Introduction to diffusion models for image generation and inpainting

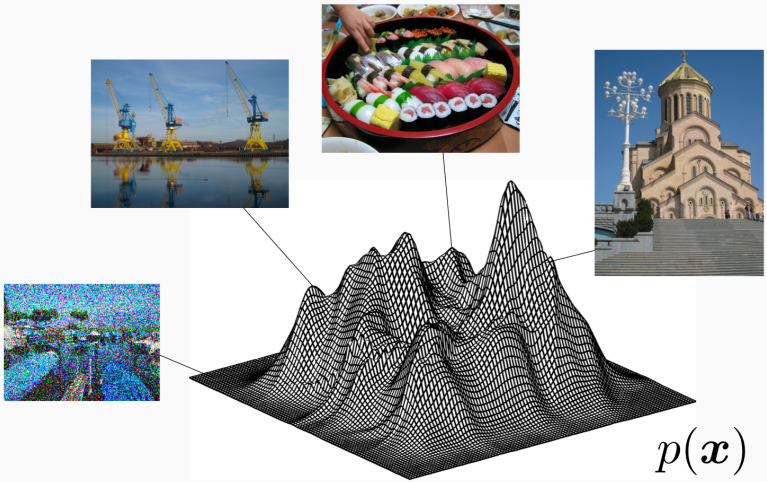
Émile Pierret, supervised by Bruno Galerne

July, 5^{th}

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

Introduction on generative models

What is a generative model ?



22/09 : Variational Auto-Encoder (VAE):

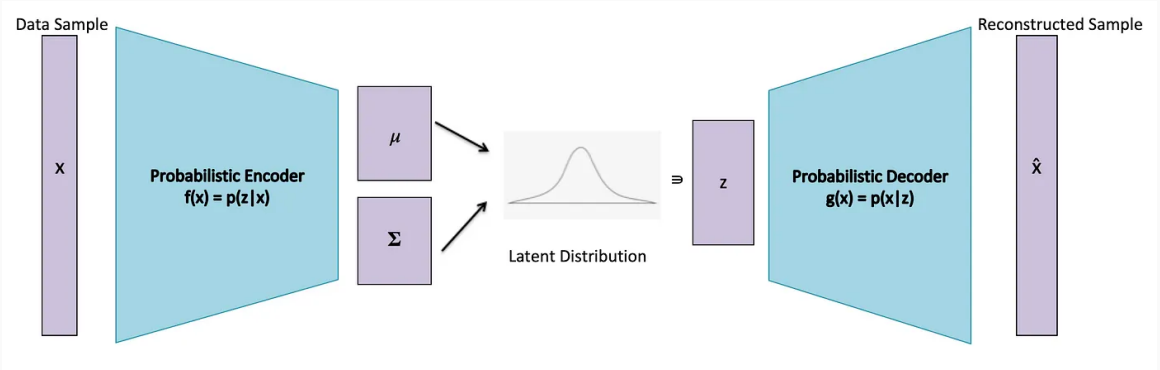


Image extracted from https://medium.com/@elzettevanrensburg/generating-the-intuition-behind-variational-auto-encoders-vaes-c7d2f8631a87

25/10: Generative Aversarial Netowrk (GAN):

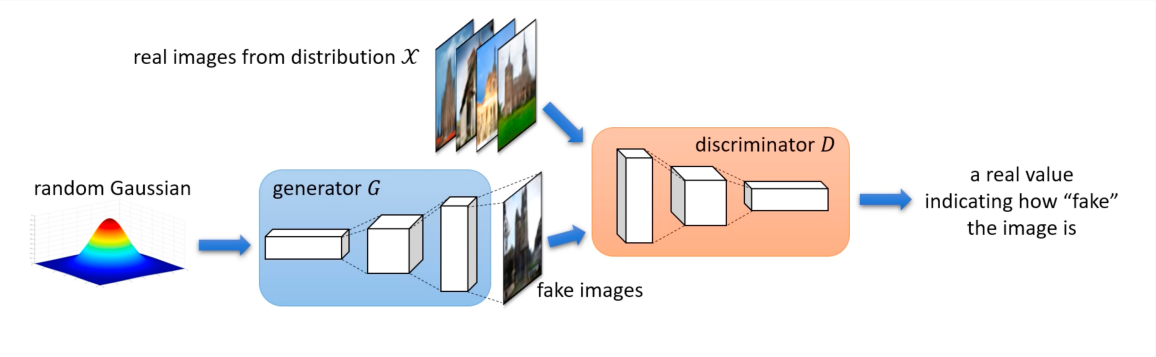
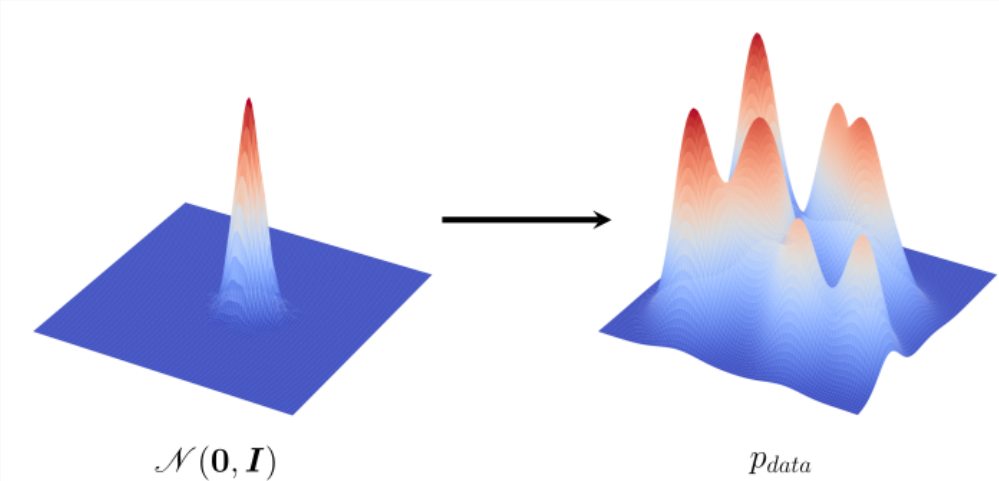


Image extracted from https://www.microsoft.com/en-us/research/blog/how-can-generative-adversarial-networks-learn-real-life-distributions-easily/

Main idea



To the diffusion models

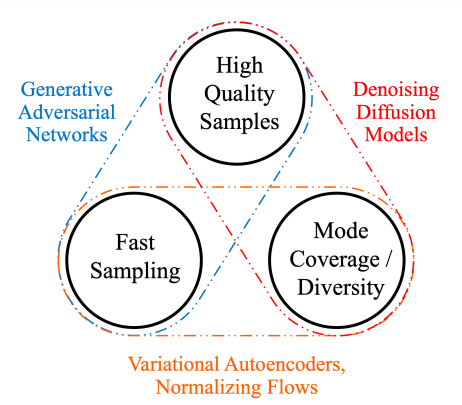


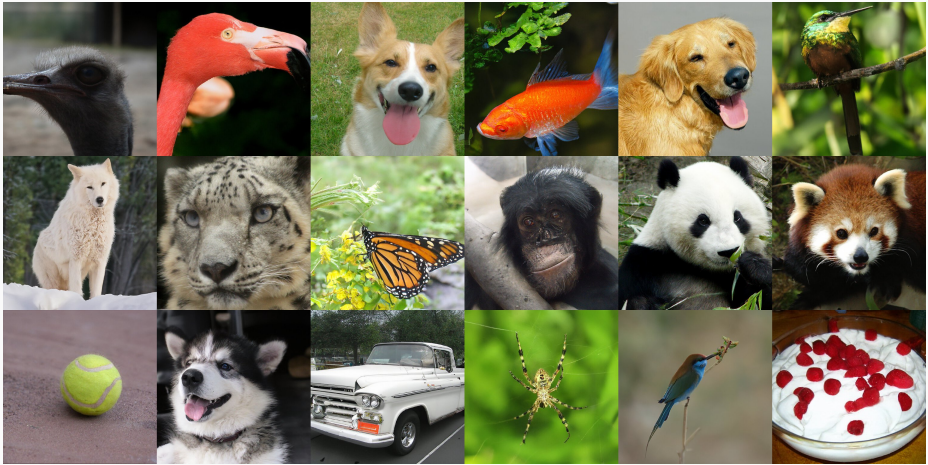
Image extrated from [Xiao et al., 2022][1]

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

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

Presentation of the diffusion models

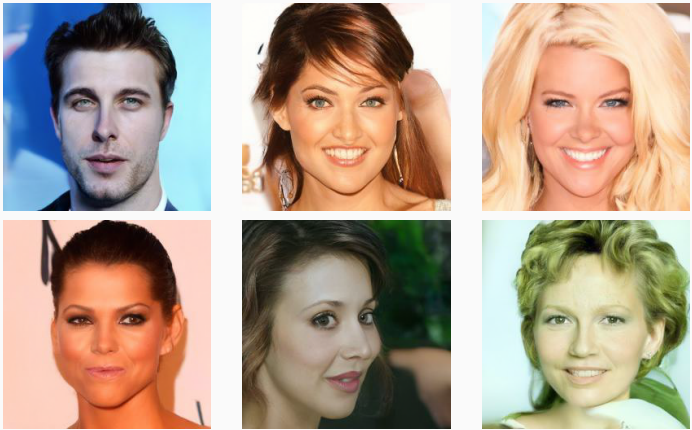
Samples from diffusion models



Images extracted from [Dhariwal and Nichol, 2021][2] from a model trained on ImageNet.

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

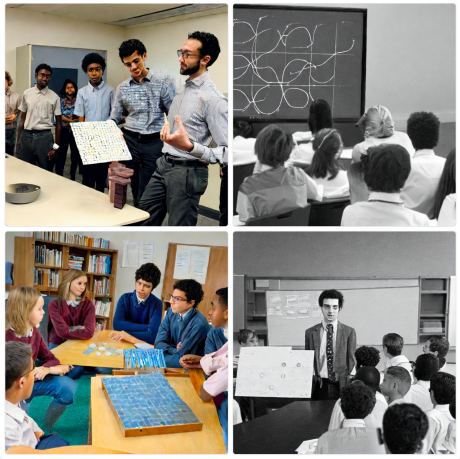
Home-made with the code of Lugmayr et al., 2022[3]



NB: 100s for 10 samples.

[3] Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., & Van Gool, L. (2022). RePaint: Inpainting using denoising diffusion probabilistic models. *RePaint*

Stable diffusion[4]



A young mathematician presenting diffusion models to his colleagues

[4] https://huggingface.co/spaces/stabilityai/stable-diffusion

Stable diffusion[5]

A young mathematician sadly discovers that he should study stochastic differential equations for his thesis.

