A Precise Examination of Diffusion Models via Their Application to Gaussian Distributions

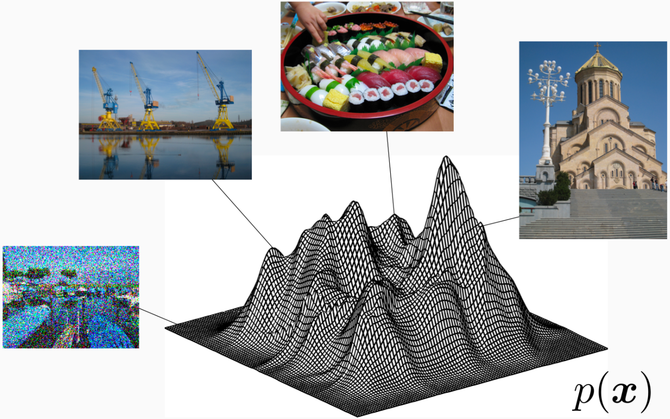
Émile Pierret[a], supervised by Bruno Galerne[a,b]
Séminaire Image Optimisations Probabilités de Bordeaux
June 26^{th} 2025

[a] Institut Denis Poisson – Université d'Orléans, Université de Tours, CNRS
[b] Institut universitaire de France (IUF)

Introduction

What is a generative model ?

Goal: Sample from a data distribution of images.



CelebA dataset

Dataset samples

50K samples

Dataset samples

50K samples

Generated (Fake) samples

Style GAN, (Karras et al., 2018) (NVIDIA)

Challenge: Given a model $G(\cdot;\Theta)$, find $\Theta^\star$ such that $G(\mathcal{N}(\mathbf{0}, \boldsymbol{I}_N), \Theta^\star) \approx p$

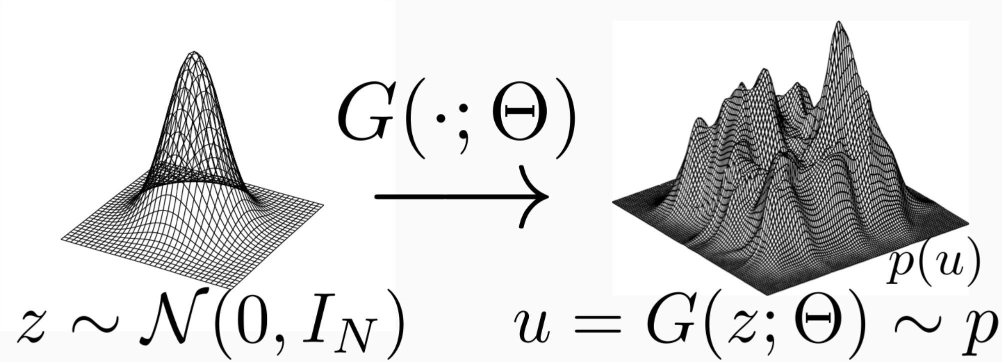


Image extracted from Bruno Galerne's slides

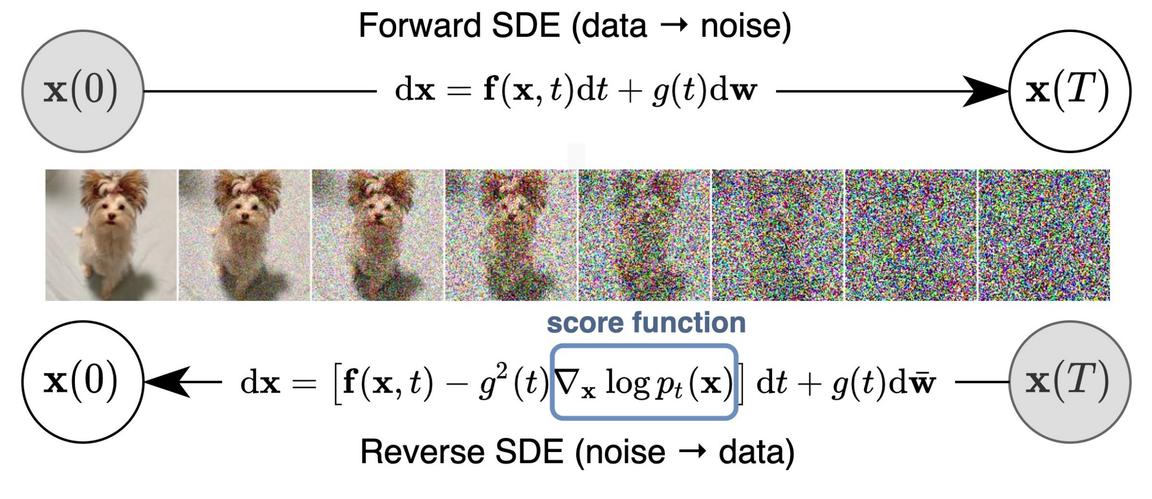
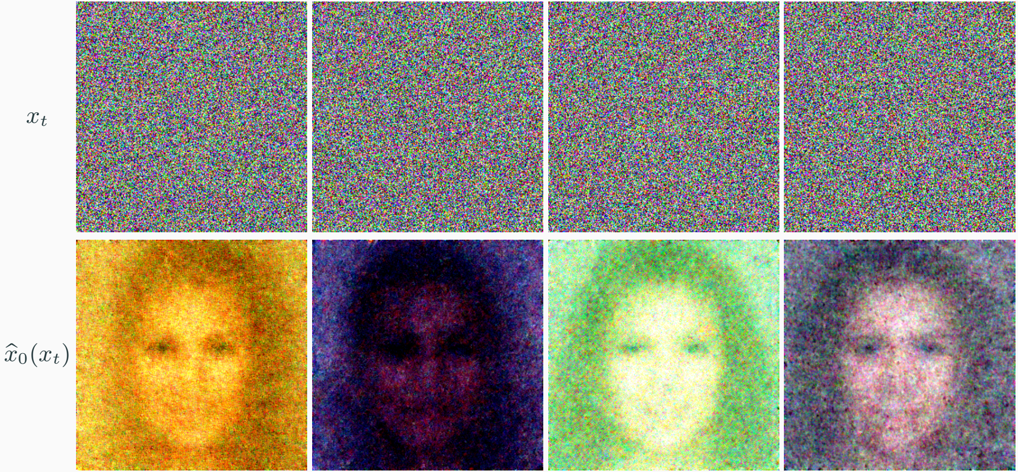


Image extracted from [Song et al. 2021]

Examples (generated with [Lugmayr et al. 2022][1])



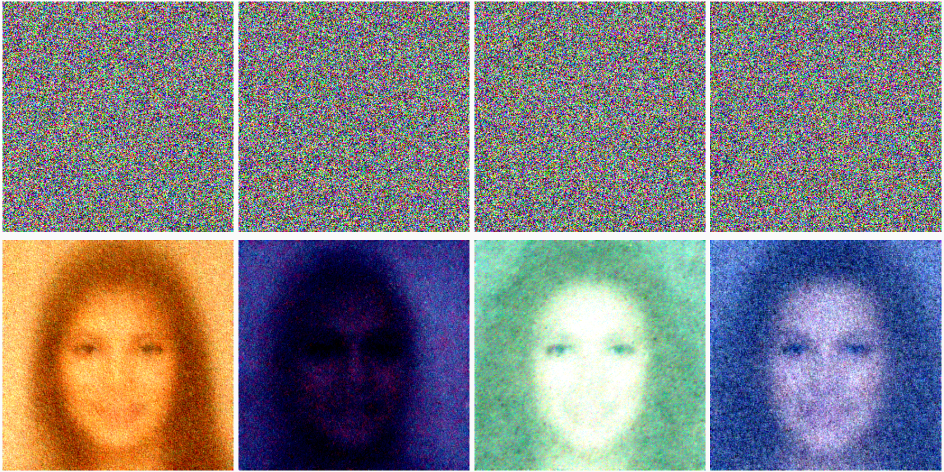
$t = 249$

[1] Andreas Lugmayr et al. (2022). "RePaint: Inpainting using Denoising Diffusion Probabilistic Models". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461. URL: https://api.semanticscholar.org/CorpusID:246240274

Examples (generated with [Lugmayr et al. 2022][1])

x_t

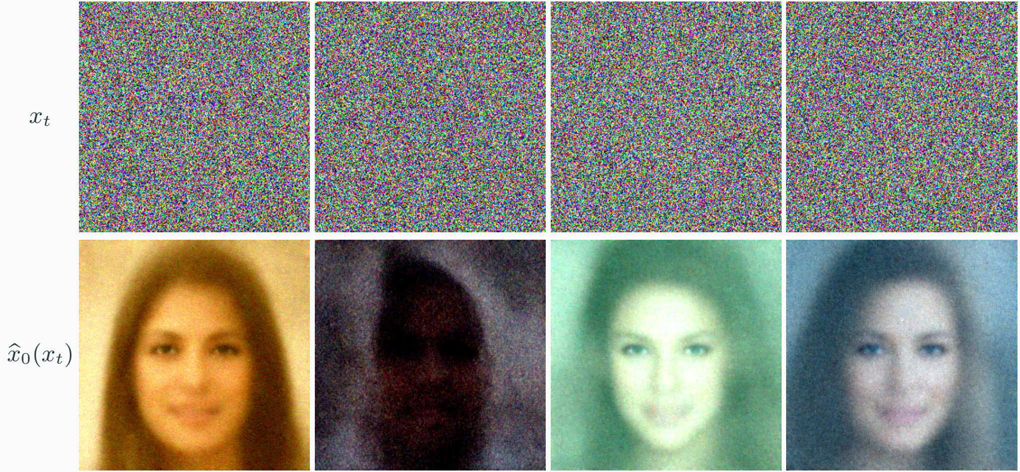
$\widehat{x}_0(x_t)$



t = 230

[1] Andreas Lugmayr et al. (2022). "RePaint: Inpainting using Denoising Diffusion Probabilistic Models". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461. URL: `https://api.semanticscholar.org/CorpusID:246240274`

Examples (generated with [Lugmayr et al. 2022][1])



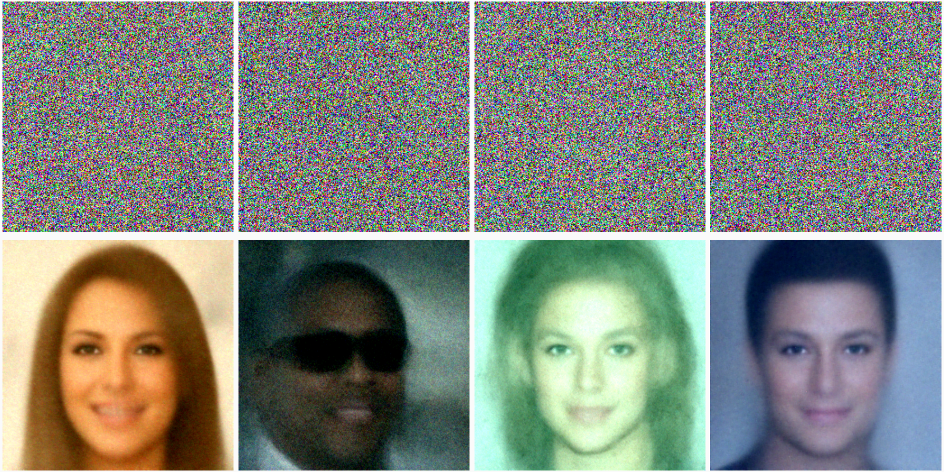
t = 210

[1] Andreas Lugmayr et al. (2022). "RePaint: Inpainting using Denoising Diffusion Probabilistic Models". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461. URL: `https://api.semanticscholar.org/CorpusID:246240274`

Examples (generated with [Lugmayr et al. 2022][1])

x_t

$\widehat{x}_0(x_t)$



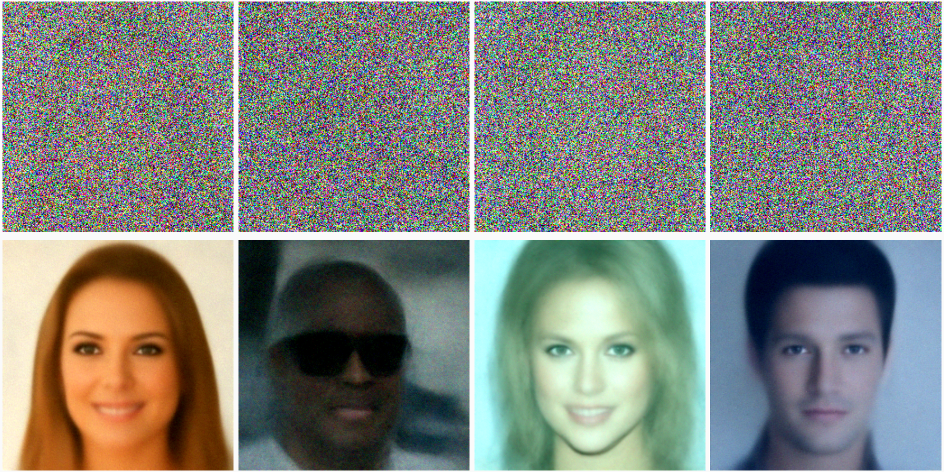
t = 190

[1] Andreas Lugmayr et al. (2022). “RePaint: Inpainting using Denoising Diffusion Probabilistic Models”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461. URL: https://api.semanticscholar.org/CorpusID:246240274

Examples (generated with [Lugmayr et al. 2022][1])

x_t

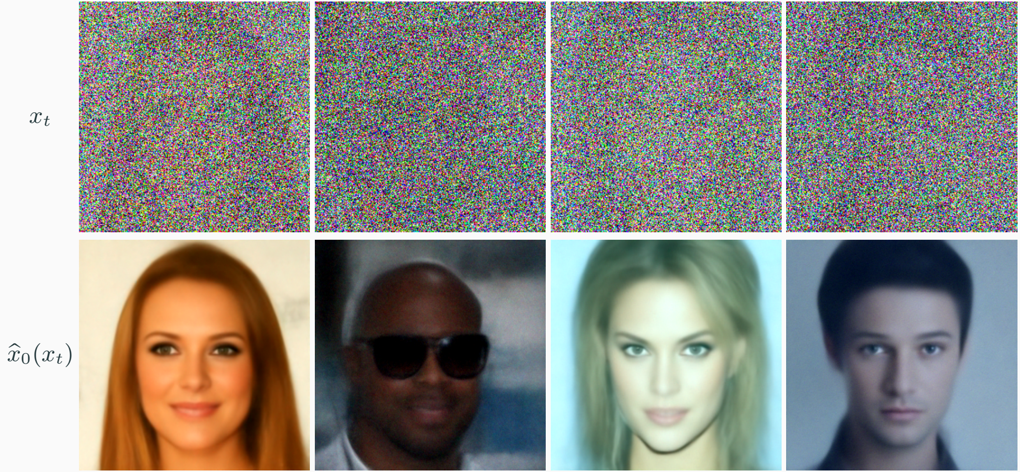
$\widehat{x}_0(x_t)$



t = 170

[1] Andreas Lugmayr et al. (2022). "RePaint: Inpainting using Denoising Diffusion Probabilistic Models". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461. URL: https://api.semanticscholar.org/CorpusID:246240274

