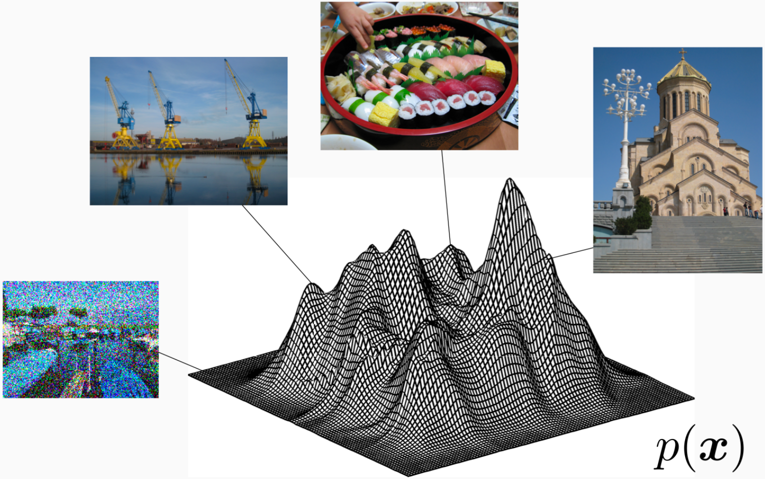
Introduction to diffusion models and study of their restriction to the Gaussian case

Émile Pierret, supervised by Bruno Galerne
May, 27^{th}, CANUM 2024

Institut Denis Poisson – Université d'Orléans, Université de Tours, CNRS

Introduction to generative models

What is a generative model ?



Variational Auto-Encoder (VAE)

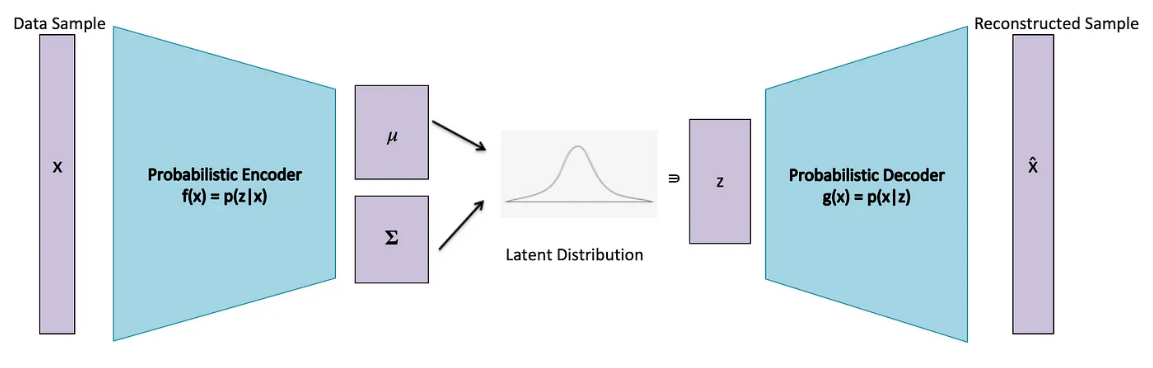


Image extracted from this url

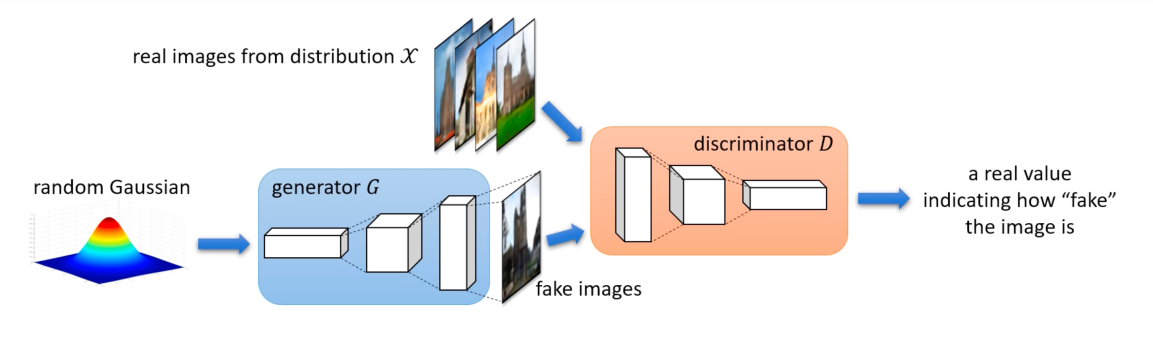


Image extracted from this url

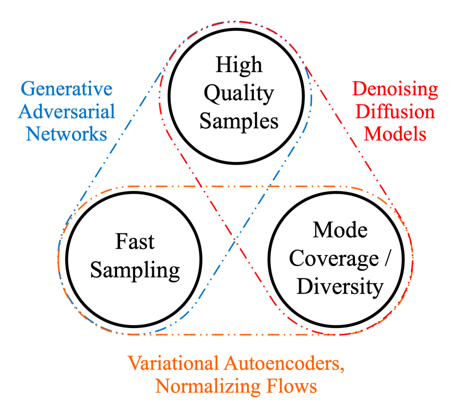
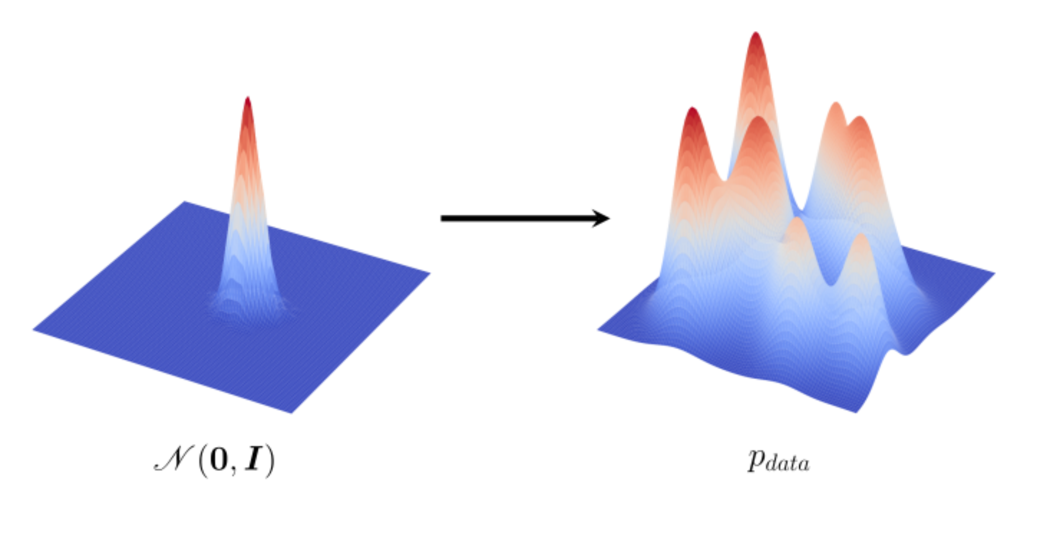


Image extrated from [Xiao et al., 2022][1]

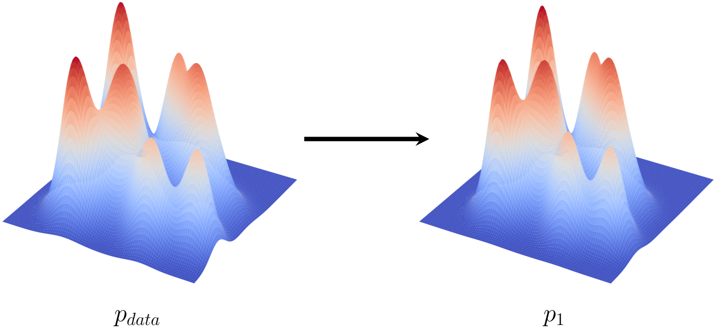
- Dhariwal, P., & Nichol, A. (2021). Diffusion models beat GANs on image synthesis. *Advances in Neural Information Processing Systems*

[1] Xiao, Z., Kreis, K., & Vahdat, A. (2022). Tackling the generative learning trilemma with denoising diffusion GANs. *International Conference on Learning Representations*

