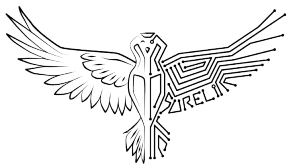
Diffusion Processes

Paul-Antoine Le Tolguenec[1], Maud Biquard[1]

[1]ISAE-Supaero



Summary

- The problem addressed by diffusion models
- Score-based methods
- Flow Matching methods

Notations

- $\{X_t\}_{t\in[0,T]}$: Probability path X_0 and X_T
- $p_t = p_{X_t}$: Marginal distribution of X_t at time t
- $s_t(x) = \nabla_x \log p_t(x)$: Score (Stein) function at time t
- $p_\theta(x) = \frac{e^{-f_\theta(x)}}{Z_\theta}$: parametrization of a density model
- $f_\theta(x)$: energy-based model (unnormalized probabilistic model)

Introduction: Generative Modeling Approaches

Likelihood-based	**Implicit**	**Score-based**
• Learn $p(x)$ directly • ARs, Normalizing Flows, VAEs • Limited by tractability (normalizing constant)	• Model sampling process • GANs • Training instability	• Model $\nabla_x \log p(x)$ • Foundation for diffusion • No normalizing constant

Score-Based Generative Models (Intuition)

- Model the score function $\nabla_x \log p(x)$ instead of density $p(x)$
- **Key advantage**: No tractable normalizing constant needed

$$p_\theta(x) = \frac{e^{-f_\theta(x)}}{Z(\theta)} \quad \text{vs.} \quad s_\theta(x) = \nabla_x \log(p_\theta)(x) = -\nabla_x f_\theta(x)$$

- Train by minimizing Fisher divergence through score matching:

$$\mathbb{E}_{p_{\text{data}}(x)}[\|s_\theta(x) - \nabla_x \log p_{\text{data}}(x)\|^2]$$

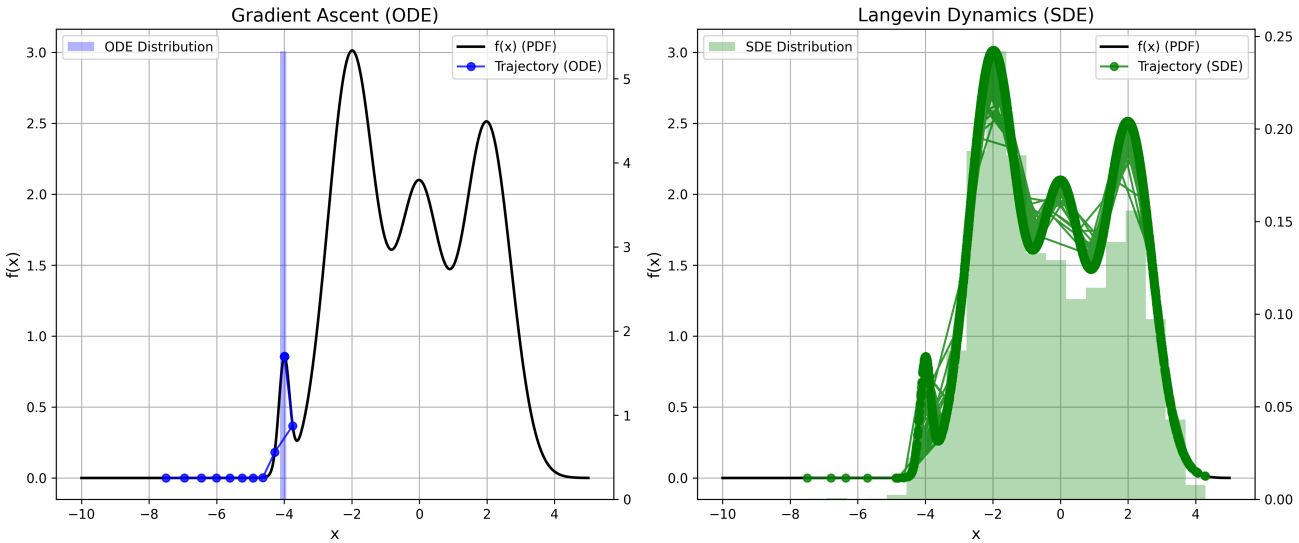
Integration of Langevin Dynamics (Intuition)

Langevin Dynamics Integration Scheme (MCMC)

The stochastic differential equation for Langevin dynamics:

$$x_{t+\Delta t} = x_t + \nabla_x \log p(x_t)\Delta t + \sqrt{2\Delta t}\,\epsilon_t \tag{1}$$

where $\epsilon_t \sim \mathcal{N}(0, I)$ is normally distributed noise.



Aapo Hyvärinen [2005](Intuition)

The gradient of $\log p_{\text{data}}(x)$:

$$\nabla_x \log p_{\text{data}}(x) = \left[\frac{\partial \log p_{\text{data}}}{\partial x_1}(x), \dots, \frac{\partial \log p_{\text{data}}}{\partial x_i}(x), \dots, \frac{\partial \log p_{\text{data}}}{\partial x_n}(x) \right]^\top.$$

$$s_\theta(x) = \left[s_\theta^1(x), \dots, s_\theta^i(x), \dots, s_\theta^n(x) \right]^\top.$$