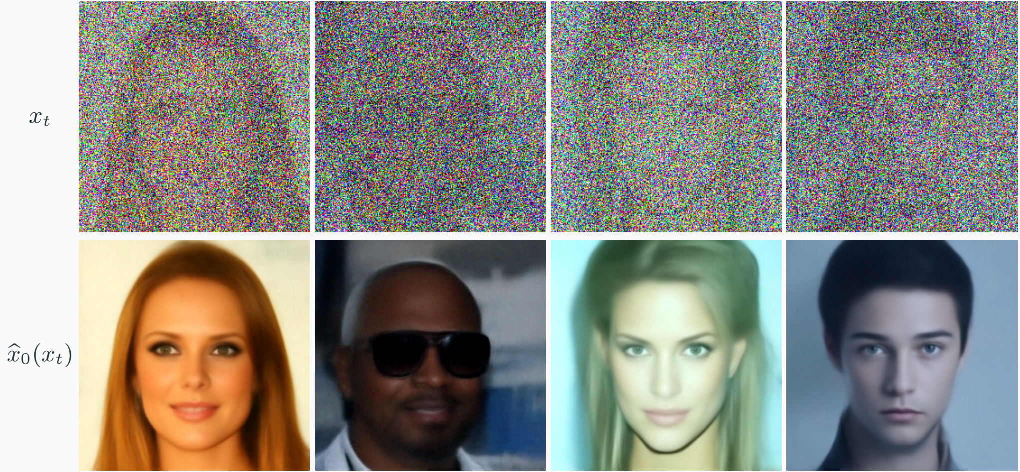
Examples (generated with [Lugmayr et al. 2022][1])



t = 150

[1] Andreas Lugmayr et al. (2022). "RePaint: Inpainting using Denoising Diffusion Probabilistic Models". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461. URL: https://api.semanticscholar.org/CorpusID:246240274

Examples (generated with [Lugmayr et al. 2022][1])



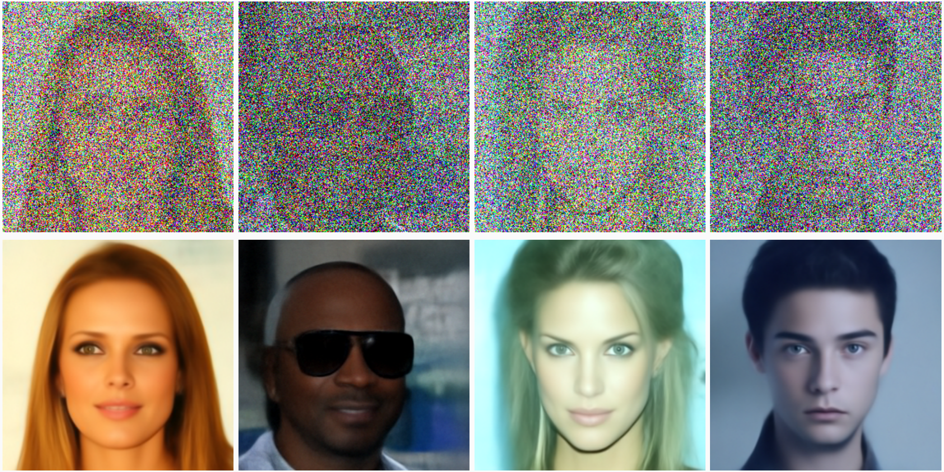
t = 130

[1] Andreas Lugmayr et al. (2022). "RePaint: Inpainting using Denoising Diffusion Probabilistic Models". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461. URL: https://api.semanticscholar.org/CorpusID:246240274

Examples (generated with [Lugmayr et al. 2022][1])

x_t

$\widehat{x}_0(x_t)$



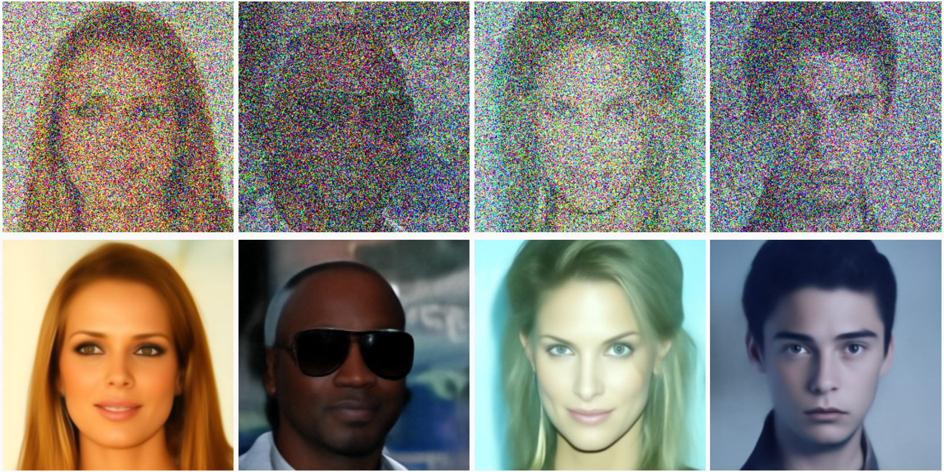
$\mathrm{t} = 110$

[1] Andreas Lugmayr et al. (2022). "RePaint: Inpainting using Denoising Diffusion Probabilistic Models". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461. URL: https://api.semanticscholar.org/CorpusID:246240274

Examples (generated with [Lugmayr et al. 2022][1])

x_t

$\widehat{x}_0(x_t)$



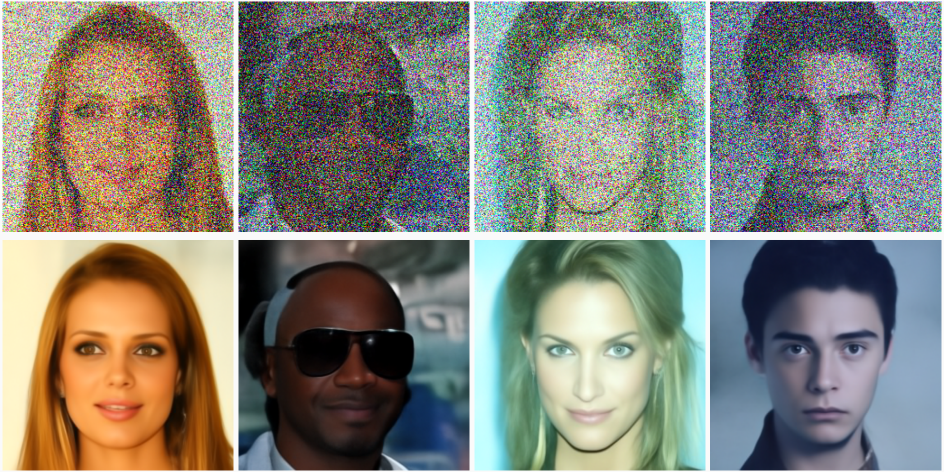
t = 90

[1] Andreas Lugmayr et al. (2022). "RePaint: Inpainting using Denoising Diffusion Probabilistic Models". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461. URL: `https://api.semanticscholar.org/CorpusID:246240274`

Examples (generated with [Lugmayr et al. 2022][1])

x_t

$\widehat{x}_0(x_t)$



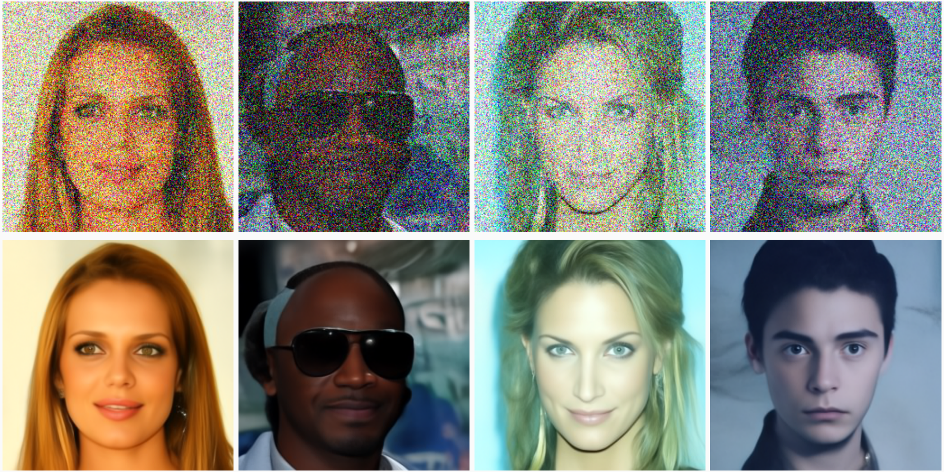
t = 70

[1] Andreas Lugmayr et al. (2022). "RePaint: Inpainting using Denoising Diffusion Probabilistic Models". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461. URL: https://api.semanticscholar.org/CorpusID:246240274

Examples (generated with [Lugmayr et al. 2022][1])

x_t

$\widehat{x}_0(x_t)$



$t = 50$

[1] Andreas Lugmayr et al. (2022). "RePaint: Inpainting using Denoising Diffusion Probabilistic Models". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461. URL: https://api.semanticscholar.org/CorpusID:246240274

Examples (generated with [Lugmayr et al. 2022][1])

x_t

$\widehat{x}_0(x_t)$

t = 30

[1] Andreas Lugmayr et al. (2022). "RePaint: Inpainting using Denoising Diffusion Probabilistic Models". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461. URL: https://api.semanticscholar.org/CorpusID:246240274

Examples (generated with [Lugmayr et al. 2022][1])

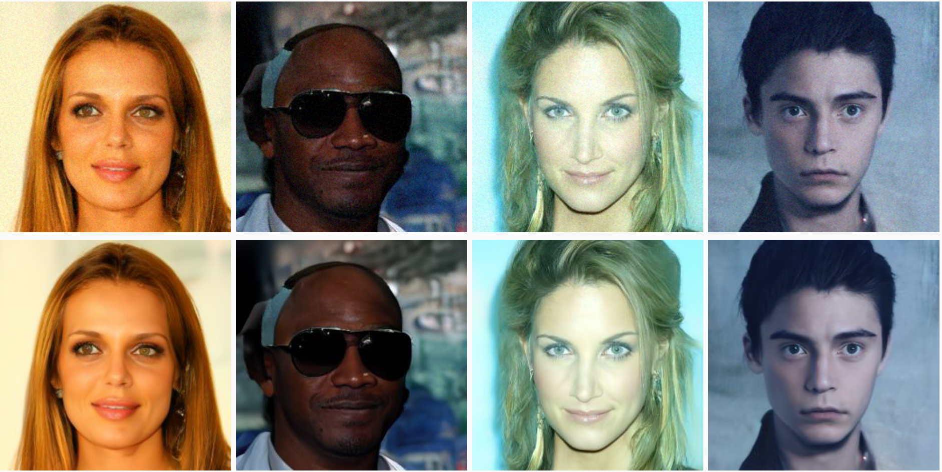
x_t

$\widehat{x}_0(x_t)$

t = 10

[1] Andreas Lugmayr et al. (2022). "RePaint: Inpainting using Denoising Diffusion Probabilistic Models". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461. URL: `https://api.semanticscholar.org/CorpusID:246240274`

Examples (generated with [Lugmayr et al. 2022][1])



x_t

$\widehat{x}_0(x_t)$

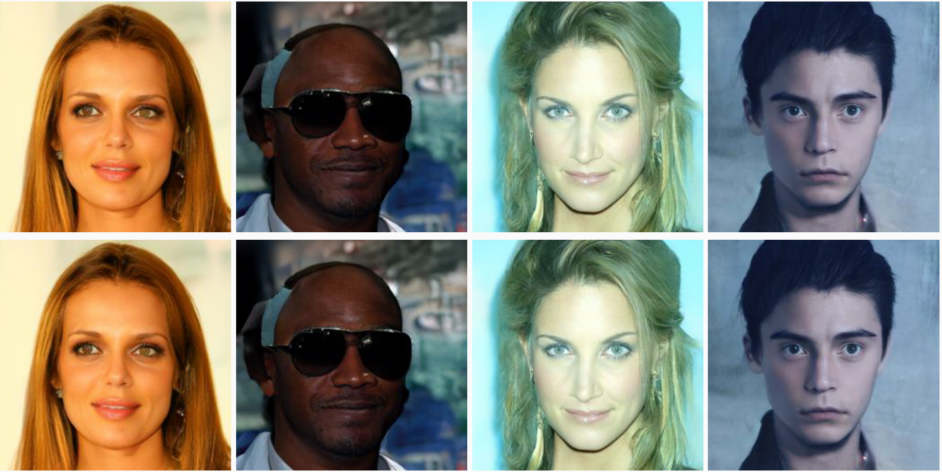
$t = 5$

[1] Andreas Lugmayr et al. (2022). "RePaint: Inpainting using Denoising Diffusion Probabilistic Models". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461. URL: `https://api.semanticscholar.org/CorpusID:246240274`

Examples (generated with [Lugmayr et al. 2022][1])

x_t

$\widehat{x}_0(x_t)$



$\mathrm{t} = 0$

[1] Andreas Lugmayr et al. (2022). "RePaint: Inpainting using Denoising Diffusion Probabilistic Models". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461. URL: https://api.semanticscholar.org/CorpusID:246240274

Introduction to diffusion models through SDEs

Focus on the VP-SDE: the forward process

$$dx_t = -\beta_t x_t dt + \sqrt{2\beta_t} dw_t, \quad 0 \leqslant t \leqslant T, \quad x_0 \sim p_{\mathrm{data}} \tag{1}$$

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

Focus on the VP-SDE: the forward process

$$dx_t = -\beta_t x_t dt + \sqrt{2\beta_t} dw_t, \quad 0 \leqslant t \leqslant T, \quad x_0 \sim p_{\text{data}} \tag{1}$$

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

The strong solution of Equation (1) is:

$$x_t = e^{-B_t} x_0 + \boldsymbol{\eta}_t, \quad 0 \leqslant t \leqslant T. \tag{2}$$

with $\eta_t \sim \mathcal{N}(\mathbf{0}, \left(1 - e^{-2B_t}\right) \boldsymbol{I}), B_t = \int_0^t \beta_u du.$

Focus on the VP-SDE: the forward process

$$dx_t = -\beta_t x_t dt + \sqrt{2\beta_t} dw_t, \quad 0 \leqslant t \leqslant T, \quad x_0 \sim p_{\text{data}} \tag{1}$$

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

The strong solution of Equation (1) is:

$$x_t = e^{-B_t} x_0 + \boldsymbol{\eta}_t, \quad 0 \leqslant t \leqslant T. \tag{2}$$

with $\eta_t \sim \mathcal{N}(\mathbf{0}, \left(1 - e^{-2B_t}\right) \boldsymbol{I}), B_t = \int_0^t \beta_u du$.

Consequently, if $t \to +\infty$, $x_\infty \sim \mathcal{N}_0$.

Focus on the VP-SDE: the forward process

$$dx_t = -\beta_t x_t dt + \sqrt{2\beta_t} dw_t, \quad 0 \leqslant t \leqslant T, \quad x_0 \sim p_{\text{data}} \tag{1}$$

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

The strong solution of Equation (1) is:

$$x_t = e^{-B_t} x_0 + \boldsymbol{\eta}_t, \quad 0 \leqslant t \leqslant T. \tag{2}$$

with $\eta_t \sim \mathcal{N}(\mathbf{0}, \left(1 - e^{-2B_t}\right) \boldsymbol{I}), B_t = \int_0^t \beta_u du$.

Consequently, if $t \to +\infty$, $x_\infty \sim \mathcal{N}_0$.

