

Diffusion Models



Tweedie's formula

Let σ be a constant and consider a random variable \mathbf{X} with probability density function $f_{\mathbf{X}}$ as well as a random Gaussian noise $\mathcal{E} \sim \mathcal{N}(0, 1)$ that is independent from \mathbf{X} . If $\mathbf{Y} = \mathbf{X} + \sigma\mathcal{E}$ then

$$\mathbb{E}[X|Y = y] = y + \sigma^2 \nabla_y \log f_Y(y) \quad \mathbb{E}[\mathcal{E}|Y = y] = -\sigma \nabla_y \log f_Y(y).$$

► Proof

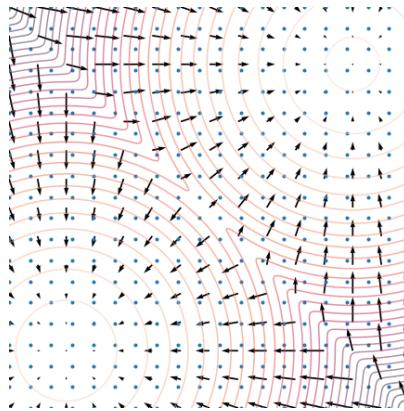
From Tweedie's formula, ϵ is estimated to be $-\sigma \nabla_y \log f_Y(y)$.

Sampler Langevin dynamics:

$$y_{k+1} = y_k + \delta_k \nabla_y \log f_Y(y_k) + \sqrt{2\delta_k} w_k$$

where $w_k \sim \mathcal{N}(0, 1)$

[Image source.](#)



Score matching ↵

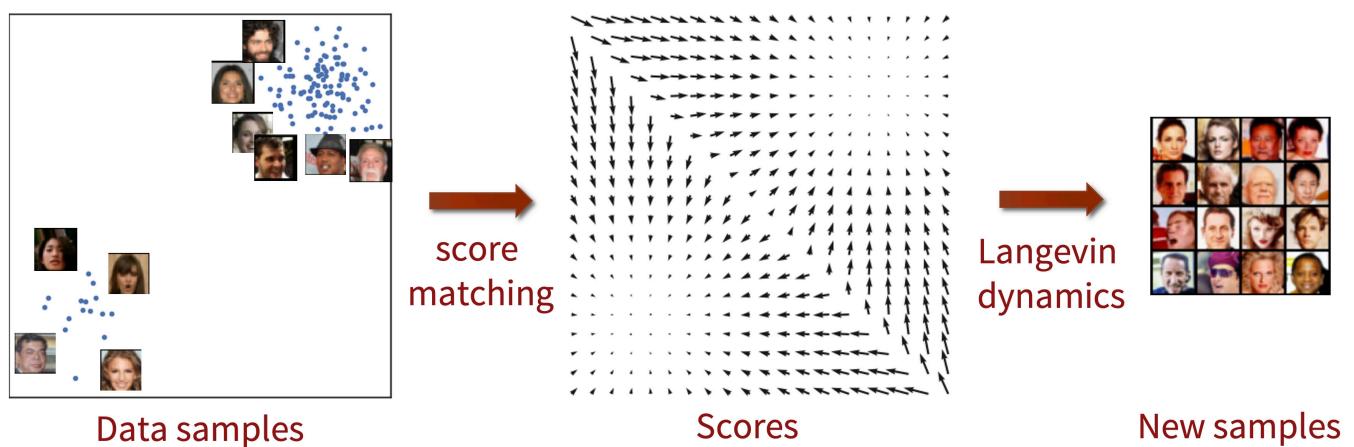
Diffusion models are also known as *energy-based models* and then *score-matching*.

For **fixed** σ , the matching is $\epsilon_\theta(y) \approx \mathbb{E}[\mathcal{E}|X + \sigma\mathcal{E} = y] = -\sigma\nabla_y \log f_{X+\sigma\mathcal{E}}(y)$.

Training: sample x / pick x in dataset, sample ε , update θ to minimize (e.g., using gradient descent), the loss:

$$\mathbb{E}[\|\epsilon_\theta(X + \sigma\mathcal{E}) - \mathcal{E}\|^2]$$

Image source.

