Diffusion Model Journal Club

Flow Matching for Generative Modeling

Annalena Kofler, 13.11.2024

The secret of modeling is not being perfect.

Karl Lagerfeld

Stolen from Antoine Wehenkel’s presentation at the SBI workshop

Similar Work

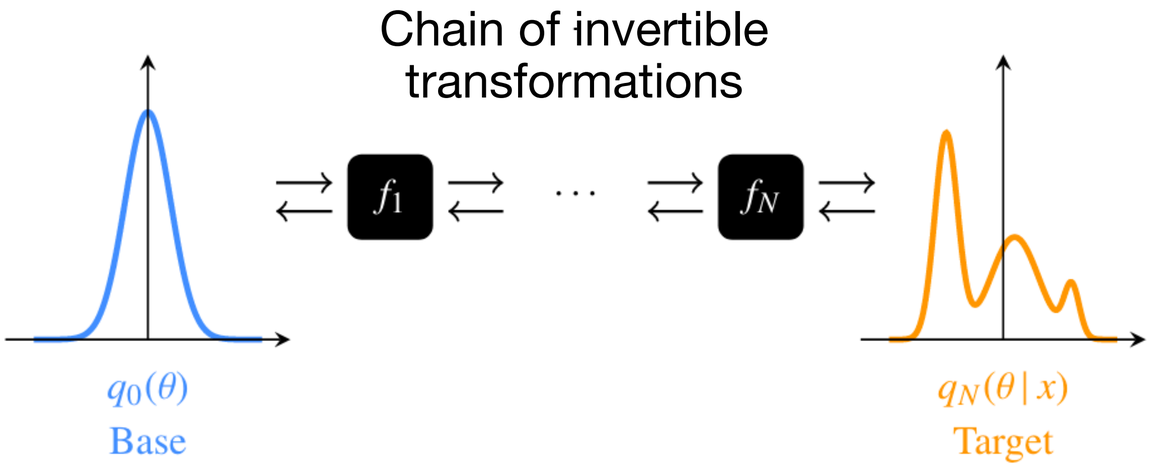
3 papers postulated similar concepts:

- "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow" (07.09.2022)
- "Building Normalizing Flows with Stochastic Interpolants" (30.09.2022)
- **"Flow Matching for Generative Modeling"** (06.10.2022)

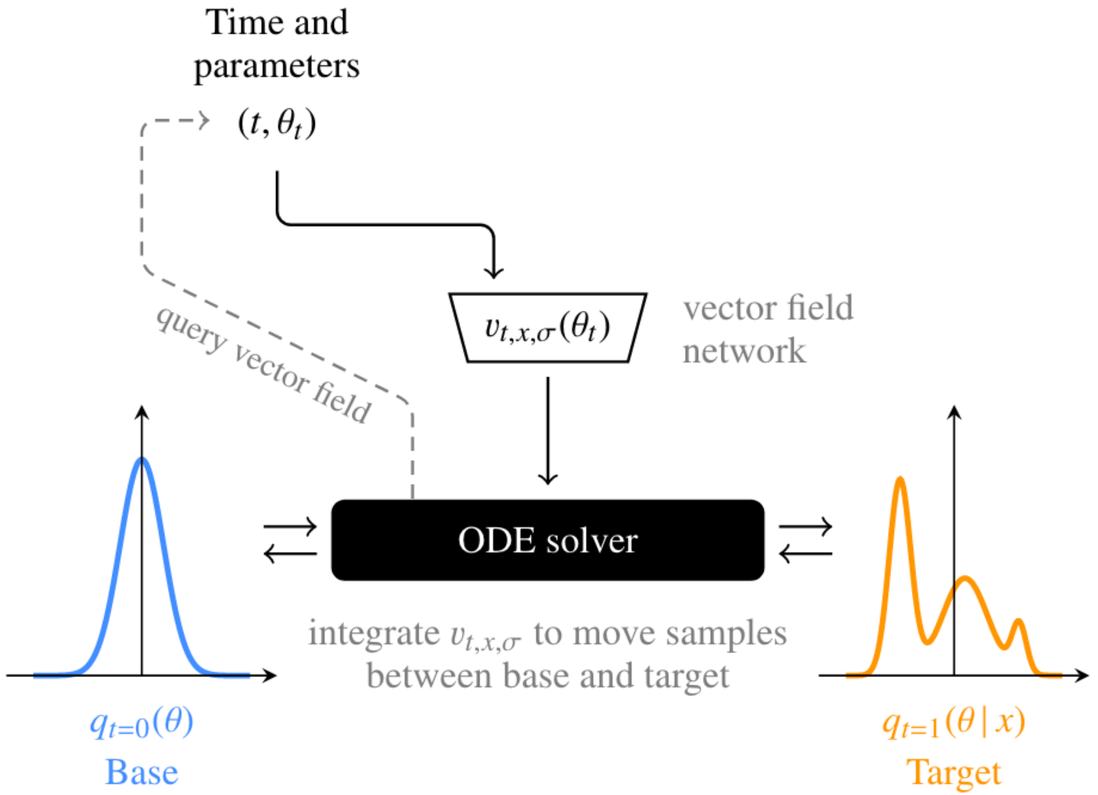
Overview

Background: Continuous Normalizing Flows

- What is the difference between a discrete and a continuous normalizing flow?



Discrete number of transformations



Continuous transformation

Background: Continuous Normalizing Flows

- Discrete change of variables

$$\mathbf{z}_1 = f(\mathbf{z}_0) \implies \log p(\mathbf{z}_1) = \log p(\mathbf{z}_0) - \log \left| \det \frac{\partial f}{\partial \mathbf{z}_0} \right|$$

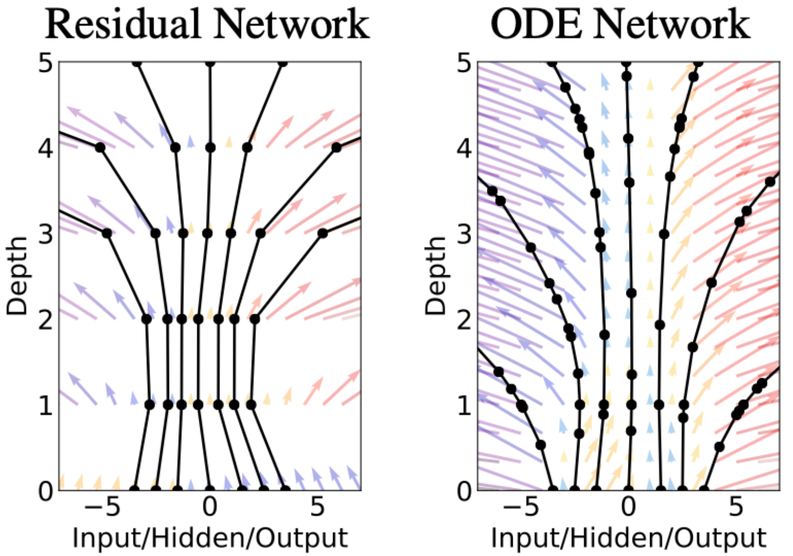
→ Loss $L = -\sum_{i=1}^{N} \log p(\mathbf{z}_1^{(i)})$

