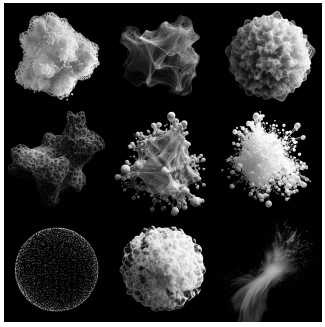
An Introduction to Diffusion Models

March 18, 2026

Workshop “Extreme Events and Generative AI”

What is generative modeling?



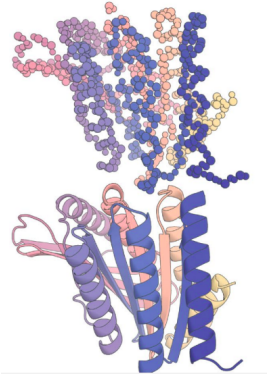
Original dataset

new data aiming to resemble the dataset

What is generative modeling?

Generative modeling is a cornerstone of modern AI, powering applications from image generation to drug discovery.

Image generated by Midjourney



Molecule generated by RFdiffusion

What is generative modeling?

Setup:

We are given a dataset $X_1, X_2, ..., X_n \in \mathbb{R}^d$.

What is generative modeling?

Setup:

We are given a dataset $X_1, X_2, ..., X_n \in \mathbb{R}^d$.

Model:

We suppose that there exists $p^\star \in \mathcal{P}(\mathbb{R}^d)$ such that $X_i \sim p^\star$ are iid.

What is generative modeling?

Setup:
We are given a dataset $X_1, X_2, ..., X_n \in \mathbb{R}^d$.

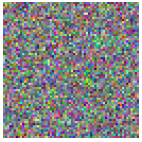
Model:
We suppose that there exists $p^\star \in \mathcal{P}(\mathbb{R}^d)$ such that $X_i \sim p^\star$ are iid.

Goal:
Find an algorithm $\mathcal{A} : (\mathbb{R}^d)^n \times \mathbb{R}^d \to \mathbb{R}^d$ such that given the dataset $X_1, ..., X_n$, $\mathcal{A}(X_1, ..., X_n, \cdot) : \mathbb{R}^d \to \mathbb{R}^d$ is a function that takes in entry an independent seed $Z \sim \nu$ and outputs a new point $\mathcal{A}(X_1, ..., X_n, Z) \in \mathbb{R}^d$ that "resembles" the dataset.

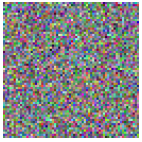
What is generative modeling?

$X_1, ..., X_n \sim p^\star$

$$\longrightarrow \quad \mathcal{A}(X_1, \ldots, X_n, \cdot)$$



$Z_1 \sim \nu$



$Z_2 \sim \nu$

$\longrightarrow$

$\mathcal{A}(X_1, \ldots, X_n, Z_1)$

$\mathcal{A}(X_1, \ldots, X_n, Z_2)$

Goal: $\mathcal{A}(X_1, \ldots, X_n, \cdot)_{\#}\nu \approx p^\star$

What is generative modeling?

Setup:

We are given a dataset $X_1, X_2, ..., X_n \in \mathbb{R}^d$.

Model:

We suppose that there exists $p^\star \in \mathcal{P}(\mathbb{R}^d)$ such that $X_i \sim p^\star$ are iid.

Goal:

Find an algorithm $\mathcal{A} : (\mathbb{R}^d)^n \times \mathbb{R}^d \to \mathbb{R}^d$ such that for $\nu \in \mathcal{P}(\mathbb{R}^d)$ easy-to-sample :

$$\text{taking} \quad \hat{p}(X_1, ..., X_n) = \mathcal{A}(X_1, ..., X_n, \cdot)_{\#}\nu,$$

$$\mathbb{E}_{X_i \sim p^\star}\Big[\mathbb{M}\big(\hat{p}(X_1, ..., X_n), p^\star\big)\Big] \quad \text{is small,}$$

for $\mathbb{M}(\cdot, \cdot)$ a metric on $\mathcal{P}(\mathbb{R}^d)$.

A brief history of diffusion models

Sohl-Dickstein et al. (2015)

- First clear diffusion-based generative model.
- Forward noising + learned reverse dynamics.
- Beautiful idea, but limited visual quality.

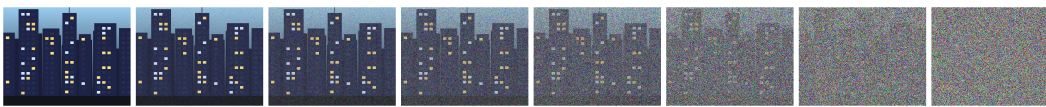
Song et al. (2020)

- Score-based generative modeling through SDEs.
- Modern mathematical formulation of diffusion models.
- Strong sample quality and major revival of the field.

Forward SDE (data → noise)

$$d\vec{X}_t = -\vec{X}_t dt + \sqrt{2} dB_t$$

$\vec{X}_0$ → $\vec{X}_T$



X_T → X_0

$$dX_t = (X_t + 2s_{T-t}(X_t))dt + \sqrt{2} dB_t$$

Reverse SDE (noise → data)

Recap on SDEs

A stochastic differential equation (SDE) has the form

$$\mathrm{d}Y_t = a(t, Y_t)\,\mathrm{d}t + b(t, Y_t)\,\mathrm{d}B_t, \qquad Y_0 \sim p_0,$$

where:

- $a(t, x)$ is the drift,
- $b(t, x)$ is the diffusion coefficient,
- $(B_t)_{t \geq 0}$ is a Brownian motion.