[5] https://huggingface.co/spaces/stabilityai/stable-diffusion

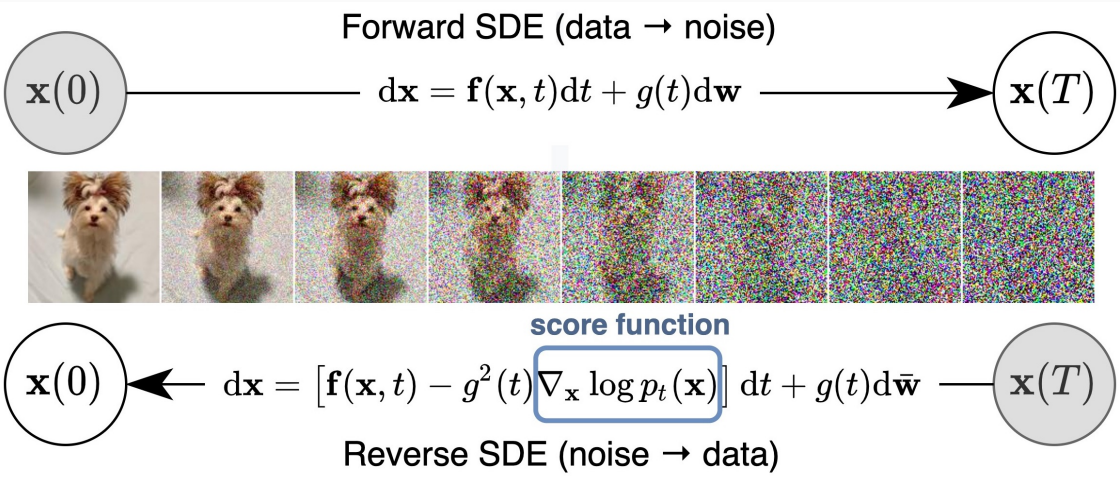
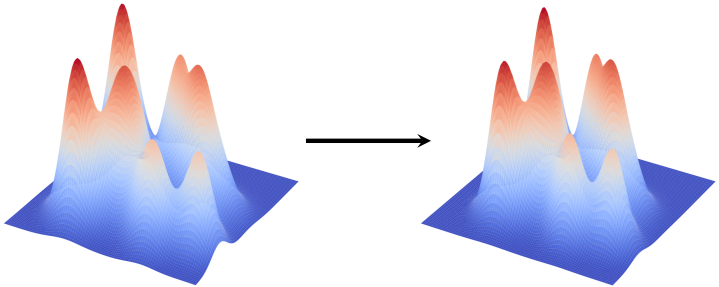


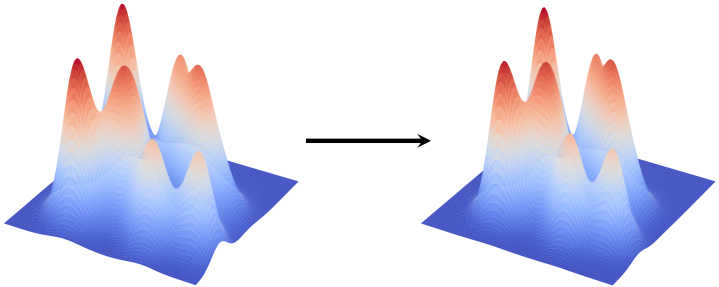
Image extracted from [Y. Song et al., 2023]

The forward process



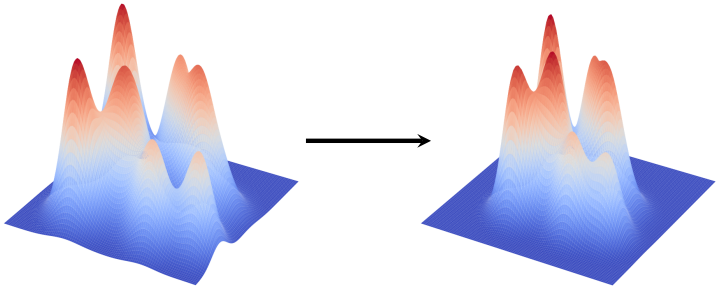
p_{data} p_1

The forward process

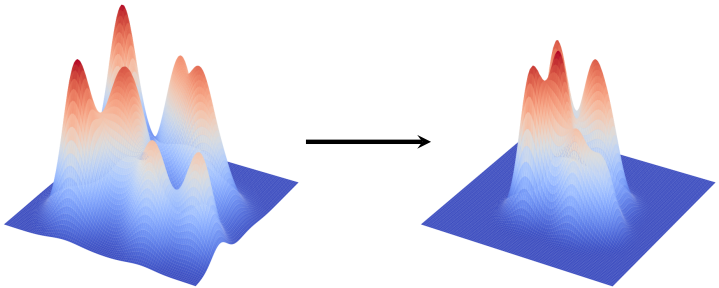


p_{data} p_2

The forward process

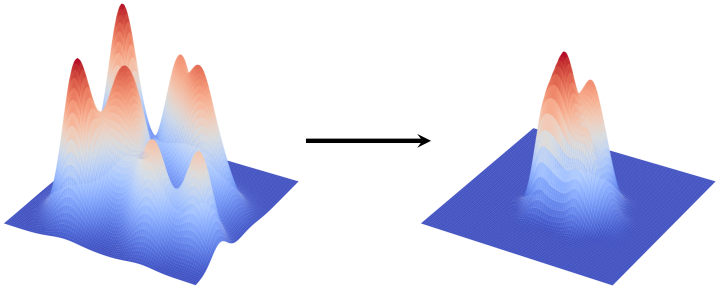


The forward process



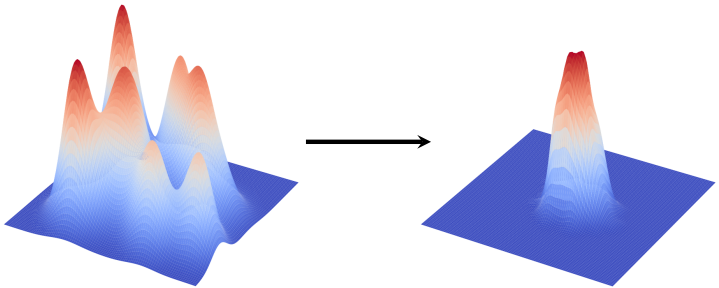
p_{data} p_4

The forward process



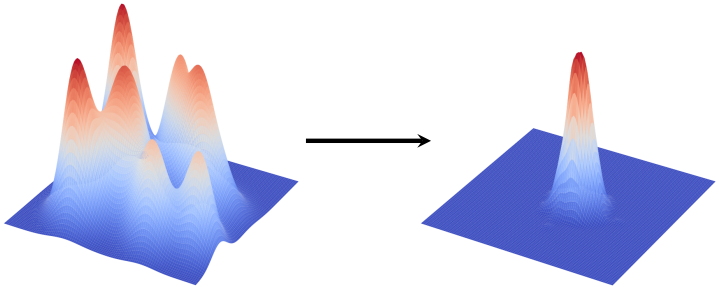
p_{data} p_5

The forward process



p_{data} p_6

The forward process

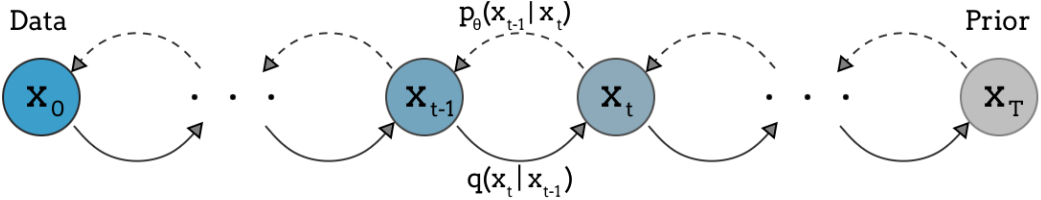


2.1. The forward process

The forward process

Let $\boldsymbol{x} \in \mathbb{R}^{3 \times M \times N}$ be an image, sample from p_{data}. We denote $p_0 := p_{\text{data}}$.

Our objective is to transform p_0 into $p_T \approx \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ with T steps. I will present the process presented in DDPM [Ho et al., 2020][6].



The forward process

Let $\boldsymbol{x} \in \mathbb{R}^{3 \times M \times N}$ be an image, sample from p_{data}. We denote $p_0 := p_{\text{data}}$.

Our objective is to transform p_0 into $p_T \approx \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ with T steps. I will present the process presented in DDPM [Ho et al., 2020][6].

Let $(\beta_t)_{0 \leqslant t \leqslant T}$ be an increasing sequence of real numbers in $[0, 1]$. Let $\alpha_t = 1 - \beta_t$, we would like to construct $(\boldsymbol{x}_t)_{0 \leqslant t \leqslant T}$.

[6] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*

The forward process

Let $\boldsymbol{x} \in \mathbb{R}^{3 \times M \times N}$ be an image, sample from p_{data}. We denote $p_0 := p_{\text{data}}$.

Our objective is to transform p_0 into $p_T \approx \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ with T steps. I will present the process presented in DDPM [Ho et al., 2020][6].

Let $(\beta_t)_{0 \leqslant t \leqslant T}$ be an increasing sequence of real numbers in $[0, 1]$. Let $\alpha_t = 1 - \beta_t$, we would like to construct $(\boldsymbol{x}_t)_{0 \leqslant t \leqslant T}$. For all $t \geqslant 1$,

$$\boldsymbol{x}_t = \sqrt{1 - \beta_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\varepsilon_t, \text{ with } \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

[6] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*

The forward process

Let $\boldsymbol{x} \in \mathbb{R}^{3 \times M \times N}$ be an image, sample from p_{data}. We denote $p_0 := p_{\text{data}}$.

Our objective is to transform p_0 into $p_T \approx \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ with T steps. I will present the process presented in DDPM [Ho et al., 2020][6].

Let $(\beta_t)_{0 \leqslant t \leqslant T}$ be an increasing sequence of real numbers in $[0, 1]$. Let $\alpha_t = 1 - \beta_t$, we would like to construct $(\boldsymbol{x}_t)_{0 \leqslant t \leqslant T}$. For all $t \geqslant 1$,

$$\boldsymbol{x}_t = \sqrt{1 - \beta_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\varepsilon_t, \text{ with } \varepsilon_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

Denoting $\alpha_t = 1 - \beta_t$

$$\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{1 - \alpha_t}\varepsilon_t, \text{ with } \varepsilon_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

[6] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*

The forward process

Let $\boldsymbol{x} \in \mathbb{R}^{3 \times M \times N}$ be an image, sample from p_{data}. We denote $p_0 := p_{\text{data}}$.

Our objective is to transform p_0 into $p_T \approx \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ with T steps. I will present the process presented in DDPM [Ho et al., 2020][6].

Let $(\beta_t)_{0 \leqslant t \leqslant T}$ be an increasing sequence of real numbers in $[0, 1]$. Let $\alpha_t = 1 - \beta_t$, we would like to construct $(\boldsymbol{x}_t)_{0 \leqslant t \leqslant T}$. For all $t \geqslant 1$,

$$\boldsymbol{x}_t = \sqrt{1 - \beta_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\varepsilon_t, \text{ with } \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

Denoting $\alpha_t = 1 - \beta_t$

$$\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{1 - \alpha_t}\varepsilon_t, \text{ with } \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

Other formulation:

$$\boldsymbol{x}_t \sim \mathcal{N}(\boldsymbol{x}_t; \sqrt{1 - \alpha_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I})$$

[6] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*

The forward process

Denoting $\overline{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, For $t \geqslant 1$,

$$\boldsymbol{x}_t \sim \mathcal{N}(\boldsymbol{x}_t; \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0, (1-\overline{\alpha}_t)\boldsymbol{I})$$

In other words, $\boldsymbol{x}_t$ can be expressed as

$$\boldsymbol{x}_t = \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{z}_t, \text{ with } \boldsymbol{z}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$

Example:

If $\boldsymbol{x}_1 = \sqrt{\alpha}_1\boldsymbol{x}_0 + \sqrt{1-\alpha_1}\boldsymbol{\varepsilon}_1$,

$$\begin{aligned}
\boldsymbol{x}_2 &= \sqrt{\alpha}_2\boldsymbol{x}_1 + \sqrt{1-\alpha_2}\boldsymbol{\varepsilon}_2 \\
&= \sqrt{\alpha_2}\sqrt{\alpha_1}\boldsymbol{x}_0 + \sqrt{\alpha_2}\sqrt{1-\alpha_1}\boldsymbol{\varepsilon}_1 + \sqrt{1-\alpha_2}\boldsymbol{\varepsilon}_2 \\
&= \sqrt{\overline{\alpha}_2}\boldsymbol{x}_0 + (\sqrt{1-\alpha_1}\boldsymbol{\varepsilon}_1 + \sqrt{1-\alpha_2}\boldsymbol{\varepsilon}_2)
\end{aligned}$$

And:

