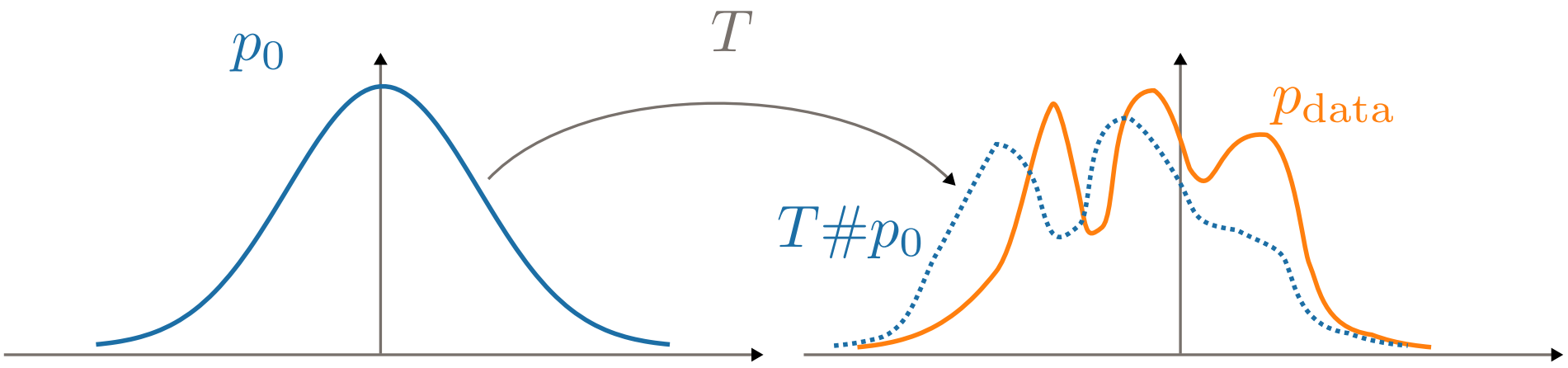
From Diffusion Models to Flow Matching: A Gentle Introduction to Modern Generative Sampling