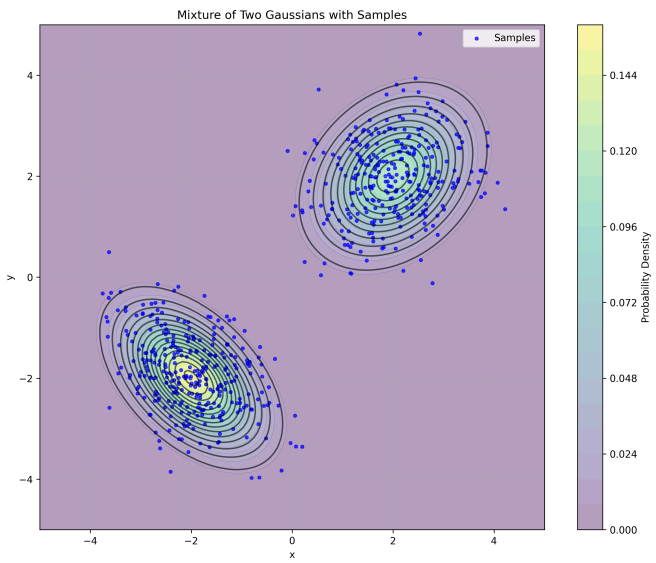
We can minimize the Fisher Divergence without de ground truth score:

$$\begin{aligned}
\mathcal{L}(\theta) &= \frac{1}{2} \mathbb{E}_{p_{\text{data}}(x)} [\| s_\theta(x) - \nabla_x \log p_{\text{data}}(x) \|^2] \\
&= \frac{1}{2} \int_{\mathcal{X}} p_{\text{data}}(x) \| s_\theta(x) - \nabla_x \log p_{\text{data}}(x) \|^2 dx \\
&= \int_{\mathcal{X}} p_{\text{data}}(x) \sum_{i=1}^{n} \left(\partial_i s_\theta^i(x) + \frac{1}{2} s_\theta^i(x)^2 \right) dx + C.
\end{aligned}$$

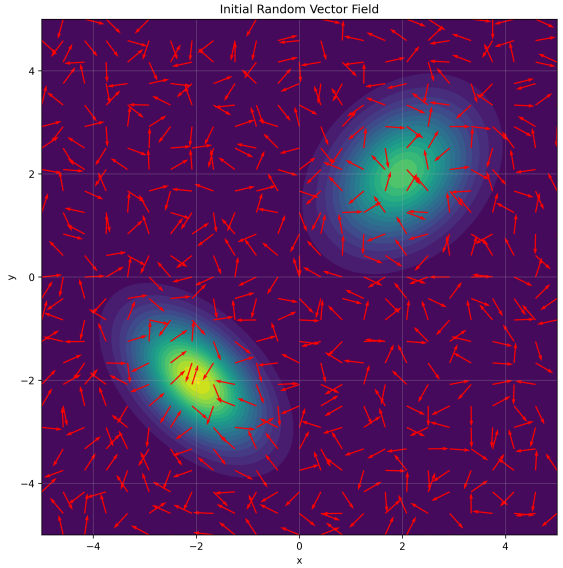
Here $\partial_i s_\theta^i(x)$ estimate the elements of the Laplacian vector of $\log p_{\text{data}}(x)$

Using Aapo's Theorem in a 2D Case (Intuition)

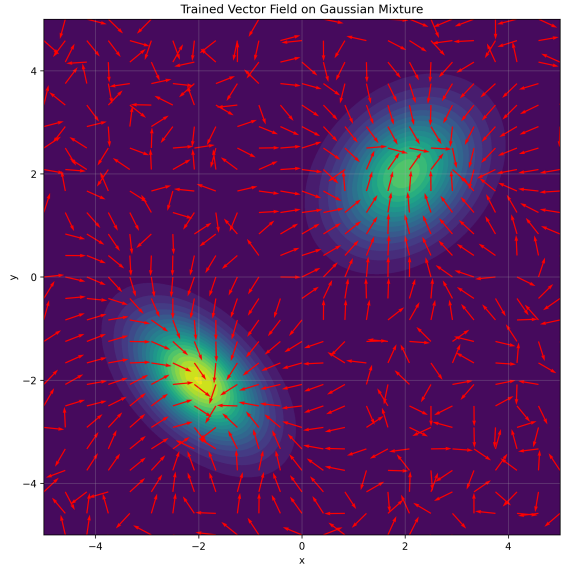
Let us illustrate Aapo's theorem on a 2-dimensional example. Below are three visualizations:



