Deep Generative Models

Flow-Matching Models

Hamid Beigy

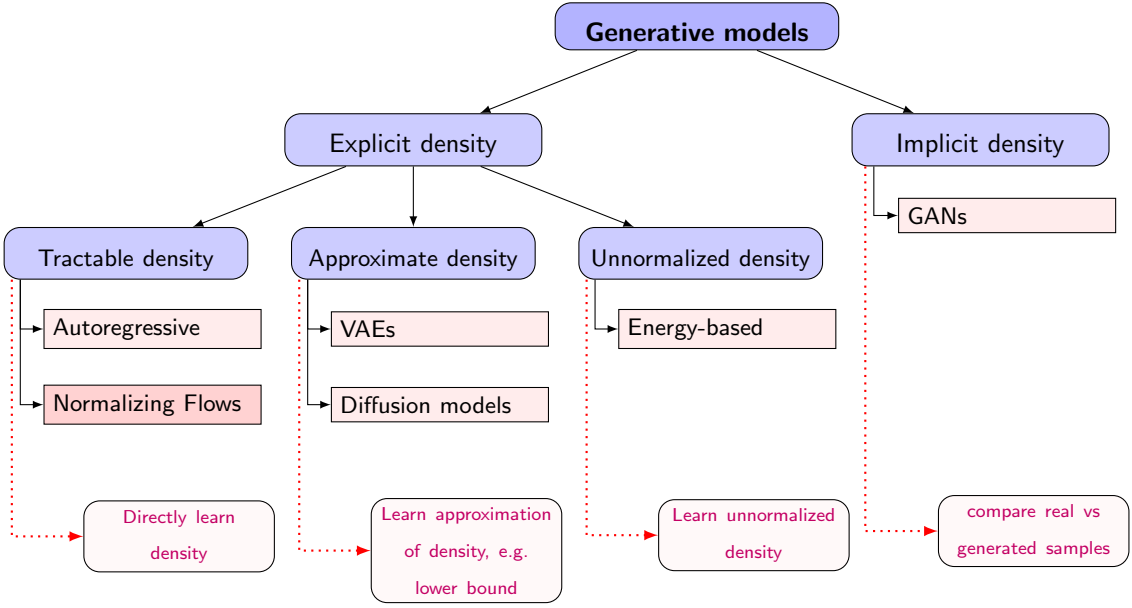
Sharif University of Technology

May 24, 2025

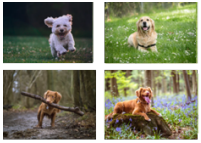
Table of contents

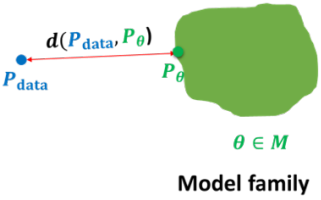
Introduction

Generative models categories



1. Assume that the observed variable $\mathbf{x}$ is a random sample from an underlying process, whose true distribution $p_{\mathrm{d}}(\mathbf{x})$ is unknown.





2. We attempt to approximate this process with a chosen model, $p_\theta(\mathbf{x})$, with parameters θ such that $\mathbf{x} \sim p_\theta(\mathbf{x})$.

3. Learning is the process of searching for the parameter θ such that $p_\theta(\mathbf{x})$ well approximates $p_{\mathrm{d}}(\mathbf{x})$ for any observed $\mathbf{x}$, i.e.

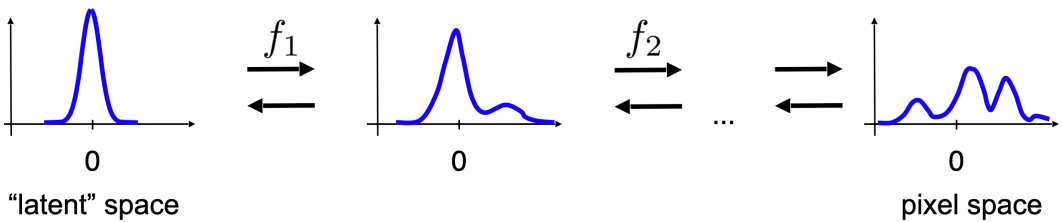
$$p_\theta(\mathbf{x}) \approx p_{\mathrm{d}}(\mathbf{x})$$

4. We wish $p_\theta(\mathbf{x})$ to be sufficiently flexible to be able to adapt to the data for obtaining sufficiently accurate model and to be able to incorporate prior knowledge.

Credit: Aditya Grover

What is Normalizing Flows?

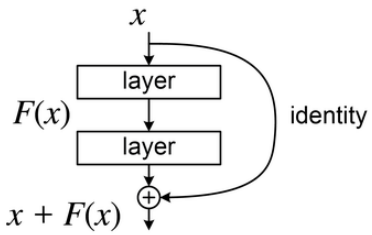
1. Normalizing Flow (NF) models are used for better and more powerful distribution approximation (Rezende and Mohamed 2015).
2. A normalizing flow transforms a simple distribution into a complex one by applying a sequence of invertible transformation functions.



3. Some methods for constructing normalizing flows
   - Coupling flows
   - Autoregressive flows
   - Residual flows

Residual flows

1. A residual network is a composition of residual connections, which are functions of the form $f(\mathbf{z}) = \mathbf{z} + F(\mathbf{z})$.
2. The function $F : \mathbb{R}^D \mapsto \mathbb{R}^D$ is called the residual block.
3. Under certain conditions on F, the residual connection f becomes invertible.



4. Flows composed of invertible residual connections are referred as residual flows.

Ordinary differential equations

Initial value problem is expressed as

$$\frac{d\mathbf{x}_t}{dt} = f_\theta(\mathbf{x}_t, t) \qquad \mathbf{x}_{t_0} = \mathbf{x}_0 \qquad \mathbf{x}_{t_1} = ?$$

Solution

$$\mathbf{x}_{t_1} = \mathbf{x}_{t_0} + \int_{t_0}^{t_1} f_\theta(\mathbf{x}_t, t)\mathrm{d}t$$

Example

Let

$$\frac{d_t}{dt} = 2t \qquad x_0 = 2 \qquad x_1 = ?$$

We have

$$\begin{aligned} x_1 &= x_0 + \int_0^1 2t\mathrm{d}t \\ &= 2 + t^2\Big|_0^1 \\ &= 2 + 1 - 0 = 3 \end{aligned}$$

Example

Let

$$\frac{dx_t}{dt} = 2xt \qquad x_0 = 2 \qquad x_1 = ?$$

We have

$$\int \frac{1}{2x}\mathrm{d}x = \int t\mathrm{d}t$$
$$\frac{1}{2}\log x = \frac{1}{2}t^2 + c_0$$
$$x_t = ce^{t^2}$$
$$x_0 = 3 \qquad \Rightarrow \quad c = 2$$
$$x_t = 2e^{t^2} \Rightarrow \quad x_1 = 5.436$$

1. What if $\int_{t_0}^{t_1} f_\theta(\mathbf{x}_t, t)\mathrm{d}t$ can not be analytically integrated?
2. We use approximation to $\int_{t_0}^{t_1} f_\theta(\mathbf{x}_t, t)\mathrm{d}t$, i.e. **numerical integration**
   - Euler method
   - Runge-Kutta method

1. Consider an ODE of the form

$$\frac{dy(t)}{dt} = f(y(s), t) \qquad y(t_0) = y_0$$

where $f(y(s), t)$ is a known function.

2. The exact solution to this ODE can be expressed in integral form:

$$y(t) = y_0 + \int_0^t f(y(s), s)ds$$

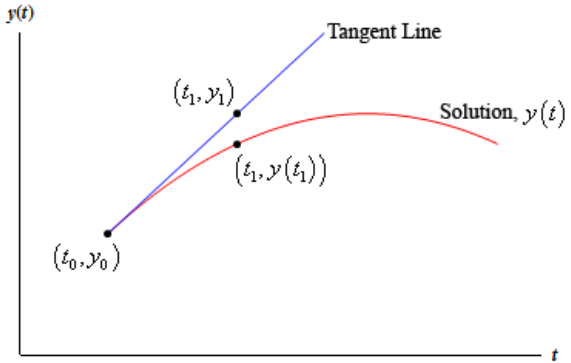
3. We want to approximate the solution near $t = t_0$.
4. We start with two pieces of information that we know about the solution:
   - We know the value of solution at $t = t_0$ from the initial condition.
   - We know the value of derivative at $t = t_0$ by plugging the initial condition into the differential equation.
5. Hence, the derivative equals to

$$\left.\frac{dy(t)}{dt}\right|_{t=t_0} = f(y_0, t_0)$$

1. These information are enough to write down the equation of tangent line to the solution at $t = t_0$ as

$$y(t) = y_0 + f(y_0, t_0) \times (t - t_0)$$

2. Now, consider the following figure



3. When t_1 is sufficiently close to t_0, point y_1 on the tangent line should be fairly close to the actual value of the solution at t_1.

1. We can find $y_1 = y(t_1)$ easily by plugging t_1 in the equation for tangent line as:

$$y_1 = y_0 + f(y_0, t_0) \times (t_1 - t_0)$$

2. When y_1 is accurate approximation of solution, it is used to estimate the tangent line at t_1 by constructing a line through the point (t_1, y_1) that has slope $f(y_1, t_1)$.