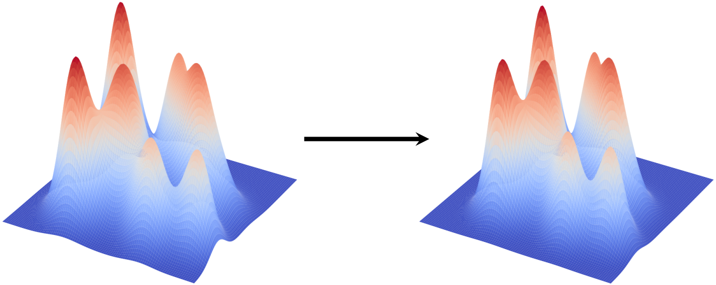
Main idea



The forward process of Denoising diffusion models

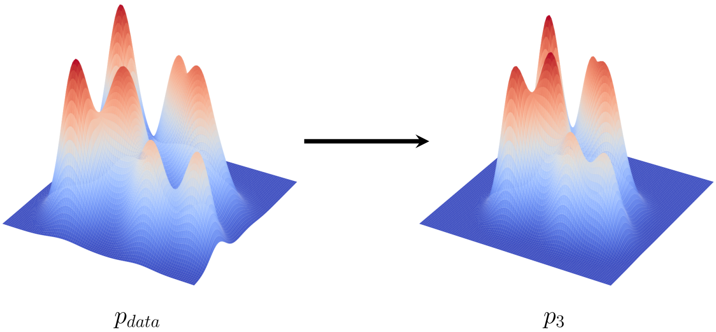


The forward process of Denoising diffusion models

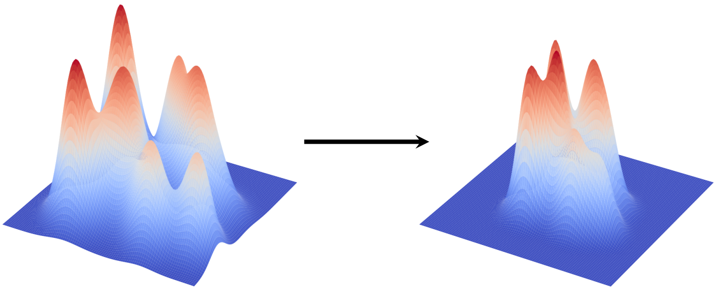


p_{data} $\longrightarrow$ p_2

The forward process of Denoising diffusion models

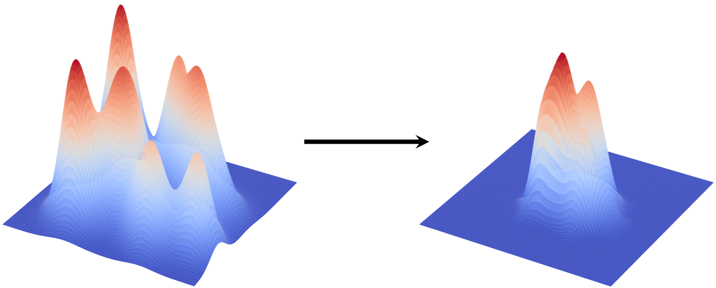


The forward process of Denoising diffusion models



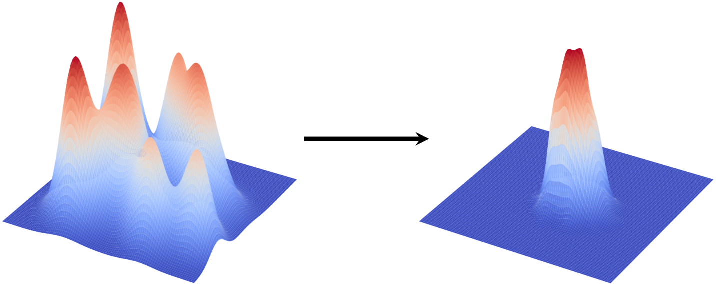
p_{data} p_4

The forward process of Denoising diffusion models



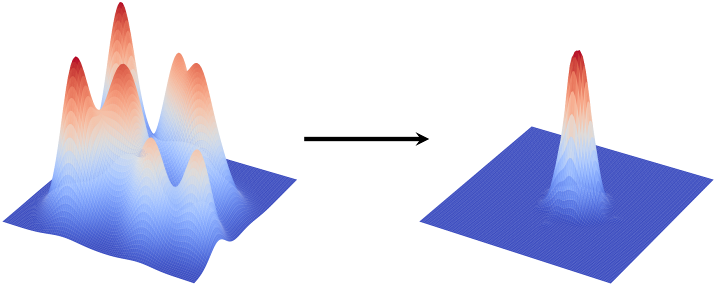
p_{data} p_5

The forward process of Denoising diffusion models



p_{data} $\quad$ p_6

The forward process of Denoising diffusion models



p_{data} p_7

Diffusion models through SDE

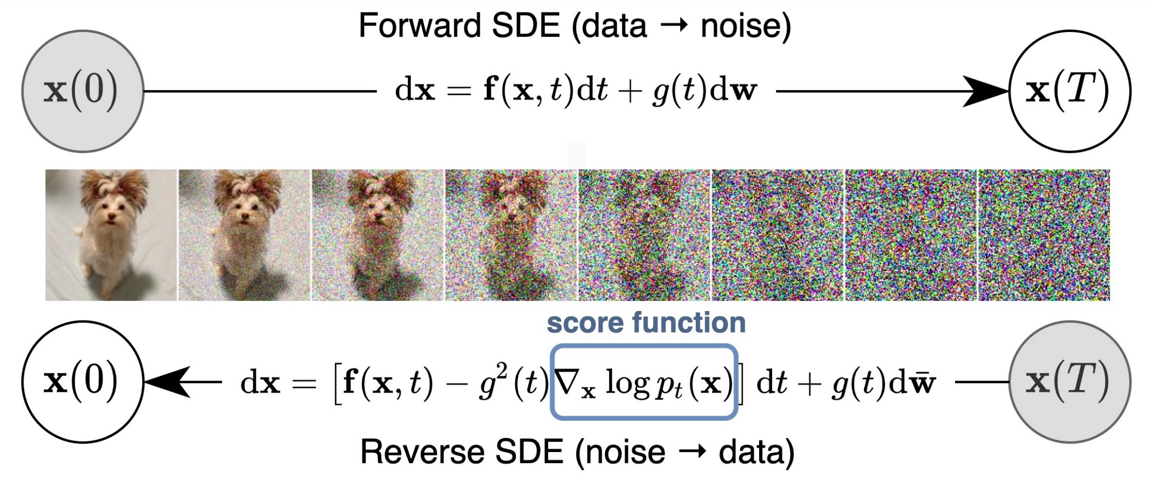


Image extracted from [Song et al., 2021]

The forward process

$$d\boldsymbol{x}_t = -\beta_t \boldsymbol{x}_t dt + \sqrt{2\beta_t} d\boldsymbol{w}_t, \quad 0 \leqslant t \leqslant T, \quad \boldsymbol{x}_0 \sim p_{\text{data}} \tag{1}$$

where β_t is an affine non-decreasing function. We denote $(p_t)_{0<\leqslant t \leqslant T}$ the density of $\boldsymbol{x}_t$.

The forward process

$$d\boldsymbol{x}_t = -\beta_t \boldsymbol{x}_t dt + \sqrt{2\beta_t} d\boldsymbol{w}_t, \quad 0 \leqslant t \leqslant T, \quad \boldsymbol{x}_0 \sim p_{\mathrm{data}} \tag{1}$$

where β_t is an affine non-decreasing function. We denote $(p_t)_{0<\leqslant t \leqslant T}$ the density of $\boldsymbol{x}_t$.

By considering $\boldsymbol{z}_t = e^{B_t}\boldsymbol{x}_t$ where $B_t = \int_0^t \beta_s ds$

The forward process

$$d\boldsymbol{x}_t = -\beta_t \boldsymbol{x}_t dt + \sqrt{2\beta_t} d\boldsymbol{w}_t, \quad 0 \leqslant t \leqslant T, \quad \boldsymbol{x}_0 \sim p_{\mathrm{data}} \tag{1}$$

where β_t is an affine non-decreasing function. We denote $(p_t)_{0<\leqslant t \leqslant T}$ the density of $\boldsymbol{x}_t$.

By considering $\boldsymbol{z}_t = e^{B_t}\boldsymbol{x}_t$ where $B_t = \int_0^t \beta_s ds$

$$\begin{aligned} d\boldsymbol{z}_t &= \beta_t e^{B_t}\boldsymbol{x}_t + e^{B_t} d\boldsymbol{x}_t \\ &= \sqrt{2\beta_t} e^{B_t} d\boldsymbol{w}_t. \end{aligned}$$

The forward process

$$d\boldsymbol{x}_t = -\beta_t \boldsymbol{x}_t dt + \sqrt{2\beta_t} d\boldsymbol{w}_t, \quad 0 \leqslant t \leqslant T, \quad \boldsymbol{x}_0 \sim p_{\text{data}} \tag{1}$$

where β_t is an affine non-decreasing function. We denote $(p_t)_{0<\leqslant t\leqslant T}$ the density of $\boldsymbol{x}_t$.

By considering $\boldsymbol{z}_t = e^{B_t}\boldsymbol{x}_t$ where $B_t = \int_0^t \beta_s ds$

$$\begin{aligned} d\boldsymbol{z}_t &= \beta_t e^{B_t} \boldsymbol{x}_t + e^{B_t} d\boldsymbol{x}_t \\ &= \sqrt{2\beta_t} e^{B_t} d\boldsymbol{w}_t. \end{aligned}$$

and for $0 \leqslant t \leqslant T$,

$$\boldsymbol{x}_t = e^{-B_t}\boldsymbol{z}_t = e^{-B_t}\boldsymbol{x}_0 + e^{-B_t}\int_0^t e^{B_s}\sqrt{2\beta_s} d\boldsymbol{w}_s = e^{-B_t}\boldsymbol{x}_0 + \boldsymbol{\eta}_t. \tag{2}$$

with $\eta_t \sim \mathcal{N}(\boldsymbol{0}, \left(1 - e^{-2B_t}\right)\boldsymbol{I})$. In particular, $\boldsymbol{\Sigma}_t := \operatorname{Cov}(\boldsymbol{x}_t) = e^{-2B_t}\operatorname{Cov}(\boldsymbol{x}_0) + \left(1 - e^{-2B_t}\right)\boldsymbol{I}$.

$$d\boldsymbol{x}_t = -\beta_t \boldsymbol{x}_t dt + \sqrt{2\beta_t} d\boldsymbol{w}_t, \quad 0 \leqslant t \leqslant T, \quad \boldsymbol{x}_0 \sim p_{\text{data}} \tag{1}$$

where β_t is an affine non-decreasing function. We denote $(p_t)_{0<\leqslant t \leqslant T}$ the density of $\boldsymbol{x}_t$.