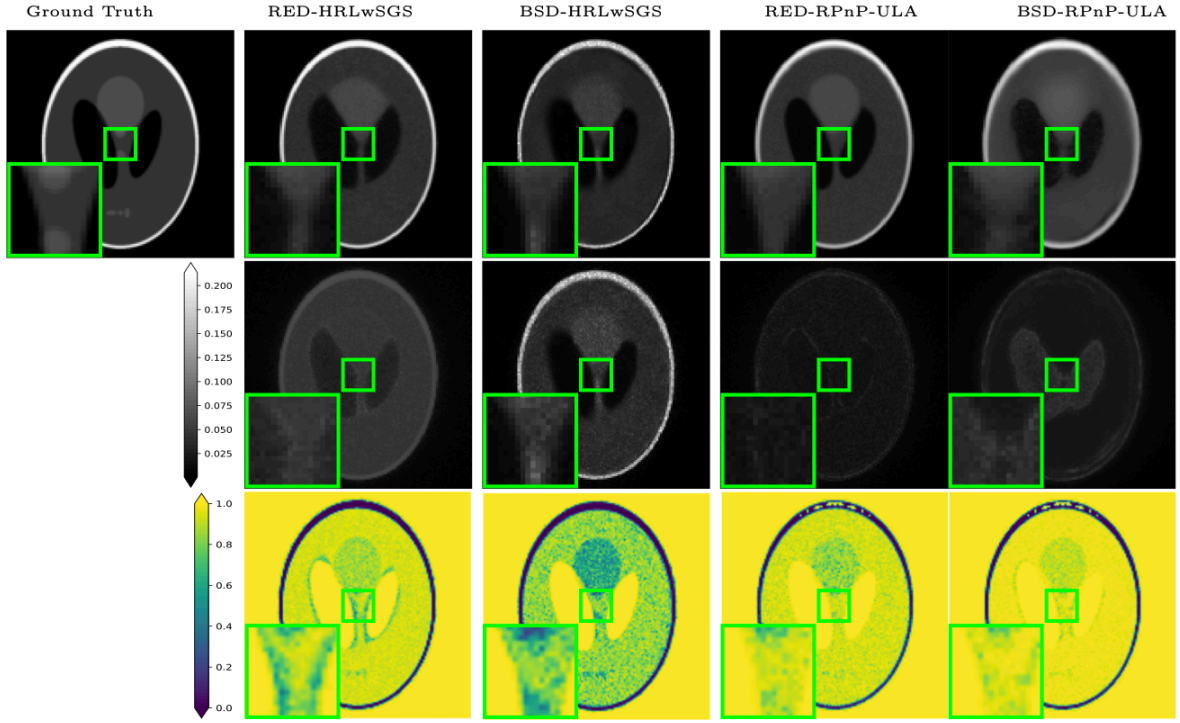


Emile Pierret, CSD ENS Ulm, pierret@math.cnrs.fr
Journées CASCIMODOT, Orléans, 18 novembre 2025

Introduction: Why generative models ?

Generating images allows to have a better understanding of images in general

$\Longrightarrow$ Allows for solving inverse problems and quantifying uncertainty



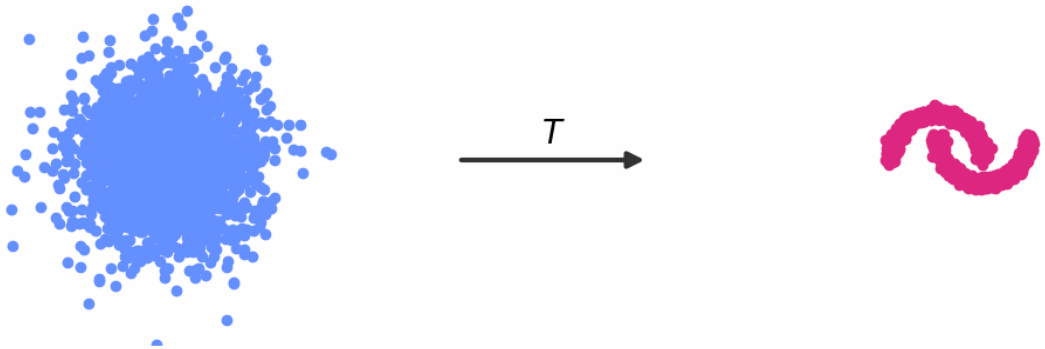
[EC Faye et al., 2024]

Generating text for storytelling, summarization, translation, code generation

$\Longrightarrow$ Si tu veux, je peux fusionner images + texte en une seule slide.

Generative models and push-forward

GAN [Goodfellow et al., 2014], VAE [Kingma and Welling, 2019]



Objective: If $X_0 \sim \mathcal{N}(0,I)$, then $X_1 = T(X_0) \sim p_{\text{data}}$

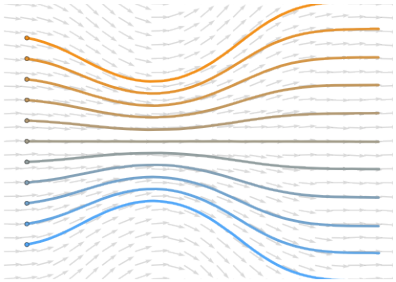
Training: minimize $\mathbb{E}[D(T_\theta(X_0), X_1)]$

Drawbacks: instable training, samples images can be blurred

$(T_t)_{0\leq t\leq 1}$ encoded as a flow (Continuous Normalizing Flows) [Chen et al., 2025]

Given an ODE

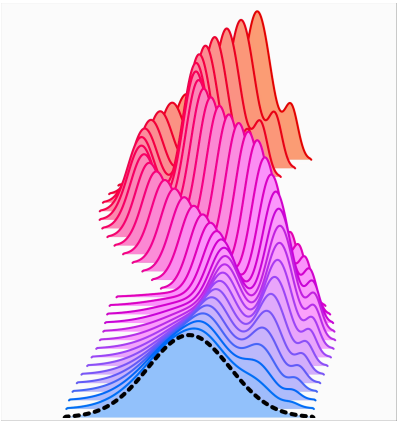
$$\frac{\partial}{\partial_t} x(t) = v_t(x(t))$$



The flow is written as

$$\frac{\partial \varphi_t}{\partial t}(x) = v_t(\varphi_t(x)), \qquad \varphi_0(x) = x \qquad \text{or} \qquad \varphi_t(x) = x + \int_0^t v_s(\varphi_s(x))ds\,.$$