3. This estimation gives

$$y(t) = y_1 + f(y_1, t_1) \times (t - t_0)$$

4. Next, we approximate the solution at $t = t_2$ and proceed accordingly.

5. Then, we can obtain the next approximation as

$$\begin{aligned} y_2 &= y_1 + f(y_1, t_1) \times (t_2 - t_1) \\ y_3 &= y_2 + f(y_2, t_2) \times (t_3 - t_2) \\ &\vdots \\ y_{n+1} &= y_n + f(y_n, t_n) \times (t_{n+1} - t_n) \end{aligned}$$

6. Assume that step sizes $t_0, t_1, t_2, \ldots$ are of a uniform size of h, i.e. $t_{n+1} - t_n = h$, for all n.

7. The next approximation is $y_{n+1} = y_n + h \times f(y_n, t_n)$

Example

1. Let $\frac{dy(t)}{dt} = 2 - 2y(t) - e^{4t}$, where $y(0) = 1$.
2. Also let $h = 0.1$.
3. Then, we approximate values of solution at $t = 0.1, 0.2, 0.3, 0.4, 0.5$ and compare them with the exact solution of ODE, given by

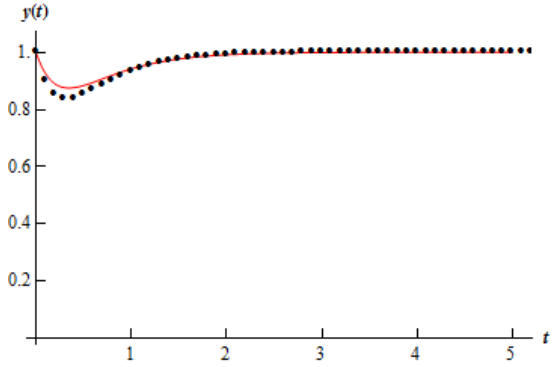
$$y(t) = 1 + \frac{1}{2}\left(e^{-4t} - e^{-2t}\right).$$

4. We have $f(y(t), t) = 2 - 2y(t) - e^{4t}$. Then, we can approximate the solution as

$$\begin{aligned} y_1 &= y_0 + h \times f(y_0, t_0) = 0.900 \\ y_3 &= y_1 + h \times f(y_1, t_1) = 0.850 \\ y_3 &= y_2 + h \times f(y_2, t_2) = 0.837 \\ y_4 &= y_3 + h \times f(y_3, t_3) = 0.835 \\ y_5 &= y_4 + h \times f(y_4, t_4) = 0.851 \end{aligned}$$

Example

Comparison of exact solution (continuous line) and approximation (discrete dots) for $h = 0.1$.



1. Now, extending the method to a vector field. Let ODE of form

$$\frac{d\mathbf{y}(t)}{dt} = f(\mathbf{y}(t), t) \qquad \mathbf{y}(t_0) = \mathbf{y}_0$$

2. The Euler's method starts from $t = 0$ and proceeding with a step size of h, so

$$\mathbf{y}(t + h) = f(\mathbf{y}(t), t) \times h + \mathbf{y}(t)$$

Initial value problem is expressed as

$$\frac{d\mathbf{x}_t}{dt} = f_\theta(\mathbf{x}_t, t) \qquad \mathbf{x}_{t_0} = \mathbf{x}_0 \qquad \mathbf{x}_{t_1} = ?$$

Solution

$$\mathbf{x}_{t_1} = \mathbf{x}_{t_0} + \int_{t_0}^{t_1} f_\theta(\mathbf{x}_t, t)\mathrm{d}t$$

$$\mathbf{x}_{t_1} = ODESolver(f_\theta(\mathbf{x}_t, t), \mathbf{x}_{t_0}, t_0, t_1)$$

- Final time
- Initial time
- Initial value
- Differential
- Any ODE solver

Neural ODE

1. Initial value problem is expressed as

$$\frac{d\mathbf{x}_t}{dt} = f_\theta(\mathbf{x}_t, t) \qquad \mathbf{x}_{t_0} = \mathbf{x}_0 \qquad \mathbf{x}_{t_1} = ?$$

2. Solution

Exact	Numerical
$\mathbf{x}_{t_1} = \mathbf{x}_{t_0} + \int_{t_0}^{t_1} f_\theta(\mathbf{x}_t, t)\mathrm{d}t$	$\mathbf{x}_{t_1} = \mathit{ODESolver}(f_\theta(\mathbf{x}_t, t), \mathbf{x}_{t_0}, t_0, t_1)$

3. In neural ODE, f_θ is a neural network parametrized by θ (T. Q. Chen et al. 2018).
4. This is a **paradigm shift**:
   - In earlier methods, f_θ was pre-defined/hand-designed according to the domain.
   - In neural ODE, we want to estimate f_θ that suits our objective.

ODE

1. Initial value problem is expressed as

$$\frac{d\mathbf{x}_t}{dt} = f_\theta(\mathbf{x}_t, t) \qquad \mathbf{x}_{t_0} = \mathbf{x}_0$$

2. Using Euler discretization

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h f_\theta(\mathbf{x}_n, n)$$

3. Forward propagation

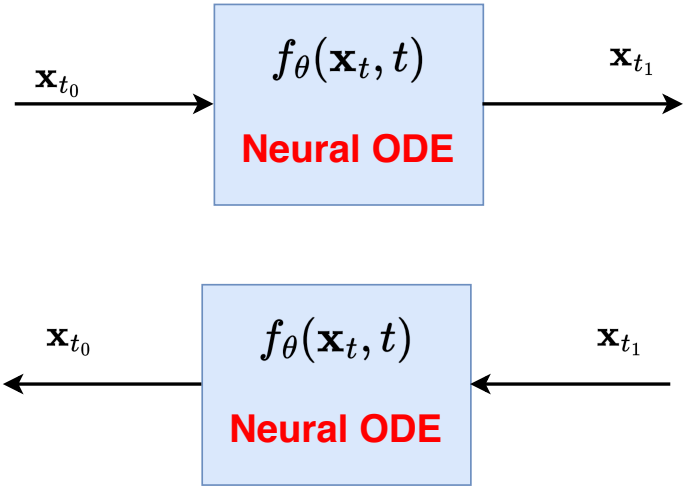
$$\mathbf{x}_{t_1} = ODESolver(f_\theta(\mathbf{x}_t, t), \mathbf{x}_{t_0}, t_0, t_1)$$

4. Update θ using gradient-based learning
5. **How to compute gradient of loss function?**
6. Back-propagate through ODESolver! **High memory cost!!**
7. Better method: Adjoint method

Residual networks

1. The output of a residual block is

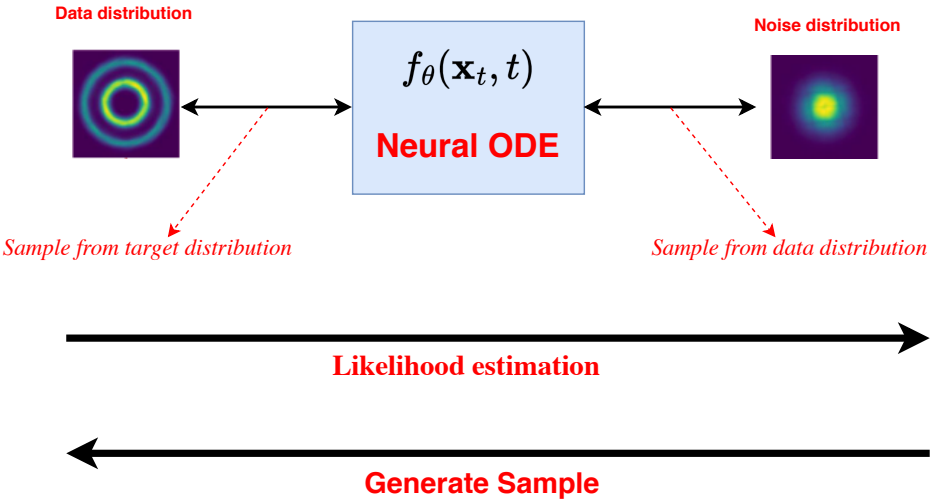
1. Neural ODEs are reversible models.



2. **They integrate forward/backward in time.**

Continuous flows

Continuous flows are continuous version of normalizing flows (Grathwohl et al. 2019).



1. In residual flows, the transformation is expressed as

$$\mathbf{x}_k = \psi_k(\mathbf{x}_{k-1}) = \mathbf{x}_{k-1} + \delta v(\mathbf{x}_{k-1})$$

for some $\delta > 0$ and Lipschitz residual connection v.

2. By rearranging this equation, we obtain

$$v(\mathbf{x}_{k-1}) = \frac{\mathbf{x}_k - \mathbf{x}_{k-1}}{\delta}$$

3. Setting $\delta = \frac{1}{K}$ and $K \to \infty$, then $\psi = \psi_K \circ \psi_{K-1} \circ \ldots \circ \psi_2 \circ \psi_1$ is given by ODE:

$$\frac{d\mathbf{x}_t}{dt} = \lim_{\delta \to 0} \frac{\mathbf{x}_{t+\delta} - \mathbf{x}_t}{\delta} = \lim_{\delta \to 0} \frac{\psi_t(\mathbf{x}_t) - \mathbf{x}_t}{\delta} = v(\mathbf{x}_t, t),$$

for $t \in [0, 1]$.