- Continuous change of variables

$$\frac{\partial \log p(\mathbf{z}(t))}{\partial t} = -\mathrm{tr}\left(\frac{df}{d\mathbf{z}(t)}\right)$$

→ Loss involves ODE integration → costly

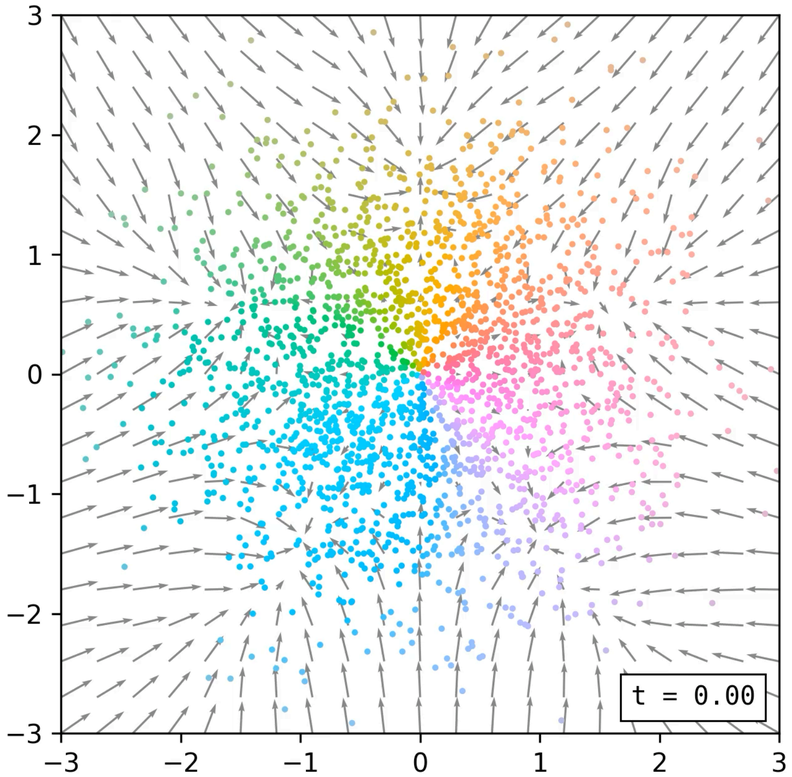
$$L(\mathbf{z}(t_1)) = L\left(\mathbf{z}(t_0) + \int_{t_0}^{t_1} f(\mathbf{z}(t), t, \theta) dt\right) = L\left(\mathrm{ODESolve}(\mathbf{z}(t_0), f, t_0, t_1, \theta)\right)$$



Chen et al., Neural Ordinary Differential Equations, NeurIPS 2018

Why is flow matching so popular?

- Because it allows to train a CNF **without ODE integration**!
- But: ODE integration required at inference ..



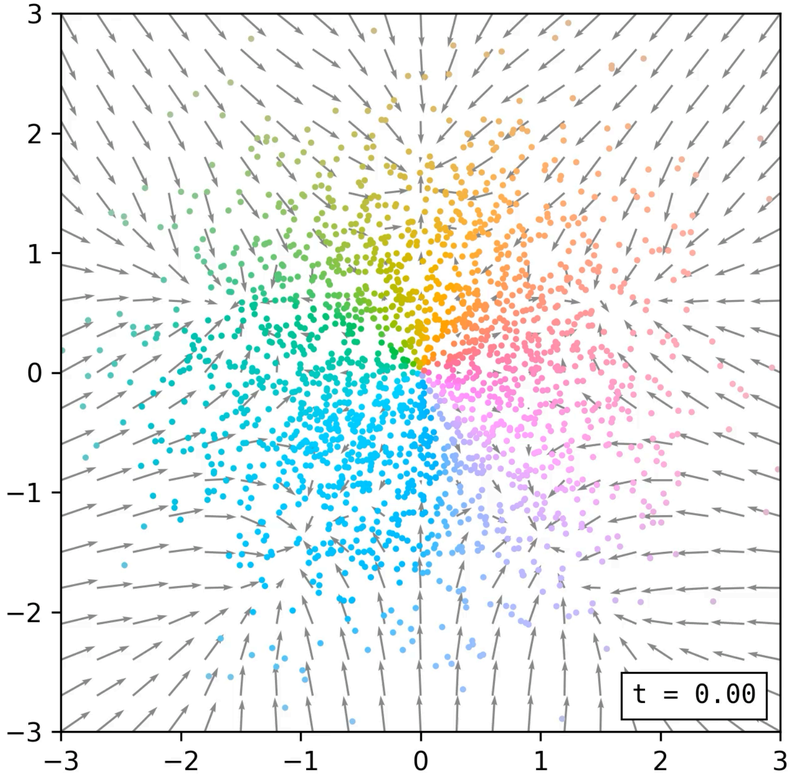
Paper: Flow Matching for Generative Modeling

Preliminaries: Vector fields and flows

- Data space x
- "Probability density path" p_t = time-dependent probability density function
- Time-dependent vector field v_t
  $\rightarrow$ constructs time-dependent map (=flow) ϕ_t via

$$\frac{\mathrm{d}}{\mathrm{d}t}\phi_t(x) = v_t\left(\phi_t(x)\right)$$

  starting from $\phi_0(x) = x$
  $\rightarrow$ parametrize vector field with CNF



Preliminaries: CNF

- CNF reshapes simple prior p_0 (noise) to complicated p_1 via push-forward equation

$$p_t = [\phi_t]_* p_0(x) = p_0(\phi_t^{-1}(x)) \det \left[\frac{\partial \phi_t^{-1}}{\partial x}(x) \right]$$

- Here: $p_0 = \mathcal{N}(0,1)$, $p_1 \approx q(x_1)$ with data distribution $q(x_1)$
- Goal: construct best probability path p_t that leads to $q(x_1)$

Flow Matching Objective

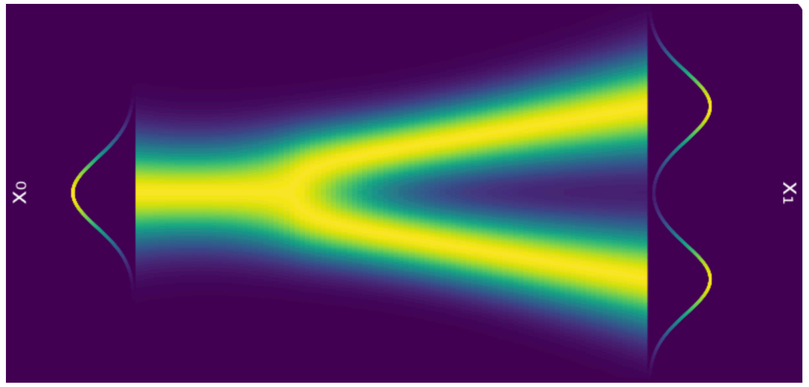
- Target probability path $p_t(x)$ and corresponding vector field $u_t(x)$

$$\mathscr{L}_{\text{FM}}(\theta) = \mathbb{E}_{t,p_t(x)} \| v_t(x;\theta) - u_t(x) \|^2 \qquad \text{with } x \sim p_t(x), t \sim \mathscr{U}[0,1]$$