Recap on SDEs

A stochastic differential equation (SDE) has the form

$$\mathrm{d}Y_t = a(t, Y_t)\,\mathrm{d}t + b(t, Y_t)\,\mathrm{d}B_t, \qquad Y_0 \sim p_0,$$

where:

- $a(t, x)$ is the drift,
- $b(t, x)$ is the diffusion coefficient,
- $(B_t)_{t \geq 0}$ is a Brownian motion.

It is understood in integral form:

$$Y_t = Y_0 + \int_0^t a(s, Y_s)\,\mathrm{d}s + \int_0^t b(s, Y_s)\,\mathrm{d}B_s.$$

Recap on SDEs

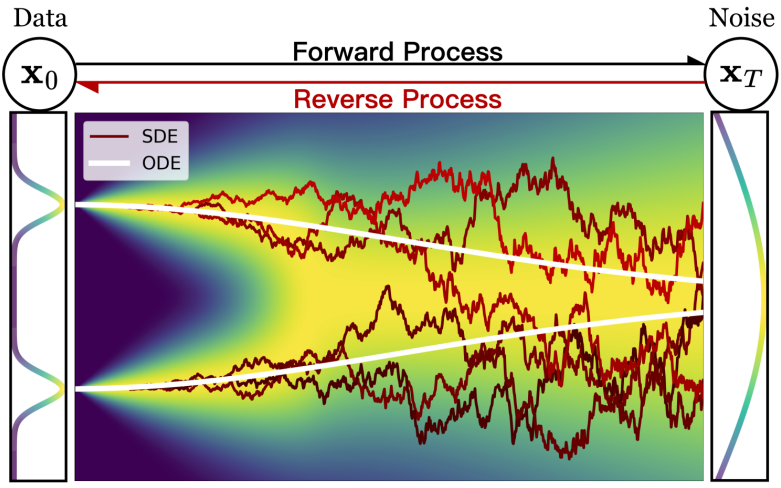


Figure 1: Figure by Andrew Chan

Recap on Itô integral

For an adapted square-integrable process $(Y_t)_{t\in[a,b]}$, the Itô integral is defined as

$$\int_a^b Y_t \,\mathrm{d}B_t := \lim_{n\to\infty} \sum_{j=1}^{p_n} Y_{t_{j-1}^n}\big(B_{t_j^n} - B_{t_{j-1}^n}\big),$$

where

$$a = t_0^n < t_1^n < \cdots < t_{p_n}^n = b, \qquad \max_j |t_j^n - t_{j-1}^n| \xrightarrow[n\to\infty]{} 0,$$

and

$$B_{t_j^n} - B_{t_{j-1}^n} \sim \mathcal{N}\big(0, t_j^n - t_{j-1}^n\big),$$

are independent.

The forward process

The forward process corresponds to

$$d\vec{X}_t = -\vec{X}_t\, dt + \sqrt{2}\, dB_t, \qquad \vec{X}_0 \sim p^\star.$$

The forward process

The forward process corresponds to

$$d\vec{X}_t = -\vec{X}_t\, dt + \sqrt{2}\, dB_t, \qquad \vec{X}_0 \sim p^\star.$$

The solution can be written as

$$\vec{X}_t = e^{-t}\vec{X}_0 + \sqrt{1 - e^{-2t}}\xi, \qquad \xi \sim \mathcal{N}(0, I_d), \quad \xi \perp \vec{X}_0.$$

- At $t = 0$: $\vec{X}_t = \vec{X}_0 \sim p^\star$.
- For large t: $p_t = \mathcal{L}(\vec{X}_t)$ becomes close to a Gaussian law.

Fokker–Planck equation

Consider the diffusion

$$dY_t = a(t, Y_t)\,dt + b(t, Y_t)\,dB_t, \qquad \Sigma(t,x) := b(t,x)b(t,x)^\top.$$

Fokker–Planck equation

Consider the diffusion

$$dY_t = a(t, Y_t)\,dt + b(t, Y_t)\,dB_t, \qquad \Sigma(t,x) := b(t,x)b(t,x)^\top.$$

Let p_t denote the density of Y_t. Using Itô's formula we get for every smooth test function φ,

$$\frac{d}{dt}\mathbb{E}[\varphi(Y_t)] = \mathbb{E}\Big[a(t, Y_t)\cdot\nabla\varphi(Y_t) + \frac{1}{2}\mathrm{Tr}(\Sigma(t, Y_t)\nabla^2\varphi(Y_t))\Big].$$

Fokker–Planck equation

Consider the diffusion

$$dY_t = a(t, Y_t)\,dt + b(t, Y_t)\,dB_t, \qquad \Sigma(t,x) := b(t,x)b(t,x)^\top.$$

Let p_t denote the density of Y_t. Using Itô's formula we get for every smooth test function φ,

$$\frac{d}{dt}\mathbb{E}[\varphi(Y_t)] = \mathbb{E}\Big[a(t, Y_t)\cdot\nabla\varphi(Y_t) + \frac{1}{2}\mathrm{Tr}(\Sigma(t, Y_t)\nabla^2\varphi(Y_t))\Big].$$

Since $\mathbb{E}[\varphi(Y_t)] = \int \varphi(x)p_t(x)\,dx$, an integration by parts gives

$$\partial_t p_t(x) = -\nabla\cdot\big(a(t,x)p_t(x)\big) + \frac{1}{2}\sum_{i,j=1}^{d}\partial_{ij}\big(\Sigma_{ij}(t,x)p_t(x)\big).$$

The reverse process

Start from the forward Ornstein–Uhlenbeck diffusion

$$d\vec{X}_t = -\vec{X}_t\, dt + \sqrt{2}\, dB_t, \qquad \vec{X}_0 \sim p^\star,$$

whose density p_t solves

$$\partial_t p_t = \nabla \cdot (x p_t) + \Delta p_t.$$

The reverse process

Start from the forward Ornstein–Uhlenbeck diffusion

$$d\vec{X}_t = -\vec{X}_t\, dt + \sqrt{2}\, dB_t, \qquad \vec{X}_0 \sim p^\star,$$

whose density p_t solves

$$\partial_t p_t = \nabla \cdot (x p_t) + \Delta p_t.$$

To reverse time, fix $T > 0$ and set $q_t := p_{T-t}$.