By considering $\boldsymbol{z}_t = e^{B_t}\boldsymbol{x}_t$ where $B_t = \int_0^t \beta_s ds$

$$\begin{aligned} d\boldsymbol{z}_t &= \beta_t e^{B_t} \boldsymbol{x}_t + e^{B_t} d\boldsymbol{x}_t \\ &= \sqrt{2\beta_t} e^{B_t} d\boldsymbol{w}_t. \end{aligned}$$

and for $0 \leqslant t \leqslant T$,

$$\boldsymbol{x}_t = e^{-B_t}\boldsymbol{z}_t = e^{-B_t}\boldsymbol{x}_0 + e^{-B_t}\int_0^t e^{B_s}\sqrt{2\beta_s} d\boldsymbol{w}_s = e^{-B_t}\boldsymbol{x}_0 + \boldsymbol{\eta}_t. \tag{2}$$

with $\eta_t \sim \mathcal{N}(\mathbf{0}, \left(1 - e^{-2B_t}\right)\boldsymbol{I})$. In particular, $\boldsymbol{\Sigma}_t := \text{Cov}\left(\boldsymbol{x}_t\right) = e^{-2B_t}\,\text{Cov}\left(\boldsymbol{x}_0\right) + \left(1 - e^{-2B_t}\right)\boldsymbol{I}$.

Consequently, if $t \to +\infty$, $\boldsymbol{x}_\infty \sim \mathcal{N}_0$

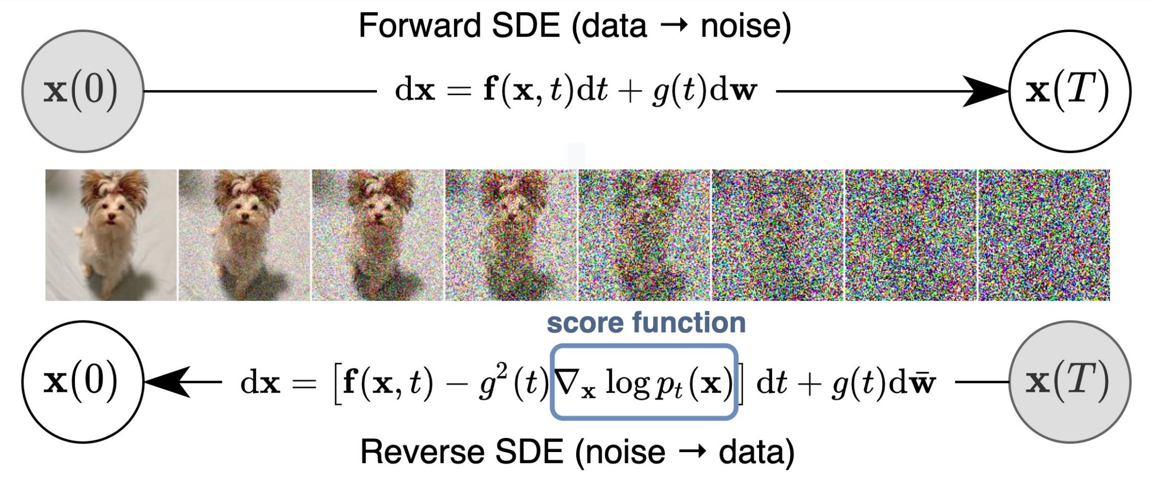


Image extracted from [Song et al., 2021]

Backward SDE

Under some assumptions on the distribution p_{data} [Pardoux, 1986][2], the backward process $(\boldsymbol{x}_{T-t})_{0 \leqslant t \leqslant T}$ verifies the backward SDE

$$d\boldsymbol{y}_t = \beta_{T-t}(\boldsymbol{y}_t + 2\nabla \log p_{T-t}(\boldsymbol{y}_t))dt + \sqrt{2\beta_{T-t}}d\overline{\boldsymbol{w}}_t, \quad 0 \leqslant t < T, \quad \boldsymbol{y}_0 \sim p_T. \tag{3}$$

[2] Pardoux, E. (1986). Grossissement d'une filtration et retournement du temps d'une diffusion. In J. Azéma & M. Yor (Eds.), *Séminaire de probabilités xx 1984/85* (pp. 48–55). Springer Berlin Heidelberg

Backward SDE

Under some assumptions on the distribution p_{data} [Pardoux, 1986][2], the backward process $(\boldsymbol{x}_{T-t})_{0\leqslant t\leqslant T}$ verifies the backward SDE

$$d\boldsymbol{y}_t = \beta_{T-t}(\boldsymbol{y}_t + 2\nabla \log p_{T-t}(\boldsymbol{y}_t))dt + \sqrt{2\beta_{T-t}}d\overline{\boldsymbol{w}}_t, \quad 0 \leqslant t < T, \quad \boldsymbol{y}_0 \sim p_T. \tag{3}$$

- $\nabla \log p_{T-t}$ is called the score function.
- The backward Brownian motion $\overline{\boldsymbol{w}}$ is not defined on the same filtration than the forward $\boldsymbol{w}$
- We are unable to derive the score function.

[2] Pardoux, E. (1986). Grossissement d'une filtration et retournement du temps d'une diffusion. In J. Azéma & M. Yor (Eds.), *Séminaire de probabilités xx 1984/85* (pp. 48–55). Springer Berlin Heidelberg

How to sample p_{data} ?

1. Learn the score function $\boldsymbol{s}_\theta(\boldsymbol{x}, t) \approx \nabla \log p_t(\boldsymbol{x})$ by applying the forward process to data and minimizing

$$\mathbb{E}_t \left\{ \mathbb{E}_t \lambda(t) \mathbb{E}_{\boldsymbol{x}_0} \mathbb{E}_{(\boldsymbol{x}_t | \boldsymbol{x}_0)} [\| s_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}_t} \log p_{0t}(\boldsymbol{x}_t \mid \boldsymbol{x}_0) \|^2] \right\}. \tag{4}$$

2. Discretize the backward SDE
   - $\boldsymbol{y}_0 \sim \mathcal{N}_0$ (and not p_T)
   - By Euler Maruyama's scheme,

$$d\boldsymbol{y}_t = \beta_{T-t}(\boldsymbol{y}_t + 2\boldsymbol{\nabla} \log \boldsymbol{p}_{\boldsymbol{T-t}}(\boldsymbol{y}_t))dt + \sqrt{2\beta_{T-t}} d\overline{\boldsymbol{w}}_t, \quad 0 \leqslant t < T, \quad \boldsymbol{y}_0 \sim p_T. \tag{5}$$

   becomes:

$$\tilde{\boldsymbol{y}}_{k+1}^{\Delta,\text{EM}} = \tilde{\boldsymbol{y}}_k^{\Delta,\text{EM}} + \Delta_t \beta_{T-t_k} \left(\tilde{\boldsymbol{y}}_k^{\Delta,\text{EM}} - 2\boldsymbol{\Sigma}_{T-t_k}^{-1} \tilde{\boldsymbol{y}}_k^{\Delta,\text{EM}} \right) + \sqrt{2\Delta_t \beta_{T-t_k}} \boldsymbol{z}_k, \ \boldsymbol{z}_k \sim \mathcal{N}_0 \tag{6}$$

The flow ODE

With a SDE can be associated an ODE

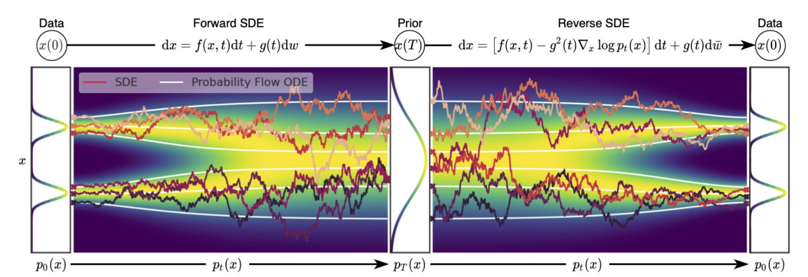


Image extracted from [Song et al., 2021][3]

[3] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021). Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*. https://openreview.net/forum?id=PxTIG12RRHS

As a reminder, the forward process is.

$$d\boldsymbol{x}_t = -\beta_t \boldsymbol{x}_t dt + \sqrt{2\beta_t} d\boldsymbol{w}_t, \quad 0 \leqslant t \leqslant T, \quad \boldsymbol{x}_0 \sim p_{\text{data}}. \tag{7}$$

With Fokker-Planck equation, we can introduce the associated flow ODE

$$d\boldsymbol{x}_t = \left[-\beta_t \boldsymbol{x}_t - \beta_t \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t)\right] dt, \quad 0 < t \leqslant T, \quad \boldsymbol{x}_0 \sim p_{\text{data}} \tag{8}$$

such that: if $\boldsymbol{y}_0 \sim p_T$ and verifies Equation (9) then for all t, $\boldsymbol{y}_t \sim p_t$.

$$d\boldsymbol{y}_t = \left[\beta_{T-t} \boldsymbol{y}_t + \beta_{T-t} \nabla_{\boldsymbol{y}} \log p_{T-t}(\boldsymbol{y}_t)\right] dt, \quad 0 \leqslant t < T. \tag{9}$$

1. Learn the score function $s_\theta(x,t) \approx \nabla \log p_t(x)$ by applying the forward process.

2. Discretize the backward SDE

- $y_0 \sim \mathcal{N}_0$ (and not p_T)
- By Euler Maruyama's scheme,

$$dy_t = \beta_{T-t}(y_t + 2\nabla \log p_{T-t}(y_t))dt + \sqrt{2\beta_{T-t}}d\overline{w}_t$$

becomes

$$\tilde{y}_{k+1}^{\Delta,\text{EM}} = \tilde{y}_k^{\Delta,\text{EM}} + \Delta_t \beta_{T-t_k}\left(\tilde{y}_k^{\Delta,\text{EM}} - 2\Sigma_{T-t_k}^{-1}\tilde{y}_k^{\Delta,\text{EM}}\right) + \sqrt{2\Delta_t\beta_{T-t_k}}z_k,\ z_k \sim \mathcal{N}_0$$