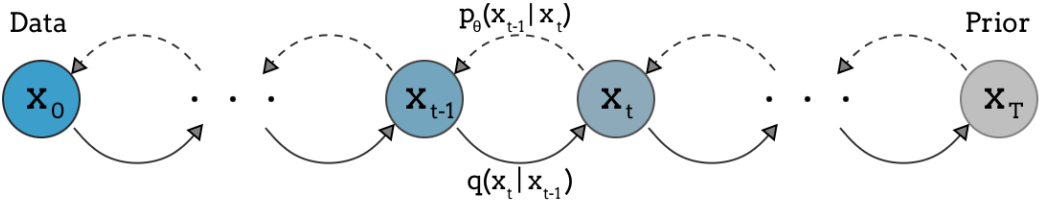
$$(\sqrt{1-\alpha_1}\boldsymbol{\varepsilon}_1 + \sqrt{1-\alpha_2}\boldsymbol{\varepsilon}_2) \sim \mathcal{N}(\boldsymbol{0}, \alpha_2(1-\alpha_1) + 1 - \alpha_2) = \mathcal{N}(\boldsymbol{0}, 1-\overline{\alpha}_2)$$

The forward decomposition

The forward decomposition is:

$$q(\boldsymbol{x}_{0:T}) = q(\boldsymbol{x}_0) \prod_{t=1}^{T} q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$$

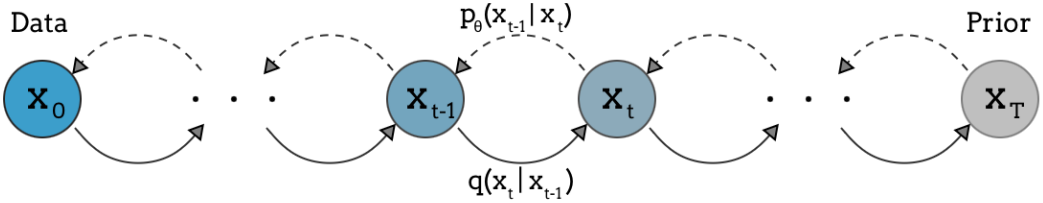
And we denote p_t the law of $\boldsymbol{x}_t$.



To keep in mind: Conditioning on $\boldsymbol{x}_0$, all is possible.

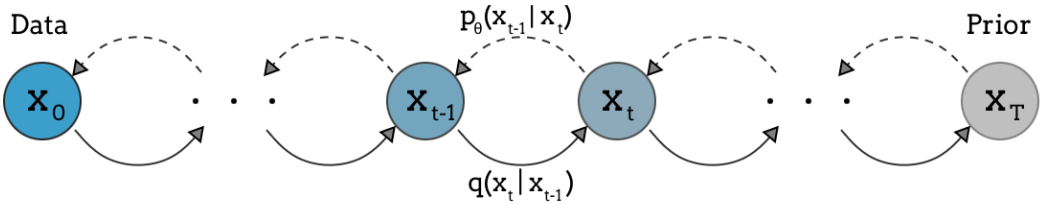
2.2. The backward process

The backward process



[7] Bishop, C. M. (2006). *Pattern recognition and machine learning* (Vol. 4). Springer

The backward process

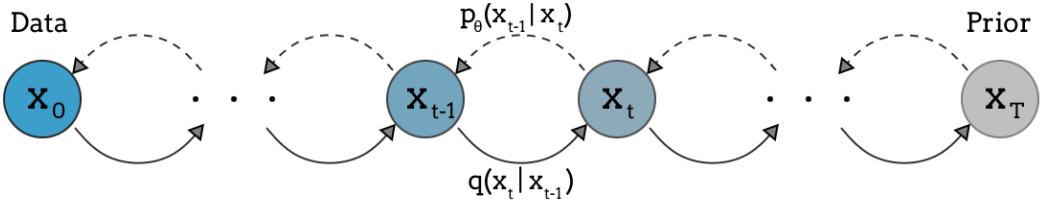


$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0, \boldsymbol{x}_t)$ is tractable.

For example, see [Bishop, 2006][7]

[7] Bishop, C. M. (2006). *Pattern recognition and machine learning* (Vol. 4). Springer

The backward process



$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0, \boldsymbol{x}_t)$ is tractable.

For example, see [Bishop, 2006][7]

$$\boldsymbol{x}_{t-1} \sim \mathcal{N}(\boldsymbol{x}_{t-1}; \tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0), \tilde{\beta}_t \boldsymbol{I})$$

$$\text{with} \quad \tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\overline{\alpha}_{t-1}}}{1-\overline{\alpha}_t}\beta_t \boldsymbol{x}_0 + \frac{\sqrt{\alpha_t}(1-\overline{\alpha}_{t-1})}{1-\overline{\alpha}_t}\boldsymbol{x}_t$$

$$\tilde{\beta}_t = \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t$$

[7] Bishop, C. M. (2006). *Pattern recognition and machine learning* (Vol. 4). Springer

The backward probability p_θ

$$p_\theta(x_{0:T}) = p_\theta(\boldsymbol{x}_T) \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$$

We will suppose that $p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ is Gaussian. [Feller, 1949][8].

This can be also explained in the discrete case by [De Bortoli et al., 2021][9].

$$p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) = \mathcal{N}(\mu_\theta(\boldsymbol{x}_t, t), \Sigma_\theta(\boldsymbol{x}_t))$$

[8] Feller, W. (1949). On the theory of stochastic processes, with particular reference to applications. *Proceedings of the [First] Berkeley Symposium on Mathematical Statistics and Probability*

[9] De Bortoli, V., Thornton, J., Heng, J., & Doucet, A. (2021). Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*

Derivation of the ELBO [Sohl-Dickstein et al., 2015]

$$\begin{aligned}
\log p_\theta(\boldsymbol{x}_0) &= \log \int p_\theta(\boldsymbol{x}_{0:T}) d\boldsymbol{x}_{1:T} \\
&= \log \int \frac{p_\theta(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)} q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0) d\boldsymbol{x}_{1:T} \\
&\geqslant \int \log \frac{p_\theta(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)} q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0) d\boldsymbol{x}_{1:T} \text{ by Jensen's inequality}
\end{aligned}$$

Consequently,

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{x}_0 \sim p_0} \left[\log p_\theta(\boldsymbol{x}_0)\right] &\geqslant \int \int \log \frac{p_\theta(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)} q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0) q(\boldsymbol{x}_0) d\boldsymbol{x}_{1:T} d\boldsymbol{x}_0 \\
&= \int \log \frac{p_\theta(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)} q(\boldsymbol{x}_{0:T}) d\boldsymbol{x}_{0:T} \\
&= \mathbb{E}_{\boldsymbol{x}_{0:T} \sim q} \left[\log \frac{p_\theta(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)}\right]
\end{aligned}$$

Derivation of the ELBO [Sohl-Dickstein et al., 2015]

By the forward and the backward decompositions,

$$\begin{aligned}\log \frac{p_\theta(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0)} &= \log \left[p_\theta(\boldsymbol{x}_T) \prod_{t=1}^{T} \frac{p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)}{q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})} \right] \\ &= \log p_\theta(\boldsymbol{x}_T) + \sum_{t=1}^{T} \log \frac{p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)}{q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})}\end{aligned}$$

Consequently:

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{x}_0 \sim p_0}[\log p_\theta(\boldsymbol{x}_0)] \\ &\geqslant \mathbb{E}_{\boldsymbol{x}_{0:T} \sim q}\left[\log p_\theta(\boldsymbol{x}_T)\right] + \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{x}_{0:T} \sim q}\left[\log \frac{p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)}{q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})}\right] \\ &= \mathbb{E}_{\boldsymbol{x}_{0:T} \sim q}\left[\log p_\theta(\boldsymbol{x}_T)\right] + \sum_{t=2}^{T} \mathbb{E}_{\boldsymbol{x}_{0:T} \sim q}\left[\log \frac{p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)}{q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{x}_0)} \frac{q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0)}{q(\boldsymbol{x}_t \mid \boldsymbol{x}_0)}\right] + \mathbb{E}_{\boldsymbol{x}_{0:T} \sim q}\left[\log \frac{p_\theta(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)}{q(\boldsymbol{x}_1 \mid \boldsymbol{x}_0)}\right] \\ &= \mathbb{E}_{\boldsymbol{x}_{0:T} \sim q}\left[\log \frac{p_\theta(\boldsymbol{x}_T)}{q(\boldsymbol{x}_T \mid \boldsymbol{x}_0)}\right] + \sum_{t=2}^{T} \mathbb{E}_{\boldsymbol{x}_{0:T} \sim q}\left[\log \frac{p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)}{q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{x}_0)}\right] + \mathbb{E}_{\boldsymbol{x}_{0:T} \sim q}\left[\log p_\theta(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)\right]\end{aligned}$$

Derivation of the ELBO [Sohl-Dickstein et al., 2015]

For $2 \leqslant t \leqslant T$,

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{x}_{0:T}}\left[\log \frac{p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)}{q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{x}_0)}\right] &= -\int \log\left(\frac{q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)}{p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)}\right) q(\boldsymbol{x}_{0:T}) d\boldsymbol{x}_{0:T} \\
&= -\int \log\left(\frac{q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)}{p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)}\right) q(\boldsymbol{x}_{0,t-1,t}) d\boldsymbol{x}_{0,t-1,t} \\
&= -\int \log\left(\frac{q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)}{p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)}\right) q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) q(\boldsymbol{x}_t, \boldsymbol{x}_0) d\boldsymbol{x}_{0,t-1,t} \\
&= -\int \left(\int \log\left(\frac{q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0)}{p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)}\right) q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) d\boldsymbol{x}_{t-1}\right) q(\boldsymbol{x}_t, \boldsymbol{x}_0) d\boldsymbol{x}_{0,t} \\
&= -\mathbb{E}_{\boldsymbol{x}_t, \boldsymbol{x}_0 \sim q}\left[\mathsf{D}_{KL}\left(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \| p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)\right)\right]
\end{aligned}$$

Derivation of the ELBO [Sohl-Dickstein et al., 2015]

Finally,

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}_0 \sim p_0}\left[\log p_\theta(\boldsymbol{x}_0)\right] \\
&\geqslant -\mathbb{E}_{\boldsymbol{x}_0}\left[\mathsf{D}_{KL}(q(\boldsymbol{x}_T \mid \boldsymbol{x}_0) \| p_\theta(\boldsymbol{x}_T))\right] - \sum_{t=2}^{T} \mathbb{E}_{\boldsymbol{x}_t, \boldsymbol{x}_0 \sim q}\left[\mathsf{D}_{KL}\left(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \| p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)\right)\right] \\
&+ \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_1}\left[\log p_\theta(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)\right]
\end{aligned}$$

For $2 \leqslant t \leqslant T$, we are looking for $\mathsf{D}_{KL}\left(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \| p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)\right)$,

By [Zhang et al., 2021][10],

$$\mathsf{D}_{KL}(\mathcal{N}(\mu_1, \Sigma_1), \mathcal{N}(\mu_2, \Sigma_2)) = \frac{1}{2}\left[\log \frac{|\Sigma_2|}{|\Sigma_1|} + \operatorname{Tr}(\Sigma_2^{-1}\Sigma_1) + (\mu_2 - \mu_1)^T \Sigma_2^{-1}(\mu_2 - \mu_1) - n\right]$$

As a reminder,

$$\boldsymbol{x}_{t-1} \sim \mathscr{N}(\boldsymbol{x}_{t-1}; \tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0), \tilde{\beta}_t \boldsymbol{I})$$

If we suppose that $p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) = \mathscr{N}(\mu_\theta(\boldsymbol{x}_t, t), \Sigma_\theta(\boldsymbol{x}_t, t))$ and $\Sigma_\theta(\boldsymbol{x}_t, t) = \tilde{\beta}_t \boldsymbol{I}$,

$$\mathsf{D}_{KL}\left(q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) \| p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)\right) = \frac{1}{2\tilde{\beta}_t^2} \|\tilde{\mu}_t(\boldsymbol{x}_t, x_0) - \mu_\theta(\boldsymbol{x}_t, t)\|^2$$

