Diffusion models for Gaussian distributions: Exact solutions and Wasserstein errors

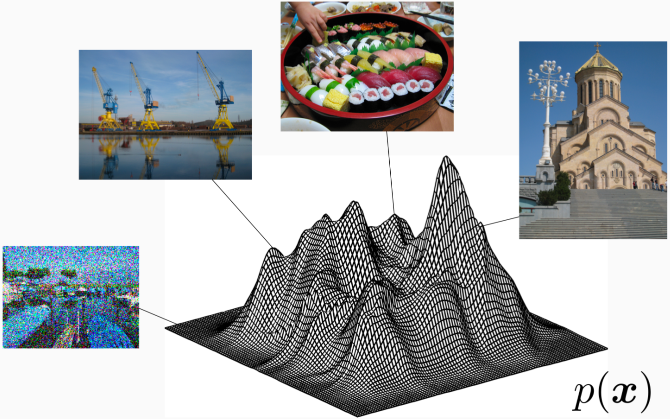
Émile Pierret[a], supervised by Bruno Galerne[a,b]
Mathematisches Forschungsinstitut Oberwolfach (MFO)
February 4^{th} 2025

[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

CelebA dataset

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 $G(\mathcal{N}(\mathbf{0}, \boldsymbol{I}_N), \Theta^\star) \approx p$

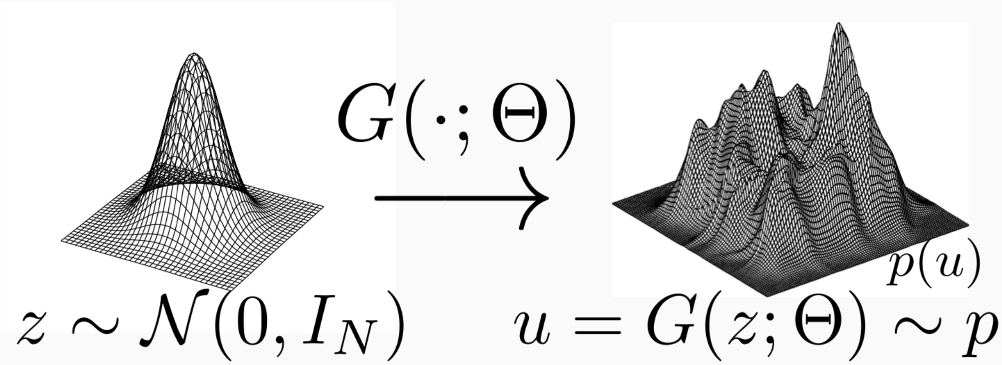


Image extracted from Bruno Galerne's slides

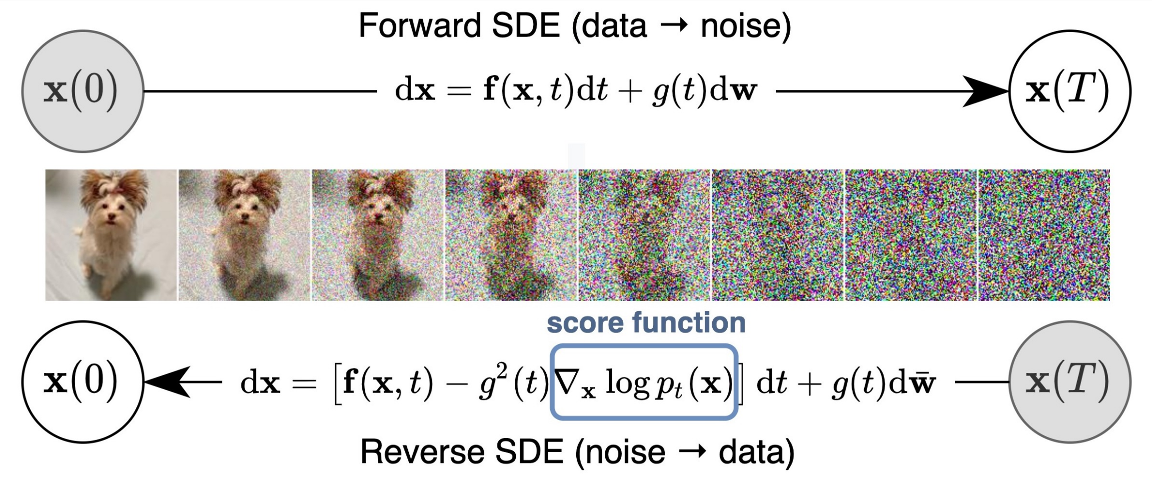


Image extracted from [Song et al. 2021]

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.

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

Furthermore, denoting $\boldsymbol{\Sigma}_t = \mathrm{Cov}(x_t)$,

$$d\boldsymbol{\Sigma}_t = 2\beta_t (\boldsymbol{I} - \boldsymbol{\Sigma}_t) dt, \quad 0 < t \leqslant T. \tag{3}$$

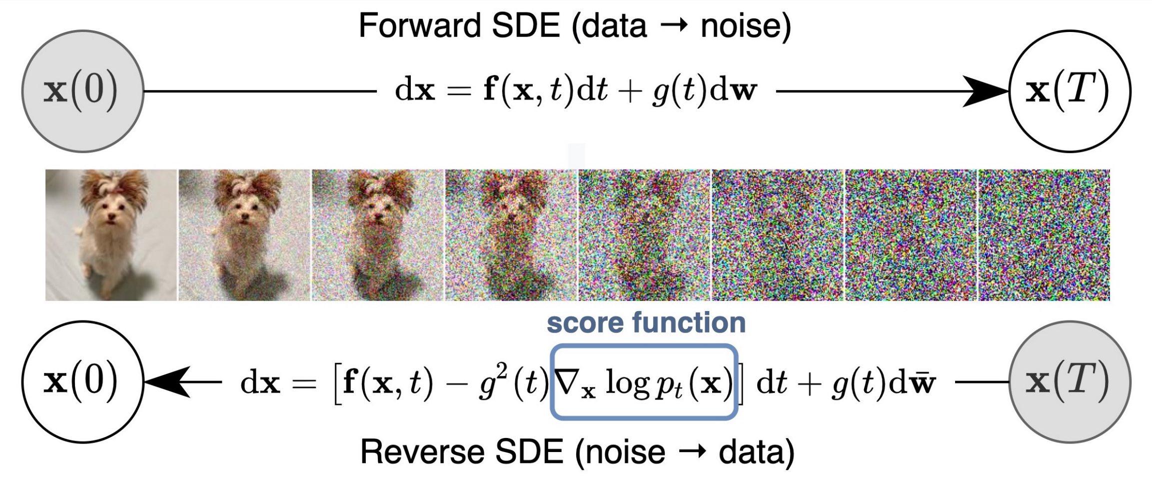


Image extracted from [Song et al. 2021]

Study of the backward process [Pardoux 1986][1]

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

$$dy_t = \beta_{T-t}(y_t + 2\nabla \log p_{T-t}(y_t))dt + \sqrt{2\beta_{T-t}}d\overline{w}_t, \quad 0 \leqslant t < T, \quad y_0 \sim p_T. \tag{4}$$

[1] 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][1]

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

$$dy_t = \beta_{T-t}(y_t + 2\nabla \log p_{T-t}(y_t))dt + \sqrt{2\beta_{T-t}}d\overline{w}_t, \quad 0 \leqslant t < T, \quad y_0 \sim p_T. \tag{4}$$

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

$$\overline{w}_t = w_t - w_T + \int_t^T \frac{1}{p(s, x_s)} \operatorname{div}(\sigma p)(s, x_s) ds. \tag{5}$$

[1] 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][1]

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

$$dy_t = \beta_{T-t}(y_t + 2\nabla \log p_{T-t}(y_t))dt + \sqrt{2\beta_{T-t}}d\overline{w}_t, \quad 0 \leqslant t < T, \quad y_0 \sim p_T. \tag{4}$$

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

$$\overline{w}_t = w_t - w_T + \int_t^T \frac{1}{p(s, x_s)} \operatorname{div}(\sigma p)(s, x_s) ds. \tag{5}$$

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

[1] 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][1]

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

$$dy_t = \beta_{T-t}(y_t + 2\nabla \log p_{T-t}(y_t))dt + \sqrt{2\beta_{T-t}}d\overline{w}_t, \quad 0 \leqslant t < T, \quad y_0 \sim p_T. \tag{4}$$

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

$$\overline{w}_t = w_t - w_T + \int_t^T \frac{1}{p(s, x_s)} \operatorname{div}(\sigma p)(s, x_s)ds. \tag{5}$$

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

[1] 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][1]

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

$$dy_t = \beta_{T-t}(y_t + 2\nabla \log p_{T-t}(y_t))dt + \sqrt{2\beta_{T-t}}d\overline{w}_t, \quad 0 \leqslant t < T, \quad y_0 \sim p_T. \tag{4}$$

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

$$\overline{w}_t = w_t - w_T + \int_t^T \frac{1}{p(s, x_s)} \operatorname{div}(\sigma p)(s, x_s)ds. \tag{5}$$

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

[1] 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][1]

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

$$dy_t = \beta_{T-t}(y_t + 2\nabla \log p_{T-t}(y_t))dt + \sqrt{2\beta_{T-t}}d\overline{w}_t, \quad 0 \leqslant t < T, \quad y_0 \sim p_T. \tag{4}$$

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

$$\overline{w}_t = w_t - w_T + \int_t^T \frac{1}{p(s, x_s)} \operatorname{div}(\sigma p)(s, x_s)ds. \tag{5}$$

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

$$dy_t = \beta_{T-t}(y_t + 2s_\theta(y_t, T-t))dt + \sqrt{2\beta_{T-t}}d\overline{w}_t, \quad 0 \leqslant t < T, \quad y_0 \sim p_T. \tag{6}$$