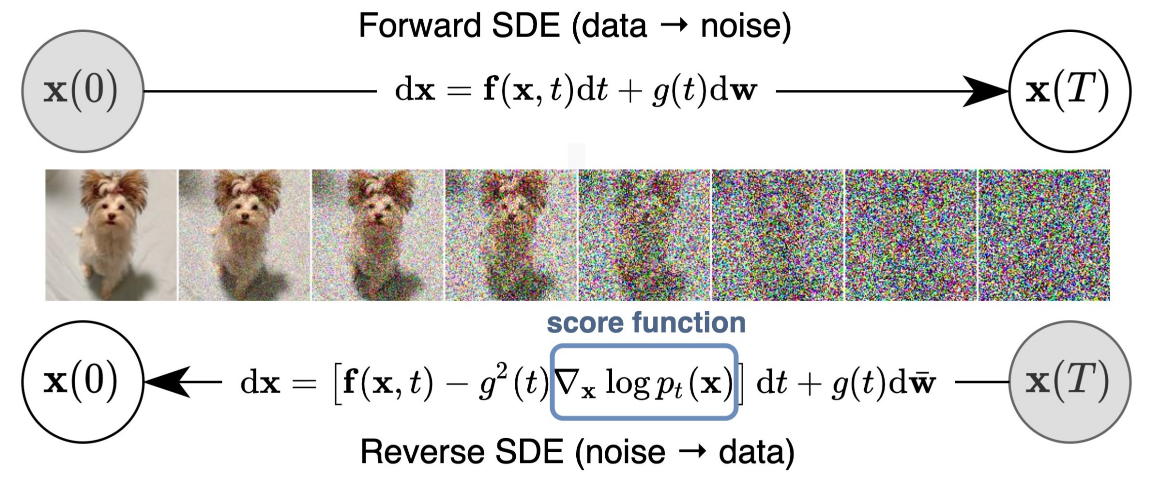


Image extracted from [Song et al. 2021]

If X_t solution of

$$d\boldsymbol{X}_t = b(t, \boldsymbol{X}_t)dt + \sigma(t, \boldsymbol{X}_t)d\boldsymbol{W}_t \tag{3}$$

Under assumptions

1. **(H1)** $\exists K > 0$ s.t. $\forall (t, x, y) \in [0, 1] \times \mathbb{R}^d \times \mathbb{R}^d$,

$$|b(t,x) - b(t,y)| + |\sigma(t,x) - \sigma(t,y)| \leqslant K|x-y|$$
$$|b(t,x)| + |\sigma(t,x)| \leqslant K(1 + |x|).$$

2. **(H2)** p_{data} has a density distribution in $L^2\left(\mathbb{R}^d, \frac{dx}{1+|x|^k}\right)$ for a certain $k \in \mathbb{N}$.
3. **(H3)** $\frac{\partial^2 \sigma^2}{\partial x_i \partial x_j}(t,x) \in L^\infty(]0,1[\times\mathbb{R}^d)$ for $1 \leqslant i, j \leqslant d$.

then $\overline{\boldsymbol{X}}_t = \boldsymbol{X}_{1-t}$ is solution of

$$d\overline{\boldsymbol{X}}_t = \bar{b}(t, \overline{\boldsymbol{X}}_t)dt + \overline{\sigma}(t, \overline{\boldsymbol{X}}_t)d\overline{\boldsymbol{W}}_t \tag{4}$$

In our case, $b(t,x) = -\beta_t x + \beta_t \nabla \log p_t(x), \sigma(t,x) = \sqrt{2\beta_t}$

[2] E. Pardoux (1986). "Grossissement d'une filtration et retournement du temps d'une diffusion". In: *Séminaire de Probabilités XX 1984/85*. Ed. by Jacques Azéma and Marc Yor. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 48–55. ISBN: 978-3-540-39860-8

Study of the backward process [Pardoux 1986][3]

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

$$dy_t = \beta_{T-t}(y_t + 2\nabla \log p_{T-t}(y_t))dt + \sqrt{2\beta_{T-t}}d\overline{w}_t, \quad 0 \leqslant t < T, \quad y_0 \sim p_T. \tag{5}$$

[3] E. Pardoux (1986). "Grossissement d'une filtration et retournement du temps d'une diffusion". In: *Séminaire de Probabilités XX 1984/85*. Ed. by Jacques Azéma and Marc Yor. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 48–55. ISBN: 978-3-540-39860-8

Study of the backward process [Pardoux 1986][3]

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

$$dy_t = \beta_{T-t}(y_t + 2\nabla \log p_{T-t}(y_t))dt + \sqrt{2\beta_{T-t}}d\overline{w}_t, \quad 0 \leqslant t < T, \quad y_0 \sim p_T. \tag{5}$$

- The backward Brownian motion $\overline{w}$ is not defined on the same filtration than the forward w:

$$\overline{w}_t = w_t - w_T + \int_t^T \frac{1}{p(s, x_s)} \operatorname{div}(\sigma p)(s, x_s) ds. \tag{6}$$

[3] E. Pardoux (1986). "Grossissement d'une filtration et retournement du temps d'une diffusion". In: *Séminaire de Probabilités XX 1984/85*. Ed. by Jacques Azéma and Marc Yor. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 48–55. ISBN: 978-3-540-39860-8

Study of the backward process [Pardoux 1986][3]

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

$$dy_t = \beta_{T-t}(y_t + 2\nabla \log p_{T-t}(y_t))dt + \sqrt{2\beta_{T-t}}d\overline{w}_t, \quad 0 \leqslant t < T, \quad y_0 \sim p_T. \tag{5}$$

- The backward Brownian motion $\overline{w}$ is not defined on the same filtration than the forward w:

$$\overline{w}_t = w_t - w_T + \int_t^T \frac{1}{p(s, x_s)} \operatorname{div}(\sigma p)(s, x_s)ds. \tag{6}$$

- $\nabla \log p_{T-t}$ is called the score function.

[3] E. Pardoux (1986). "Grossissement d'une filtration et retournement du temps d'une diffusion". In: *Séminaire de Probabilités XX 1984/85*. Ed. by Jacques Azéma and Marc Yor. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 48–55. ISBN: 978-3-540-39860-8

Study of the backward process [Pardoux 1986][3]

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

$$dy_t = \beta_{T-t}(y_t + 2\nabla \log p_{T-t}(y_t))dt + \sqrt{2\beta_{T-t}}d\overline{w}_t, \quad 0 \leqslant t < T, \quad y_0 \sim p_T. \tag{5}$$

- The backward Brownian motion $\overline{w}$ is not defined on the same filtration than the forward w:

$$\overline{w}_t = w_t - w_T + \int_t^T \frac{1}{p(s, x_s)} \operatorname{div}(\sigma p)(s, x_s) ds. \tag{6}$$

- $\nabla \log p_{T-t}$ is called the score function.
- We are unable to derive the score function: **No closed-form solution !**

[3] E. Pardoux (1986). "Grossissement d'une filtration et retournement du temps d'une diffusion". In: *Séminaire de Probabilités XX 1984/85*. Ed. by Jacques Azéma and Marc Yor. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 48–55. ISBN: 978-3-540-39860-8

Study of the backward process [Pardoux 1986][3]

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

$$dy_t = \beta_{T-t}(y_t + 2\nabla \log p_{T-t}(y_t))dt + \sqrt{2\beta_{T-t}}d\overline{w}_t, \quad 0 \leqslant t < T, \quad y_0 \sim p_T. \tag{5}$$

- The backward Brownian motion $\overline{w}$ is not defined on the same filtration than the forward w:

$$\overline{w}_t = w_t - w_T + \int_t^T \frac{1}{p(s, x_s)} \operatorname{div}(\sigma p)(s, x_s)ds. \tag{6}$$

- $\nabla \log p_{T-t}$ is called the score function.
- We are unable to derive the score function: **No closed-form solution !**
- Learn $s_\theta(x,t) \approx \nabla \log p_t(x)$

[3] E. Pardoux (1986). "Grossissement d'une filtration et retournement du temps d'une diffusion". In: *Séminaire de Probabilités XX 1984/85*. Ed. by Jacques Azéma and Marc Yor. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 48–55. ISBN: 978-3-540-39860-8

Study of the backward process [Pardoux 1986][3]

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

$$dy_t = \beta_{T-t}(y_t + 2\nabla \log p_{T-t}(y_t))dt + \sqrt{2\beta_{T-t}}d\overline{w}_t, \quad 0 \leqslant t < T, \quad y_0 \sim p_T. \tag{5}$$

- The backward Brownian motion $\overline{w}$ is not defined on the same filtration than the forward w:

$$\overline{w}_t = w_t - w_T + \int_t^T \frac{1}{p(s,x_s)} \operatorname{div}(\sigma p)(s,x_s)ds. \tag{6}$$

- $\nabla \log p_{T-t}$ is called the score function.
- We are unable to derive the score function: **No closed-form solution !**
- Learn $s_\theta(x,t) \approx \nabla \log p_t(x)$

$$dy_t = \beta_{T-t}(y_t + 2s_\theta(y_t, T-t))dt + \sqrt{2\beta_{T-t}}d\overline{w}_t, \quad 0 \leqslant t < T, \quad y_0 \sim p_T. \tag{7}$$

[3] E. Pardoux (1986). "Grossissement d'une filtration et retournement du temps d'une diffusion". In: *Séminaire de Probabilités XX 1984/85.* Ed. by Jacques Azéma and Marc Yor. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 48–55. ISBN: 978-3-540-39860-8

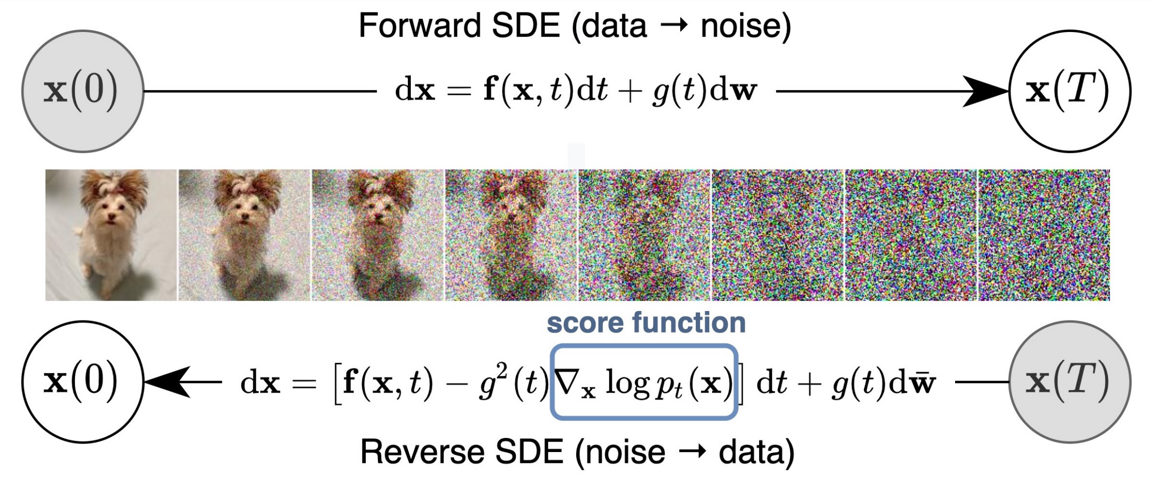


Image extracted from [Song et al. 2021]

To the probability flow-ODE: the Fokker-Planck equation

$$dx_t = f(x_t, t)dt + g(t)dw_t \tag{8}$$

The marginals $(p_t)_{0\leqslant t\leqslant T}$ follow the Fokker-Planck Equation:

$$\partial_t p_t(x) = -\operatorname{div}_x \left[f(x,t)p_t(x)\right] + \frac{1}{2}g(t)^2\Delta_x p_t(x). \tag{9}$$

To the probability flow-ODE: the Fokker-Planck equation

$$dx_t = f(x_t, t)dt + g(t)dw_t \tag{8}$$

The marginals $(p_t)_{0 \leqslant t \leqslant T}$ follow the Fokker-Planck Equation:

$$\partial_t p_t(x) = -\operatorname{div}_x \left[f(x,t) p_t(x) \right] + \frac{1}{2} g(t)^2 \Delta_x p_t(x). \tag{9}$$

By denoting $q_t = p_{T-t}$, we search for a Fokker-Planck equation associated with q_t.

To the probability flow-ODE: the Fokker-Planck equation

$$dx_t = f(x_t, t)dt + g(t)dw_t \tag{8}$$

The marginals $(p_t)_{0 \leqslant t \leqslant T}$ follow the Fokker-Planck Equation:

$$\partial_t p_t(x) = -\operatorname{div}_x \left[f(x,t)p_t(x)\right] + \frac{1}{2}g(t)^2 \Delta_x p_t(x). \tag{9}$$

By denoting $q_t = p_{T-t}$, we search for a Fokker-Planck equation associated with q_t.

$$\begin{aligned}
\partial_t q_t(x) &= -\partial_t p_{T-t}(x) \\
&= \operatorname{div}_x \left[f(x, T-t)q_t(x)\right] + \left(-1 + \frac{1}{2}\right) g(T-t)^2 \Delta_x q_t(x) \\
&= -\operatorname{div}_x \left[\left(-f(x, T-t) + g(T-t)^2 \nabla_x \log q_t(x)\right) q_t(x)\right] + \frac{1}{2}g(T-t)^2 \Delta_x q_t(x).
\end{aligned}$$

To the probability flow-ODE: the Fokker-Planck equation

$$dx_t = f(x_t, t)dt + g(t)dw_t \tag{8}$$

The marginals $(p_t)_{0 \leqslant t \leqslant T}$ follow the Fokker-Planck Equation:

$$\partial_t p_t(x) = -\operatorname{div}_x \left[f(x,t) p_t(x) \right] + \frac{1}{2} g(t)^2 \Delta_x p_t(x). \tag{9}$$

By denoting $q_t = p_{T-t}$, we search for a Fokker-Planck equation associated with q_t.

$$\begin{aligned}
\partial_t q_t(x) &= -\partial_t p_{T-t}(x) \\
&= \operatorname{div}_x \left[f(x, T-t) q_t(x) \right] + \left(-1 + \frac{1}{2} \right) g(T-t)^2 \Delta_x q_t(x) \\
&= -\operatorname{div}_x \left[\left(-f(x, T-t) + g(T-t)^2 \nabla_x \log q_t(x) \right) q_t(x) \right] + \frac{1}{2} g(T-t)^2 \Delta_x q_t(x).
\end{aligned}$$

$$\begin{aligned}
\partial_t q_t(x) &= -\partial_t p_{T-t}(x) \\
&= \operatorname{div}_x \left[f(x, T-t) q_t(x) \right] + \left(0 - \frac{1}{2} \right) g(T-t)^2 \Delta_x q_t(x) \\
&= -\operatorname{div}_x \left[\left(-f(x, T-t) + \frac{1}{2} g(T-t)^2 \nabla_x \log q_t(x) \right) q_t(x) \right].
\end{aligned}$$

Probability-flow ODE

The marginals $(p_t)_{0 \leqslant t \leqslant T}$ associated with the backward SDE

$$dy_t = -\beta_t \left[y_t + 2\nabla_y \log p_t(y_t) \right] dt + \sqrt{2\beta_t} d\overline{w}_t, \quad 0 \leqslant t \leqslant T, \quad y_T \sim p_T \tag{10}$$

Probability-flow ODE

The marginals $(p_t)_{0 \leqslant t \leqslant T}$ associated with the backward SDE

$$dy_t = -\beta_t \left[y_t + 2\nabla_y \log p_t(y_t) \right] dt + \sqrt{2\beta_t} d\overline{w}_t, \quad 0 \leqslant t \leqslant T, \quad y_T \sim p_T \tag{10}$$

are the same as those of this ODE

$$dy_t = -\beta_t \left[y_t + \nabla_y \log p_t(y_t) \right] dt, \qquad 0 \leqslant t \leqslant T, \quad y_T \sim p_T. \tag{11}$$

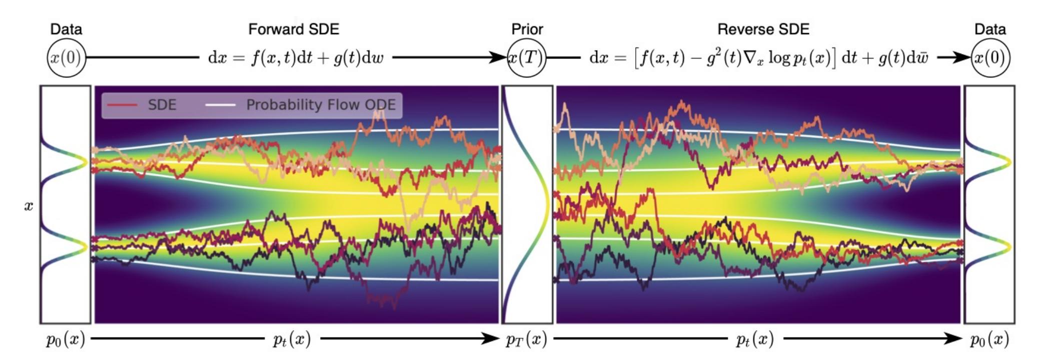
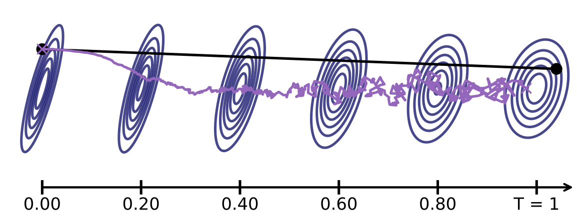