Gaussian Mixture



Initial Vector Field



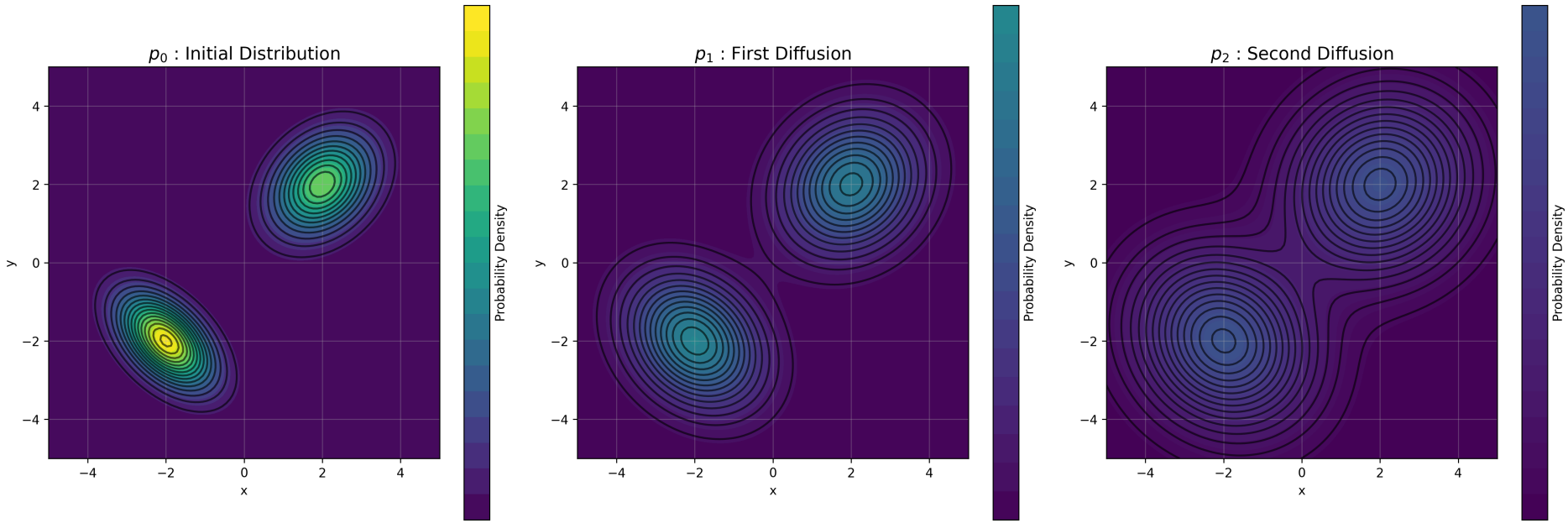
Trained Vector Field

Problem! We would like to sample "any point" from $\mathcal{X}$ and use Langevin dynamic to end up in high areas of $\log p_{\text{data}}$.

The noise trick (Intuition)

The idea of the noise trick is to define a series $\{X_i\}_{[0,n]}$ of n random variables (with $X_0 \sim p_{\text{data}}$ of increasing variance $\sigma_{i+1} > \sigma_i$. In order to sample from the target distribution, we then need to solve n Langevin dynamics processes to return to a sample of the previous random variable $X_i \xrightarrow{\text{LD}} X_{i-1}$ (Well... That's what I thought). We learn a serie of scores:

$$\mathcal{L}(\theta) = \sum_{i=0}^{n} \frac{1}{2} \mathbb{E}_{p_i(x)}[\|s_\theta(x, i) - \nabla_x \log p_i(x)\|^2]$$



Stochastic Differential Equations (SDEs)

General Form: Continuous-time Markov chain

$$dX_t = f(X_t, t)dt + \sigma(X_t, t)dW_t$$

Where:

- X_t: stochastic process
- $f(X_t, t)$: drift term
- $\sigma(X_t, t)$: diffusion term
- W_t: standard Brownian motion

Key Examples

- **Ornstein-Uhlenbeck Process**
- **Langevin Dynamics**
- **Continuous SGD**

Denoising Diffusion Probabilistic Models (DDPM)

Forward Diffusion Process (Adding Noise)

$$\begin{cases} dX_t = -\frac{1}{2}\beta(t)X_t\,dt + \sqrt{\beta(t)}\,dW_t \text{ Variance preserving (VP)} \\ dX_t = \sqrt{\frac{d\sigma^2}{dt}(t)}\,dW_t \text{ Variance exploding (VE)} \end{cases} \quad \text{with } X_0 \sim p_{\text{data}}$$

These equations are design choices. Their designs should be considered at the same level as the model design itself.

- $\beta(t) > 0$ is the noise variance function (schedule)
- W_t is a standard Brownian motion

Step 1: Closed form for X_t (Bert Oksendal[2000])

$$dX_t = -\frac{1}{2}\beta(t)X_t dt + \sqrt{\beta(t)}\, dW_t$$

$$\implies I(t)dX_t \underbrace{\frac{1}{2}\beta(t)I(t)}_{\frac{dI}{dt}(t)} X_t dt = I(t)\sqrt{\beta(t)}\, dW_t \quad \text{with} \quad \underbrace{I(t) = \exp\left(\int_0^t \frac{1}{2}\beta(s)ds\right)}_{\text{Integrating Factor}}$$

$$\implies I(t)dX_t + dIX_t = I(t)\sqrt{\beta(t)}\, dW_t$$

$$\implies d\left(IX_t\right) = I(t)\sqrt{\beta(t)}\, dW_t$$

Let's integrate between 0 and t

$$\int_0^t d\left(IX_s\right) = \int_0^t \sqrt{\beta(s)}\, dW_s$$

$$X_t \cdot \exp\left(\frac{1}{2}\int_0^t \beta(s)\, ds\right) = X_0 + \int_0^t \sqrt{\beta(s)} \exp\left(\int_0^s \frac{1}{2}\beta(u)du\right) dW_s$$

$$X_t = X_0 \cdot \exp\left(-\frac{1}{2}\int_0^t \beta(s)\, ds\right) + \int_0^t \sqrt{\beta(s)} \cdot \exp\left(-\frac{1}{2}\int_s^t \beta(u)\, du\right) dW_s$$

Step 1: Closed form for X_t

Stochastic process X_t

$$X_t = \underbrace{X_0 \cdot \exp\left(-\frac{1}{2}\int_0^t \beta(s)\,ds\right)}_{\text{Deterministic after first sample}} + \underbrace{\int_0^t \sqrt{\beta(s)} \cdot \exp\left(-\frac{1}{2}\int_s^t \beta(u)\,du\right) dW_s}_{\text{Stochastic term}} \tag{2}$$

Key result: (Sample without the need to integrate.)