The reverse process

Start from the forward Ornstein–Uhlenbeck diffusion

$$d\vec{X}_t = -\vec{X}_t\, dt + \sqrt{2}\, dB_t, \qquad \vec{X}_0 \sim p^\star,$$

whose density p_t solves

$$\partial_t p_t = \nabla \cdot (x p_t) + \Delta p_t.$$

To reverse time, fix $T > 0$ and set $q_t := p_{T-t}$. For $b_t \geq 0$,

$$\partial_t q_t = -\nabla \cdot (x q_t) - \Delta q_t$$

The reverse process

Start from the forward Ornstein–Uhlenbeck diffusion

$$d\vec{X}_t = -\vec{X}_t\, dt + \sqrt{2}\, dB_t, \qquad \vec{X}_0 \sim p^\star,$$

whose density p_t solves

$$\partial_t p_t = \nabla \cdot (x p_t) + \Delta p_t.$$

To reverse time, fix $T > 0$ and set $q_t := p_{T-t}$. For $b_t \geq 0$,

$$\begin{aligned} \partial_t q_t &= -\nabla \cdot (x q_t) - \Delta q_t \\ &= -\nabla \cdot \Big(\big(x + (1 + b_t^2) \nabla \log q_t \big) q_t \Big) + b_t^2 \Delta q_t. \end{aligned}$$

The reverse process

Start from the forward Ornstein–Uhlenbeck diffusion

$$d\vec{X}_t = -\vec{X}_t\, dt + \sqrt{2}\, dB_t, \qquad \vec{X}_0 \sim p^\star,$$

whose density p_t solves

$$\partial_t p_t = \nabla \cdot (x p_t) + \Delta p_t.$$

To reverse time, fix $T > 0$ and set $q_t := p_{T-t}$. For $b_t \geq 0$,

$$\begin{aligned} \partial_t q_t &= -\nabla \cdot (x q_t) - \Delta q_t \\ &= -\nabla \cdot \Big(\big(x + (1 + b_t^2) \nabla \log q_t \big) q_t \Big) + b_t^2 \Delta q_t. \end{aligned}$$

Hence we can take as backward process

$$dX_t = \Big(X_t + (1 + b_t^2) \nabla \log p_{T-t}(X_t) \Big)\, dt + \sqrt{2} b_t\, d\overline{B}_t, \quad X_0 \sim p_T.$$

Sampling with Score-Based Generative models

Forward Process (OU):

$$d\vec{X}_t = -\vec{X}_t dt + \sqrt{2} dB_t, \quad \vec{X}_0 \sim p^\star$$

$$\text{Fix } T > 0, \quad b_t \geq 0 \quad \text{and write} \quad p_t := \text{law}(\vec{X}_t), \quad s(t,x) := \nabla \log p_t(x)$$

Backward Process:

$$dX_t = \Big(X_t + (1 + b_t^2) \cdot s(T-t, X_t)\Big) dt + \sqrt{2} b_t dB_t, \quad X_0 \sim p_T$$

$$\text{law}(X_{T-t}) = \text{law}(\vec{X}_t)$$

Sampling with Score-Based Generative models

Forward Process (OU):

$$d\vec{X}_t = -\vec{X}_t dt + \sqrt{2} dB_t, \quad \vec{X}_0 \sim p^\star$$

$$\text{Fix } T > 0, \quad b_t \geq 0 \quad \text{and write} \quad p_t := \text{law}(\vec{X}_t), \quad s(t,x) := \nabla \log p_t(x)$$

Backward Process:

$$dX_t = \Big(X_t + (1 + b_t^2) \cdot s(T-t, X_t)\Big) dt + \sqrt{2} b_t dB_t, \quad X_0 \sim p_T$$

$$\text{law}(X_{T-t}) = \text{law}(\vec{X}_t)$$

In practice:

$$d\hat{X}_t = \Big(\hat{X}_t + (1 + b_t^2) \cdot \hat{s}(T-t, \hat{X}_t)\Big) dt + \sqrt{2} b_t dB_t, \quad \hat{X}_0 \sim \mathcal{N}(0, \text{Id})$$

with $\hat{s}$ an estimator of s.

