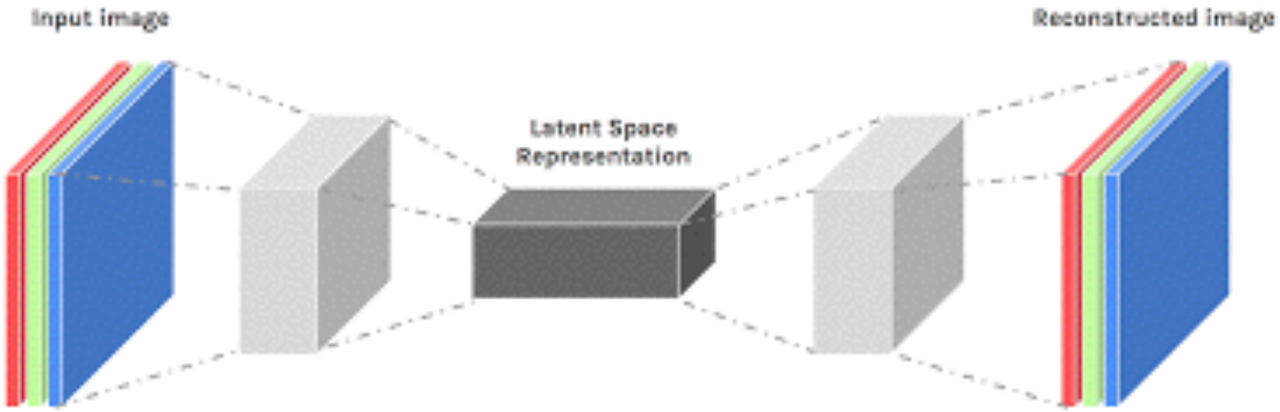
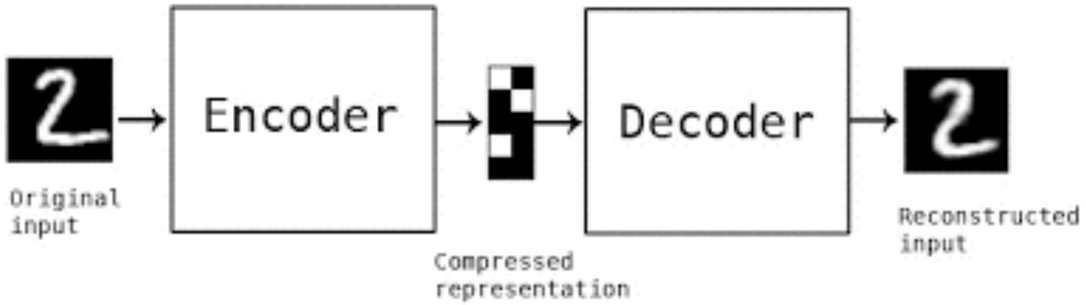
Achitectures

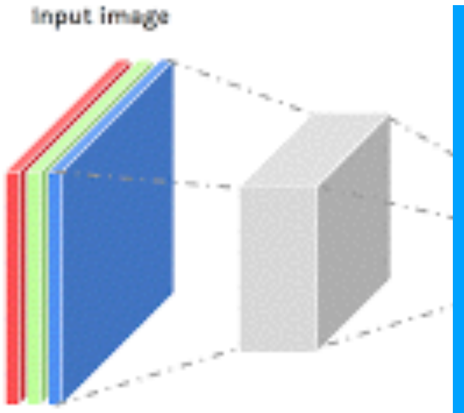
auto-encoders

GANs



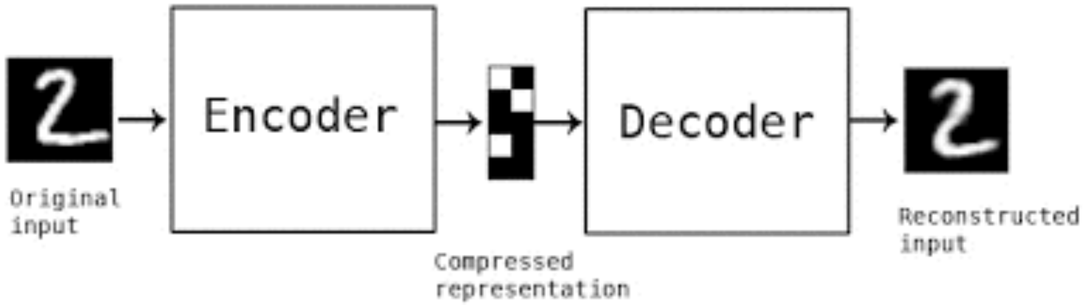


Learn representation without labels



Cute here and add a single layer!