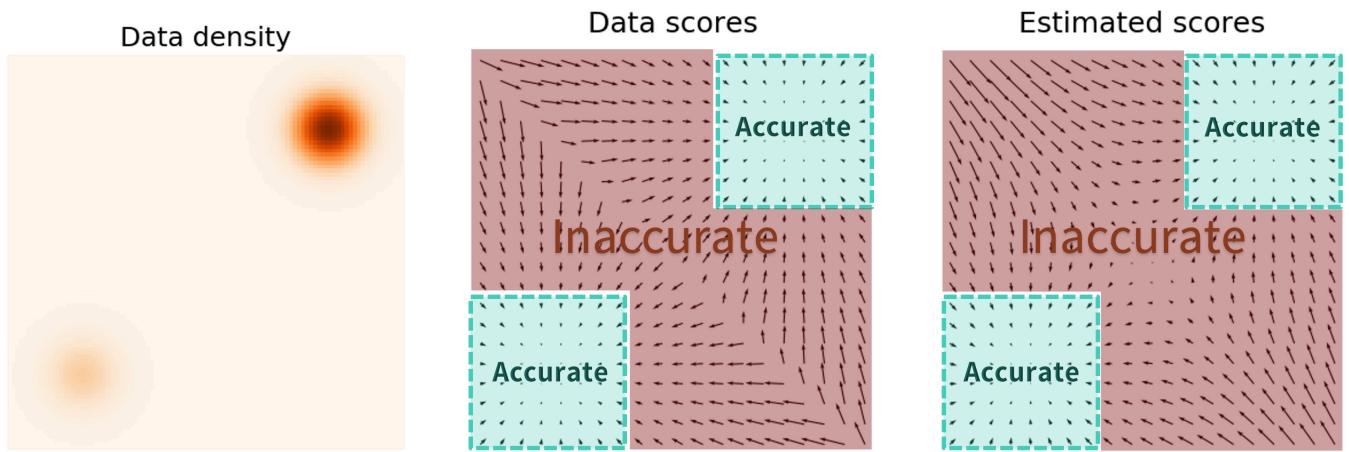


Issue with small variance

If σ is too small then the support of $\mathbf{X} + \sigma\mathcal{E}$ may not cover the whole state space \rightarrow inaccurate $\epsilon_\theta(\mathbf{y})$ in these parts. More formally, the loss is the Fisher divergence:

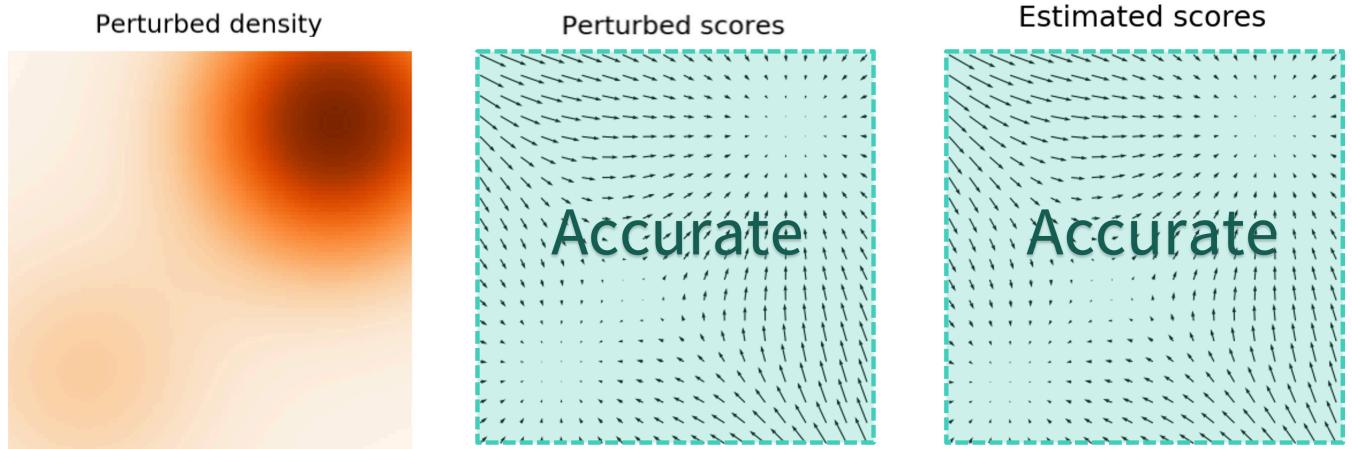
$$\mathbb{E}[\|\epsilon_\theta(\mathbf{y}) + \sigma\nabla_y \log f_Y(\mathbf{y})\|^2] = \int_y f_Y(\mathbf{y}) \|\epsilon_\theta(\mathbf{y}) + \sigma\nabla_y \log f_Y(\mathbf{y})\|^2 d\mathbf{y}$$

