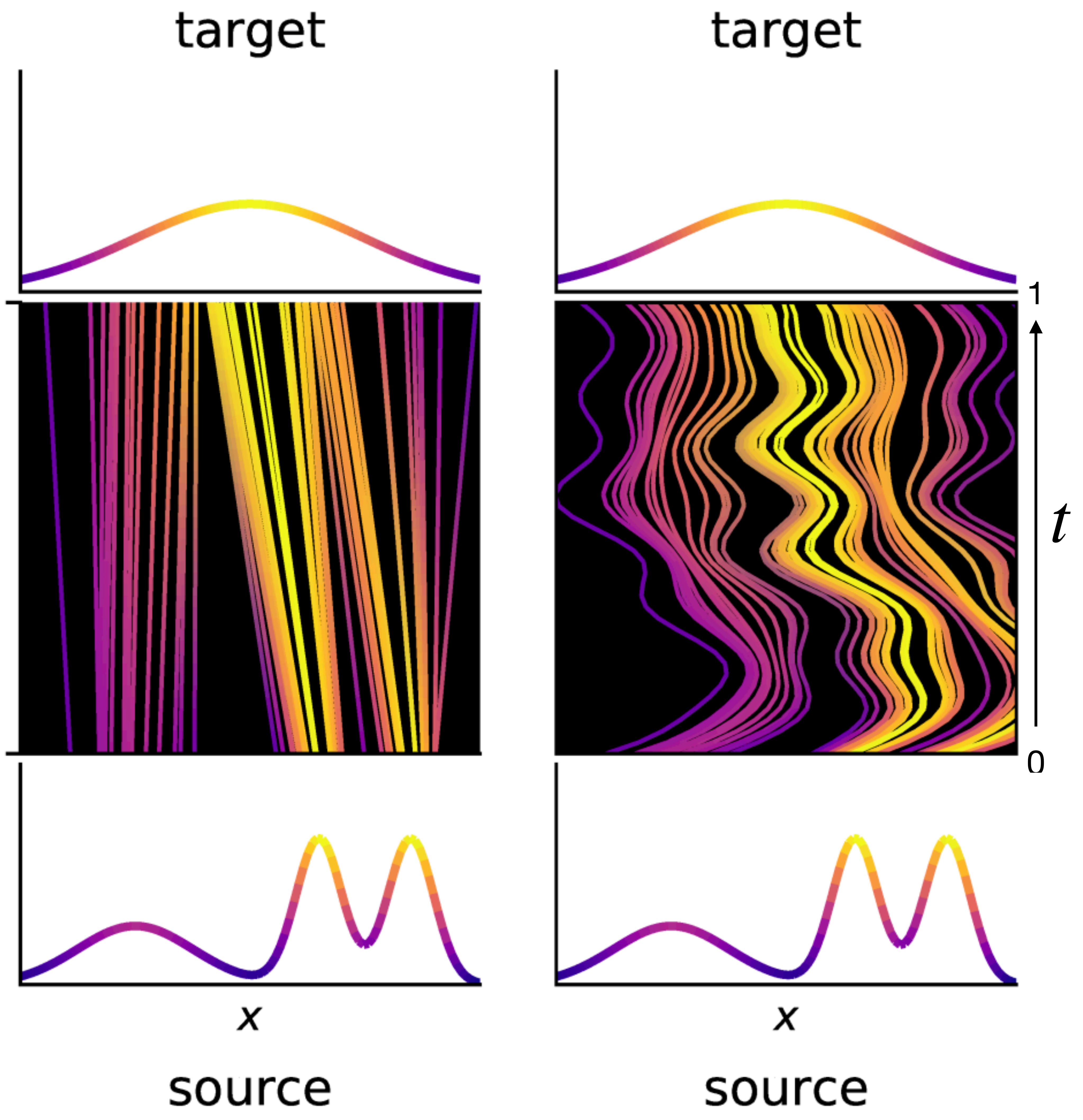


Image from: Finlay et al., *How to train your neural ODE: the world of Jacobian and kinetic regularization.* (ICML 2020)

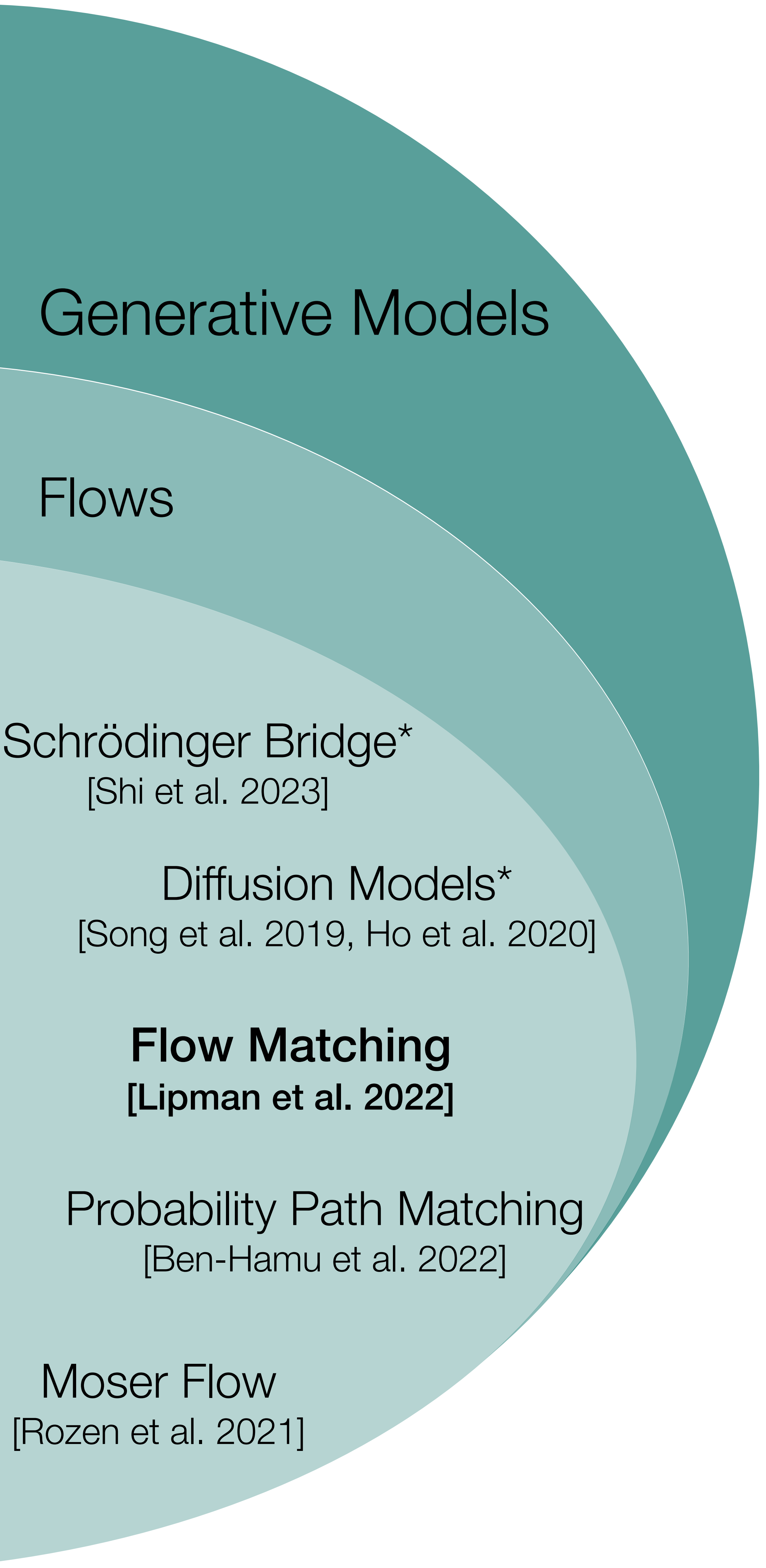
Degrees of Freedom

How to reduce degrees of freedom?

Regularization

Supervision

Simulation-Free Continuous Normalizing Flows for Generative Modeling



Achieve simulation free training by adding supervision!

Can we add more supervision?

Goal: find velocity field $v_t(x)$ s.t. $p_1 \approx q$

Degrees of Freedom

- Probability path p_t
- Velocity field v_t

Can we add more supervision?

Goal: find velocity field $v_t(x)$ s.t. $p_1 \approx q$

Degrees of Freedom

- Probability path p_t
- Velocity field v_t

Moser Flow: Divergence-based Generative Modeling on Manifolds

Noam Rozen[1] **Aditya Grover**[2,3] **Maximilian Nickel**[2] **Yaron Lipman**[1,2]

[1]Weizmann Institute of Science [2]Facebook AI Research [3]UCLA

NeurIPS 2021

Matching Normalizing Flows and Probability Paths on Manifolds

Heli Ben-Hamu [*1] **Samuel Cohen** [*2 3] **Avishek Joey Bose** [2] **Brandon Amos** [2] **Aditya Grover** [2] **Maximilian Nickel** [2] **Ricky T. Q. Chen** [2] **Yaron Lipman** [1 2]

ICML 2022

Flow Matching

Degrees of Freedom

✗ Probability path p_t
✗ Velocity field v_t

FLOW MATCHING FOR GENERATIVE MODELING

Yaron Lipman[1,2] **Ricky T. Q. Chen**[1] **Heli Ben-Hamu**[2] **Maximilian Nickel**[1] **Matt Le**[1]
[1]Meta AI (FAIR) [2]Weizmann Institute of Science

ICLR 2023

FLOW STRAIGHT AND FAST: LEARNING TO GENERATE AND TRANSFER DATA WITH RECTIFIED FLOW

Xingchao Liu*, **Chengyue Gong***, **Qiang Liu**
Department of Computer Science
University of Texas at Austin
{xcliu, cygong, lqiang}@cs.utexas.edu

ICLR 2023

BUILDING NORMALIZING FLOWS WITH STOCHASTIC INTERPOLANTS

Michael S. Albergo
Center for Cosmology and Particle Physics
New York University
New York, NY 10003, USA
albergo@nyu.edu

Eric Vanden-Eijnden
Courant Institute of Mathematical Sciences
New York University
New York, NY 10012, USA
eve2@cims.nyu.edu

ICLR 2023

Flow Matching

$$L_{\mathrm{FM}}(q \| p_1) = \min \mathbb{E}_{t, p_t(x)} \| v_t(x) - u_t(x) \|^2$$

Degrees of Freedom

- ✗ Probability path p_t
- ✗ Velocity field v_t

Construct:

- Target probability path p_t s.t. $p_0 = p$, $p_1 \approx q$
- Generating velocity field u_t

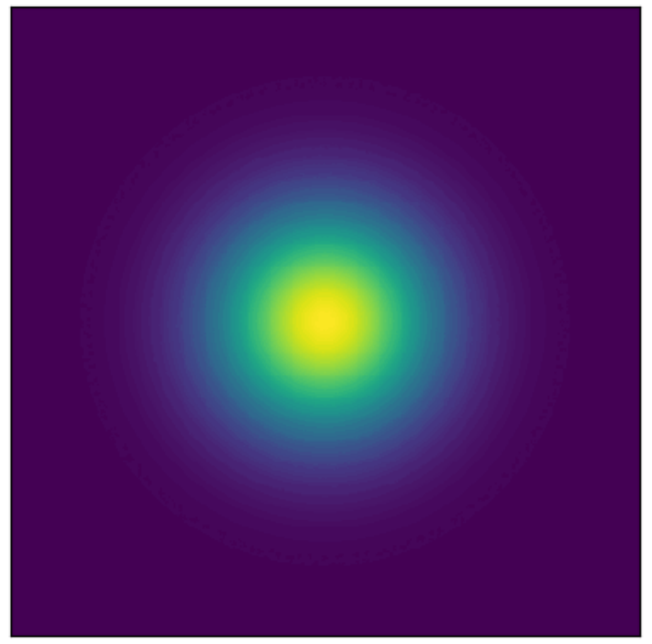
Find a tractable optimization objective

Lipman et al., *Flow Matching for Generative Modeling.* (ICLR 2023)

Conditional Probability Paths

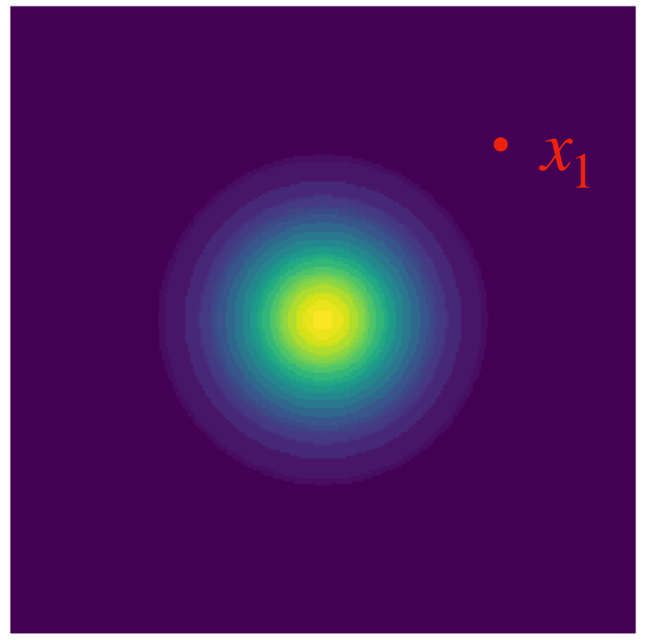
Law of total probability

$$p_t(x) = \int p_t(x|x_1)q(x_1)dx_1$$



Marginal path

$$p_t(x|x_1)$$

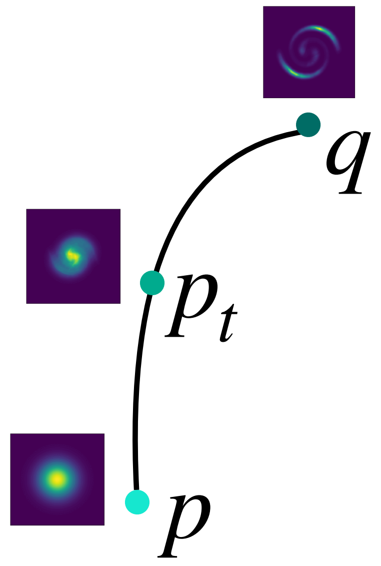


Conditional path

Boundary conditions:

$$p_0 = p$$

$$p_1 = q$$



$$p_0(\cdot|x_1) = p$$

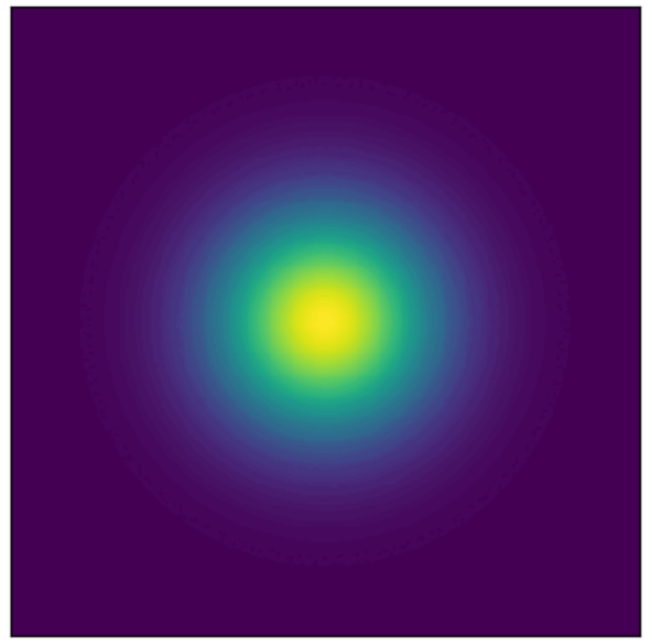
$$p_1(\cdot|x_1) = \delta_{x_1}$$

Lipman et al., *Flow Matching for Generative Modeling.* (ICLR 2023)

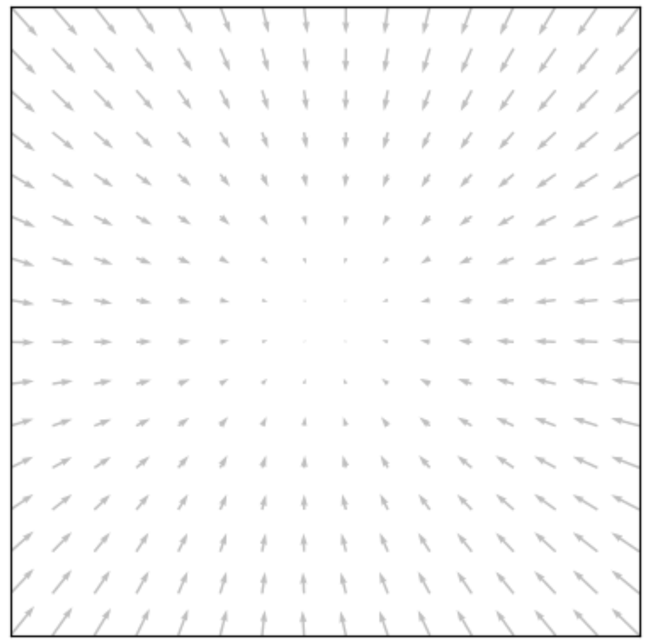
The marginalization “trick”