For $x_0 \sim X_0$ and $x_t \sim X_t$, we have :

$$p(x_t|x_0) = \mathcal{N}\left(\mu(x_0,t), \sigma_t^2\right) \tag{3}$$

where:

$$\mu(x_0,t) = x_0 \cdot \exp\left(-\frac{1}{2}\int_0^t \beta(s)\,ds\right) \quad \text{Use "le bon sens"} \tag{4}$$

$$\sigma_t^2 = \int_0^t \beta(s) \cdot \exp\left(-\int_s^t \beta(u)\,du\right) ds \quad \text{Use Itô isometry} \tag{5}$$

Step 2: Estimate $\nabla_{x_t} \log p_{X_t}$ (M.C.K. Tweedie[1940])

The optimization performed in DDPM

$$\mathcal{L}(\theta) = \mathop{\mathbb{E}}_{\substack{x_0 \sim X_0 \\ t \sim U([0,1] \\ \varepsilon \sim \mathcal{N}(0,I)}} \left[\| \varepsilon - s_\theta(\underbrace{\mu(x_0,t) + \sigma_t^2 \varepsilon}_{x_t}, t) \|_2 \right]$$

Tweedie formula

Let X be a random variable and Y a noisy observation defined as:

$$Y = X + \varepsilon * \sigma^2, \quad \varepsilon \sim \mathcal{N}(0, I).$$

The goal is to estimate the conditional expectation $\mathbb{E}[X \mid Y = y]$. Tweedie's formula states:

$$\mathbb{E}[X \mid Y = y] = y + \sigma^2 \frac{\mathrm{d}}{\mathrm{d}y} \log p_Y(y),$$

Step 2: Estimate $\nabla_{x_t} \log p_{X_t}$ (M.C.K. Tweedie[1940])

Tweedie formula in our case

Let $\mu(X_0, t)$ be a random variable and X_t a noisy observation defined as:

$$X_t = \mu(X_0, t) + \varepsilon * \sigma_t^2, \quad \varepsilon \sim \mathcal{N}(0, I).$$

The goal is to estimate the conditional expectation $\mathbb{E}[\mu(X_0, t) \mid X_t = y]$. Tweedie's formula states:

$$\nabla_{x_t} \log p_{X_t} = \frac{\mathop{\mathbb{E}}_{x_0 \sim X_0}[\mu(x_0, t) \mid X_t = x_t] - x_t}{\sigma_t^2} = \underbrace{\mathop{\mathbb{E}}_{x_0 \sim X_0}\left[\frac{\overbrace{\mu(x_0, t) - x_t}^{-\varepsilon}}{\sigma_t^2} \mid X_t = x_t\right]}_{s_\theta(x_t)}$$

Using Aapo objectif (Laplacian) is more expensive (Song [2019]). Computing the Laplacian is d times the cost of the gradient cost. (However, this needs to compute expectation for each x_t.)

Step 2: Estimate $\nabla_{x_t} \log p_{X_t}$ (M.C.K. Tweedie[1940])

Different objectives

$$\begin{cases} \mathcal{L}_{\text{Logical}}(\theta) = \underset{\substack{t\sim U([0,1]) \\ x_t \sim X_t}}{\mathbb{E}} \left[\| \underset{x_0 \sim p_{\text{data}}}{\mathbb{E}} [q(x_0|x_t)] - s_\theta(x_t, t) \|_2 \right], \\ \mathcal{L}_{\text{Practical}}(\theta) = \underset{\substack{x_0 \sim p_{\text{data}} \\ t\sim U([0,1]) \\ x_t \sim X_t}}{\mathbb{E}} [\| q(x_0|x_t) - s_\theta(x_t, t) \|_2]. \end{cases}$$

$$\mathcal{L}_{\text{Logical}}(\theta) \neq \mathcal{L}_{\text{Pratical}}(\theta)$$

Same gradients (For Bregman divergence)

$$\nabla \mathcal{L}_{\text{Logical}}(\theta) = \nabla \mathcal{L}_{\text{Pratical}}(\theta)$$

Step 3: Go back (Naïve approach)

Jump from random variable to random variable using LD

Sampling from a specific random variable:

Let say, I want to sample from a specific distribution p^* and make the assumption that for the following SDE:

$$dX_s = a(s)\,ds + b(s)\,dW_s,$$

I can find $a(s)$ and $b(s)$ such that $\lim_{s\to\infty} p_{X_s} = p*$

Fokker-Planck equation (1-dimension):

$$\frac{\partial P(x,s)}{\partial s} = -\frac{\partial}{\partial x}\left[a(s)P(x,s)\right] + \frac{\partial^2}{\partial x^2}\left[\frac{1}{2}b(s)b(s)^T P(x,s)\right]$$

Here, $P(x,s) = p_{X_s}(x)$ the probability measure of the random variable X_s in x.

Step 3: Naive approach

We need to find $a(s)$ and $b(s)$ such that when $\frac{\partial P(x,s)}{\partial s} = 0$, we have $P = p^*$.

$$\underbrace{\frac{\partial P(x,s)}{\partial s}}_{=0 \text{ At equilibrium}} = -\frac{\partial}{\partial x}\left[a(s)P(x,s)\right] + \frac{\partial^2}{\partial x^2}\left[\frac{1}{2}b(s)b(s)^T P(x,s)\right]$$

$$\frac{\partial}{\partial x}\left[a(s)p^*(x)\right] = \frac{\partial^2}{\partial x^2}\left[\frac{1}{2}b(s)b(s)^T p^*(x)\right]$$

We suppose $b(s)b(s)^T = 2I$:

$$a(s)p^*(x) = \nabla_x p^* + \underbrace{C}_{=0}$$

$$a(s) = \frac{\nabla_x p^*}{p^*} = \nabla_x \log(p^*)$$

Sampling from p^*

$$\underbrace{dX_s = \nabla_{x_s} \log(p^*) + \sqrt{2}dW_s}_{\text{LD}(X_0, p^*, s) \mapsto X_s}$$

Naïve reverse process

$$X_u \overset{\text{LD}(X_u, p_{X_{u+du}}, h)}{\underset{\text{Transition model}}{\longrightarrow}} X_{u+du}$$

Where u is the reverse time and h the simulated time.