[1] 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

Diffusion models through SDE

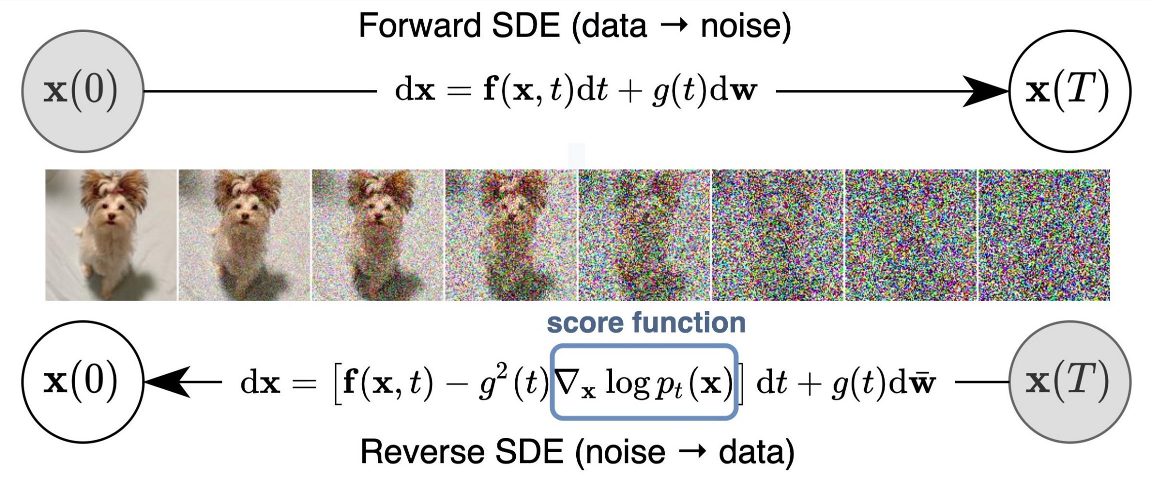
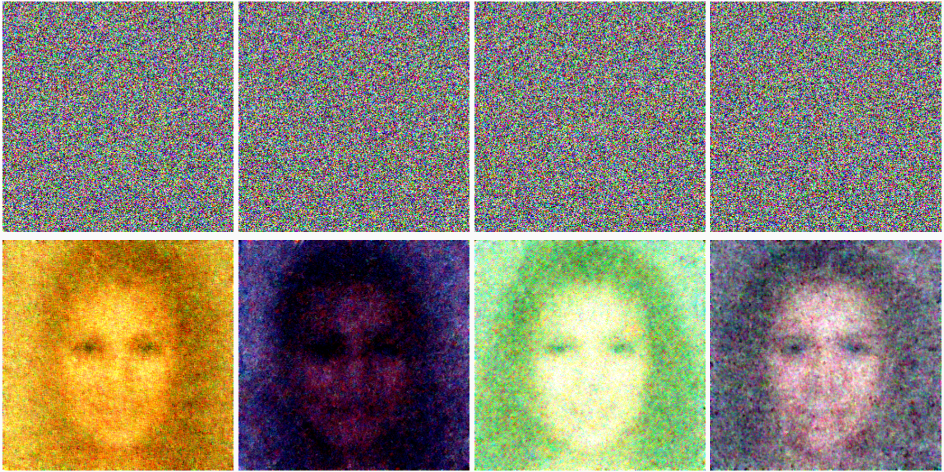


Image extracted from [Song et al. 2021]

Examples (generated with [Lugmayr et al. 2022][2])

x_t

$\widehat{x}_0(x_t)$



t = 249

[2] 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][2])

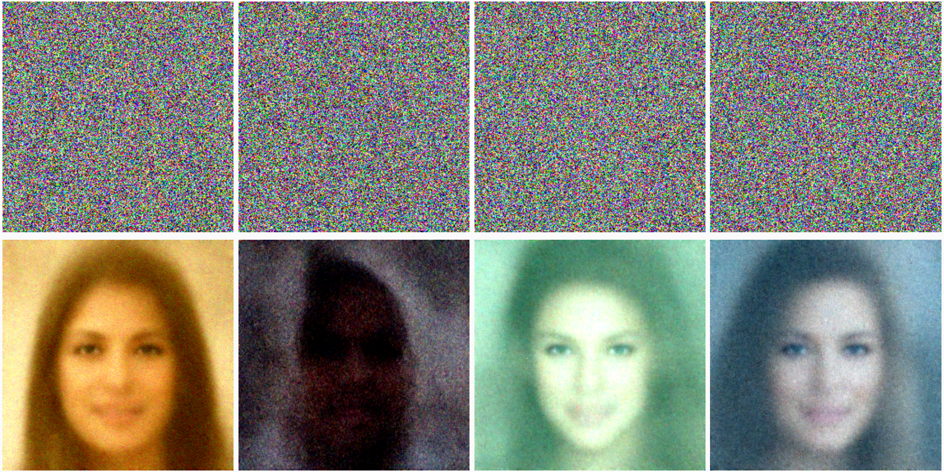
t = 230

[2] 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][2])

x_t

$\widehat{x}_0(x_t)$



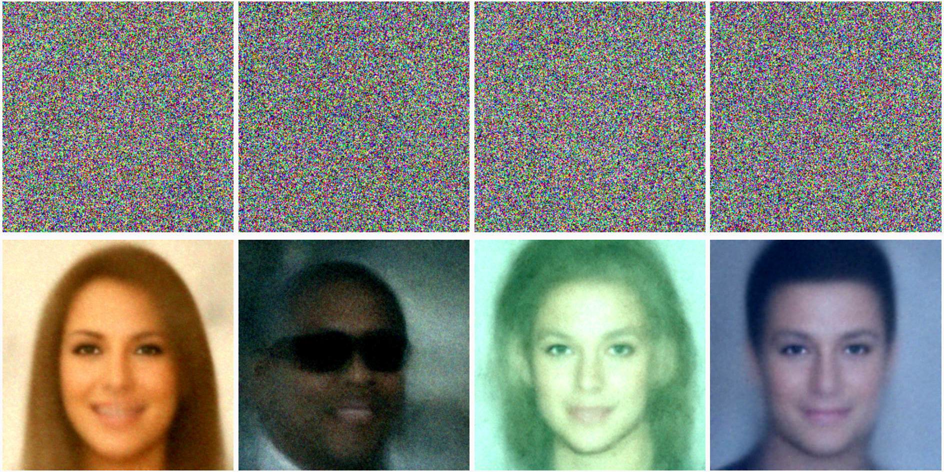
$t = 210$

[2] 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][2])

x_t

$\widehat{x}_0(x_t)$



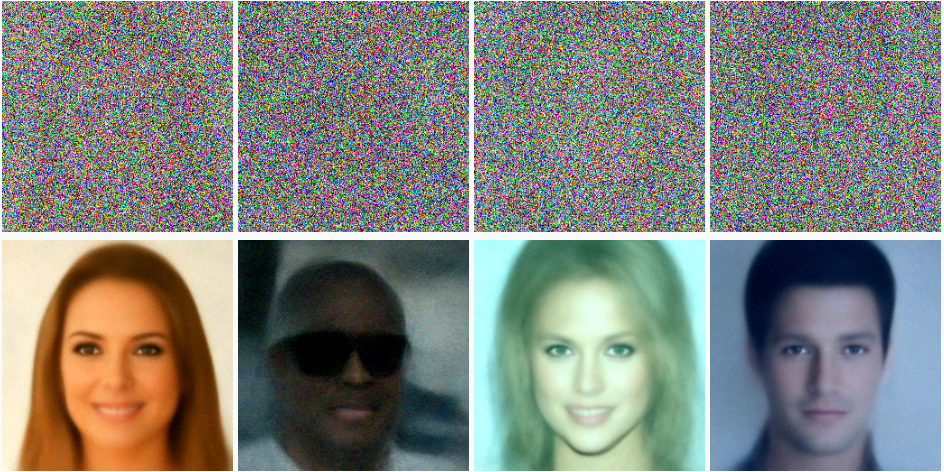
$\mathrm{t} = 190$

[2] 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][2])

x_t

$\widehat{x}_0(x_t)$



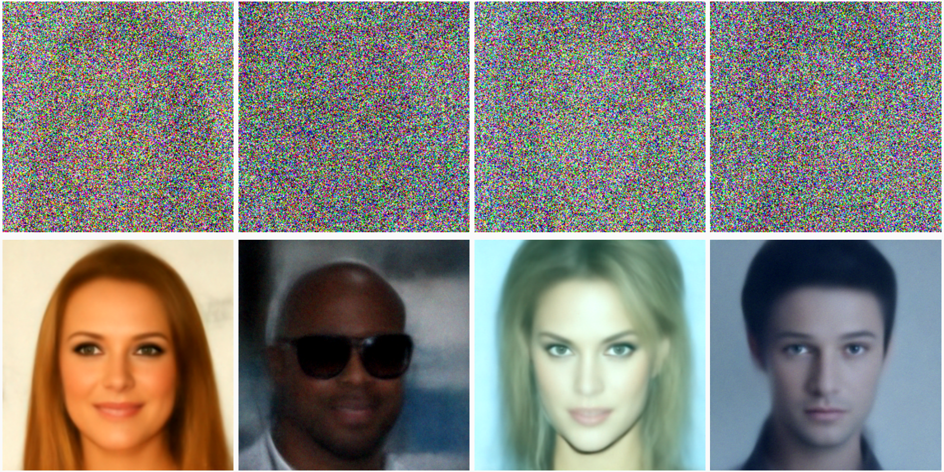
$t = 170$

[2] 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][2])

x_t

$\widehat{x}_0(x_t)$



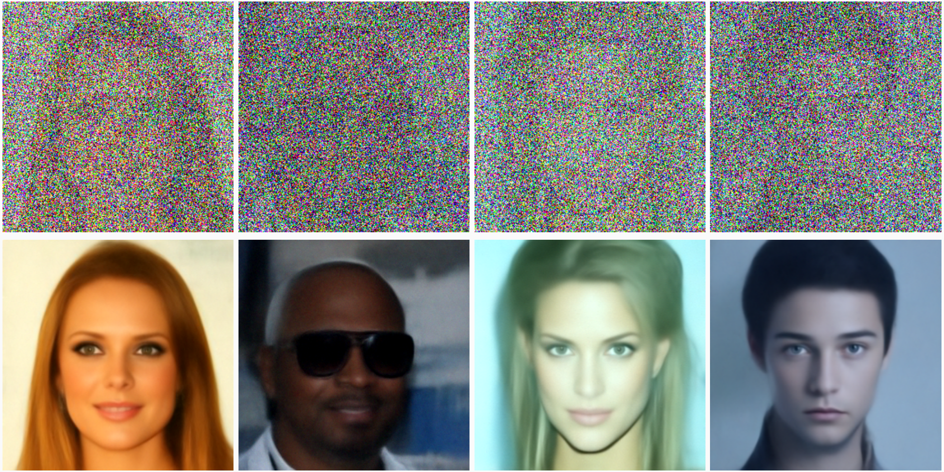
$t = 150$

[2] 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][2])