Objective: If $Z_0 \sim \mathcal{N}(0,I)$, then $Z_t = \varphi_t(Z_0) \sim p_t$



Objective: Learn v_θ such that

$$\frac{\partial}{\partial_t} x(t) = v_\theta(t, x(t))$$

Training procedure: For each training data, integrate the ODE and minimize a loss

Drawbacks: Unstable and costly training

Today: models and flow matching

Guide:

1) Prescribe a path $(p_t)_{0\leq t\leq 1}$ between $\mathcal{N}(0,I)$ and p_{data}

2) Learn the associated velocity field $(v_t)_{0\leq t\leq 1}$
(without integrating the whole equation at each training step)

$$\frac{\partial}{\partial_t}x(t) = v_t(x(t))$$

3) Integrate the ODE via a discretization scheme

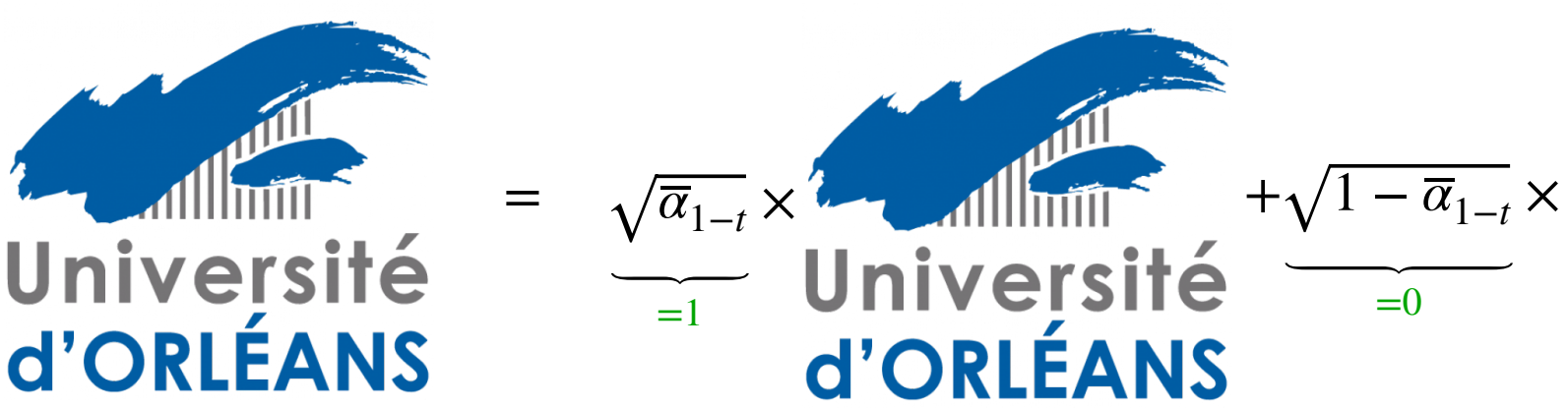
We need probabilistic tools to guarantee that $(v_t)_{0\leq t\leq 1}$ *generates* $(p_t)_{0\leq t\leq 1}$

Diffusion models

Diffusion models [Y. Song et al., 2021], [Ho et al., 2020], [Y. Song et al., 2019]

Main idea: 1) Build a probability path from p_{data} to $\mathcal{N}(0,I)$

$t = 1$



This dynamics is described by the Stochastic Differential Equation (SDE)

$$dx_t = -\beta_t x_t dt + \sqrt{2\beta_t} dw_t \quad \text{with } \overline{\alpha}_{1-t} = e^{-2\int_0^t \beta_u du}$$

Diffusion models

Main idea: 1) Build a probability path from p_{data} to $\mathcal{N}(0,I)$

$t = 0.9$

$=$ $\underbrace{\sqrt{\overline{\alpha}_{1-t}}}_{=0.95} \times$

$+\underbrace{\sqrt{1-\overline{\alpha}_{1-t}}}_{=0.32} \times$

This dynamics is described by the Stochastic Differential Equation (SDE)

$$dx_t = -\beta_t x_t dt + \sqrt{2\beta_t} dw_t \quad \text{with } \overline{\alpha}_{1-t} = e^{-2\int_0^t \beta_u du}$$