Marginal path

$$p_t(x) = \int p_t(x|x_1)q(x_1)dx_1$$

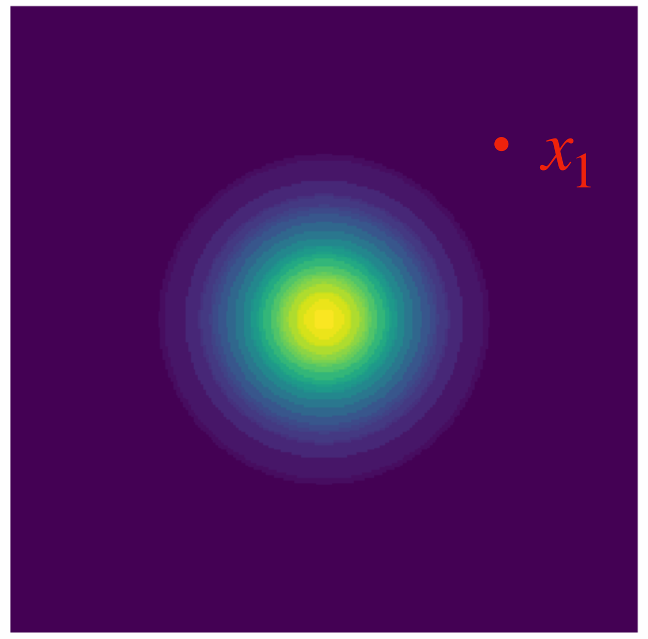


$$u_t(x) = \int u_t(x|x_1)\frac{p_t(x|x_1)q(x_1)}{p_t(x)}dx_1$$

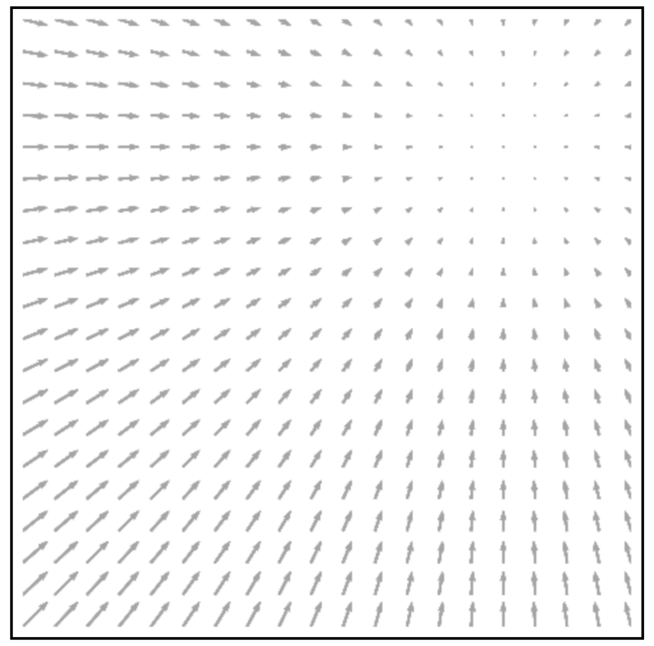


Conditional path

$$p_t(x|x_1)$$



$$u_t(x|x_1)$$



Lipman et al., *Flow Matching for Generative Modeling.* (ICLR 2023)

The marginalization "trick"

Continuity Equation PDE

$$\dot{p}_t = -\operatorname{div}(p_t v_t)$$

Claim: $u_t(x) = \int u_t(x|x_1)\frac{p_t(x|x_1)q(x_1)}{p_t(x)}dx_1 \xrightarrow{\text{generates}} p_t(x) = \int p_t(x|x_1)q(x_1)dx_1$

Proof:

$$\begin{aligned}
\operatorname{div}(p_t(x)u_t(x)) &= \\
&= \operatorname{div}\left(\int u_t(x|x_1)p_t(x|x_1)q(x_1)dx_1\right) = \\
&= \int \operatorname{div}(u_t(x|x_1)p_t(x|x_1))q(x_1)dx_1 = \\
&= \int -\frac{d}{dt}p_t(x|x_1)q(x_1)dx_1 = \\
&= -\frac{d}{dt}\int p_t(x|x_1)q(x_1)dx_1 = \\
&= -\frac{d}{dt}p_t(x)
\end{aligned}$$

Lipman et al., *Flow Matching for Generative Modeling.* (ICLR 2023)

Flow Matching

$$L_{\mathrm{FM}}(q \| p_1) = \min \mathbb{E}_{t, p_t(x)} \| v_t(x) - u_t(x) \|^2$$

Degrees of Freedom

✗ Probability path p_t

✗ Velocity field v_t

$$u_t(x) = \int u_t(x \mid x_1) \frac{p_t(x \mid x_1) q(x_1)}{p_t(x)} dx_1$$

Construct:

- Target probability path p_t s.t. $p_0 = p$, $p_1 \approx q$
- Generating velocity field u_t ✓

Find a tractable optimization objective

Lipman et al., *Flow Matching for Generative Modeling.* (ICLR 2023)

Conditional Flow Matching Loss

$$L_{\mathrm{FM}}(q||p_1) = \min \mathbb{E}_{t,\, p_t(x)} ||v_t(x) - u_t(x)||^2$$

$$L_{\mathrm{CFM}}(q||p_1) = \min \mathbb{E}_{t,\, q(x_1),\, p_t(x|x_1)} ||v_t(x) - u_t(x \,|\, x_1)||^2$$

The gradients of losses coincide:

$$\nabla_\theta L_{\mathrm{FM}} = \nabla_\theta L_{\mathrm{CFM}}$$

Lipman et al., *Flow Matching for Generative Modeling.* (ICLR 2023)

Conditional Flow Matching Loss

$$L_{\mathrm{FM}}(q\|p_1) = \min \mathbb{E}_{t,\,p_t(x)}\|v_t(x) - u_t(x)\|^2 \qquad L_{\mathrm{CFM}}(q\|p_1) = \min \mathbb{E}_{t,\,q(x_1),\,p_t(x|x_1)}\|v_t(x) - u_t(x\,|\,x_1)\|^2$$

$$\|v_t(x) - u_t(x)\|^2 = \underline{\|v_t(x)\|^2} - 2\langle v_t(x), u_t(x)\rangle + \|\cancel{u_t(x)}\|^2$$

Do not depend on θ

$$\|v_t(x) - u_t(x\,|\,x_1)\|^2 = \underline{\|v_t(x)\|^2} - 2\langle v_t(x), u_t(x\,|\,x_1)\rangle + \|\cancel{u_t(x\,|\,x_1)}\|^2$$

Law of total expectation $\quad \mathbb{E}_{t,\,p_t(x)}\|v_t(x)\|^2 = \mathbb{E}_{t,\,q(x_1),p_t(x|x_1)}\|v_t(x)\|^2$

Lipman et al., *Flow Matching for Generative Modeling.* (ICLR 2023)

Conditional Flow Matching Loss

$$L_{\mathrm{FM}}(q\|p_1) = \min \mathbb{E}_{t,\, p_t(x)} \|v_t(x) - u_t(x)\|^2 \qquad L_{\mathrm{CFM}}(q\|p_1) = \min \mathbb{E}_{t,\, q(x_1),\, p_t(x|x_1)} \|v_t(x) - u_t(x \mid x_1)\|^2$$

$$\|v_t(x) - u_t(x)\|^2 = \underline{\|v_t(x)\|^2} - 2\langle v_t(x), u_t(x)\rangle + \cancel{\|u_t(x)\|^2}$$

Do not depend on θ

$$\|v_t(x) - u_t(x \mid x_1)\|^2 = \underline{\|v_t(x)\|^2} - 2\langle v_t(x), u_t(x \mid x_1)\rangle + \cancel{\|u_t(x \mid x_1)\|^2}$$

Lipman et al., *Flow Matching for Generative Modeling.* (ICLR 2023)

Conditional Flow Matching Loss

$$L_{\mathrm{FM}}(q\|p_1) = \min \mathbb{E}_{t,\, p_t(x)} \|v_t(x) - u_t(x)\|^2 \qquad L_{\mathrm{CFM}}(q\|p_1) = \min \mathbb{E}_{t,\, q(x_1),\, p_t(x|x_1)} \|v_t(x) - u_t(x\,|\,x_1)\|^2$$

$$\mathbb{E}_{p_t(x)} \langle v_t(x), u_t(x) \rangle = \int \langle v_t(x), \frac{\int u_t(x\,|\,x_1) p_t(x\,|\,x_1) q(x_1)}{p_t(x)} dx_1 \rangle p_t(x) dx =$$

$$= \int \langle v_t(x), \int u_t(x\,|\,x_1) p_t(x\,|\,x_1) q(x_1) dx_1 \rangle dx =$$

$$= \int \langle v_t(x), u_t(x\,|\,x_1) \rangle p_t(x\,|\,x_1) q(x_1) dx_1 dx =$$

$$= \mathbb{E}_{q(x_1), p_t(x|x_1)} \langle v_t(x), u_t(x\,|\,x_1) \rangle$$

$$u_t(x) = \int u_t(x\,|\,x_1) \frac{p_t(x\,|\,x_1) q(x_1)}{p_t(x)} dx_1$$

Lipman et al., *Flow Matching for Generative Modeling.* (ICLR 2023)

Conditional Flow Matching Loss

$$L_{\mathrm{FM}}(q\|p_1) = \min \mathbb{E}_{t,\, p_t(x)} \|v_t(x) - u_t(x)\|^2 \qquad L_{\mathrm{CFM}}(q\|p_1) = \min \mathbb{E}_{t,\, q(x_1),\, p_t(x|x_1)} \|v_t(x) - u_t(x \mid x_1)\|^2$$

$$\|v_t(x) - u_t(x)\|^2 = \underline{\|v_t(x)\|^2} - 2\underline{\langle v_t(x), u_t(x)\rangle} + \cancel{\|u_t(x)\|^2}$$

Do not depend on θ

$$\|v_t(x) - u_t(x \mid x_1)\|^2 = \underline{\|v_t(x)\|^2} - 2\underline{\langle v_t(x), u_t(x \mid x_1)\rangle} + \cancel{\|u_t(x \mid x_1)\|^2}$$

The gradients of losses coincide:

$$\nabla_\theta L_{\mathrm{FM}} = \nabla_\theta L_{\mathrm{CFM}}$$

Lipman et al., *Flow Matching for Generative Modeling.* (ICLR 2023)

Flow Matching

$$L_{\mathrm{FM}}(q||p_1) = \min \mathbb{E}_{t,\, p_t(x)} \| v_t(x) - u_t(x) \|^2$$

$$L_{\mathrm{CFM}}(q||p_1) = \min \mathbb{E}_{t,\, q(x_1),\, p_t(x|x_1)} \| v_t(x) - u_t(x \mid x_1) \|^2$$

Degrees of Freedom

- ✗ Probability path p_t
- ✗ Velocity field v_t

Construct:

- Target probability path p_t s.t. $p_0 = p$, $p_1 \approx q$
- Generating velocity field u_t

✔

Find a tractable optimization objective

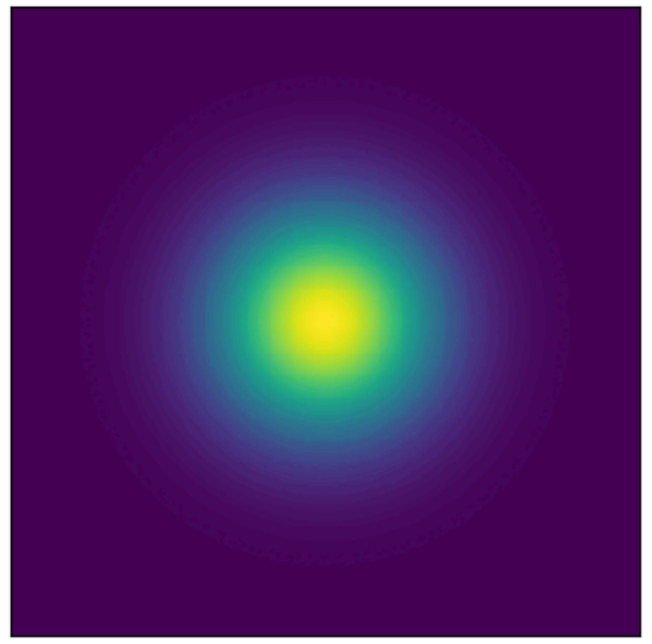
✔

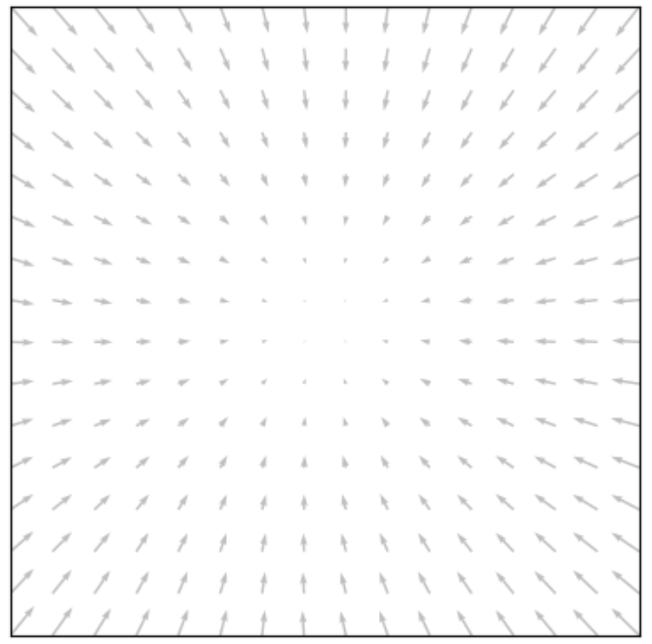
Lipman et al., *Flow Matching for Generative Modeling.* (ICLR 2023)

Conditional Flows

Marginal path

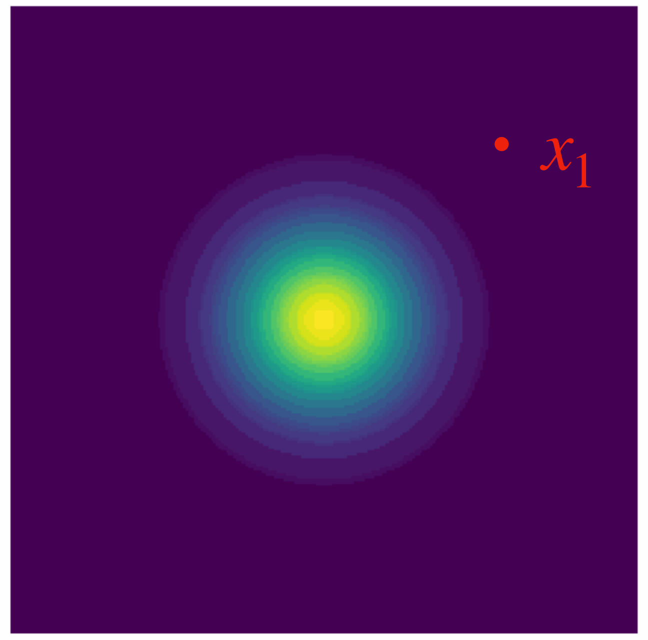
$$p_t(x) = \int p_t(x|x_1)q(x_1)dx_1 \qquad u_t(x) = \int u_t(x|x_1)\frac{p_t(x|x_1)q(x_1)}{p_t(x)}dx_1$$



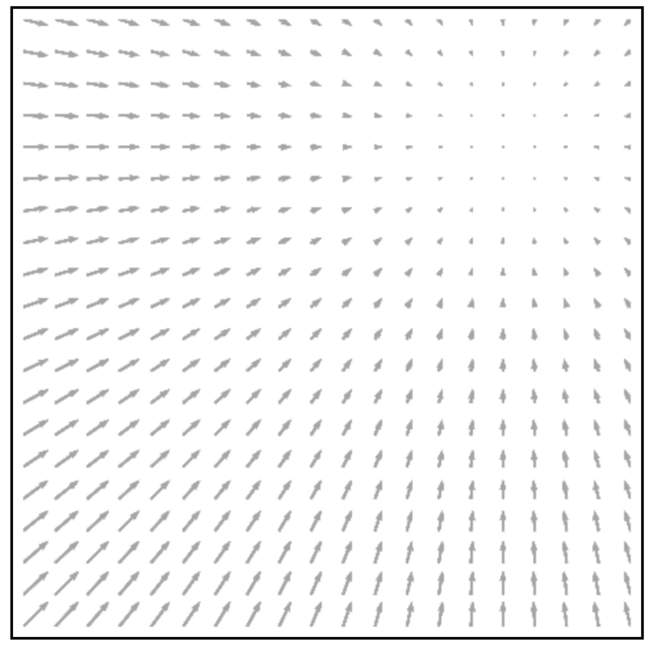


Conditional path

$p_t(x|x_1)$

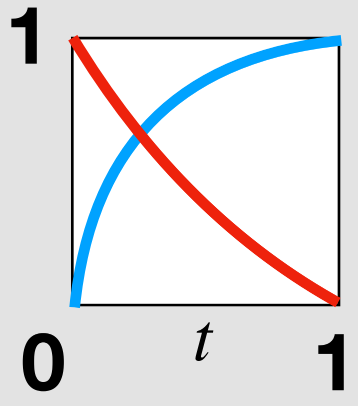


$u_t(x|x_1)$



$$x_t = \psi_t(x_0|x_1) \sim p_t(x|x_1)$$

$$u_t(\psi_t(x_0|x_1)|x_1) = \dot{\psi}_t(x_0|x_1)$$



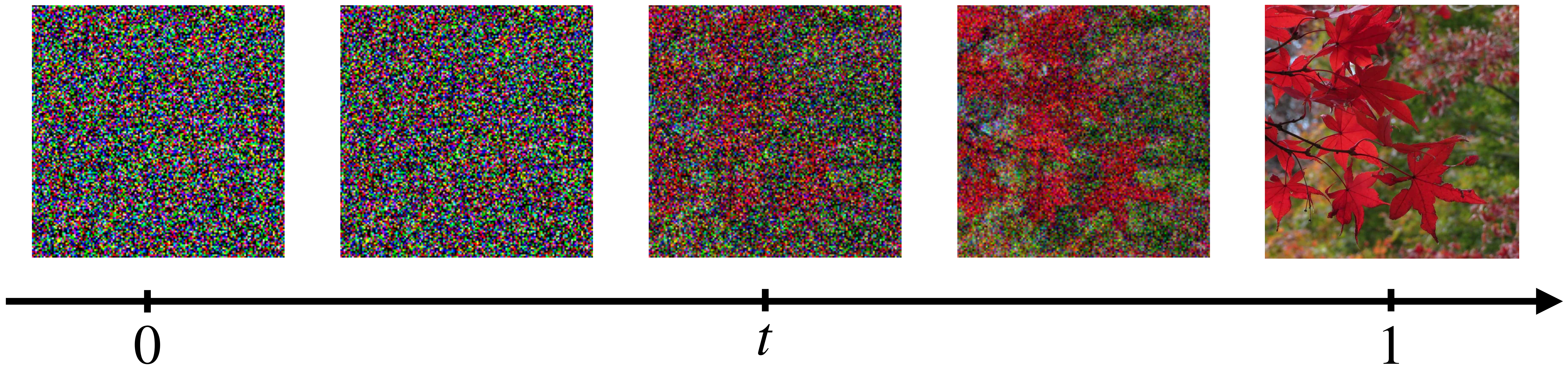
$$x_t = \psi_t(x_0|x_1) = \sigma_t x_0 + \alpha_t x_1$$

$$u_t(x|x_1) = \dot{\sigma}_t \frac{x - \alpha_t x_1}{\sigma_t} + \dot{\alpha}_t x_1$$

Lipman et al., *Flow Matching for Generative Modeling.* (ICLR 2023)

Conditional Flows

$x_t \sim p_t$



$$x_t = \psi_t(x_0 \,|\, x_1) = \sigma_t x_0 + \alpha_t x_1$$

Flow Matching

$$L_{\mathrm{FM}}(q||p_1) = \min \mathbb{E}_{t,\, p_t(x)} ||v_t(x) - u_t(x)||^2$$

$$\Updownarrow$$

$$L_{\mathrm{CFM}}(q||p_1) = \min \mathbb{E}_{t,\, q(x_1),\, p_t(x|x_1)} ||v_t(x) - u_t(x\,|\,x_1)||^2$$

Degrees of Freedom

- ✗ Probability path p_t
- ✗ Velocity field v_t

Construct:

- Target probability path p_t s.t. $p_0 = p$, $p_1 \approx q$
- Generating velocity field u_t

✓

Find a tractable optimization objective

✓

Lipman et al., *Flow Matching for Generative Modeling.* (ICLR 2023)

Flow Matching vs. Diffusion

Algorithm 1: Flow Matching training.