so it is inaccurate for \mathbf{y} such that $f_Y(\mathbf{y})$ is too small. [Image source](#).



Issue with large variance

If σ is too large then $\epsilon_\theta(y) \approx -\sigma \nabla_y \log f_{X+\sigma\mathcal{E}}(y)$ everywhere but the distribution $Y \sim X + \sigma\mathcal{E}$ is too noisy, less specific to X (small signal to noise ratio). [Image source](#).



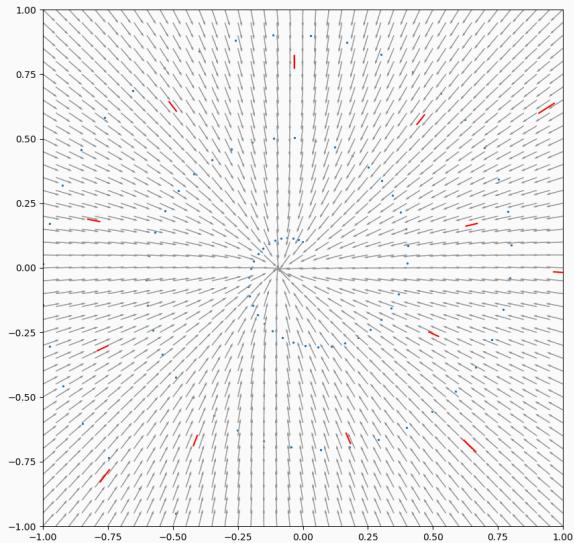
Variance-dependent score ↵

The matching is $\epsilon_\theta(y, \sigma) \approx -\sigma \nabla_y \log f_{X+\sigma\mathcal{E}}(y)$

Training: sample x / pick x in dataset, sample σ, ε , update θ to minimize (e.g., using gradient descent), the loss:

$$\mathbb{E}[\|\epsilon_\theta(X + \sigma\mathcal{E}, \sigma) - \mathcal{E}\|^2]$$

Animation generated with [smallldiffusion](#) using a deterministic (i.e. $w_k = 0$) sampler.



Sampling ↳

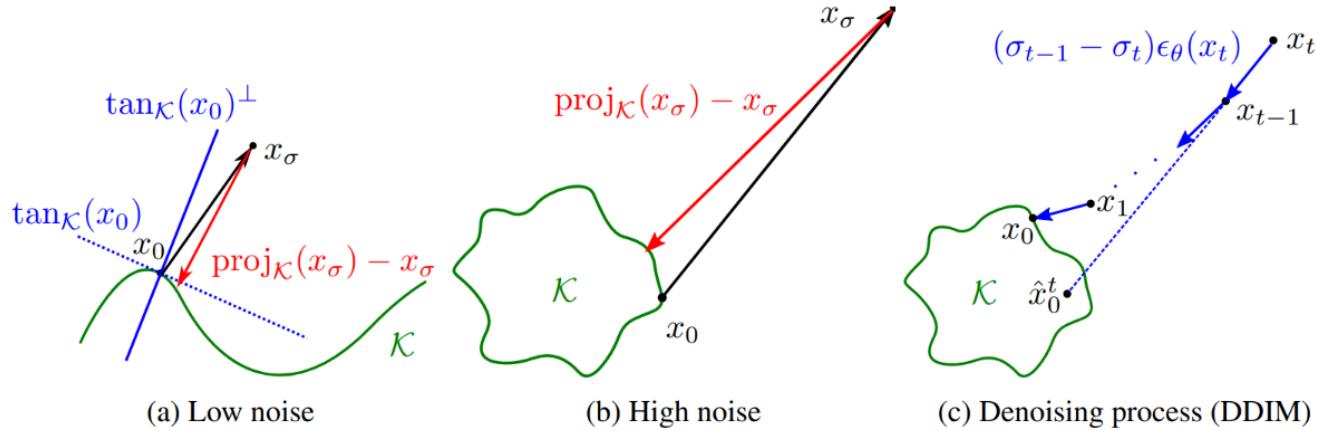


Image from [PY24; Figure 1]. Deterministic DDIM : move from a variance estimate σ_t to σ_{t-1} :