Diffusion models

Main idea: 1) Build a probability path from p_{data} to $\mathcal{N}(0,I)$

$t = 0.8$

$$= \underbrace{\sqrt{\overline{\alpha}_{1-t}}}_{=0.81} \times$$

$$+ \underbrace{\sqrt{1-\overline{\alpha}_{1-t}}}_{=0.58} \times$$

This dynamics is described by the Stochastic Differential Equation (SDE)

$$dx_t = -\beta_t x_t dt + \sqrt{2\beta_t} dw_t \quad \text{with } \overline{\alpha}_{1-t} = e^{-2\int_0^t \beta_u du}$$

Diffusion models

Main idea: 1) Build a probability path from p_{data} to $\mathcal{N}(0,I)$

$t = 0.7$

$$= \underbrace{\sqrt{\overline{\alpha}_{1-t}}}_{=0.63} \times$$

$$+ \underbrace{\sqrt{1-\overline{\alpha}_{1-t}}}_{=0.78} \times$$

This dynamics is described by the Stochastic Differential Equation (SDE)

$$dx_t = -\beta_t x_t dt + \sqrt{2\beta_t} dw_t \quad \text{with } \overline{\alpha}_{1-t} = e^{-2\int_0^t \beta_u du}$$

Diffusion models

Main idea: 1) Build a probability path from p_{data} to $\mathcal{N}(0,I)$

$t = 0.6$

$$= \underbrace{\sqrt{\overline{\alpha}_{1-t}}}_{=0.44} \times \text{(Université d'ORLÉANS)} + \underbrace{\sqrt{1-\overline{\alpha}_{1-t}}}_{=0.90} \times$$

This dynamics is described by the Stochastic Differential Equation (SDE)

$$dx_t = -\beta_t x_t dt + \sqrt{2\beta_t} dw_t \quad \text{with } \overline{\alpha}_{1-t} = e^{-2\int_0^t \beta_u du}$$

Diffusion models

Main idea: 1) Build a probability path from p_{data} to $\mathcal{N}(0,I)$

$t = 0.5$

$$= \underbrace{\sqrt{\overline{\alpha}_{1-t}}}_{=0.28} \times \text{Image} + \underbrace{\sqrt{1-\overline{\alpha}_{1-t}}}_{=0.96} \times$$

This dynamics is described by the Stochastic Differential Equation (SDE)

$$dx_t = -\beta_t x_t dt + \sqrt{2\beta_t} dw_t \quad \text{with } \overline{\alpha}_{1-t} = e^{-2\int_0^t \beta_u du}$$

Diffusion models

Main idea: 1) Build a probability path from p_{data} to $\mathcal{N}(0,I)$

$t = 0.4$

$$= \underbrace{\sqrt{\overline{\alpha}_{1-t}}}_{=0.16} \times \text{Université d'ORLÉANS} + \underbrace{\sqrt{1-\overline{\alpha}_{1-t}}}_{=0.99} \times$$

This dynamics is described by the Stochastic Differential Equation (SDE)

$$dx_t = -\beta_t x_t dt + \sqrt{2\beta_t} dw_t \quad \text{with } \overline{\alpha}_{1-t} = e^{-2\int_0^t \beta_u du}$$

Diffusion models

Main idea: 1) Build a probability path from p_{data} to $\mathcal{N}(0,I)$

$t = 0.3$

$$= \underbrace{\sqrt{\overline{\alpha}_{1-t}}}_{=0.08} \times \text{Université d'ORLÉANS} + \underbrace{\sqrt{1-\overline{\alpha}_{1-t}}}_{=0.99} \times$$

This dynamics is described by the Stochastic Differential Equation (SDE)

$$dx_t = -\beta_t x_t dt + \sqrt{2\beta_t} dw_t \quad \text{with } \overline{\alpha}_{1-t} = e^{-2\int_0^t \beta_u du}$$

Diffusion models

Main idea: 1) Build a probability path from p_{data} to $\mathcal{N}(0,I)$

$t = 0.2$

$$= \underbrace{\sqrt{\overline{\alpha}_{1-t}}}_{=0.08} \times \text{Université d'ORLÉANS} + \underbrace{\sqrt{1-\overline{\alpha}_{1-t}}}_{=0.99} \times$$

This dynamics is described by the Stochastic Differential Equation (SDE)

$$dx_t = -\beta_t x_t dt + \sqrt{2\beta_t} dw_t \quad \text{with } \overline{\alpha}_{1-t} = e^{-2\int_0^t \beta_u du}$$