[10] Zhang, Y., Liu, W., Chen, Z., Li, K., & Wang, J. (2021). On the properties of kullback-leibler divergence between gaussians. *CoRR, abs/2102.05485*. https://arxiv.org/abs/2102.05485

For $t = T$, $\mathbb{E}_{\boldsymbol{x}_0}\left[\mathrm{D}_{KL}(q(\boldsymbol{x}_T \mid \boldsymbol{x}_0) \| p_\theta(\boldsymbol{x}_T))\right]$ is neglected because

$$q(\boldsymbol{x}_T \mid \boldsymbol{x}_0) = \mathscr{N}(\mathbb{E}(\boldsymbol{x}_0), \sqrt{1-\overline{\alpha}_T}\,\mathrm{Var}(\boldsymbol{x}_0)) \approx \mathscr{N}(\boldsymbol{0}, \boldsymbol{I})$$

For $t = 0$, $\mu_t heta(\boldsymbol{x}_1, 1)$ is penalized to obtain data images in $[0, 1]$

The loss is finally,

$$L(\theta) = \sum_{t=2}^{T} \frac{1}{2\tilde{\beta}_t^2} \|\tilde{\mu}_t(\boldsymbol{x}_t, x_0) - \mu_\theta(\boldsymbol{x}_t, t)\|^2 - \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{x}_1}\left[\log p_\theta(\boldsymbol{x}_0 \mid \boldsymbol{x}_1)\right]$$

Problem: This loss is difficult to train, even for 2D distribution (see tutorials at this url[11])

[11] https://github.com/acids-ircam/diffusion_models

Solution:

Denoting $\boldsymbol{x}_t = \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\varepsilon}$ with $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$,

$$L_{t-1} = \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{\varepsilon}} \left[\frac{1}{2\sigma_t^2} \left\| \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{x}_t(\boldsymbol{x}_0, \boldsymbol{\varepsilon}) - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}} \boldsymbol{\varepsilon} \right) - \mu_\theta(\boldsymbol{x}_t(\boldsymbol{x}_0, \boldsymbol{\varepsilon}), t) \right\|^2 \right].$$

Solution:

Denoting $\boldsymbol{x}_t = \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\varepsilon}$ with $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$,

$$L_{t-1} = \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{\varepsilon}} \left[\frac{1}{2\sigma_t^2} \left\| \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{x}_t(\boldsymbol{x}_0, \boldsymbol{\varepsilon}) - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}} \boldsymbol{\varepsilon} \right) - \mu_\theta(\boldsymbol{x}_t(\boldsymbol{x}_0, \boldsymbol{\varepsilon}), t) \right\|^2 \right].$$

$$\mu_\theta(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{x}_t(\boldsymbol{x}_0, \boldsymbol{\varepsilon}) - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}} \boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t) \right)$$

$$\Longleftrightarrow$$

$$\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t) = \frac{\sqrt{1-\overline{\alpha}_t}}{\beta_t} \left(\boldsymbol{x}_t - \sqrt{\alpha_t} \mu_\theta(\boldsymbol{x}_t, t) \right)$$

Solution:

Denoting $\boldsymbol{x}_t = \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\varepsilon}$ with $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$,

$$L_{t-1} = \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\varepsilon}}\left[\frac{1}{2\sigma_t^2}\left\|\frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t(\boldsymbol{x}_0,\boldsymbol{\varepsilon}) - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\boldsymbol{\varepsilon}\right) - \mu_\theta(\boldsymbol{x}_t(\boldsymbol{x}_0,\boldsymbol{\varepsilon}),t)\right\|^2\right].$$

$$\mu_\theta(\boldsymbol{x}_t,t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t(\boldsymbol{x}_0,\boldsymbol{\varepsilon}) - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t,t)\right)$$

$$\Longleftrightarrow$$

$$\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t,t) = \frac{\sqrt{1-\overline{\alpha}_t}}{\beta_t}\left(\boldsymbol{x}_t - \sqrt{\alpha_t}\mu_\theta(\boldsymbol{x}_t,t)\right)$$

$$L_{t-1} = \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\varepsilon}}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\overline{\alpha}_t)}\left\|\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t(\boldsymbol{x}_0,\boldsymbol{\varepsilon}))\right\|^2\right].$$

Solution:

Denoting $\boldsymbol{x}_t = \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\varepsilon}$ with $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$,

$$L_{t-1} = \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\varepsilon}}\left[\frac{1}{2\sigma_t^2}\left\|\frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t(\boldsymbol{x}_0,\boldsymbol{\varepsilon}) - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\boldsymbol{\varepsilon}\right) - \mu_\theta(\boldsymbol{x}_t(\boldsymbol{x}_0,\boldsymbol{\varepsilon}),t)\right\|^2\right].$$

$$\mu_\theta(\boldsymbol{x}_t,t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t(\boldsymbol{x}_0,\boldsymbol{\varepsilon}) - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t,t)\right)$$

$$\Longleftrightarrow$$

$$\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t,t) = \frac{\sqrt{1-\overline{\alpha}_t}}{\beta_t}\left(\boldsymbol{x}_t - \sqrt{\alpha_t}\mu_\theta(\boldsymbol{x}_t,t)\right)$$

$$L_{t-1} = \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\varepsilon}}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\overline{\alpha}_t)}\left\|\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t(\boldsymbol{x}_0,\boldsymbol{\varepsilon}))\right\|^2\right].$$

$$L_{\text{simple}}(\theta) = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\varepsilon}}\left[\left\|\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t(\boldsymbol{x}_0,\boldsymbol{\varepsilon}),t)\right\|^2\right].$$

To convince yourself: Residual learning for denoiser.

Discrete backward process

The backward process becomes:

$$\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}} \boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t) \right) + \tilde{\beta}_t \boldsymbol{z}_t \text{ with } \boldsymbol{z}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

Discrete backward process

The backward process becomes:

$$\begin{aligned}\boldsymbol{x}_T &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \\ \boldsymbol{x}_{t-1} &= \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)\right) + \tilde{\beta}_t \boldsymbol{z}_t \text{ with } \boldsymbol{z}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})\end{aligned}$$

Ho et al., 2020 has the same results with:

$$\begin{aligned}\boldsymbol{x}_T &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \\ \boldsymbol{x}_{t-1} &= \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)\right) + \beta_t \boldsymbol{z}_t \text{ with } \boldsymbol{z}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})\end{aligned}$$

As a reminder, $\tilde{\beta}_t = \frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\beta_t$

$$L(\theta) = \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\boldsymbol{\epsilon}_t}\left[\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2\right]$$
$$= \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\boldsymbol{\epsilon}_t}\left[\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t, t)\|^2\right]$$

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t\mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Some implementation tricks: Exponential Moving Average (EMA)

During the training, for each training step,

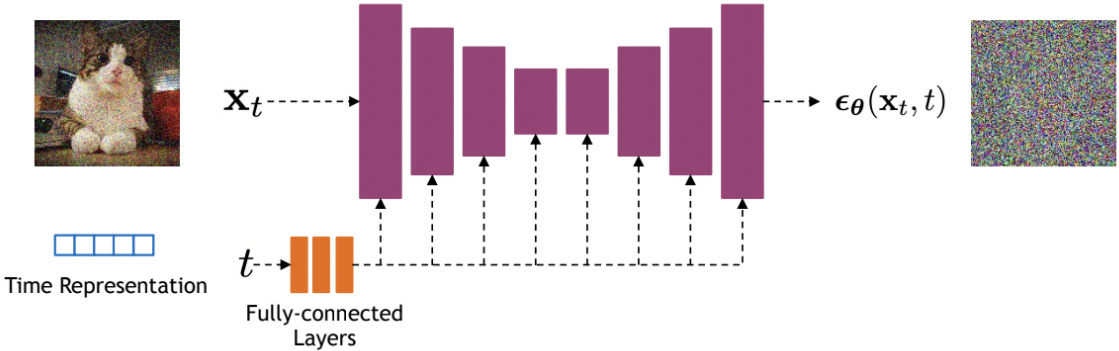
$\tilde{\theta}_{n+1} = (1 - \mu)\theta_{n+1} + \mu\theta_n$

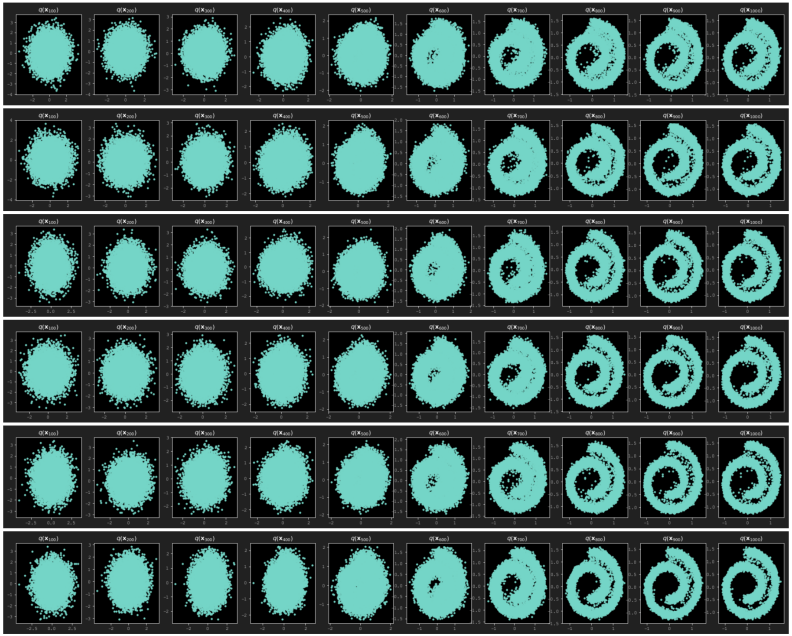
with $\mu = 0.9$ for example.

Time representation

Network architecture: UNET

Time representation: sinusoidal positional embeddings or random Fourier features and the network is a U-NET.

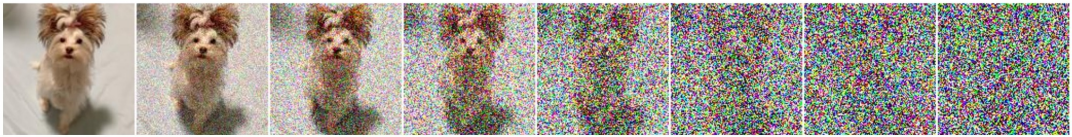




2.3. The continuous framework

Forward SDE (data → noise)

$\mathbf{x}(0)$ — $\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$ → $\mathbf{x}(T)$



score function

$\mathbf{x}(0)$ ← $\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$ — $\mathbf{x}(T)$

Reverse SDE (noise → data)

Image extracted from [Y. Song et al., 2023]

Study of the continuous framework [Y. Song et al., 2023][12]

GROSSISSEMENT D'UNE FILTRATION

ET RETOURNEMENT DU TEMPS D'UNE DIFFUSION

E.PARDOUX

1. Introduction .

Soit $\{ X_t, 0 \leqslant t \leqslant 1 \}$ un processus de diffusion dans $\mathbb{R}^d$ solution de l'E.D.S:

$$dX_t = b(t,X_t)dt + \sigma(t,X_t)dW_t$$

où $\{W_t, 0 \leqslant t \leqslant 1 \}$ est un mouvement brownien standard dans $\mathbb{R}^\ell$. Continuant le travail de [2], nous nous posons la question suivante: existe-t-il un brownien standard dans $\mathbb{R}^\ell$ $\{\overline{W}_t, 0 \leqslant t \leqslant 1 \}$ et des coefficients $\{ \overline{b}(t,x), \overline{\sigma}(t,x); 0 \leqslant t \leqslant 1, x \in \mathbb{R}^d \}$ tels que le processus $\overline{X}_t = X_{1-t}, 0 \leqslant t \leqslant 1$, soit solution de :

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

Notre méthode consiste à identifier $\{\overline{W}_t\}$,en résolvant un problème de grossissement de filtration . On pourrait probablement déduire le résultat ci-dessous de ceux de Jeulin [4] et de Jacod

[12] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2023). Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*

SDE in one slide

What means $d\boldsymbol{x}_t = b(t, \boldsymbol{x}_t)dt + \sigma(t)d\boldsymbol{w}_t$?