$$\begin{aligned}
 X_t &= X_0 + \sigma_t \mathcal{E} \\
 \mathbb{E}[X_{t-1}|X_t = x_t] &= \mathbb{E}[X_0|X_t = x_t] + \sigma_{t-1} \mathbb{E}[\mathcal{E}|X_t = x_t] \\
 &= \mathbb{E}[X_t|X_t = x_t] - \sigma_t \mathbb{E}[\mathcal{E}|X_t = x_t] + \sigma_{t-1} \mathbb{E}[\mathcal{E}|X_t = x_t] \\
 &= x_t + (\sigma_{t-1} - \sigma_t) \mathbb{E}[\mathcal{E}|X_t = x_t]
 \end{aligned}$$

Denoising with randomness ↗

Given $0 \leq \mu < 1$, pick $\sigma_{t'}$ such that $\sigma_{t-1} = \sigma_t^\mu \sigma_{t'}^{1-\mu}$ we have the following sampler of [PY24]:

$$x_{t-1} = x_t + (\sigma_{t'} - \sigma_t) \epsilon_\theta(x_t, \sigma_t) + \eta w_t \quad w_t \sim \mathcal{N}(0, I)$$

For $\mu = 0$, $\sigma_{t'} = \sigma_{t-1}$ and we recover deterministic DDIM. For $\mu = 1/2$, we have DDPM of [H]A20]

► What is the order between σ_{t-1} , σ_t and $\sigma_{t'}$?

► What should η be ?

[H]A20] J. Ho, A. Jain and P. Abbeel. *Denoising Diffusion Probabilistic Models*. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20* (Curran Associates Inc., Red Hook, NY, USA, Dec 2020); pp. 6840–6851. Accessed on Nov 26, 2024.

[PY24] F. Permenter and C. Yuan. *Interpreting and Improving Diffusion Models from an Optimization Perspective* (Jun 2024), arXiv:2306.04848. Accessed on Nov 26, 2024.

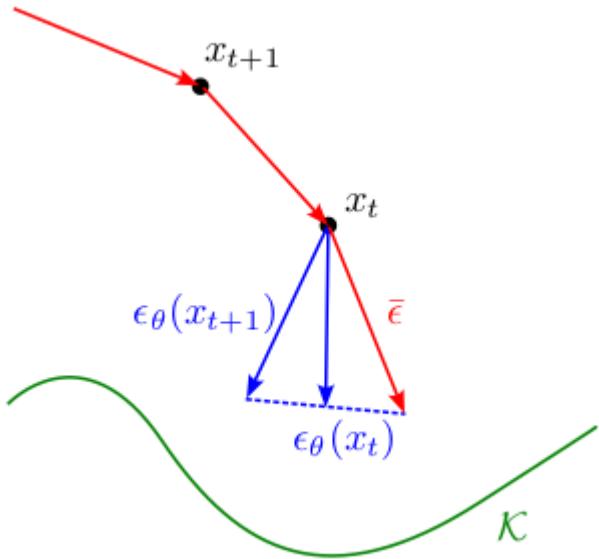
Acceleration ↵

As $\epsilon_\theta(x_t, \sigma_t)$ is more accurate than $\epsilon_\theta(x_{t+1}, \sigma_{t+1})$, [PY24; Section 5] suggests accelerating the convergence by correcting part of the previous step. That is, we take:

$$\bar{\epsilon}_t = \gamma \epsilon_\theta(x_t, \sigma_t) + (1 - \gamma) \epsilon_\theta(x_{t+1}, \sigma_{t+1})$$

with $\gamma > 1$.

Image is [PY24; Figure 3].