4. The flow of ODE $\psi_t : [0, 1] \times \mathbb{R}^d \mapsto \mathbb{R}^d$ is defined such that

$$\frac{d\psi_t}{dt} = v(\psi_t(\mathbf{x}_0), t).$$

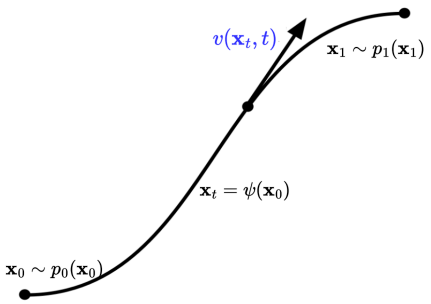
1. The flow of ODE is

$$\frac{d\psi_t}{dt} = v(\psi_t(\mathbf{x}_0), t).$$

where

- $\mathbf{x}_t$ is the state of the system.
- $v(\mathbf{x}_t, t)$ is vector field, called **velocity field**.

2. At the time
   - 0 : $p_0(\mathbf{x}_0)$ is the the standard Gaussian distribution.
   - 1 : $p_1(\mathbf{x}_1)$ is the distribution of data such. We need to be close to $p_{\text{d}}(\mathbf{x})$.



1. The $\frac{d\psi_t}{dt} = v(\psi_t(\mathbf{x}_0), t)$ states that transformation ψ_t maps initial condition $\mathbf{x}_0$ to the solution at time t denoted by $\mathbf{x}_t$ as:

$$\mathbf{x}_t \triangleq \psi_t(\mathbf{x}_0) = \mathbf{x}_0 + \int_0^t v(\mathbf{x}_s, s)ds \tag{1}$$

2. This ODE is called an **initial value problem**, controlled by **velocity field** $v(\mathbf{x}_t, t)$.
3. Additionally, two important objects in continuous normalizing flow are
   - the flow $\psi_t(\mathbf{x})$ and
   - the probability path $p_t(\mathbf{x})$, which is the distribution of $\psi_t(\mathbf{x})$
4. The continuity equation (transport equation) links $p_t(\mathbf{x})$ and $v(\mathbf{x}_t, t)$.
5. In probability, **continuity equation** is analogous to **conservation of mass** in fluid dynamics:

$$\frac{\partial\, p_t(\mathbf{x})}{\partial t} + \nabla \cdot \mathbf{j}(\mathbf{x}_t, t) = 0,$$

6. $\mathbf{j}(\mathbf{x}_t, t) = v(\mathbf{x}_t, t)\, p_t(\mathbf{x})$ is the **probability flux** describing flow of probability density.

1. The **continuity equation** maintains conservation of probability:

$$\frac{\partial\, p_t(\mathbf{x})}{\partial t} + \nabla \cdot \mathbf{j}(\mathbf{x}_t, t) = 0,$$

2. The divergence of a d-dimensional vector field $\mathbf{g}$:

$$\nabla \cdot \mathbf{g}(\mathbf{x}) = \sum_{k=1}^{d} \frac{\partial \mathbf{g}_k(\mathbf{x})}{\partial x_k} = \operatorname{tr}\big(J_{\mathbf{g}(\mathbf{x})}\big)$$

   where $J_{\mathbf{g}(\mathbf{x})}$ is the Jacobian of vector field $\mathbf{g}(\mathbf{x})$.

3. $\nabla \cdot \mathbf{j}(\mathbf{x}_t, t)$ measures the rate at which probability density is **expanding/contracting** in a given region of space.

4. Multiplying the **continuity equation** with $\frac{1}{p_t(\mathbf{x})}$, results in:

$$\frac{1}{p_t(\mathbf{x}_t)} \frac{\partial\, p_t(\mathbf{x}_t)}{\partial t} + \frac{1}{p_t(\mathbf{x}_t)} \nabla \cdot \big(v(\mathbf{x}_t, t)\, p_t(\mathbf{x}_t)\big) = 0,$$

$$\frac{\partial \log\, p_t(\mathbf{x}_t)}{\partial t} + \langle \nabla_{\mathbf{x}} \log\, p_t(\mathbf{x}_t), v(\mathbf{x}_t, t) \rangle + \nabla \cdot v(\mathbf{x}_t, t) = 0,$$

$$\langle \nabla_{\mathbf{x}} \log\, p_t(\mathbf{x}_t), v(\mathbf{x}_t, t) \rangle = -\frac{\partial \log\, p_t(\mathbf{x}_t)}{\partial t} - \nabla \cdot v(\mathbf{x}_t, t).$$

1. The **continuity equation** maintains conservation of probability:

$$\langle \nabla_{\mathbf{x}} \log\, p_t(\mathbf{x}_t), v(\mathbf{x}_t, t) \rangle = -\frac{\partial \log\, p_t(\mathbf{x}_t)}{\partial t} - \nabla \cdot v(\mathbf{x}_t, t).$$

2. Calculating the total derivative of $\frac{d \log\, p_t(\mathbf{x}_t)}{dt}$:

$$\begin{aligned}
\frac{d \log\, p_t(\mathbf{x}_t)}{dt} &= \frac{\partial \log\, p_t(\mathbf{x}_t)}{\partial t} + \langle \nabla_{\mathbf{x}} \log\, p_t(\mathbf{x}_t), \frac{\partial \mathbf{x}_t}{\partial t} \rangle, \\
&= \frac{\partial \log\, p_t(\mathbf{x}_t)}{\partial t} + \langle \nabla_{\mathbf{x}} \log\, p_t(\mathbf{x}_t), v(\mathbf{x}_t, t) \rangle, \\
&= \frac{\partial \log\, p_t(\mathbf{x}_t)}{\partial t} - \frac{\partial \log\, p_t(\mathbf{x}_t)}{\partial t} - \nabla \cdot v(\mathbf{x}_t, t) \\
&= -\nabla \cdot v(\mathbf{x}_t, t) \\
&= -\operatorname{tr}\left(\frac{\partial v(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \right).
\end{aligned}$$

1. Computing the total change in log-density by integrating $\frac{d \log\ p_t(\mathbf{x}_t)}{dt}$ across time:

$$\int_0^1 \left(\frac{d \log\ p_t(\mathbf{x}_t)}{dt} + \operatorname{tr}\left(\frac{\partial v(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \right) \right) \mathrm{d}t = 0.$$

2. Simplifying the above integral:

$$\log\ p_1(\mathbf{x}_1) = \log\ p_0(\mathbf{x}_0) - \int_0^1 \operatorname{tr}\left(\frac{\partial v(\mathbf{x}_t, t)}{\partial \mathbf{x}_t} \right) \mathrm{d}t.$$

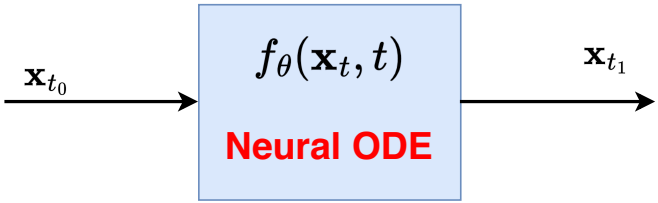
3. To compute $log\ p_t(\mathbf{x}_t)$, we can either solve both the time evolution of $\mathbf{x}_t$ and its log density $log\ p_t(\mathbf{x}_t)$ together,

$$\frac{d \begin{pmatrix} \mathbf{x}_t \\ \log\ p_t(\mathbf{x}_t) \end{pmatrix}}{dt} = \begin{pmatrix} v(\mathbf{x}_t, t) \\ -\nabla \cdot v(\mathbf{x}_t, t) \end{pmatrix}$$

or solve only for $\mathbf{x}_t$ and then estimate $log\ p_t(\mathbf{x}_t)$ using quadrature methods.

1. Parameterizing the vector field with a neural network with weights θ, $v_\theta(\mathbf{x}_t, t)$, leads to **neural ODE** (T. Q. Chen et al. 2018).
2. Let $\mathbf{x}_0$ be the initial condition for this ODE.
3. By integrating over time t, we solve it and get the output as given below.

$$\mathbf{x}_t = \int_0^t v_\theta(\mathbf{x}_s, s)\mathrm{d}s$$

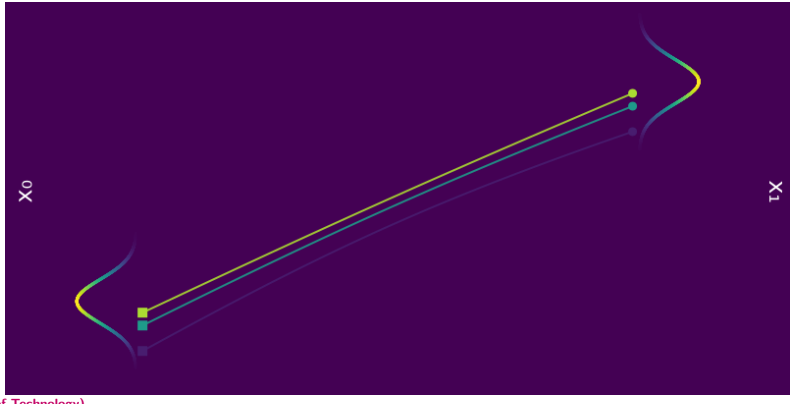


Example: Gaussian to a Gaussian (1D)