SDE in one slide

What means $d\boldsymbol{x}_t = b(t, \boldsymbol{x}_t)dt + \sigma(t)d\boldsymbol{w}_t$?

$$\boldsymbol{x}_t = \boldsymbol{x}_0 + \int_0^t b(s, \boldsymbol{x}_s)ds + \int_0^t \sigma(s)ds$$

SDE in one slide

What means $d\boldsymbol{x}_t = b(t, \boldsymbol{x}_t)dt + \sigma(t)d\boldsymbol{w}_t$?

$$\boldsymbol{x}_t = \boldsymbol{x}_0 + \int_0^t b(s, \boldsymbol{x}_s)ds + \int_0^t \sigma(s)ds$$

Discretly,

$$\boldsymbol{x}(t + \Delta t) - \boldsymbol{x}(t) \approx \Delta t b(t, \boldsymbol{x}(t)) + \sigma(t)\sqrt{\Delta t}\xi_t, \text{ with } \xi_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

Continuous framework

Forward process: Let $(\boldsymbol{x}_t)_{0 \leqslant t \leqslant 1}$ be a diffusion process verifying the SDE:

$$d\boldsymbol{x}_t = b(t, \boldsymbol{x}_t)dt + \sigma(t)d\boldsymbol{w}_t$$

where $(\boldsymbol{w}_t)_{0 \leqslant t \leqslant 1}$ is a standard Brownian motion.

Continuous framework

Forward process: Let $(\boldsymbol{x}_t)_{0 \leqslant t \leqslant 1}$ be a diffusion process verifying the SDE:

$$d\boldsymbol{x}_t = b(t, \boldsymbol{x}_t)dt + \sigma(t)d\boldsymbol{w}_t$$

where $(\boldsymbol{w}_t)_{0 \leqslant t \leqslant 1}$ is a standard Brownian motion. **Example:**

$$d\boldsymbol{x}_t = -\frac{1}{2}\beta(t)\boldsymbol{x}_t dt + \sqrt{\beta(t)}d\boldsymbol{w}_t$$

Forward process: Let $(\boldsymbol{x}_t)_{0 \leqslant t \leqslant 1}$ be a diffusion process verifying the SDE:

$$d\boldsymbol{x}_t = b(t, \boldsymbol{x}_t)dt + \sigma(t)d\boldsymbol{w}_t$$

where $(\boldsymbol{w}_t)_{0 \leqslant t \leqslant 1}$ is a standard Brownian motion. **Example:**

$$d\boldsymbol{x}_t = -\frac{1}{2}\beta(t)\boldsymbol{x}_t dt + \sqrt{\beta(t)}d\boldsymbol{w}_t$$

$$\boldsymbol{x}(t + \Delta t) = \boldsymbol{x}_t - \frac{1}{2}\beta(t)\Delta t\boldsymbol{x}_t + \sqrt{\beta(t)\Delta t}\boldsymbol{\varepsilon}_t$$

Forward process: Let $(\boldsymbol{x}_t)_{0 \leqslant t \leqslant 1}$ be a diffusion process verifying the SDE:

$$d\boldsymbol{x}_t = b(t, \boldsymbol{x}_t)dt + \sigma(t)d\boldsymbol{w}_t$$

where $(\boldsymbol{w}_t)_{0 \leqslant t \leqslant 1}$ is a standard Brownian motion. **Example:**

$$d\boldsymbol{x}_t = -\frac{1}{2}\beta(t)\boldsymbol{x}_t dt + \sqrt{\beta(t)}d\boldsymbol{w}_t$$

$$\begin{aligned}\boldsymbol{x}(t + \Delta t) &= \boldsymbol{x}_t - \frac{1}{2}\beta(t)\Delta t\boldsymbol{x}_t + \sqrt{\beta(t)\Delta t}\boldsymbol{\varepsilon}_t \\ &= (1 - \frac{1}{2}\beta(t)\Delta t)\boldsymbol{x}_t + \sqrt{\beta(t)\Delta t}\boldsymbol{\varepsilon}_t\end{aligned}$$

Forward process: Let $(\boldsymbol{x}_t)_{0 \leqslant t \leqslant 1}$ be a diffusion process verifying the SDE:

$$d\boldsymbol{x}_t = b(t, \boldsymbol{x}_t)dt + \sigma(t)d\boldsymbol{w}_t$$

where $(\boldsymbol{w}_t)_{0 \leqslant t \leqslant 1}$ is a standard Brownian motion. **Example:**

$$d\boldsymbol{x}_t = -\frac{1}{2}\beta(t)\boldsymbol{x}_t dt + \sqrt{\beta(t)}d\boldsymbol{w}_t$$

$$\begin{aligned}
\boldsymbol{x}(t + \Delta t) &= \boldsymbol{x}_t - \frac{1}{2}\beta(t)\Delta t\boldsymbol{x}_t + \sqrt{\beta(t)\Delta t}\boldsymbol{\varepsilon}_t \\
&= (1 - \frac{1}{2}\beta(t)\Delta t)\boldsymbol{x}_t + \sqrt{\beta(t)\Delta t}\boldsymbol{\varepsilon}_t \\
&\approx \sqrt{1 - \beta(t)\Delta t}\boldsymbol{x}_t + \sqrt{\beta(t)\Delta t}\boldsymbol{\varepsilon}_t
\end{aligned}$$

Forward process: Let $(\boldsymbol{x}_t)_{0 \leqslant t \leqslant 1}$ be a diffusion process verifying the SDE:

$$d\boldsymbol{x}_t = b(t, \boldsymbol{x}_t)dt + \sigma(t)d\boldsymbol{w}_t$$

where $(\boldsymbol{w}_t)_{0 \leqslant t \leqslant 1}$ is a standard Brownian motion. **Example:**

$$d\boldsymbol{x}_t = -\frac{1}{2}\beta(t)\boldsymbol{x}_t dt + \sqrt{\beta(t)}d\boldsymbol{w}_t$$

$$\begin{aligned}
\boldsymbol{x}(t + \Delta t) &= \boldsymbol{x}_t - \frac{1}{2}\beta(t)\Delta t\boldsymbol{x}_t + \sqrt{\beta(t)\Delta t}\boldsymbol{\varepsilon}_t \\
&= (1 - \frac{1}{2}\beta(t)\Delta t)\boldsymbol{x}_t + \sqrt{\beta(t)\Delta t}\boldsymbol{\varepsilon}_t \\
&\approx \sqrt{1 - \beta(t)\Delta t}\boldsymbol{x}_t + \sqrt{\beta(t)\Delta t}\boldsymbol{\varepsilon}_t \\
&\approx \sqrt{1 - \beta_t}\boldsymbol{x}_t + \sqrt{\beta_t}\boldsymbol{z}_t
\end{aligned}$$

Forward process: Let $(\boldsymbol{x}_t)_{0 \leqslant t \leqslant 1}$ be a diffusion process verifying the SDE:

$$d\boldsymbol{x}_t = b(t, \boldsymbol{x}_t)dt + \sigma(t)d\boldsymbol{w}_t$$

where $(\boldsymbol{w}_t)_{0 \leqslant t \leqslant 1}$ is a standard Brownian motion. **Example:**

$$d\boldsymbol{x}_t = -\frac{1}{2}\beta(t)\boldsymbol{x}_t dt + \sqrt{\beta(t)}d\boldsymbol{w}_t$$

$$\begin{aligned}
\boldsymbol{x}(t + \Delta t) &= \boldsymbol{x}_t - \frac{1}{2}\beta(t)\Delta t\boldsymbol{x}_t + \sqrt{\beta(t)\Delta t}\boldsymbol{\varepsilon}_t \\
&= (1 - \frac{1}{2}\beta(t)\Delta t)\boldsymbol{x}_t + \sqrt{\beta(t)\Delta t}\boldsymbol{\varepsilon}_t \\
&\approx \sqrt{1 - \beta(t)\Delta t}\boldsymbol{x}_t + \sqrt{\beta(t)\Delta t}\boldsymbol{\varepsilon}_t \\
&\approx \sqrt{1 - \beta_t}\boldsymbol{x}_t + \sqrt{\beta_t}\boldsymbol{z}_t
\end{aligned}$$

It is DDPM ! With $\beta_t = \Delta_t \beta \left(\frac{t}{T}\right)$

Question from [Pardoux, 1986][13]: Is there $\overline{b}, \overline{\sigma}$ such that $\overline{\boldsymbol{x}}_t = \boldsymbol{x}_{1-t}$ is solution of the reverse SDE:

$$d\overline{\boldsymbol{x}}_t = \overline{b}(t, \overline{\boldsymbol{x}}_t)dt + \overline{\sigma}(t)d\overline{\boldsymbol{w}}_t$$

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

Question from [Pardoux, 1986][13]: Is there $\overline{b}, \overline{\sigma}$ such that $\overline{\boldsymbol{x}}_t = \boldsymbol{x}_{1-t}$ is solution of the reverse SDE:

$$d\overline{\boldsymbol{x}}_t = \overline{b}(t, \overline{\boldsymbol{x}}_t)dt + \overline{\sigma}(t)d\overline{\boldsymbol{w}}_t$$

Backward process: Under certain assumptions, $\boldsymbol{x}_t$ is a weak solution of the following SDE, backward in time:

$$d\boldsymbol{x}_t = \left[b(t, \boldsymbol{x}_t) - \sigma(t)^2 \nabla_{\boldsymbol{x}} \log p_t(\overline{\boldsymbol{x}}_t)\right] dt + \sigma(t) \otimes d\widehat{\boldsymbol{w}}_t$$

with $\widehat{\boldsymbol{w}}_t = \boldsymbol{w}_t - \boldsymbol{w}_1 - \int_t^1 \nabla \log p_s(\boldsymbol{x}_s) ds$ which is a backward Brownian motion adapted to the filtration $\sigma(\boldsymbol{x}_t \cup \{\boldsymbol{w}_s - w_t, t \leqslant s \leqslant 1\})$

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

Question from [Pardoux, 1986][13]: Is there $\overline{b}, \overline{\sigma}$ such that $\overline{\boldsymbol{x}}_t = \boldsymbol{x}_{1-t}$ is solution of the reverse SDE:

$$d\overline{\boldsymbol{x}}_t = \overline{b}(t, \overline{\boldsymbol{x}}_t)dt + \overline{\sigma}(t)d\overline{\boldsymbol{w}}_t$$

Backward process: Under certain assumptions, $\boldsymbol{x}_t$ is a weak solution of the following SDE, backward in time:

$$d\boldsymbol{x}_t = \left[b(t, \boldsymbol{x}_t) - \sigma(t)^2 \nabla_{\boldsymbol{x}} \log p_t(\overline{\boldsymbol{x}}_t)\right] dt + \sigma(t) \otimes d\widehat{\boldsymbol{w}}_t$$

with $\widehat{\boldsymbol{w}}_t = \boldsymbol{w}_t - \boldsymbol{w}_1 - \int_t^1 \nabla \log p_s(\boldsymbol{x}_s) ds$ which is a backward Brownian motion adapted to the filtration $\sigma(\boldsymbol{x}_t \cup \{\boldsymbol{w}_s - w_t, t \leqslant s \leqslant 1\})$

That means:

$$\begin{aligned}
\overline{b}(t, \overline{\boldsymbol{x}}_t) &= -b(1-t, \overline{\boldsymbol{x}}_t) - \sigma(1-t)^2 \nabla_{\boldsymbol{x}} \log p_{1-t}(\overline{\boldsymbol{x}}) \\
\overline{\sigma}(t) &= -\sigma(1-t) \\
\overline{\boldsymbol{w}}_t &= \boldsymbol{w}_{1-t} - \boldsymbol{w}_1 - \int_{1-t}^1 \nabla \log p_s(\boldsymbol{x}_s) ds
\end{aligned}$$