Intuition on the score function

Assume that $p^\star$ is supported in $B(0, R)$, and let

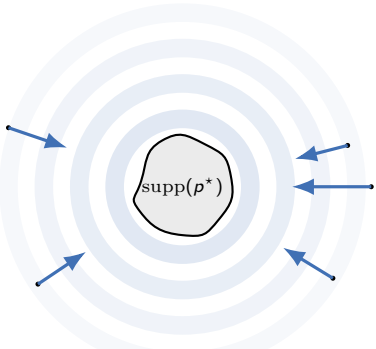
$$\tilde{p}_t = p^\star * \mathcal{N}(0, \sigma_t^2 I_d).$$

Far from the support ($\|x\| \gg R$), the Gaussian factor dominates:

$$\tilde{p}_t(x) = \int p^\star(y) \, \frac{e^{-\|x-y\|^2/(2\sigma_t^2)}}{(2\pi\sigma_t^2)^{d/2}} \, dy \;\approx\; C_t \, e^{-\|x\|^2/(2\sigma_t^2)}.$$

Hence

$$\nabla \log \tilde{p}_t(x) \approx -\frac{x}{\sigma_t^2}.$$



$\nabla \log \tilde{p}_t(x)$ points inward
and grows like $\|x\|/\sigma_t^2$ in the tails

Score trick

We have

$$p_t(x) = \int p_t(x \mid x_0)\, p_0(x_0)\, dx_0,$$

Score trick

We have

$$p_t(x) = \int p_t(x \mid x_0)\, p_0(x_0)\, dx_0,$$

so differentiating under the integral:

$$\nabla \log p_t(x) = \frac{\nabla p_t(x)}{p_t(x)} = \frac{1}{p_t(x)} \int \nabla_x p_t(x \mid x_0)\, p_0(x_0)\, dx_0$$

Score trick

We have

$$p_t(x) = \int p_t(x \mid x_0)\, p_0(x_0)\, dx_0,$$

so differentiating under the integral:

$$\begin{aligned}\nabla \log p_t(x) &= \frac{\nabla p_t(x)}{p_t(x)} = \frac{1}{p_t(x)} \int \nabla_x p_t(x \mid x_0)\, p_0(x_0)\, dx_0 \\ &= \int \nabla_x \log p_t(x \mid x_0) \frac{p_t(x \mid x_0) p_0(x_0)}{p_t(x)}\, dx_0\end{aligned}$$

Score trick

We have

$$p_t(x) = \int p_t(x \mid x_0)\, p_0(x_0)\, dx_0,$$

so differentiating under the integral:

$$\begin{aligned}\nabla \log p_t(x) &= \frac{\nabla p_t(x)}{p_t(x)} = \frac{1}{p_t(x)} \int \nabla_x p_t(x \mid x_0)\, p_0(x_0)\, dx_0 \\ &= \int \nabla_x \log p_t(x \mid x_0) \frac{p_t(x \mid x_0) p_0(x_0)}{p_t(x)}\, dx_0 = \mathbb{E}[\nabla_x \log p_t(x \mid X_0) \mid X_t = x]\,.\end{aligned}$$

Score trick

We have

$$p_t(x) = \int p_t(x \mid x_0)\, p_0(x_0)\, dx_0,$$

so differentiating under the integral:

$$\begin{aligned}\nabla \log p_t(x) &= \frac{\nabla p_t(x)}{p_t(x)} = \frac{1}{p_t(x)} \int \nabla_x p_t(x \mid x_0)\, p_0(x_0)\, dx_0 \\ &= \int \nabla_x \log p_t(x \mid x_0) \frac{p_t(x \mid x_0) p_0(x_0)}{p_t(x)}\, dx_0 = \mathbb{E}[\nabla_x \log p_t(x \mid X_0) \mid X_t = x]\,.\end{aligned}$$

For the Gaussian channel $X_t = \alpha_t X_0 + \sigma_t \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, I_d)$,

Score trick

We have

$$p_t(x) = \int p_t(x \mid x_0)\, p_0(x_0)\, dx_0,$$

so differentiating under the integral:

$$\begin{aligned}\nabla \log p_t(x) &= \frac{\nabla p_t(x)}{p_t(x)} = \frac{1}{p_t(x)} \int \nabla_x p_t(x \mid x_0)\, p_0(x_0)\, dx_0 \\ &= \int \nabla_x \log p_t(x \mid x_0) \frac{p_t(x \mid x_0) p_0(x_0)}{p_t(x)}\, dx_0 = \mathbb{E}[\nabla_x \log p_t(x \mid X_0) \mid X_t = x]\,.\end{aligned}$$

For the Gaussian channel $X_t = \alpha_t X_0 + \sigma_t \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, I_d)$,

$$\nabla_x \log p_t(x \mid X_0) = -\frac{x - \alpha_t X_0}{\sigma_t^2},$$

Score trick

We have

$$p_t(x) = \int p_t(x \mid x_0)\, p_0(x_0)\, dx_0,$$

so differentiating under the integral:

$$\begin{aligned}\nabla \log p_t(x) &= \frac{\nabla p_t(x)}{p_t(x)} = \frac{1}{p_t(x)} \int \nabla_x p_t(x \mid x_0)\, p_0(x_0)\, dx_0 \\ &= \int \nabla_x \log p_t(x \mid x_0) \frac{p_t(x \mid x_0) p_0(x_0)}{p_t(x)}\, dx_0 = \mathbb{E}[\nabla_x \log p_t(x \mid X_0) \mid X_t = x]\,.\end{aligned}$$

For the Gaussian channel $X_t = \alpha_t X_0 + \sigma_t \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, I_d)$,

$$\nabla_x \log p_t(x \mid X_0) = -\frac{x - \alpha_t X_0}{\sigma_t^2},$$

$$\nabla \log p_t(x) = -\frac{1}{\sigma_t^2}\Big(x - \alpha_t\, \mathbb{E}[X_0 \mid X_t = x]\Big) = -\frac{1}{\sigma_t}\, \mathbb{E}[\varepsilon \mid X_t = x].$$

Learning the score by regression

From the score trick we have

$$\arg\min_{f \in L^2} \mathbb{E}\left[\left\| f(X_t) + \tfrac{1}{\sigma_t}\varepsilon \right\|^2\right] = \left(x \mapsto \mathbb{E}\left[-\tfrac{1}{\sigma_t}\varepsilon \,\middle|\, X_t = x\right]\right) = (x \mapsto \nabla \log p_t(x))$$