x_t

$\widehat{x}_0(x_t)$



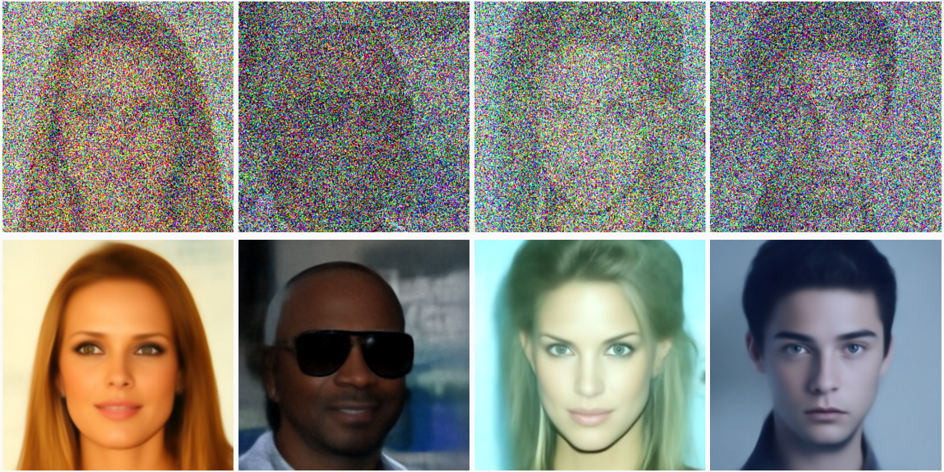
$t = 130$

[2] 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][2])

x_t

$\widehat{x}_0(x_t)$



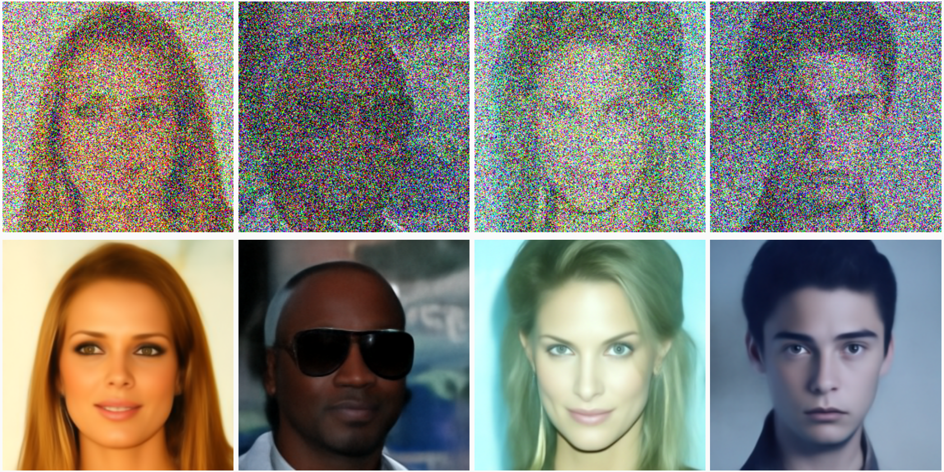
$\mathrm{t} = 110$

[2] 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][2])

x_t

$\widehat{x}_0(x_t)$

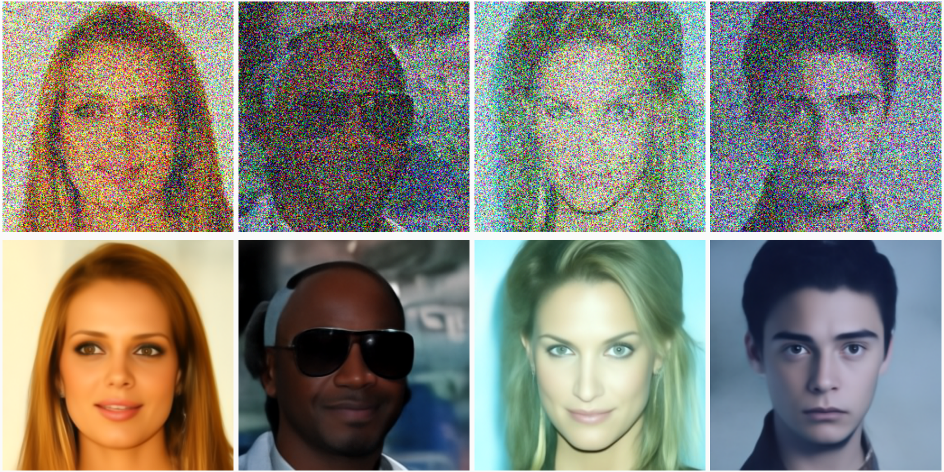


t = 90

[2] 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`

x_t

$\widehat{x}_0(x_t)$



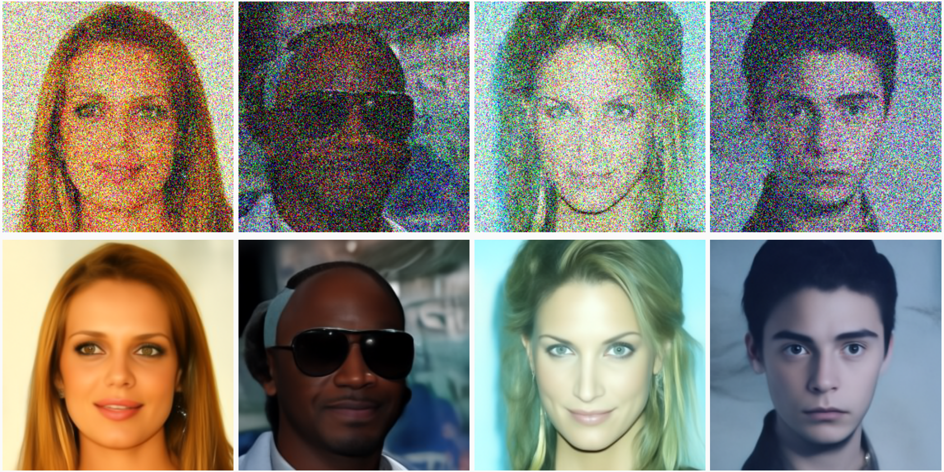
t = 70

[2] 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][2])

x_t

$\widehat{x}_0(x_t)$



t = 50

[2] 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][2])

x_t

$\widehat{x}_0(x_t)$

t = 30

[2] 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][2])

x_t

$\widehat{x}_0(x_t)$

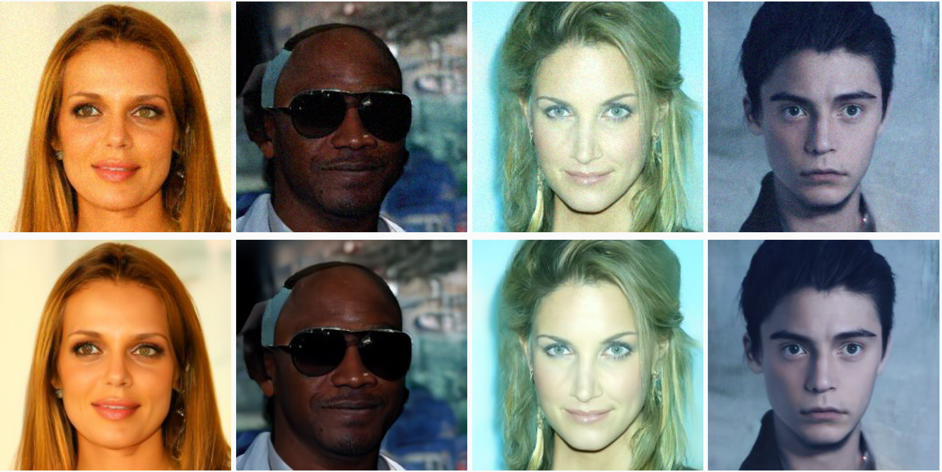
$t = 10$

[2] 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][2])

x_t

$\widehat{x}_0(x_t)$



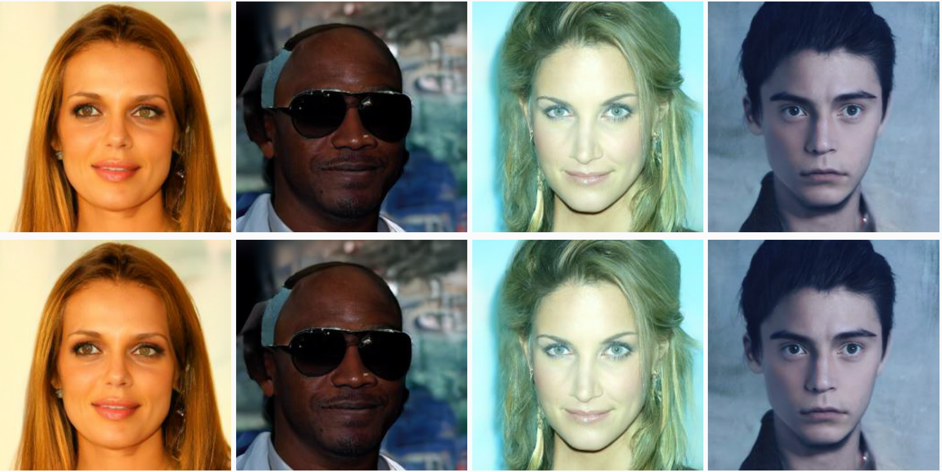
$t = 5$

[2] 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][2])

x_t

$\widehat{x}_0(x_t)$



$\mathrm{t} = 0$

[2] 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`

Probability-flow ODE

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

$$dy_t = -\beta_t \left[y_t + 2\nabla_y \log p_t(y_t)\right] dt + \sqrt{2\beta_t} d\overline{w}_t, \quad 0 \leqslant t \leqslant T, \quad y_T \sim p_T \tag{7}$$

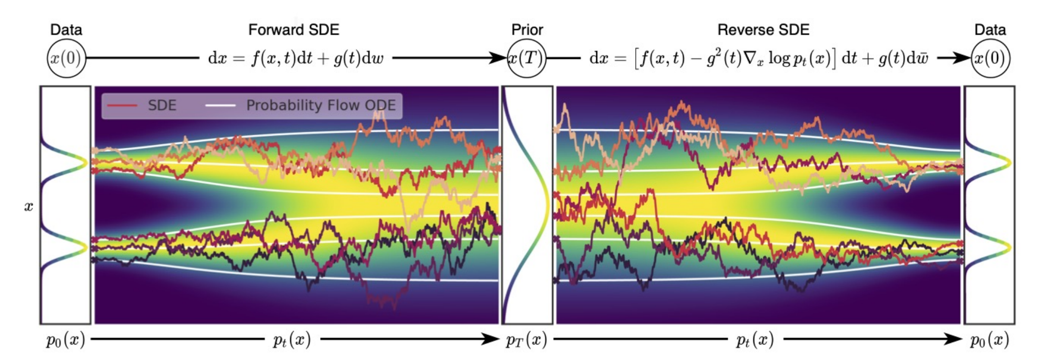
Probability-flow ODE

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

$$dy_t = -\beta_t \left[y_t + 2\nabla_y \log p_t(y_t)\right] dt + \sqrt{2\beta_t} d\overline{w}_t, \quad 0 \leqslant t \leqslant T, \quad y_T \sim p_T \tag{7}$$

are the same as those of this ODE

$$dy_t = -\beta_t \left[y_t + \nabla_y \log p_t(y_t)\right] dt, \qquad 0 \leqslant t \leqslant T, \quad y_T \sim p_T. \tag{8}$$