1. How to map a 1D Gaussian to another one with different mean?
2. We can derive a one-shot (i.e. discrete) flow bridging between two Gaussian distributions.
3. We want derive a time-continuous flow $\psi_t(x)$, corresponding to integration of $v(x_t, t)$.
4. Let

$$p_0(x) = \mathcal{N}(0, 1)$$
$$p_1(x) = \mathcal{N}(\mu, 1)$$

5. We can continuously bridge with a simple linear transformation $\psi_t(x_0) = x_0 + \mu t$ as



Example: Gaussian to a Gaussian (1D)

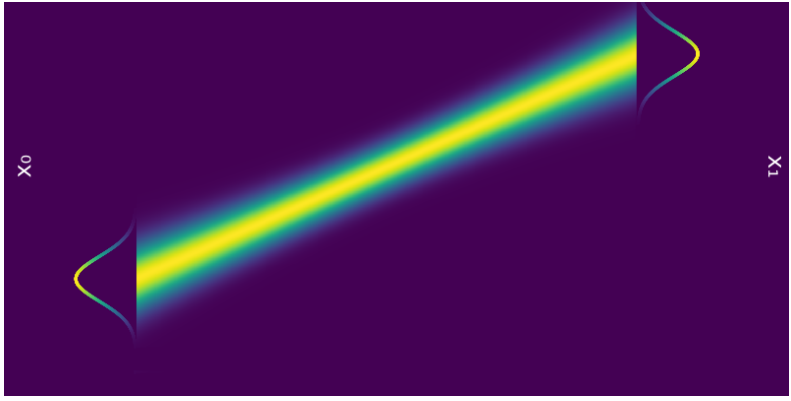
1. Every marginal $p_t(x)$ is a Gaussian, and also

$$\mathbb{E}_{p_0(x)}[\psi_t(x_0)] = \mu t$$

2. This implies that $\mathbb{E}_{p_0(x)}[\psi_1(x_0)] = \mu = \mathbb{E}_{p_1(x)}[x_1]$ and

$$\text{var}_{p_0(x)}[\psi_t(x_0)] = 1$$
$$\text{var}_{p_0(x)}[\psi_1(x_0)] = 1 = \text{var}_{p_1(x)}[x_1]$$

3. The probability path $p_t(x)]x = \mathcal{N}(\mu t, 1)$ bridges $p_0(x)$ and $p_1(x)$.



Example: Gaussian to a Gaussian (1D)

1. Now determine the vector field $v(x_t, t)$, which satisfies

$$\frac{d\psi_t(x_0)}{dt} = v(x_t, t)$$

2. We can plug $\psi(x_0, t) = x_0 + \mu t$ in on the left hand side to get

$$\frac{d\psi_t(x_0)}{dt} = \frac{d(x_0 + \mu t)}{dt} = \mu$$
$$v(x_t, t) = v(x_0 + \mu t, t)$$

3. It is easy to see that one such solution is the constant vector field

$$v(x_t, t) = \mu$$

4. We can also define $v(x_t, t)$ such that $p_0(x) \xrightarrow{v(x_t,t)} p_1(x)$ and derive the corresponding $\psi_t(x_0)$ by solving the ODE.

Training continuous flows

1. To construct flow, we maximize the log-likelihood of $p_t(\mathbf{x})$.
2. Maximizing the log-likelihood minimizes $\mathrm{D}_{KL}(\, p_{\mathrm{d}}(\mathbf{x}) \,||\, p_t(\mathbf{x}))$.
3. The log-likelihood of $p_t(\mathbf{x})$ can be written as

$$\begin{aligned}\mathrm{LL}(\theta) &= \mathbb{E}_{\mathbf{x}\sim p_{\mathrm{d}}(\mathbf{x})}[\log\, p_1(\mathbf{x})] \\ &= \mathbb{E}_{\mathbf{x}\sim p_{\mathrm{d}}(\mathbf{x})}\left[\log\, p_0(\mathbf{x}_0) - \int_0^1 \mathrm{tr}\left(\frac{\partial v(\mathbf{x}_t, t)}{\partial \mathbf{x}_t}\right)\mathrm{d}t\right],\end{aligned}$$

4. Expectation is taken over data distribution, $\log\, p_1(\mathbf{x})$ represents parametric distribution.
5. Maximizing the log-likelihood requires:
   - Expensive numerical ODE simulations at training time!
   - Estimators for the divergence to scale nicely with high dimension.
6. Change of variables:

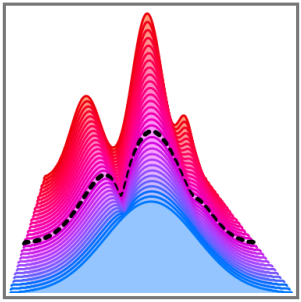
$$\log\, p_1(\mathbf{x}) - \log\, p_0(\mathbf{x}) = \log\det\left(\frac{dv_\theta}{d\mathbf{x}_t}\right)$$

7. Instantaneous change of variables:

$$\frac{\partial \log\, p_t(\mathbf{x})}{\partial t} = -\,\mathrm{tr}\left(\frac{\partial v_\theta(\mathbf{x}_t, t)}{\partial \mathbf{x}_t}\right)$$

Training continuous flows

1. This expectation necessitates expensive numerical ODE simulations during training.
2. This numerical ODE simulations affect the scalability of estimators when dealing with high dimensions.
3. Continuous normalizing flows are highly expressive because they parameterize a wide variety of flows and can represent many probability distributions.
4. Training CNFs can be very slow due to the need for ODE integration at each iteration.



- Blue distribution: Noise distribution $p_0(\mathbf{x})$.
- Red distribution: Data distribution $p_1(\mathbf{x}) \approx p_d(\mathbf{x})$.
- Dashed distribution: The probability path $p_t(\mathbf{x})$.

Calculating the log-likelihood for CNFs

1. Unlike in discrete time normalizing flows, we do not require invertibility of v,
2. Hence, we cannot invert the transformation to obtain $\mathbf{x}_0$ for given datapoint $\mathbf{x}_1$.
3. Under some conditions, we can uniquely solve the following problem (Grathwohl et al. 2019).

$$\begin{bmatrix} \mathbf{x}_0 \\ \ln\, p_1(\mathbf{x}_1) - \ln\, p_0(\mathbf{x}_0) \end{bmatrix} = \int_1^0 \begin{bmatrix} v_\theta(\mathbf{x}_t, t) \\ -\operatorname{tr}\left(\frac{\partial v_\theta(\mathbf{x}_t, t)}{\partial \mathbf{x}_t}\right) \end{bmatrix} \mathrm{d}t$$

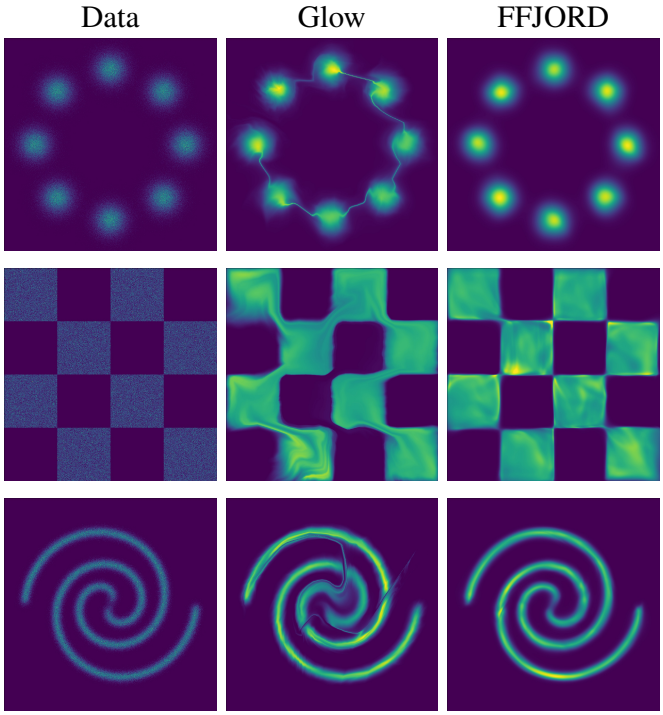
with initial conditions:

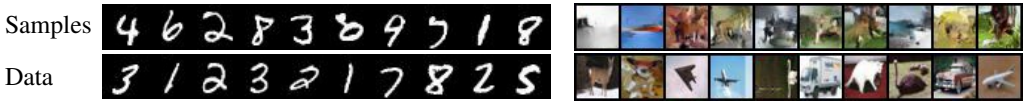
$$\begin{bmatrix} \mathbf{x}_1 \\ \ln\, p_1(\mathbf{x}_\mathrm{d}) - \ln\, p_1(\mathbf{x}_1) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_\mathrm{d} \\ 0, \end{bmatrix}$$

where $\mathbf{x}_1$ is a datapoint $\mathbf{x}_\mathrm{d}$.

4. We do the following steps:
   - Take a datapoint $\mathbf{x}_1 = \mathbf{x}_\mathrm{d}$.
   - Solve the above integral by applying a solver to find $\mathbf{x}_0$ and keeping track of traces over time.
   - Calculate the log-likelihood by adding $\ln\, p_0(\mathbf{x}_0)$ to the sum of negative traces $-\operatorname{tr}\left(\frac{\partial v_\theta(\mathbf{x}_t, t)}{\partial \mathbf{x}_t}\right)$.