  $\rightarrow$ Learn $v_t(x)$

- Problem: Intractable because we don't have access to closed form of $u_t(x)$

Conditional Probability Paths

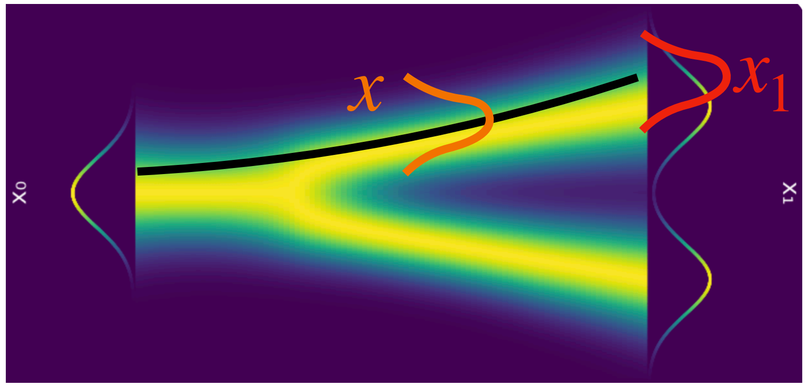


- Target probability path $p_t(x)$ (full distribution) unknown
  → Can we find a valid, simpler substitute?

- Construct target probability path $p_t(x)$ via mixture of simpler probability paths
  → **conditional probability path** $p_t(x \mid x_1)$ for data sample x_1

  - $t = 0$: $p_0(x \mid x_1) = p(x)$

  - $t = 1$: distribution centered around $x = x_1$, e.g. $p_1(x \mid x_1) = \mathcal{N}(x \mid x_1, \sigma^2 I)$



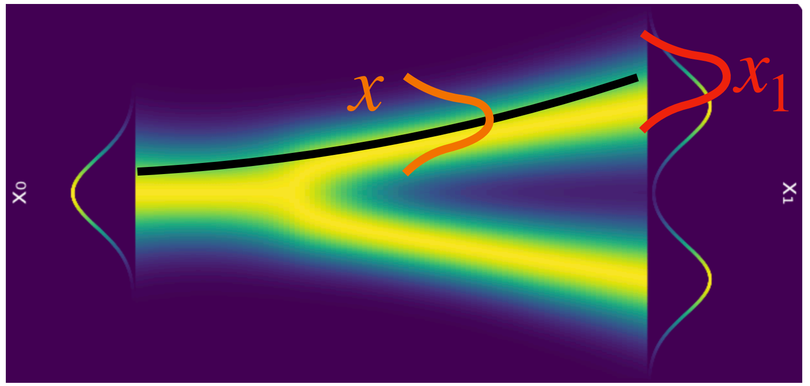
Conditional Probability Paths

- How do we get from the conditional probability path $p_t(x \mid x_1)$ back to the full target probability path $p_t(x)$?
  $\rightarrow$ Marginalize over all data samples x_1

$$p_t(x) = \int p_t(x \mid x_1) q(x_1) \, \mathrm{d}x_1$$

- At $t = 1$: recover data distribution $q(x)$

$$p_1(x) = \int p_1(x \mid x_1) q(x_1) \, \mathrm{d}x_1 \approx q(x)$$



Marginal vector field

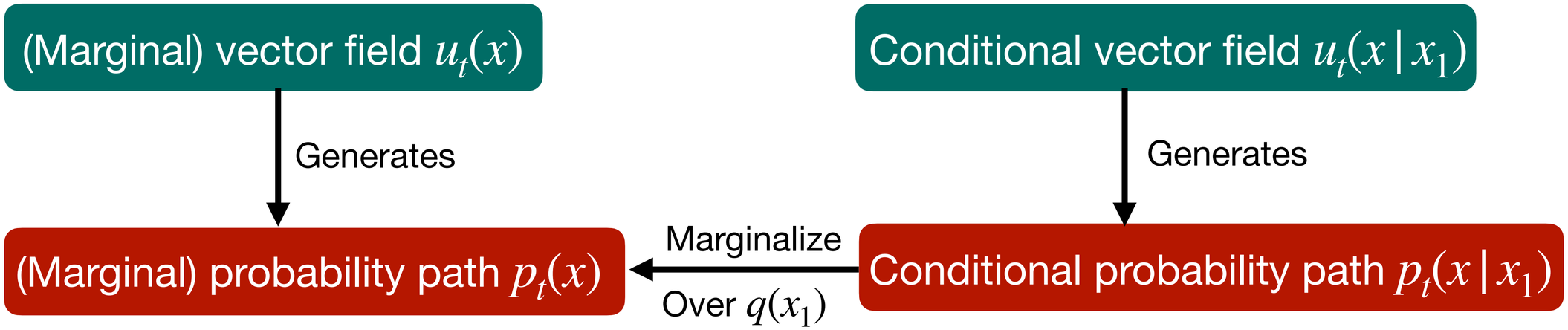
- Can we marginalize the vector fields equivalent to the probability paths?
  → Yes!

$$u_t(x) = \int u_t(x \mid x_1) \frac{p_t(x \mid x_1) q(x_1)}{p_t(x)} \, \mathrm{d}x_1$$

- We broke down the unknown and intractable marginal vector field $u_t(x)$ into simpler conditional vector fields $u_t(x \mid x_1)$!

Vector fields and probability paths

$$u_t(x) = \int u_t(x \mid x_1) \frac{p_t(x \mid x_1) q(x_1)}{p_t(x)} \, \mathrm{d}x_1$$



Proof (Appendix A)

Theorem 1. *Given vector fields $u_t(x|x_1)$ that generate conditional probability paths $p_t(x|x_1)$, for any distribution $q(x_1)$, the marginal vector field u_t in equation 8 generates the marginal probability path p_t in equation 6, i.e., u_t and p_t satisfy the continuity equation (equation 26).*

Proof. To verify this, we check that p_t and u_t satisfy the continuity equation (equation 26):

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

where in the second equality we used the fact that $u_t(\cdot|x_1)$ generates $p_t(\cdot|x_1)$, in the last equality we used equation 8. Furthermore, the first and third equalities are justified by assuming the integrands satisfy the regularity conditions of the Leibniz Rule (for exchanging integration and differentiation). □

Are we done yet?