2. Discretize the flow ODE in reverse-time

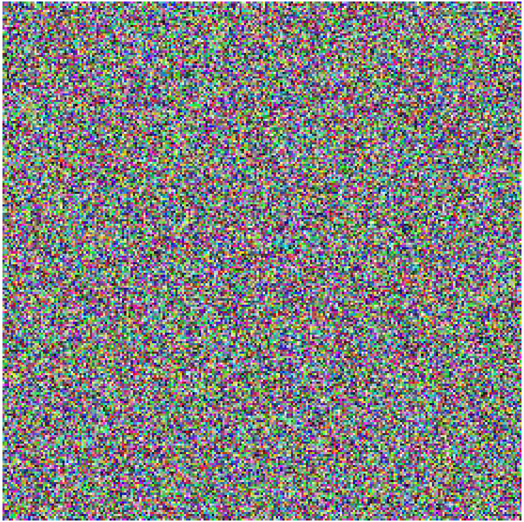
- $y_0 \sim \mathcal{N}_0$ (and not p_T)
- By Euler's scheme,

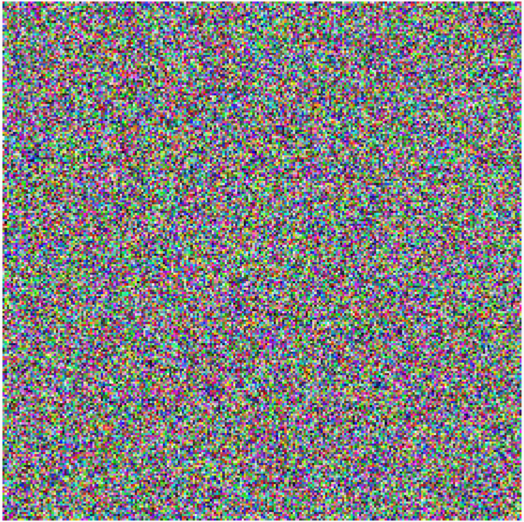
$$dy_t = [\beta_{T-t}y_t + \beta_{T-t}\nabla_y \log p_{T-t}(y_t)]\, dt$$

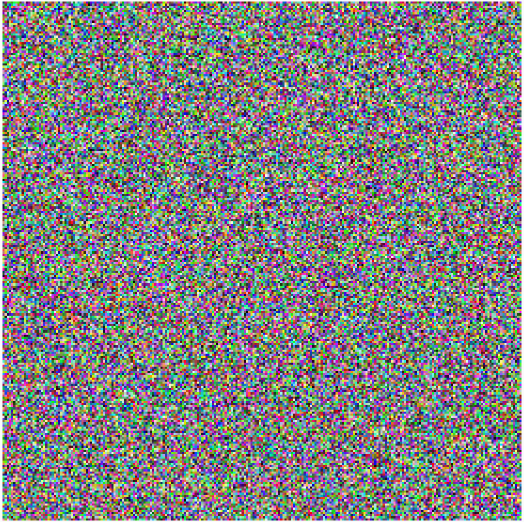
becomes

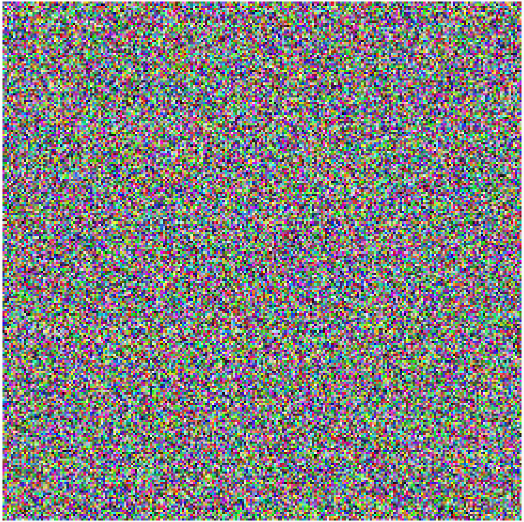
$$\widehat{y}_{k+1}^{\Delta,\text{Euler}} = \widehat{y}_k^{\Delta,\text{Euler}} + \Delta_t f(t_k, \widehat{y}_k^{\Delta,\text{Euler}})$$

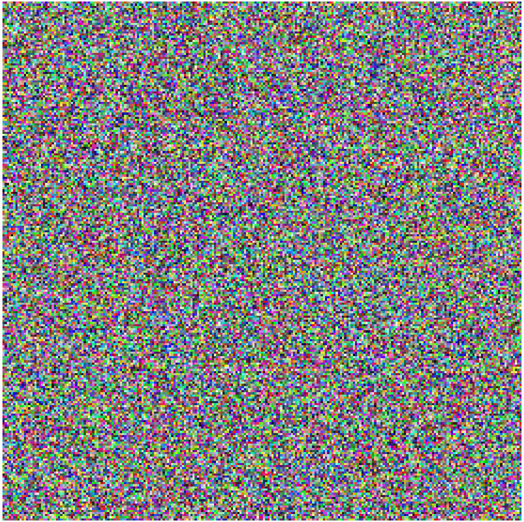
with $f(t,y) = \beta_{T-t}y - \beta_{T-t}\Sigma_{T-t}^{-1}y$

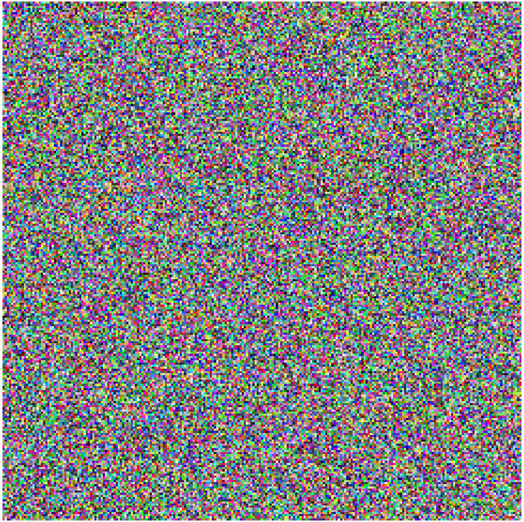


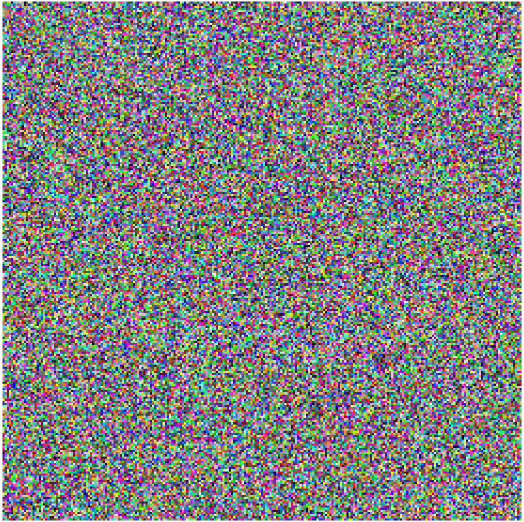
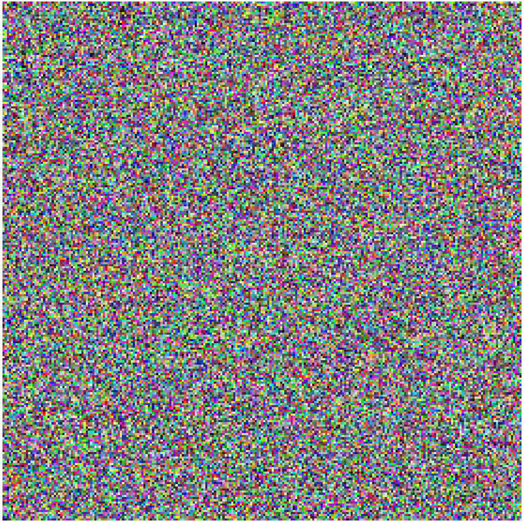