Learning the score by regression

From the score trick we have

$$\arg\min_{f \in L^2} \mathbb{E}\left[\left\| f(X_t) + \tfrac{1}{\sigma_t}\varepsilon \right\|^2\right] = \left(x \mapsto \mathbb{E}\left[-\tfrac{1}{\sigma_t}\varepsilon \,\middle|\, X_t = x\right]\right) = (x \mapsto \nabla \log p_t(x))$$

so for all $t > 0$

$$\nabla \log p_t \in \arg\min_{\phi_t \in L^2} \mathbb{E}_{\substack{X \sim p^\star \\ Y \sim \mathcal{N}(0, \mathrm{Id})}} \left[\| \phi_t(e^{-t}X + \sigma_t Y) + \frac{1}{\sigma_t} Y \|^2\right].$$

Learning the score by regression

From the score trick we have

$$\arg\min_{f \in L^2} \mathbb{E}\Big[\big\|f(X_t) + \tfrac{1}{\sigma_t}\varepsilon\big\|^2\Big] = \Big(x \mapsto \mathbb{E}\Big[-\tfrac{1}{\sigma_t}\varepsilon \,\Big|\, X_t = x\Big]\Big) = (x \mapsto \nabla \log p_t(x))$$

so for all $t > 0$

$$\nabla \log p_t \in \arg\min_{\phi_t \in L^2} \; \mathbb{E}_{\substack{X \sim p^\star \\ Y \sim \mathcal{N}(0,\mathrm{Id})}} \Big[\| \phi_t(e^{-t}X + \sigma_t Y) + \frac{1}{\sigma_t} Y \|^2\Big].$$

How to approximate it in practice (Denoising score matching):
Choose a neural network class $\mathcal{S}$ and compute

$$\hat{s}_t \in \arg\min_{\phi_t \in \mathcal{S}} \; \frac{1}{n}\sum_{i=1}^{n} \| \phi_t(e^{-t}X_i + \sigma_t Y_i) + \frac{1}{\sigma_t} Y_i \|^2,$$

with $X_i \sim p^\star$, $Y_i \sim \mathcal{N}(0, \mathrm{Id})$ and $\sigma_t = \sqrt{1 - e^{-2t}}$.

The algorithm in practice

Training phase
We learn a score estimator $\hat{s}(t, x) \approx \nabla \log p_t(x)$.

The algorithm in practice

Training phase
We learn a score estimator $\hat{s}(t, x) \approx \nabla \log p_t(x)$.

For each gradient step:

1. Sample X_i from the dataset, $t \sim \mathrm{Unif}([0, T])$ and $\varepsilon \sim \mathcal{N}(0, I_d)$.
2. Form a noisy sample
$$\tilde{X}_t = e^{-t} X_i + \sigma_t \varepsilon, \qquad \sigma_t = \sqrt{1 - e^{-2t}}.$$
3. Update the network by doing one gradient step on
$$\left\| \hat{s}(t, \tilde{X}_t) + \tfrac{1}{\sigma_t} \varepsilon \right\|^2.$$

The algorithm in practice

Training phase
We learn a score estimator $\hat{s}(t, x) \approx \nabla \log p_t(x)$.

For each gradient step:

1. Sample X_i from the dataset, $t \sim \mathrm{Unif}([0, T])$ and $\varepsilon \sim \mathcal{N}(0, I_d)$.
2. Form a noisy sample
$$\tilde{X}_t = e^{-t} X_i + \sigma_t \varepsilon, \qquad \sigma_t = \sqrt{1 - e^{-2t}}.$$
3. Update the network by doing one gradient step on
$$\left\| \hat{s}(t, \tilde{X}_t) + \tfrac{1}{\sigma_t} \varepsilon \right\|^2.$$

Sampling phase
We sample $\bar{X}_N$ which has law close to $p^\star$.

The algorithm in practice

Training phase
We learn a score estimator $\hat{s}(t, x) \approx \nabla \log p_t(x)$.

For each gradient step:

1. Sample X_i from the dataset, $t \sim \mathrm{Unif}([0, T])$ and $\varepsilon \sim \mathcal{N}(0, I_d)$.
2. Form a noisy sample
$$\tilde{X}_t = e^{-t} X_i + \sigma_t \varepsilon, \qquad \sigma_t = \sqrt{1 - e^{-2t}}.$$
3. Update the network by doing one gradient step on
$$\left\| \hat{s}(t, \tilde{X}_t) + \tfrac{1}{\sigma_t} \varepsilon \right\|^2.$$

Sampling phase
We sample $\bar{X}_N$ which has law close to $p^\star$.

1. Initialize from pure noise $\bar{X}_0 \sim \mathcal{N}(0, I_d)$.
2. Discretize time $0 = t_0 < t_1 < \cdots < t_N < T$.
3. For $k = 0, \ldots, N-1$, simulate the reverse SDE
$$\bar{X}_{k+1} = \bar{X}_k + \Big(\bar{X}_k + 2\hat{s}\big(T - t_k, \bar{X}_k\big) \Big) \Delta t + \sqrt{2} \sqrt{\Delta t}\, \xi_k,$$
with $\xi_k \sim \mathcal{N}(0, I_d)$.