Pre-training without labels!

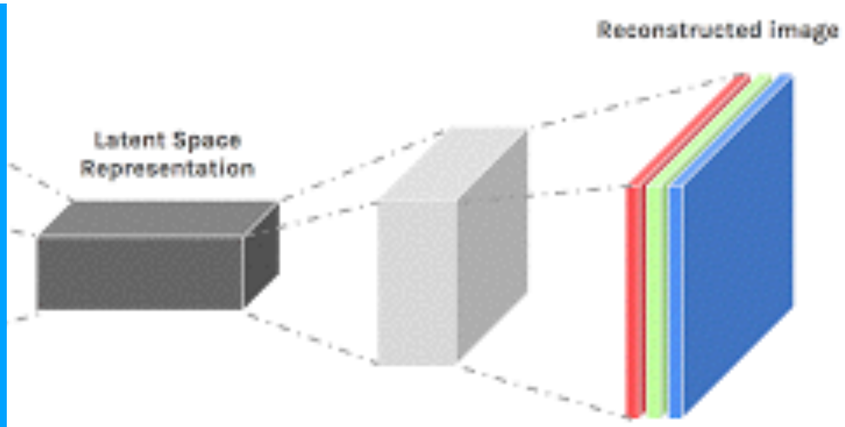


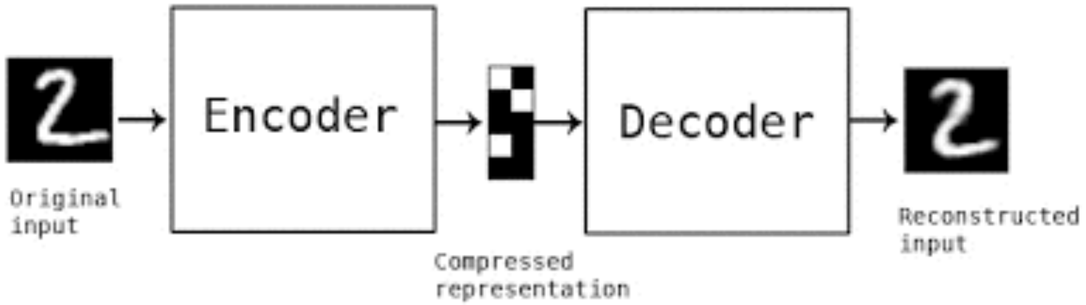
Learn representation without labels

Fundamental in the early day of deep learning for pre-training

Cute here!