- Still intractable because of integration in marginalization:

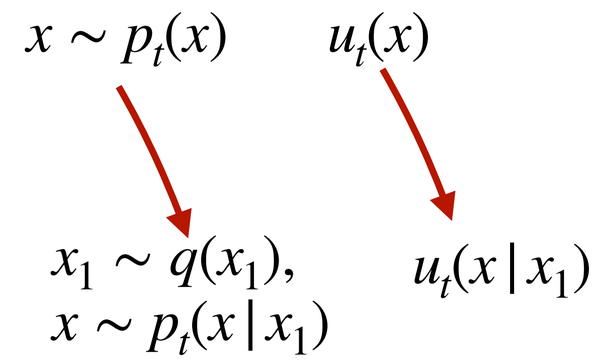
$$p_t(x) = \int p_t(x \mid x_1) q(x_1) \, \mathrm{d}x_1 \qquad u_t(x) = \int u_t(x \mid x_1) \frac{p_t(x \mid x_1) q(x_1)}{p_t(x)} \, \mathrm{d}x_1$$

- Standard flow matching objective intractable

$$\mathscr{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t, p_t(x)} \| v_t(x; \theta) - u_t(x) \|^2$$

- Instead: Conditional flow matching objective

$$\mathscr{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t, q(x_1) p_t(x \mid x_1)} \| v_t(x; \theta) - u_t(x \mid x_1) \|^2$$



Are the FM and CFM objective equivalent?

- Flow matching objective

$$\mathscr{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t,p_t(x)} \| v_t(x;\theta) - u_t(x) \|^2$$

- Conditional flow matching objective

$$\mathscr{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t,q(x_1)p_t(x|x_1)} \| v_t(x;\theta) - u_t(x \,|\, x_1) \|^2$$

Theorem 2. *Assuming that* $p_t(x) > 0$ *for all* $x \in \mathbb{R}^d$ *and* $t \in [0,1]$*, then, up to a constant independent of* θ*,* $\mathcal{L}_{CFM}$ *and* $\mathcal{L}_{FM}$ *are equal. Hence,* $\nabla_\theta \mathcal{L}_{FM}(\theta) = \nabla_\theta \mathcal{L}_{CFM}(\theta)$.

Proof (Appendix A)

Theorem 2. *Assuming that* $p_t(x) > 0$ *for all* $x \in \mathbb{R}^d$ *and* $t \in [0, 1]$*, then, up to a constant independent of* θ*,* $\mathcal{L}_{CFM}$ *and* $\mathcal{L}_{FM}$ *are equal. Hence,* $\nabla_\theta \mathcal{L}_{FM}(\theta) = \nabla_\theta \mathcal{L}_{CFM}(\theta)$.

Proof. To ensure existence of all integrals and to allow the changing of integration order (by Fubini's Theorem) in the following we assume that $q(x)$ and $p_t(x|x_1)$ are decreasing to zero at a sufficient speed as $\|x\| \to \infty$, and that $u_t, v_t, \nabla_\theta v_t$ are bounded.

First, using the standard bilinearity of the 2-norm we have that

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

...

Proof (Appendix A)

Next, remember that u_t is independent of θ and note that

$$\begin{aligned}\mathbb{E}_{p_t(x)}\|v_t(x)\|^2 &= \int \|v_t(x)\|^2 p_t(x)dx = \int \|v_t(x)\|^2 p_t(x|x_1)q(x_1)dx_1dx \\ &= \mathbb{E}_{q(x_1),p_t(x|x_1)}\|v_t(x)\|^2,\end{aligned}$$

where in the second equality we use equation 6, and in the third equality we change the order of integration. Next,

$$\begin{aligned}\mathbb{E}_{p_t(x)}\left\langle v_t(x), u_t(x)\right\rangle &= \int \left\langle v_t(x), \frac{\int u_t(x|x_1)p_t(x|x_1)q(x_1)dx_1}{p_t(x)}\right\rangle p_t(x)dx \\ &= \int \left\langle v_t(x), \int u_t(x|x_1)p_t(x|x_1)q(x_1)dx_1\right\rangle dx \\ &= \int \left\langle v_t(x), u_t(x|x_1)\right\rangle p_t(x|x_1)q(x_1)dx_1dx \\ &= \mathbb{E}_{q(x_1),p_t(x|x_1)}\left\langle v_t(x), u_t(x|x_1)\right\rangle,\end{aligned}$$

where in the last equality we change again the order of integration. □

Conclusion (so far)

- CFM objective valid substitute for FM objective

$$\mathscr{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,q(x_1)p_t(x|x_1)} \| v_t(x;\theta) - u_t(x \,|\, x_1) \|^2$$

  $\rightarrow$ Train CNF to generate $p_t(x)$ without accessing $p_t(x)$ or $u_t(x)$!

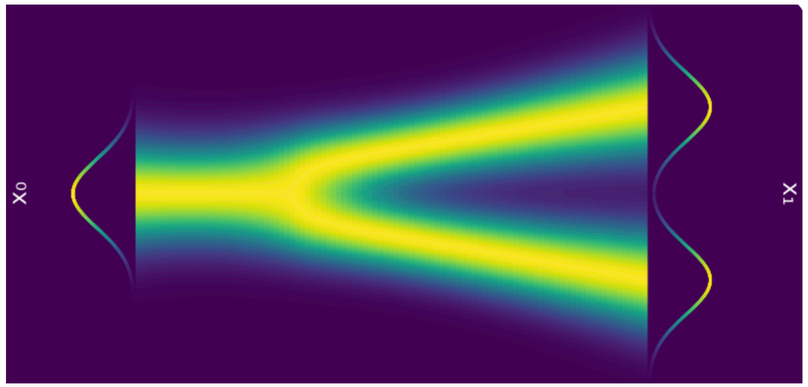
- Requirement: Suitable definitions of $p_t(x \,|\, x_1)$ and $u_t(x \,|\, x_1)$

How to construct $p_t(x \mid x_1)$ and $u_t(x \mid x_1)$?

- Family of Gaussian conditional probability paths

$$p_t(x \mid x_1) = \mathcal{N}(x \mid \mu_t(x_1), \sigma_t(x_1)^2 I) \qquad \text{With } \mu_0(x_1) = 0, \sigma_0(x_1) = 1$$

- At $t = 0$: $\mu_0(x_1) = 0, \sigma_0(x_1) = 1$
- At $t = 1$: $\mu_1(x_1) = x_1, \sigma_1(x_1) = \sigma_{\min}$



$t = 0$ $\qquad$ $t = 1$

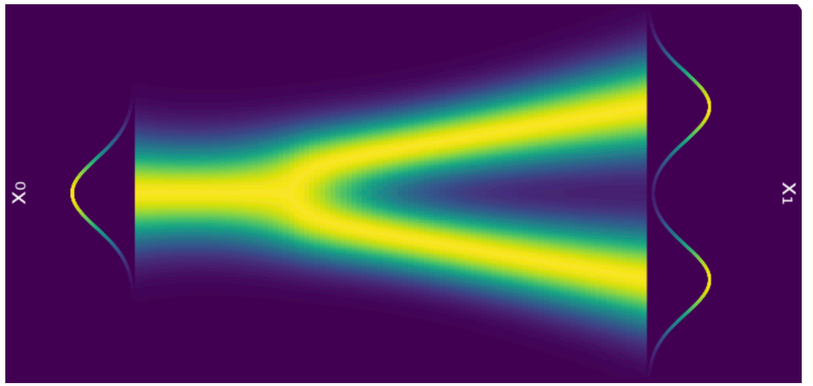
- Infinitely many options for definition of vector field
  $\rightarrow$ simplest: flow defined by affine transformation

$$\psi_t(x) = \sigma_t(x_1)x + \mu_t(x_1) \longrightarrow \frac{\mathrm{d}}{\mathrm{d}t}\psi_t(x) = u_t\left(\psi_t(x) \mid x_1\right)$$

How does the loss function change?