Step 3: Anderson[1982]

The reverse process is a specific SDE

$$\begin{cases} dX_t = f(X_t, t)dt + g(t)dW_t & \text{Forward} \\ dX_t = \left[f(X_t, t) - g^2(t)\nabla \log(p_{X_t})\right] dt + g(t)d\bar{W}_t & \text{Reverse} \end{cases}$$

The reverse SDE for VP

$$dX_t = \left[-\frac{1}{2}X_t - \nabla \log(p_{X_t})\right] \beta(t)dt + \beta(t)d\bar{W}_t$$

Score-based summary

Summary

- Closed form for X_t (Because no need to integrate to sample from X_t)
- Learn $\nabla_x \log P_{X_t}$ (Tweedie or Hyvärinen)
- Use the reverse SDE (Anderson[1982]) to go from $X_{t_{\text{final}}}$ to X_0

Question to answer

- Hyvärinen vs Tweedie ?

Flow Matching (Distributions are "Pâte à modeler")

Change of variable

Let X and Y be two random variables such that $Y = \psi(X)$ with $\psi : \mathbb{R}^d \to \mathbb{R}^d$ a $\mathcal{C}^r$ diffeomorphism (endomorphism + bijection + $\psi^{-1} \in \mathcal{C}^1$), we can write p_Y given p_X as:

$$p_Y(y) = p_X(\psi^{-1}(y))|\det \partial_y \psi^{-1}(y)|$$

Normalizing flows: approximate the pdf

Given m data points $\{x_k\}_{k\in[0,m]}$ drawn from p_{data}, start from Gaussian $p_{X_1} = \mathcal{N}(0,1)$, apply n 'discrete' parametrized transformations $\theta = \{\theta_i\}_{i\in[1,n]}$ such that $X_{n+1} = f_{\theta_n} \circ \cdots \circ f_{\theta_1}(X_1)$ and minimize:

$$\mathcal{L}(\theta) = \sum_{k=0}^{m} -\log\left(p^{\theta}_{X_{n+1}}(x_k)\right)$$

After optimization, we can draw samples from p_{data} by sampling from p_{X_1} and apply the transformation series to the sample.

Flows using vector field

Continuous-time flow definition

A C^r flow ψ_t can be defined in terms of a $C^r([0,1] \times \mathbb{R}^d, \mathbb{R}^d)$ velocity field $u : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ implementing $u : (t,x) \mapsto u_t(x)$ via the following ODE:

$$\frac{d}{dt}\psi_t(x) = u_t(\psi_t(x)) \qquad \text{(flow ODE)}$$

$$\psi_0(x) = x \qquad \text{(flow initial conditions)}$$

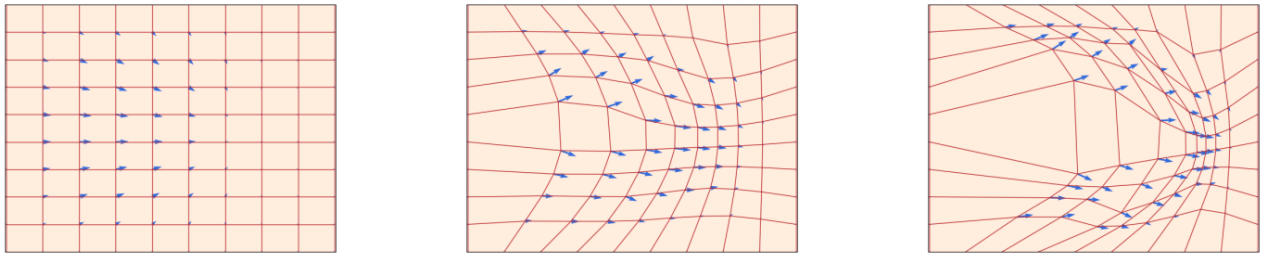
Coddington[1956] shows the existence and unicity of the solution.
In short, you can define a flow by the series of vector fields that shaped it.
In practice, we learn u_θ^t

Probability path

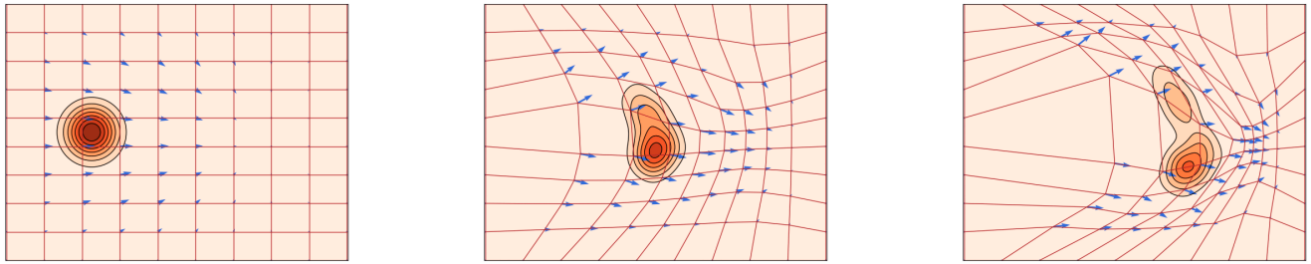
A time-dependent probability $(p_t)_{0 \le t \le 1}$ is a probability path. So, the marginal PDF of a flow model $X_t = \psi_t(X_0)$ at time t:

$$X_t \sim p_t$$

Continuous-time flow visualization



Flow defined as a sum of vector fields



Distribution transformed by the flow

Training with simulation

The Continuity Equation (Villani[2009])

$$\frac{d}{dt}p_t(x) + \text{div}(p_t u_t)(x) = 0$$

The Instantaneous Change of Variable (Chen et al.[2018])

$$\frac{d}{dt}\log p_t(\psi_t(x)) = -\text{div}(u_t)(\psi_t(x)) \implies \log p_1(\psi_1(x)) = \log p_0(\psi_0(x)) + \int_0^1 -\text{div}(u_t)(\psi_t(x))dt$$