Continuous flows





Negative log-likelihood on test data for density estimation models:

	POWER	GAS	HEPMASS	MINIBOONE	BSDS300	MNIST	CIFAR10
Real NVP	-0.17	-8.33	18.71	13.55	-153.28	1.06*	3.49*
Glow	-0.17	-8.15	18.92	11.35	-155.07	1.05*	**3.35***
FFJORD	**-0.46**	**-8.59**	**14.92**	**10.43**	**-157.40**	**0.99*** (1.05†)	3.40*
MADE	3.08	-3.56	20.98	15.59	-148.85	2.04	5.67
MAF	-0.24	-10.08	17.70	11.75	-155.69	1.89	4.31
TAN	-0.48	-11.19	15.12	11.01	-157.03	-	-
MAF-DDSF	-0.62	-11.96	15.09	8.86	-157.73	-	-

Flow matching

1. The main benefits of continuous flows are
   - Constraints are much less than in the discrete case:

     for the solution of the ODE to be unique, only needs v to be Lipschitz continuous in $\mathbf{x}$ and continuous in t.
   - Inverting the flow can be achieved by solving the ODE in reverse.
   - Computing the likelihood does not require inverting the flow, nor to compute a log determinant;

     only the trace of the Jacobian is required, that can be approximated using the Hutchinson trick. Please study it.

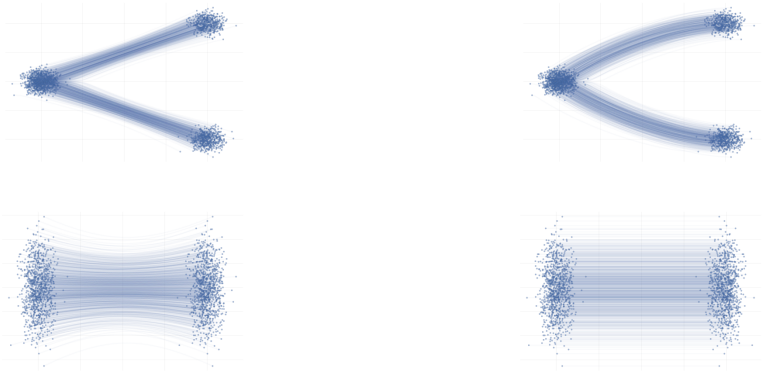
     $$\mathrm{tr}(\mathbf{A}) = \mathbb{E}_{\epsilon}[\epsilon^{\intercal}\mathbf{A}\epsilon]$$

     where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.
2. However, training a neural ODE with log-likelihood does not scale well to high-dimensional spaces, and the process tends to be unstable, likely due
   - to numerical approximations and
   - to the (infinite) number of possible probability paths.

1. Flow matching is a simulation-free way to train CNF models.
2. We directly formulate a regression objective w.r.t. $v_\theta(\mathbf{x}_t, t)$ of the form

$$\mathcal{L}_{fm}(\theta) = \mathbb{E}_{\substack{t\sim U(0,1) \\ \mathbf{x}\sim p_t(\mathbf{x})}} \left[\| v_\theta(\mathbf{x}_t, t) - v(\mathbf{x}_t, t) \|^2 \right]$$

3. This requires knowledge of a valid $v(\mathbf{x}_t, t)$ (we assume we know it!).



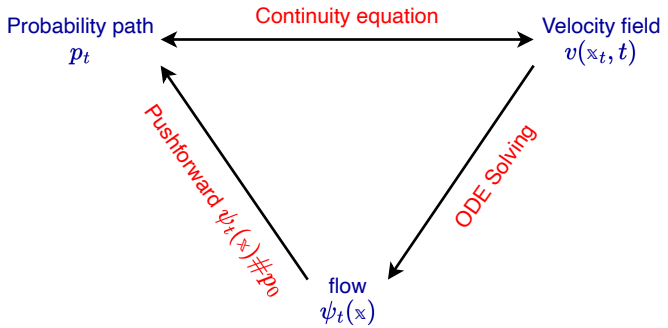
4. This objective function can not be minimized, due to inaccessibility of $v(\mathbf{x}_t, t)$, similar to the basic score matching.
5. This is where Conditional Flow Matching (CFM) comes.

Conditional Flow Matching

1. CFM was introduced by three simultaneous papers through different approaches:
   - conditional matching (Lipman, R. T. Q. Chen, et al. 2023).
   - rectifying flows (X. Liu, Gong, and Q. Liu 2023).
   - stochastic interpolants (Albergo and Vanden-Eijnden 2023).
2. The transport equation relates a vector field $v(\mathbf{x}_t, t)$ to a probability path $p_t(\mathbf{x})$ as

$$\frac{\partial\, p_t(\mathbf{x})}{\partial t} = -\nabla \cdot v(\mathbf{x}_t, t)\, p_t(\mathbf{x})$$

3. Thus, constructing $v(\mathbf{x}_t, t)$ or $p_t(\mathbf{x})$ is equivalent.



1. Let $\mathbf{z} \in \mathbb{R}^d$ be a random variable sampled from a given distribution $p_{\mathbf{z}}(\mathbf{z})$.
2. The conditional ODE becomes

$$\frac{d\mathbf{x}_t}{dt} = v(\mathbf{x}_t, t \mid \mathbf{z})$$

3. Then, the objective function becomes

$$\mathcal{L}_{cfm}(\theta) = \mathbb{E}_{\substack{t\sim U(0,1) \\ \mathbf{x}\sim p_t(\mathbf{x}),\ \mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}} \left[\| v_\theta(\mathbf{x}_t, t) - v(\mathbf{x}_t, t \mid \mathbf{z}) \|^2 \right]$$

Theorem (Lipman, R. T. Q. Chen, et al. 2023)

If for all $\mathbf{x} \in \mathbb{R}^d$, we have $p_t(\mathbf{x}) > 0$ and $t \in [0, 1]$, then $\mathcal{L}_{cfm}(\theta) = \mathcal{L}_{fm}(\theta) + c$, where c is independent of θ. Therefore, we have $\nabla_\theta \mathcal{L}_{cfm}(\theta) = \nabla_\theta \mathcal{L}_{fm}(\theta)$.

4. We can use CFM instead of FM.
5. What this conditioning $\mathbf{z}$ should be, and what is its distribution?

   There are multiple options (Tong et al. 2024).

1. How to obtain conditional distributions $p_t(\mathbf{x} \mid \mathbf{z})$?
2. The continuity equation allows us to calculate the probability path. But, we need to know the vector field.
3. How to avoid it?
   - First, consider the form of $p_t(\mathbf{x} \mid \mathbf{z})$.
   - Then, use form of $p_t(\mathbf{x} \mid \mathbf{z})$ and derive the vector field $v(\mathbf{x}, t \mid \mathbf{z})$.
4. CFM expresses probability path as a marginal over a joint involving a latent variable $\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})$:

$$p_t(\mathbf{x}_t) = \int p_{\mathbf{z}}(\mathbf{z})\, p_{t \mid \mathbf{z}}(\mathbf{x}_t \mid \mathbf{z}) \mathrm{d}\mathbf{z}$$

   - Term $p_{t \mid \mathbf{z}}(\mathbf{x}_t \mid \mathbf{z})$ is called **conditional probability path**.
   - Term $p_{t \mid \mathbf{z}}(\mathbf{x}_t \mid \mathbf{z})$ satisfies some boundary conditions at $t = 0$ and $t = 1$ such that $p_t(\mathbf{x}_t)$ be a valid path interpolating between $p_0(\mathbf{x}_0)$ and $p_{\mathrm{d}}(\mathbf{x})$.
5. Regarding $\mathbf{z}$, we can think of it as extra information like data $\mathbf{x}_1$ or anything else like a class label, a piece of text, an audio signal, or an additional image.