[PY24] F. Permenter and C. Yuan. [Interpreting and Improving Diffusion Models from an Optimization Perspective](#) (Jun 2024), arXiv:2306.04848. Accessed on Nov 26, 2024.

Auto-Encoder

Find encoder E and decoder D that minimize the loss:

$$\mathbb{E}[\|X - D(E(X))\|_2^2]$$

The *code* (aka *latent variable*) $z = E(x)$ typically has a smaller size compared to x to force the model to only keep essential features.

► **What is the solution if E and D were linear (i.e. matrices) ?**

Variational Auto-Encoder

We add artificial noise to improve learning:

$$\mathbb{E}[\|X - D(E_\mu(X) + \mathcal{E} \odot E_\sigma(X))\|_2^2]$$

An insightful way to model this is as follows.

- Assume our dataset come from a distribution X that is obtained from a latent space Z using a decoder $X = D(Z)$.
- If we knew the latent variable z , we would search for a decoder D that maximizes $\mathbb{E}[\log(f_{X|Z}(x|z))]$.
- Since we do not know z , we make an estimate of y using an encoder $Y = E(X)$.
- The validity of these encoder and decoder models can now be measured with a Maximum Likelihood Estimator that searches for the encoder and decoder models that maximize $\mathbb{E}[\log(f_X(X))]$.

Evidence Lower BOund (ELBO) ↗

For any random variables X , Y and Z , we have

$$\log(f_X(x)) = D_{\text{KL}}((Y|X=x) \parallel (Z|X=x)) + \mathcal{L}(x)$$

where the *evidence lower bound* [KW13]

$$\mathcal{L}(x) = -D_{\text{KL}}((Y|X=x) \parallel Z) + \mathbb{E}[\log(f_{X|Z}(x|Y))|X=x]$$

► Proof

$\mathcal{L}(x)$ is a **lower bound** to $\log(f_X(x))$ as the Kullback-Leibler divergence $D_{\text{KL}}((Y|X=x) \parallel (Z|X=x))$ is always nonnegative.

In the context of VAEs, Z represents the actual latent variable and Y represents the estimated random variables, so $D_{\text{KL}}((Y|X=x) \parallel (Z|X=x))$ is a measure of the error made by our estimator.

[KW13] D. P. Kingma and M. Welling. [Auto-Encoding Variational Bayes](#) (Dec 2013). Accessed on Nov 26, 2024.

Gaussian ELBO ↵

Consider two deterministic functions $E_\mu, E_\sigma : \mathbb{R}^n \rightarrow \mathbb{R}^r$ and $D_\mu, D_\sigma : \mathbb{R}^r \rightarrow \mathbb{R}^n$. Suppose $X = D_\mu(Z) + \mathcal{E}_1 \odot D_\sigma(Z)$ and $Y = E_\mu(X) + \mathcal{E}_2 \odot E_\sigma(X)$ with $\mathcal{E}_1, \mathcal{E}_2 \sim \mathcal{N}(0, I)$. We have (see [KW13; Appendix B] for a proof):

$$2D_{\text{KL}}((Y|X=x) \parallel Z) = \|E_\mu(X)\|_2^2 + \|E_\sigma(X)\|_2^2 - r - \sum_{i=1}^r \log((E_\sigma(X))_i^2)$$

For the second part of the ELBO, we have

$$\begin{aligned} & \mathbb{E}[\log(f_{X|Z}(x|Y))] \\ &= \mathbb{E}[\log(f_{X|Z}(x|E_\mu(x) + \mathcal{E}_2 \odot E_\sigma(x)))] \\ &= \mathbb{E}[\log(f_{\mathcal{E}_1}(\text{Diag}(D_\sigma(E_\mu(x) + \mathcal{E}_2 \odot E_\sigma(x)))^{-1}(x - D_\mu(E_\mu(x) + \mathcal{E}_2 \odot E_\sigma(x)))))] \\ &= -\frac{\log(2\pi)}{2} + \mathbb{E}[\|\text{Diag}(D_\sigma(E_\mu(x) + \mathcal{E}_2 \odot E_\sigma(x)))^{-1}(x - D_\mu(E_\mu(x) + \mathcal{E}_2 \odot E_\sigma(x)))\|_2^2]. \end{aligned}$$

[KW13] D. P. Kingma and M. Welling. *Auto-Encoding Variational Bayes* (Dec 2013). Accessed on Nov 26, 2024.