Image extracted from [Song et al. 2021]

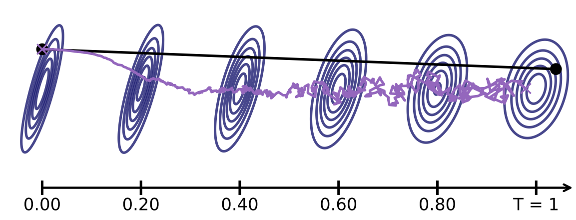
Study of the convergence of diffusion models

Illustration of the different error types

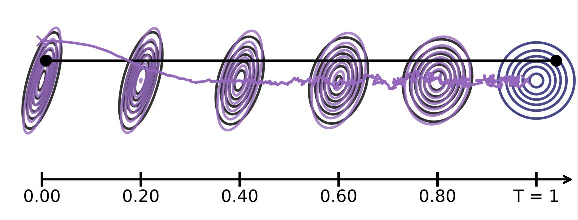
Theoretical setting



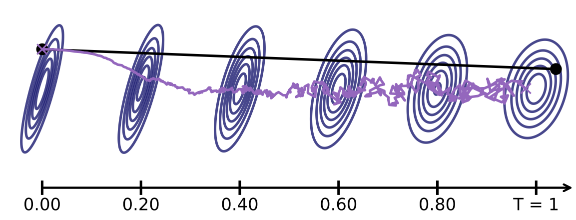
Theoretical setting



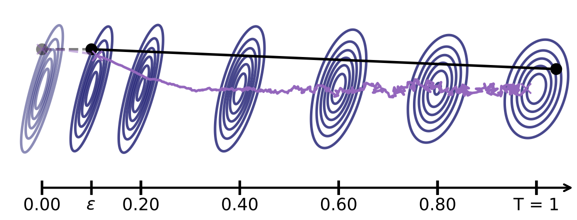
Initialization error



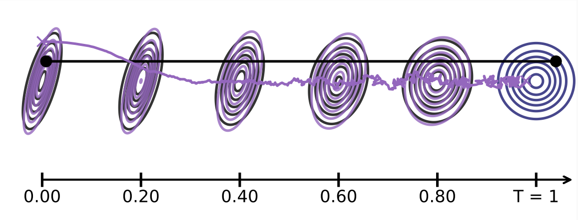
Theoretical setting

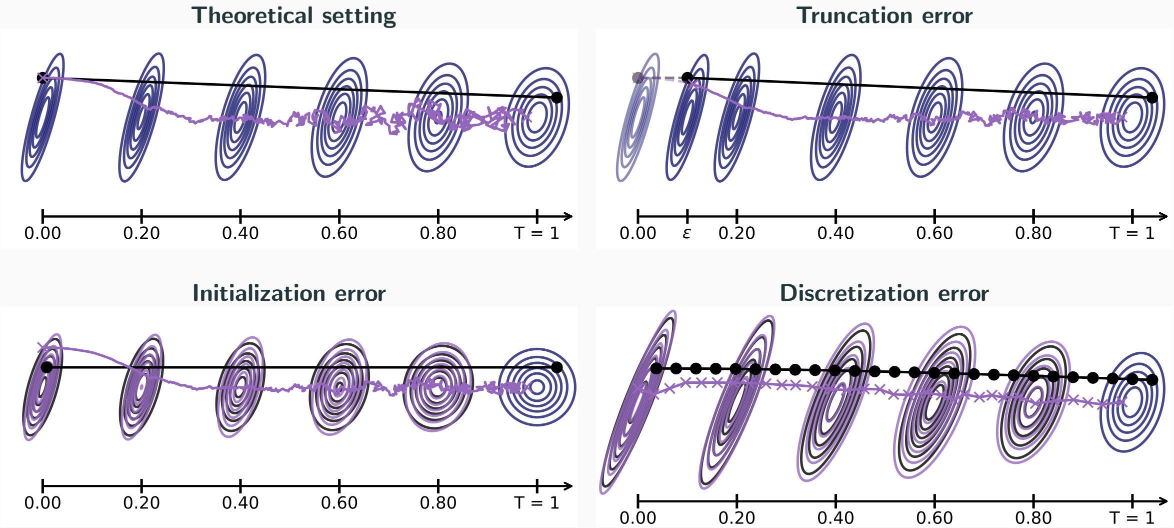


Truncation error



Initialization error




Theoretical setting
0.00
0.20
0.40
0.60
0.80
T = 1
Truncation error
0.00
ε
0.20
0.40
0.60
0.80
T = 1
Initialization error
0.00
0.20
0.40
0.60
0.80
T = 1
Discretization error
0.00
0.20
0.40
0.60
0.80
T = 1

$$
\begin{aligned}
&dy_t = -\beta_t \left[y_t + 2\nabla_y \log p_t(y_t)\right] dt + \sqrt{2\beta_t} d\overline{w}_t, \\
&\text{or} \qquad\qquad\qquad\qquad\qquad\qquad \text{where} \quad 0 \leqslant t \leqslant T, \quad y_T \sim p_T. \qquad (12)\\
&dy_t = -\beta_t \left[y_t + \nabla_y \log p_t(y_t)\right] dt,
\end{aligned}
$$

Sampling a distribution using diffusion models implies different choices and error types:

$$dy_t = -\beta_t \left[y_t + 2\nabla_y \log p_t(y_t) \right] dt + \sqrt{2\beta_t} d\overline{w}_t,$$

or

$$dy_t = -\beta_t \left[y_t + \nabla_y \log p_t(y_t) \right] dt,$$

where $0 \leqslant t \leqslant T$, $\cancel{y_T \sim p_T}$ $y_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. (12)

Sampling a distribution using diffusion models implies different choices and error types:

- p_T, which is unknown, is replaced by $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ → **initialization error**

$$dy_t = -\beta_t \left[y_t + 2\nabla_y \log p_t(y_t) \right] dt + \sqrt{2\beta_t} d\overline{w}_t,$$
or
$$dy_t = -\beta_t \left[y_t + \nabla_y \log p_t(y_t) \right] dt,$$
where $\cancel{0}_{\varepsilon} \leqslant t \leqslant T, \quad y_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}).$ (12)

Sampling a distribution using diffusion models implies different choices and error types:

- p_T, which is unknown, is replaced by $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ → **initialization error**
- In fact, another time ε to consider them on $[\varepsilon, T]$ → **truncation error**

$$dy_t = -\beta_t \left[y_t + 2\nabla_y \log p_t(y_t)\right] dt + \sqrt{2\beta_t} d\overline{w}_t,$$

or   where $\varepsilon \leqslant t \leqslant T, \quad y_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}).$ (12)

$$dy_t = -\beta_t \left[y_t + \nabla_y \log p_t(y_t)\right] dt,$$

Sampling a distribution using diffusion models implies different choices and error types:

- p_T, which is unknown, is replaced by $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ → **initialization error**
- In fact, another time ε to consider them on $[\varepsilon, T]$ → **truncation error**
- A scheme to discretize the equations → **discretization error**

$$dy_t = -\beta_t[y_t + 2\underbrace{\cancel{\nabla_y \log p_t(y_t)}}_{s_\theta(t, y_t)}]dt + \sqrt{2\beta_t}d\overline{w}_t,$$

or where $\varepsilon \leqslant t \leqslant T, \quad y_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}).$ (12)

$$dy_t = -\beta_t[y_t + \underbrace{\cancel{\nabla_y \log p_t(y_t)}}_{s_\theta(t, y_t)}]dt,$$

Sampling a distribution using diffusion models implies different choices and error types:

- p_T, which is unknown, is replaced by $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ → **initialization error**
- In fact, another time ε to consider them on $[\varepsilon, T]$ → **truncation error**
- A scheme to discretize the equations → **discretization error**
- A model/neural network s_θ to learn the score → **score approximation error**

Sampling through diffusion models

$$
\begin{aligned}
&dy_t = -\beta_t[y_t + 2s_\theta(t, y_t)]dt + \sqrt{2\beta_t}d\overline{w}_t, \\
&\quad \text{or} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{where} \quad \varepsilon \leqslant t \leqslant T, \quad y_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}). \\
&dy_t = -\beta_t[y_t + s_\theta(t, y_t)]dt,
\end{aligned}
\tag{12}
$$

Sampling a distribution using diffusion models implies different choices and error types:

- p_T, which is unknown, is replaced by $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ → **initialization error**
- In fact, another time ε to consider them on $[\varepsilon, T]$ → **truncation error**
- A scheme to discretize the equations → **discretization error**
- A model/neural network s_θ to learn the score → **score approximation error**

Theorem 1. *Assume* **A**1, **A**2, **A**3, **A**4 *that* $T \geq 2\bar{\beta}(1 + \log(1 + \mathrm{diam}(\mathcal{M})))$, $\gamma_K = \varepsilon$ *and* $\varepsilon, \mathtt{M}, \delta \leq 1/32$. *Then, there exists* $\mathtt{D}_0 \geq 0$ *such that*

$$\mathbf{W}_1(\mathcal{L}(Y_K), \pi) \leq \mathtt{D}_0(\exp[\kappa/\varepsilon](\mathtt{M} + \delta^{1/2})/\varepsilon^2 + \exp[\kappa/\varepsilon]\exp[-T/\bar{\beta}] + \varepsilon^{1/2}),$$

with $\kappa = \mathrm{diam}(\mathcal{M})^2(1 + \bar{\beta})/2$ *and*

$$\mathtt{D}_0 = D(1 + \bar{\beta})^7(1 + d + \mathrm{diam}(\mathcal{M})^4)(1 + \log(1 + \mathrm{diam}(\mathcal{M}))), \tag{7}$$

and D *is a numerical constant.*

Theorem 3. *Assume* **A**1, **A**2, **A**3, **A**4 *that* $T \geq 2\beta(1 + \log(1 + \mathrm{diam}(\mathcal{M})))$, $\gamma_K = \varepsilon$ *and* $\varepsilon, \mathtt{M}, \delta \leq 1/32$. *In addition, assume that there exists* $\Gamma \geq 0$ *such that for any* $t \in (0, T]$ *and* $x_t \in \mathbb{R}^d$

$$\|\nabla^2 \log p_t(x_t)\| \leq \Gamma/\sigma_t^2. \tag{9}$$

Then, there exists $\mathtt{D}_0 \geq 0$ *such that*

$$\mathbf{W}_1(\mathcal{L}(Y_K), \pi) \leq \mathtt{D}_0((\mathtt{M} + \delta^{1/2})/\varepsilon^{\Gamma+2} + \exp[-T/\bar{\beta}]/\varepsilon^{\Gamma} + \varepsilon^{1/2}),$$

From Valentin De Bortoli (2022). “Convergence of denoising diffusion models under the manifold hypothesis”. In: *Transactions on Machine Learning Research*. ISSN: 2835-8856. URL: https://openreview.net/forum?id=MhK5aXo3gB

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)

Gaussian assumption

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

$$\nabla \log p_t(x) = -\mathbf{\Sigma}_t^{-1} x, \quad 0 < t \leqslant T \tag{13}$$

with $\mathbf{\Sigma}_t = e^{-2B_t}\mathbf{\Sigma} + (1 - e^{-2B_t})\boldsymbol{I}$.

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

$$\nabla \log p_t(x) = -\mathbf{\Sigma}_t^{-1} x, \quad 0 < t \leqslant T \tag{13}$$

with $\mathbf{\Sigma}_t = e^{-2B_t}\mathbf{\Sigma} + (1 - e^{-2B_t})\boldsymbol{I}$.

Proposition 3: Lineartiy of the score

he three following propositions are equivalent:

(i) $x_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ for some covariance $\mathbf{\Sigma}$.

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

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

Initialization error

Explicit solution of the backward equations

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

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

Proposition 4: Solution to the equations under Gaussian assumption

Under Gaussian assumption, the strong solution to SDE (5) can be written as:

$$y_t^{\mathsf{SDE}} = e^{-(B_T - B_t)} \mathbf{\Sigma}_t \mathbf{\Sigma}_T^{-1} y_T + \boldsymbol{\xi}_t, \quad 0 \leqslant t \leqslant T \tag{14}$$

Under Gaussian assumption, the solution to ODE (11) can be written as:

$$y_t^{\mathsf{ODE}} = \mathbf{\Sigma}_T^{-1/2} \mathbf{\Sigma}_t^{1/2} y_T, \quad 0 \leqslant t \leqslant T, \tag{15}$$

with $\mathbf{\Sigma}_t = e^{-2B_t}\mathbf{\Sigma} + (1 - e^{-2B_t})\boldsymbol{I}$.

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

Proposition 5: Solution to the equations under Gaussian assumption

If $\mathrm{Cov}(y_T)\mathbf{\Sigma} = \mathbf{\Sigma}\,\mathrm{Cov}(y_T)$, for $0 \leqslant t \leqslant T$,

$$\mathbf{\Sigma}_t^{\mathsf{SDE}} = \mathrm{Cov}(y_t^{\mathsf{SDE}}) = \mathbf{\Sigma}_t + e^{-2(B_T - B_t)}\mathbf{\Sigma}_t^2\mathbf{\Sigma}_T^{-2}\left[\mathrm{Cov}(y_T) - \mathbf{\Sigma}_T\right], \tag{14}$$

$$\mathbf{\Sigma}_t^{\mathsf{ODE}} = \mathrm{Cov}(y_t^{\mathsf{ODE}}) = \mathbf{\Sigma}_t\mathbf{\Sigma}_T^{-1}\,\mathrm{Cov}(y_T), \tag{15}$$

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

Proposition 6: Solution to the equations under Gaussian assumption

If $\mathrm{Cov}(y_T)\mathbf{\Sigma} = \mathbf{\Sigma}\,\mathrm{Cov}(y_T)$, for $0 \leqslant t \leqslant T$,

$$\mathbf{\Sigma}_t^{\mathsf{SDE}} = \mathrm{Cov}(y_t^{\mathsf{SDE}}) = \mathbf{\Sigma}_t + e^{-2(B_T - B_t)}\mathbf{\Sigma}_t^2\mathbf{\Sigma}_T^{-2}\left[\mathrm{Cov}(y_T) - \mathbf{\Sigma}_T\right], \tag{14}$$

$$\mathbf{\Sigma}_t^{\mathsf{ODE}} = \mathrm{Cov}(y_t^{\mathsf{ODE}}) = \mathbf{\Sigma}_t\mathbf{\Sigma}_T^{-1}\,\mathrm{Cov}(y_T), \tag{15}$$

If $y_T \sim p_T = \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_T)$,

$$\mathbf{\Sigma}_t^{\mathsf{SDE}} = \mathbf{\Sigma}_t^{\mathsf{ODE}} = \mathbf{\Sigma}_t, \quad 0 \leqslant t \leqslant T.$$

Explicit solution of the backward equations

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

Proposition 7: Solution to the equations under Gaussian assumption

If $\operatorname{Cov}(y_T)\mathbf{\Sigma} = \mathbf{\Sigma}\operatorname{Cov}(y_T)$, for $0 \leqslant t \leqslant T$,

$$\mathbf{\Sigma}_t^{\mathsf{SDE}} = \operatorname{Cov}(y_t^{\mathsf{SDE}}) = \mathbf{\Sigma}_t + e^{-2(B_T - B_t)}\mathbf{\Sigma}_t^2\mathbf{\Sigma}_T^{-2}\left[\operatorname{Cov}(y_T) - \mathbf{\Sigma}_T\right], \tag{14}$$

$$\mathbf{\Sigma}_t^{\mathsf{ODE}} = \operatorname{Cov}(y_t^{\mathsf{ODE}}) = \mathbf{\Sigma}_t\mathbf{\Sigma}_T^{-1}\operatorname{Cov}(y_T), \tag{15}$$

If $y_T \sim p_T = \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_T)$,

$$\mathbf{\Sigma}_t^{\mathsf{SDE}} = \mathbf{\Sigma}_t^{\mathsf{ODE}} = \mathbf{\Sigma}_t, \quad 0 \leqslant t \leqslant T.$$