1. Since we have access to samples $\mathbf{x}_1 \sim p_{\mathrm{d}}(\mathbf{x})$, it is good idea to condition on $\mathbf{z} = \mathbf{x}_1$:

$$p_t(\mathbf{x}_t) = \int p_{\mathrm{d}}(\mathbf{x}_1)\, p_{t\,|\,1}(\mathbf{x}_t \mid \mathbf{x}_1) \mathrm{d}\mathbf{x}_1$$

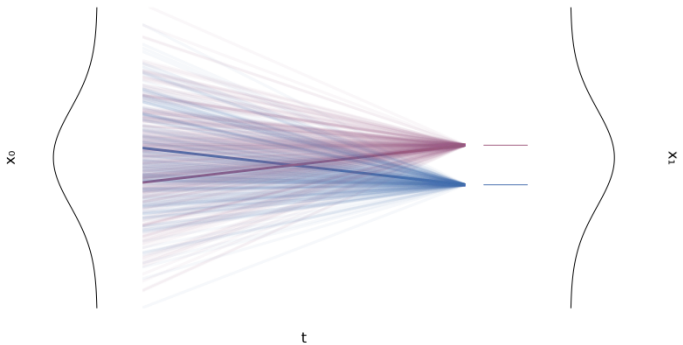
2. Conditional probability path $p_{t\,|\,1}(\mathbf{x}_t \mid \mathbf{x}_1)$ needs to satisfy the boundary conditions

$$p_0(\mathbf{x} \mid \mathbf{x}_1) = p_0(\mathbf{x})$$ reference distribution, usually $p_0(\mathbf{x}) = \mathcal{N}(0, \mathbf{I})$

$$p_1(\mathbf{x} \mid \mathbf{x}_1) = \mathcal{N}(\mathbf{x}_1, \sigma_{min}^2 \mathbf{I})$$

$$\sigma_{min} > 0$$ small value

3. Choosing reference distribution as $p_0(\mathbf{x}) = \mathcal{N}(0, \mathbf{I})$.



1. Conditional probability path satisfies transport equation with conditional vector field $v(\mathbf{x}, t \mid \mathbf{x}_1)$

$$\frac{\partial\, p_t(\mathbf{x} \mid \mathbf{x}_1)}{\partial t} = -\nabla \cdot \left(v(\mathbf{x}, t \mid \mathbf{x}_1)\, p_t(\mathbf{x} \mid \mathbf{x}_1)\right)$$

2. Lipman et al. (2023) introduced the notion of **conditional flow matching** (CFM) uses $v(\mathbf{x}, t \mid \mathbf{x}_1)$ to express marginal vector $v(\mathbf{x}, t)$ as

$$\begin{aligned} v(\mathbf{x}, t) &= \mathbb{E}_{\mathbf{x}_1 \sim p_{1 \mid t}(\mathbf{x})}[v(\mathbf{x}, t \mid \mathbf{x}_1)] \\ &= \int v(\mathbf{x}, t \mid \mathbf{x}_1) \frac{p_t(\mathbf{x} \mid \mathbf{x}_1)\, p_{\mathrm{d}}(\mathbf{x}_1)}{p_t(\mathbf{x})} \mathrm{d}\mathbf{x}_1 \end{aligned}$$

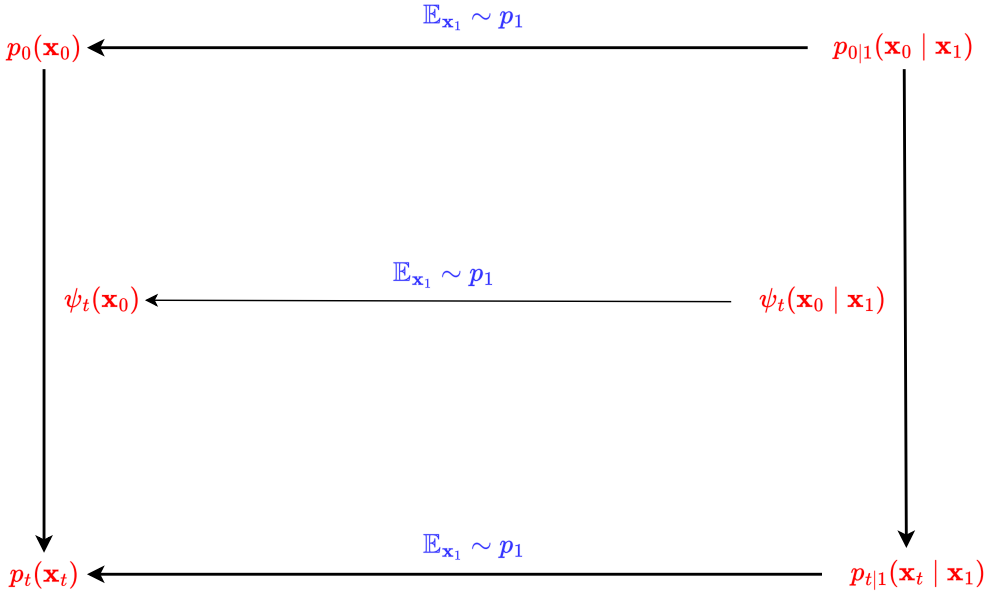
3. We need to show that the marginal vector field $v(\mathbf{x}, t)$ satisfies the **transport equation**:

$$\frac{\partial\, p_t(\mathbf{x})}{\partial t} = -\nabla \cdot \left(v(\mathbf{x}, t)\, p_t(\mathbf{x})\right)$$

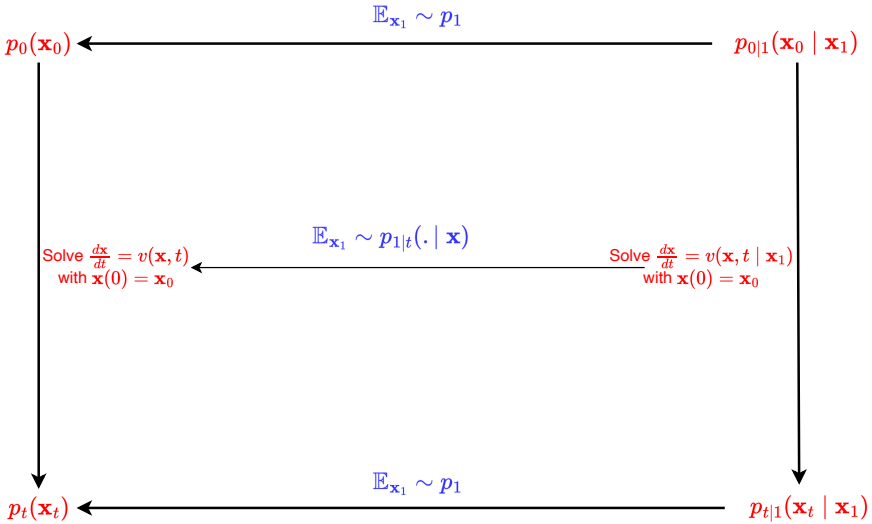
The $\frac{\partial\, p_t(\mathbf{x})}{\partial t}$ can be written as

$$
\begin{aligned}
\frac{\partial\, p_t(\mathbf{x})}{\partial t} &= \frac{\partial}{\partial t} \int p_t(\mathbf{x} \mid \mathbf{x}_1)\, p_{\mathrm{d}}(\mathbf{x}_1) \mathrm{d}\mathbf{x}_1 \\
&= \int \frac{\partial}{\partial t} (\, p_t(\mathbf{x} \mid \mathbf{x}_1))\, p_{\mathrm{d}}(\mathbf{x}_1) \mathrm{d}\mathbf{x}_1 \\
&= -\int \nabla \cdot (v(\mathbf{x}, t \mid \mathbf{x}_1)\, p_t(\mathbf{x} \mid \mathbf{x}_1))\, p_{\mathrm{d}}(\mathbf{x}_1) \mathrm{d}\mathbf{x}_1 \\
&= -\int \nabla \cdot (v(\mathbf{x}, t \mid \mathbf{x}_1)\, p_t(\mathbf{x} \mid \mathbf{x}_1)\, p_{\mathrm{d}}(\mathbf{x}_1)) \mathrm{d}\mathbf{x}_1 \\
&= -\nabla \cdot \int v(\mathbf{x}, t \mid \mathbf{x}_1)\, p_t(\mathbf{x} \mid \mathbf{x}_1)\, p_{\mathrm{d}}(\mathbf{x}_1) \mathrm{d}\mathbf{x}_1 \\
&= -\nabla \cdot \left(\int v(\mathbf{x}, t \mid \mathbf{x}_1) \frac{p_t(\mathbf{x} \mid \mathbf{x}_1)\, p_{\mathrm{d}}(\mathbf{x}_1)}{p_t(\mathbf{x})}\, p_t(\mathbf{x})\ \mathrm{d}\mathbf{x}_1 \right) \\
&= -\nabla \cdot \left(\underbrace{\int v(\mathbf{x}, t \mid \mathbf{x}_1) \frac{p_t(\mathbf{x} \mid \mathbf{x}_1)\, p_{\mathrm{d}}(\mathbf{x}_1)}{p_t(\mathbf{x})}\ \mathrm{d}\mathbf{x}_1}_{v(\mathbf{x}, t)}\ p_t(\mathbf{x}) \right) \\
&= -\nabla \cdot (v(\mathbf{x}, t)\, p_t(\mathbf{x}))
\end{aligned}
$$

The relation between $\psi_t(\mathbf{x}_0)$ and $\psi_t(\mathbf{x}_0 \mid \mathbf{x}_1)$



Relation between $v(\mathbf{x}_0, t)$ and $v(\mathbf{x}_0, t \mid \mathbf{x}_1)$

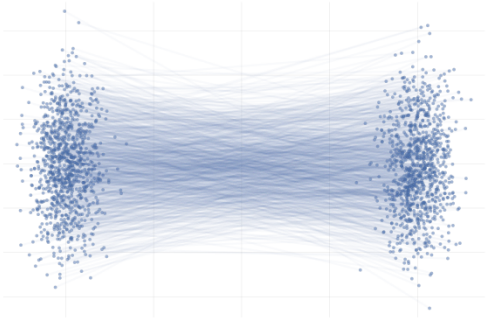


Example: Gaussian to Gaussian (2D)

1. Let $\mu = 10$ and

$$p_0(\mathbf{x}) = \mathcal{N}([-\mu, 0], \mathbf{I})$$
$$p_1(\mathbf{x}) = \mathcal{N}([+\mu, 0], \mathbf{I})$$
$$\psi_t(\mathbf{x}_0 \mid \mathbf{x}_1) = (1 - t)\mathbf{x}_0 + t\mathbf{x}_1$$

2. Example conditional paths $\psi_t(\mathbf{x}_0 \mid \mathbf{x}_1)$

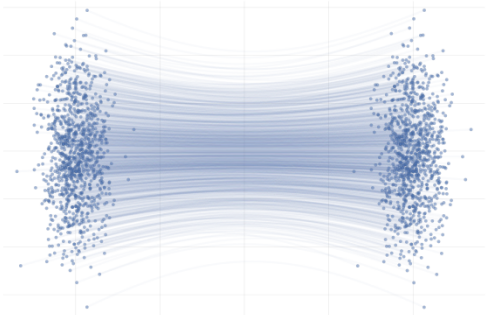


3. We are interested in learning the marginal paths $\psi_t(\mathbf{x}_0)$ for initial points $\mathbf{x}_0 \sim p_0(\mathbf{x})$.