Monte-Carlo sampling ↵

This can be approximated using Monte-Carlo given L samples $\epsilon_1, \dots, \epsilon_L$ from the distribution $\mathcal{N}(0, I)$ as

$$\mathbb{E}[\log(f_{X|Z}(x|Y))] \approx \frac{1}{L} \sum_{i=1}^L \log(f_{X|Z}(x|E_\mu(x) + \epsilon_i \odot E_\sigma(x))).$$

In the simpler case where $D_\sigma(z) = \mathbf{1}$, we recognize the classical L2 norm:

$$\mathbb{E}[\log(f_{X|Z}(x|Y))] \approx -\frac{\log(2\pi)}{2} + \frac{1}{L} \sum_{i=1}^L \|x - D_\mu(E_\mu(x) + \epsilon_i \odot E_\sigma(x))\|_2^2.$$

Variational AutoEncoders (VAEs) ↵

- We want to learn the distribution of our data represented by the random variable \mathbf{X} .
- The encoder maps a data point \mathbf{x} to a Gaussian distribution $\mathbf{Y} \sim \mathcal{N}(E_\mu(\mathbf{x}), E_\sigma(\mathbf{x}))$ over the latent space
- The decoder maps a latent variable $\mathbf{z} \sim \mathbf{Z}$ to a the Gaussian distribution $\mathcal{N}(D_\mu(\mathbf{z}), D_\sigma(\mathbf{z}))$

The Maximum Likelihood Estimator (MLE) maximizes the following sum over our datapoints \mathbf{x} with its ELBO:

$$\sum_{\mathbf{x}} \log(f_{\mathbf{X}}(\mathbf{x})) \geq \sum_{\mathbf{x}} -D_{\text{KL}}((\mathbf{Y}|X = \mathbf{x}) \parallel \mathbf{Z}) + \mathbb{E}[\log(f_{\mathbf{X}|Z}(\mathbf{x}|Y))]$$

So the MLE minimizes the loss

$$-\mathbb{E}[\log(f_{\mathbf{X}|Z}(\mathbf{x}|Y))].$$

with the KL-regularizer

$$D_{\text{KL}}((\mathbf{Y}|X = \mathbf{x}) \parallel \mathbf{Z})$$

Denoising Auto-Encoder

Auto-Encoder	$D(E(X))$
Variational Auto-Encoder	$D(E(X) + \mathcal{E})$
Denoising Auto-Encoder	$D(E(X + \sigma\mathcal{E}))$

The goal of the diffusion model is, given a noisy image $\mathbf{Y} = \mathbf{X} + \sigma\mathcal{E}$ and σ , to find the noise \mathcal{E} .