Diffusion models

Main idea: 1) Build a probability path from p_{data} to $\mathcal{N}(0,I)$

$t = 0.1$

$$= \underbrace{\sqrt{\overline{\alpha}_{1-t}}}_{=0.02} \times \text{Université d'ORLÉANS} + \underbrace{\sqrt{1-\overline{\alpha}_{1-t}}}_{=0.99} \times$$

This dynamics is described by the Stochastic Differential Equation (SDE)

$$dx_t = -\beta_t x_t dt + \sqrt{2\beta_t} dw_t \quad \text{with } \overline{\alpha}_{1-t} = e^{-2\int_0^t \beta_u du}$$

Diffusion models

Main idea: 1) Build a probability path from p_{data} to $\mathcal{N}(0,I)$

$t = 0$

$$= \underbrace{\sqrt{\overline{\alpha}_{1-t}}}_{=0.01} \times \text{(Université d'ORLÉANS)} + \underbrace{\sqrt{1-\overline{\alpha}_{1-t}}}_{=0.99} \times$$

This dynamics is described by the Stochastic Differential Equation (SDE)

$$dx_t = -\beta_t x_t dt + \sqrt{2\beta_t} dw_t \quad \text{with } \overline{\alpha}_{1-t} = e^{-2\int_0^t \beta_u du}$$

Precisely, if $X_1 \sim p_{\text{data}}$ and follows the SDE, $X_t \sim p_t$

Diffusion models

Main tool: Fokker-Planck-equation

Given a SDE, $dx_t = -\beta_t x_t dt + \sqrt{2\beta_t} dw_t$

$$\frac{\partial p_t}{\partial_t}(x) = -\operatorname{div}\left(-\beta_t x p_t(x)\right) + \frac{1}{2} 2\beta_t \Delta_x p_t(x) \qquad \text{(Fokker-Planck)}$$

By aiming $q_t = p_{1-t}$ which is a path from $\mathcal{N}(0, I)$ to p_{data},

$$\frac{\partial q_t}{\partial_t}(x) = -\operatorname{div}\left(-\beta_t [x_t + \nabla \log p_t(x_t)] q_t(x)\right) \qquad \text{(Reversed Fokker-Planck)}$$

that corresponds to the ODE

$$dx_t = -\beta_t [x_t + \nabla \log p_t(x_t)] dt \qquad (1)$$

Precisely, if $Z_0 \sim p_0$ and follows the ODE (1), $Z_1 \sim p_{\text{data}}$

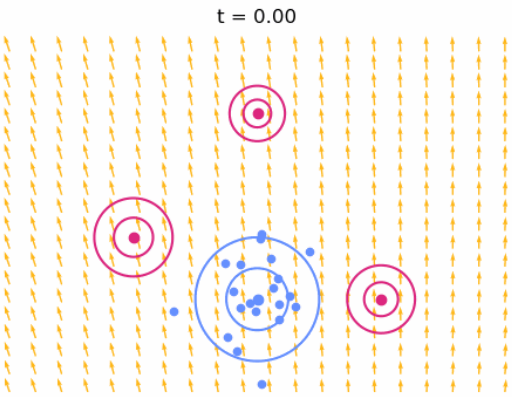
Training procedure

1) Approximate the score function $\nabla \log p_t(x)$ by a neural network $s_\theta(t, x)$

By Tweedie's formula, the score function can be learned as a denoiser

$$s_\theta(x, t) \propto \text{argmin}_{t\sim\text{Unif}([0,1]),x_1\sim p_{\text{data}},\varepsilon_0\sim\mathcal{N}(0,I)}\mathbb{E}[\|s_\theta(x, t) - (\sqrt{\overline{\alpha}_{t-1}}x_1 + \sqrt{1-\overline{\alpha}_{t-1}}\varepsilon_0)\|^2]$$

$\Longrightarrow$ It is not necessary to integrate the whole ODE for the training.