Now you have a generative model



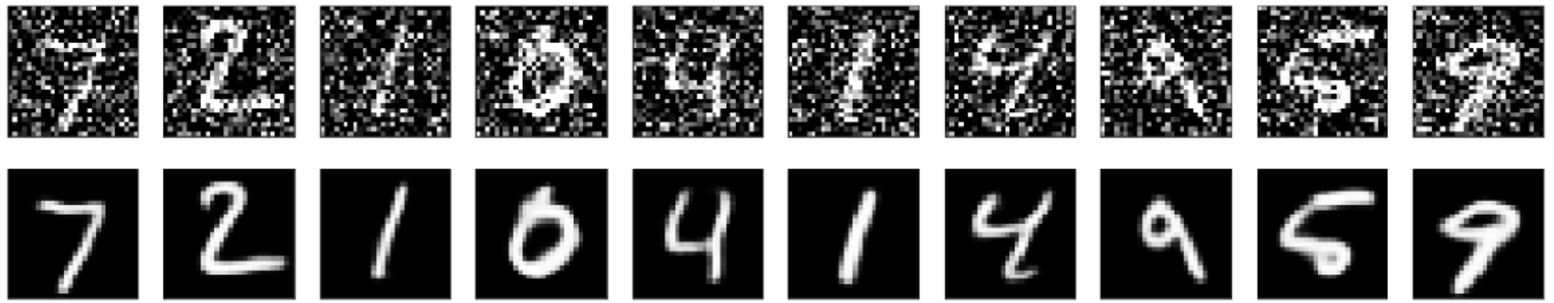


Learn representation without labels

Fundamental in the early day of deep learning for pre-training

Denoising auto-encoder

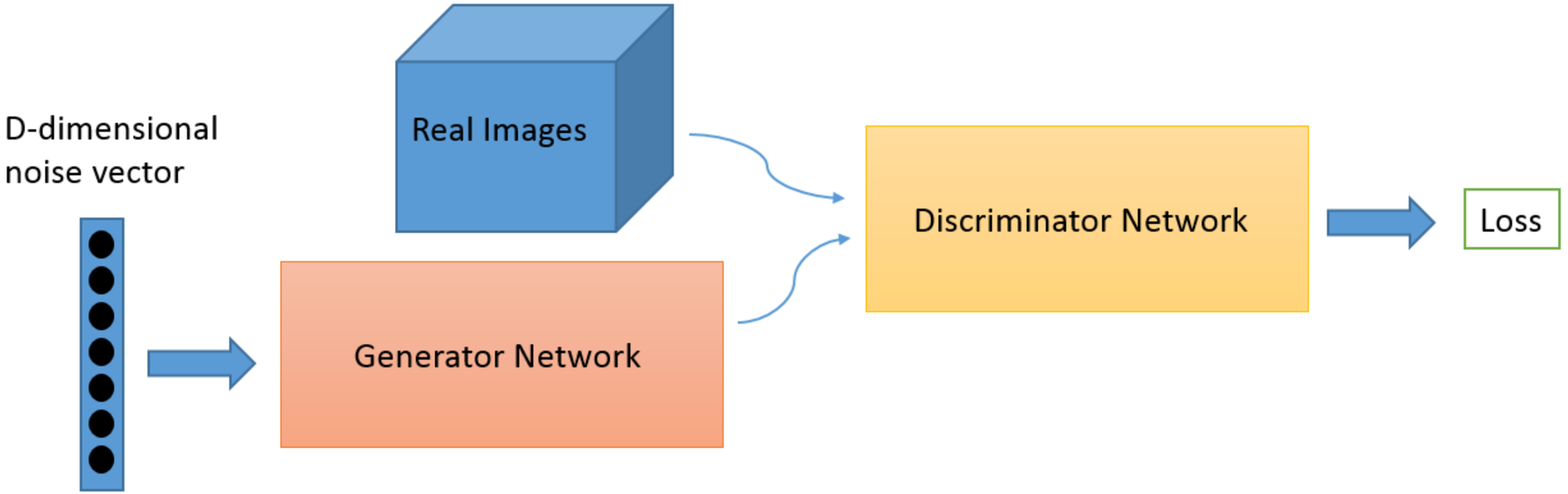
```
decoded_imgs = denoiser_cnn.predict(x_test_noisy)

n = 10  # how many digits we will display
plt.figure(figsize=(20, 4))
for i in range(n):
    # display original
    ax = plt.subplot(2, n, i + 1)
    plt.imshow(x_test_noisy[i].reshape(28, 28))
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)

    # display reconstruction
    ax = plt.subplot(2, n, i + 1 + n)
    plt.imshow(decoded_imgs[i].reshape(28, 28))
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)
plt.show()
```