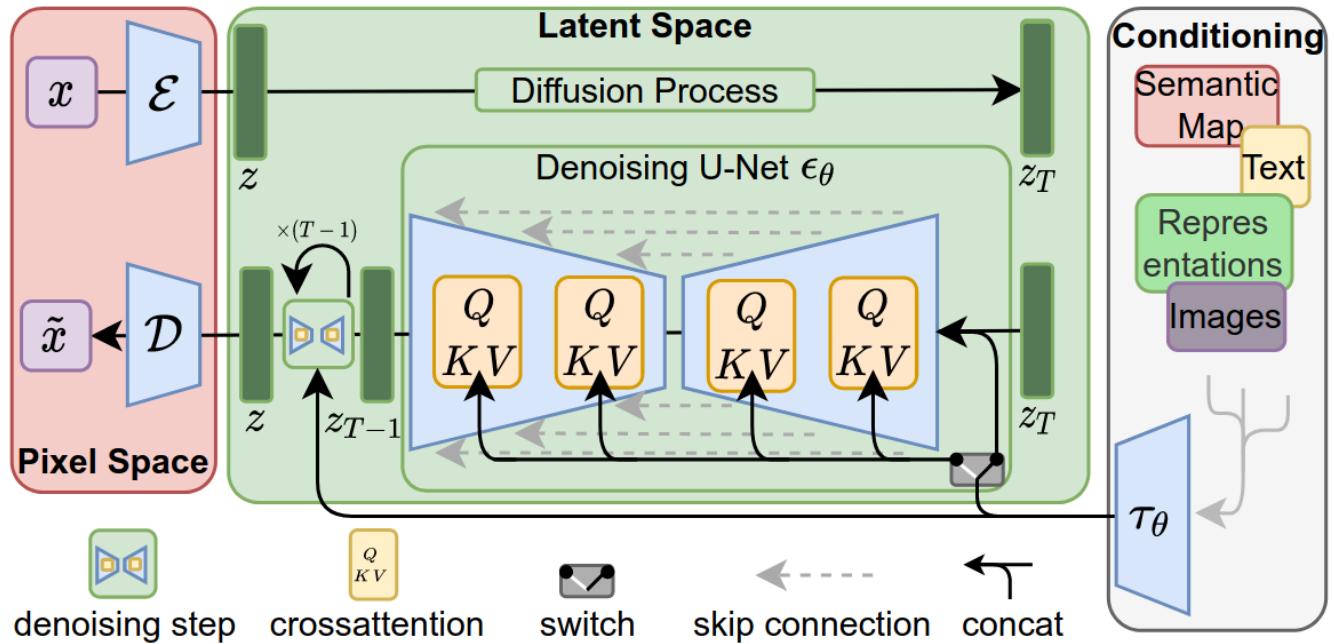
The denoising Auto-Encoder instead attempts to find the original image \mathbf{X} . At the limit $\sigma \rightarrow 0$, the denoising Auto-Encoder $D(E(X + \sigma\mathcal{E}))$ is a classical Auto-Encoder, hence the name.

To train the denoising Auto-Encoder, we can use the Evidence Lower-Bound with $\mathbf{Y} = \mathbf{X} + \sigma\mathcal{E}$ and \mathbf{Z} such that $\mathbf{X} = D(E(\mathbf{Z}))$ [HJA20]:

$$-\log(f_X(x)) \leq \mathbb{E}[-\log(f_{X|Z}(x|Y))] + D((Y|X=x) \parallel Z)$$

[HJA20] J. Ho, A. Jain and P. Abbeel. *Denoising Diffusion Probabilistic Models*. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20* (Curran Associates Inc., Red Hook, NY, USA, Dec 2020); pp. 6840–6851. Accessed on Nov 26, 2024.

Conditioned diffusion



Source: [RBL+22; Figure 3]

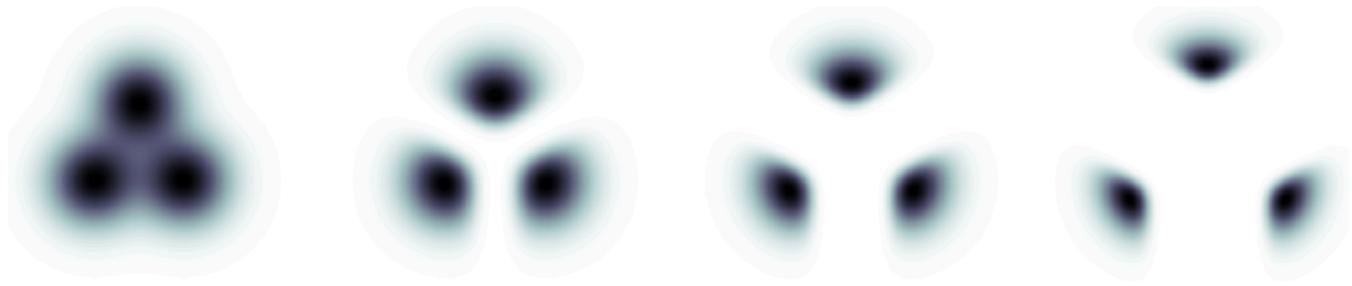
[RBL+22] R. Rombach, A. Blattmann, D. Lorenz et al. *High-Resolution Image Synthesis With Latent Diffusion Models*. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022); pp. 10684–10695. Accessed on Aug 26, 2024.

Classifier-Free Guidance

Improvement for conditioned diffusion for multi-modal distributions: use classifier [DN24]. **Issue:** need to train a classifier...