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

Link with the discrete case

$\boldsymbol{x}_t \approx \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)$ and:

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) \propto \exp\left(-\frac{\|\boldsymbol{x}_t - \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0\|^2}{2(1-\overline{\alpha}_t)}\right)$$

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

Link with the discrete case

$\boldsymbol{x}_t \approx \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)$ and:

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) \propto \exp\left(-\frac{\|\boldsymbol{x}_t - \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0\|^2}{2(1-\overline{\alpha}_t)}\right)$$

$$\nabla \log q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = -\frac{\boldsymbol{x}_t - \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0}{1-\overline{\alpha}_t} \approx -\frac{\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)}{1-\overline{\alpha}_t}$$

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

Link with the discrete case

$\boldsymbol{x}_t \approx \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)$ and:

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) \propto \exp\left(-\frac{\|\boldsymbol{x}_t - \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0\|^2}{2(1-\overline{\alpha}_t)}\right)$$

$$\nabla \log q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = -\frac{\boldsymbol{x}_t - \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0}{1-\overline{\alpha}_t} \approx -\frac{\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)}{1-\overline{\alpha}_t}$$

Consequently, $-\frac{\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)}{1-\overline{\alpha}_t}$ minimizes $\mathbb{E}_{\boldsymbol{x}_0}\left(\|\nabla \log q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) - X\|^2\right)$

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

Link with the discrete case

$\boldsymbol{x}_t \approx \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)$ and:

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) \propto \exp\left(-\frac{\|\boldsymbol{x}_t - \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0\|^2}{2(1-\overline{\alpha}_t)}\right)$$

$$\nabla \log q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = -\frac{\boldsymbol{x}_t - \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0}{1-\overline{\alpha}_t} \approx -\frac{\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)}{1-\overline{\alpha}_t}$$

Consequently, $-\frac{\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)}{1-\overline{\alpha}_t}$ minimizes $\mathbb{E}_{\boldsymbol{x}_0}\left(\|\nabla \log q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) - X\|^2\right)$

and, $s_\theta(\boldsymbol{x}_t, t) = -\frac{\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)}{1-\overline{\alpha}_t} \approx \nabla \log q(\boldsymbol{x}_t)$

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

Link with the discrete case

$\boldsymbol{x}_t \approx \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)$ and:

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) \propto \exp\left(-\frac{\|\boldsymbol{x}_t - \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0\|^2}{2(1-\overline{\alpha}_t)}\right)$$

$$\nabla \log q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = -\frac{\boldsymbol{x}_t - \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0}{1-\overline{\alpha}_t} \approx -\frac{\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)}{1-\overline{\alpha}_t}$$

Consequently, $-\frac{\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)}{1-\overline{\alpha}_t}$ minimizes $\mathbb{E}_{\boldsymbol{x}_0}\left(\|\nabla \log q(\boldsymbol{x}_t \mid \boldsymbol{x}_0) - X\|^2\right)$

and, $s_\theta(\boldsymbol{x}_t, t) = -\frac{\boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t)}{1-\overline{\alpha}_t} \approx \nabla \log q(\boldsymbol{x}_t)$

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t + (1-\alpha_t)s_\theta(\boldsymbol{x}_t, t)\right) + \tilde{\beta}_t \boldsymbol{z}_t \text{ with } \boldsymbol{z}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$

This is well explained in [Chung et al., 2022][14]

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

With $b(t, \boldsymbol{x}) = -\frac{1}{2}\beta(t)\boldsymbol{x}, \sigma(t) = \sqrt{\beta(t)}$,

$\overline{\boldsymbol{x}}_t = \left[b(t, \overline{\boldsymbol{x}}_t) - \sigma(t)^2 \nabla_{\boldsymbol{x}} \log p_t(\overline{\boldsymbol{x}})\right] dt + \sigma(t) d\overline{\boldsymbol{w}}_t$

becomes:

$$\begin{array}{ccccc} d\overline{\boldsymbol{x}}_t & = & \frac{1}{2}\beta(t)\boldsymbol{x}dt & -\beta(t)\nabla_{\boldsymbol{x}} \log p_t(\overline{\boldsymbol{x}})dt & \sqrt{\beta(t)}d\overline{\boldsymbol{w}}_t \\ \boldsymbol{x}_{t-1} & = & \frac{1}{\sqrt{1-\beta_t}}\boldsymbol{x}_t & +\frac{\beta_t}{\sqrt{1-\beta_t}} s_\theta(\boldsymbol{x}_t, t) & \tilde{\beta}_t \boldsymbol{z}_t \end{array}$$

And $\frac{1}{\sqrt{1-\beta_t}} \approx 1 + \frac{1}{2}\beta_t$, $\frac{\beta_t}{\sqrt{1-\beta_t}} \approx \beta_t(1 + \frac{1}{2}\beta_t) \approx \beta_t$

To other approaches

Can be applied to other SDE as the model SMLD [Y. Song and Ermon, 2019][15] where $d\boldsymbol{x}_t = \sqrt{\frac{d[\sigma^2(t)]}{dt}} d\boldsymbol{w}_t$

[15] Song, Y., & Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*

To other approaches

To a SDE $d\boldsymbol{x}_t = f(\boldsymbol{x}_t, t)dt + g(t)d\boldsymbol{w}_t$ can be associated an ODE: $d\boldsymbol{x} = \left[f(\boldsymbol{x}, t) - \frac{1}{2}g(t)^2 \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right] dt$

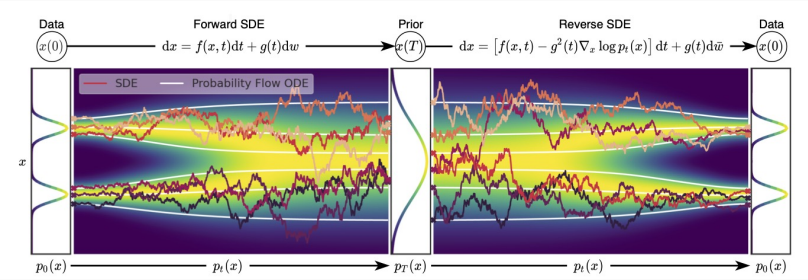


Image extracted from [Y. Song et al., 2023][16]

[16] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2023). Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*

2.4. Conclusion

Conclusion

- Speed-up the convergence. [De Bortoli et al., 2021][17], [Dhariwal and Nichol, 2021][18]
- The data distribution does not have density Bortoli, 2022[19]
- Not presented here: DDIM [J. Song et al., 2021][20].
- Interesting tutorials: Song's code: https://github.com/yang-song/score_sde, 2D models: https://github.com/acids-ircam/diffusion_models/tree/main
- Reading recommendation : [Ho et al., 2020][21], [Y. Song et al., 2023][22]

[17] De Bortoli, V., Thornton, J., Heng, J., & Doucet, A. (2021). Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*

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

[19] Bortoli, V. D. (2022). Convergence of denoising diffusion models under the manifold hypothesis. *Transactions on Machine Learning Research*

[20] Song, J., Meng, C., & Ermon, S. (2021). Denoising diffusion implicit models. *International Conference on Learning Representations*

[21] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*

[22] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2023). Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*

Presentation of RePaint

Introduction

Various approaches:

- Re-train a score-based model. [Saharia et al., 2022][23]
- Classifier quidance method. [Y. Song et al., 2023][24]
- The replacement method, presented today. [Lugmayr et al., 2022][25],[Chung et al., 2022][26]
- Pseudo-inverse reasonning [Choi et al., 2021][27]

[23] Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., & Norouzi, M. (2022). Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*

[24] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2023). Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*

[25] Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., & Van Gool, L. (2022). RePaint: Inpainting using denoising diffusion probabilistic models. *RePaint*

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

[27] Choi, J., Kim, S., Jeong, Y., Gwon, Y., & Yoon, S. (2021). ILVR: Conditioning method for denoising diffusion probabilistic models. *ILVR*

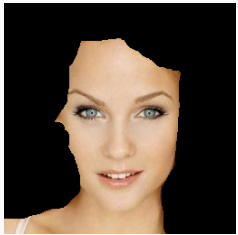
Diverse impaiting

Let suppose that $\boldsymbol{x} \sim p_{\text{data}}$, we suppose that there exists a mask $\boldsymbol{m}$ such that $\boldsymbol{mx}$ is **known**.

Objective: Sample $\boldsymbol{x}$ conditioned on $\boldsymbol{mx} = \boldsymbol{y}$.

Example:

x



$\boldsymbol{y} = \boldsymbol{mx}$

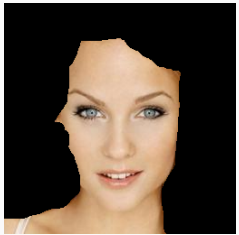
Diverse impaiting

Let suppose that $\boldsymbol{x} \sim p_{\text{data}}$, we suppose that there exists a mask $\boldsymbol{m}$ such that $\boldsymbol{mx}$ is **known**.

Objective: Sample $\boldsymbol{x}$ conditioned on $\boldsymbol{mx} = \boldsymbol{y}$.

Example:

x



$\boldsymbol{y} = \boldsymbol{mx}$

Sample 1

Sample 2

How to build a conditional diffusion model ? Y. Song et al., 2023

Let denote $\overline{\boldsymbol{m}} = 1 - \boldsymbol{m}$

Objective: Sample $\boldsymbol{x}_0$ conditioned on $\boldsymbol{m}\boldsymbol{x}_0 = \boldsymbol{y}$.

Let suppose that we know a diffusion model for $\boldsymbol{x}_0 \sim p_{\text{data}}$.

We would like to model the evolution of $\boldsymbol{z}_t = \overline{\boldsymbol{m}}\boldsymbol{x}_t$ for the forward and the backward processes.

How to build a conditional diffusion model ? Y. Song et al., 2023

Let denote $\overline{\boldsymbol{m}} = 1 - \boldsymbol{m}$

Objective: Sample $\boldsymbol{x}_0$ conditioned on $\boldsymbol{m}\boldsymbol{x}_0 = \boldsymbol{y}$.

Let suppose that we know a diffusion model for $\boldsymbol{x}_0 \sim p_{\text{data}}$.

We would like to model the evolution of $\boldsymbol{z}_t = \overline{\boldsymbol{m}}\boldsymbol{x}_t$ for the forward and the backward processes.

The forward process is:

$$d\boldsymbol{z}_t = -\frac{1}{2}\beta(t)\boldsymbol{z}_t dt + \sqrt{\beta(t)}d\boldsymbol{w}_t$$

How to build a conditional diffusion model ? Y. Song et al., 2023

Let denote $\overline{\boldsymbol{m}} = 1 - \boldsymbol{m}$

Objective: Sample $\boldsymbol{x}_0$ conditioned on $\boldsymbol{m}\boldsymbol{x}_0 = \boldsymbol{y}$.

Let suppose that we know a diffusion model for $\boldsymbol{x}_0 \sim p_{\text{data}}$.

We would like to model the evolution of $\boldsymbol{z}_t = \overline{\boldsymbol{m}}\boldsymbol{x}_t$ for the forward and the backward processes.

The forward process is:

$$d\boldsymbol{z}_t = -\frac{1}{2}\beta(t)\boldsymbol{z}_t dt + \sqrt{\beta(t)}d\boldsymbol{w}_t$$

The reversed SDE is conditionned on $m\boldsymbol{x}_0 = \boldsymbol{y}$ and becomes:

$$d\boldsymbol{z}_t = \left[-\frac{1}{2}\beta(t)\boldsymbol{z}_t - \beta(t)\nabla_{\boldsymbol{z}} \log p\left(\boldsymbol{z}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right)\right] dt + \sqrt{\beta(t)}d\overline{\boldsymbol{w}}_t$$

How to build a conditional diffusion model ? Y. Song et al., 2023

Let denote $\overline{\boldsymbol{m}} = 1 - \boldsymbol{m}$

Objective: Sample $\boldsymbol{x}_0$ conditioned on $\boldsymbol{m}\boldsymbol{x}_0 = \boldsymbol{y}$.