If $y_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$,

$$\mathbf{\Sigma}_t^{\mathsf{SDE}} = \mathbf{\Sigma}_t + e^{-2(B_T - B_t)}\mathbf{\Sigma}_t^2\mathbf{\Sigma}_T^{-2}(\boldsymbol{I} - \mathbf{\Sigma}_T), \quad 0 \leqslant t \leqslant T.$$

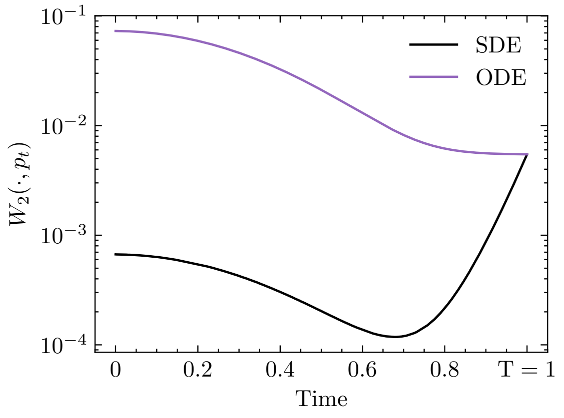
$$\mathbf{\Sigma}_t^{\mathsf{ODE}} = \mathbf{\Sigma}_t\mathbf{\Sigma}_T^{-1}$$

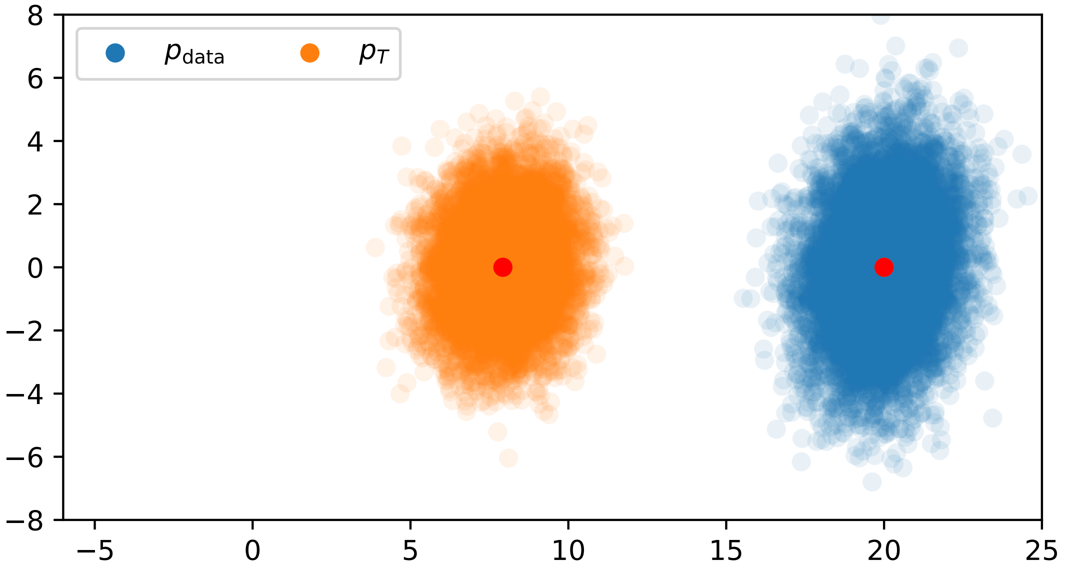
SDE vs ODE

Proposition 8: Marginals of the generative processes under Gaussian assumption

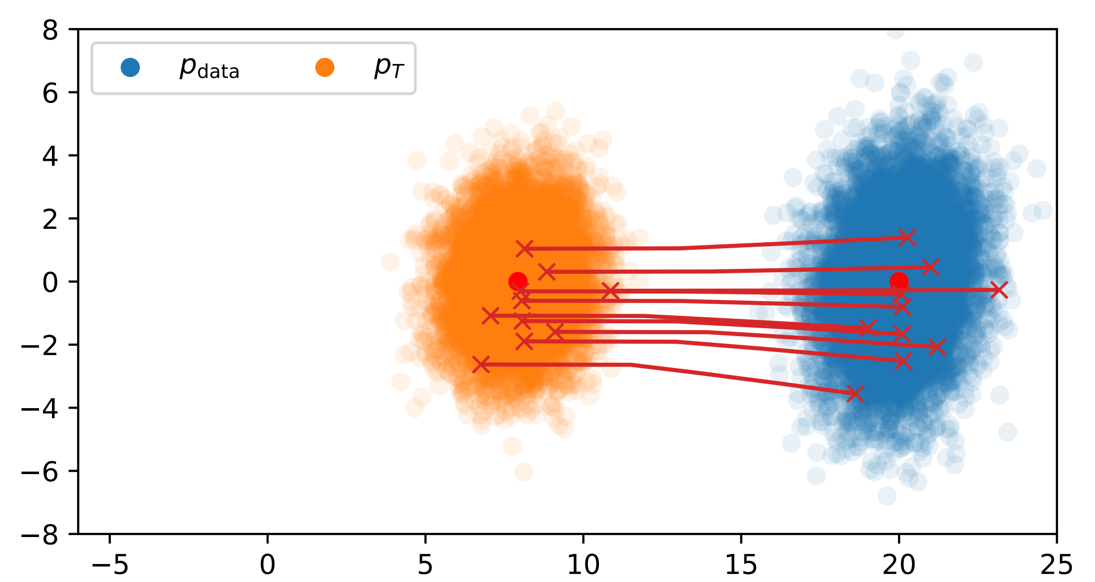
Under Gaussian assumption, for all $0 \leqslant t \leqslant T$,

$$\mathbf{W}_2(p_t^{\mathsf{SDE}}, p_t) \leqslant \mathbf{W}_2(p_t^{\mathsf{ODE}}, p_t) \qquad (16)$$