This classifier-guided strategy was replaced in [HS22] by a simpler trick: Train both a conditioned (with condition τ) and an unconditioned diffusion model, and combine them with:

$$\bar{\epsilon}_t = \lambda \epsilon_{\theta}(x_t, \sigma_t, \tau) + (1 - \lambda) \epsilon_{\theta}(x_t, \sigma_t) \quad \text{with } \lambda > 1$$



[DN24] P. Dhariwal and A. Nichol. *Diffusion Models Beat GANs on Image Synthesis*. In: *Proceedings of the 35th International Conference on Neural Information Processing Systems, NIPS '21* (Curran Associates Inc., Red Hook, NY, USA, Jun 2024); pp. 8780–8794. Accessed on Nov 26, 2024.

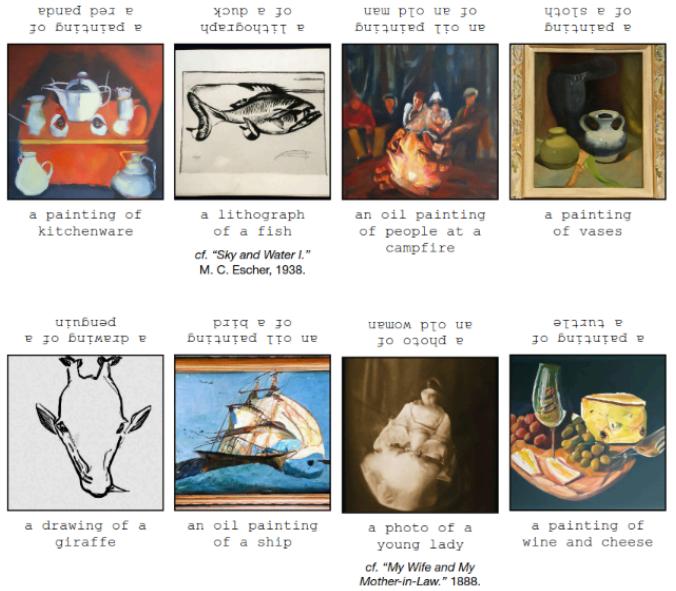
[HS22] J. Ho and T. Salimans. *Classifier-Free Diffusion Guidance* (Jul 2022), arXiv:2207.12598. Accessed on Nov 26, 2024.

Optical illusions ↗

It was shown in [BLL+23] how to train a diffusion model to generate illusions. Surprisingly, [GPO24] showed that you don't need a specialized model and you can use a pre-trained diffusion model.

Given N transformations v_1, \dots, v_N [GPO24; Equation (2)]:

$$\bar{\epsilon}_t = \frac{1}{N} \sum_{i=1}^N v_i^{-1}(\epsilon_\theta(v_i(x_t), \sigma_t, \tau))$$



[BLL+23] R. Burgert, X. Li, A. Leite *et al.* [Diffusion Illusions: Hiding Images in Plain Sight](#) (Dec 2023), arXiv:2312.03817. Accessed on Nov 27, 2024.

[GPO24] D. Geng, I. Park and A. Owens. [Visual Anagrams: Generating Multi-View Optical Illusions with Diffusion Models](#) (Apr 2024), arXiv:2311.17919. Accessed on Nov 27, 2024.

Utils ↗

```
1 using PlutoUI, DataFrames, PrettyTables, LinearAlgebra, Luxor, LaTeXStrings,  
  MathTeXEngine
```

```
1 import DocumenterCitations, CSV, Logging
```

```
qa (generic function with 2 methods)
```

```
1 include("utils.jl")
```

```
biblio =
```

```
► CitationBibliography("/home/runner/work/LINMA2472/LINMA2472/Lectures/biblio.bib", AlphaSt
```

```
1 biblio = load_biblio!()
```

ⓘ Loading bibliography from `/home/runner/work/LINMA2472/LINMA2472/Lectures/biblio.bib`...

ⓘ Loading completed.

```
cite (generic function with 1 method)
```

```
1 cite(args...) = bibcite(biblio, args...)
```

```
refs (generic function with 1 method)
```

```
1 refs(args...) = bibrefs(biblio, args...)
```