Denoising auto-encoder

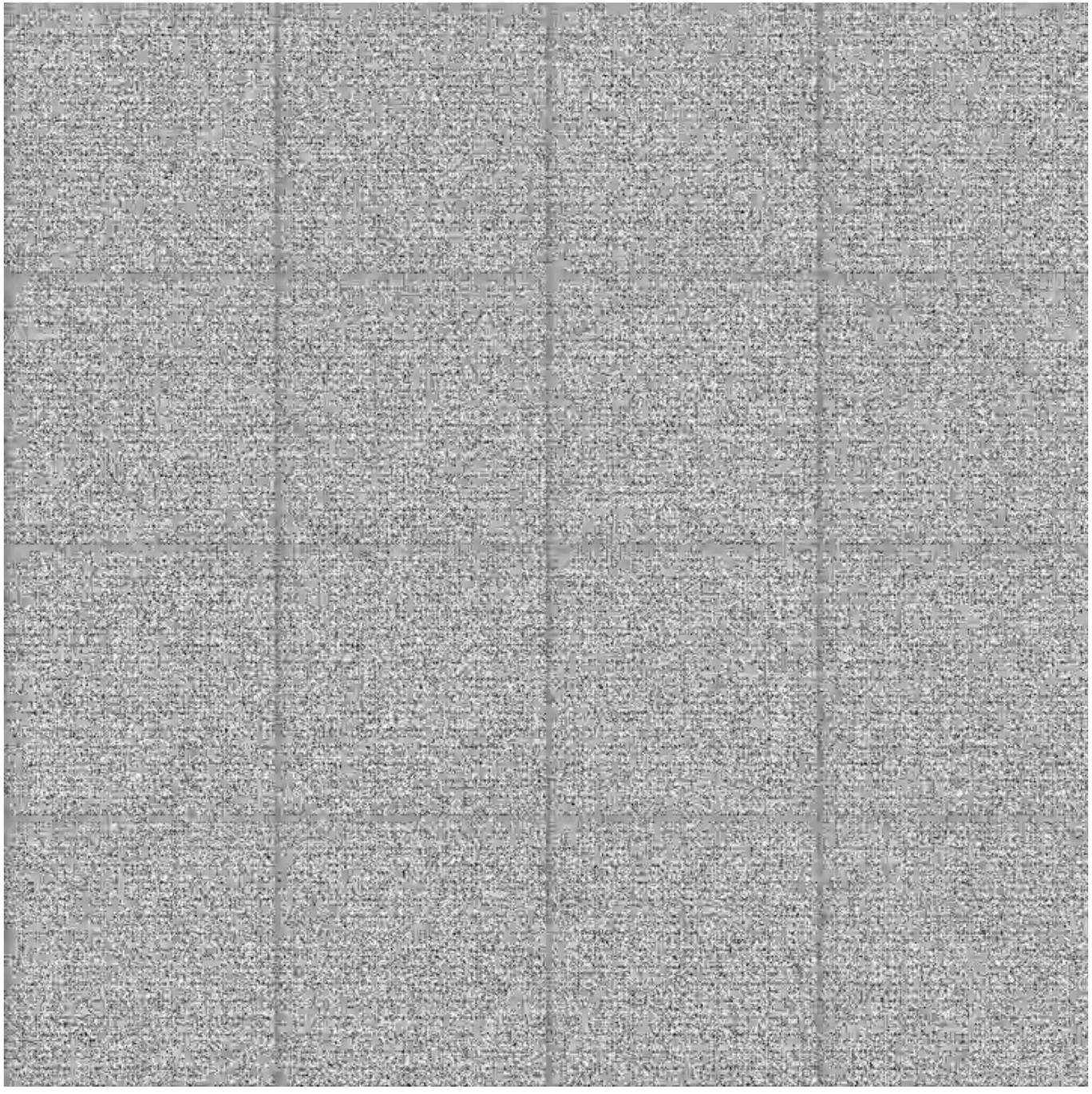
GANs

Generative Adversarial Networks

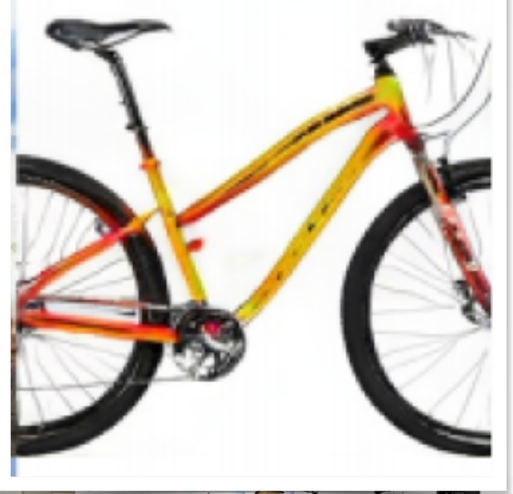


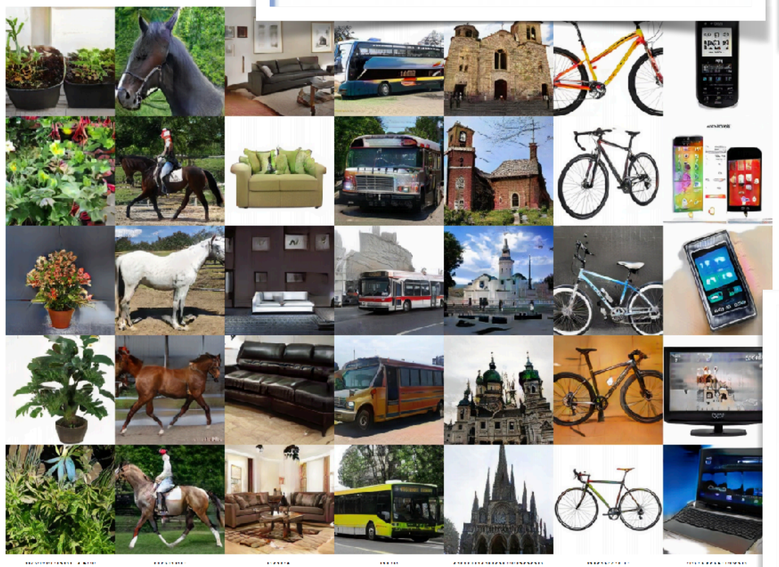
The loss function is measuring how good the discriminator can distinguish between real and generated images!