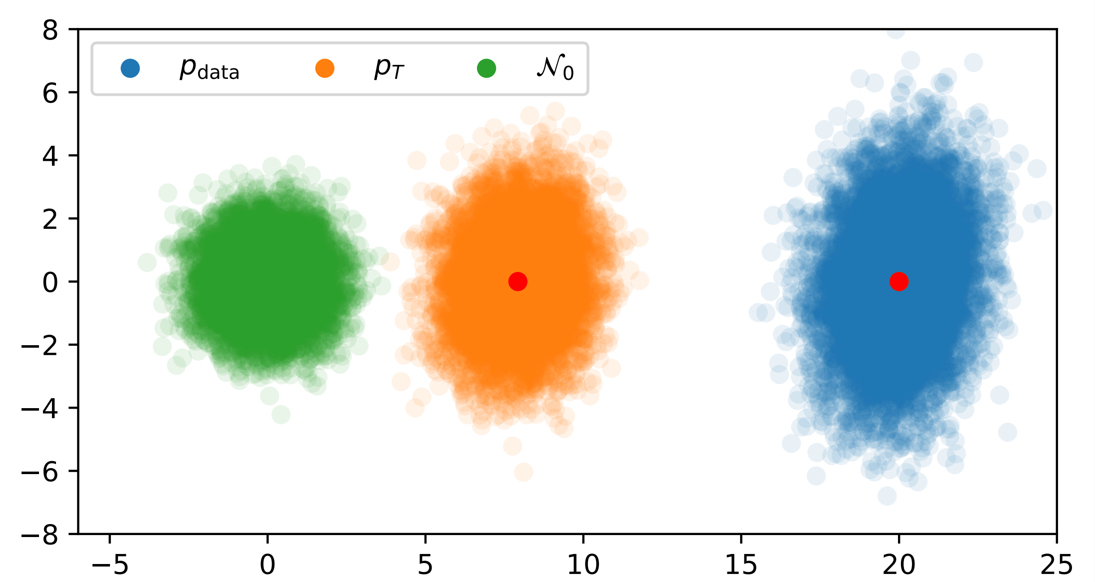


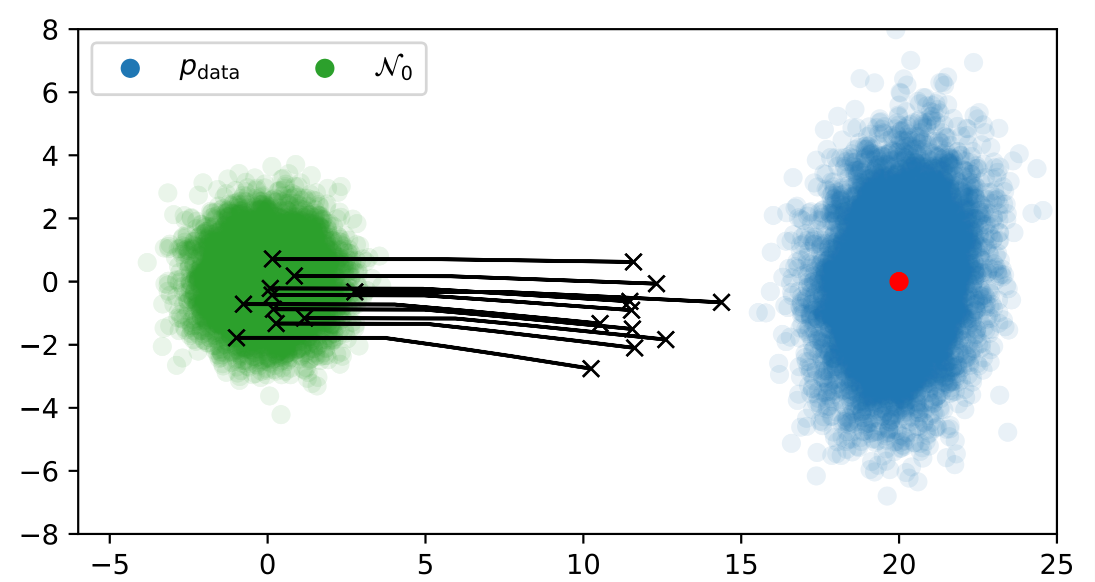

p_{data}
p_T
8
6
4
2
0
−2
−4
−6
−8
−5
0
5
10
15
20
25

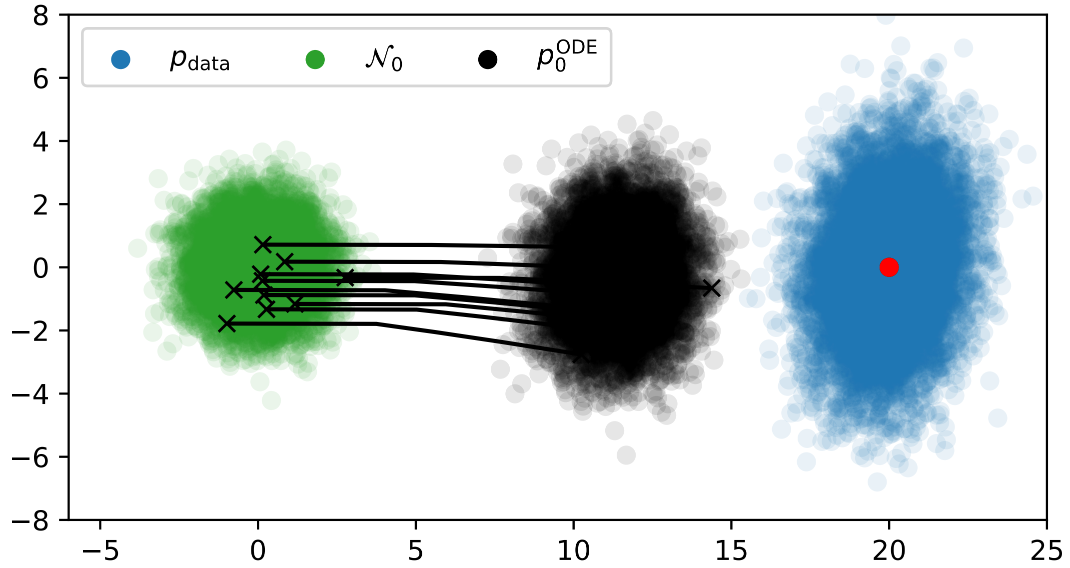
Initialization error: Focus on the ODE

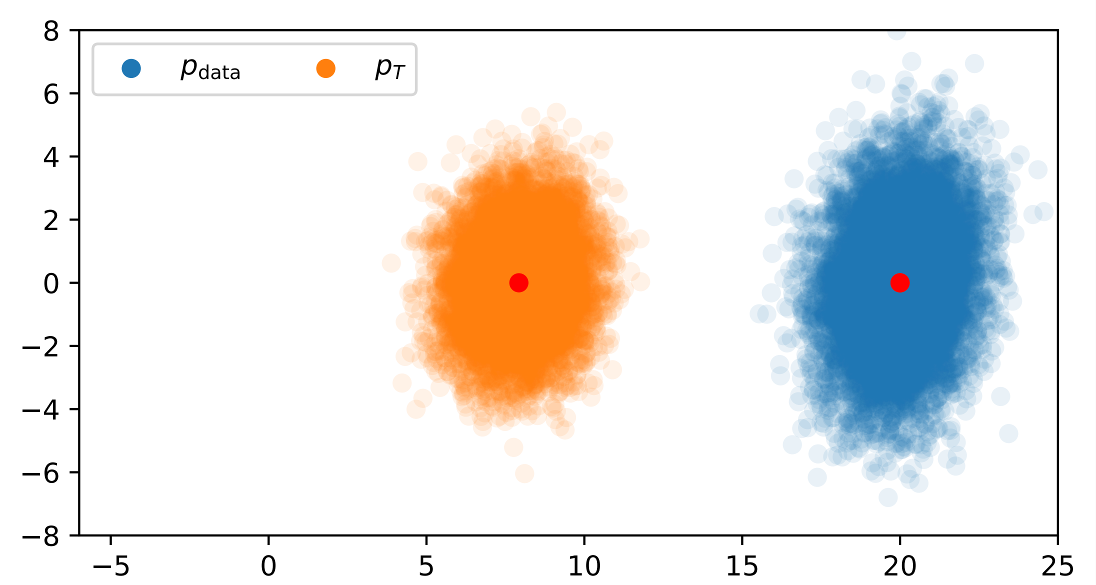


Initialization error: Focus on the ODE

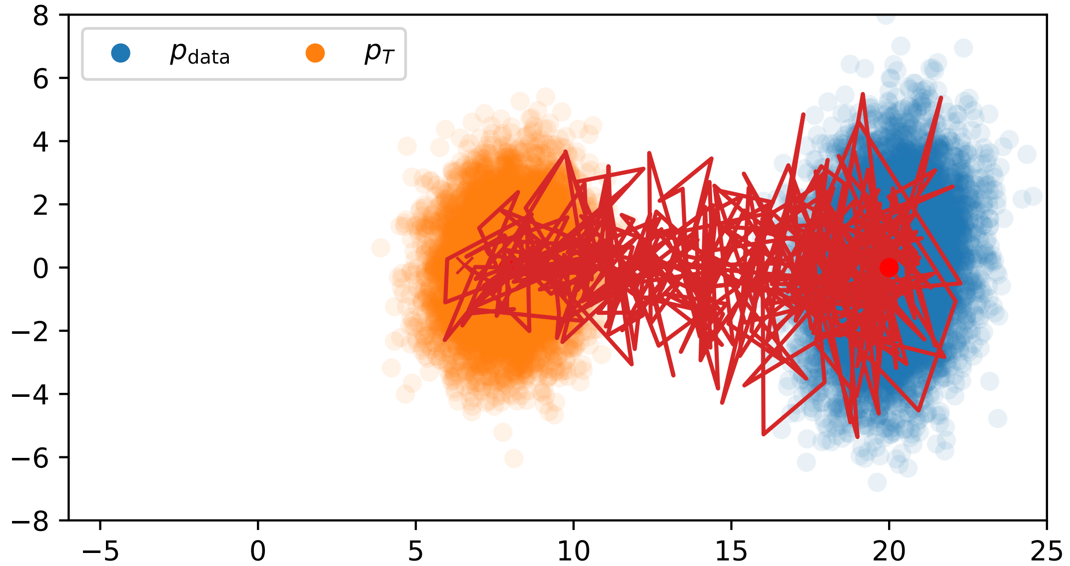


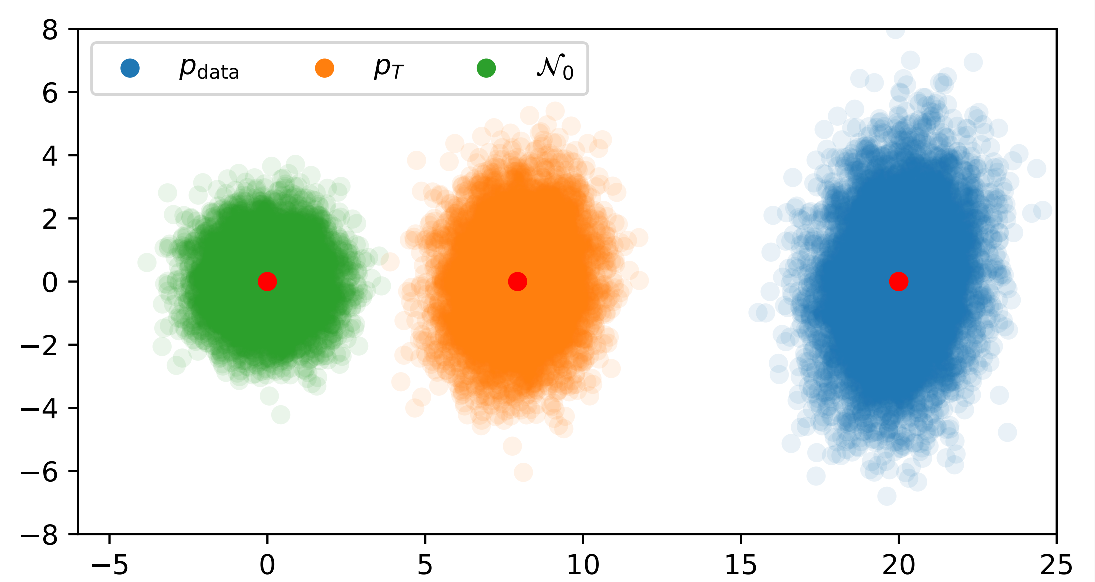

p_{data}
$\mathcal{N}_0$
8
6
4
2
0
−2
−4
−6
−8
−5
0
5
10
15
20
25


p_{data}
$\mathcal{N}_0$
p_0^{ODE}
8
6
4
2
0
−2
−4
−6
−8
−5
0
5
10
15
20
25

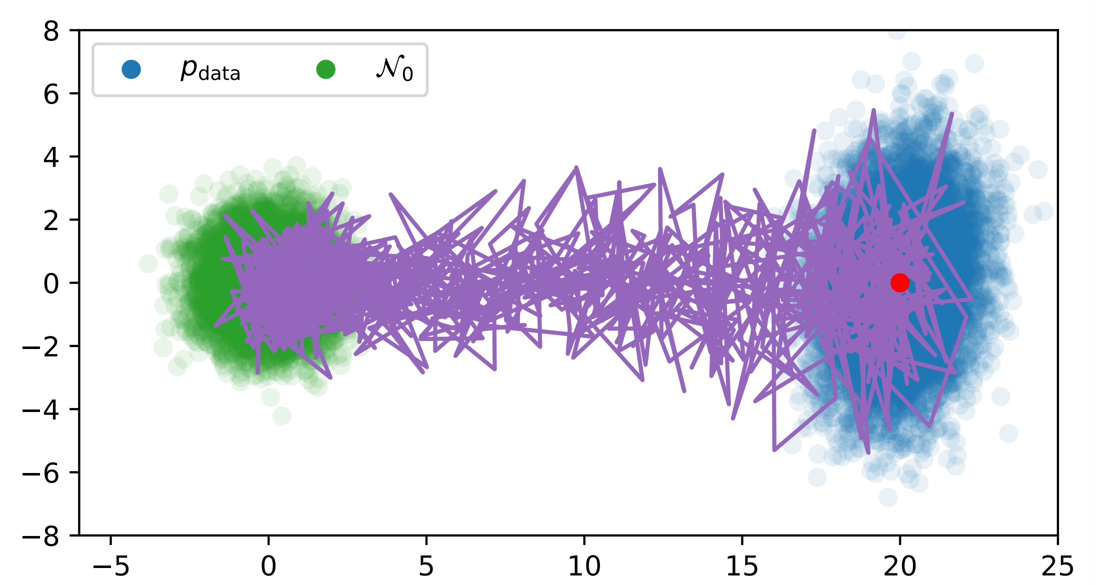

p_{data}
p_T
8
6
4
2
0
−2
−4
−6
−8
−5
0
5
10
15
20
25

Initialization error: Focus on the SDE

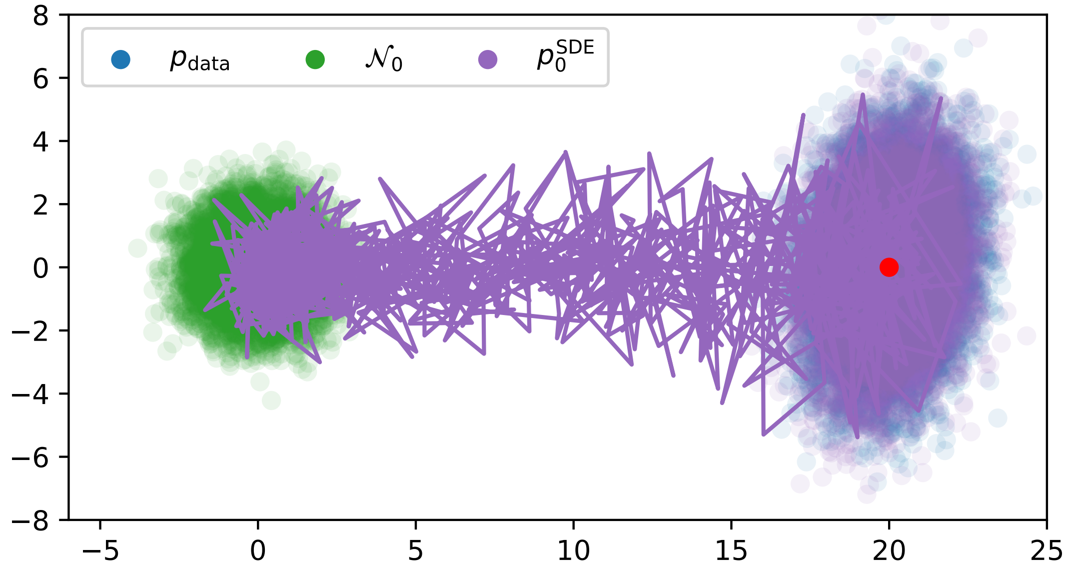


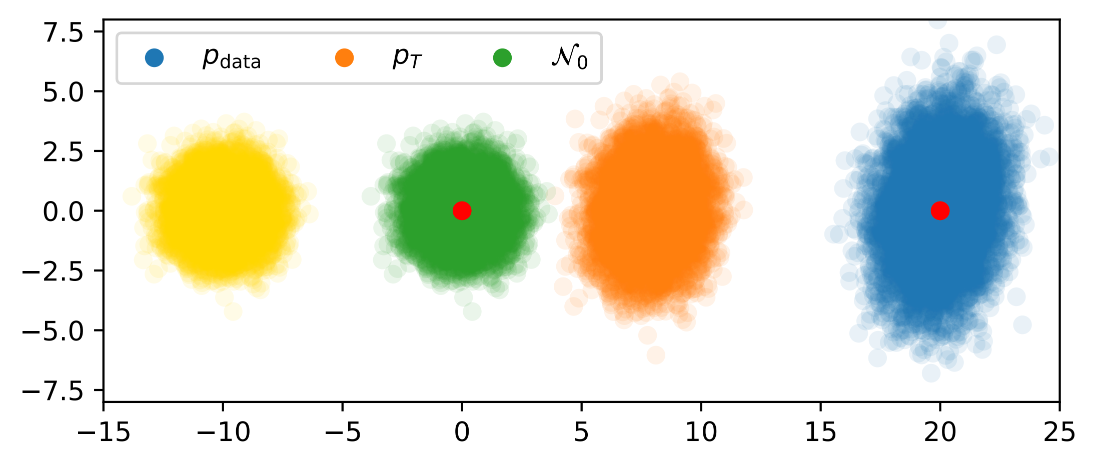

p_{data}
p_T
$\mathcal{N}_0$
8
6
4
2
0
−2
−4
−6
−8
−5
0
5
10
15
20
25

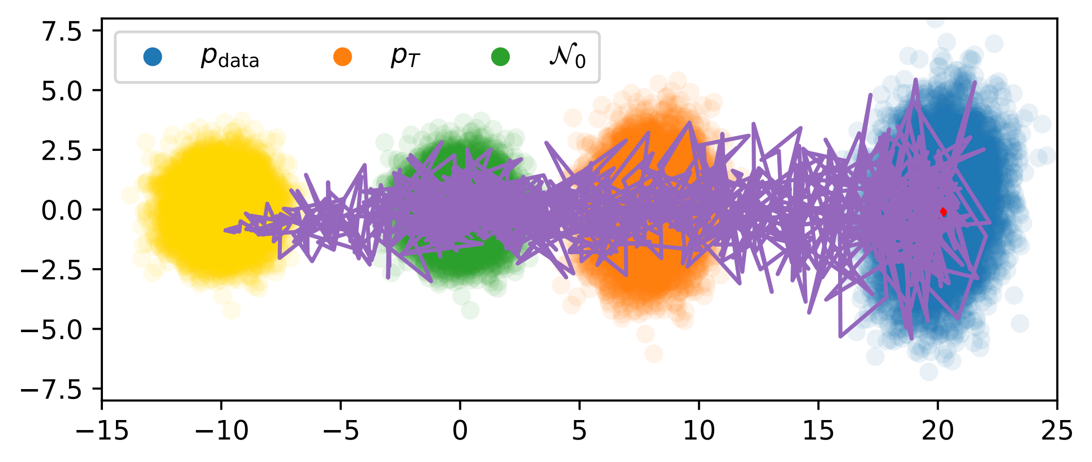
Initialization error: Focus on the SDE

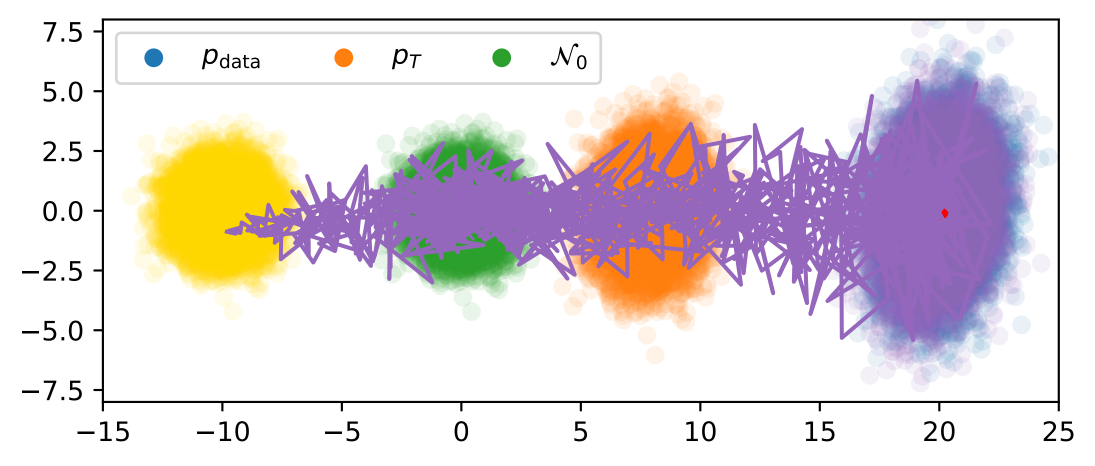


Initialization error: Focus on the SDE




p_{data}
p_T
$\mathcal{N}_0$
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−15
−10
−5
0
5
10
15
20
25


p_{data}
p_T
$\mathcal{N}_0$
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−15
−10
−5
0
5
10
15
20
25



p_{data}
p_T
$\mathcal{N}_0$
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−15
−10
−5
0
5
10
15
20
25

Exponential forgetting of the initial condition

Under Gaussian assumption, the strong solution to SDE (5) can be written as:

$$y_t^{\text{SDE}} = e^{-(B_T - B_t)} \boldsymbol{\Sigma}_t \boldsymbol{\Sigma}_T^{-1} y_T + \boldsymbol{\xi}_t, \quad 0 \leqslant t \leqslant T \tag{17}$$

Under Gaussian assumption, the solution to ODE (11) can be written as:

$$y_t^{\text{ODE}} = \boldsymbol{\Sigma}_T^{-1/2} \boldsymbol{\Sigma}_t^{1/2} y_T, \quad 0 \leqslant t \leqslant T, \tag{18}$$

- $y \mapsto \boldsymbol{\Sigma}_T^{-1/2} \boldsymbol{\Sigma}_t^{1/2} y$ is the transport map between p_T and p_t.
- False in general:, see [Lavenant and Santambrogio 2022][4]
- However, used in [Khrulkov et al. 2023][5]

---

[4] Hugo Lavenant and Filippo Santambrogio (2022). “The flow map of the Fokker–Planck equation does not provide optimal transport”. In: *Applied Mathematics Letters* 133, p. 108225. ISSN: 0893-9659. DOI: https://doi.org/10.1016/j.aml.2022.108225. URL: https://www.sciencedirect.com/science/article/pii/S089396592200180X

[5] Valentin Khrulkov et al. (2023). “Understanding DDPM Latent Codes Through Optimal Transport”. In: *The Eleventh International Conference on Learning Representations*. URL: https://openreview.net/forum?id=6PIrhAx1j4i

Truncation error

Truncation error

$$\begin{aligned} dy_t &= -\beta_t[y_t + 2s_\theta(t, y_t)]dt + \sqrt{2\beta_t}d\overline{w}_t, \\ &\text{or} \\ dy_t &= -\beta_t[y_t + s_\theta(t, y_t)]dt, \end{aligned} \quad \text{where} \quad \varepsilon \leqslant t \leqslant T, \quad y_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}). \tag{19}$$

Sampling a distribution using diffusion models implies different choices and error types:

- p_T, which is unknown, is replaced by $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ → **initialization error**
- In fact, another time ε to consider them on $[\varepsilon, T]$ → **truncation error**
- A scheme to discretize the equations → **discretization error**
- A model/neural network s_θ to learn the score → **score approximation error**

Truncation error under Gaussian assumption

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

$$\nabla \log p_t(x) = -\mathbf{\Sigma}_t^{-1} x, \quad 0 < t \leqslant T \tag{20}$$

with $\mathbf{\Sigma}_t = e^{-2B_t}\mathbf{\Sigma} + (1 - e^{-2B_t})\boldsymbol{I}$.

Truncation error under Gaussian assumption

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

$$\nabla \log p_t(x) = -\mathbf{\Sigma}_t^{-1} x, \quad 0 < t \leqslant T \tag{20}$$

with $\mathbf{\Sigma}_t = e^{-2B_t}\mathbf{\Sigma} + (1 - e^{-2B_t})\boldsymbol{I}$. At time $t = 0$,

$$\mathbf{\Sigma}_0 = \mathbf{\Sigma}$$

Truncation error under Gaussian assumption

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

$$\nabla \log p_t(x) = -\mathbf{\Sigma}_t^{-1} x, \quad 0 < t \leqslant T \tag{20}$$

with $\mathbf{\Sigma}_t = e^{-2B_t}\mathbf{\Sigma} + (1 - e^{-2B_t})\boldsymbol{I}$. At time $t = 0$,

$$\mathbf{\Sigma}_0 = \mathbf{\Sigma}$$

Consequently, $\nabla \log p_0(x)$ is not defined in general.