Understanding the error

Four levels of approximation

$$\begin{cases} dX_t = \Big(X_t + (1+b_t^2)\, s(T-t, X_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, \qquad X_0 \sim p_T, \end{cases}$$

Error decomposition

Understanding the error

Four levels of approximation

$$\begin{cases} dX_t = \Big(X_t + (1+b_t^2)\, s(T-t, X_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & X_0 \sim p_T, \\ d\tilde{X}_t = \Big(\tilde{X}_t + (1+b_t^2)\, s(T-t, \tilde{X}_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & \tilde{X}_0 \sim \mathcal{N}(0, I_d), \end{cases}$$

Error decomposition

Understanding the error

Four levels of approximation

$$\begin{cases} dX_t = \Big(X_t + (1+b_t^2)\, s(T-t, X_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & X_0 \sim p_T, \\ d\tilde{X}_t = \Big(\tilde{X}_t + (1+b_t^2)\, s(T-t, \tilde{X}_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & \tilde{X}_0 \sim \mathcal{N}(0, I_d), \\ d\hat{X}_t = \Big(\hat{X}_t + (1+b_t^2)\, \hat{s}(T-t, \hat{X}_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & \hat{X}_0 \sim \mathcal{N}(0, I_d), \end{cases}$$

Error decomposition

Understanding the error

Four levels of approximation

$$\begin{cases} dX_t = \Big(X_t + (1+b_t^2)\, s(T-t, X_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & X_0 \sim p_T, \\ d\tilde{X}_t = \Big(\tilde{X}_t + (1+b_t^2)\, s(T-t, \tilde{X}_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & \tilde{X}_0 \sim \mathcal{N}(0, I_d), \\ d\hat{X}_t = \Big(\hat{X}_t + (1+b_t^2)\, \hat{s}(T-t, \hat{X}_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & \hat{X}_0 \sim \mathcal{N}(0, I_d), \\ \bar{X}_{k+1} = \bar{X}_k + \Big(\bar{X}_k + 2\,\hat{s}(T-t_k, \bar{X}_k)\Big)\Delta t + \sqrt{2\,\Delta t}\,\xi_k, & \bar{X}_0 \sim \mathcal{N}(0, I_d). \end{cases}$$

Error decomposition

Understanding the error

Four levels of approximation

$$\begin{cases} dX_t = \Big(X_t + (1+b_t^2)\, s(T-t, X_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & X_0 \sim p_T, \\ d\tilde{X}_t = \Big(\tilde{X}_t + (1+b_t^2)\, s(T-t, \tilde{X}_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & \tilde{X}_0 \sim \mathcal{N}(0, I_d), \\ d\hat{X}_t = \Big(\hat{X}_t + (1+b_t^2)\, \hat{s}(T-t, \hat{X}_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & \hat{X}_0 \sim \mathcal{N}(0, I_d), \\ \bar{X}_{k+1} = \bar{X}_k + \Big(\bar{X}_k + 2\,\hat{s}(T-t_k, \bar{X}_k)\Big)\Delta t + \sqrt{2\,\Delta t}\,\xi_k, & \bar{X}_0 \sim \mathcal{N}(0, I_d). \end{cases}$$

Error decomposition

$$W_2\big(p^\star, \mathcal{L}(\bar{X}_N)\big) = W_2\big(\mathcal{L}(X_T), \mathcal{L}(\bar{X}_N)\big)$$

Understanding the error

Four levels of approximation

$$\begin{cases} dX_t = \Big(X_t + (1+b_t^2)\, s(T-t, X_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & X_0 \sim p_T, \\ d\tilde{X}_t = \Big(\tilde{X}_t + (1+b_t^2)\, s(T-t, \tilde{X}_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & \tilde{X}_0 \sim \mathcal{N}(0, I_d), \\ d\hat{X}_t = \Big(\hat{X}_t + (1+b_t^2)\, \hat{s}(T-t, \hat{X}_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & \hat{X}_0 \sim \mathcal{N}(0, I_d), \\ \bar{X}_{k+1} = \bar{X}_k + \Big(\bar{X}_k + 2\,\hat{s}(T-t_k, \bar{X}_k)\Big)\Delta t + \sqrt{2\,\Delta t}\,\xi_k, & \bar{X}_0 \sim \mathcal{N}(0, I_d). \end{cases}$$

Error decomposition

$$\begin{aligned} W_2\big(p^\star, \mathcal{L}(\bar{X}_N)\big) &= W_2\big(\mathcal{L}(X_T), \mathcal{L}(\bar{X}_N)\big) \\ &\leq \underbrace{W_2\big(\mathcal{L}(X_T), \mathcal{L}(X_{t_N})\big)}_{\text{early stopping}} + \end{aligned}$$

Understanding the error

Four levels of approximation

$$\begin{cases} dX_t = \Big(X_t + (1+b_t^2)\, s(T-t, X_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & X_0 \sim p_T, \\ d\tilde{X}_t = \Big(\tilde{X}_t + (1+b_t^2)\, s(T-t, \tilde{X}_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & \tilde{X}_0 \sim \mathcal{N}(0, I_d), \\ d\hat{X}_t = \Big(\hat{X}_t + (1+b_t^2)\, \hat{s}(T-t, \hat{X}_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & \hat{X}_0 \sim \mathcal{N}(0, I_d), \\ \bar{X}_{k+1} = \bar{X}_k + \Big(\bar{X}_k + 2\,\hat{s}(T-t_k, \bar{X}_k)\Big)\Delta t + \sqrt{2\,\Delta t}\,\xi_k, & \bar{X}_0 \sim \mathcal{N}(0, I_d). \end{cases}$$