Input : dataset q, noise p
Initialize v^θ
while *not converged* **do**
- $t \sim \mathcal{U}([0,1])$ ▷ sample time
- $x_1 \sim q(x_1)$ ▷ sample data
- $x_0 \sim p(x_0)$ ▷ sample noise
- $x_t = \Psi_t(x_0|x_1)$ ▷ conditional flow
- Gradient step with $\nabla_\theta \|v_t^\theta(x_t) - \dot{x}_t\|^2$

Output: v^θ

Algorithm 2: Diffusion training.

Input : dataset q, noise p
Initialize s^θ
while *not converged* **do**
- $t \sim \mathcal{U}([0,1])$ ▷ sample time
- $x_1 \sim q(x_1)$ ▷ sample data
- $x_t = p_t(x_t|x_1)$ ▷ sample conditional prob
- Gradient step with $\nabla_\theta \|s_t^\theta(x_t) - \nabla_{x_t} \log p_t(x_t|x_1)\|^2$

Output: v^θ

$p_t(x_t | x_1)$ general
$p(x_0)$ is general

$p_t(x_t | x_1)$ closed-form from an SDE $dx_t = f_t dt + g_t dw$

- *Variance Exploding*: $p_t(x | x_1) = \mathcal{N}(x | x_1, \sigma_{1-t}^2 I)$
- *Variance Preserving*: $p_t(x | x_1) = \mathcal{N}(x | \alpha_{1-t} x_1, (1 - \alpha_{1-t}^2) I)$

$$\alpha_t = e^{-\frac{1}{2}T(t)}$$

$p(x_0)$ is Gaussian
$p_0(\cdot | x_1) \approx p$

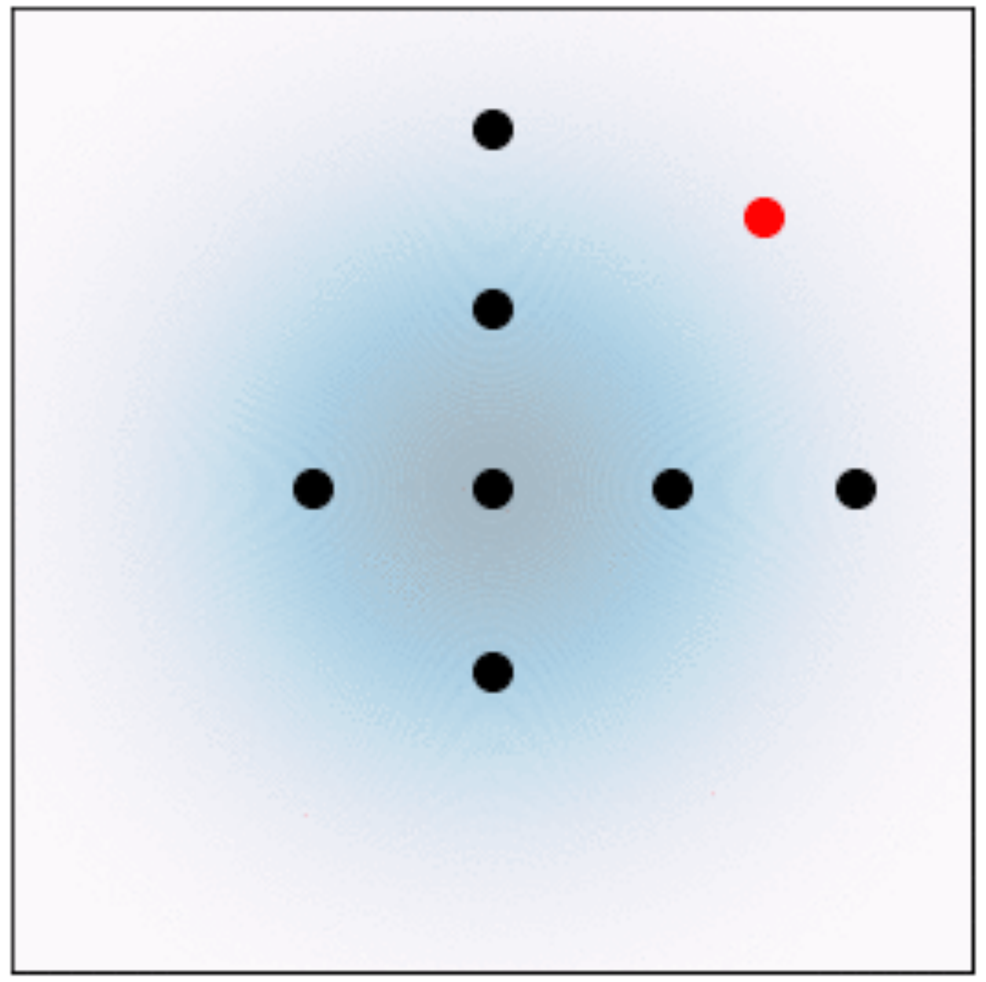
Lipman et al., *Flow Matching for Generative Modeling.* (ICLR 2023)

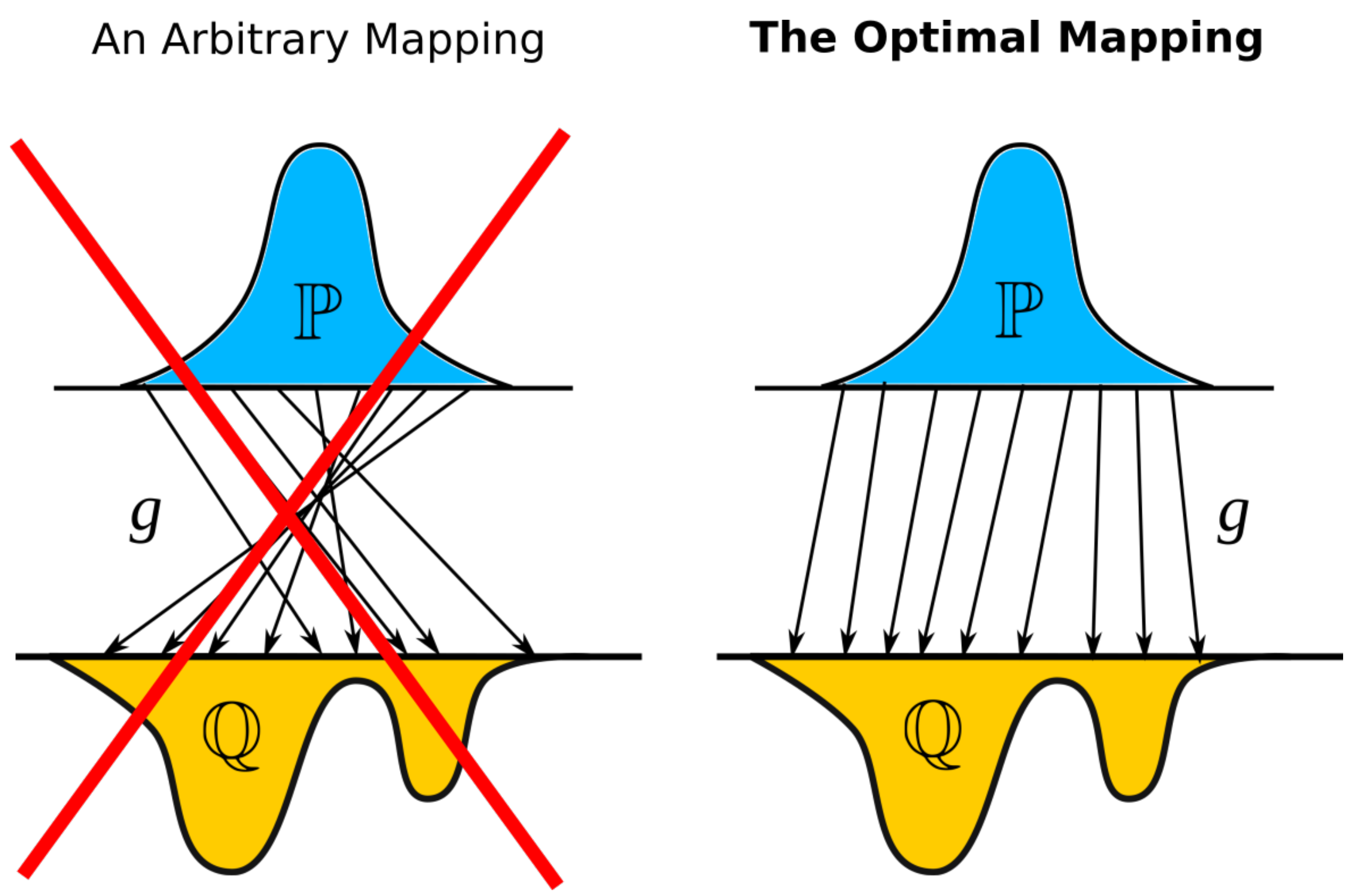
Conditional Optimal Transport Probability Path

- Example of non-diffusion choice

$$p_t(x \mid x_1), u_t(x \mid x_1) \quad \Leftrightarrow \quad \psi_t(x_0 \mid x_1)$$

$$\mathcal{N}(tx_1, (1-t)^2 I), \frac{x_1 - x}{1-t} \qquad\qquad \psi_t(x_0 \mid x_1) = (1-t)x_0 + tx_1$$





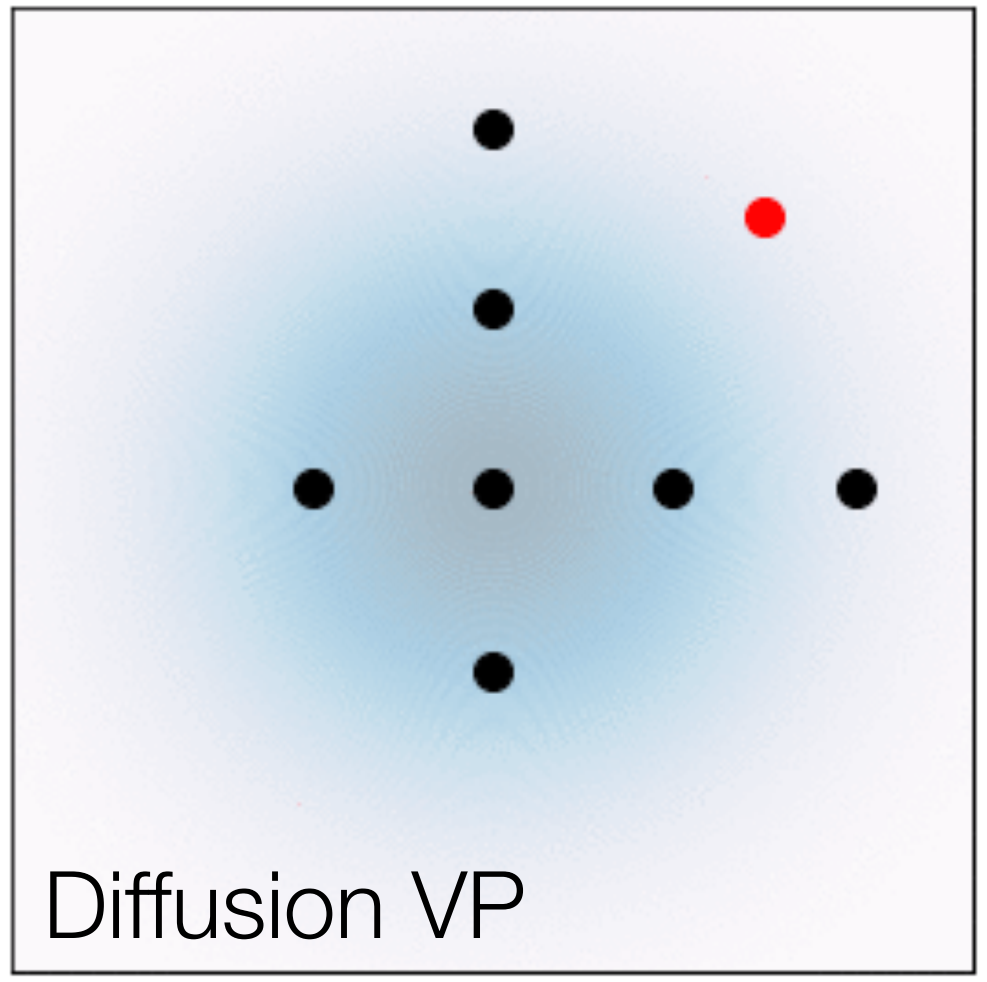
Lipman et al., *Flow Matching for Generative Modeling.* (ICLR 2023)

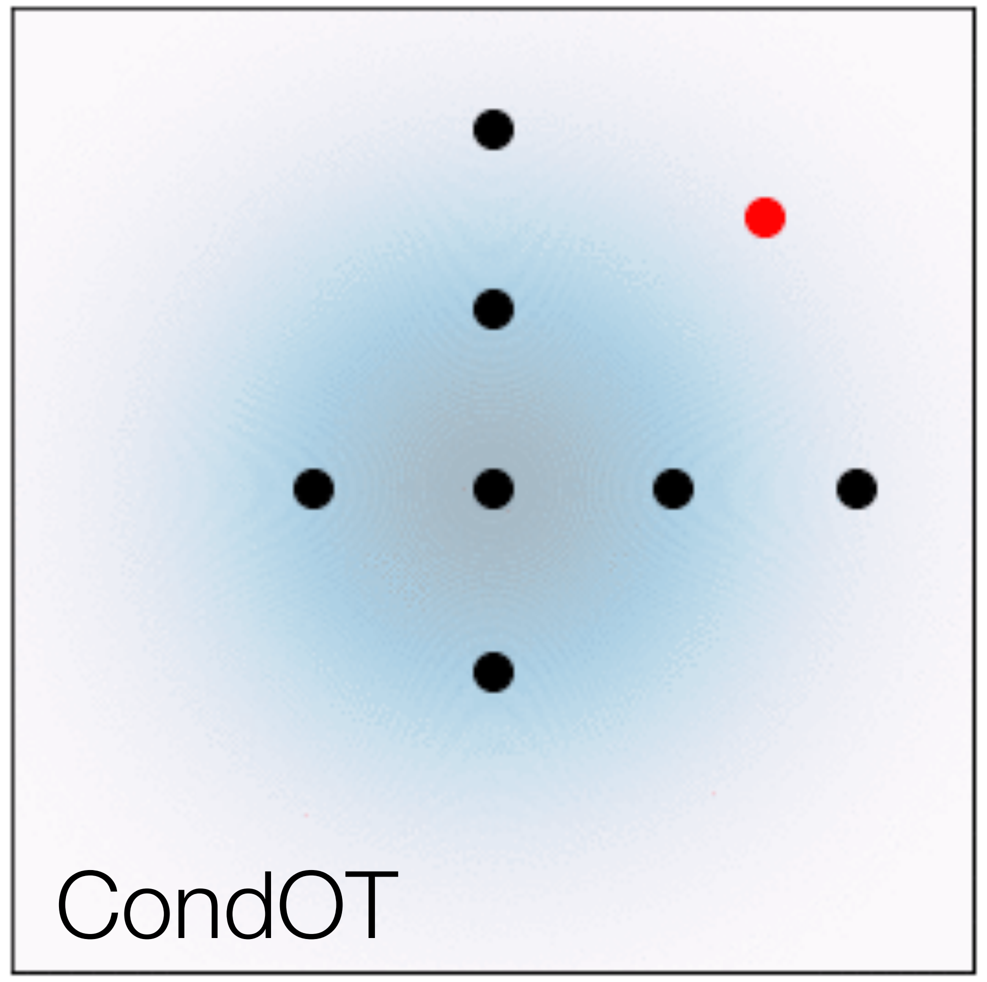
Conditional Optimal Transport Probability Path

- Example of non-diffusion choice

$$p_t(x \mid x_1), u_t(x \mid x_1) \quad \Leftrightarrow \quad \psi_t(x_0 \mid x_1)$$

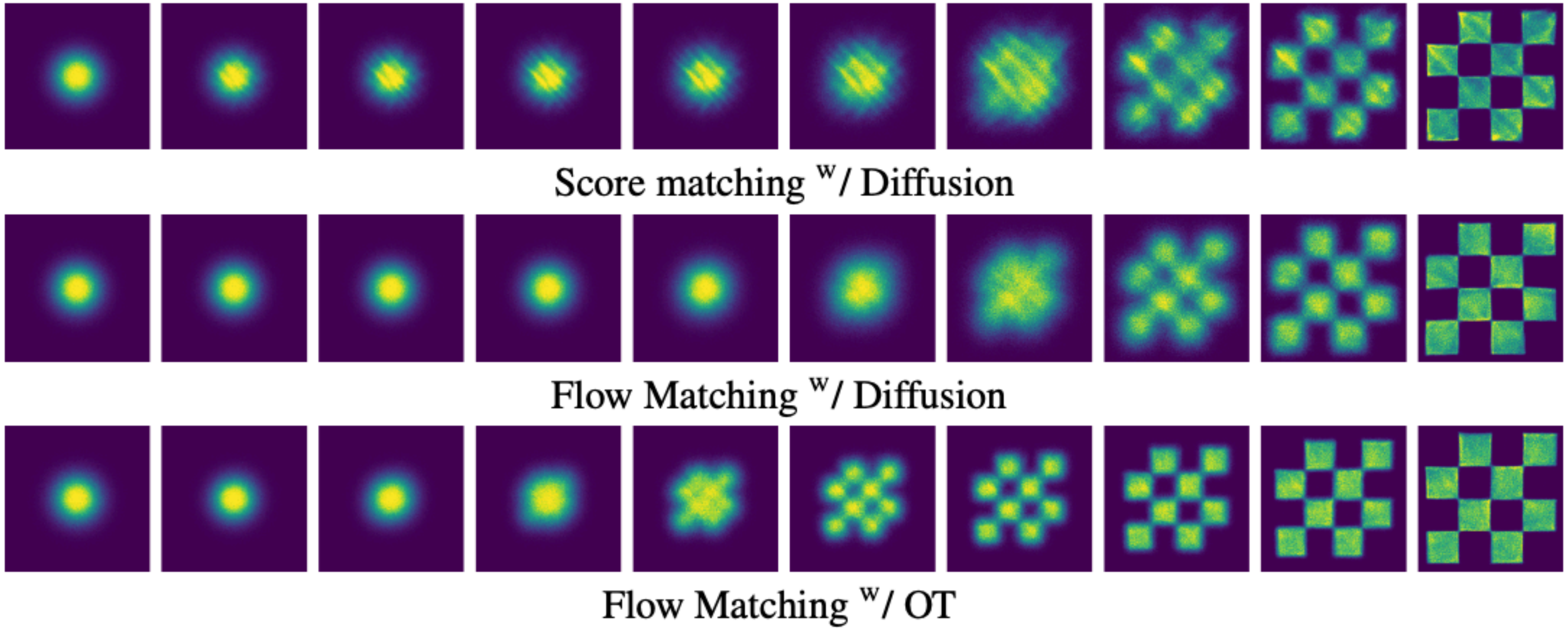
$$\mathcal{N}(tx_1, (1-t)^2 I), \frac{x_1 - x}{1-t} \qquad\qquad \psi_t(x_0 \mid x_1) = (1-t)x_0 + tx_1$$