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$



Nvidia @NIPS 2017





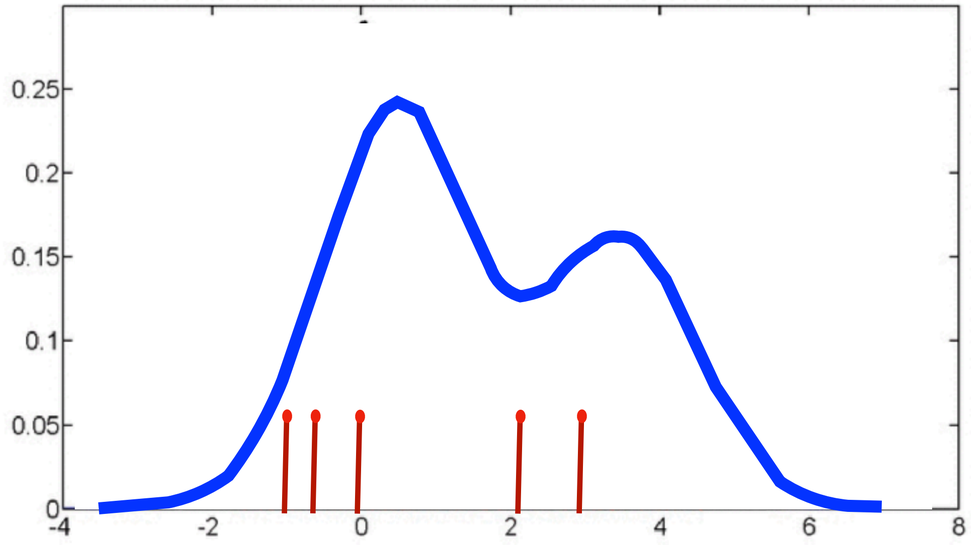
https://thispersondoesnotexist.com/image

Generating models & sampling

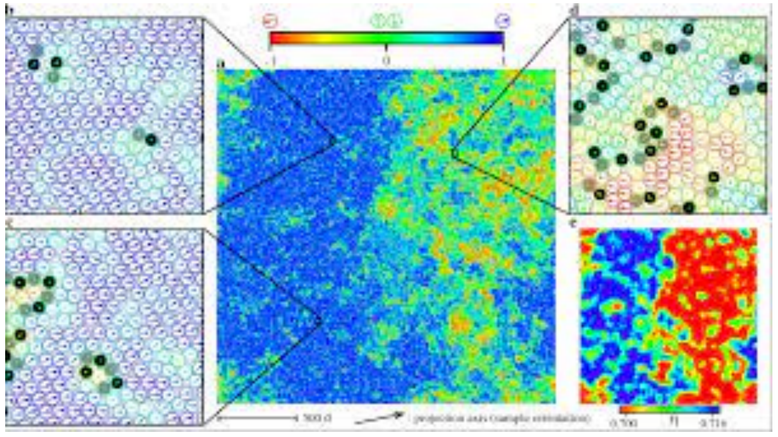
A tale of diffusion, flow, and stochastic interpolants