4. Then, we use $\mathbf{x}_0$ to generate samples $\psi_1(\mathbf{x}_0)$.

Example: Gaussian to Gaussian (2D)

1. Example marginal paths $\psi_1(\mathbf{x}_0)$:



2. Pick a point $\mathbf{x}_0 \sim p_0(\mathbf{x})$, and compute a MC estimator for $v(\mathbf{x}, t)$ at different t along path $\psi_1(\mathbf{x}_0)$.

3. We look at

$$v(\psi_1(\mathbf{x}_0), t) = \mathbb{E}_{p_{1\mid t}(\mathbf{x})}[v(\psi_1(\mathbf{x}_0), t \mid \mathbf{x}_1)]$$
$$\approx \frac{1}{n}\sum_{i=1}^{n} v(\psi_1(\mathbf{x}_0), t \mid \mathbf{x}_1^{(i)}) \qquad \text{with} \qquad \mathbf{x}_1^{(i)} \sim p_{1\mid t}(\mathbf{x}_1 \mid \psi_1(\mathbf{x}_0))$$

4. In practice we don not have access to the posterior $p_{1\mid t}(\mathbf{x}_1 \mid \psi_1(\mathbf{x}_0))$.

1. We had

$$
\begin{aligned}
v(\mathbf{x}, t) &= \mathbb{E}_{\mathbf{x}_1 \sim p_{1 \mid t}(\mathbf{x})}[v(\mathbf{x}, t \mid \mathbf{x}_1)] \\
&= \int v(\mathbf{x}, t \mid \mathbf{x}_1) \frac{p_t(\mathbf{x} \mid \mathbf{x}_1)\, p_{\mathrm{d}}(\mathbf{x}_1)}{p_t(\mathbf{x})} \mathrm{d}\mathbf{x}_1
\end{aligned}
$$

2. Now consider the loss of flow matching as

$$
\mathcal{L}_{fm}(\theta) = \mathbb{E}_{\substack{t \sim U(0,1) \\ \mathbf{x} \sim p_t(\mathbf{x})}} \left[\| v_\theta(\mathbf{x}_t, t) - v(\mathbf{x}_t, t) \|^2 \right]
$$

3. Using $v(\mathbf{x}, t) = \mathbb{E}_{\mathbf{x}_1 \sim p_{1 \mid t}(\mathbf{x})}[v(\mathbf{x}, t \mid \mathbf{x}_1)]$, we obtain

$$
\mathcal{L}_{cfm}(\theta) = \mathbb{E}_{\substack{t \sim U(0,1) \\ \mathbf{x} \sim p_t(\mathbf{x}),\ \mathbf{x}_1 \sim p_{\mathrm{d}}(\mathbf{x}_1)}} \left[\| v_\theta(\mathbf{x}_t, t) - v(\mathbf{x}_t, t \mid \mathbf{x}_1) \|^2 \right]
$$

4. This implies that we can use $\mathcal{L}_{cfm}(\theta)$ for training parametric vector field $v_\theta(\mathbf{x}, t)$

1. Consider a practical example of conditional vector field and corresponding probability path.
2. Suppose we want conditional vector field which generates a path of Gaussian, i.e.

$$p_t(\mathbf{x} \mid \mathbf{x}_1) = \mathcal{N}(\mu_t(\mathbf{x}_1), \sigma_t^2(\mathbf{x}_1)\mathbf{I})$$

   for some mean $(\mu_t(\mathbf{x}_1)$ and standard deviation $\sigma_t(\mathbf{x}_1)$.
3. **In general, there is no unique ODE that generates these distributions.**
4. **However, the following theorem shows that there is a unique vector field that leads to those!**

Theorem (Lipman, R. T. Q. Chen, et al. 2023)

The unique vector field with initial conditions $p_0(\mathbf{x}) = \mathcal{N}(\mu_0, \sigma_0^2\mathbf{I})$ *that generates* $p_t(\mathbf{x} \mid \mathbf{x}_1) = \mathcal{N}(\mu_t(\mathbf{x}_1), \sigma_t^2(\mathbf{x}_1)\mathbf{I})$ *has the following form:*

$$v(\mathbf{x}, t \mid \mathbf{x}_1) = \frac{\sigma_t'(\mathbf{x}_1)}{\sigma_t(\mathbf{x}_1)}(\mathbf{x} - \mu_t(\mathbf{x}_1)) + \mu_t'(\mathbf{x}_1)$$

where

- $\mu_t'(\mathbf{x}_1)$ *denote the time derivate of* $\mu_t(\mathbf{x}_1)$.
- $\sigma_t'(\mathbf{x}_1)$ *denote the time derivate of* $\sigma_t'(\mathbf{x}_1)$.

Result: If we consider a class of conditional probability paths in the form of Gaussian, we can analytically calculate the conditional vector field as long as the means and the standard deviations are differentiable.

Proof of theorem (Lipman, R. T. Q. Chen, et al. 2023).

1. Let

$$\psi_t(\mathbf{x} \mid \mathbf{x}_1) = \mu_t(\mathbf{x}_1) + \sigma_t(\mathbf{x}_1)\mathbf{x}$$

2. We want to determine $v(\mathbf{x} \mid \mathbf{x}_1)$ such that

$$\frac{d}{dt}\psi_t(\mathbf{x}) = v(\psi_t(\mathbf{x}), t \mid \mathbf{x}_1)$$

3. The left hand side is

$$\begin{aligned}\frac{d}{dt}\psi_t(\mathbf{x}) &= \frac{d}{dt}(\mu_t(\mathbf{x}_1) + \sigma_t(\mathbf{x}_1)\mathbf{x}) \\ &= \frac{d\mu_t(\mathbf{x}_1)}{dt} + \frac{d\sigma_t(\mathbf{x}_1)}{dt}\mathbf{x} \\ &= \mu_t'(\mathbf{x}_1) + \sigma_t'(\mathbf{x}_1)\mathbf{x}\end{aligned}$$

1. Thus, we obtain

$$\mu_t'(\mathbf{x}_1) + \sigma_t'(\mathbf{x}_1)\mathbf{x} = v(\psi_t(\mathbf{x} \mid \mathbf{x}_1), t \mid \mathbf{x}_1)$$

2. Suppose that $v(\psi_t(\mathbf{x} \mid \mathbf{x}_1), t \mid \mathbf{x}_1)$ is of the form

$$v(\psi_t(\mathbf{x} \mid \mathbf{x}_1), t \mid \mathbf{x}_1) = h(t, \psi_t(\mathbf{x}), \mathbf{x}_1)\mu_t'(\mathbf{x}_1) + g(t, \psi_t(\mathbf{x}), \mathbf{x}_1)\sigma_t'(\mathbf{x}_1)$$

for some functions h and g.

3. In the previous equation, we had

$$h(t, \psi_t(\mathbf{x}), \mathbf{x}_1) = 1 \qquad\qquad g(t, \psi_t(\mathbf{x}), \mathbf{x}_1) = \mathbf{x}$$

4. The simplest solution to this equation is

$$h(t, \mathbf{x}, \mathbf{x}_1) = 1 \qquad\qquad g(t, \mathbf{x}, \mathbf{x}_1) = {\psi_t}^{-1}(\mathbf{x}) = \frac{\mathbf{x} - \mu_t(\mathbf{x}_1)}{\sigma_t(\mathbf{x}_1)}$$

such that $g(t, \psi_t(\mathbf{x}), \mathbf{x}_1) = {\psi_t}^{-1}(\psi_t(\mathbf{x})) = \mathbf{x}$, resulting in

$$v(\mathbf{x}, t \mid \mathbf{x}_1) = \frac{\sigma_t'(\mathbf{x}_1)}{\sigma_t(\mathbf{x}_1)}(\mathbf{x} - \mu_t(\mathbf{x}_1)) + \mu_t'(\mathbf{x}_1)$$