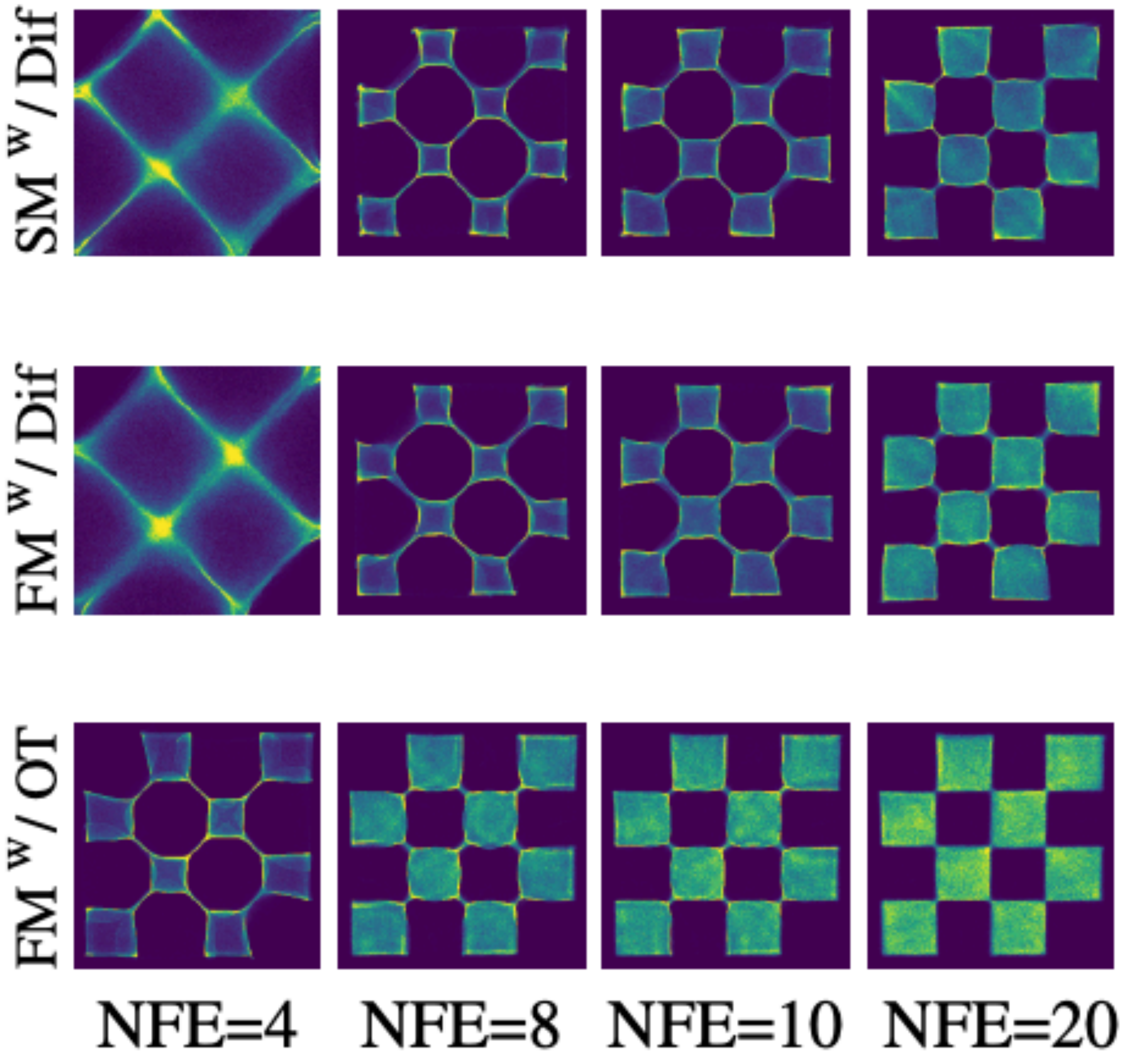
Lipman et al., *Flow Matching for Generative Modeling.* (ICLR 2023)

Conditional Optimal Transport Probability Path



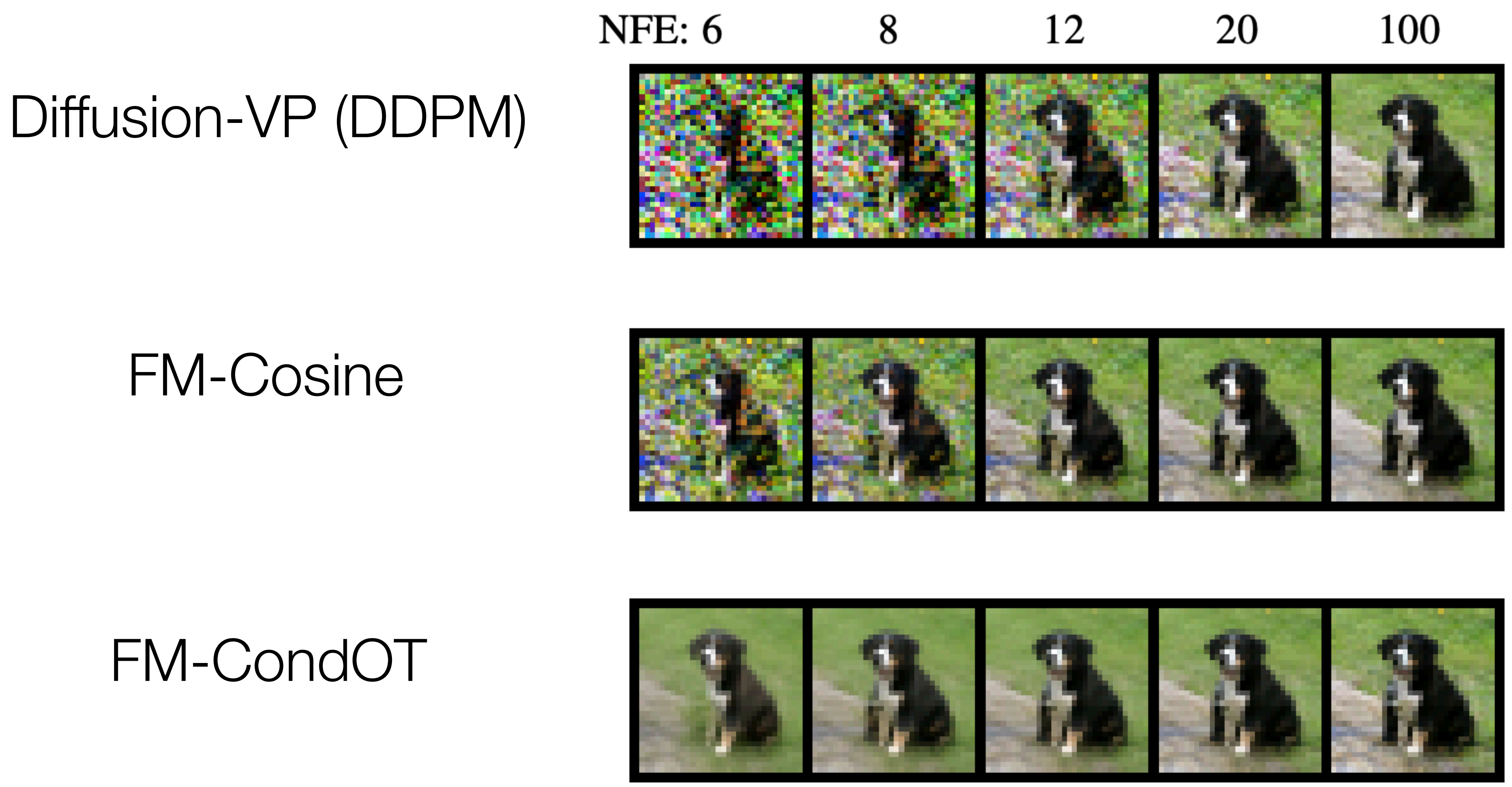
Lipman et al., *Flow Matching for Generative Modeling.* (ICLR 2023)

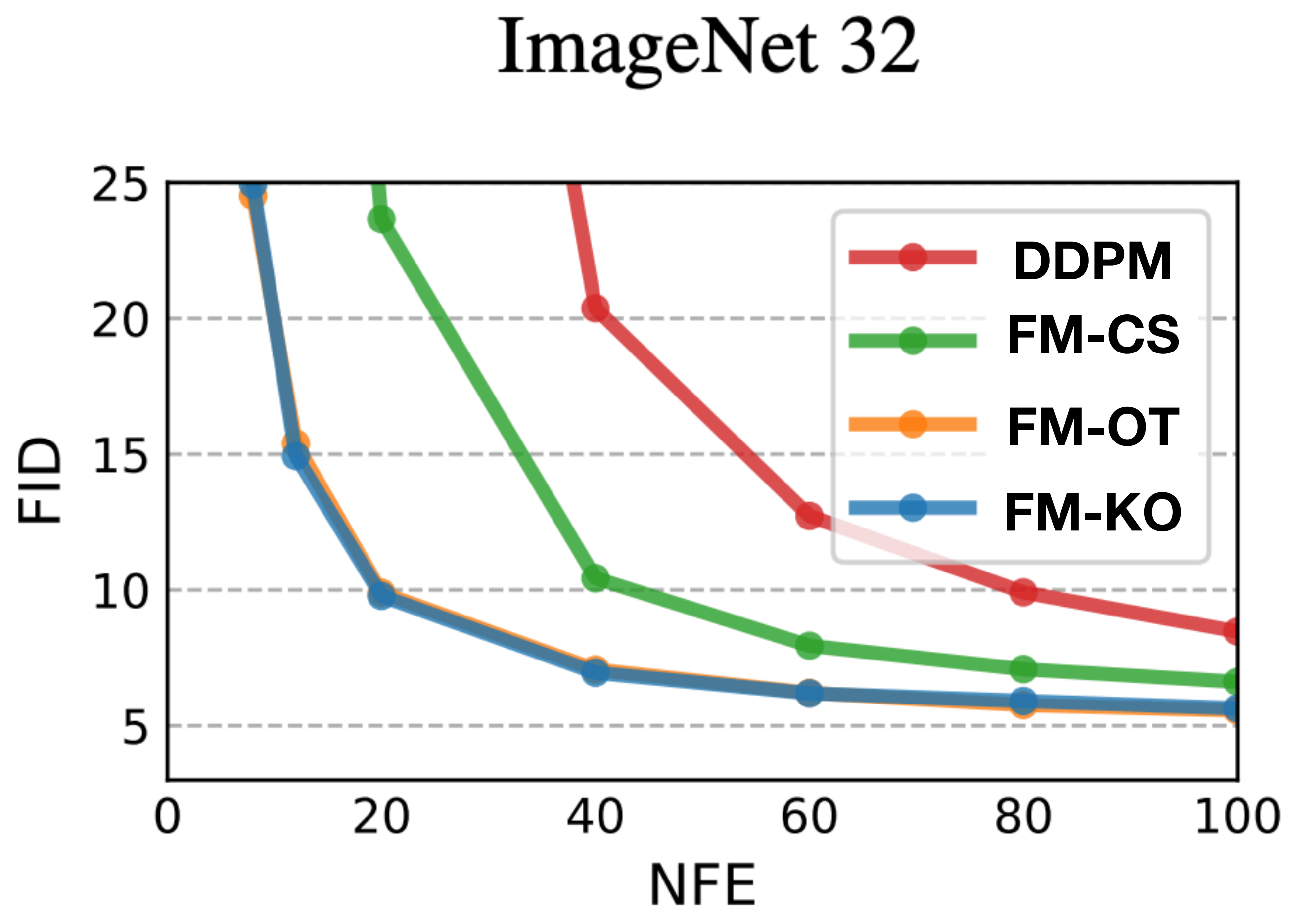
Conditional Optimal Transport Probability Path



Lipman et al., *Flow Matching for Generative Modeling.* (ICLR 2023)

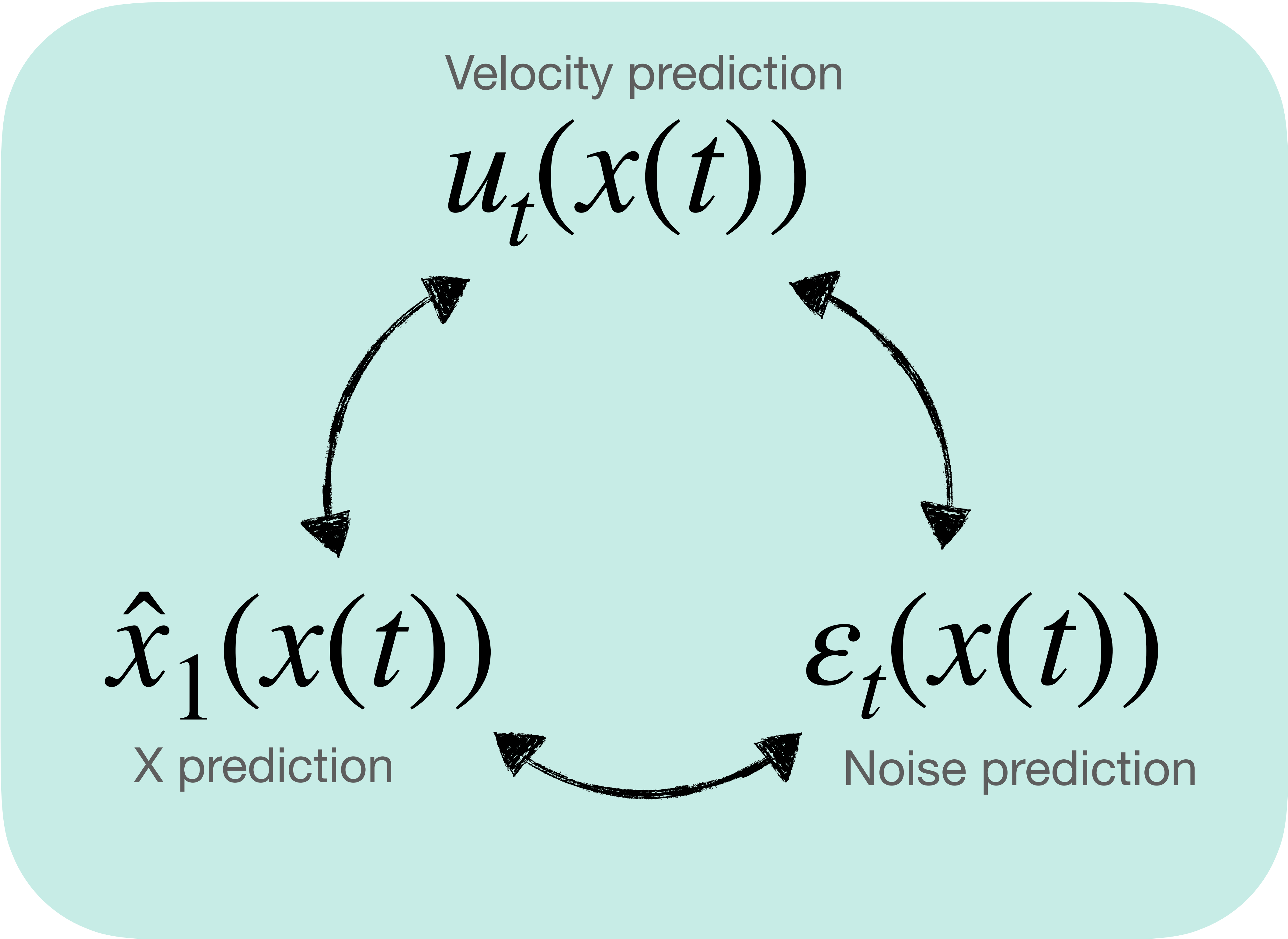
Conditional Optimal Transport Probability Path