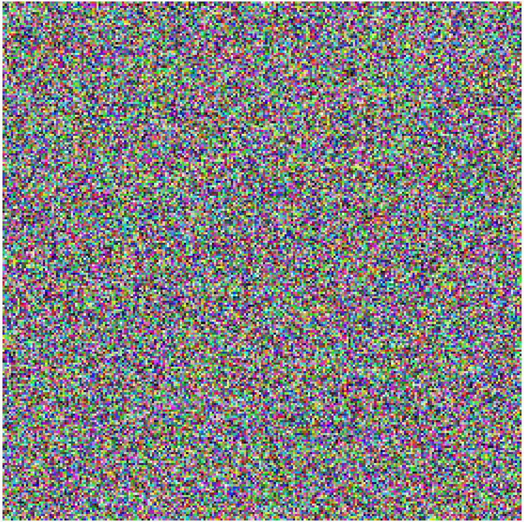


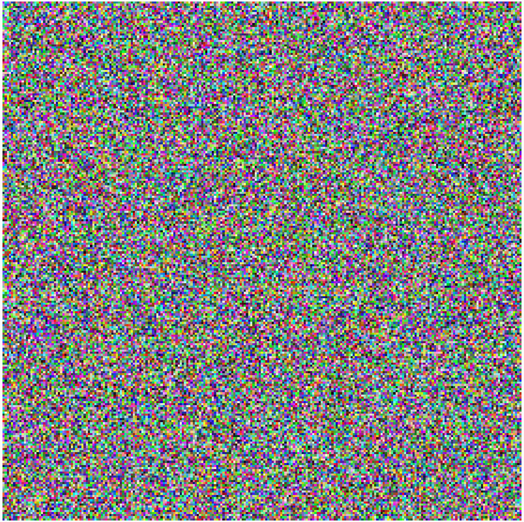


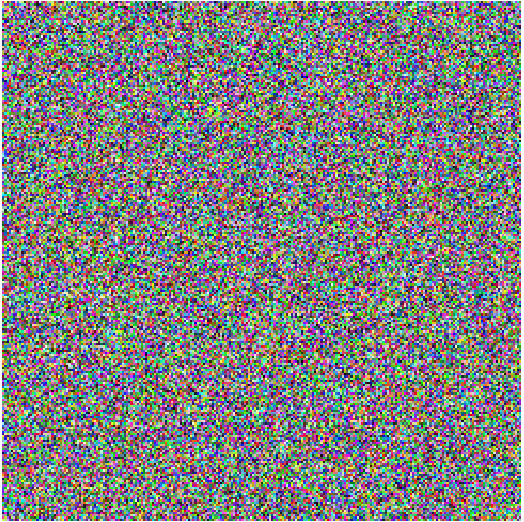


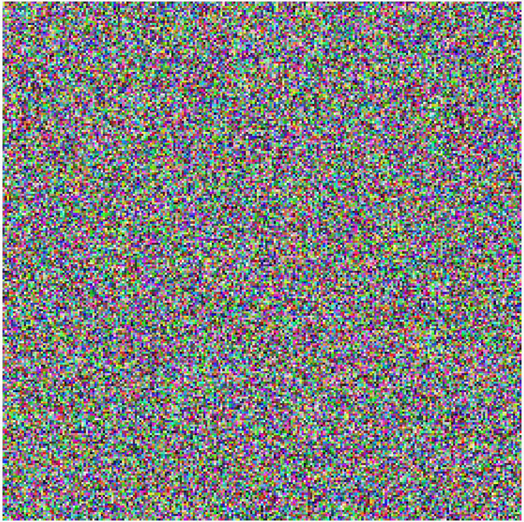


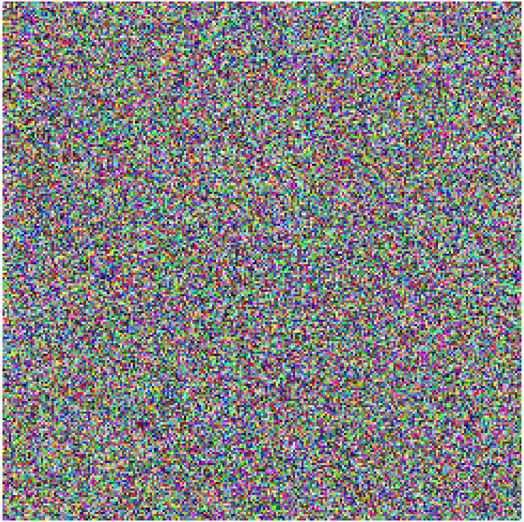
 

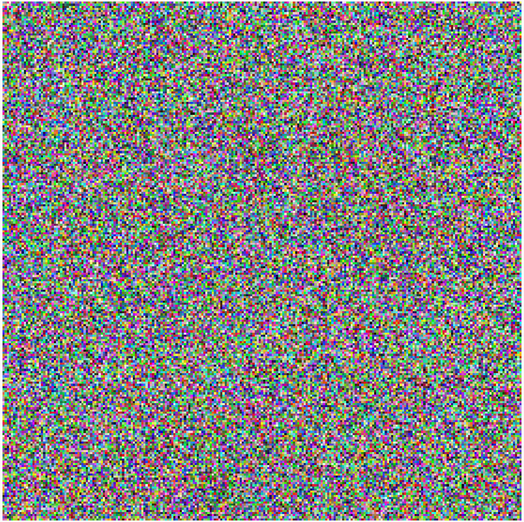


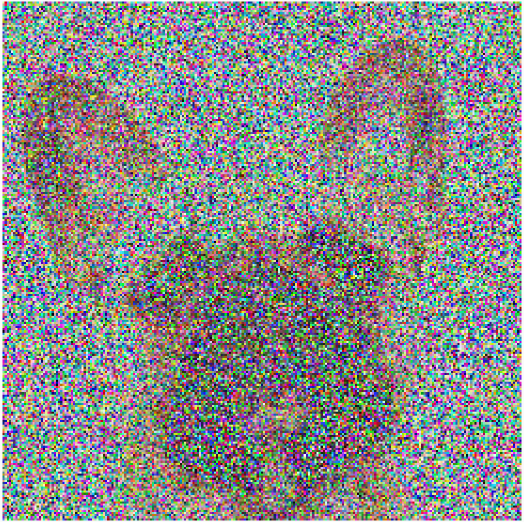


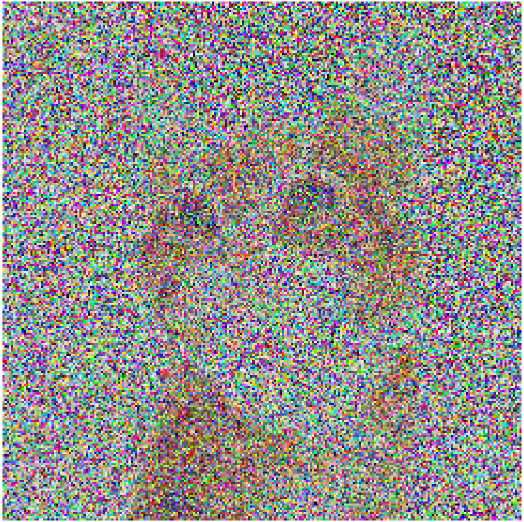


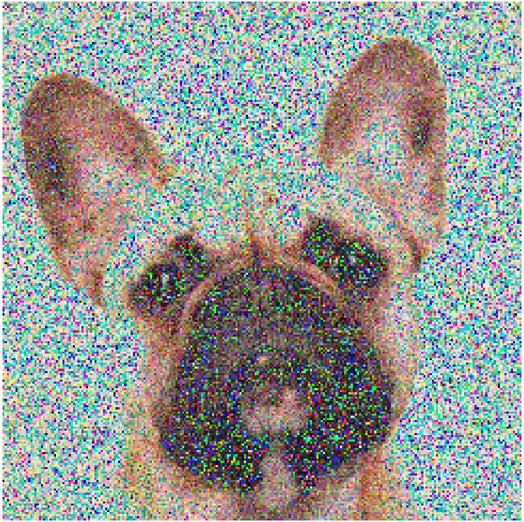


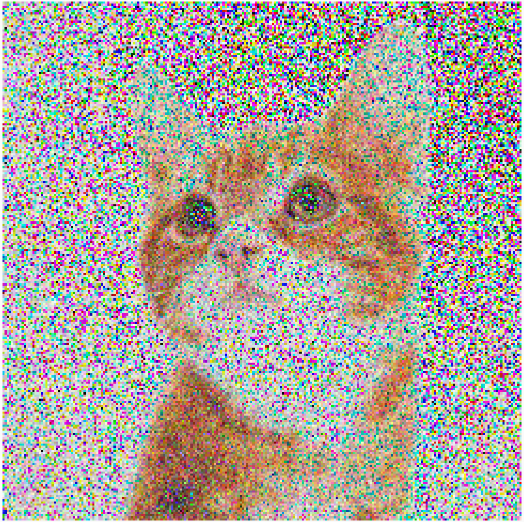


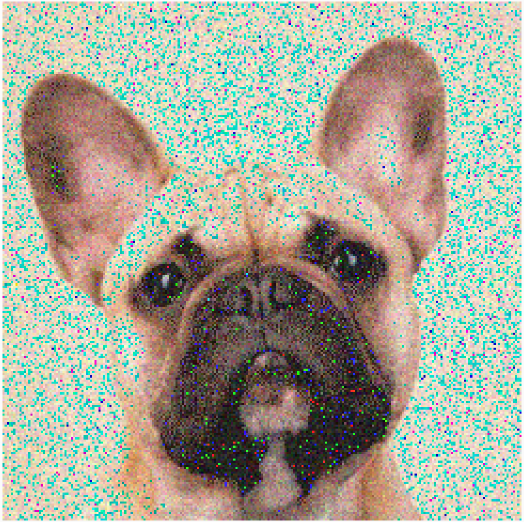


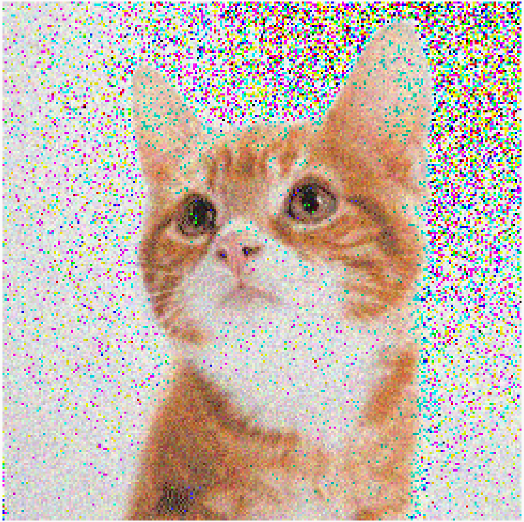













To image restoration

The sampling of p_{data} provides a prior knowledge on data to achieve restoration tasks on images (inpainting, super-resolution, deblurring,...) [Song et al., 2021],[Lugmayr et al., 2022],[Chung et al., 2022], Pseudo-inverse reasonning [Choi et al., 2021]

Study of the convergence

- Experimental study: [S. Chen, Chewi, Lee, et al., 2023; Franzese et al., 2023; Karras et al., 2022]
- Theoretical study: [Benton et al., 2024; S. Chen, Chewi, Li, et al., 2023; De Bortoli et al., 2021; Lee et al., 2022, 2024]
- Under manifold assumption: [M. Chen et al., 2023; De Bortoli, 2022; Wenliang and Moran, 2022]
- Upper bounds on the 1-Wasserstein or TV distance between the data and the model distributions by making assumptions on the L^2-error between the ideal and learned score functions and on the compacity of the support of the data
- In practice, the convergence of diffusion models is observed using the Frechet Inception Distance (FID) which is 2-Wasserstein distance between Gaussians fitted to datasets.