Study of the convergence

Sampling through diffusion models

$$\begin{aligned} &dy_t = -\beta_t \left[y_t + 2\nabla_y \log p_t(y_t) \right] dt + \sqrt{2\beta_t} d\overline{w}_t, \\ &\text{or} \qquad\qquad\qquad\qquad\qquad\qquad \text{where} \quad 0 \leqslant t \leqslant T, \quad y_T \sim p_T. \\ &dy_t = -\beta_t \left[y_t + \nabla_y \log p_t(y_t) \right] dt, \end{aligned} \tag{9}$$

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 $0 \leqslant t \leqslant T$, ~~$y_T \sim p_T$~~ $y_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. (9)

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**

Sampling through diffusion models

$$
\begin{aligned}
&dy_t = -\beta_t\left[y_t + 2\nabla_y \log p_t(y_t)\right]dt + \sqrt{2\beta_t}d\overline{w}_t, \\
&\text{or} \qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{where} \quad \underset{\varepsilon}{\cancel{0}} \leqslant t \leqslant T, \quad y_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}). \qquad (9)\\
&dy_t = -\beta_t\left[y_t + \nabla_y \log p_t(y_t)\right]dt,
\end{aligned}
$$

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

- 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**

Sampling through diffusion models

$$
\begin{aligned}
dy_t &= -\beta_t \left[y_t + 2\nabla_y \log p_t(y_t)\right] dt + \sqrt{2\beta_t} d\overline{w}_t, \\
&\text{or} \qquad\qquad \text{where} \quad \varepsilon \leqslant t \leqslant T, \quad y_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}). \\
dy_t &= -\beta_t \left[y_t + \nabla_y \log p_t(y_t)\right] dt,
\end{aligned} \tag{9}
$$

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**

Sampling through diffusion models

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

or where $\varepsilon \leqslant t \leqslant T, \quad y_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}).$ (9)

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

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

$$
\begin{aligned}
&dy_t = -\beta_t[y_t + 2s_\theta(t, y_t)]dt + \sqrt{2\beta_t}d\overline{w}_t, \\
&\text{or} \qquad\qquad\qquad\qquad\qquad\qquad \text{where} \quad \varepsilon \leqslant t \leqslant T, \quad y_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}). \qquad (9)\\
&dy_t = -\beta_t[y_t + s_\theta(t, y_t)]dt,
\end{aligned}
$$

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 s_θ to learn the score $\longrightarrow$ **score approximation error**

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 $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. ($\boldsymbol{\Sigma}$ is not necessarily invertible) In this case,

$$\nabla \log p_t(x) = -\boldsymbol{\Sigma}_t^{-1} x, \quad 0 < t \leqslant T \tag{10}$$

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

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,

$$\nabla \log p_t(x) = -\mathbf{\Sigma}_t^{-1} x, \quad 0 < t \leqslant T \tag{10}$$

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 $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. ($\boldsymbol{\Sigma}$ is not necessarily invertible)

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 4: Solution to the equations under Gaussian assumption

Under Gaussian assumption, the strong solution to SDE (4) can be written as:

$$y_t^{\mathsf{SDE}} = e^{-(B_T - B_t)} \mathbf{\Sigma}_t \mathbf{\Sigma}_T^{-1} y_T + \boldsymbol{\xi}_t, \quad 0 \leqslant t \leqslant T \tag{11}$$

Under Gaussian assumption, the solution to ODE (8) can be written as:

$$y_t^{\mathsf{ODE}} = \mathbf{\Sigma}_T^{-1/2} \mathbf{\Sigma}_t^{1/2} y_T, \quad 0 \leqslant t \leqslant T, \tag{12}$$

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

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

Proposition 5: Solution to the equations under Gaussian assumption

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

$$\boldsymbol{\Sigma}_t^{\mathsf{SDE}} = \mathrm{Cov}(y_t^{\mathsf{SDE}}) = \boldsymbol{\Sigma}_t + e^{-2(B_T - B_t)} \boldsymbol{\Sigma}_t^2 \boldsymbol{\Sigma}_T^{-2} \left[\mathrm{Cov}(y_T) - \boldsymbol{\Sigma}_T\right], \tag{11}$$

$$\boldsymbol{\Sigma}_t^{\mathsf{ODE}} = \mathrm{Cov}(y_t^{\mathsf{ODE}}) = \boldsymbol{\Sigma}_t \boldsymbol{\Sigma}_T^{-1}\,\mathrm{Cov}(y_T), \tag{12}$$

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

Proposition 6: Solution to the equations under Gaussian assumption

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

$$\boldsymbol{\Sigma}_t^{\mathsf{SDE}} = \mathrm{Cov}(y_t^{\mathsf{SDE}}) = \boldsymbol{\Sigma}_t + e^{-2(B_T - B_t)}\boldsymbol{\Sigma}_t^2 \boldsymbol{\Sigma}_T^{-2}\left[\mathrm{Cov}(y_T) - \boldsymbol{\Sigma}_T\right], \tag{11}$$

$$\boldsymbol{\Sigma}_t^{\mathsf{ODE}} = \mathrm{Cov}(y_t^{\mathsf{ODE}}) = \boldsymbol{\Sigma}_t \boldsymbol{\Sigma}_T^{-1}\,\mathrm{Cov}(y_T), \tag{12}$$

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

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

Explicit solution of the backward equations

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

Proposition 7: Solution to the equations under Gaussian assumption

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

$$\boldsymbol{\Sigma}_t^{\mathsf{SDE}} = \mathrm{Cov}(y_t^{\mathsf{SDE}}) = \boldsymbol{\Sigma}_t + e^{-2(B_T - B_t)}\boldsymbol{\Sigma}_t^2\boldsymbol{\Sigma}_T^{-2}\left[\mathrm{Cov}(y_T) - \boldsymbol{\Sigma}_T\right], \tag{11}$$

$$\boldsymbol{\Sigma}_t^{\mathsf{ODE}} = \mathrm{Cov}(y_t^{\mathsf{ODE}}) = \boldsymbol{\Sigma}_t\boldsymbol{\Sigma}_T^{-1}\,\mathrm{Cov}(y_T), \tag{12}$$

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

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

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

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

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

Link between ODE and OT

Under Gaussian assumption, the solution to ODE (8) can be written as:

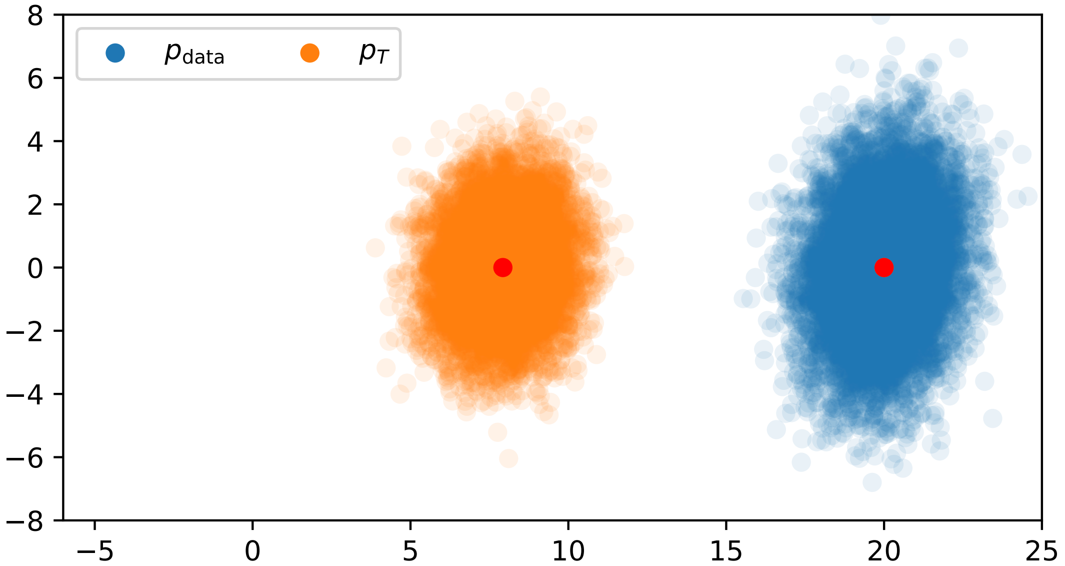
$$y_t^{\text{ODE}} = \boldsymbol{\Sigma}_T^{-1/2} \boldsymbol{\Sigma}_t^{1/2} y_T, \quad 0 \leqslant t \leqslant T, \tag{13}$$

- $y \mapsto \boldsymbol{\Sigma}_T^{-1/2} \boldsymbol{\Sigma}_t^{1/2} y$ is the transport map between p_T and p_t.
- False in general:, see [Lavenant and Santambrogio 2022][3]
- However, used in [Khrulkov et al. 2023][4]