Example: Linear interpolation

1. A simple choice for the mean $\mu_t(\mathbf{x}_1)$ and standard deviation $\sigma_t(\mathbf{x}_1)$ is the linear interpolation as:

$$\mu_t(\mathbf{x}_1) \triangleq t\mathbf{x}_1 \qquad \mu'_t(\mathbf{x}_1) = \mathbf{x}_1$$
$$\sigma_t(\mathbf{x}_1) \triangleq (1-t) + t\sigma_{min} \qquad \sigma'_t(\mathbf{x}_1) = -1 + \sigma_{min}$$

such that

$$(\mu_0(\mathbf{x}_1) + \sigma_0(\mathbf{x}_1)\mathbf{x}_1) \sim p_0(\mathbf{x})$$
$$(\mu_1(\mathbf{x}_1) + \sigma_1(\mathbf{x}_1)\mathbf{x}_1) \sim p_1(\mathbf{x}) = \mathcal{N}(\mathbf{x}_1, \sigma^2_{min}\mathbf{I})$$

2. Also for some $\mu > 0$, let

$$p_0(\mathbf{x}) = \mathcal{N}([-\mu, 0], \mathbf{I})$$
$$p_1(\mathbf{x}) = \mathcal{N}([+\mu, 0], \mathbf{I})$$
$$\psi_t(\mathbf{x}_0 \mid \mathbf{x}_1) = (1-t)\mathbf{x}_0 + t\mathbf{x}_1$$

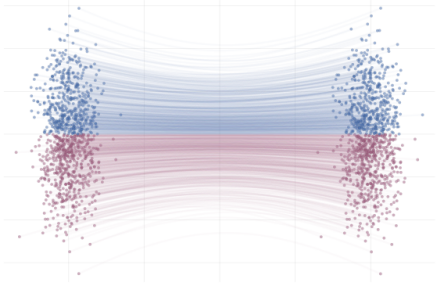
Example: Linear interpolation

1. Writing $\sigma_t(\mathbf{x}_1)$ as $\sigma_t(\mathbf{x}_1) = 1 - (1 - \sigma_{min})$, the conditional vector field becomes as

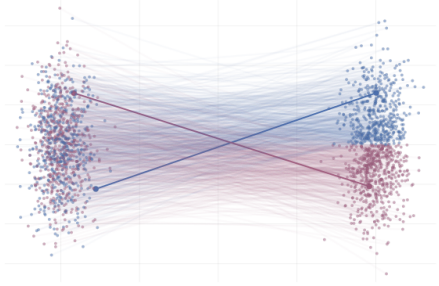
$$
\begin{aligned}
v(\mathbf{x}, t \mid \mathbf{x}_1) &= \frac{-(1 - \sigma_{min})}{1 - (1 - \sigma_{min})t}(\mathbf{x} - t\mathbf{x}_1) + \mathbf{x}_1 \\
&= \frac{1}{(1 - t) + t\sigma_{min}}\big[- (1 - \sigma_{min})(\mathbf{x} - t\mathbf{x}_1) + (1 - (1 - \sigma_{min})t)\mathbf{x}_1\big] \\
&= \frac{1}{(1 - t) + t\sigma_{min}}\big[- (1 - \sigma_{min})\mathbf{x} + \mathbf{x}_1\big] \\
&= \frac{\mathbf{x}_1 - (1 - \sigma_{min})\mathbf{x}}{1 - (1 - \sigma_{min})t}
\end{aligned}
$$

2. Example paths from $p_0(\mathbf{x})$ to $p_1(\mathbf{x})$ following

the true vector field $v(\mathbf{x}, t)$

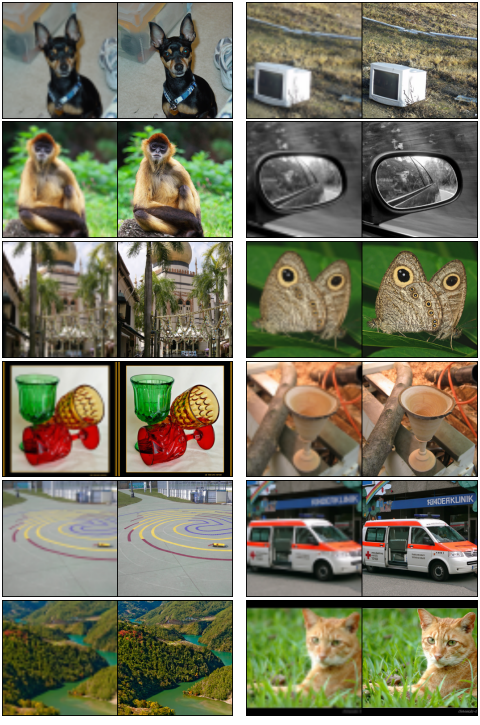


the conditional vector field $v(\mathbf{x}, t \mid \mathbf{x}_1)$



Conditional Flow Matching Results

The results from (Lipman, R. T. Q. Chen, et al. 2023)



Training/Sampling Conditional Flow Matching

Training:

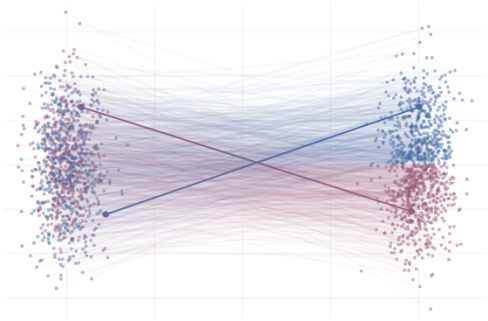
1. Sample $t \sim U(0,1)$ and $\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})$.
2. Calculate $\mu_t(\mathbf{z})$ and $\sigma_t(\mathbf{z})$
3. Sample from $\mathbf{x}_t \sim \mathcal{N}(\mu_t(\mathbf{z}), \sigma_t^2(\mathbf{z})\mathbf{I})$
4. Calculate the vector field $v(\mathbf{x}_t, t \mid \mathbf{z})$
5. Calculate $\nabla_\theta \mathcal{L}_{cfm}(\theta)$ and update θ.

Sampling:

1. Sample $\mathbf{x}_0 \sim p_0(\mathbf{x})$.
2. Run forward Euler method from $t = 0$ to $t = 1$ with step size h

$$\mathbf{x}_{t+h} = \mathbf{x}_t + h \times v_\theta(\mathbf{x}_t, t)$$

There are two issues arising from crossing conditional paths.



1. ODE hard to integrate and slow sampling at inference
2. Consider $v(\mathbf{x}_t, t \mid \mathbf{x}_1)$ with two data samples $\mathbf{x}_1^{(1)}$ and $\mathbf{x}_1^{(2)}$.
3. SGD approximates the CFM loss as:

$$\mathcal{L}_{cfm}(\theta) \approx \frac{1}{2}\|v_\theta(\mathbf{x}_t^{(1)}, t) - v(\mathbf{x}_t^{(1)}, t, \mid \mathbf{x}_t^{(1)})\|^2 + \frac{1}{2}\|v_\theta(\mathbf{x}_t^{(2)}, t) - v(\mathbf{x}_t^{(2)}, t, \mid \mathbf{x}_t^{(2)})\|^2$$

4. We are attempting to align $v_\theta(\mathbf{x}_t, t)$ with two different vector fields.
5. This can lead to increased variance in the gradient estimate, and thus slower convergence.

References

Reading

1. Chapter 9 of Deep Generative Modeling (Tomczak 2024).
2. Flow Matching for Generative Modeling (Lipman, R. T. Q. Chen, et al. 2023).
3. Improving and generalizing flow-based generative models with minibatch optimal transport (Tong et al. 2024).
4. Introduction to Flow Matching and Diffusion Models (Holderrieth and Erives 2025).
5. Flow Matching Guide and Code (Lipman, Havasi, et al. 2024).
6. An Introduction to Flow Matching (Fjelde, Mathieu, and Dutordoir 2024).

References i

- Albergo, Michael S. and Eric Vanden-Eijnden (2023). "Building Normalizing Flows with Stochastic Interpolants". In: *International Conference on Learning Representations*.
- Chen, Tian Qi et al. (2018). "Neural Ordinary Differential Equations". In: *Advances in Neural Information Processing Systems*, pp. 6572–6583.
- Fjelde, Tor, Emile Mathieu, and Vincent Dutordoir (2024). *An Introduction to Flow Matching*. URL: https://mlg.eng.cam.ac.uk/blog/2024/01/20/flow-matching.html.
- Grathwohl, Will et al. (2019). "FFJORD: Free-Form Continuous Dynamics for Scalable Reversible Generative Models". In: *International Conference on Learning Representations*.
- Holderrieth, Peter and Ezra Erives (2025). *Introduction to Flow Matching and Diffusion Models*. URL: https://diffusion.csail.mit.edu/.
- Lipman, Yaron, Ricky T. Q. Chen, et al. (2023). "Flow Matching for Generative Modeling". In: *International Conference on Learning Representations*.
- Lipman, Yaron, Marton Havasi, et al. (2024). *Flow Matching Guide and Code*. arXiv: 2412.06264 [cs.LG]. URL: https://arxiv.org/abs/2412.06264.
- Liu, Xingchao, Chengyue Gong, and Qiang Liu (2023). "Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow". In: *International Conference on Learning Representations*.
- Rezende, Danilo Jimenez and Shakir Mohamed (2015). "Variational Inference with Normalizing Flows". In: *International Conference on Machine Learning*. Vol. 37, pp. 1530–1538.

References ii

- Tomczak, Jakub M. (2024). *Deep Generative Modeling*. Springer.
- Tong, Alexander et al. (2024). "Improving and generalizing flow-based generative models with minibatch optimal transport". In: *Transactions on Machine Learning Research* 2024.

Questions?