Let suppose that we know a diffusion model for $\boldsymbol{x}_0 \sim p_{\text{data}}$.

We would like to model the evolution of $\boldsymbol{z}_t = \overline{\boldsymbol{m}}\boldsymbol{x}_t$ for the forward and the backward processes.

The forward process is:

$$d\boldsymbol{z}_t = -\frac{1}{2}\beta(t)\boldsymbol{z}_t dt + \sqrt{\beta(t)}d\boldsymbol{w}_t$$

The reversed SDE is conditionned on $m\boldsymbol{x}_0 = \boldsymbol{y}$ and becomes:

$$d\boldsymbol{z}_t = \left[-\frac{1}{2}\beta(t)\boldsymbol{z}_t - \beta(t)\nabla_{\boldsymbol{z}} \log p\left(\boldsymbol{z}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right)\right] dt + \sqrt{\beta(t)}d\overline{\boldsymbol{w}}_t$$

Problem: $\nabla_{\boldsymbol{z}} \log p_t\left(\boldsymbol{z}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right)$ is not tractable.

Let denote $\overline{\boldsymbol{m}} = 1 - \boldsymbol{m}$

$$p_t \left[\boldsymbol{z}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y} \right]$$

Let denote $\overline{\boldsymbol{m}} = 1 - \boldsymbol{m}$

$$p_t\left[\boldsymbol{z}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right] = p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right]$$

Let denote $\overline{\boldsymbol{m}} = 1 - \boldsymbol{m}$

$$
\begin{aligned}
p_t\left[\boldsymbol{z}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right] &= p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right] \\
&= \int p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}, \boldsymbol{m}\boldsymbol{x}_t\right] p_t\left[\boldsymbol{m}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{z}_0 = \boldsymbol{y}\right] d(\boldsymbol{m}\boldsymbol{x}_t)
\end{aligned}
$$

Let denote $\overline{\boldsymbol{m}} = 1 - \boldsymbol{m}$

$$
\begin{aligned}
p_t\left[\boldsymbol{z}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right] &= p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right] \\
&= \int p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}, \boldsymbol{m}\boldsymbol{x}_t\right] p_t\left[\boldsymbol{m}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{z}_0 = \boldsymbol{y}\right] d(\boldsymbol{m}\boldsymbol{x}_t) \\
&= \mathbb{E}_{p_t[\boldsymbol{m}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}]}\Big(p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}, \boldsymbol{m}\boldsymbol{x}_t\right]\Big)
\end{aligned}
$$

Let denote $\overline{\boldsymbol{m}} = 1 - \boldsymbol{m}$

$$
\begin{aligned}
p_t\left[\boldsymbol{z}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right] &= p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right] \\
&= \int p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}, \boldsymbol{m}\boldsymbol{x}_t\right] p_t\left[\boldsymbol{m}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{z}_0 = \boldsymbol{y}\right] d(\boldsymbol{m}\boldsymbol{x}_t) \\
&= \mathbb{E}_{p_t[\boldsymbol{m}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}]}\left(p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}, \boldsymbol{m}\boldsymbol{x}_t\right]\right) \\
&\approx \mathbb{E}_{p_t[\boldsymbol{m}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}]}\left(p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{x}_t\right]\right)
\end{aligned}
$$

Let denote $\overline{\boldsymbol{m}} = 1 - \boldsymbol{m}$

$$
\begin{aligned}
p_t\left[\boldsymbol{z}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right] &= p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right] \\
&= \int p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}, \boldsymbol{m}\boldsymbol{x}_t\right] p_t\left[\boldsymbol{m}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{z}_0 = \boldsymbol{y}\right] d(\boldsymbol{m}\boldsymbol{x}_t) \\
&= \mathbb{E}_{p_t[\boldsymbol{m}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}]}\Big(p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}, \boldsymbol{m}\boldsymbol{x}_t\right]\Big) \\
&\approx \mathbb{E}_{p_t[\boldsymbol{m}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}]}\Big(p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{x}_t\right]\Big) \\
&\approx p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{x}_t\right]
\end{aligned}
$$

Let denote $\overline{\boldsymbol{m}} = 1 - \boldsymbol{m}$

$$
\begin{aligned}
p_t\left[\boldsymbol{z}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right] &= p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right] \\
&= \int p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}, \boldsymbol{m}\boldsymbol{x}_t\right] p_t\left[\boldsymbol{m}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{z}_0 = \boldsymbol{y}\right] d(\boldsymbol{m}\boldsymbol{x}_t) \\
&= \mathbb{E}_{p_t[\boldsymbol{m}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}]}\Big(p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}, \boldsymbol{m}\boldsymbol{x}_t\right]\Big) \\
&\approx \mathbb{E}_{p_t[\boldsymbol{m}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}]}\Big(p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{x}_t\right]\Big) \\
&\approx p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{x}_t\right]
\end{aligned}
$$

Now,

$$\log p_t\left(\boldsymbol{x}_t\right) = \log p_t\left(\overline{\boldsymbol{m}}\boldsymbol{x}_t, \boldsymbol{x}_t\right)$$

Let denote $\overline{\boldsymbol{m}} = 1 - \boldsymbol{m}$

$$
\begin{aligned}
p_t\left[\boldsymbol{z}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right] &= p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right] \\
&= \int p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}, \boldsymbol{m}\boldsymbol{x}_t\right] p_t\left[\boldsymbol{m}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{z}_0 = \boldsymbol{y}\right] d(\boldsymbol{m}\boldsymbol{x}_t) \\
&= \mathbb{E}_{p_t[\boldsymbol{m}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}]}\Big(p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}, \boldsymbol{m}\boldsymbol{x}_t\right]\Big) \\
&\approx \mathbb{E}_{p_t[\boldsymbol{m}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}]}\Big(p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{x}_t\right]\Big) \\
&\approx p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{x}_t\right]
\end{aligned}
$$

Now,

$$
\log p_t\left(\boldsymbol{x}_t\right) = \log p_t\left(\overline{\boldsymbol{m}}\boldsymbol{x}_t, \boldsymbol{x}_t\right) = \log p_t\left(\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{x}_t\right) + \log p_t\left(\boldsymbol{m}\boldsymbol{x}_t\right)
$$

Let denote $\overline{\boldsymbol{m}} = 1 - \boldsymbol{m}$

$$
\begin{aligned}
p_t\left[\boldsymbol{z}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right] &= p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right] \\
&= \int p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}, \boldsymbol{m}\boldsymbol{x}_t\right] p_t\left[\boldsymbol{m}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{z}_0 = \boldsymbol{y}\right] d(\boldsymbol{m}\boldsymbol{x}_t) \\
&= \mathbb{E}_{p_t[\boldsymbol{m}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}]}\Big(p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}, \boldsymbol{m}\boldsymbol{x}_t\right]\Big) \\
&\approx \mathbb{E}_{p_t[\boldsymbol{m}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}]}\Big(p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{x}_t\right]\Big) \\
&\approx p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{x}_t\right]
\end{aligned}
$$

Now,

$$
\log p_t\left(\boldsymbol{x}_t\right) = \log p_t\left(\overline{\boldsymbol{m}}\boldsymbol{x}_t, \boldsymbol{x}_t\right) = \log p_t\left(\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{x}_t\right) + \log p_t\left(\boldsymbol{m}\boldsymbol{x}_t\right)
$$

Thus,

$$
\nabla_{\boldsymbol{z}} \log p_t\left(\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{x}_t\right) = \nabla_{\overline{\boldsymbol{m}}\boldsymbol{x}} \log p_t\left(\boldsymbol{x}_t\right)
$$

Let denote $\overline{\boldsymbol{m}} = 1 - \boldsymbol{m}$

$$
\begin{aligned}
p_t\left[\boldsymbol{z}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right] &= p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right] \\
&= \int p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}, \boldsymbol{m}\boldsymbol{x}_t\right] p_t\left[\boldsymbol{m}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{z}_0 = \boldsymbol{y}\right] d(\boldsymbol{m}\boldsymbol{x}_t) \\
&= \mathbb{E}_{p_t[\boldsymbol{m}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}]}\Big(p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}, \boldsymbol{m}\boldsymbol{x}_t\right]\Big) \\
&\approx \mathbb{E}_{p_t[\boldsymbol{m}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}]}\Big(p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{x}_t\right]\Big) \\
&\approx p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{x}_t\right]
\end{aligned}
$$

Now,

$$\log p_t\left(\boldsymbol{x}_t\right) = \log p_t\left(\overline{\boldsymbol{m}}\boldsymbol{x}_t, \boldsymbol{x}_t\right) = \log p_t\left(\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{x}_t\right) + \log p_t\left(\boldsymbol{m}\boldsymbol{x}_t\right)$$

Thus,

$$\nabla_{\boldsymbol{z}} \log p_t\left(\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{x}_t\right) = \nabla_{\overline{\boldsymbol{m}}\boldsymbol{x}} \log p_t\left(\boldsymbol{x}_t\right) = \overline{\boldsymbol{m}} \nabla_{\boldsymbol{x}} \log p_t\left(\boldsymbol{x}_t\right)$$

Let denote $\overline{\boldsymbol{m}} = 1 - \boldsymbol{m}$

$$
\begin{aligned}
p_t\left[\boldsymbol{z}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right] &= p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}\right] \\
&= \int p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}, \boldsymbol{m}\boldsymbol{x}_t\right] p_t\left[\boldsymbol{m}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{z}_0 = \boldsymbol{y}\right] d(\boldsymbol{m}\boldsymbol{x}_t) \\
&= \mathbb{E}_{p_t[\boldsymbol{m}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}]}\Big(p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}, \boldsymbol{m}\boldsymbol{x}_t\right]\Big) \\
&\approx \mathbb{E}_{p_t[\boldsymbol{m}\boldsymbol{x}_t \mid m\boldsymbol{x}_0 = \boldsymbol{y}]}\Big(p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{x}_t\right]\Big) \\
&\approx p_t\left[\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{x}_t\right]
\end{aligned}
$$

Now,

$$\log p_t\left(\boldsymbol{x}_t\right) = \log p_t\left(\overline{\boldsymbol{m}}\boldsymbol{x}_t, \boldsymbol{x}_t\right) = \log p_t\left(\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{x}_t\right) + \log p_t\left(\boldsymbol{m}\boldsymbol{x}_t\right)$$

Thus,

$$\nabla_{\boldsymbol{z}} \log p_t\left(\overline{\boldsymbol{m}}\boldsymbol{x}_t \mid \boldsymbol{m}\boldsymbol{x}_t\right) = \nabla_{\overline{\boldsymbol{m}}\boldsymbol{x}} \log p_t\left(\boldsymbol{x}_t\right) = \overline{\boldsymbol{m}} \nabla_{\boldsymbol{x}} \log p_t\left(\boldsymbol{x}_t\right)$$

Consequently, the approached backward SDE becomes:

$$d\boldsymbol{z}_t = \left[-\frac{1}{2}\overline{\boldsymbol{m}}\beta(t)\boldsymbol{x}_t - \overline{\boldsymbol{m}}\beta(t)\nabla_{\boldsymbol{x}} \log p_t\left(\boldsymbol{x}_t\right)\right] dt + \sqrt{\beta(t)} d\overline{\boldsymbol{w}}_t$$

How to build a conditional diffusion model ?