Train with simulation:

We can simulate the ODE to obtain p_1^θ:

$$\begin{cases} \psi_{t+dt} = \psi_t + dt \cdot u_\theta(\psi_t) \\ \text{Initial conditions: } \psi_0(x) = x \text{ and } p_0 = \mathcal{N}(0, I) \end{cases} \qquad \underbrace{\mathcal{L}(\theta) = -\mathop{\mathbb{E}}_{x\sim p_{\text{data}}}\left[\log p_1^\theta(x)\right]}_{\text{We learn } u_\theta(t)}$$

Simulation free flow matching

Hard Objective: Searching for Ψ, $u_{t\in[0,1]}$ or $p_{X_{t\in[0,1]}}$

$$\psi* = \arg\min_{\psi} \int_{\mathcal{X}} |p_{\underbrace{\psi(X_0)}_{X_1}}(x) - p_{\text{data}}(x)|dx$$

$$= \left(\psi_0 + \int_0^1 u_t(\psi_t)dt\right)^* = \arg\min_{u_{t\in[0,1]}} \int_{\mathcal{X}} |p_{\left(\psi_0+\int_0^1 u_t(\psi_t)dt\right)(X_0)}(x) - p_{\text{data}}(x)|dx$$

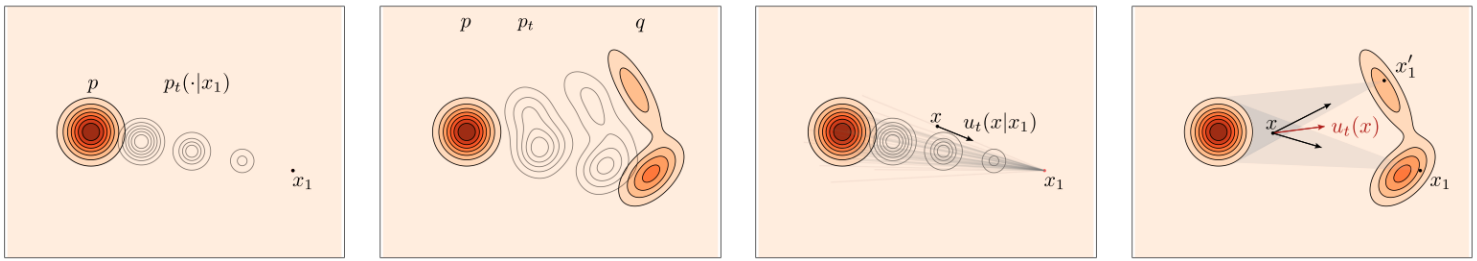
Looking for probability paths

Conditional probability paths

Consider conditioning the design of p_t on a realization $X_1 = x_1$, yielding $p_{t|1}(x|x_1)$. The marginal probability path p_t is:

$$p_t(x) = \int p_{t|1}(x|x_1) p_{\text{data}}(x_1) dx_1 \tag{6}$$

$p_{t|1}(x|x_1)$: The probability of $x \sim X_t$ given that we will be in x_1 when $t = 1$



(a) Conditional probability path $p_t(x|x_1)$. **(b)** (Marginal) Probability path $p_t(x)$. **(c)** Conditional velocity field $u_t(x|x_1)$. **(d)** (Marginal) Velocity field $u_t(x)$.

The Conditional Strategy: One probability path !

Flow Matching

Boundary conditions:

- $p_0 = \mathcal{N}(0, 1)$ (not always)
- $p_1 = p_{\text{data}}$

Conditional Flow Matching

Boundary conditions:

- $p_{0|1}(\cdot|x_1) = \mathcal{N}(0, 1)$ (not always)
- $p_{1|1}(\cdot|x_1) = \delta_{x_1}(x)$ (boundaries = nb samples)

One such conditional probability path

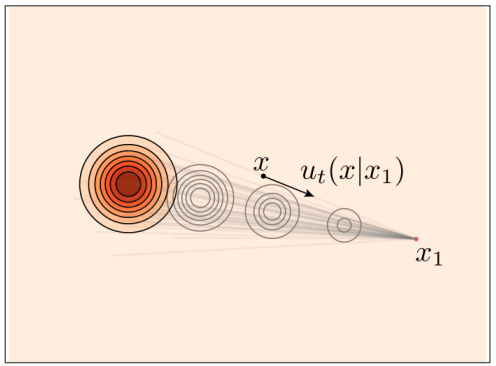
$$p_{t|1}(\cdot|x_1) = \mathcal{N}(\cdot| \underbrace{t x_1}_{\mu_t}, \underbrace{(1-t)^2 I}_{\sigma_t^2})$$

Indeed, we have $p_{1|1}(\cdot|x_1) = \delta_{x_1}(\cdot)$ and $p_{0|1}(\cdot|x_1) = \mathcal{N}(\cdot|0, I)$.

From probability path to vector field

Only one sample x_1

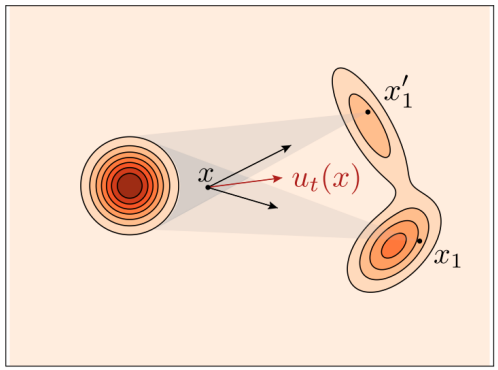
$u_t(\cdot|x_1)$ generates $p_{t|1}(\cdot|x_1)$



(c) Conditional velocity field $u_t(x|x_1)$.