- Reparametrize x_1 by x_0

$$\psi_t(x) = \sigma_t(x_1)x + \mu_t(x_1) \longrightarrow \psi_t(x_0)$$



$t = 0$ $t = 1$

- CFM loss:

$$\mathscr{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,q(x_1)p_t(x|x_1)} \| v_t(x;\theta) - u_t(x \mid x_1) \|^2$$

$$\downarrow$$

$$\mathscr{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,q(x_1)p_t(x_0)} \left\| v_t(\psi_t(x_0)) - \frac{\mathrm{d}}{\mathrm{d}t}\psi_t(x_0) \right\|^2$$

How to parametrize the conditional vector field?

- ψ_t is affine map → Solve for u_t in closed form

$$\psi_t(x) = \sigma_t(x_1)x + \mu_t(x_1)$$

$$u_t\left(\psi_t(x)\,|\,x_1\right) = \frac{\mathrm{d}}{\mathrm{d}t}\psi_t(x) = \frac{\mathrm{d}}{\mathrm{d}t}\left(\sigma_t(x_1)x + \mu_t(x_1)\right) = \sigma_t'(x_1)x + \mu_t'(x_1)$$

$$= u_t\left(\sigma_t(x_1)x + \mu_t(x_1)\,|\,x_1\right) = \sigma_t(x_1)u_t\left(x\,|\,x_1\right) + \mu_t(x_1)$$

$$\sigma_t(x_1)u_t(x\,|\,x_1) + \mu_t(x_1) = \sigma_t'(x_1)x + \mu_t'(x_1)$$

$$u_t(x\,|\,x_1) = \frac{\sigma'(x_1)}{\sigma_t(x_1)}(x - \mu_t(x_1)) + \mu_t'(x_1)$$

Special instances of probability paths

How to select $\mu_t(x_1)$ and $\sigma_t(x_1)$?

1. Diffusion conditional vector fields
2. Optimal Transport conditional vector fields

1. Diffusion conditional vector fields

$$u_t(x \mid x_1) = \frac{\sigma'(x_1)}{\sigma_t(x_1)}(x - \mu_t(x_1)) + \mu_t'(x_1)$$

- Reverse variance exploding (VE) probability path

$$p_t(x) = \mathcal{N}(x \mid x_1, \sigma_{1-t}^2 I) \qquad \mu_t(x_1) = x_1 \qquad \sigma_t(x_1) = \sigma_{1-t}$$

$$u_t(x \mid x_1) = -\frac{\sigma_{1-t}'}{\sigma_{1-t}}(x - x_1)$$

- Reversed variance preserving (VP) probability path

$$\alpha_t = \exp -\frac{1}{2}\int_0^t \beta(s)\mathrm{d}s$$

$$p_t(x) = \mathcal{N}\left(x \mid \alpha_{1-t}x_1, (1 - \alpha_{1-t}^2)I\right) \qquad \mu_t(x_1) = \alpha_{1-t}x_1 \qquad \sigma_t(x_1) = \sqrt{1 - \alpha_{1-t}^2}$$

$$u_t(x \mid x_1) = \frac{\alpha_{1-t}'}{1 - \alpha_{1-t}^2}(\alpha_{1-t}x - x_1)$$

Side note: not defined for all t

1. Diffusion conditional vector fields

- Using diffusion probability paths with CFM objective
  → training alternative
  → apparently more stable and robust (compared to score matching)

2. Optimal Transport conditional vector fields

- More natural choice: linear change in time

$$\mu_t(x_x) = tx_1 \qquad \sigma_t(x) = 1 - (1 - \sigma_{min})$$

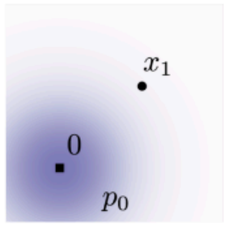
$$u_t(x \mid x_1) = \frac{x_1 - (1 - \sigma_{min})x}{1 - (1 - \sigma_{min})t}$$

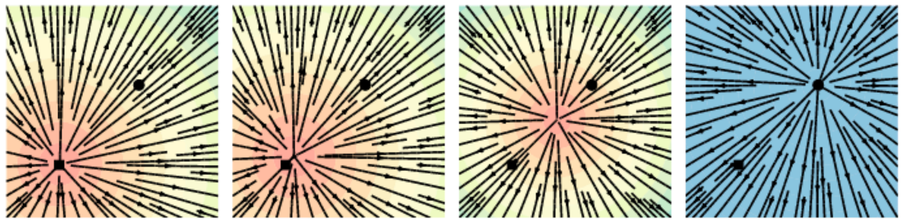
- CFM loss:

$$\mathscr{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t, q(x_1)p_t(x_0)} \left\| v_t(\psi_t(x_0)) - \left(x_1 - (1 - \sigma_{min})x_0\right) \right\|^2$$

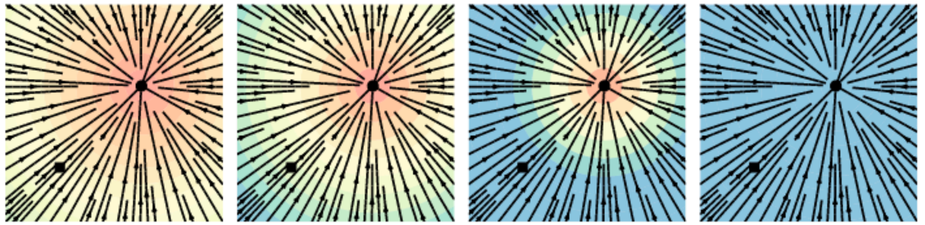
2. Optimal Transport conditional vector fields

- Why are these paths more intuitive?
  → linear in time



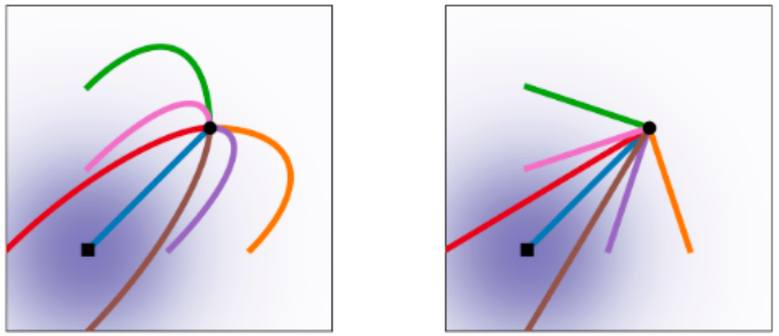


Diffusion path – conditional score function



OT path – conditional vector field

- Straight lines
  → no overshooting



Diffusion

OT

Advantages of FM

- More stable training
- Faster training, i.e. converges faster
- More efficient sampling
  due to more efficient ODE solvers