Error types

There are four types of error:

- The initialization error
- The discretization error
- The truncation error
- The score approximation error

The initialization error

$$d\boldsymbol{y}_t = \beta_{T-t}(\boldsymbol{y}_t + 2\nabla \log p_{T-t}(\boldsymbol{y}_t))dt + \sqrt{2\beta_{T-t}}d\boldsymbol{w}_t, \quad 0 \leqslant t < T, \quad \boldsymbol{y}_0 \sim p_T. \tag{10}$$

is replaced by:

$$d\boldsymbol{y}_t = \beta_{T-t}(\boldsymbol{y}_t + 2\nabla \log p_{T-t}(\boldsymbol{y}_t))dt + \sqrt{2\beta_{T-t}}d\boldsymbol{w}_t, \quad 0 \leqslant t < T, \quad \boldsymbol{y}_0 \sim \mathcal{N}_0. \tag{11}$$

- The resullt is: if $\boldsymbol{y}_t$ verifies Equation (14), $\boldsymbol{y}_T \sim p_{T-t}$
- Equation (15) produces another stochastic process.
- This holds also for the ODE.

The discretization error

Several choice for the discretization:

SDE schemes	Euler-Maruyama (EM)	$\begin{cases} \tilde{\boldsymbol{y}}_0^{\Delta,\text{EM}} & \sim \mathcal{N}_0 \\ \tilde{\boldsymbol{y}}_{k+1}^{\Delta,\text{EM}} & = \tilde{\boldsymbol{y}}_k^{\Delta,\text{EM}} + \Delta_t \beta_{T-t_k} \left(\tilde{\boldsymbol{y}}_k^{\Delta,\text{EM}} - 2\boldsymbol{\Sigma}_{T-t_k}^{-1} \tilde{\boldsymbol{y}}_k^{\Delta,\text{EM}} \right) + \sqrt{2\Delta_t \beta_{T-t_k}} \boldsymbol{z}_k, \ \boldsymbol{z}_k \sim \mathcal{N}_0 \end{cases}$	(12)
	Exponential integrator (EI)	$\begin{cases} \tilde{\boldsymbol{y}}_0^{\Delta,\text{EI}} & \sim \mathcal{N}_0 \\ \tilde{\boldsymbol{y}}_{k+1}^{\Delta,\text{EI}} & = \tilde{\boldsymbol{y}}_k^{\Delta,\text{EI}} + \gamma_{1,k} \left(\tilde{\boldsymbol{y}}_k^{\Delta,\text{EI}} - 2\boldsymbol{\Sigma}_{T-t_k}^{-1} \tilde{\boldsymbol{y}}_k^{\Delta,\text{EI}} \right) + \sqrt{2\gamma_{2,k}} z_k, \ \boldsymbol{z}_k \sim \mathcal{N}_0 \end{cases}$ where $\gamma_{1,k} = \exp(B_{T-t_k} - B_{T-t_{k+1}}) - 1$ and $\gamma_{2,k} = \frac{1}{2}(\exp(2B_{T-t_k} - 2B_{T-t_{k+1}}) - 1)$	(13)
ODE schemes	Explicit Euler	$\begin{cases} \hat{\boldsymbol{y}}_0^{\Delta,\text{Euler}} & \sim \mathcal{N}_0 \\ \hat{\boldsymbol{y}}_{k+1}^{\Delta,\text{Euler}} & = \hat{\boldsymbol{y}}_k^{\Delta,\text{Euler}} + \Delta_t f(t_k, \hat{\boldsymbol{y}}_k^{\Delta,\text{Euler}}) \quad \text{with } f(t,\boldsymbol{y}) = \beta_{T-t}\boldsymbol{y} - \beta_{T-t}\boldsymbol{\Sigma}_{T-t}^{-1}\boldsymbol{y} \end{cases}$	(14)
	Heun's method	$\begin{cases} \hat{\boldsymbol{y}}_0^{\Delta,\text{Heun}} & \sim \mathcal{N}_0 \\ \hat{\boldsymbol{y}}_{k+1/2}^{\Delta,\text{Heun}} & = \hat{\boldsymbol{y}}_k^{\Delta,\text{Heun}} + \Delta_t f(t_k, \hat{\boldsymbol{y}}_k^{\Delta,\text{Heun}}) \quad \text{with } f(t,\boldsymbol{y}) = \beta_{T-t}\boldsymbol{y} - \beta_{T-t}\boldsymbol{\Sigma}_{T-t}^{-1}\boldsymbol{y} \\ \hat{\boldsymbol{y}}_{k+1}^{\Delta,\text{Heun}} & = \hat{\boldsymbol{y}}_k^{\Delta,\text{Heun}} + \frac{\Delta_t}{2} \left(f(t_k, \hat{\boldsymbol{y}}_k^{\Delta,\text{Heun}}) + f(t_{k+1}, \hat{\boldsymbol{y}}_{k+1/2}^{\Delta,\text{Heun}}) \right) \end{cases}$	(15)

The truncation error

$$d\boldsymbol{y}_t = \beta_{T-t}(\boldsymbol{y}_t + 2\nabla \log p_{T-t}(\boldsymbol{y}_t))dt + \sqrt{2\beta_{T-t}}d\boldsymbol{w}_t, \quad 0 \leqslant t < T, \quad \boldsymbol{y}_0 \sim p_T. \tag{16}$$

- At time 0, p_0 does not necessary exists.
- It is preferable to solve Equation (20) from 0 to $T - \varepsilon$.
- In general, $\varepsilon = 10^{-3}$ (Karras et al., 2022; Song et al., 2021)

The score approximation error

$$d\boldsymbol{y}_t = \beta_{T-t}(\boldsymbol{y}_t + 2\nabla \log p_{T-t}(\boldsymbol{y}_t))dt + \sqrt{2\beta_{T-t}}d\boldsymbol{w}_t, \quad 0 \leqslant t < T, \quad \boldsymbol{y}_0 \sim p_T. \tag{17}$$

$$d\boldsymbol{y}_t = \beta_{T-t}(\boldsymbol{y}_t + 2\boldsymbol{s}_\theta(T-t, \boldsymbol{y}_t))dt + \sqrt{2\beta_{T-t}}d\boldsymbol{w}_t, \quad 0 \leqslant t < T, \quad \boldsymbol{y}_0 \sim p_T. \tag{18}$$

where s_θ is a neural network.

1. The most difficult to estimate theoretically.
2. In general, bounds on the L^2 norm.

Restriction to the Gaussian case

Gaussian assumption

Gaussian assumption: p_{data} is a centered Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. ($\mathbf{\Sigma}$ is not necessarily invertible)

- $p_t \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_t)$, with $\mathbf{\Sigma}_t = e^{-2B_t} \operatorname{Cov}(\boldsymbol{x}_0) + \left(1 - e^{-2B_t}\right) \boldsymbol{I}$
- $\nabla \log p_t(\boldsymbol{x}) = -\mathbf{\Sigma}_t^{-1} \boldsymbol{x}$
- Also known if p_{data} is a Gaussian mixture [Shah et al., 2023; Zach et al., 2024; Zach et al., 2023].

Note that $\nabla \log p_t$ is linear.

Gaussian assumption: p_{data} is a centered Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. ($\mathbf{\Sigma}$ is not necessarily invertible)

- $p_t \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_t)$, with $\mathbf{\Sigma}_t = e^{-2B_t} \operatorname{Cov}(\boldsymbol{x}_0) + \left(1 - e^{-2B_t}\right) \boldsymbol{I}$
- $\nabla \log p_t(\boldsymbol{x}) = -\mathbf{\Sigma}_t^{-1} \boldsymbol{x}$
- Also known if p_{data} is a Gaussian mixture [Shah et al., 2023; Zach et al., 2024; Zach et al., 2023].

Note that $\nabla \log p_t$ is linear.

Proposition 2: Characterization of Gaussian distributions through diffusion models

The three following propositions are equivalent:

(i) $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ for some covariance $\mathbf{\Sigma}$.

(ii) $\forall t > 0, \nabla_x \log p_t(\boldsymbol{x})$ is linear w.r.t $\boldsymbol{x}$.

(iii) $\exists t > 0, \nabla_x \log p_t(\boldsymbol{x})$ is linear w.r.t $\boldsymbol{x}$.

In this case, for $t > 0$, $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}) = -\mathbf{\Sigma}_t^{-1} \boldsymbol{x}$.

Explicit solution of the backward SDE

Proposition 3: Solution of the backward SDE under Gaussian assumption

Under Gaussian assumption, the strong solution to Equation (3) can be written as:

$$\boldsymbol{y}_t = e^{-(B_T - B_{T-t})} \boldsymbol{\Sigma}_{T-t} \boldsymbol{\Sigma}_T^{-1} \boldsymbol{y}_0 + \boldsymbol{\xi}_t, \quad 0 \leqslant t \leqslant T \tag{19}$$

where $\boldsymbol{\xi}_t$ is a Gaussian process. Finally:

$$\mathrm{Cov}(\boldsymbol{y}_t) = \boldsymbol{\Sigma}_{T-t} + e^{-2(B_T - B_{T-t})} \boldsymbol{\Sigma}_{T-t}^2 \boldsymbol{\Sigma}_T^{-1} \left(\boldsymbol{\Sigma}_{T-t}^{-1} \mathrm{Cov}(\boldsymbol{y}_0) \boldsymbol{\Sigma}_T^{-1} \boldsymbol{\Sigma}_{T-t} - \boldsymbol{I} \right), \tag{20}$$

and in particular, if $\mathrm{Cov}(\boldsymbol{y}_0)$ and $\boldsymbol{\Sigma}$ commute,

$$\mathrm{Cov}(\boldsymbol{y}_t) = \boldsymbol{\Sigma}_{T-t} + e^{-2(B_T - B_{T-t})} \boldsymbol{\Sigma}_{T-t}^2 \boldsymbol{\Sigma}_T^{-1} \left[\boldsymbol{\Sigma}_T^{-1} \mathrm{Cov}(\boldsymbol{y}_0) - \boldsymbol{I} \right] \tag{21}$$

- $\boldsymbol{y}_0$ can follow any law.