2) Integrate the ODE

$$dx_t = -\beta_t[x_t + s_\theta(t, x_t)]dt \qquad (dx_t = -\beta_t[x_t + \nabla \log p_t(x_t)]dt)$$

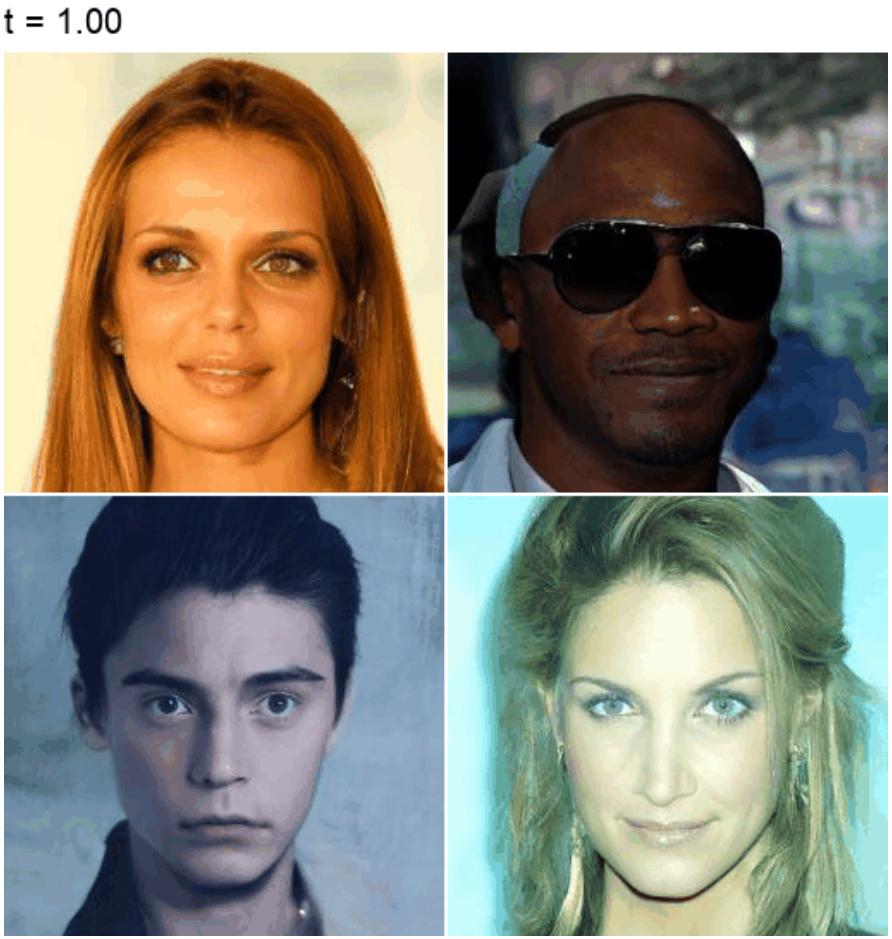
Illustration with images

1) Learning of $s_\theta(t, x_t)$ on a database of faces

2) Integration of

$$dx_t = -\beta_t[x_t + s_\theta(t, x_t)]dt$$

with a Heun's scheme and 250 steps.



Initialization error

Note that the probability path, given previously is not exactly between p_{data} and $\mathcal{N}(0,I)$

$t = 0$

$\sim p_0 \neq \mathcal{N}(0,I)$

$= \underbrace{\sqrt{\overline{\alpha}_{1-t}}}_{=0.01} \times$

$+\underbrace{\sqrt{1-\overline{\alpha}_{1-t}}}_{=0.99} \times$

The Fokker-Planck equation builds a bridge between p_{data} and p_0

In practice, $D_{KL}(p_0,\mathcal{N}(0,I)) < \varepsilon$ and we adopt this approximation, called *initialization error.*

Flow matching

Flow matching [Lpiman et al., 2023, Liu et al., 2023]

We directly build a probability path $(p_t)_{0\leq t\leq 1}$ from $\mathcal{N}(0,I)$ to p_{data},

Main tool: Continuity equation

Given a velocity field v such that

$$\frac{\partial}{\partial t}p_t + \text{div}(p_t v_t) = 0 \qquad \text{(CE)}$$

Under regularity assumptions, the flow verifying

$$\frac{\partial \varphi_t}{\partial t}(x) = v_t(\varphi_t(x)), \qquad \varphi_0(x) = x$$

is such that if $Z_0 \sim \mathcal{N}(0,I)$ then $Z_1 := \varphi_1(Z_0) \sim p_1$