Error decomposition

$$\begin{aligned} W_2\big(p^\star, \mathcal{L}(\bar{X}_N)\big) &= W_2\big(\mathcal{L}(X_T), \mathcal{L}(\bar{X}_N)\big) \\ &\leq \underbrace{W_2\big(\mathcal{L}(X_T), \mathcal{L}(X_{t_N})\big)}_{\text{early stopping}} + \underbrace{W_2\big(\mathcal{L}(X_{t_N}), \mathcal{L}(\tilde{X}_{t_N})\big)}_{\text{finite horizon}} + \end{aligned}$$

Understanding the error

Four levels of approximation

$$\begin{cases} dX_t = \Big(X_t + (1+b_t^2)\, s(T-t, X_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & X_0 \sim p_T, \\ d\tilde{X}_t = \Big(\tilde{X}_t + (1+b_t^2)\, s(T-t, \tilde{X}_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & \tilde{X}_0 \sim \mathcal{N}(0, I_d), \\ d\hat{X}_t = \Big(\hat{X}_t + (1+b_t^2)\, \hat{s}(T-t, \hat{X}_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & \hat{X}_0 \sim \mathcal{N}(0, I_d), \\ \bar{X}_{k+1} = \bar{X}_k + \Big(\bar{X}_k + 2\, \hat{s}(T-t_k, \bar{X}_k)\Big)\Delta t + \sqrt{2\,\Delta t}\, \xi_k, & \bar{X}_0 \sim \mathcal{N}(0, I_d). \end{cases}$$

Error decomposition

$$\begin{aligned} W_2\big(p^\star, \mathcal{L}(\bar{X}_N)\big) &= W_2\big(\mathcal{L}(X_T), \mathcal{L}(\bar{X}_N)\big) \\ &\leq \underbrace{W_2\big(\mathcal{L}(X_T), \mathcal{L}(X_{t_N})\big)}_{\text{early stopping}} + \underbrace{W_2\big(\mathcal{L}(X_{t_N}), \mathcal{L}(\tilde{X}_{t_N})\big)}_{\text{finite horizon}} + \underbrace{W_2\big(\mathcal{L}(\tilde{X}_{t_N}), \mathcal{L}(\hat{X}_{t_N})\big)}_{\text{score approximation}} + \end{aligned}$$

Understanding the error

Four levels of approximation

$$\begin{cases} dX_t = \Big(X_t + (1+b_t^2)\, s(T-t, X_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & X_0 \sim p_T, \\ d\tilde{X}_t = \Big(\tilde{X}_t + (1+b_t^2)\, s(T-t, \tilde{X}_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & \tilde{X}_0 \sim \mathcal{N}(0, I_d), \\ d\hat{X}_t = \Big(\hat{X}_t + (1+b_t^2)\, \hat{s}(T-t, \hat{X}_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & \hat{X}_0 \sim \mathcal{N}(0, I_d), \\ \bar{X}_{k+1} = \bar{X}_k + \Big(\bar{X}_k + 2\, \hat{s}(T-t_k, \bar{X}_k)\Big)\Delta t + \sqrt{2\,\Delta t}\, \xi_k, & \bar{X}_0 \sim \mathcal{N}(0, I_d). \end{cases}$$

Error decomposition

$$\begin{aligned} W_2\big(p^\star, \mathcal{L}(\bar{X}_N)\big) &= W_2\big(\mathcal{L}(X_T), \mathcal{L}(\bar{X}_N)\big) \\ &\leq \underbrace{W_2\big(\mathcal{L}(X_T), \mathcal{L}(X_{t_N})\big)}_{\text{early stopping}} + \underbrace{W_2\big(\mathcal{L}(X_{t_N}), \mathcal{L}(\tilde{X}_{t_N})\big)}_{\text{finite horizon}} + \underbrace{W_2\big(\mathcal{L}(\tilde{X}_{t_N}), \mathcal{L}(\hat{X}_{t_N})\big)}_{\text{score approximation}} + \underbrace{W_2\big(\mathcal{L}(\hat{X}_{t_N}), \mathcal{L}(\bar{X}_N)\big)}_{\text{time discretization}} \end{aligned}$$

Understanding the error

Four levels of approximation

$$\begin{cases} dX_t = \Big(X_t + (1+b_t^2)\, s(T-t, X_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & X_0 \sim p_T, \\ d\tilde{X}_t = \Big(\tilde{X}_t + (1+b_t^2)\, s(T-t, \tilde{X}_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & \tilde{X}_0 \sim \mathcal{N}(0, I_d), \\ d\hat{X}_t = \Big(\hat{X}_t + (1+b_t^2)\, \hat{s}(T-t, \hat{X}_t)\Big)\, dt + \sqrt{2}\, b_t\, dB_t, & \hat{X}_0 \sim \mathcal{N}(0, I_d), \\ \bar{X}_{k+1} = \bar{X}_k + \Big(\bar{X}_k + 2\,\hat{s}(T-t_k, \bar{X}_k)\Big)\Delta t + \sqrt{2\,\Delta t}\,\xi_k, & \bar{X}_0 \sim \mathcal{N}(0, I_d). \end{cases}$$

Error decomposition

$$\begin{aligned} W_2\big(p^\star, \mathcal{L}(\bar{X}_N)\big) &= W_2\big(\mathcal{L}(X_T), \mathcal{L}(\bar{X}_N)\big) \\ &\le \underbrace{W_2\big(\mathcal{L}(X_T), \mathcal{L}(X_{t_N})\big)}_{\text{early stopping}} + \underbrace{W_2\big(\mathcal{L}(X_{t_N}), \mathcal{L}(\tilde{X}_{t_N})\big)}_{\text{finite horizon}} + \underbrace{W_2\big(\mathcal{L}(\tilde{X}_{t_N}), \mathcal{L}(\hat{X}_{t_N})\big)}_{\text{score approximation}} + \underbrace{W_2\big(\mathcal{L}(\hat{X}_{t_N}), \mathcal{L}(\bar{X}_N)\big)}_{\text{time discretization}} \\ &\lesssim \sqrt{T - t_N} + e^{-T} + \mathrm{Poly}(d)\, n^{-1/d} + \frac{\sqrt{d}}{N}. \end{aligned}$$

Conditional generation

New setup:
We now observe paired data (X_i, C_i), where C_i is a condition (label, covariate, regime, side information, ...).