Truncation error under Gaussian assumption

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

$$\nabla \log p_t(x) = -\boldsymbol{\Sigma}_t^{-1} x, \quad 0 < t \leqslant T \tag{20}$$

with $\boldsymbol{\Sigma}_t = e^{-2B_t}\boldsymbol{\Sigma} + (1 - e^{-2B_t})\boldsymbol{I}$. At time $t = 0$,

$$\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}$$

Consequently, $\nabla \log p_0(x)$ is not defined in general.

Discretization error

Discretization error

$$dy_t = -\beta_t[y_t + 2s_\theta(t, y_t)]dt + \sqrt{2\beta_t}d\overline{w}_t,$$

or \quad where $\varepsilon \leqslant t \leqslant T, \quad y_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}).$ (21)

$$dy_t = -\beta_t[y_t + s_\theta(t, y_t)]dt,$$

Sampling a distribution using diffusion models implies different choices and error types:

- p_T, which is unknown, is replaced by $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ → **initialization error**
- In fact, another time ε to consider them on $[\varepsilon, T]$ → **truncation error**
- A scheme to discretize the equations → **discretization error**
- A model/neural network s_θ to learn the score → **score approximation error**

Discretization schemes

The EM discretized process is a Gaussian processe:

$$\begin{cases} \tilde{\boldsymbol{y}}_0^{\Delta,\mathsf{EM}} & \sim \mathcal{N}_0 \\ \tilde{\boldsymbol{y}}_1^{\Delta,\mathsf{EM}} & = \tilde{\boldsymbol{y}}_0^{\Delta,\mathsf{EM}} + \Delta_t \beta_{T-t_0} \left(\tilde{\boldsymbol{y}}_0^{\Delta,\mathsf{EM}} - 2\boldsymbol{\Sigma}_{T-t_0}^{-1} \tilde{\boldsymbol{y}}_0^{\Delta,\mathsf{EM}} \right) + \sqrt{2\Delta_t \beta_{T-t_0}} z_0, \; z_0 \sim \mathcal{N}_0 \end{cases} \tag{14}$$

Discretization schemes

The EM discretized process is a Gaussian processe:

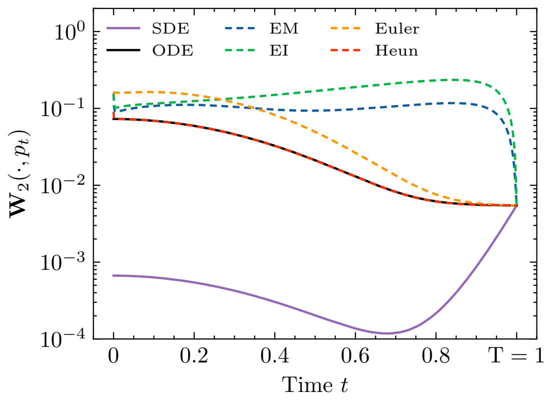
$$\begin{cases} \tilde{\boldsymbol{y}}_0^{\Delta,\mathsf{EM}} & \sim \mathcal{N}_0 \\ \tilde{\boldsymbol{y}}_1^{\Delta,\mathsf{EM}} & = \tilde{\boldsymbol{y}}_0^{\Delta,\mathsf{EM}} + \Delta_t \beta_{T-t_0} \left(\tilde{\boldsymbol{y}}_0^{\Delta,\mathsf{EM}} - 2\boldsymbol{\Sigma}_{T-t_0}^{-1} \tilde{\boldsymbol{y}}_0^{\Delta,\mathsf{EM}} \right) + \sqrt{2\Delta_t \beta_{T-t_0}} z_0, \; z_0 \sim \mathcal{N}_0 \end{cases} \tag{14}$$

We studied

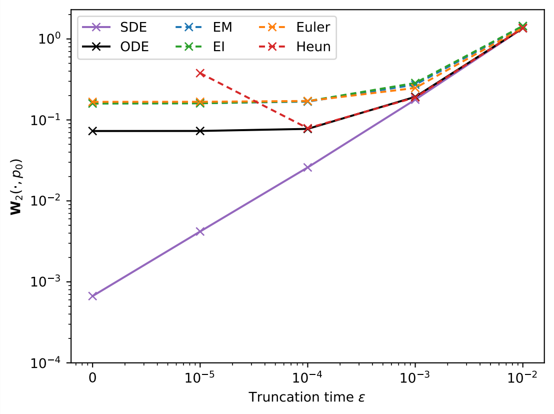
SDE schemes	ODE schemes
Euler-Maruyama (EM)	Euler
Exponential Integrator (EI)	Heun
DDPM [Ho et al., 2020]	Runge-Kutta 4 (RK4)

$\Longrightarrow$ With the perfect score, the discretized processes remain Gaussian processes.

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



Initialization + discretization



Truncation + Initialization

Score approximation error

Score approximation error

$$
\begin{aligned}
&dy_t = -\beta_t[y_t + 2s_\theta(t, y_t)]dt + \sqrt{2\beta_t}d\overline{w}_t, \\
&\text{or} \qquad\qquad\qquad\qquad\qquad\qquad \text{where} \quad \varepsilon \leqslant t \leqslant T, \quad y_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}). \\
&dy_t = -\beta_t[y_t + s_\theta(t, y_t)]dt,
\end{aligned}
\tag{22}
$$

Sampling a distribution using diffusion models implies different choices and error types:

- p_T, which is unknown, is replaced by $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ $\longrightarrow$ **initialization error**
- In fact, another time ε to consider them on $[\varepsilon, T]$ $\longrightarrow$ **truncation error**
- A scheme to discretize the equations $\longrightarrow$ **discretization error**
- A model/neural network alerts_θ to learn the score $\longrightarrow$ **score approximation error**

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne, Gousseau, and Morel 2011] [6]

A Gaussian distribution, named $\mathrm{ADSN}(\boldsymbol{u})$, can be associated with a texton $\boldsymbol{u}$:

$\boldsymbol{u}$

Samples of $\mathrm{ADSN}(\boldsymbol{u})$

[6] Bruno Galerne, Yann Gousseau, and Jean-Michel Morel (2011). "Random Phase Textures: Theory and Synthesis". In: *IEEE Transactions on Image Processing* 20.1, pp. 257–267. DOI: 10.1109/TIP.2010.2052822

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne, Gousseau, and Morel 2011] [6]

A Gaussian distribution, named $\mathrm{ADSN}(\boldsymbol{u})$, can be associated with a texton $\boldsymbol{u}$:

$\boldsymbol{u}$

Samples of $\mathrm{ADSN}(\boldsymbol{u})$

[6] Bruno Galerne, Yann Gousseau, and Jean-Michel Morel (2011). "Random Phase Textures: Theory and Synthesis". In: *IEEE Transactions on Image Processing* 20.1, pp. 257–267. DOI: 10.1109/TIP.2010.2052822

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne, Gousseau, and Morel 2011] [6]

A Gaussian distribution, named $\mathrm{ADSN}(\boldsymbol{u})$, can be associated with a texton $\boldsymbol{u}$:

$\boldsymbol{u}$

Samples of $\mathrm{ADSN}(\boldsymbol{u})$

[6] Bruno Galerne, Yann Gousseau, and Jean-Michel Morel (2011). "Random Phase Textures: Theory and Synthesis". In: *IEEE Transactions on Image Processing* 20.1, pp. 257–267. DOI: 10.1109/TIP.2010.2052822

A Gaussian distribution, named $\mathrm{ADSN}(\boldsymbol{u})$, can be associated with a texton $\boldsymbol{u}$:

$\boldsymbol{u}$

Samples of $\mathrm{ADSN}(\boldsymbol{u})$

[6] Bruno Galerne, Yann Gousseau, and Jean-Michel Morel (2011). "Random Phase Textures: Theory and Synthesis". In: *IEEE Transactions on Image Processing* 20.1, pp. 257–267. DOI: 10.1109/TIP.2010.2052822

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne, Gousseau, and Morel 2011] [6]

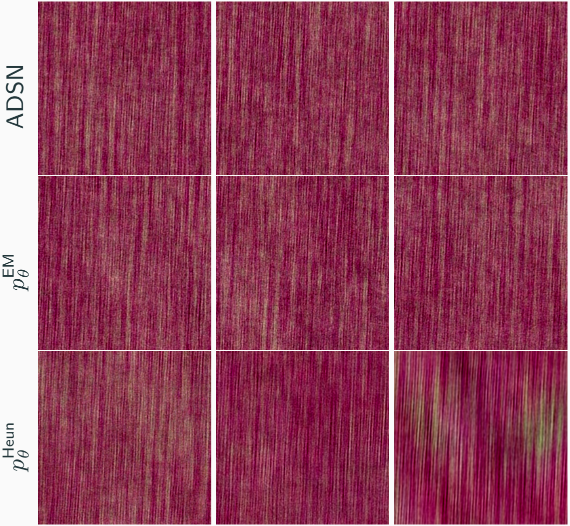
A Gaussian distribution, named $\mathrm{ADSN}(\boldsymbol{u})$, can be associated with a texton $\boldsymbol{u}$:

$\boldsymbol{u}$

Samples of $\mathrm{ADSN}(\boldsymbol{u})$

[6] Bruno Galerne, Yann Gousseau, and Jean-Michel Morel (2011). "Random Phase Textures: Theory and Synthesis". In: *IEEE Transactions on Image Processing* 20.1, pp. 257–267. DOI: 10.1109/TIP.2010.2052822

The score approximation error



- We train a diffusion model to generate an ADSN model.
- The stochastic EM is more resilient to the initialization error than the deterministic Heun's scheme.

Conclusion

- Theoretically, the backward SDE is more resilient to initialization errors than the ODE.
- Theoretically, Heun's scheme applied to the ODE is the recommended method for using diffusion models.
- In practice, the SDE is more robust to score approximation: the noise dilutes the errors at each step.
- For further study on score approximation
  → see Samuel Hurault et al. (2025). *From Denoising Score Matching to Langevin Sampling: A Fine-Grained Error Analysis in the Gaussian Setting*. arXiv: 2503.11615 [cs.LG]. URL: https://arxiv.org/abs/2503.11615

Extension to conditional diffusion models

Conditional diffusion models

We define the inverse problem

$$\boldsymbol{v} = \boldsymbol{A}x_0 + \sigma\boldsymbol{n}, \quad \boldsymbol{n} \sim \mathcal{N}_0 \tag{23}$$

where $\boldsymbol{A}$ is a linear degradation operator.

Conditional diffusion models

We define the inverse problem

$$\boldsymbol{v} = \boldsymbol{A}x_0 + \sigma\boldsymbol{n}, \quad \boldsymbol{n} \sim \mathcal{N}_0 \tag{23}$$

where $\boldsymbol{A}$ is a linear degradation operator. To sample the posterior $p(x_0 \mid \boldsymbol{v})$, the backward

$$dy_t = \beta_{T-t}(y_t + 2\nabla \log p_{T-t}(y_t))dt + \sqrt{2\beta_{T-t}}d\overline{w}_t, \quad 0 \leqslant t \leqslant T, \quad y_0 \sim p_T, \tag{24}$$

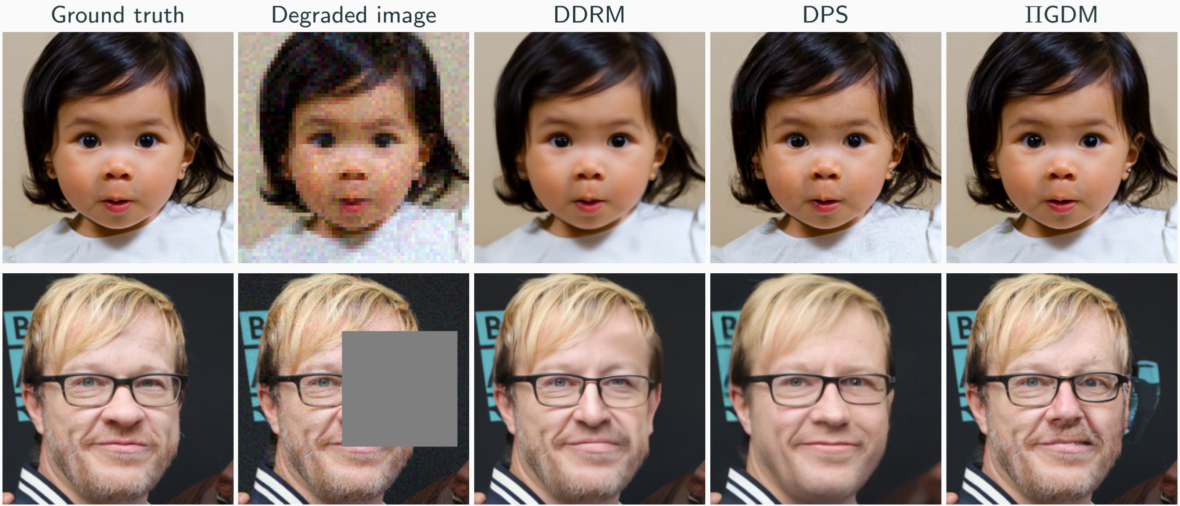
can be replaced by

$$dy_t = \beta_t(y_t + 2\nabla \log p_t(y_t \mid \boldsymbol{v}))dt + \sqrt{2\beta_t}d\overline{w}_t, \quad 0 \leqslant t \leqslant T, \quad y_T \sim p_T. \tag{25}$$

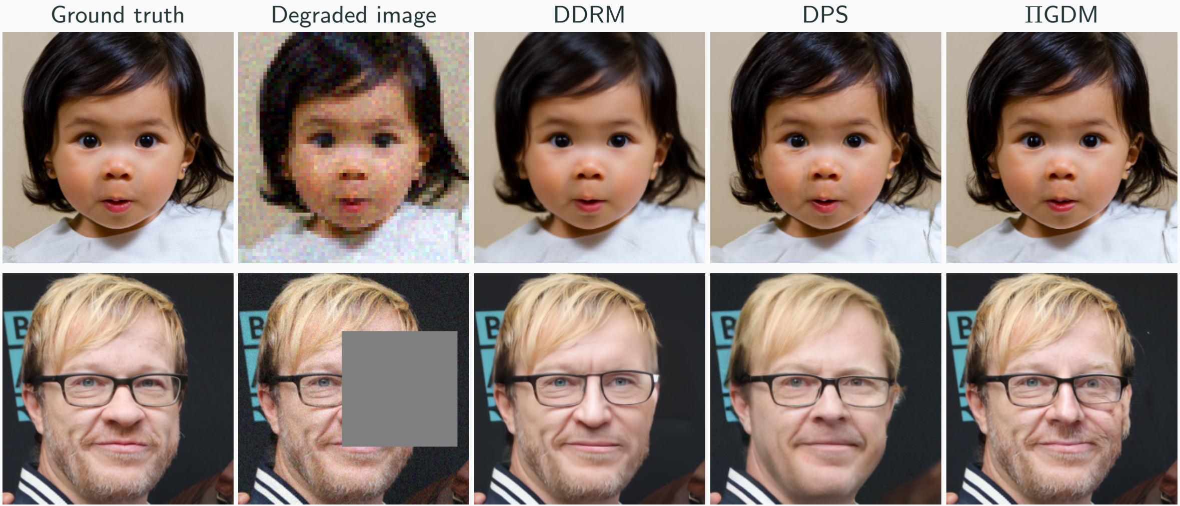
By Bayes' formula,

$$\nabla \log p_t(y_t \mid \boldsymbol{v}) = \nabla \log p_t(y_t) + \nabla \log p_t(\boldsymbol{v} \mid y_t). \tag{26}$$