All samples:

$$u_t(x) = \int u_t(x|x_1) p_{1|t}(x_1|x) dx_1$$



(d) (Marginal) Velocity field $u_t(x)$.

The final objective for Conditional Flow Matching (CFM)

Literal expression for $u_t(\cdot|x_1)$

$$\begin{cases} x_{t|1} = \left(t \cdot x_1 + (1-t) \cdot \underbrace{\varepsilon}_{\sim X_0} \right) \sim X_{t|1} \\ dX_{t|1} = u_t(X_{t|1}|x_1)dt \implies u_t\left(X_{t|1}|x_1\right) = x_1 - X_0 \end{cases}$$

Gradient Equality

$$\nabla \mathcal{L}_{\text{Logical}}(\theta) = \nabla \mathcal{L}_{\text{Pratical}}(\theta)$$

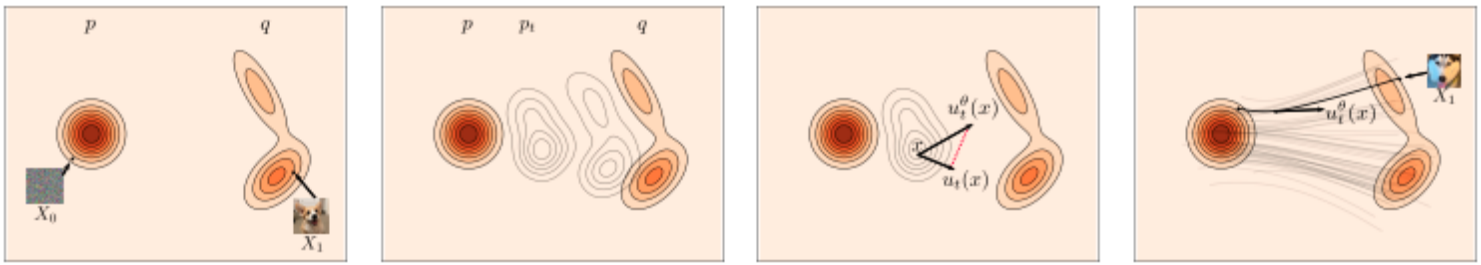
Loss for CFM

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathop{\mathbb{E}}_{\substack{t\sim U[0,1] \\ x_0\sim\mathcal{N}(0,I) \\ x_1\sim p_{\text{data}}}} \left[\| u_\theta(\underbrace{(t \cdot x_1 + (1-t)) \cdot x_0}_{=x_t}, t) - (x_1 - x_0) \|_2 \right]$$

Flow matching summary

Summary

- Searching for ψ or $u_{t\in[0,1]}$ is complicated so we look for probability paths $p_{X_{t\in[0,1]}}$.
- We know a (conditional) probability path $p_{X_{(t\in[0,1])|1}}$ with the right boundary conditions.
- We learn the vector field $u_{(t\in[0,1])|1}$ producing that probability path.
- To sample from p_{data}, sample $x_0 \sim X_0$ and simulate $dX_t = u_\theta(X_t)dt$ (until 1).



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

Rectified flows: Optimal Transport

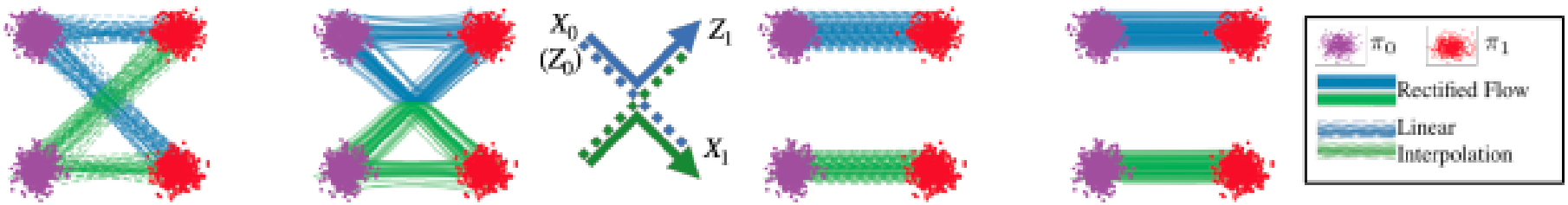
Recursive flow

$$Z^{k+1}_{t\in[0,1]} = \mathcal{T}\left(Z^{k}_{t\in[0,1]}\right)$$

s.t. $(Z^k_0, Z^k_1) = (X_0, X_1) \quad \forall k \in [|0, n|]$

Minimize Straightness

$$\begin{cases} S(Z) = \int_0^1 \mathbb{E}\left[\| (Z_1 - Z_0) - \dot{Z}_t \|^2\right] dt \\ \min_{k\in\{0,\dots,K\}} S(Z^k) \leq \dfrac{\mathbb{E}\left[\| X_1 - X_0 \|^2\right]}{K} \end{cases}$$



(a) Linear interpolation $X_t = tX_1 + (1-t)X_0$

(b) Rectified flow Z_t induced by (X_0, X_1)

(c) Linear interpolation $Z_t = tZ_1 + (1-t)Z_0$

(d) Rectified flow Z'_t induced by (Z_0, Z_1)

General intuition

Score-based and flow matching methods share similarities.