Goal:
For a fixed condition c, generate a new sample approximately distributed as

$$X \sim p^\star(\cdot \mid C = c).$$

Conditional generation

New setup:
We now observe paired data (X_i, C_i), where C_i is a condition (label, covariate, regime, side information, ...).

Goal:
For a fixed condition c, generate a new sample approximately distributed as

$$X \sim p^\star(\cdot \mid C = c).$$

Unconditional diffusion: $\nabla_x \log p_t(x)$ $\quad\Longrightarrow\quad$ Conditional diffusion: $\nabla_x \log p_t(x \mid c)$.

Conditional denoising score matching

For the Gaussian channel

$$X_t = e^{-t} X_0 + \sigma_t \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, I_d),$$

the score trick becomes

$$\nabla_x \log p_t(x \mid c) = \mathbb{E}\left[-\frac{1}{\sigma_t} \varepsilon \;\middle|\; X_t = x,\ C = c \right].$$

Conditional denoising score matching

For the Gaussian channel

$$X_t = e^{-t} X_0 + \sigma_t \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, I_d),$$

the score trick becomes

$$\nabla_x \log p_t(x \mid c) = \mathbb{E}\left[-\frac{1}{\sigma_t} \varepsilon \;\middle|\; X_t = x,\ C = c \right].$$

Hence the conditional score is characterized by the regression problem

$$s^\star \in \arg\min_f \mathbb{E}\left[\left\| f(t, X_t, C) + \frac{1}{\sigma_t} \varepsilon \right\|^2 \right].$$

Conditional denoising score matching

For the Gaussian channel

$$X_t = e^{-t} X_0 + \sigma_t \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, I_d),$$

the score trick becomes

$$\nabla_x \log p_t(x \mid c) = \mathbb{E}\left[-\frac{1}{\sigma_t} \varepsilon \;\middle|\; X_t = x,\ C = c \right].$$

Hence the conditional score is characterized by the regression problem

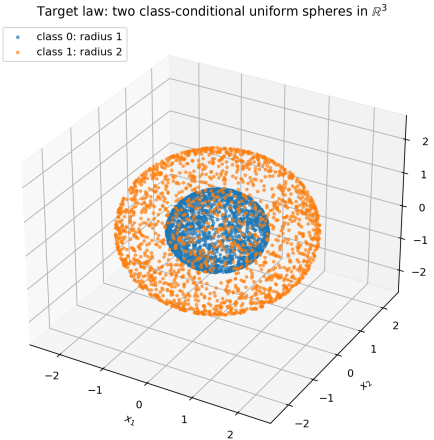
$$s^\star \in \arg\min_f \mathbb{E}\left[\left\| f(t, X_t, C) + \frac{1}{\sigma_t} \varepsilon \right\|^2 \right].$$

Empirical loss:

$$\hat{s} \in \arg\min_{\phi \in \mathcal{S}} \frac{1}{n} \sum_{i=1}^{n} \left\| \phi\big(t_i, e^{-t_i} X_i + \sigma_{t_i} \varepsilon_i, C_i\big) + \frac{1}{\sigma_{t_i}} \varepsilon_i \right\|^2.$$

Sampling: for a fixed c, run the reverse SDE with $\hat{s}(t, x, c)$ in place of $\hat{s}(t, x)$.

Conditional score-based generation on two spheres



Conditional score-based generation on two spheres

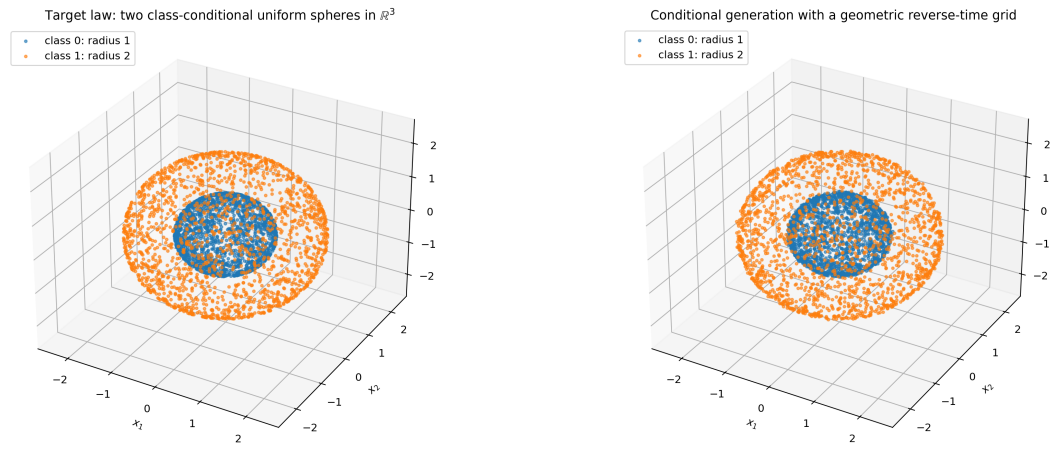


Figure 2: Left: target conditional distributions (uniform on two spheres in $\mathbb{R}^3$). Right: samples generated by a conditional score-based model.