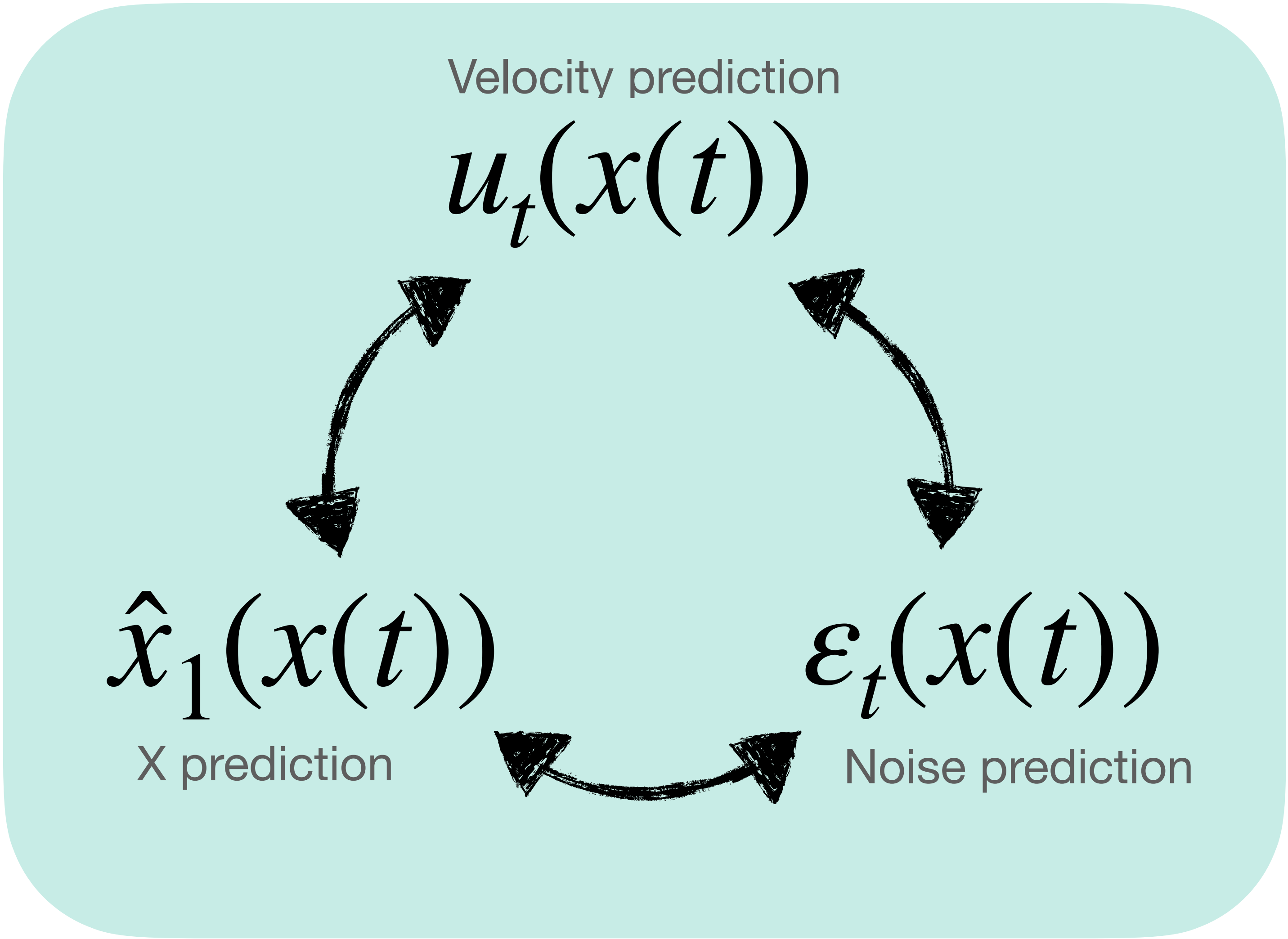


Lipman et al., *Flow Matching for Generative Modeling.* (ICLR 2023)

On flow-matching, denoisers, noise and score prediction



On flow-matching, denoisers, noise and score prediction



$$x_t = \psi_t(x_0 \mid x_1) = \sigma_t x_0 + \alpha_t x_1$$

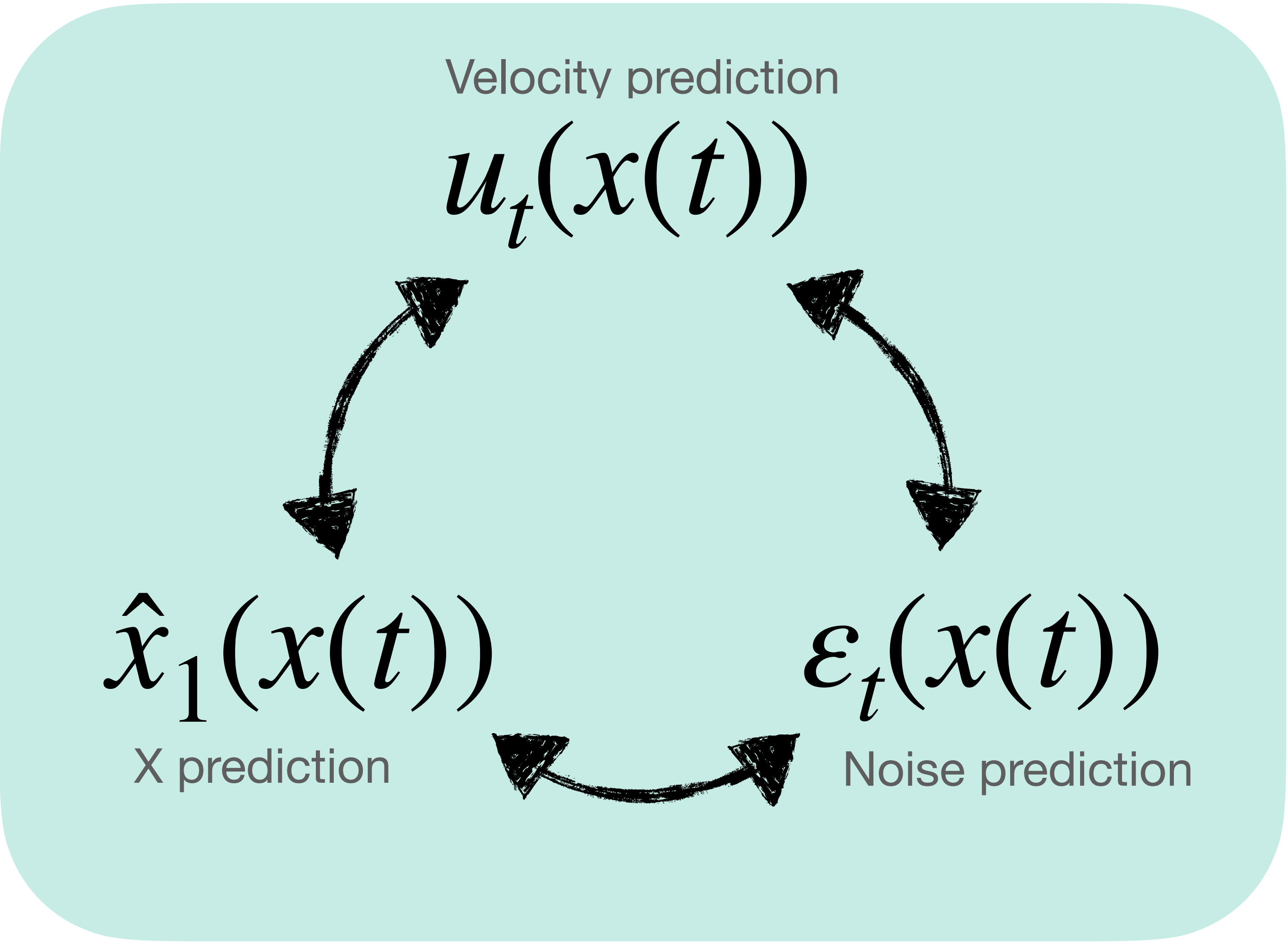
$$u_t(x \mid x_1) = \frac{\dot{\sigma}_t}{\sigma_t} x - \left(\frac{\alpha_t \dot{\sigma}_t}{\sigma_t} - \dot{\alpha}_t\right) x_1$$

$$u_t(x \mid x_0) = \frac{\dot{\alpha}_t}{\alpha_t} x - \left(\frac{\sigma_t \dot{\alpha}_t}{\alpha_t} - \dot{\sigma}_t\right) x_0$$

$$u_t(x \mid x_0, x_1) = \dot{\sigma}_t x_0 + \dot{\alpha}_t x_1$$

Tong et al., *Improving and Generalizing Flow-Based Generative Models with Minibatch Optimal Transport.* (TMLR)

On flow-matching, denoisers, noise and score prediction



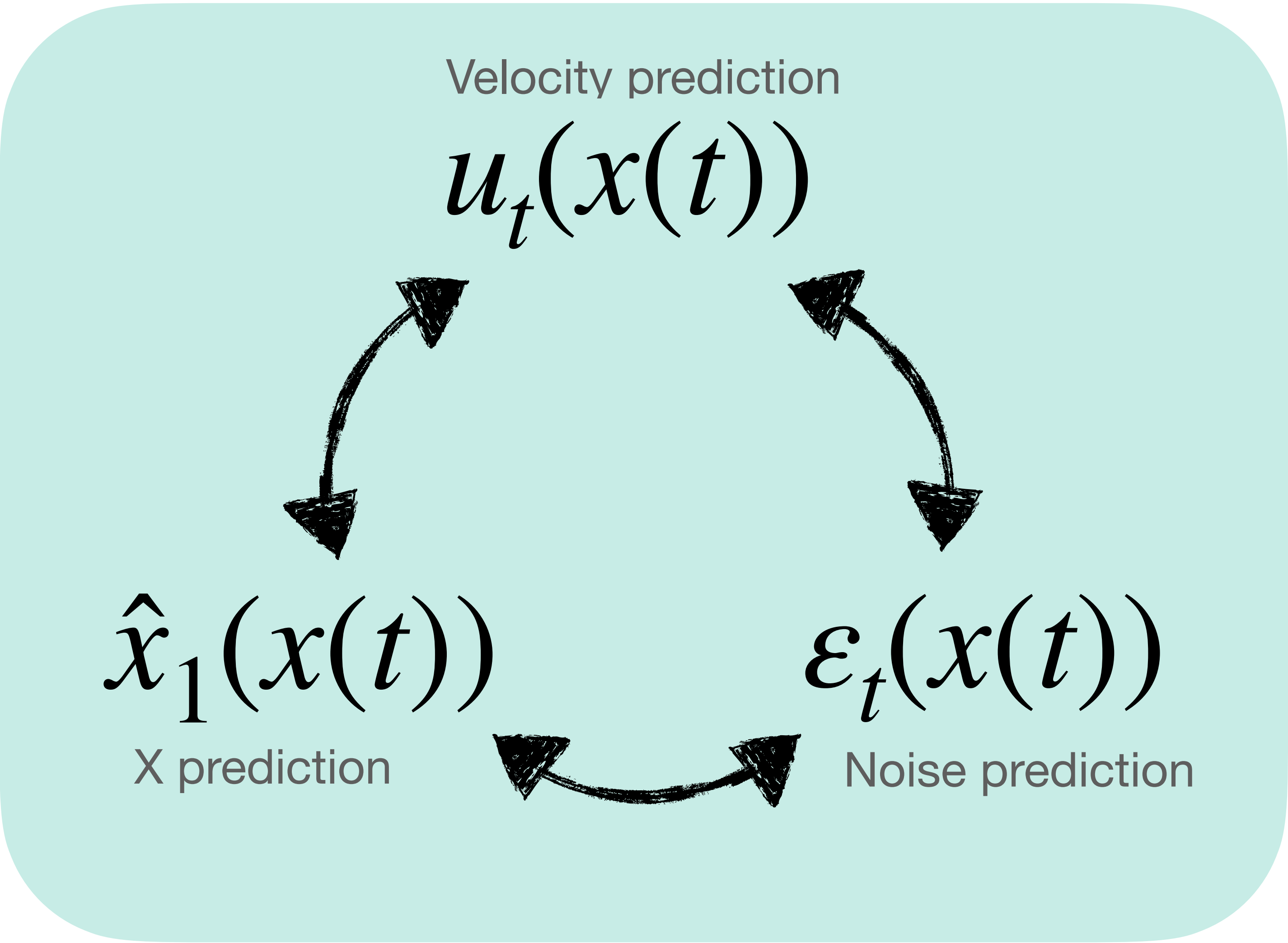
$$x_t = \psi_t(x_0 \mid x_1) = \sigma_t x_0 + \alpha_t x_1$$

$$u_t(x \mid x_1) = \frac{\dot{\sigma}_t}{\sigma_t} x - \left(\frac{\alpha_t \dot{\sigma}_t}{\sigma_t} - \dot{\alpha}_t\right) x_1$$

$$u_t(x \mid x_0) = \frac{\dot{\alpha}_t}{\alpha_t} x - \left(\frac{\sigma_t \dot{\alpha}_t}{\alpha_t} - \dot{\sigma}_t\right) x_0$$

$$u_t(x) = \int u_t(x \mid x_1) \frac{p_t(x \mid x_1) q(x_1)}{p_t(x)} dx_1$$

On flow-matching, denoisers, noise and score prediction



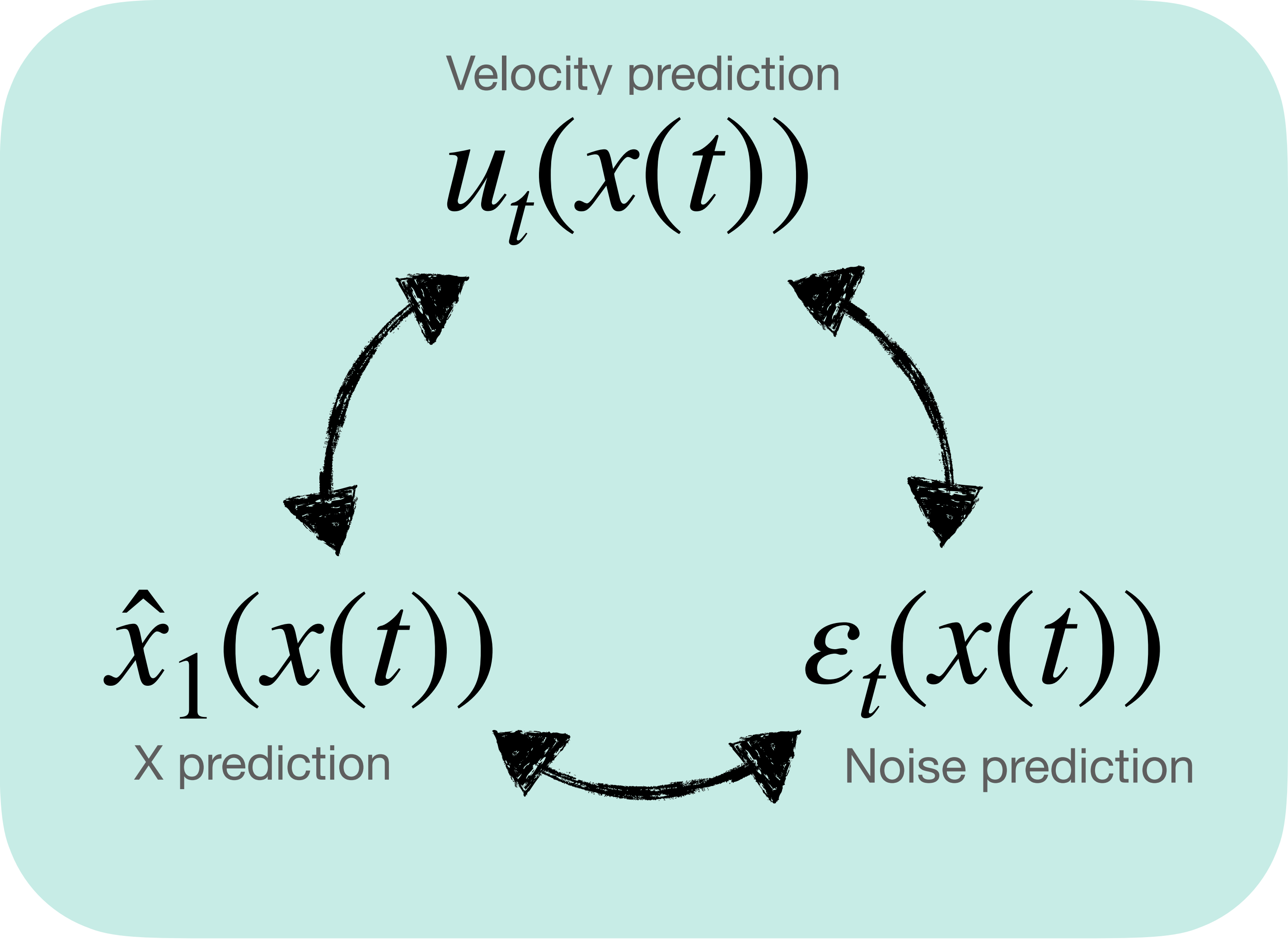
$$x_t = \psi_t(x_0 \,|\, x_1) = \sigma_t x_0 + \alpha_t x_1$$

$$u_t(x \,|\, x_1) = \frac{\dot{\sigma}_t}{\sigma_t} x - \left(\frac{\alpha_t \dot{\sigma}_t}{\sigma_t} - \dot{\alpha}_t\right) x_1$$

$$u_t(x \,|\, x_0) = \frac{\dot{\alpha}_t}{\alpha_t} x - \left(\frac{\sigma_t \dot{\alpha}_t}{\alpha_t} - \dot{\sigma}_t\right) x_0$$

$$u_t(x) = \int u_t(x \,|\, x_1) p_t(x_1 \,|\, x) dx_1$$

On flow-matching, denoisers, noise and score prediction



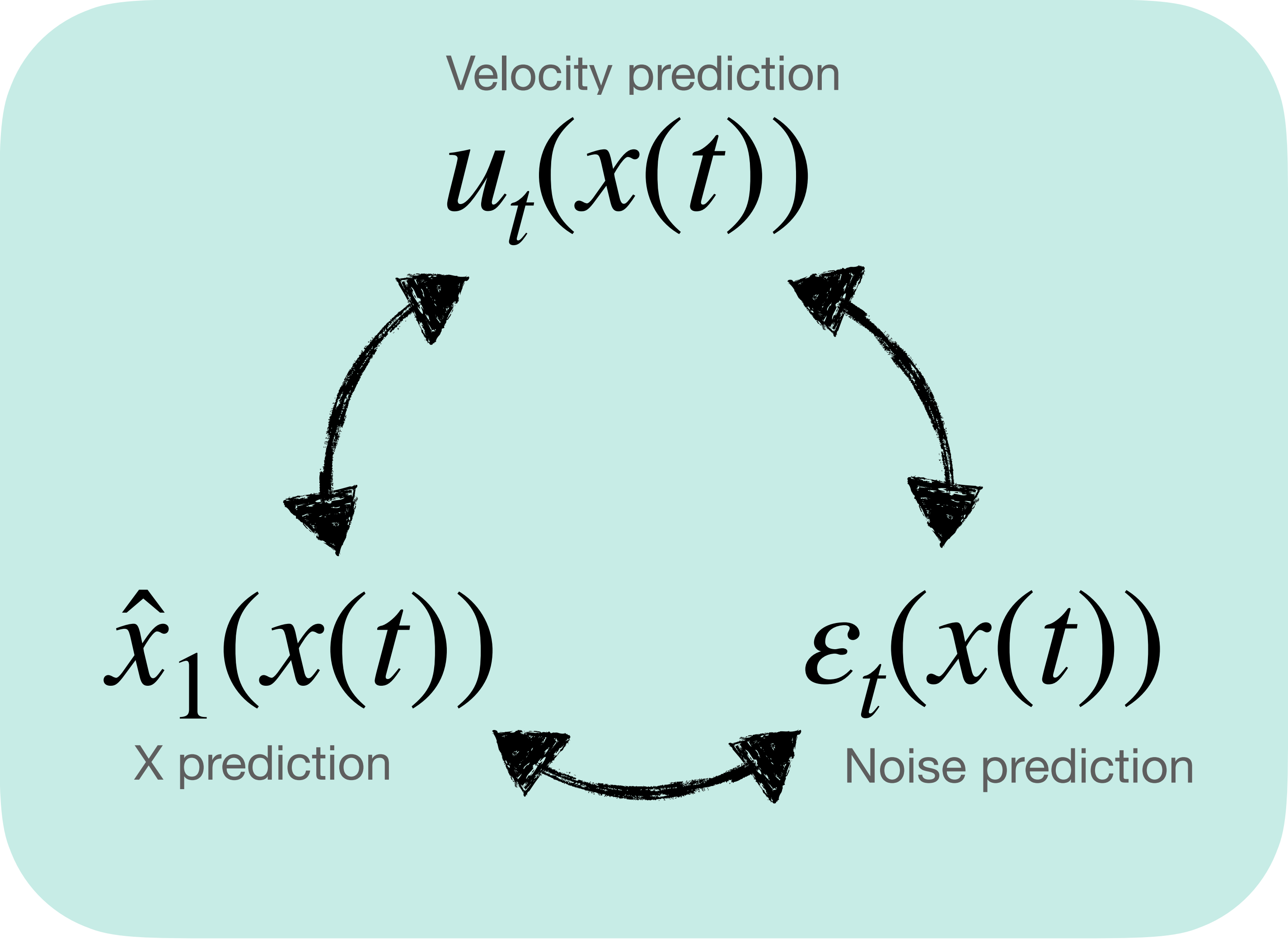
$$x_t = \psi_t(x_0 | x_1) = \sigma_t x_0 + \alpha_t x_1$$

$$u_t(x | x_1) = \frac{\dot{\sigma}_t}{\sigma_t} x - \left(\frac{\alpha_t \dot{\sigma}_t}{\sigma_t} - \dot{\alpha}_t\right) x_1$$

$$u_t(x | x_0) = \frac{\dot{\alpha}_t}{\alpha_t} x - \left(\frac{\sigma_t \dot{\alpha}_t}{\alpha_t} - \dot{\sigma}_t\right) x_0$$

$$u_t(x) = \int u_t(x | x_1) p_t(x_1 | x) dx_1 = \int u_t(x | x_0) p_t(x_0 | x) dx_0$$