Training

- Score matching: $\mathcal{L} = \mathbb{E}_{p_{\text{data}}(x)}[\|s_\theta(x) - \nabla_x \log p_{\text{data}}(x)\|^2]$
- Flow matching: $\mathcal{L}_{FM}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1],x\sim p_t(x)}\left[||u_\theta(x,t) - u(x,t)||^2\right]$

Sampling

- Score-based: by simulating the reverse SDE $dX_t = \left[f(X_t,t) - g^2(t)\nabla \log p_t(X_t)\right] dt + g(t)dW_t$
- Flow matching: by simulating the transport ODE $dX_t = u(X_t,t)dt$

Fokker-Planck and probability paths

Fokker-Planck equation

Given the SDE $dX_t = \mu(X_t, t)dt + \sigma(X_t, t)dW_t$, Fokker-Planck equation describes the evolution of the probability path $p_t(x)$:

$$\frac{\partial p_t(x)}{\partial t} = -\text{div}_x(\mu(x,t)p_t(x)) + \frac{1}{2}\left[\sum_{i,j}\frac{\partial^2}{\partial_j \partial_j}\sigma(.,t)p_t(.)\right](x) \tag{7}$$

Special cases

- Deterministic case (FM): $\sigma(x,t) = 0 \Rightarrow \frac{\partial p_t(x)}{\partial t} = -\text{div}_x(\mu(x,t)p_t(x))$
- Diffusion case: $\sigma(x,t) = \sigma(t) \Rightarrow \frac{\partial p_t(x)}{\partial t} = -\text{div}_x\left(\left[\mu(x,t) - \frac{1}{2}\sigma(t)^2\nabla \log p_t(x)\right]p_t(x)\right)$

Fokker-Planck and probability paths

Special cases

- Deterministic case (FM): $\sigma(x,t) = 0 \Rightarrow \frac{\partial p_t(x)}{\partial t} = -\text{div}_x(\mu(x,t)p_t(x))$
- Diffusion case: $\sigma(x,t) = \sigma(t) \Rightarrow \frac{\partial p_t(x)}{\partial t} = -\text{div}_x\left(\left[\mu(x,t) - \frac{1}{2}\sigma(t)^2\nabla \log p_t(x)\right] p_t(x)\right)$

Shared probability paths

The dynamics defined by

$$dX_t = \mu(X_t,t)dt + \sigma(t)dW_t \quad \text{(stochastic)} \tag{7}$$

$$dX_t = \left[\mu(X_t,t) - \frac{1}{2}\sigma(t)^2\nabla \log p_t(X_t)\right] dt \quad \text{(deterministic)} \tag{8}$$

share the same probability path, given by

$$\frac{\partial p_t(x)}{\partial t} = -\text{div}_x\left(\left[\mu(x,t) - \frac{1}{2}\sigma(t)^2\nabla \log p_t(x)\right] p_t(x)\right). \tag{9}$$

Formally

Time convention

- Flow matching: t from 0 (”noise”) to 1 (data)
- Diffusion: r from $r = +\infty$ (noise) to 0 (data)

Let us denote by $r = k(t)$ the reparameterization.

Forward diffusion VS Conditional probability path

Considering the forward diffusion $dX_r = f(X_r, r)dr + g(r)dW_r$, the density of $(X_r|X_0)$ is given by

$$p(x_r|x_{r=0}) = \mathcal{N}(\tilde{f}(r)x_{r=0}, \tilde{g}(r)^2 I) \tag{10}$$

With the above time reparameterization, it corresponds to the conditional probability path

$$p_{t|1}(x_t|x_1 = x_{r=0}) = \mathcal{N}(\alpha_t x_1, \sigma_t^2 I) \tag{11}$$

with $\alpha_t = \tilde{f}(k(t))$ and $\sigma_t = \tilde{g}(k(t))$. This probability path is an affine Gaussian probability paths, constructed with the conditional flow

$$X_t = \alpha_t X_1 + \sigma_t X_0 \tag{12}$$

Formally - Sampling

Link between score and velocity field

Considering an affine Gaussian probability path $p_{t|1}(x_t|x_0) = \mathcal{N}(\alpha_t x_0, \sigma_t^2 I)$, we have

$$u(x,t) = \frac{\dot{\alpha}_t}{\alpha_t} x - \frac{\dot{\sigma}_t \sigma_t \alpha_t - \dot{\alpha}_t \sigma_t^2}{\alpha_t} \nabla \log p_t(x) \tag{13}$$

$$= \dot{k}(t) \left[f(x,t) - \frac{g(t)^2}{2} \nabla \log p_t(x) \right] \tag{14}$$

$u(x,t)$ is classically approximated using *flow matching*, while, $\nabla \log p_t(x)$ is classically approximated using *score matching*.

Then, ODE or SDE sampling following the equations

$$dX_t = \dot{k}(t) \underbrace{\left[f(X_t,t) - \frac{1}{2} g(t)^2 \nabla \log p_t(X_t) \right]}_{=u(X_t,t)} dt \quad \text{(deterministic)} \tag{15}$$

$$dX_t = \dot{k}(t) \left(f(X_t,t) - g(t)^2 \nabla \log p_t(X_t) \right) dt + \dot{k}(t) g(t) dW_t \quad \text{(stochastic)} \tag{16}$$

Score-based VS Flow matching summary

Score-based methods "=" Gaussian affine conditional flow

- The probability paths defined by the reverse diffusion SDE and the flow ODE of a Gaussian affine conditional flow are the same.
- In this setting, the velocity field $u(x, t)$ is an affine combination of x and the score $\nabla \log p_t(x)$
- In both frameworks, we can sample from p_{data} by solving a SDE or an ODE.

$\rightarrow$ FM more general framework, as it does not necessarily interpolate from a Gaussian to p_{data}.

Source: "Le poly de Meta"

Conclusion & Takeaways

- **Challenge:** Generate sample from an unknown distribution.

Thanks for your attention!