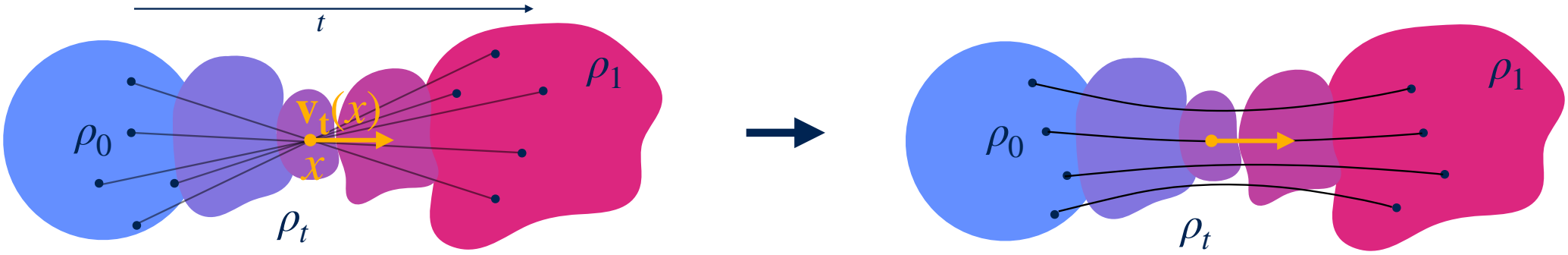
An ingenious of v

We prescribe the probability path $(p_t)_{t\in[0,1]}$ which is the distribution of $X_t = (1-t)X_0 + tX_1$

We can show that

$$v_t(x) = \mathbb{E}(X_1 - X_0 \mid X_t = x)$$

satisfies the continuity equation (CE)

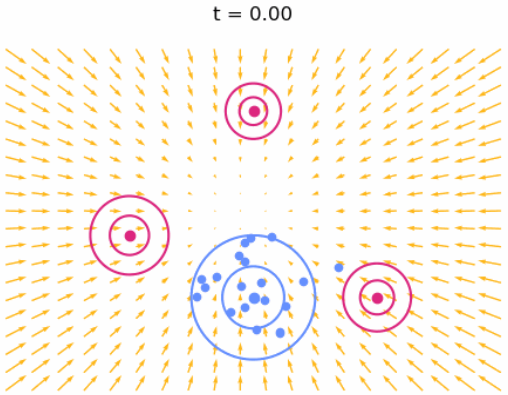


Training procedure

1) Approximate the velocity field $v_t(x) = \mathbb{E}(X_1 - X_0 \mid X_t = x)$ by a neural network $v_\theta(t, x)$ by minimizing

$$v_\theta(x, t) \propto \text{argmin}_{t\sim\text{Unif}([0,1]),x_1\sim p_{\text{data}},\varepsilon_0\sim\mathcal{N}(0,I)}\mathbb{E}[\|s_\theta(x, t) - (x_1 - \varepsilon_0)\|^2]$$

$\Longrightarrow$ It is not necessary to integrate the whole ODE for the training.



2) Integrate

$$dx_t = v_\theta(x, t)dt$$

Illustration with images

with Euler scheme and 90 steps.

[Dao et al., 2023]

Comparison of both approaches

	Diffusion models	**Flow matching**
Main tool	Fokker-Planck equation	Continuity equation
Guide	Reverse a *forward SDE*	Follow a path given by random variables
Learning	Learn the score function	Learn the velocity field
Sampling	Euler/Heun or Euler-Maruyama schemes	Euler scheme in general
Initialization error	Yes 😢	No 😀

Correspondence between both approaches

We can write the Flow Matching's path a Diffusion Model's forward SDE

$$dx_t = -\frac{1}{1-t}x_t dt + \sqrt{\frac{2t}{1-t}}dw_t \qquad \text{with } \sqrt{\frac{2t}{1-t}} \underset{t\longrightarrow 1}{\longrightarrow} +\infty$$

This is a memoryless process. [Domingo-Enrich et al., 2024]

In the Gaussian case, with $p_{\text{data}} = \mathcal{N}(0,\Sigma)$

Because of the initialization error,

Diffusion models (with ODE) by replacing p_0 by $\mathcal{N}(0,I)$ is not Optimal Transport

[EP, Galerne, 2025]

There is not initialization error,

Flow Matching corresponds exactly to Optimal Transport

[Hertrich et al., 2025]

Conclusion

Not discussed today:

- The similar study of the *truncature error*

- Link between Flow Matching and Optimal Transport

- General Flow matching: between another distribution than $\mathcal{N}(0,I)$ and p_{data}

To appear in January: *An Introduction to Flow Matching for Applied Mathematicians, joint work with Julie Delon, Alasdair Newson, Valentine Tosel.*

Thank you for your attention !