On flow-matching, denoisers, noise and score prediction



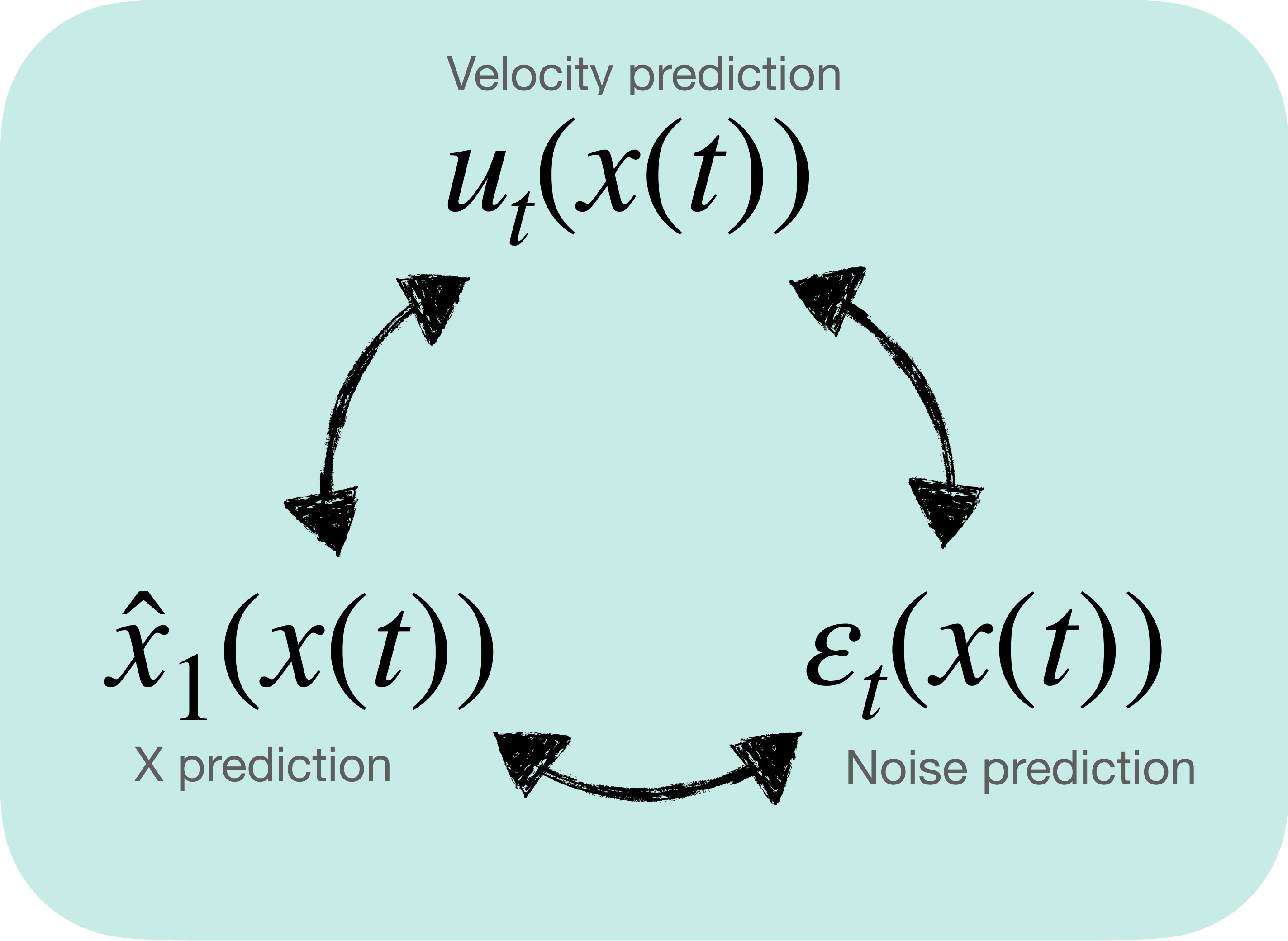
$$x_t = \psi_t(x_0 \,|\, x_1) = \sigma_t x_0 + \alpha_t x_1$$

$$u_t(x \,|\, x_1) = \frac{\dot{\sigma}_t}{\sigma_t} x - \left(\frac{\alpha_t \dot{\sigma}_t}{\sigma_t} - \dot{\alpha}_t\right) x_1$$

$$u_t(x \,|\, x_0) = \frac{\dot{\alpha}_t}{\alpha_t} x - \left(\frac{\sigma_t \dot{\alpha}_t}{\alpha_t} - \dot{\sigma}_t\right) x_0$$

$$u_t(x) = \frac{\dot{\sigma}_t}{\sigma_t} x - \left(\frac{\alpha_t \dot{\sigma}_t}{\sigma_t} - \dot{\alpha}_t\right) \hat{x}_1(x, t) = \frac{\dot{\alpha}_t}{\alpha_t} x - \left(\frac{\sigma_t \dot{\alpha}_t}{\alpha_t} - \dot{\sigma}_t\right) \hat{\varepsilon}(x, t)$$

On flow-matching, denoisers, noise and score prediction



$$x_t = \psi_t(x_0 \,|\, x_1) = \sigma_t x_0 + \alpha_t x_1$$

$$u_t(x \,|\, x_0) = \frac{\dot{\alpha}_t}{\alpha_t} x - \left(\frac{\sigma_t \dot{\alpha}_t}{\alpha_t} - \dot{\sigma}_t\right) x_0$$

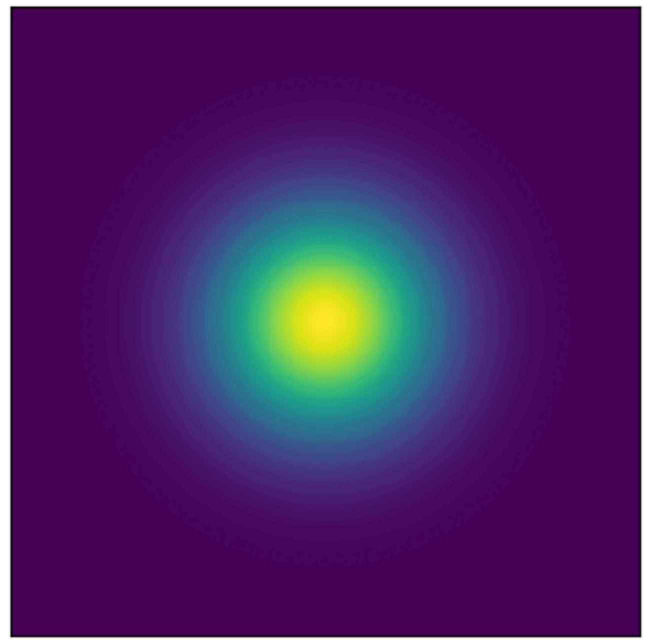
For the Gaussian case, where $x_0 \sim \mathcal{N}(0, I)$

$$u_t(x) = \frac{\dot{\alpha}_t}{\alpha_t} x - \left(\frac{\sigma_t \dot{\alpha}_t}{\alpha_t} - \dot{\sigma}_t\right) \hat{\varepsilon}(x, t) = \frac{\dot{\alpha}_t}{\alpha_t} x - \left(\frac{\sigma_t \dot{\alpha}_t}{\alpha_t} + \dot{\sigma}_t\right) \sigma_t \nabla \log p_t(x)$$

Joint Flow Matching

Marginal path

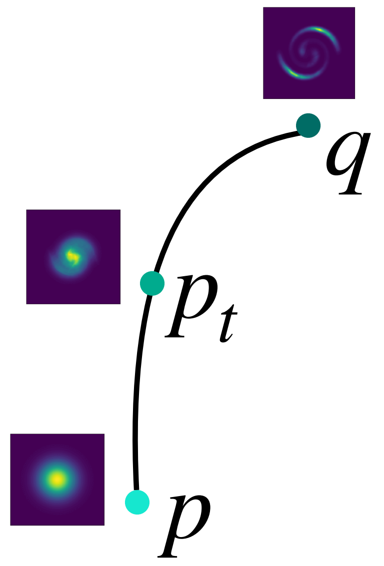
$$p_t(x) = \int p_t(x \mid x_1) q(x_1) dx_1$$



Boundary conditions:

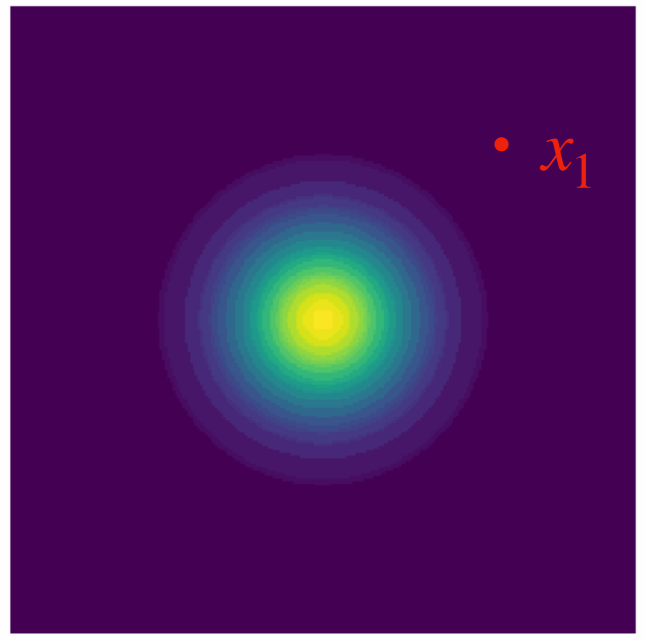
$$p_0 = p$$

$$p_1 = q$$



Conditional path

$$p_t(x \mid x_1)$$



$$p_0(\cdot \mid x_1) = p$$

$$p_1(\cdot \mid x_1) = \delta_{x_1}$$

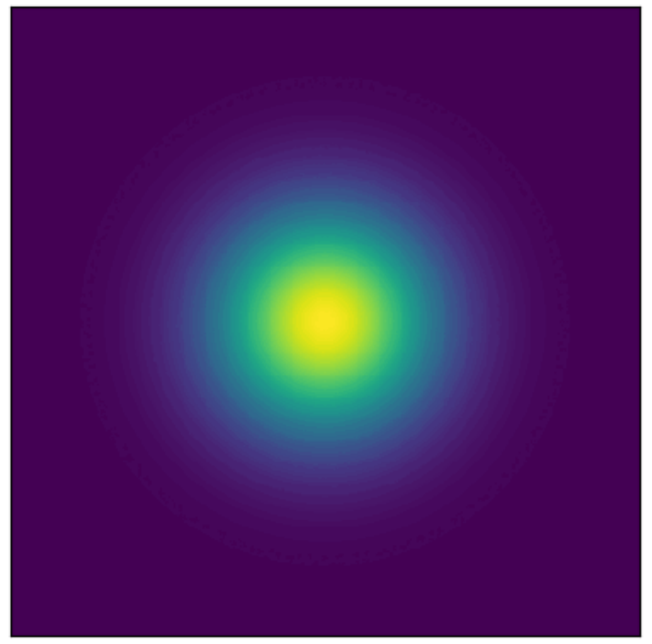
Pooladian*, **Ben-Hamu*,** Enrich* et al., *Multisample Flow Matching: Straightening Flows with Minibatch Couplings.* (ICML 2023)

Tong et al., *Improving and Generalizing Flow-Based Generative Models with Minibatch Optimal Transport.* (TMLR)

Joint Flow Matching

Marginal path

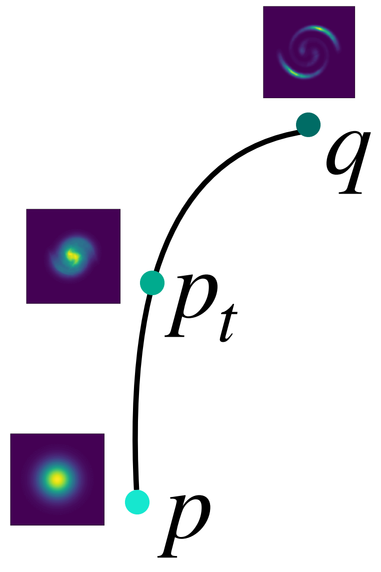
$$p_t(x) = \int p_t(x \mid x_1) q(x_1) dx_1$$



Boundary conditions:

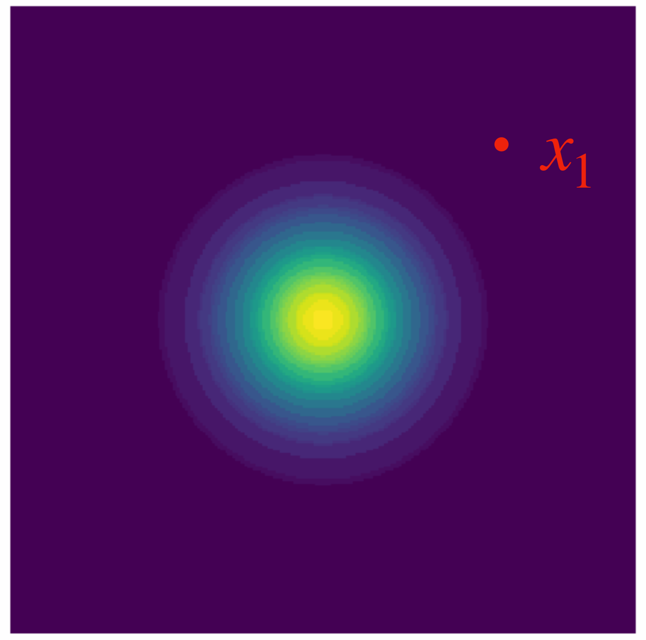
$$p_0 = p$$

$$p_1 = q$$



Conditional path

$$p_t(x \mid x_1)$$



$$\cancel{p_0(\cdot \mid x_1) = p} \quad \int \underbrace{p_0(x_0 \mid x_1) q(x_1)}_{q(x_0, x_1)} dx_1 = p(x_0)$$

$$p_1(\cdot \mid x_1) = \delta_{x_1}$$

Pooladian*, **Ben-Hamu*,** Enrich* et al., *Multisample Flow Matching: Straightening Flows with Minibatch Couplings.* (ICML 2023)

Tong et al., *Improving and Generalizing Flow-Based Generative Models with Minibatch Optimal Transport.* (TMLR)

Joint Flow Matching

Arbitrary source-target coupling

Algorithm 3: Flow Matching training.

Input : dataset and noise joint q

Initialize v^θ

while *not converged* **do**

- $t \sim \mathcal{U}([0,1])$ ▷ sample time
- $\boxed{(x_0, x_1) \sim q(x_0, x_1)}$ ▷ sample noise-data
- $x_t = \Psi_t(x_0|x_1)$ ▷ conditional flow
- Gradient step with $\nabla_\theta \|v_t^\theta(x_t) - \dot{x}_t\|^2$

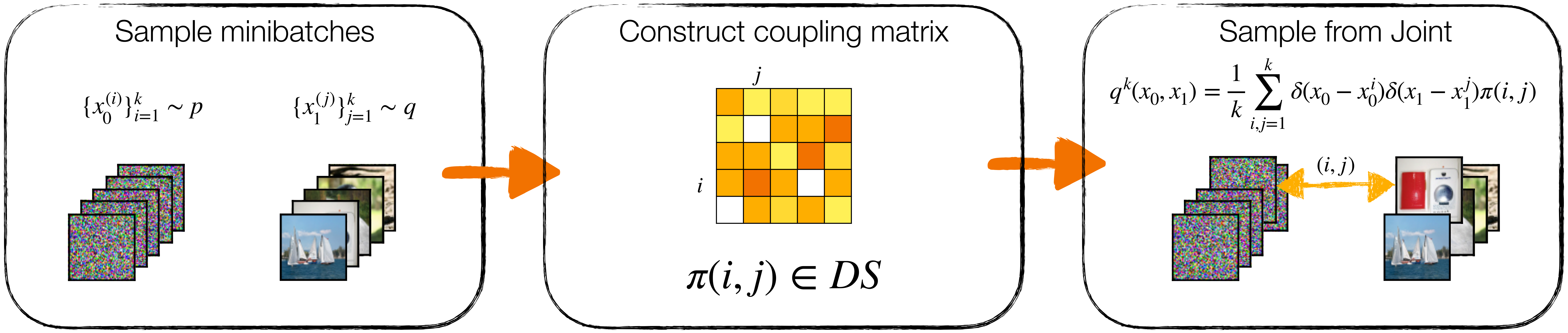
Output: v^θ

How can we construct $q(x_0, x_1)$?