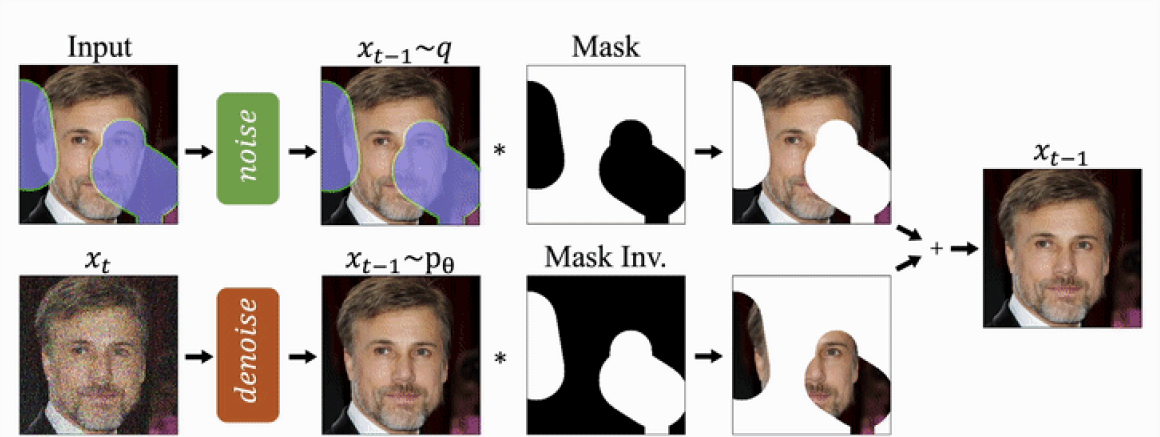


Image extracted from [Lugmayr et al., 2022][28]

[28] Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., & Van Gool, L. (2022). RePaint: Inpainting using denoising diffusion probabilistic models. *RePaint*

Pseudo-code

1. **Input:** A masked image $\boldsymbol{m}\boldsymbol{x} = \boldsymbol{y}$
2. Sample $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
3. For $t = T, \dots, 1$
4. Sample $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
5. $\boldsymbol{x}_{t-1}^{\text{known}} = \sqrt{\overline{\alpha}_t}\boldsymbol{y} + (1 - \overline{\alpha}_t)\boldsymbol{\varepsilon}$
6. Sample $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ if $t > 1$ else $\boldsymbol{z} = \mathbf{0}$
7. $\boldsymbol{x}_{t-1}^{\text{unknown}} = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}} \boldsymbol{\varepsilon}_\theta(\boldsymbol{x}_t, t) \right) + \sigma_t \boldsymbol{z}$
8. $\boldsymbol{x}_{t-1} = \boldsymbol{m} \odot \boldsymbol{x}_{t-1}^{\text{known}} + (1 - \boldsymbol{m}) \odot \boldsymbol{x}_{t-1}^{\text{unknown}}$
9. **Output:** $\boldsymbol{x}_0$

- $\boldsymbol{x}_{t-1}^{\text{known}}$ is resampled at each step knowing:

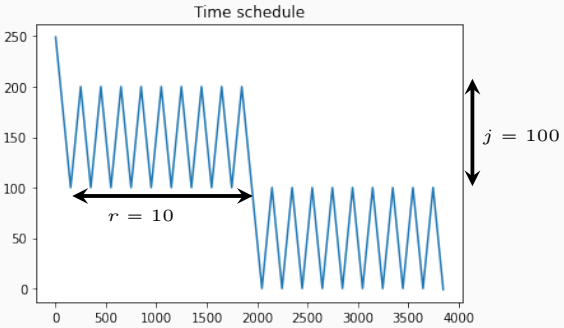
$$\boldsymbol{x}_{t-1} \sim \mathcal{N}(\sqrt{\overline{\alpha}_t}\boldsymbol{x}_0, (1 - \overline{\alpha}_t)\boldsymbol{I})$$

Let's try it

Let's try it

Let's try it

Time schedule



1. For each group of times of size j
2. Do r times:
3. For $s = T, \dots, T - j$
4. Do the backward from s to $s - 1$
5. For $s = T - j, \dots, T$
6. Do the forward from s to $s + 1$
7. For $s = T, \dots, T - j$
8. Do the backward from s to $s - 1$

Learned variance

Problem: From 1000 steps to 18820 steps with $j = r = 10$

In DDPM, there are two choices for the variance schedule of the backward process: β_t of $\tilde{\beta}_t$. Nichol and Dhariwal, 2021 proposes to learn the variance as

$$\Sigma_\theta(\boldsymbol{x}, t) = \exp\left[v \log(\beta_t) + (1 - v) \log(\tilde{\beta}_t)\right]$$

This reduces the number of steps from 1000 steps to 250 steps.

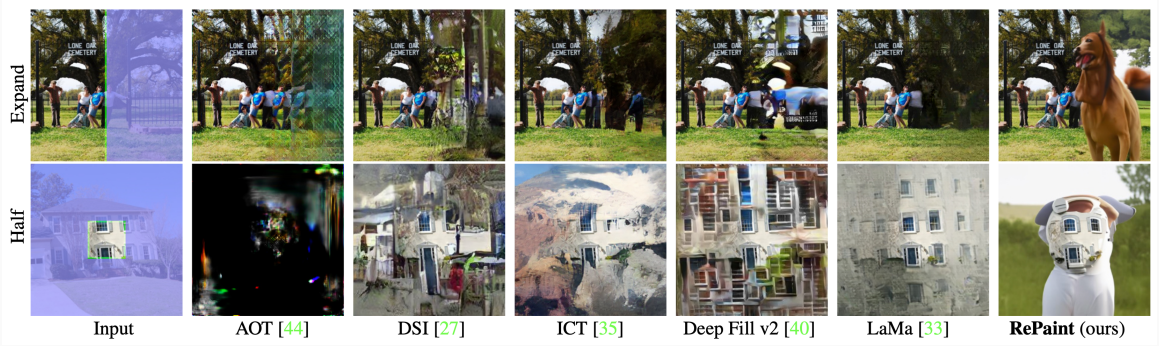
Now: From 250 steps to 4570 steps with $j = r = 10$.

"Justification" of the algorithm

From the article: "However, the sampling of the known pixels using is performed without considering the generated parts of the image, which introduces disharmony. Although the model tries to harmonize the image again in every step, it can never fully converge because the same issue occurs in the next step. Moreover, in each reverse step, the maximum change to an image declines due to the variance schedule of β_t. Thus, the method cannot correct mistakes that lead to disharmonious boundaries in the subsequent steps due to restricted flexibility. **As a consequence, the model needs more time to harmonize the conditional information x^{known} with the generated information x^{unknown} in one step before advancing to the next denoising step.**"

Let's try it

Fails


Expand
Half
Input
AOT [44]
DSI [27]
ICT [35]
Deep Fill v2 [40]
LaMa [33]
RePaint (ours)

Conclusion

- Free learning method is an interesting approach.
- Probably far from the true conditional distribution [Trippe et al., 2023][29] .
- An other approach with manifold constraints [Chung et al., 2022][30].

[29] Trippe, B. L., Yim, J., Tischer, D., Baker, D., Broderick, T., Barzilay, R., & Jaakkola, T. S. (2023). Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. *International Conference on Learning Representations*

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

The end

Thank you for your attention !

References

Bishop, C. M. (2006). *Pattern recognition and machine learning* (Vol. 4). Springer.

Bortoli, V. D. (2022). Convergence of denoising diffusion models under the manifold hypothesis. *Transactions on Machine Learning Research*.

Choi, J., Kim, S., Jeong, Y., Gwon, Y., & Yoon, S. (2021). ILVR: Conditioning method for denoising diffusion probabilistic models. *ILVR*.

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

De Bortoli, V., Thornton, J., Heng, J., & Doucet, A. (2021). Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*.

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

Feller, W. (1949). On the theory of stochastic processes, with particular reference to applications. *Proceedings of the [First] Berkeley Symposium on Mathematical Statistics and Probability*.

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*.

Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., & Van Gool, L. (2022). RePaint: Inpainting using denoising diffusion probabilistic models. *RePaint*.

Nichol, A. Q., & Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th international conference on machine learning* (pp. 8162–8171). PMLR. https://proceedings.mlr.press/v139/nichol21a.html

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

Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., & Norouzi, M. (2022). Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., & Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics [ISSN: 1938-7228]. *Proceedings of the 32nd International Conference on Machine Learning*, 2256–2265. Retrieved 2022-12-01, from https://proceedings.mlr.press/v37/sohl-dickstein15.html

Song, J., Meng, C., & Ermon, S. (2021). Denoising diffusion implicit models. *International Conference on Learning Representations*.

Song, Y., & Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*.

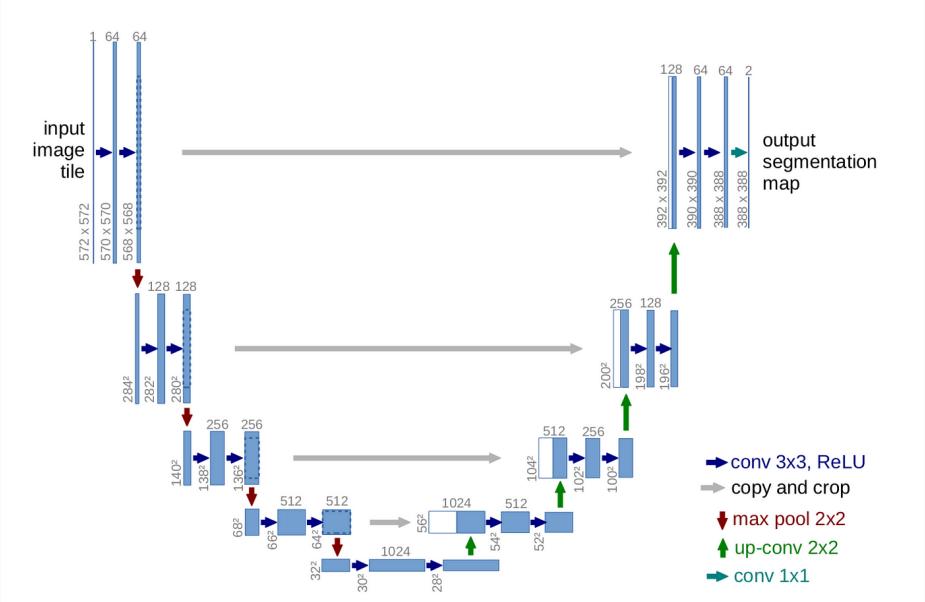
Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2023). Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations*.

Trippe, B. L., Yim, J., Tischer, D., Baker, D., Broderick, T., Barzilay, R., & Jaakkola, T. S. (2023). Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. *International Conference on Learning Representations*.

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

Zhang, Y., Liu, W., Chen, Z., Li, K., & Wang, J. (2021). On the properties of kullback-leibler divergence between gaussians. *CoRR*, *abs/2102.05485*. https://arxiv.org/abs/2102.05485

Appendix


input image tile
572 x 572
570 x 570
568 x 568
1
64
64
128
128
284²
282²
280²
140²
138²
136²
256
256
68²
66²
64²
512
512
32²
30²
1024
28²
56²
54²
52²
1024
512
104²
102²
100²
512
256
200²
198²
196²
256
128
392 x 392
390 x 390
388 x 388
388 x 388
128
64
64
2
output segmentation map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

We can read in Ho et al., 2020, "Experimentally, both $\sigma_t^2 = \beta_t$ and $\sigma_t^2 = \tilde{\beta}_t$ had similar results. The first choice is optimal for $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and the second is optimal for $\boldsymbol{x}_0$ deterministically set to one point. These are the two extreme choices corresponding to upper and lower bounds on reverse process entropy for data with coordinatewise unit variance Sohl-Dickstein et al., 2015."

1. $\widehat{\boldsymbol{w}}_t = \boldsymbol{w}_t - \boldsymbol{w}_1 - \int_t^1 \nabla \log p_s(\boldsymbol{x}_s) ds$ is a backward Brownian motion adapted to the filtration $\sigma(\boldsymbol{x}_t \cup \{\boldsymbol{w}_s - w_t, t \leqslant s \leqslant 1\})$ because it is a local backward martingale.
2. Rewriting of the SDE in the Stratonovich sense.
3. Insertion of $\widehat{\boldsymbol{w}}_t$ in the SDE.
4. Change of variable.