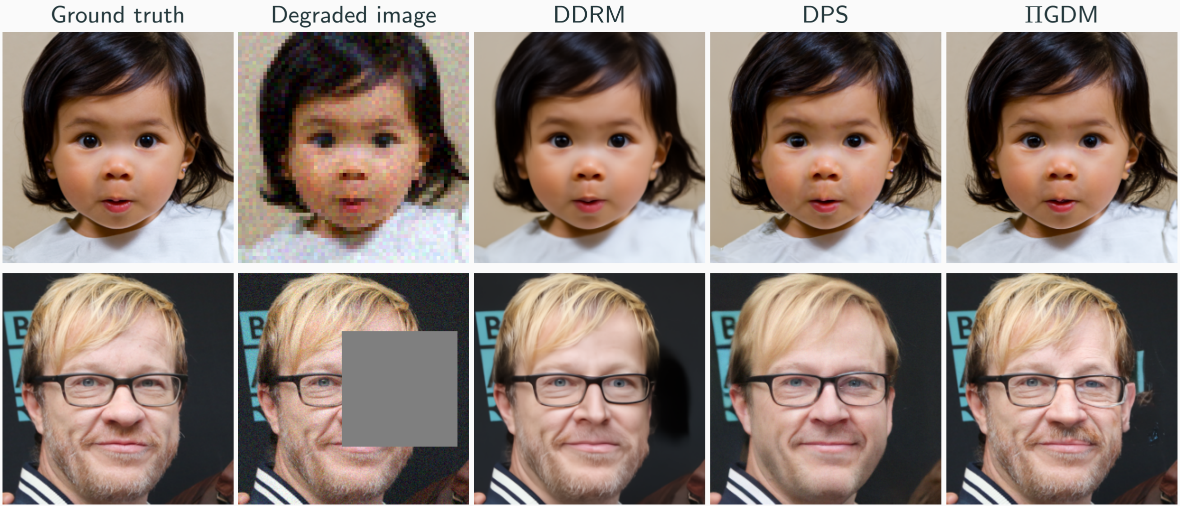
Example from state of the art



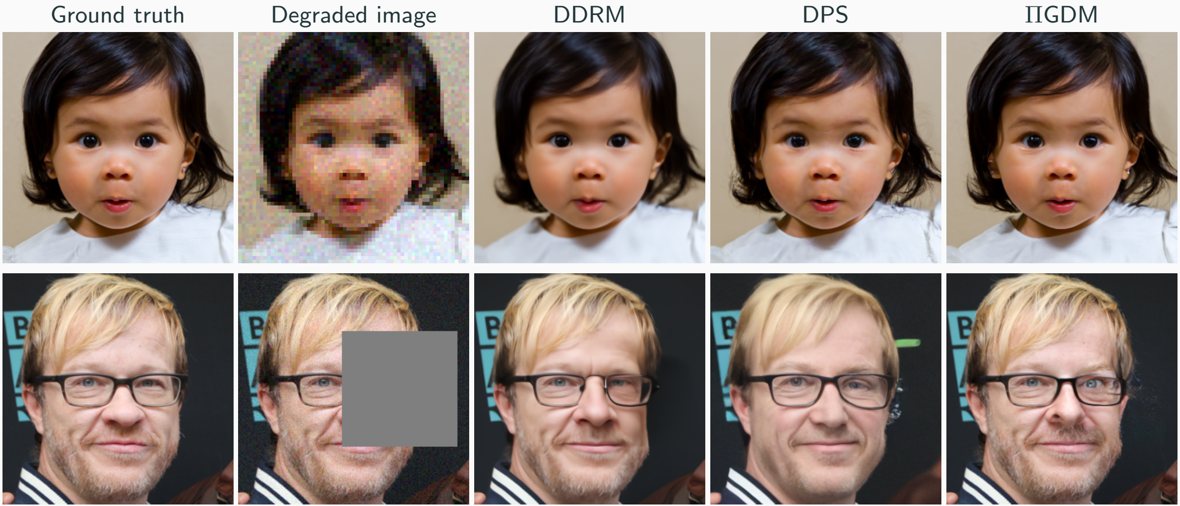
Example from state of the art



Example from state of the art



Example from state of the art



The evaluation of conditional diffusion models in the literature is essentially empirical, using the Frechet Inception Distance (FID).

Method	SR (×4)		Inpaint (box)		Inpaint (random)		Deblur (gauss)		Deblur (motion)	
	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓
DPS (ours)	**39.35**	**0.214**	**33.12**	**0.168**	**21.19**	**0.212**	**44.05**	**0.257**	**39.92**	**0.242**
DDRM (Kawar et al., 2022)	62.15	0.294	42.93	0.204	69.71	0.587	74.92	0.332	-	-
MCG (Chung et al., 2022a)	87.64	0.520	40.11	0.309	29.26	0.286	101.2	0.340	310.5	0.702
PnP-ADMM (Chan et al., 2016)	66.52	0.353	151.9	0.406	123.6	0.692	90.42	0.441	89.08	0.405
Score-SDE (Song et al., 2021b) (ILVR (Choi et al., 2021))	96.72	0.563	60.06	0.331	76.54	0.612	109.0	0.403	292.2	0.657
ADMM-TV	110.6	0.428	68.94	0.322	181.5	0.463	186.7	0.507	152.3	0.508

Table extracted from [Chung, Sim, and Ye 2022]

Evaluation of conditional diffusion models in the literature

The evaluation of conditional diffusion models in the literature is essentially empirical, using the Frechet Inception Distance (FID).

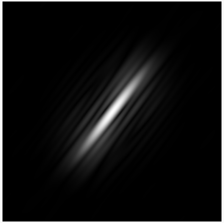
	SR (×4)		**Inpaint (box)**		**Inpaint (random)**		**Deblur (gauss)**		**Deblur (motion)**	
Method	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓
DPS (ours)	**39.35**	**0.214**	**33.12**	**0.168**	**21.19**	**0.212**	**44.05**	**0.257**	**39.92**	**0.242**
DDRM (Kawar et al., 2022)	62.15	0.294	42.93	0.204	69.71	0.587	74.92	0.332	-	-
MCG (Chung et al., 2022a)	87.64	0.520	40.11	0.309	29.26	0.286	101.2	0.340	310.5	0.702
PnP-ADMM (Chan et al., 2016)	66.52	0.353	151.9	0.406	123.6	0.692	90.42	0.441	89.08	0.405
Score-SDE (Song et al., 2021b) (ILVR (Choi et al., 2021))	96.72	0.563	60.06	0.331	76.54	0.612	109.0	0.403	292.2	0.657
ADMM-TV	110.6	0.428	68.94	0.322	181.5	0.463	186.7	0.507	152.3	0.508

Table extracted from [Chung, Sim, and Ye 2022]

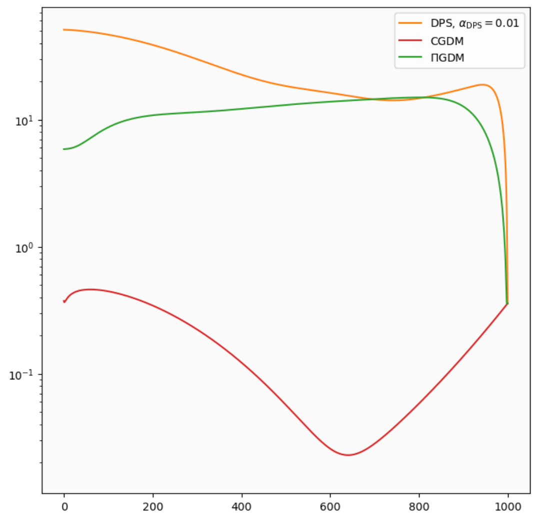
$\Longrightarrow$ We propose an exact Wassertein error evaluation.

Exact Wasserstein error for deblurring

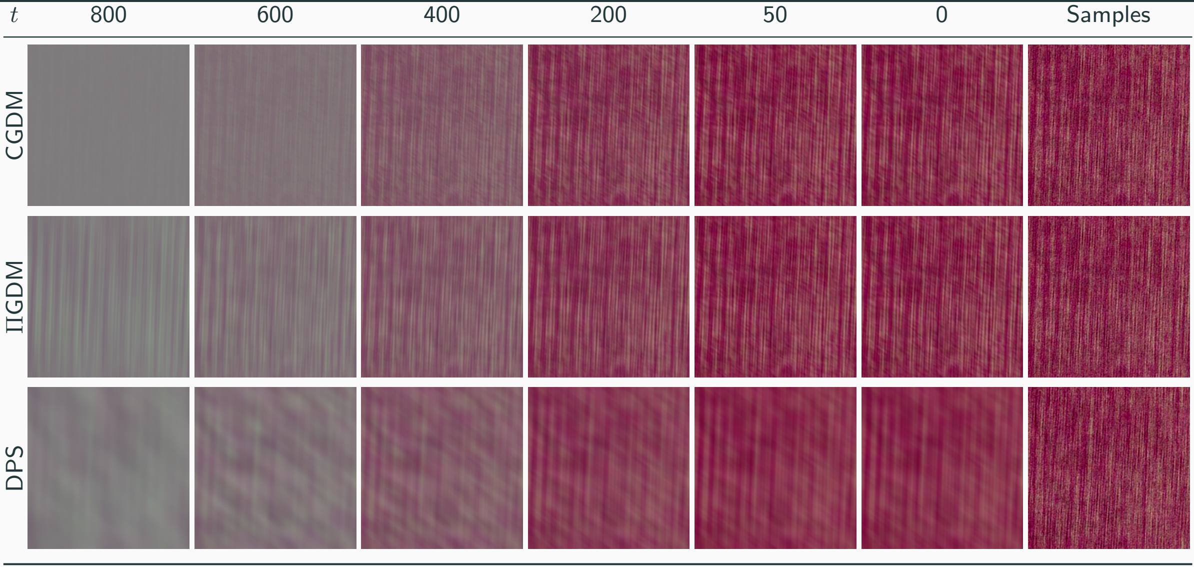
Blur kernel



Blurred image v



Study of the bias



Conclusion

Conclusion

- Extension to GMM ?

Conclusion

- Extension to GMM ?
  - Exact solutions are not known.

Conclusion

- Extension to GMM ?
  - Exact solutions are not known.
  - A closed form for the $\mathbf{W}_2$ distance is not known.

Conclusion

- Extension to GMM ?
  - Exact solutions are not known.
  - A closed form for the $\mathbf{W}_2$ distance is not known.
  - The discretized processes are no more GMM.

Conclusion

- Extension to GMM ?
  - Exact solutions are not known.
  - A closed form for the $\mathbf{W}_2$ distance is not known.
  - The discretized processes are no more GMM.
- To be compared with the bounds in the literature.
  → Stanislas Strasman et al. (2024). *An analysis of the noise schedule for score-based generative models*. arXiv: 2402.04650 [math.ST]. URL: https://arxiv.org/abs/2402.04650

Conclusion

- Extension to GMM ?
  - Exact solutions are not known.
  - A closed form for the $\mathbf{W}_2$ distance is not known.
  - The discretized processes are no more GMM.
- To be compared with the bounds in the literature.
  → Stanislas Strasman et al. (2024). *An analysis of the noise schedule for score-based generative models*. arXiv: 2402.04650 [math.ST]. URL: https://arxiv.org/abs/2402.04650
- Add theoretical score approximation ?
  → Samuel Hurault et al. (2025). *From Denoising Score Matching to Langevin Sampling: A Fine-Grained Error Analysis in the Gaussian Setting*. arXiv: 2503.11615 [cs.LG]. URL: https://arxiv.org/abs/2503.11615

Thank you for your attention !

References

Choi, Jooyoung et al. (2021). "ILVR: Conditioning Method for Denoising Diffusion Probabilistic Models". In: *ILVR*. Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14367–14376. URL: https://openaccess.thecvf.com/content/ICCV2021/html/Choi_ILVR_Conditioning_Method_for_Denoising_Diffusion_Probabilistic_Models_ICCV_2021_paper.html (visited on 2022-11-28).

Chung, Hyungjin, Byeongsu Sim, Dohoon Ryu, et al. (2022). "Improving Diffusion Models for Inverse Problems using Manifold Constraints". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Chung, Hyungjin, Byeongsu Sim, and Jong Chul Ye (2022). "Come-Closer-Diffuse-Faster: Accelerating Conditional Diffusion Models for Inverse Problems Through Stochastic Contraction". In: *Come-Closer-Diffuse-Faster*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12413–12422. URL: https://openaccess.thecvf.com/content/CVPR2022/html/Chung_Come-Closer-Diffuse-Faster_Accelerating_Conditional_Diffusion_Models_for_Inverse_Problems_Through_Stochastic_CVPR_2022_paper.html (visited on 2023-01-27).

De Bortoli, Valentin (2022). “Convergence of denoising diffusion models under the manifold hypothesis”. In: *Transactions on Machine Learning Research*. ISSN: 2835-8856. URL: https://openreview.net/forum?id=MhK5aXo3gB.

Galerne, Bruno, Yann Gousseau, and Jean-Michel Morel (2011). “Random Phase Textures: Theory and Synthesis”. In: *IEEE Transactions on Image Processing* 20.1, pp. 257–267. DOI: 10.1109/TIP.2010.2052822.

Hurault, Samuel et al. (2025). *From Denoising Score Matching to Langevin Sampling: A Fine-Grained Error Analysis in the Gaussian Setting*. arXiv: 2503.11615 [cs.LG]. URL: https://arxiv.org/abs/2503.11615.

Khrulkov, Valentin et al. (2023). “Understanding DDPM Latent Codes Through Optimal Transport”. In: *The Eleventh International Conference on Learning Representations*. URL: https://openreview.net/forum?id=6PIrhAx1j4i.

Lavenant, Hugo and Filippo Santambrogio (2022). “The flow map of the Fokker–Planck equation does not provide optimal transport”. In: *Applied Mathematics Letters* 133, p. 108225. ISSN: 0893-9659. DOI: https://doi.org/10.1016/j.aml.2022.108225. URL: https://www.sciencedirect.com/science/article/pii/S089396592200180X.

Lugmayr, Andreas et al. (2022). “RePaint: Inpainting using Denoising Diffusion Probabilistic Models”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461. URL: https://api.semanticscholar.org/CorpusID:246240274.

Pardoux, E. (1986). "Grossissement d'une filtration et retournement du temps d'une diffusion". In: *Séminaire de Probabilités XX 1984/85*. Ed. by Jacques Azéma and Marc Yor. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 48–55. ISBN: 978-3-540-39860-8.

Song, Yang et al. (2021). "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=PxTIG12RRHS.

Strasman, Stanislas et al. (2024). *An analysis of the noise schedule for score-based generative models*. arXiv: 2402.04650 [math.ST]. URL: https://arxiv.org/abs/2402.04650.

To the restoration problems ?

To the restoration problems ?

My thesis title: Stochastic super resolution using deep generative models

To the restoration problems ?

My thesis title: Stochastic super resolution using deep generative models

→ We need to use **conditional** diffusion model !

How to perform conditional simulation ?

What is the link with solving inverse problems $\boldsymbol{v} = \boldsymbol{A}x + \sigma\boldsymbol{\varepsilon}$?

How to perform conditional simulation ?

What is the link with solving inverse problems $\boldsymbol{v} = \boldsymbol{A}x + \sigma\boldsymbol{\varepsilon}$?

A large literature [Song et al. 2021[7],Lugmayr et al. 2022[8],Chung, Sim, Ryu, et al. 2022[9],Choi et al. 2021[10]] uses the Bayes formula

$$\nabla_x \log p_t(x_t \mid \boldsymbol{v}) = \nabla_x \log p_t(\boldsymbol{v} \mid x_t) + \nabla_x \log p_t(x_t). \tag{27}$$

where $\nabla_x \log p_t(x_t)$ is the unconditional score. Consequently, studying the unconditional case provides information for the conditional one.

---

[7]Yang Song et al. (2021). "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=PxTIG12RRHS

[8]Andreas Lugmayr et al. (2022). "RePaint: Inpainting using Denoising Diffusion Probabilistic Models". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11451–11461. URL: https://api.semanticscholar.org/CorpusID:246240274

[9]Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, et al. (2022). "Improving Diffusion Models for Inverse Problems using Manifold Constraints". In: *Advances in Neural Information Processing Systems (NeurIPS)*

[10]Jooyoung Choi et al. (2021). "ILVR: Conditioning Method for Denoising Diffusion Probabilistic Models". In: *ILVR*. Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14367–14376. URL: https://openaccess.thecvf.com/content/ICCV2021/html/Choi_ILVR_Conditioning_Method_for_Denoising_Diffusion_Probabilistic_Models_ICCV_2021_paper.html (visited on 2022-11-28)

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