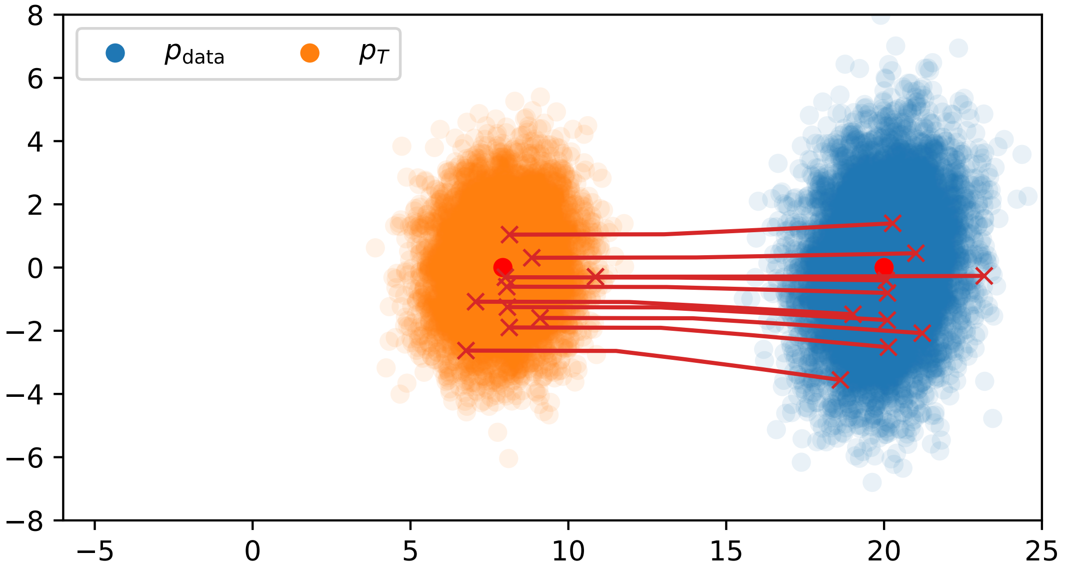
[3] Hugo Lavenant and Filippo Santambrogio (2022). "The flow map of the Fokker–Planck equation does not provide optimal transport". In: *Applied Mathematics Letters* 133, p. 108225. ISSN: 0893-9659. DOI: https://doi.org/10.1016/j.aml.2022.108225. URL: https://www.sciencedirect.com/science/article/pii/S089396592200180X