Proposition 4: Solution of the ODE probability flow under Gaussian assumption

The solution to the probability flow ODE (8) under Gaussian assumption corresponds to the optimal transport map between p_T and p_{data}. More precisely, for any $\boldsymbol{y}_0$,

$$\boldsymbol{y}_t = \boldsymbol{\Sigma}_T^{-1/2}\boldsymbol{\Sigma}_{T-t}^{1/2}\boldsymbol{y}_0, \quad 0 \leqslant t \leqslant T,$$

is the solution of the reverse-time ODE (9). Consequently, the covariance matrix $\mathrm{Cov}(\boldsymbol{y}_t)$ verifies

$$\mathrm{Cov}(\boldsymbol{y}_t) = \boldsymbol{\Sigma}_T^{-1/2}\boldsymbol{\Sigma}_{T-t}^{1/2}\,\mathrm{Cov}(\boldsymbol{y}_0)\boldsymbol{\Sigma}_{T-t}^{1/2}\boldsymbol{\Sigma}_T^{-1/2}, \quad 0 \leqslant t \leqslant T, \tag{22}$$

and in particular, if $\mathrm{Cov}(\boldsymbol{y}_0)$ and $\boldsymbol{\Sigma}$ commute,

$$\mathrm{Cov}(\boldsymbol{y}_t) = \boldsymbol{\Sigma}_T^{-1}\boldsymbol{\Sigma}_{T-t}\,\mathrm{Cov}(\boldsymbol{y}_0), \quad 0 \leqslant t \leqslant T. \tag{23}$$

- The relation between optimal transport and probability flow ODE (also called Fokker-Planck ODE) has been discussed in Khrulkov et al., 2023; Lavenant and Santambrogio, 2022[4] in the asymptotic case where $T \mapsto +\infty$.

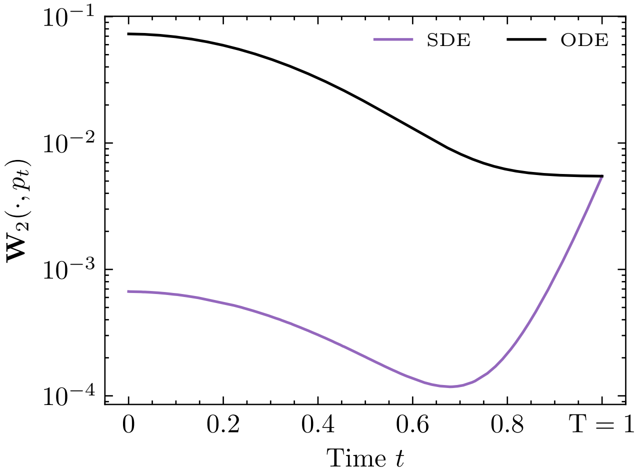
[4] Lavenant, H., & Santambrogio, F. (2022). The flow map of the fokker–planck equation does not provide optimal transport. *Applied Mathematics Letters*, *133*, 108225. https://doi.org/https://doi.org/10.1016/j.aml.2022.108225

Proposition 5: Marginals of the generative processes under Gaussian assumption

Under Gaussian assumption, $(\tilde{\boldsymbol{y}}_t)_{0\leqslant t\leqslant T}$ and $(\widehat{\boldsymbol{y}}_t)_{0\leqslant t\leqslant T}$ are Gaussian processes. At each time t, $\tilde{p}_t$ is the Gaussian distribution $\mathcal{N}(\mathbf{0}, \tilde{\boldsymbol{\Sigma}}_t)$ with $\tilde{\boldsymbol{\Sigma}}_t = \boldsymbol{\Sigma}_t + e^{-2(B_T - B_t)}\boldsymbol{\Sigma}_t^2\boldsymbol{\Sigma}_T^{-1}(\boldsymbol{\Sigma}_T^{-1} - \boldsymbol{I})$ and $\widehat{p}_t$ is the Gaussian distribution $\mathcal{N}(\mathbf{0}, \widehat{\boldsymbol{\Sigma}}_t)$ with $\widehat{\boldsymbol{\Sigma}}_t = \boldsymbol{\Sigma}_T^{-1}\boldsymbol{\Sigma}_t$. For all $0 \leqslant t \leqslant T$, the three covariance matrices $\boldsymbol{\Sigma}_t$, $\tilde{\boldsymbol{\Sigma}}_t$ and $\widehat{\boldsymbol{\Sigma}}_t$ share the same range. Furthermore, for all $0 \leqslant t \leqslant T$,

$$\mathbf{W}_2(\tilde{p}_t, p_t) \leqslant \mathbf{W}_2(\widehat{p}_t, p_t) \tag{24}$$

which shows for $t = 0$ that the SDE sampler is a better sampler than the ODE sampler when the exact score is konwn.


10^{-1}
10^{-2}
10^{-3}
10^{-4}
$\mathbf{W}_2(\cdot, p_t)$
SDE
ODE
0
0.2
0.4
0.6
0.8
T = 1
Time t

Discretization error

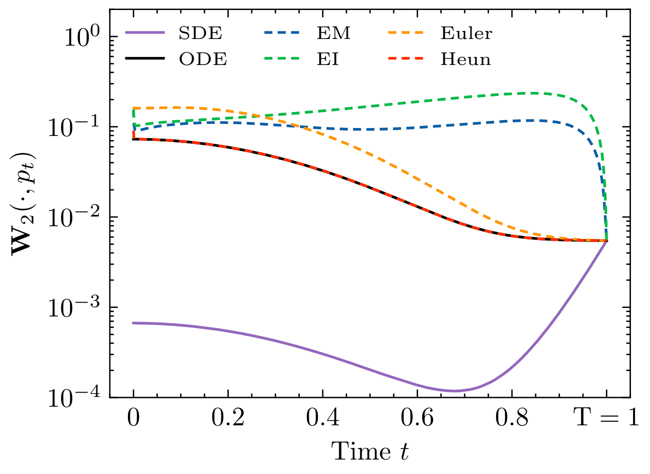
From

$$\tilde{\boldsymbol{y}}_{k+1}^{\Delta,\mathsf{EM}} = \tilde{\boldsymbol{y}}_{k}^{\Delta,\mathsf{EM}} + \Delta_t \beta_{T-t_k} \left(\tilde{\boldsymbol{y}}_{k}^{\Delta,\mathsf{EM}} - 2\boldsymbol{\Sigma}_{T-t_k}^{-1} \tilde{\boldsymbol{y}}_{k}^{\Delta,\mathsf{EM}} \right) + \sqrt{2\Delta_t \beta_{T-t_k}} \boldsymbol{z}_k, \ \boldsymbol{z}_k \sim \mathcal{N}_0$$

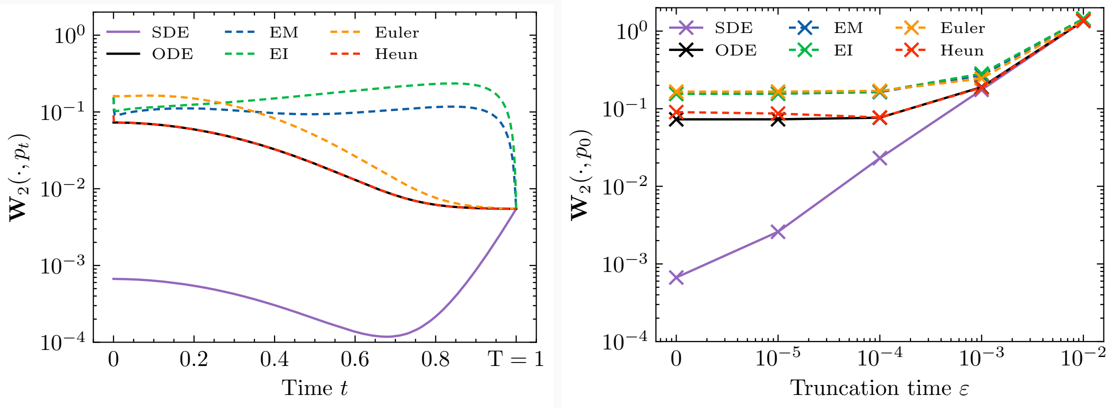
we have:

$$\lambda_i^{\mathsf{EM},k+1} = \left(1 + \Delta_t \beta_{T-t_k} (1 - \tfrac{2}{\lambda_i^{T-t_k}})\right)^2 \lambda_i^{\mathsf{EM},k} + 2\Delta_t \beta_{T-t_k}, 1 \leqslant i \leqslant d, 0 \leqslant k \leqslant N-2 \quad (25)$$

Discretization error



Truncation error

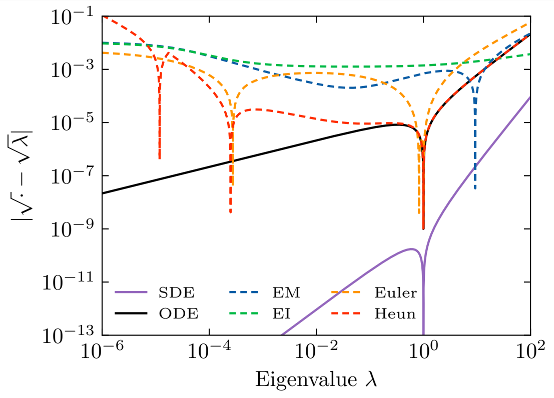


(a) Initialization error along the integration time

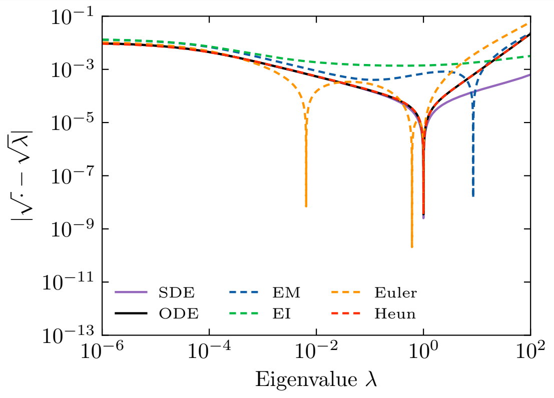
(b) Truncation error for different truncation time ε