Pooladian*, **Ben-Hamu*,** Enrich* et al., *Multisample Flow Matching: Straightening Flows with Minibatch Couplings.* (ICML 2023)

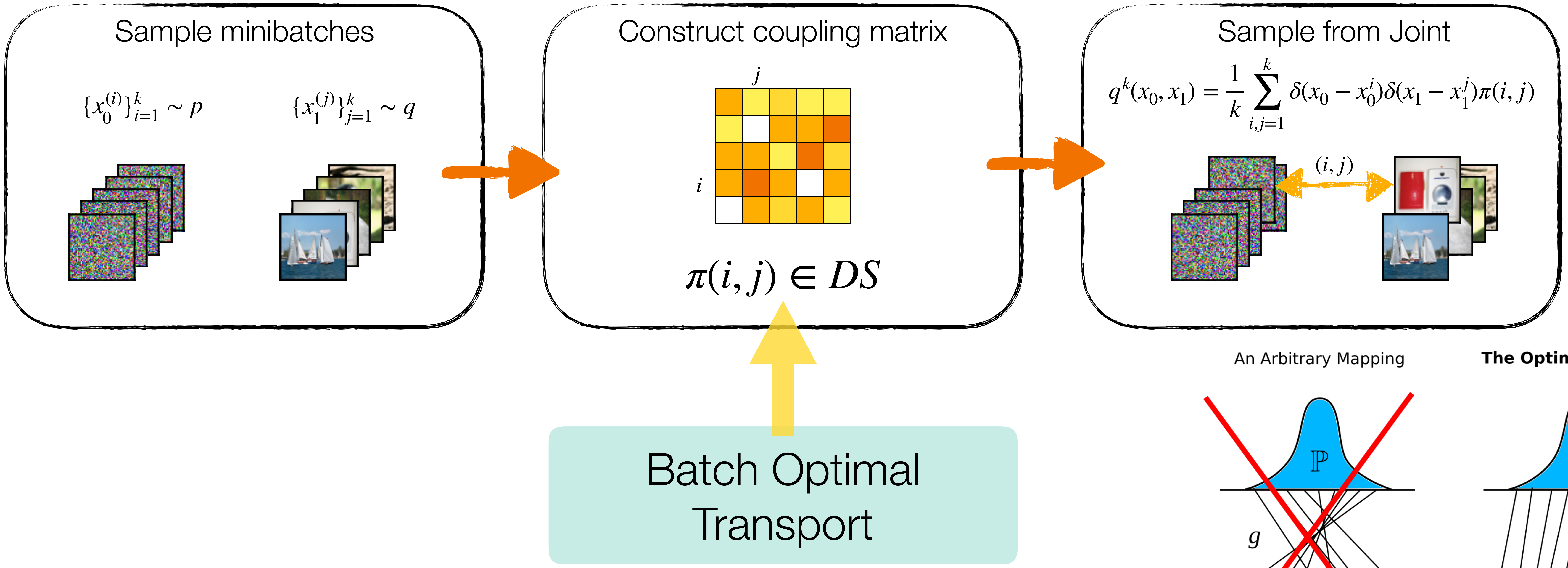
Tong et al., *Improving and Generalizing Flow-Based Generative Models with Minibatch Optimal Transport.* (TMLR)

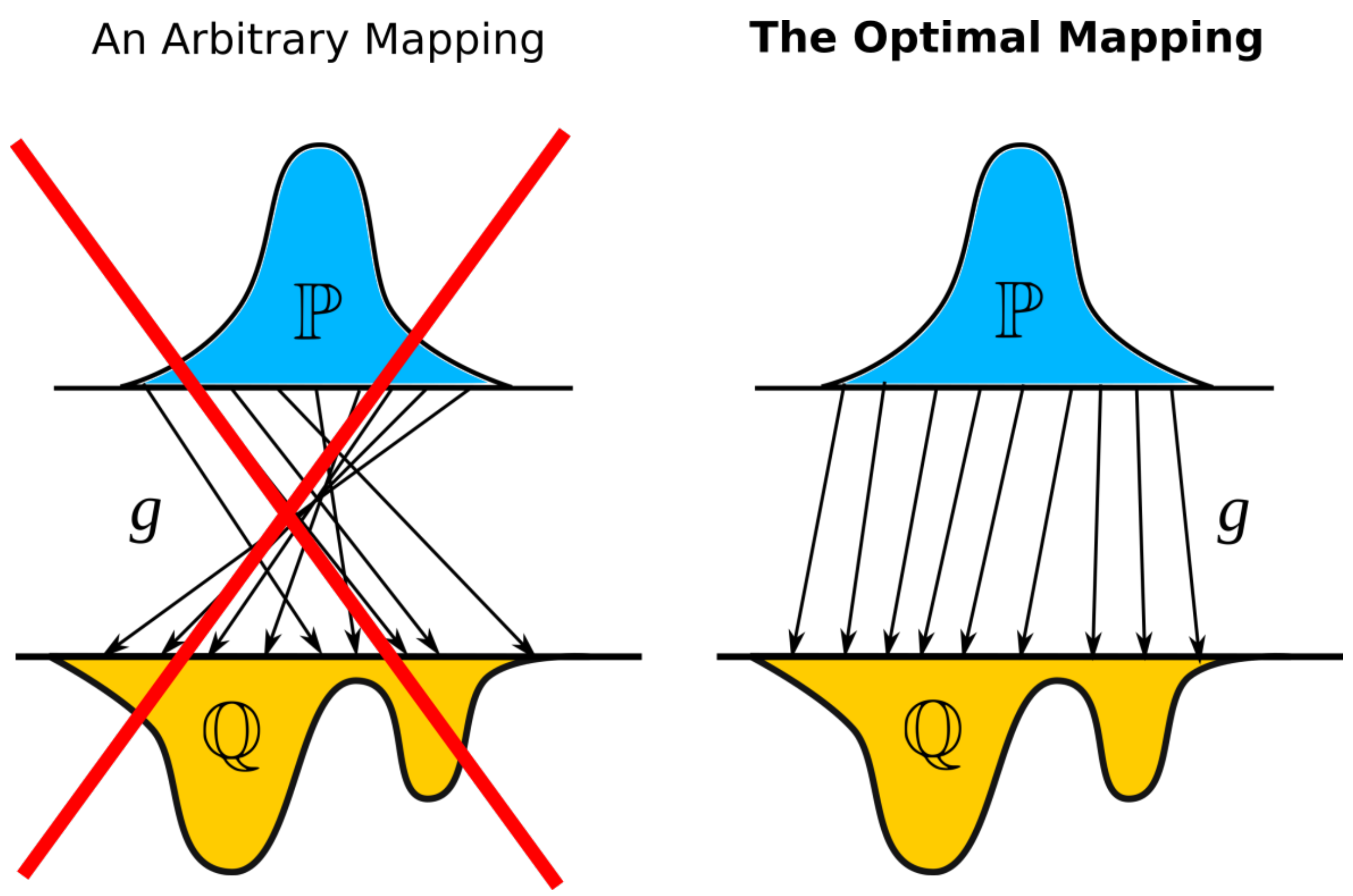
Multisample Flow Matching



Pooladian*, **Ben-Hamu*,** Enrich* et al., *Multisample Flow Matching: Straightening Flows with Minibatch Couplings.* (ICML 2023)

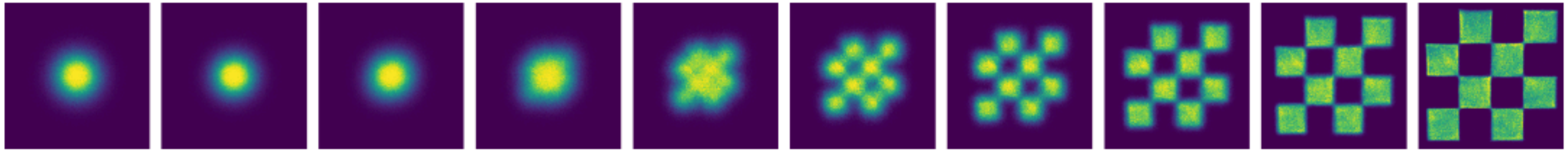
Multisample Flow Matching



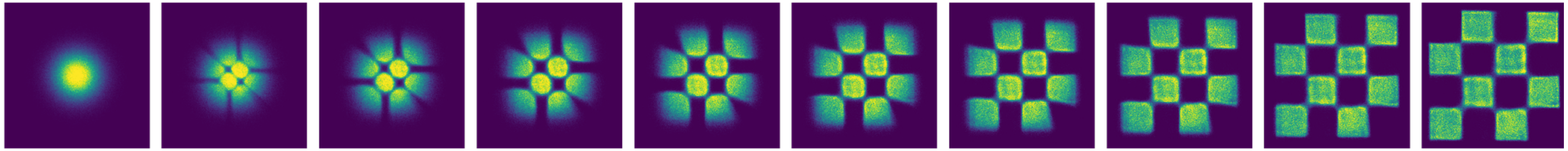


Pooladian*, **Ben-Hamu*,** Enrich* et al., *Multisample Flow Matching: Straightening Flows with Minibatch Couplings.* (ICML 2023)

Multisample Flow Matching



Flow Matching / Cond-OT

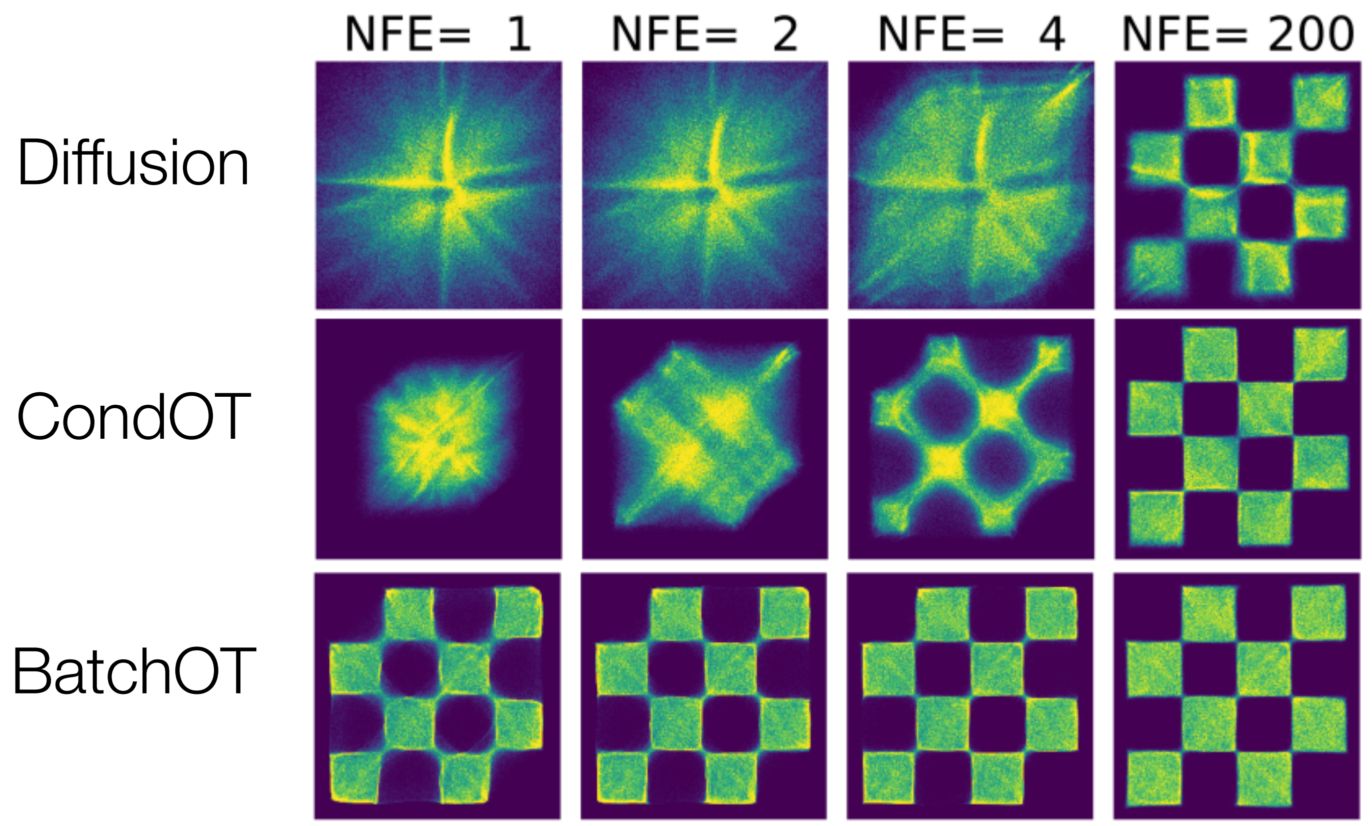


Flow Matching / Batch-OT

Batch Optimal Transport

Pooladian*, **Ben-Hamu*,** Enrich* et al., *Multisample Flow Matching: Straightening Flows with Minibatch Couplings.* (ICML 2023)

Multisample Flow Matching

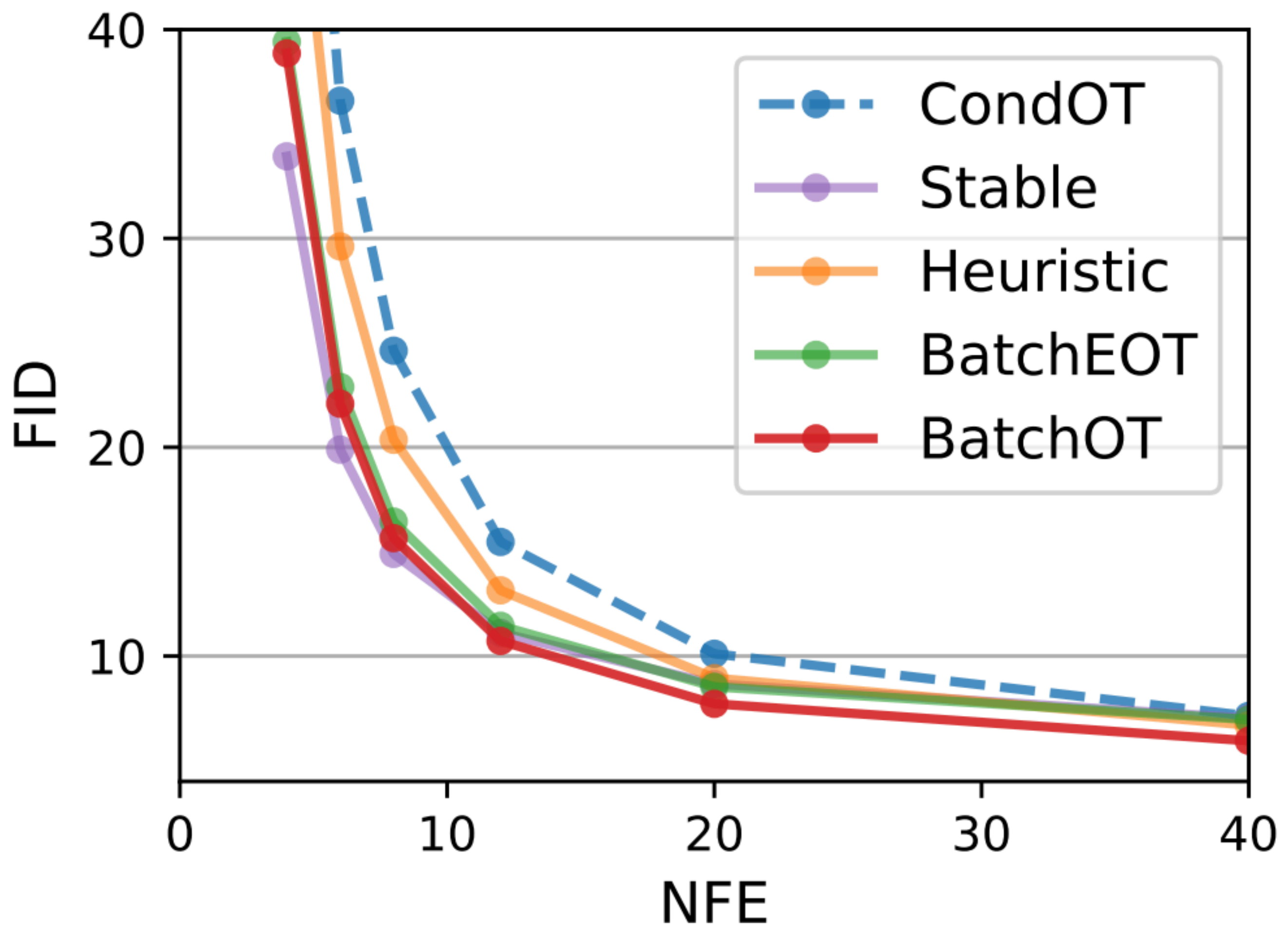


Batch Optimal Transport

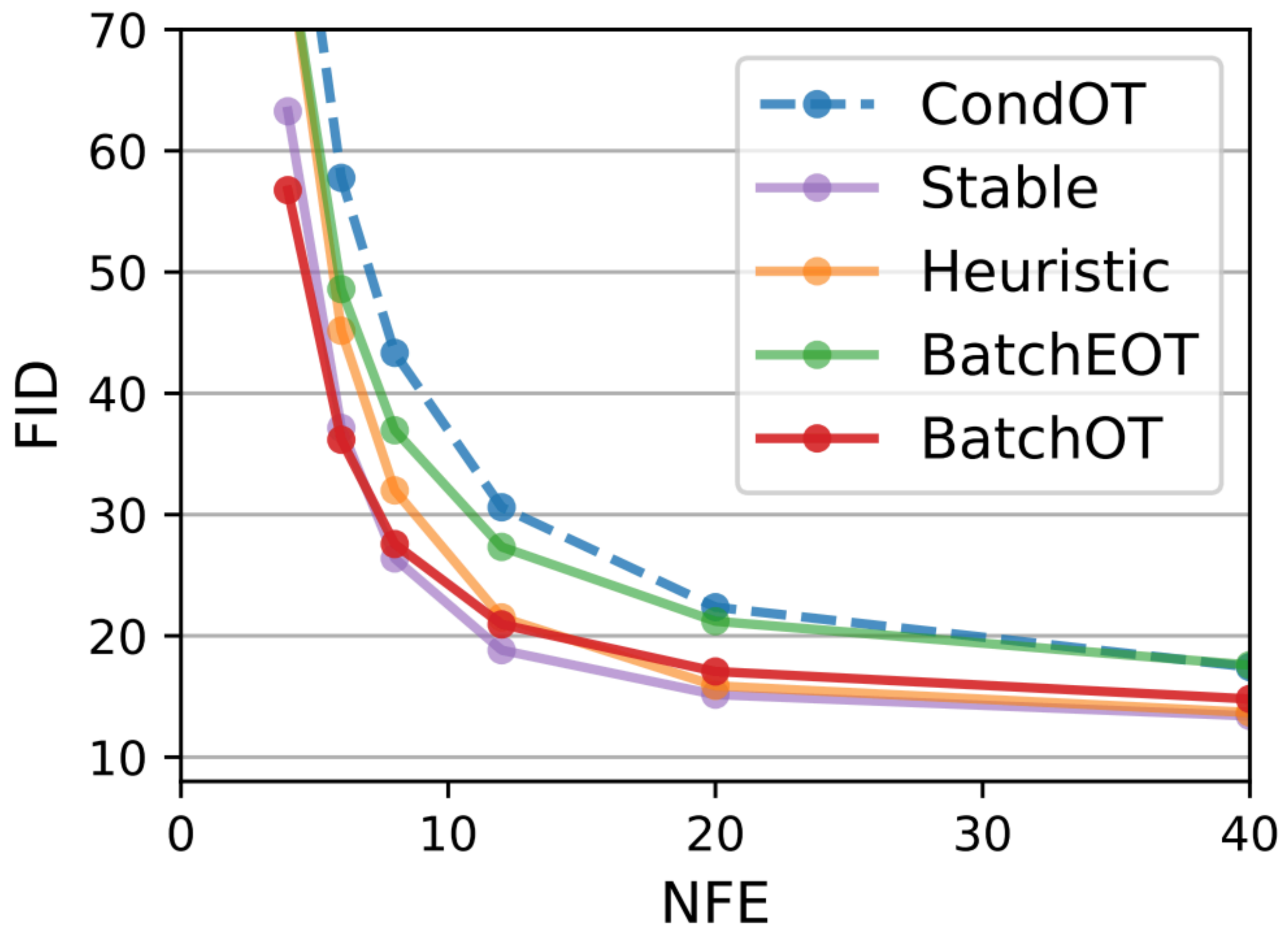
Pooladian*, **Ben-Hamu*,** Enrich* et al., *Multisample Flow Matching: Straightening Flows with Minibatch Couplings.* (ICML 2023)