[4] 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

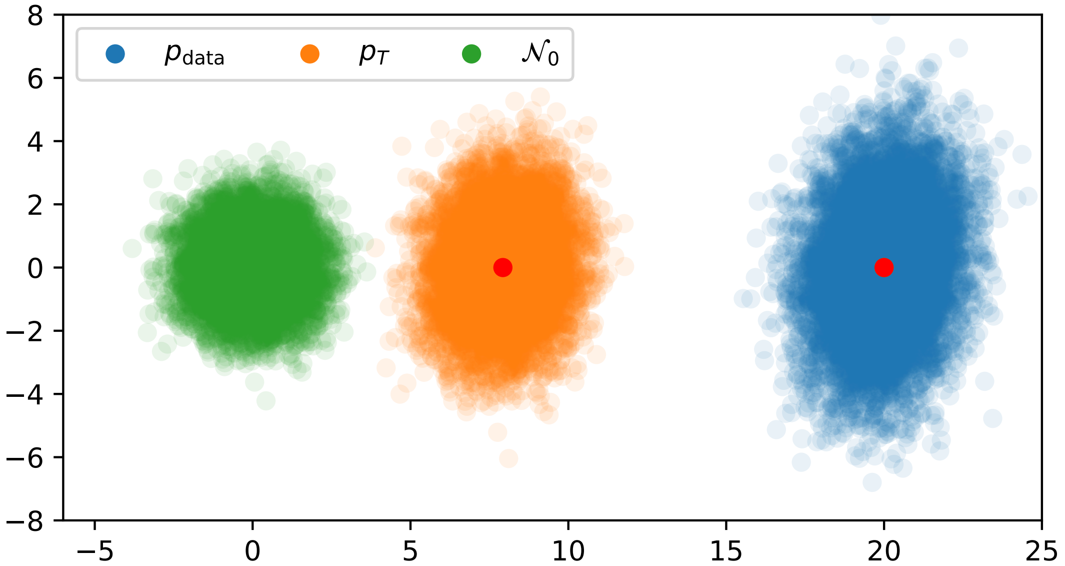
Initialization error: Focus on the ODE



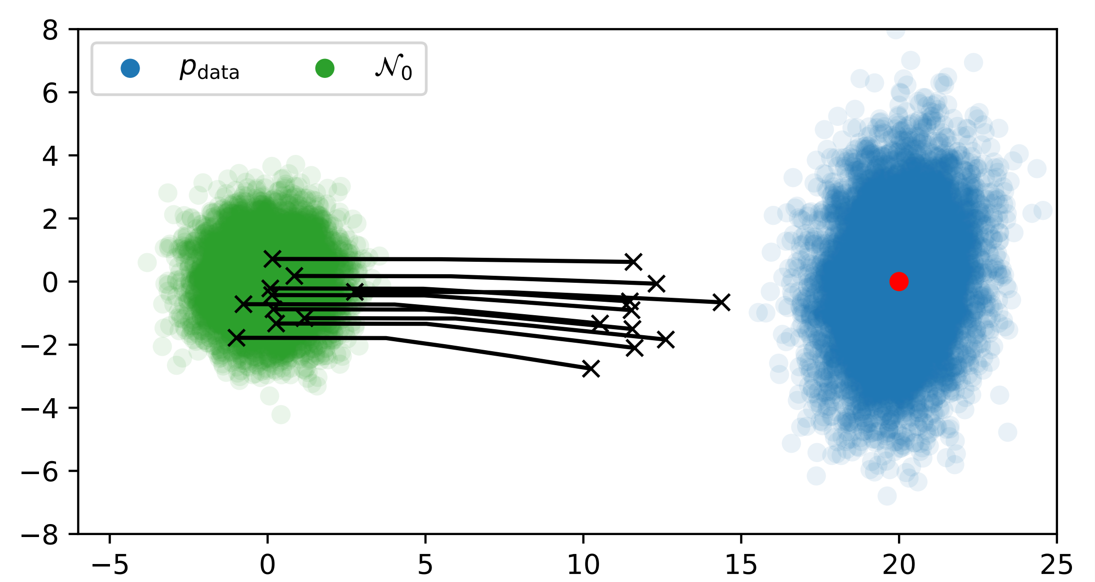
Initialization error: Focus on the ODE



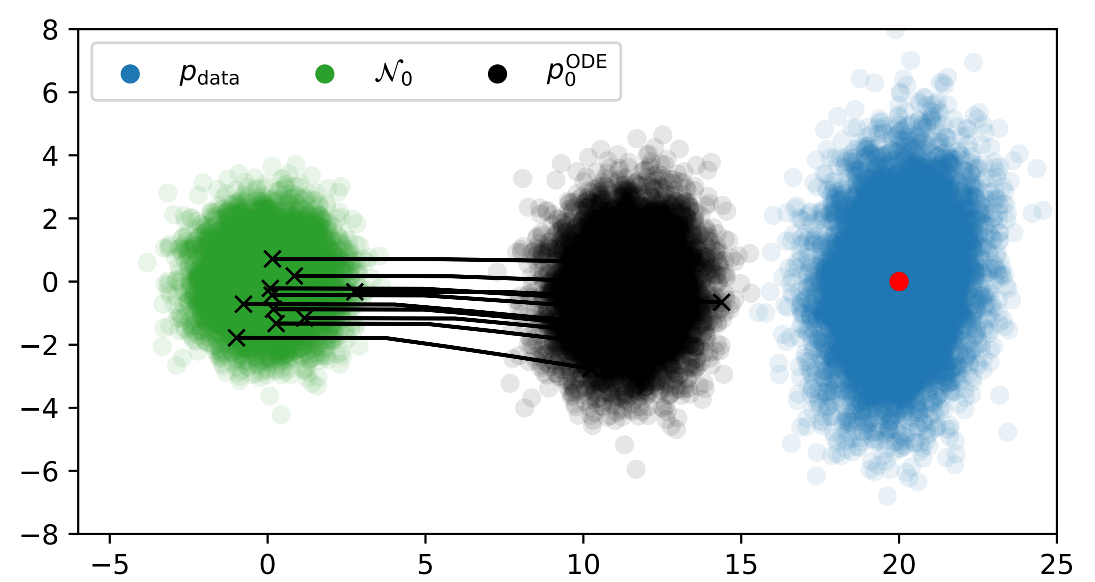
Initialization error: Focus on the ODE



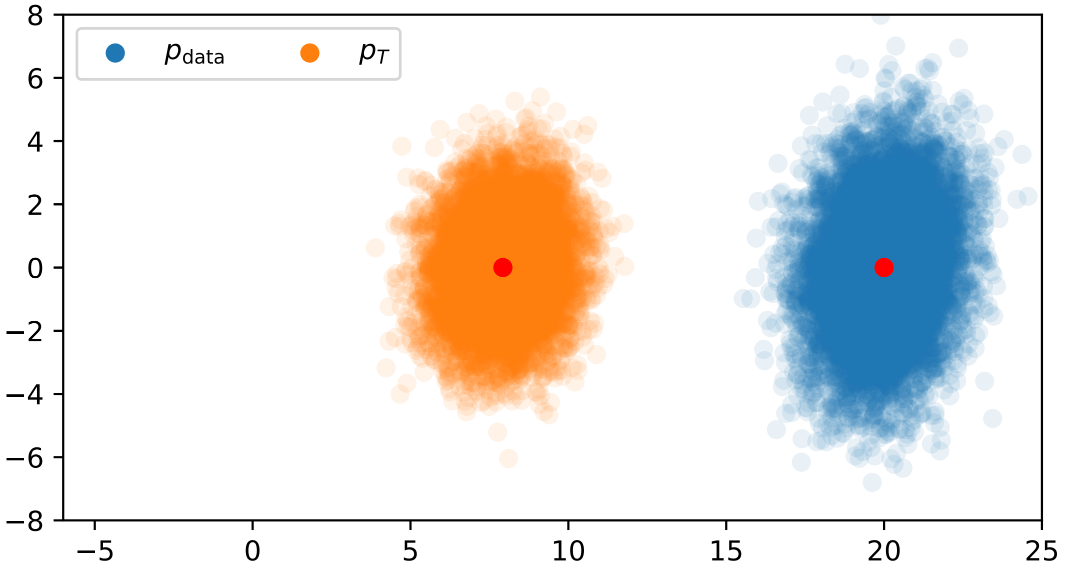
Initialization error: Focus on the ODE



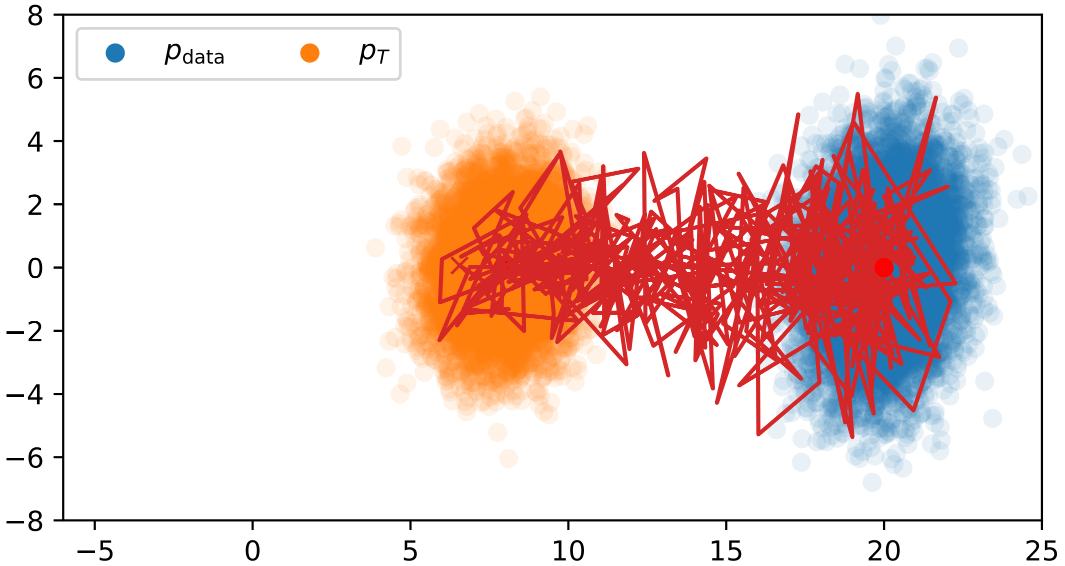
Initialization error: Focus on the ODE

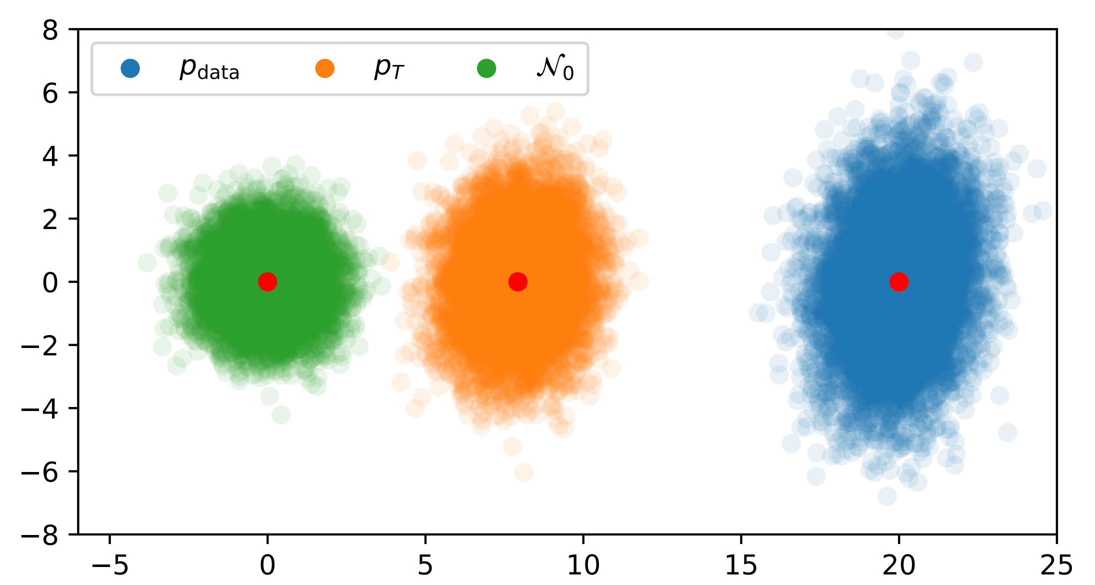


Initialization error: Focus on the SDE



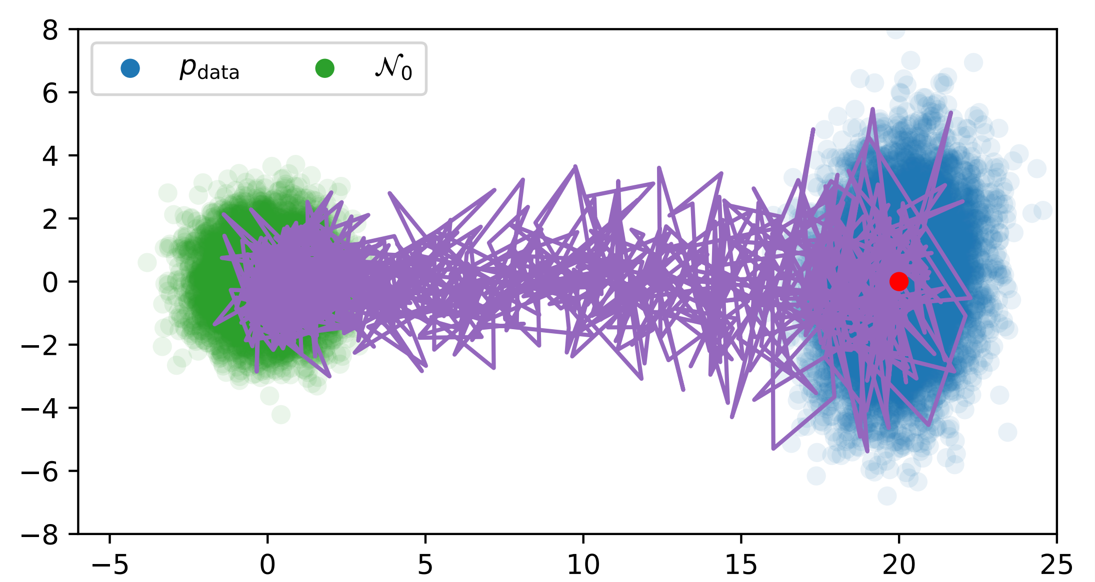
Initialization error: Focus on the SDE



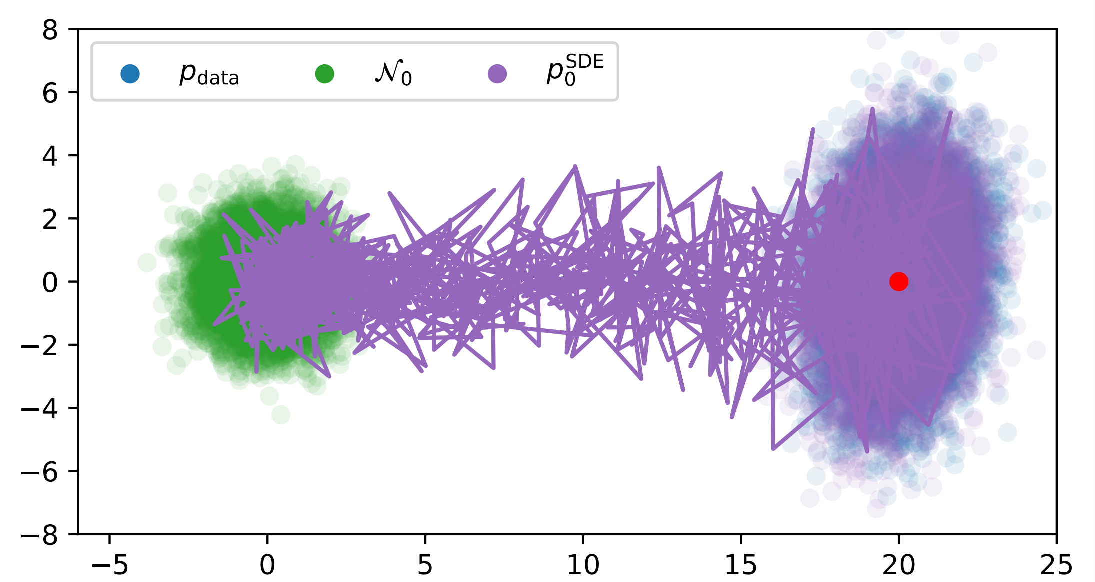


p_{data}
p_T
$\mathcal{N}_0$
8
6
4
2
0
−2
−4
−6
−8
−5
0
5
10
15
20
25

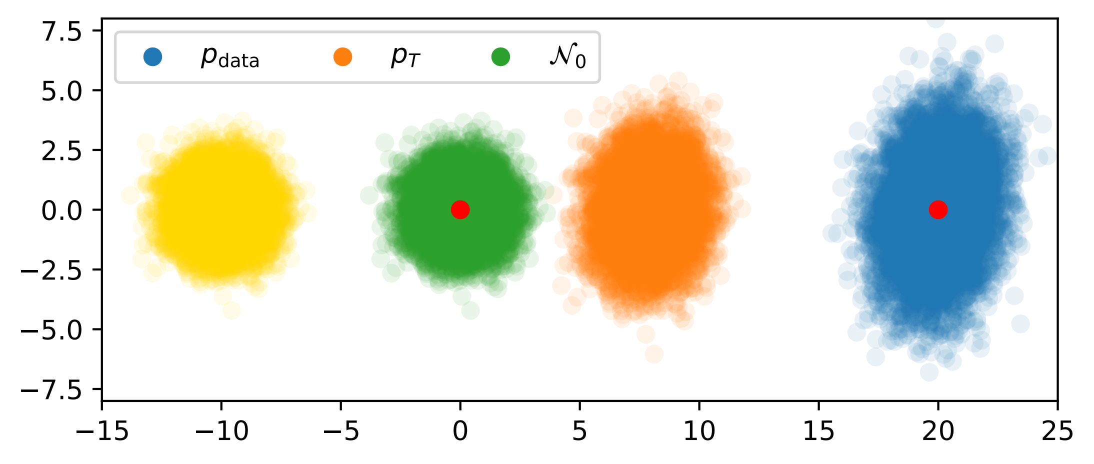
Initialization error: Focus on the SDE



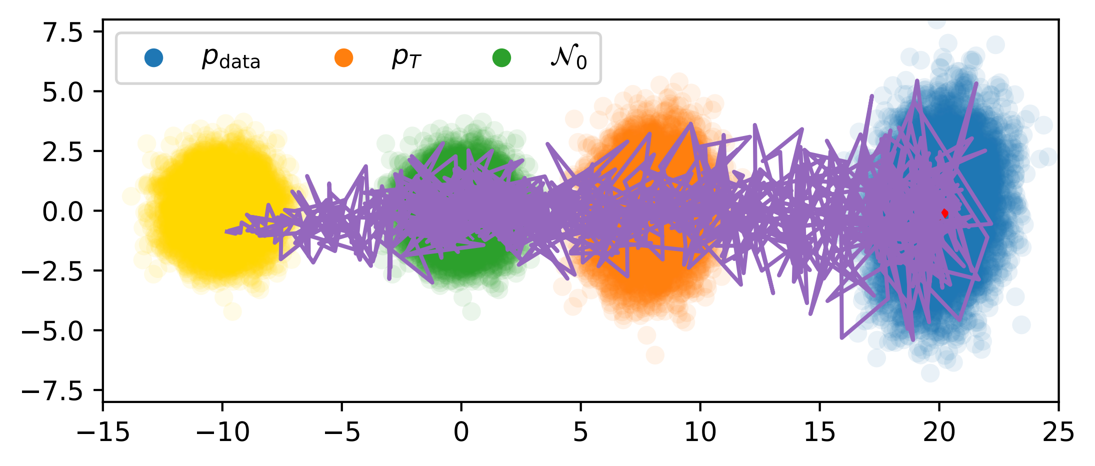
Initialization error: Focus on the SDE



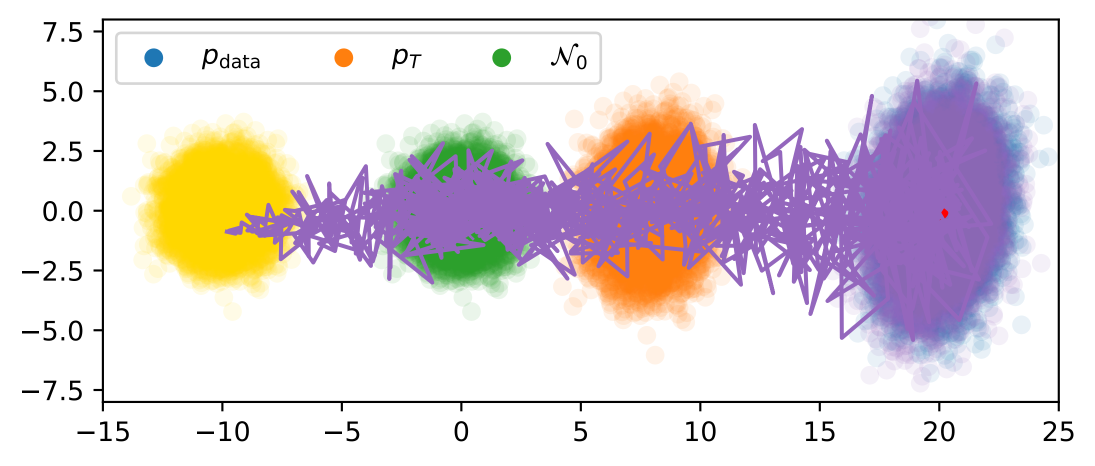
Initialization error: Focus on the SDE



Initialization error: Focus on the SDE



Initialization error: Focus on the SDE



Initialization error

Under Gaussian assumption, the strong solution to SDE (4) can be written as:

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

with $\mathrm{Cov}(\boldsymbol{\xi}_t) = \boldsymbol{\Sigma}_t - e^{-2(B_T - B_t)} \boldsymbol{\Sigma}_t^2 \boldsymbol{\Sigma}_T^{-1}$.

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

Initialization error

Under Gaussian assumption, the strong solution to SDE (4) can be written as:

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

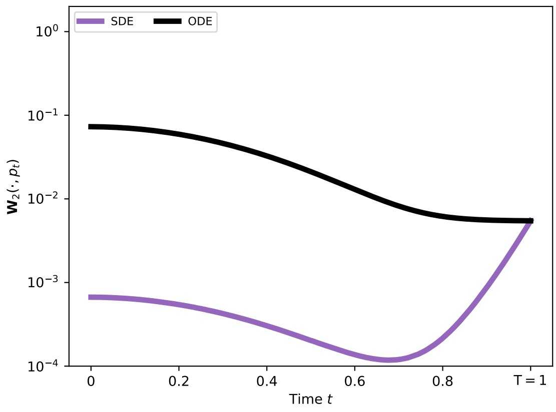
with $\mathrm{Cov}(\boldsymbol{\xi}_t) = \boldsymbol{\Sigma}_t - e^{-2(B_T - B_t)} \boldsymbol{\Sigma}_t^2 \boldsymbol{\Sigma}_T^{-1}$.

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

Proposition 8: Marginals of the generative processes under Gaussian assumption

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

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



Discretization error

Discretization processes

The EM discretized process is a Gaussian process:

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

Discretization processes

The EM discretized process is a Gaussian process:

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

$$\boldsymbol{\Sigma}_T^{\Delta,\text{EM}} = \boldsymbol{I} \tag{17}$$

$$\boldsymbol{\Sigma}_{T-1}^{\Delta,\text{EM}} = \left(\boldsymbol{I} + \Delta_t \beta_{T-t_0} \left(\boldsymbol{I} - 2\boldsymbol{\Sigma}_{T-t_0}^{-1} \right) \right)^2 \boldsymbol{\Sigma}_T^{\Delta,\text{EM}} + 2\Delta_t \beta_{T-t_0} \boldsymbol{I} \tag{18}$$

Discretization processes

The EM discretized process is a Gaussian process:

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