Ablation study

		Continuous		$N = 50$		$N = 250$		$N = 500$		$N = 1000$	
		p_T	$\mathcal{N}_0$	p_T	$\mathcal{N}_0$	p_T	$\mathcal{N}_0$	p_T	$\mathcal{N}_0$	p_T	$\mathcal{N}_0$
EM	$\varepsilon = 0$	0	6.7E-4	4.77	4.77	0.65	0.65	0.31	0.31	0.15	0.16
	$\varepsilon = 10^{-5}$	2.5E-3	2.6E-3	4.77	4.77	0.65	0.65	0.31	0.31	0.16	0.16
	$\varepsilon = 10^{-3}$	0.17	0.17	4.67	4.67	0.69	0.69	0.39	0.39	0.27	0.27
	$\varepsilon = 10^{-2}$	1.35	1.35	4.56	4.56	1.69	1.69	1.50	1.50	1.42	1.42
EI	$\varepsilon = 0$	0	6.7E-4	2.81	2.81	0.57	0.57	0.30	0.30	0.16	0.16
	$\varepsilon = 10^{-5}$	2.5E-3	2.6E-3	2.81	2.81	0.57	0.57	0.30	0.30	0.16	0.16
	$\varepsilon = 10^{-3}$	0.17	0.17	2.91	2.91	0.66	0.66	0.41	0.41	0.28	0.28
	$\varepsilon = 10^{-2}$	1.35	1.35	3.93	3.93	1.76	1.76	1.55	1.55	1.45	1.45
Euler	$\varepsilon = 0$	0	0.07	1.72	1.78	0.38	0.44	0.19	0.26	0.10	0.17
	$\varepsilon = 10^{-5}$	2.5E-3	0.07	1.72	1.78	0.38	0.44	0.20	0.26	0.10	0.17
	$\varepsilon = 10^{-3}$	0.17	0.19	1.72	1.78	0.42	0.48	0.27	0.32	0.21	0.25
	$\varepsilon = 10^{-2}$	1.35	1.36	2.21	2.25	1.41	1.43	1.37	1.38	1.36	1.37
Heun	$\varepsilon = 0$	0	0.07	7.09	7.09	0.72	0.73	0.21	0.22	0.05	0.09
	$\varepsilon = 10^{-5}$	2.5E-3	0.07	6.48	6.48	0.64	0.65	0.18	0.20	0.05	0.09
	$\varepsilon = 10^{-3}$	0.17	0.19	0.56	0.57	0.13	0.15	0.16	0.18	0.17	0.19
	$\varepsilon = 10^{-2}$	1.35	1.36	1.37	1.38	1.35	1.36	1.35	1.36	1.35	1.36



(a) Initialization error at final time



(b) Truncation error at final time for $\varepsilon = 10^{-3}$

Gaussian sample

p_θ^{EM}

p_θ^{Heun}

p_θ^{Heun}

Score approximation

	Exact score distribution			Learned score distribution	
p	$\mathbf{W}_2(p, p_{\text{data}}) \downarrow$	$\mathbf{W}_2^{\text{emp.}}(p^{\text{emp.}}, p_{\text{data}}) \downarrow$	$\text{FID}(p^{\text{emp.}}, p_{\text{data}}^{\text{emp.}}) \downarrow$	$\mathbf{W}_2^{\text{emp.}}(p_\theta^{\text{emp.}}, p_{\text{data}}^{\text{emp.}}) \downarrow$	$\text{FID}(p_\theta^{\text{emp.}}, p_{\text{data}}^{\text{emp.}}) \downarrow$
EM	5.16	5.1630±7E-5	0.0891±8E-4	15.6	1.02
Heun	3.73	3.7323±2E-4	0.0447±6E-4	56.7	19.4

- Heun's method fails.
- EM discretization more resilient to score approximation.

Conclusion

- This theoretical analysis led to conclude that Heun's scheme is the best numerical solution, in accordance with empirical previous work [Karras et al., 2022].
- We conducted an empirical analysis with a learned score function using standard architecture which showed the most important one in practice.
- This suggests that assessing the quality of learned score functions is an important research direction for future work.

The end

Thank you for your attention !

Preprint : Diffusion models for Gaussian distributions: Exact solutions and Wasserstein errors, E. Pierret, B. Galerne, 2024, hal, Arxiv

References

Benton, J., Bortoli, V. D., Doucet, A., & Deligiannidis, G. (2024). Nearly d-linear convergence bounds for diffusion models via stochastic localization. *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=r5njV3BsuD

Chen, M., Huang, K., Zhao, T., & Wang, M. (2023). Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, & J. Scarlett (Eds.), *Proceedings of the 40th international conference on machine learning* (pp. 4672–4712). PMLR. https://proceedings.mlr.press/v202/chen23o.html

Chen, S., Chewi, S., Lee, H., Li, Y., Lu, J., & Salim, A. (2023). The probability flow ODE is provably fast. *Thirty-seventh Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=KD6MFeWSAd

Chen, S., Chewi, S., Li, J., Li, Y., Salim, A., & Zhang, A. (2023). Sampling is as easy as learning the score: Theory for diffusion models with minimal data assumptions. *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=zyLVMgsZ0U_

Choi, J., Kim, S., Jeong, Y., Gwon, Y., & Yoon, S. (2021). ILVR: Conditioning method for denoising diffusion probabilistic models. *ILVR*, 14367–14376. Retrieved 2022-11-28, from https://openaccess.thecvf.com/content/ICCV2021/html/Choi_ILVR_Conditioning_Method_for_Denoising_Diffusion_Probabilistic_Models_ICCV_2021_paper.html

Chung, H., Sim, B., Ryu, D., & Ye, J. C. (2022). Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems (NeurIPS)*.

De Bortoli, V. (2022). Convergence of denoising diffusion models under the manifold hypothesis. *Transactions on Machine Learning Research*. https://openreview.net/forum?id=MhK5aXo3gB

De Bortoli, V., Thornton, J., Heng, J., & Doucet, A. (2021). Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, *34*, 17695–17709. Retrieved 2022-11-08, from https://papers.nips.cc/paper/2021/hash/940392f5f32a7ade1cc201767cf83e31-Abstract.html

Dhariwal, P., & Nichol, A. (2021). Diffusion models beat GANs on image synthesis. *Advances in Neural Information Processing Systems*.

Franzese, G., Rossi, S., Yang, L., Finamore, A., Rossi, D., Filippone, M., & Michiardi, P. (2023). How much is enough? a study on diffusion times in score-based generative models. *Entropy*, *25*(4). https://doi.org/10.3390/e25040633

Karras, T., Aittala, M., Aila, T., & Laine, S. (2022). Elucidating the design space of diffusion-based generative models. *Proc. NeurIPS*.

Khrulkov, V., Ryzhakov, G., Chertkov, A., & Oseledets, I. (2023). Understanding DDPM latent codes through optimal transport. *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=6PIrhAx1j4i

Lavenant, H., & Santambrogio, F. (2022). The flow map of the fokker–planck equation does not provide optimal transport. *Applied Mathematics Letters*, *133*, 108225. https://doi.org/https://doi.org/10.1016/j.aml.2022.108225

Lee, H., Lu, J., & Tan, Y. (2022). Convergence of score-based generative modeling for general data distributions. *NeurIPS 2022 Workshop on Score-Based Methods*.

Lee, H., Lu, J., & Tan, Y. (2024). Convergence for score-based generative modeling with polynomial complexity. *Proceedings of the 36th International Conference on Neural Information Processing Systems*.

Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., & Gool, L. V. (2022). Repaint: Inpainting using denoising diffusion probabilistic models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11451–11461. https://api.semanticscholar.org/CorpusID:246240274

Pardoux, E. (1986). Grossissement d'une filtration et retournement du temps d'une diffusion. In J. Azéma & M. Yor (Eds.), *Séminaire de probabilités xx 1984/85* (pp. 48–55). Springer Berlin Heidelberg.

Shah, K., Chen, S., & Klivans, A. (2023). Learning mixtures of gaussians using the DDPM objective. *Thirty-seventh Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=aig7sgdRfI

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021). Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*. https://openreview.net/forum?id=PxTIG12RRHS

Wenliang, L. K., & Moran, B. (2022). Score-based generative model learn manifold-like structures with constrained mixing. *NeurIPS 2022 Workshop on Score-Based Methods*. https://openreview.net/forum?id=eSZqaIrDLZR

Xiao, Z., Kreis, K., & Vahdat, A. (2022). Tackling the generative learning trilemma with denoising diffusion GANs. *International Conference on Learning Representations*.

Zach, M., Kobler, E., Chambolle, A., & Pock, T. (2024). Product of gaussian mixture diffusion models. *Journal of Mathematical Imaging and Vision*. https://doi.org/10.1007/s10851-024-01180-3

Zach, M., Pock, T., Kobler, E., & Chambolle, A. (2023). Explicit diffusion of gaussian mixture model based image priors. In L. Calatroni, M. Donatelli, S. Morigi, M. Prato, & M. Santacesaria (Eds.), *Scale space and variational methods in computer vision* (pp. 3–15). Springer International Publishing.