Multisample Flow Matching

Reduce sampling time
by 30%-60%



ImageNet 32×32



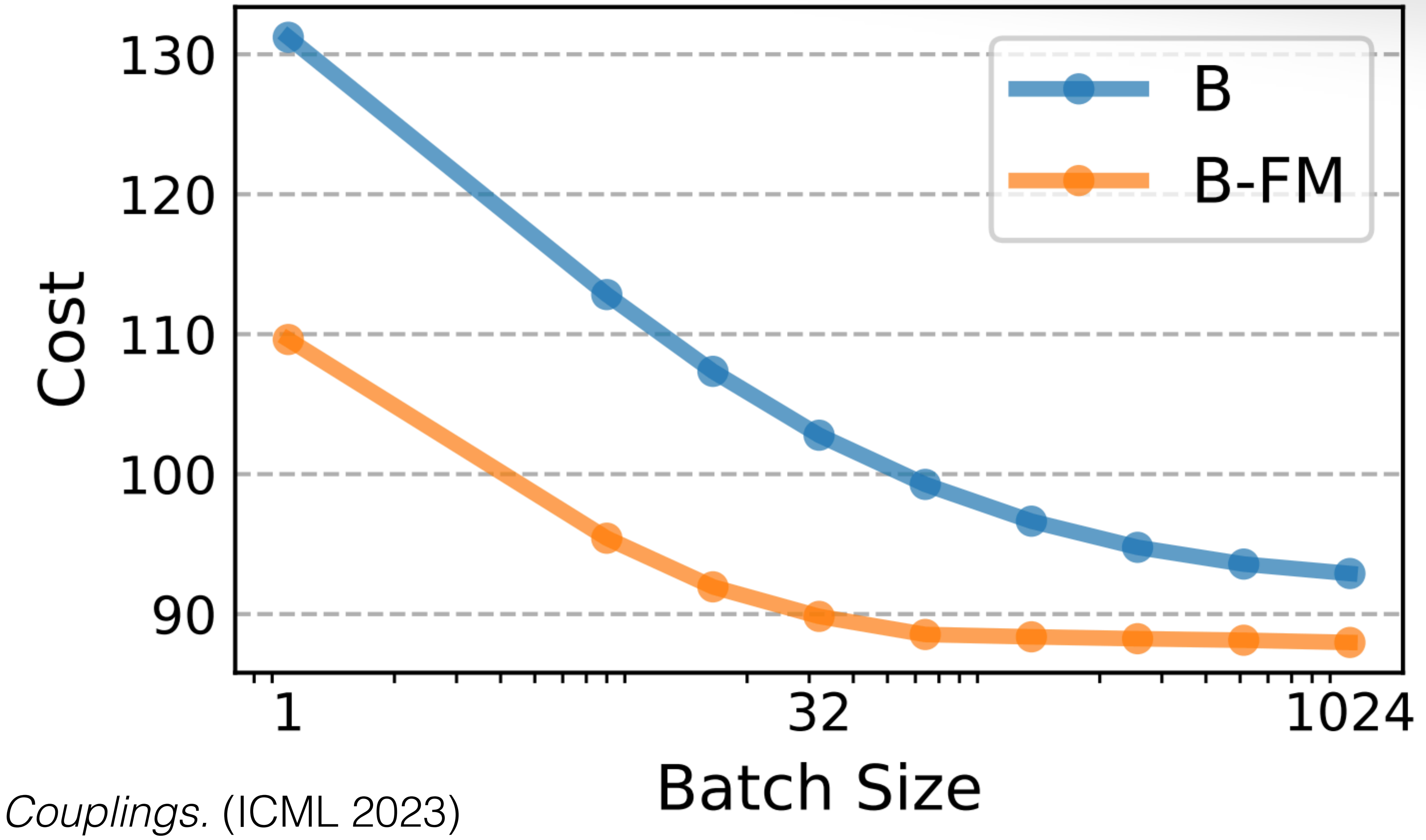
ImageNet 64×64

Pooladian*, **Ben-Hamu*,** Enrich* et al., *Multisample Flow Matching: Straightening Flows with Minibatch Couplings.* (ICML 2023)

Asymptotics of BatchOT

Theorem **[Pooladian*, Ben-Hamu*, Enrich* et al.].**

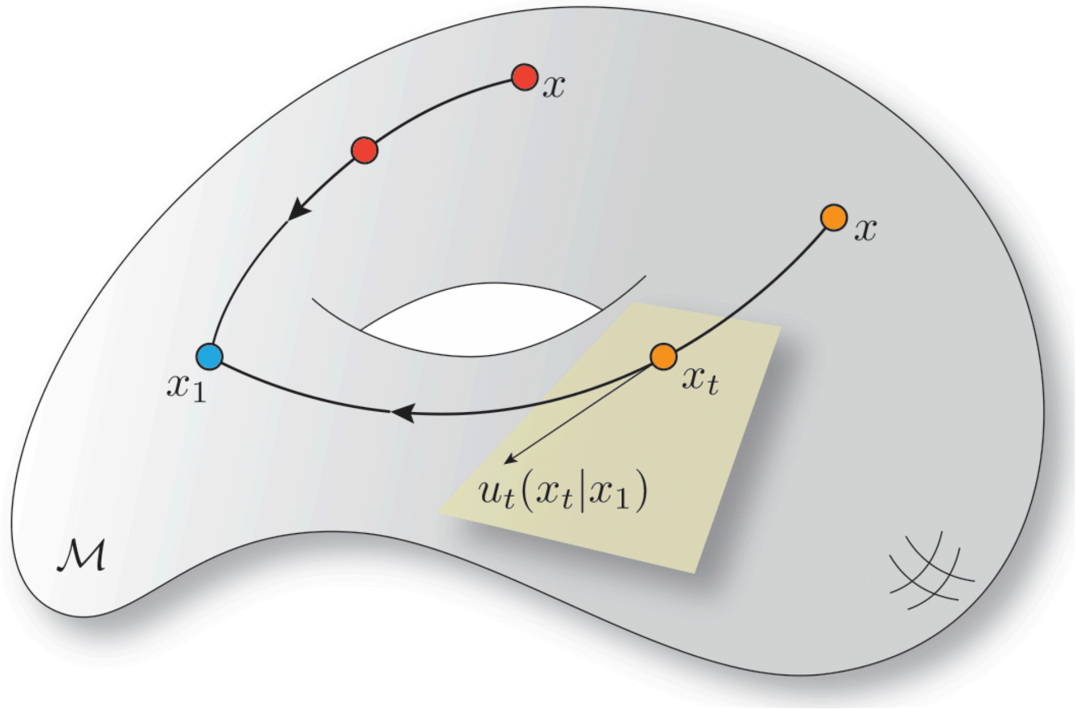
- The transport cost of the learned flow monotonically decreases with $\textbf{batch size} \uparrow$
- As $\textbf{batch size} \to \infty$ the transport cost of the learned flow converges to the optimal transport cost.



Pooladian*, **Ben-Hamu*,** Enrich* et al., *Multisample Flow Matching: Straightening Flows with Minibatch Couplings.* (ICML 2023)

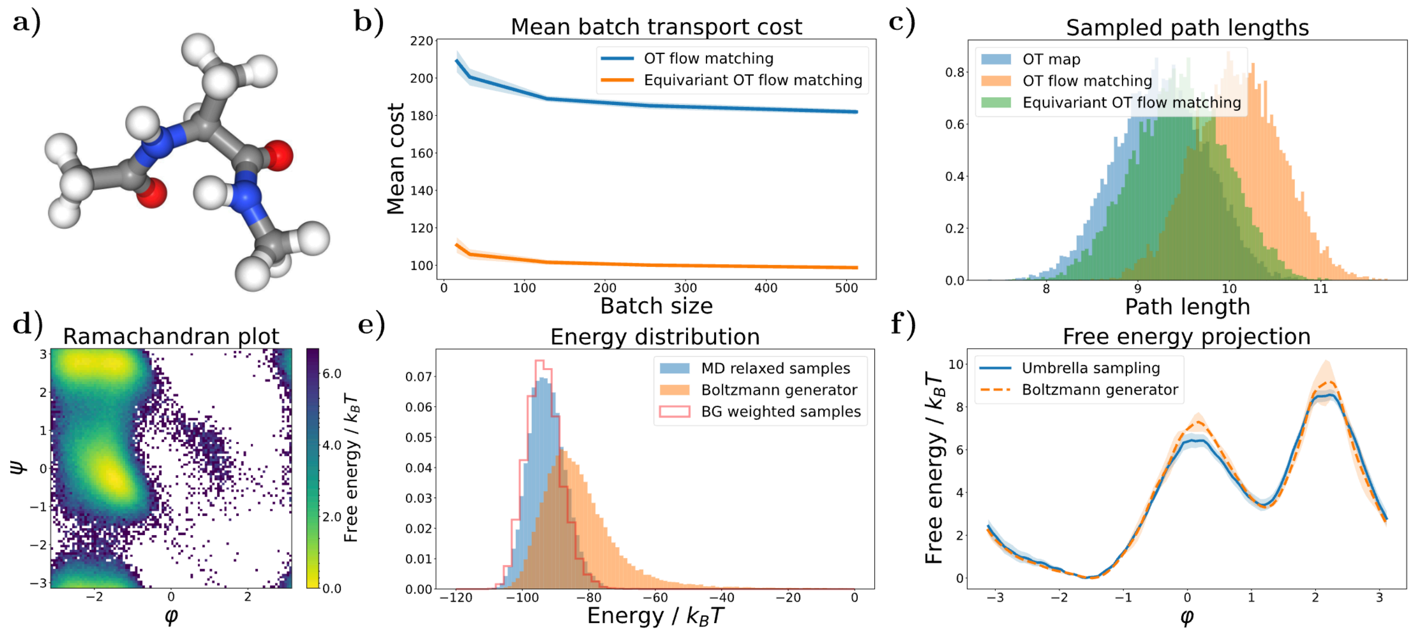
Interesting Applications

Flow Matching on general geometries, Chen & Lipman 2023

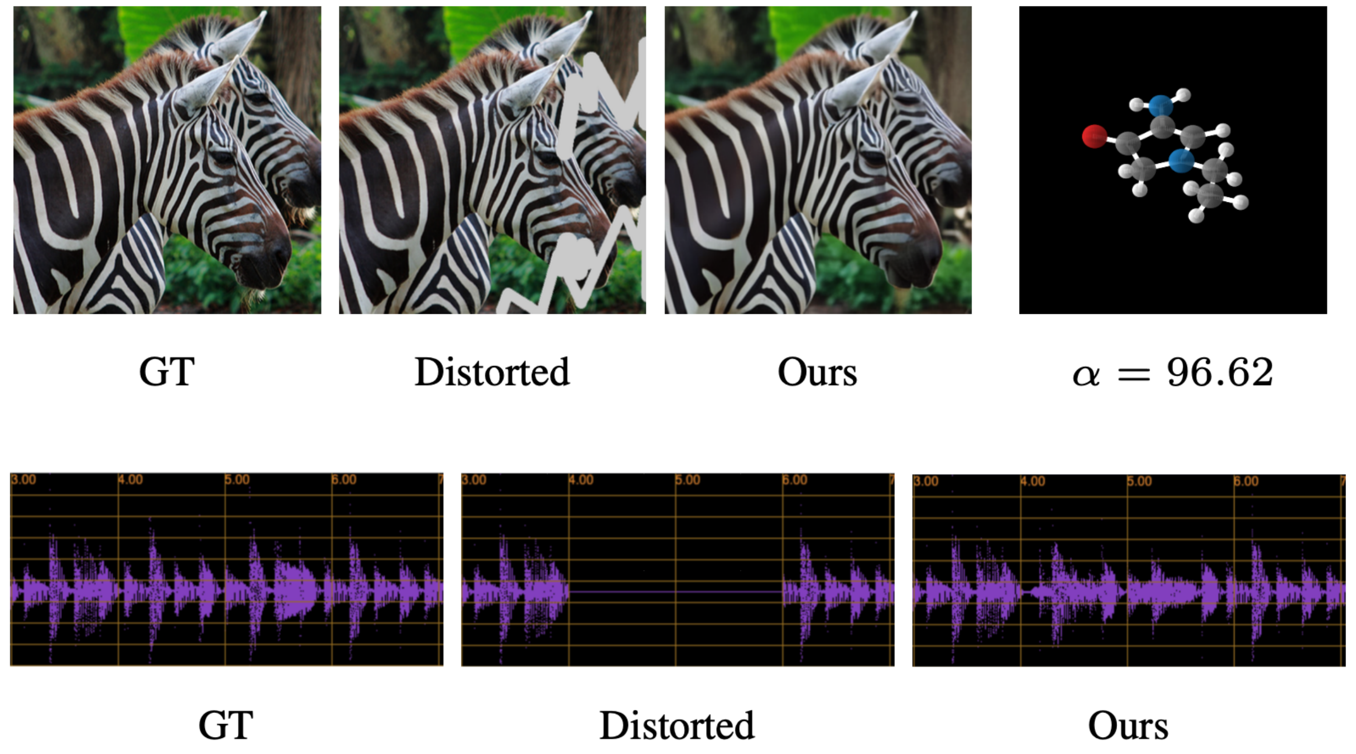


Text-2-Image, Esser et al. 2024

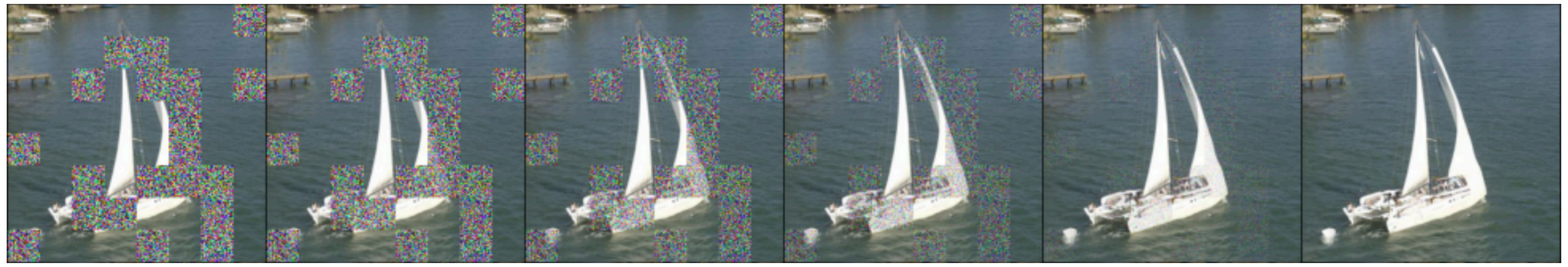
Equivariant Flow Matching, Klein et al. 2023



Controlled Generation, Ben-Hamu et al. 2024



Inverse Problems, Albergo et al. 2023



Take Home Messages

- Flows are powerful generative models when supervised adequately
    - Flow Matching is a **flexible** framework for training generative flows
- Improved sampling speed and stability
- Exciting generalizations and extensions

Image from Esser et al. 2024

Future Research

- Conditional generation
  - Experiments reveal that some data domains exhibit a gap in ability to learn conditional models.
  - What are the possible improvements? Architectures, data, etc.?

D-Flow, **Ben-Hamu** et al. 2024

Property	α	$\Delta\varepsilon$	ε_{HOMO}	ε_{LUMO}	μ	C_v
Units	Bohr^2	meV	meV	meV	D	$\frac{\text{cal}}{\text{mol}}$K
QM9*	0.10	64	39	36	0.043	0.040
EDM	2.76	655	356	584	1.111	1.101
EquiFM	2.41	591	337	530	1.106	1.033
GeoLDM	2.37	587	340	522	1.108	1.025
Ours	1.38	340	179	330	0.299	0.784

Conditional Models

- Is the success of Flow matching (and diffusion) due to…
  - Formulating the generative modeling problem as a reg
  - Breaking the problem into many small denoising proble
  - Abandoning log-likelihoods in high dimensions?

Thank You!

https://helibenhamu.github.io/

Thanks to my collaborators and mentors along the way:

Yaron Lipman, Haggai Maron, Ricky Chen, Omri Puny, Samuel Cohen, Itai Gat, Uriel Singer, Brian Karrer, Matan Atzmon, Maximillian Nickel, Aram Alexander Pooladian, Carles Domingo Enrich, Brandon Amos, Joey Bose, Matt Le, Aditya Grover, Nimrod Segol, Hadar Serviansky, Nadav Shamir.