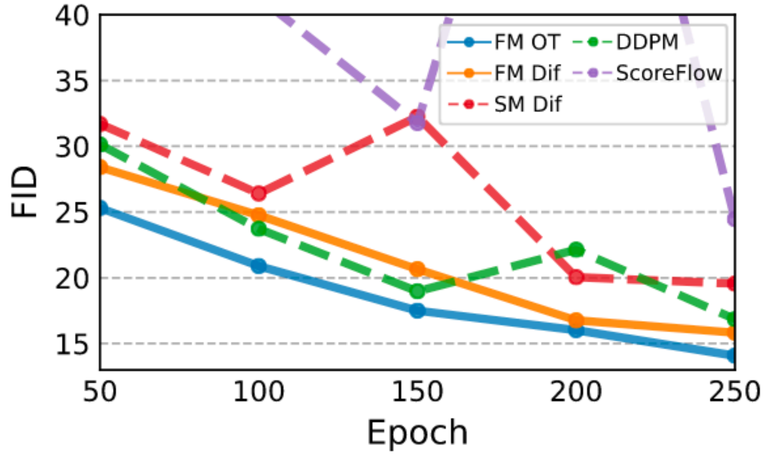


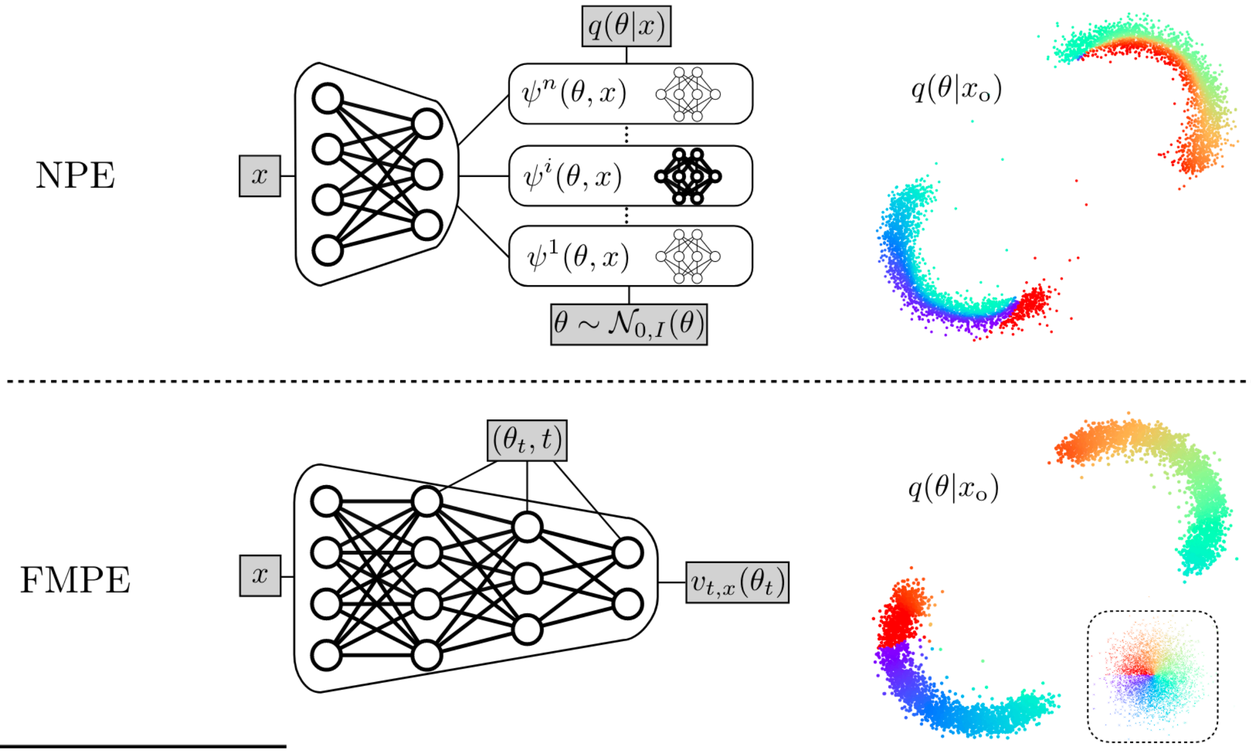
Image quality during training

Follow-ups

- Flow Matching for Scalable Simulation-based Inference
- Flow matching tutorial at NeurIPS 2024

Flow Matching for Scalable Simulation-based Inference

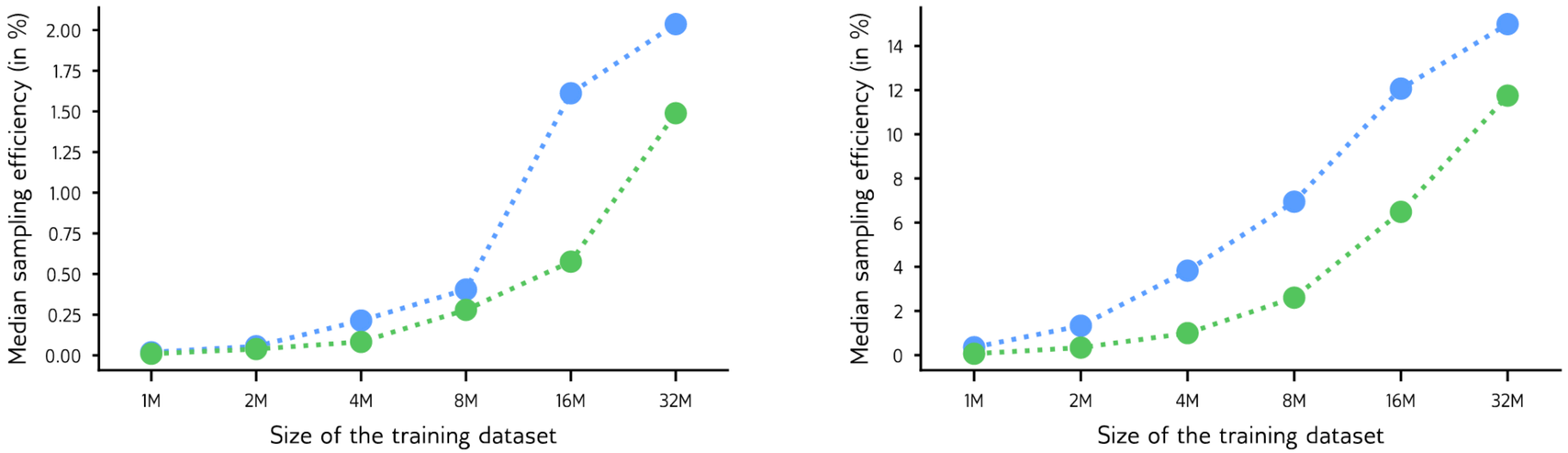
- More flexible network architecture than discrete normalizing flows
- But: significantly larger inference times



	Network Passes	Inference Time (per batch)
FMPE (sample only)	248	26s
FMPE (sample and log probs)	350	352s
NPE (sample and log probs)	1	1.5s

Flow Matching for Atmospheric Retrieval of Exoplanets: Where Reliability meets Adaptive Noise Levels

- Seem to scale better



(a) Results on default test set.