$$\boldsymbol{\Sigma}_T^{\Delta,\text{EM}} = \boldsymbol{I} \tag{17}$$

$$\boldsymbol{\Sigma}_{T-1}^{\Delta,\text{EM}} = \left(\boldsymbol{I} + \Delta_t \beta_{T-t_0} \left(\boldsymbol{I} - 2\boldsymbol{\Sigma}_{T-t_0}^{-1} \right) \right)^2 \boldsymbol{\Sigma}_T^{\Delta,\text{EM}} + 2\Delta_t \beta_{T-t_0} \boldsymbol{I} \tag{18}$$

Denoting $(\boldsymbol{v}_i)_i$ the eigenvectors of $\boldsymbol{\Sigma}$, let $1 \leqslant i \leqslant d$,

$$\boldsymbol{\Sigma}_1^{\Delta,\text{EM}} \boldsymbol{v}_i = \left[\left(1 + \Delta_t \beta_{T-t_0} \left(\boldsymbol{I} - \frac{2}{\lambda_{i,T-t_0}} \right) \right)^2 \lambda_{i,T}^{\Delta,\text{EM}} + 2\Delta_t \beta_{T-t_0} \right] \boldsymbol{v}_i \tag{19}$$

Discretization processes

The EM discretized process is a Gaussian process:

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

$$\boldsymbol{\Sigma}_T^{\Delta,\text{EM}} = \boldsymbol{I} \tag{17}$$

$$\boldsymbol{\Sigma}_{T-1}^{\Delta,\text{EM}} = \left(\boldsymbol{I} + \Delta_t \beta_{T-t_0} \left(\boldsymbol{I} - 2\boldsymbol{\Sigma}_{T-t_0}^{-1} \right) \right)^2 \boldsymbol{\Sigma}_T^{\Delta,\text{EM}} + 2\Delta_t \beta_{T-t_0} \boldsymbol{I} \tag{18}$$

Denoting $(\boldsymbol{v}_i)_i$ the eigenvectors of $\boldsymbol{\Sigma}$, let $1 \leqslant i \leqslant d$,

$$\boldsymbol{\Sigma}_1^{\Delta,\text{EM}} \boldsymbol{v}_i = \left[\left(1 + \Delta_t \beta_{T-t_0} \left(\boldsymbol{I} - \frac{2}{\lambda_{i,T-t_0}} \right) \right)^2 \lambda_{i,T}^{\Delta,\text{EM}} + 2\Delta_t \beta_{T-t_0} \right] \boldsymbol{v}_i \tag{19}$$

$$\lambda_{i,1}^{\Delta,\text{EM}} = \left(1 + \Delta_t \beta_{T-t_0} \left(\boldsymbol{I} - \frac{2}{\lambda_{i,T-t_0}} \right) \right)^2 \lambda_{i,T}^{\Delta,\text{EM}} + 2\Delta_t \beta_{T-t_0}, \quad 1 \leqslant i \leqslant d. \tag{20}$$

The EM discretized process is a Gaussian process:

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

For two centered Gaussians $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_1)$ and $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_2)$ such that $\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2$ are simultaneously diagonalizable with respective eigenvalues $(\lambda_{i,1})_{1 \leqslant i \leqslant d}, (\lambda_{i,2})_{1 \leqslant i \leqslant d}$,

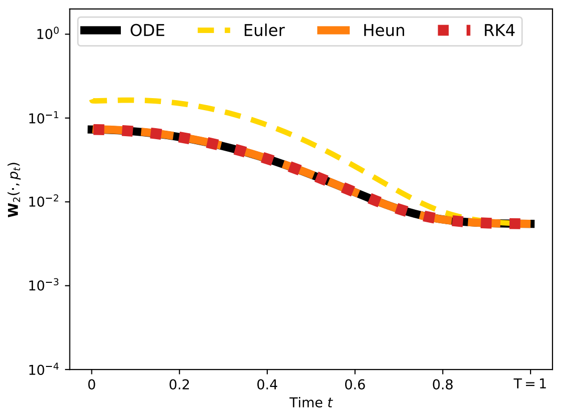
$$\mathbf{W}_2(\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_1), \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_2))^2 = \sum_{1 \leqslant i \leqslant d} (\sqrt{\lambda_{i,1}} - \sqrt{\lambda_{i,2}})^2 \tag{17}$$

Conclusion: We are able to compute exact 2-Wassertein distance between the forward process and the discretized process.

ODE discretization

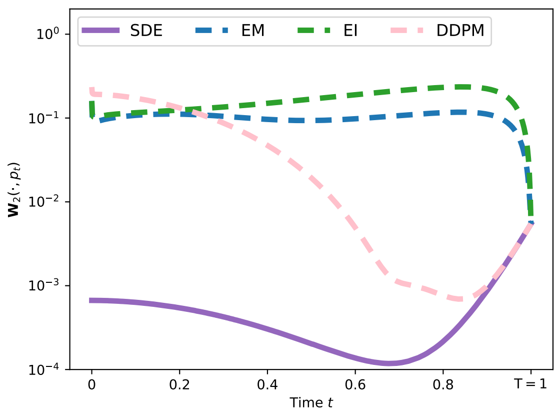
Several schemes to discretize the ODE:

- The Euler's scheme
- The Heun's scheme
- The RK4 scheme
- . . .