(b) Results on Gaussian test set.

Fig. E.1: Sampling efficiency as a function of the number of spectra used for training, for both FMPE and NPE.

Normalizing Flow Details

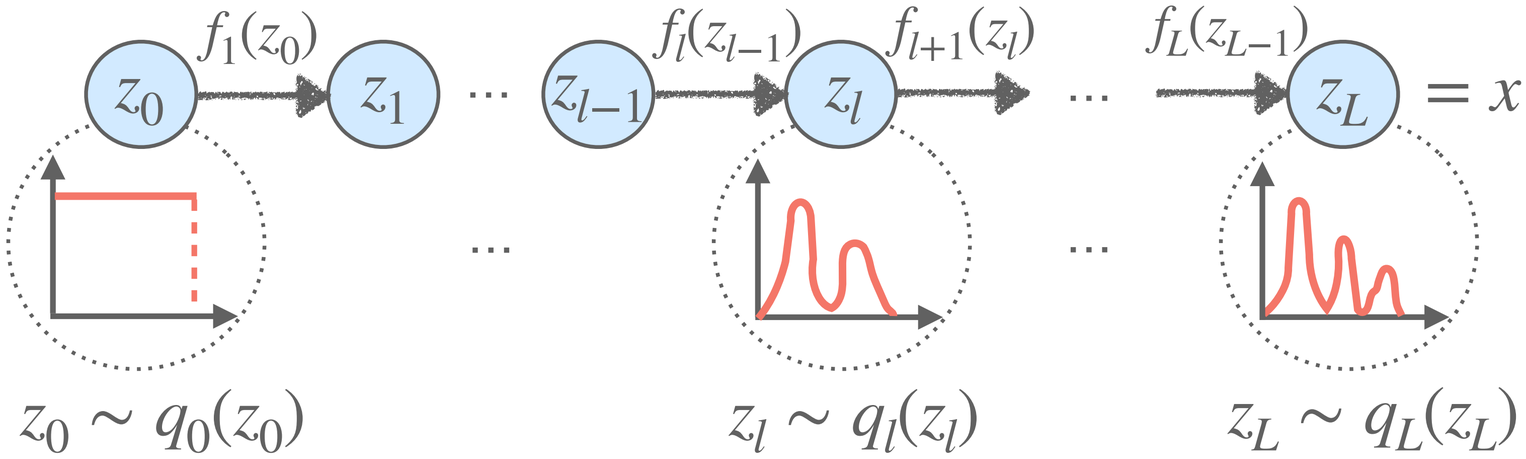
Chain of Transformations

Base Distribution
$q_0(z_0)$

Learned Flow Distribution
$q_\theta(x) = q_L(z_L)$

Chain of Transformations

Base Distribution
$q_0(z_0)$

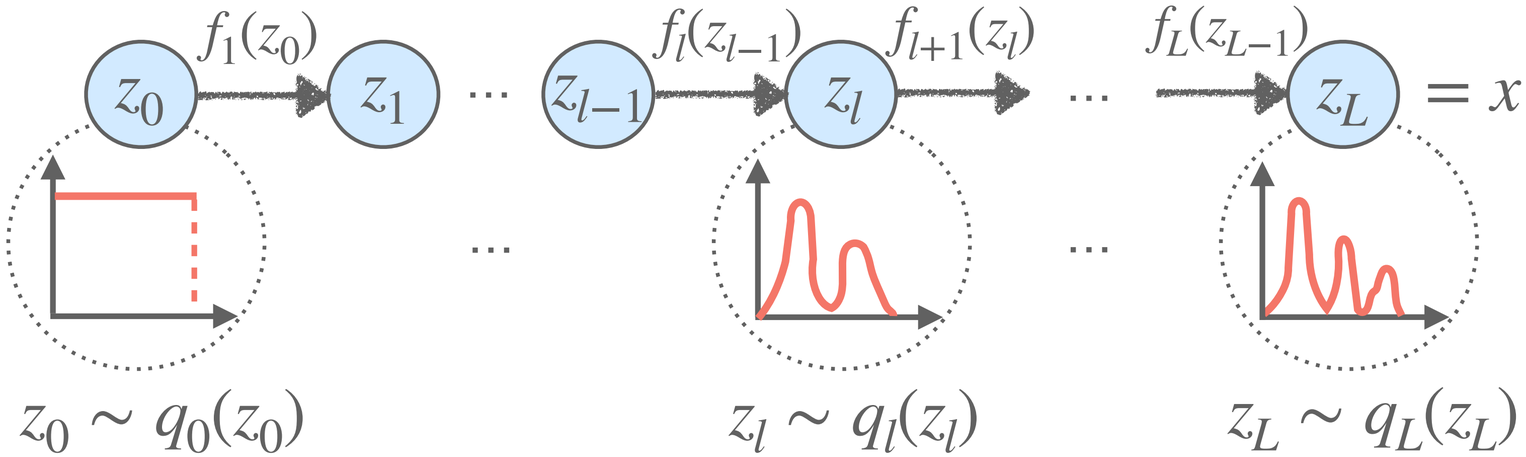


Learned Flow Distribution
$q_\theta(x) = q_L(z_L)$

Sampling: $z_0 \sim q_0(z_0)$ → $x = z_L = f_L \circ \ldots \circ f_1(z_0)$

Chain of Transformations

Base Distribution $q_0(z_0)$



Learned Flow Distribution $q_\theta(x) = q_L(z_L)$

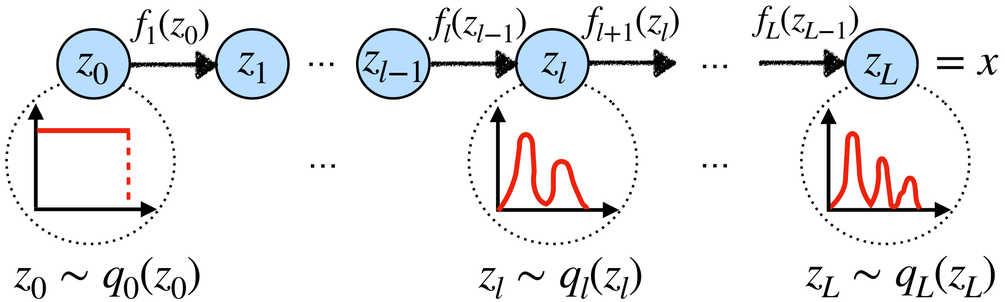
Evaluating the density: $z_0 = f_1^{-1} \circ \ldots \circ f_L^{-1}(x)$ ⟵ $x = z_L = f_L(z_{L-1})$

$$q(x) = q_0\big(f^{-1}(x)\big) \prod_{i=1}^{L} \left| \det \left(\frac{\partial f_i^{-1}(x)}{\partial x} \right) \right| = q_0(f^{-1}(x)) \prod_{i=1}^{L} \left| \det J_{f_i^{-1}} \right|$$

Example: Gaussian with two modes

Base Distribution

$q_0(z_0)$

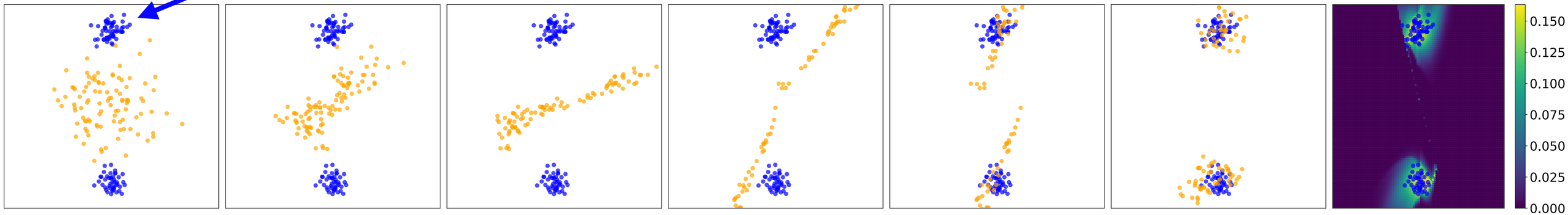


Learned Flow Distribution

$q_\theta(x) = q_L(z_L)$

Example:

Samples from target $p(x)$

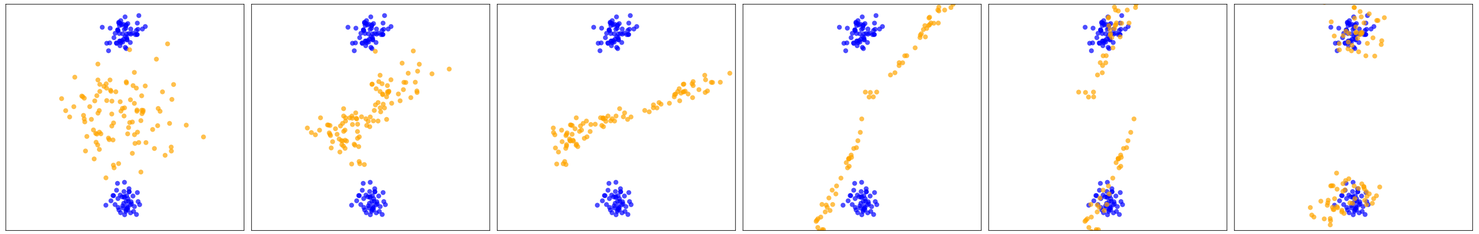


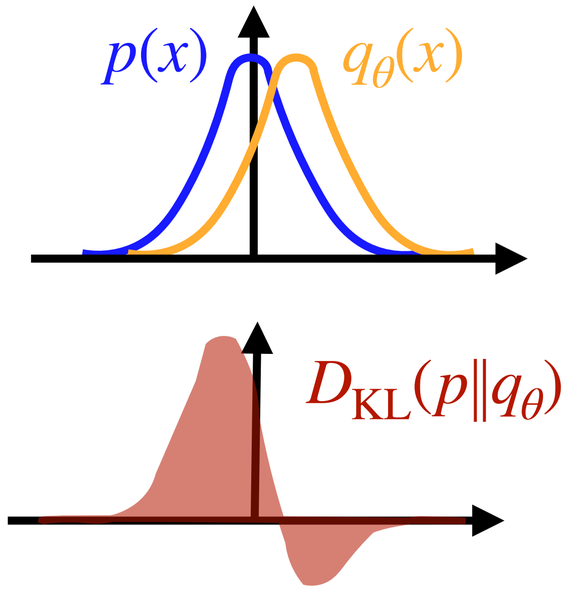
$z_0 \sim q_0(z_0)$

$z_5 \sim q_5(z_5)$ $q(x) = q_5(z_5)$

How to train a normalizing flow?

General question in generative modeling:
How can we compare two distributions?





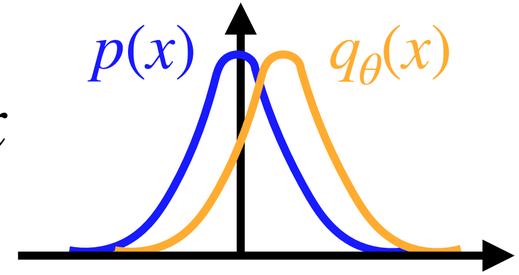
→ divergences from probability theory

- Divergence = difference measure in probability space
- Quantifies how dis-similar two distributions are → smaller is better

Standard choice: Kullback-Leibler divergence $D_{\mathrm{KL}}(p\,\|q_\theta) = \int p(x) \log \frac{p(x)}{q_\theta(x)}\,\mathrm{d}x$

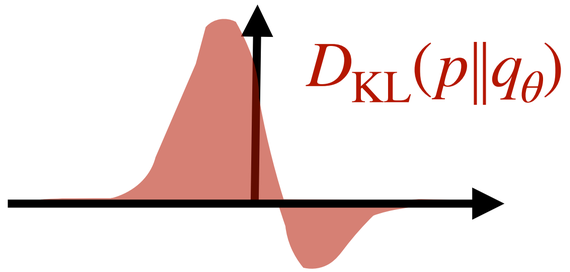
How to train a normalizing flow?

Kullback-Leibler divergence: $D_{\mathrm{KL}}(p \,\|\, q_\theta) = \int p(x) \log \frac{p(x)}{q_\theta(x)} \, \mathrm{d}x$



$$\mathscr{L} = D_{\mathrm{KL}}(p \,\|\, q_\theta) = \mathbb{E}_{x\sim p(x)}\left[\log \frac{p(x)}{q_\theta(x)}\right]$$

$$= \mathbb{E}_{x\sim p(x)}[\log p(x)] - \mathbb{E}_{x\sim p(x)}[\log q_\theta(x)]$$



$$\nabla_\theta \mathscr{L} = \nabla_\theta D_{\mathrm{KL}}(p \,\|\, q_\theta) = -\nabla_\theta \mathbb{E}_{x\sim p(x)}[\log q_\theta(x)]$$

$$= -\sum_{i=1}^{N} \nabla_\theta \log q_\theta(x_i) + \text{const}. \qquad \text{with } x_i \sim p(x_i)$$

→ Loss is negative log-likelihood: $\mathscr{L} = -\sum_{i=1}^{N} \log q_\theta(x_i)$

How to train a normalizing flow?

Loss is negative log-likelihood: $\mathscr{L} = -\sum_{i=1}^{N} \log q_\theta(x_i)$ with $x_i \sim p(x_i)$

Standard setting:

- (Simulated) data set available $\rightarrow x_i \sim p(x_i)$
- But $p(x)$ unknown/intractable