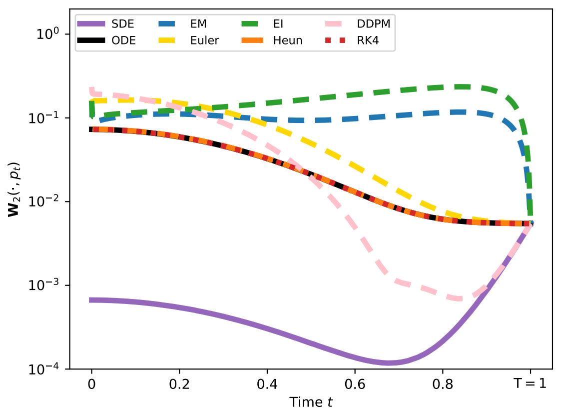
Several schemes to discretize the SDE:

- The Denoising Diffusion Probabilistic Model (DDPM) scheme
- The Euler Maruyama (EM) scheme
- The Exponential Integrator (EI) scheme
- ...



Discretization + Initialization errors

- The ODE discretization wins.
- The SDE dynamic is too fast.



Score approximation error

The Asymptotic Discrete Spot Noise (ADSN) model [Galerne, Gousseau, and Morel 2011] [5]

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

$\boldsymbol{u}$

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

[5] 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] [5]

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

$\boldsymbol{u}$

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

[5] 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] [5]

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

$\boldsymbol{u}$

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

[5] 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] [5]

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

$\boldsymbol{u}$

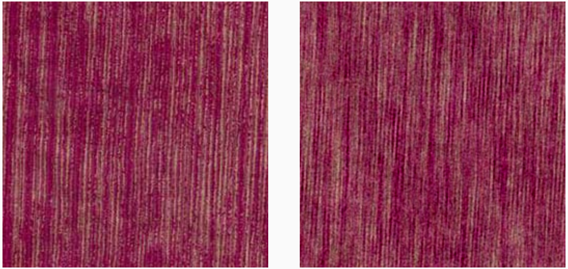
Samples of $\mathrm{ADSN}(\boldsymbol{u})$

[5] 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] [5]

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

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



[5] 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

$\boldsymbol{u}$

Gaussian samples

p_θ^{EM}

p_θ^{Heun}

Discretization + Initialization + Score approximation errors

$\boldsymbol{u}$

Gaussian samples

p_θ^{EM}

p_θ^{Heun}

Discretization + Initialization + Score approximation errors

$\boldsymbol{u}$

Gaussian samples

p_θ^{EM}

p_θ^{Heun}

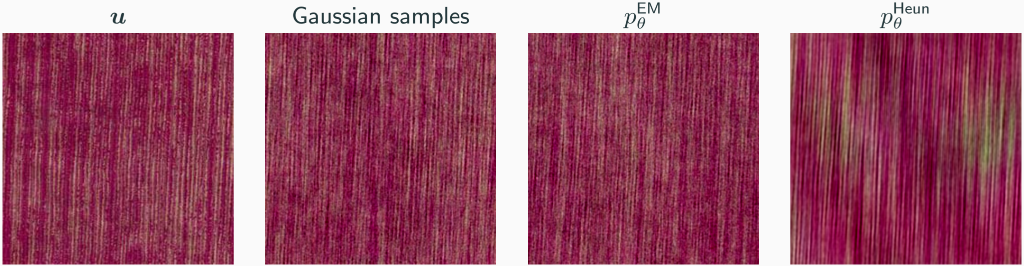
$\boldsymbol{u}$

Gaussian samples

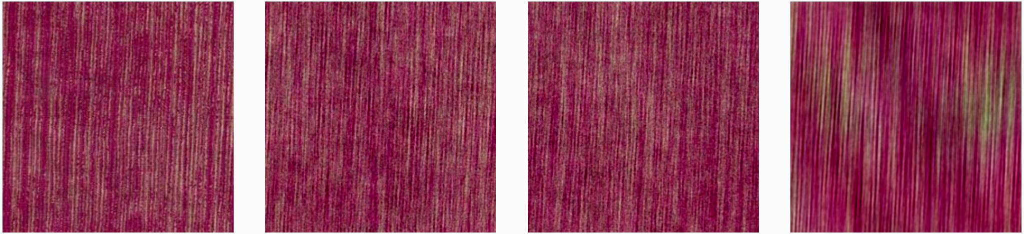
p_θ^{EM}

p_θ^{Heun}

Discretization + Initialization + Score approximation errors



Discretization + Initialization + Score approximation errors



	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	0.0891	15.6	1.02
Heun	3.73	3.7323	0.0447	56.7	19.4

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.
  (in progress) → 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.
  (in progress) → 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
- Extension to conditional diffusion models ?
  (in progress) → Joël Greffier et al. (2024). "Photon-counting CT systems: A technical review of current clinical possibilities". In: *Diagnostic and Interventional Imaging*. ISSN: 2211-5684. DOI: https://doi.org/10.1016/j.diii.2024.09.002. URL: https://www.sciencedirect.com/science/article/pii/S2211568424001955

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 et al. (2022). "Improving Diffusion Models for Inverse Problems using Manifold Constraints". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

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.

Greffier, Joël et al. (2024). "Photon-counting CT systems: A technical review of current clinical possibilities". In: *Diagnostic and Interventional Imaging*. ISSN: 2211-5684. DOI: https://doi.org/10.1016/j.diii.2024.09.002. URL: https://www.sciencedirect.com/science/article/pii/S2211568424001955.

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.

SDE schemes	Euler-Maruyama (EM)	$\begin{cases} \tilde{\boldsymbol{y}}_0^{\Delta,\text{EM}} & \sim \mathcal{N}_0 \\ \tilde{\boldsymbol{y}}_{k+1}^{\Delta,\text{EM}} & = \tilde{\boldsymbol{y}}_k^{\Delta,\text{EM}} + \Delta_t \tilde{f}(t_k, \tilde{\boldsymbol{y}}_k^{\Delta,\text{EM}}) + \sqrt{2\Delta_t \beta_{T-t_k}} z_k, \ z_k \sim \mathcal{N}_0 \end{cases}$ where $\tilde{f}(t, y) = \beta_{T-t} y - 2\beta_{T-t} \boldsymbol{\Sigma}_{T-t}^{-1} y$	(18)
	Exponential integrator (EI)	$\begin{cases} \tilde{\boldsymbol{y}}_0^{\Delta,\text{EI}} & \sim \mathcal{N}_0 \\ \tilde{\boldsymbol{y}}_{k+1}^{\Delta,\text{EI}} & = \tilde{\boldsymbol{y}}_k^{\Delta,\text{EI}} + \gamma_{1,k} \left(\tilde{\boldsymbol{y}}_k^{\Delta,\text{EI}} - 2\boldsymbol{\Sigma}_{T-t_k}^{-1} \tilde{\boldsymbol{y}}_k^{\Delta,\text{EI}} \right) + \sqrt{2\gamma_{2,k}} z_k, \ z_k \sim \mathcal{N}_0 \end{cases}$ where $\gamma_{1,k} = \exp(B_{T-t_k} - B_{T-t_{k+1}}) - 1$ and $\gamma_{2,k} = \frac{1}{2}(\exp(2B_{T-t_k} - 2B_{T-t_{k+1}}) - 1)$	(19)
	Denoising Diffusion Probabilistic Model (DDPM)	$\begin{cases} \tilde{\boldsymbol{y}}_0^{\Delta,\text{DDPM}} & \sim \mathcal{N}_0 \\ \tilde{\boldsymbol{y}}_{k+1}^{\Delta,\text{DDPM}} & = \frac{1}{\sqrt{1-\beta_k^{\text{DDPM}}}} \left(\tilde{\boldsymbol{y}}_k^{\Delta,\text{DDPM}} - \beta_k^{\text{DDPM}} \boldsymbol{\Sigma}_{T-t_k}^{-1} \tilde{\boldsymbol{y}}_k^{\Delta,\text{DDPM}} \right) + \sqrt{\beta_k^{\text{DDPM}}} z_k, \ z_k \sim \mathcal{N}_0 \end{cases}$ where $\beta_k^{\text{DDPM}} = 2\Delta_t \beta(t_k)$	(20)
ODE schemes	Explicit Euler	$\begin{cases} \hat{\boldsymbol{y}}_0^{\Delta,\text{Euler}} & \sim \mathcal{N}_0 \\ \hat{\boldsymbol{y}}_{k+1}^{\Delta,\text{Euler}} & = \hat{\boldsymbol{y}}_k^{\Delta,\text{Euler}} + \Delta_t \hat{f}(t_k, \hat{\boldsymbol{y}}_k^{\Delta,\text{Euler}}) \end{cases}$ where $\hat{f}(t, y) = \beta_{T-t} y - \beta_{T-t} \boldsymbol{\Sigma}_{T-t}^{-1} y$	(21)
	Heun's method	$\begin{cases} \hat{\boldsymbol{y}}_0^{\Delta,\text{Heun}} & \sim \mathcal{N}_0 \\ \hat{\boldsymbol{y}}_{k+1/2}^{\Delta,\text{Heun}} & = \hat{\boldsymbol{y}}_k^{\Delta,\text{Heun}} + \Delta_t \hat{f}(t_k, \hat{\boldsymbol{y}}_k^{\Delta,\text{Heun}}) \\ \hat{\boldsymbol{y}}_{k+1}^{\Delta,\text{Heun}} & = \hat{\boldsymbol{y}}_k^{\Delta,\text{Heun}} + \frac{\Delta_t}{2} \left(\hat{f}(t_k, \hat{\boldsymbol{y}}_k^{\Delta,\text{Heun}}) + \hat{f}(t_{k+1}, \hat{\boldsymbol{y}}_{k+1/2}^{\Delta,\text{Heun}}) \right) \end{cases}$ where $\hat{f}(t, y) = \beta_{T-t} y - \beta_{T-t} \boldsymbol{\Sigma}_{T-t}^{-1} y$	(22)
	Runge-	$\begin{cases} \hat{\boldsymbol{y}}_0^{\Delta,\text{RK4}} & \sim \mathcal{N}_0 \\ K_1 & = \hat{f}(t_k, \hat{\boldsymbol{y}}_k^{\Delta,\text{RK4}}) \\ K_2 & = \hat{f}(t_{k+1/2}, \hat{\boldsymbol{y}}_k^{\Delta,\text{RK4}} + \frac{\Delta_t}{2} K_1) \\ K_3 & = \hat{f}(t \ldots \end{cases}$	(23)

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[6],Lugmayr et al. 2022[7],Chung et al. 2022[8],Choi et al. 2021[9]] uses the Bayes formula

$$\nabla_x \log p_t(x_t \mid \boldsymbol{v}) = \nabla_x \log p_t(\boldsymbol{v} \mid x_t) + \nabla_x \log p_t(x_t). \tag{24}$$

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

[6] 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

[7] 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

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

[9] 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