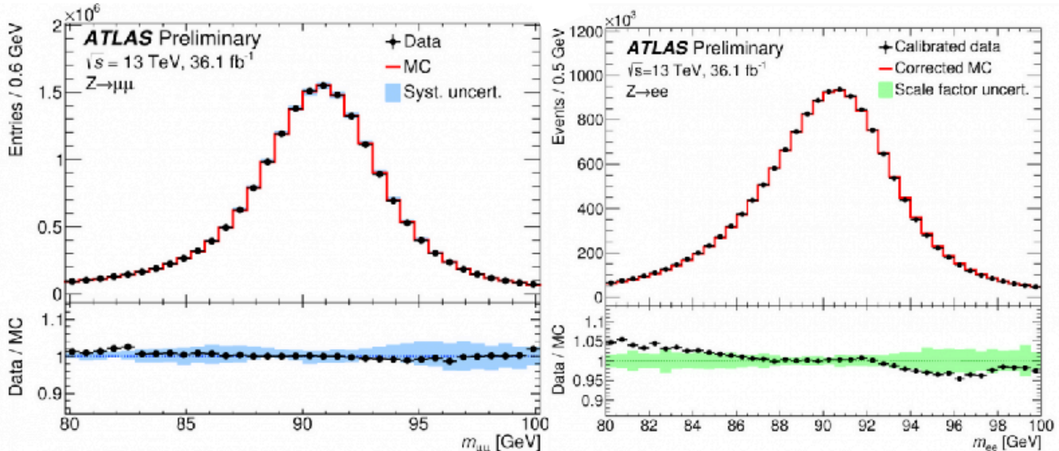
Density Estimation and Sampling

- **Sampling**: Given a probability measure $\mu \in \mathscr{P}(\Omega)$, generate a sample $\mathbf{x} \sim \mathscr{P}$.



The ability to sample is fundamental in statistical physics, quantum mechanics, high energy physics etc ….



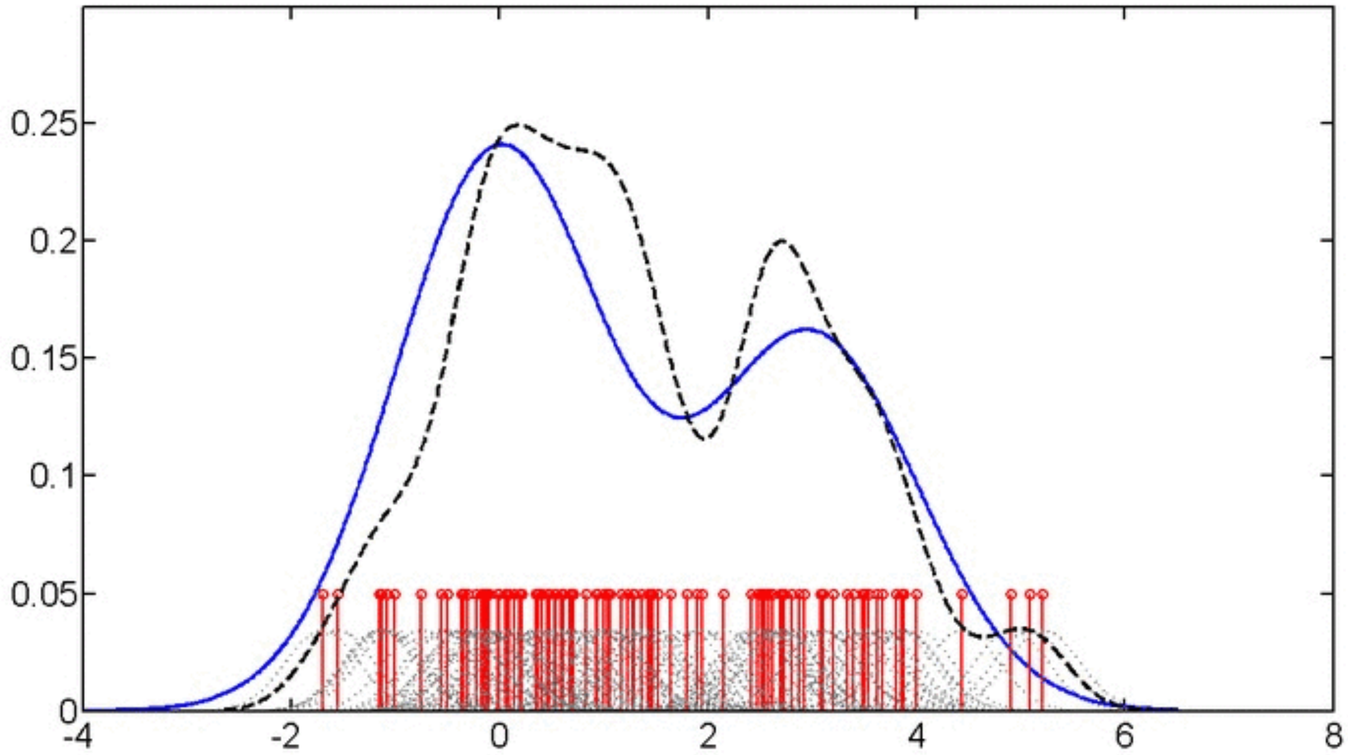


Density Estimation and Sampling

- **Sampling**: Given a probability measure $\mu \in \mathscr{P}(\Omega)$, generate a sample $\mathbf{x} \sim \mathscr{P}$.
- **Density Estimation**: Given data $\{\mathbf{x}_i\}_{i=1}^n$from the unknown probability measure $\mu \in \mathscr{P}(\Omega)$ calculate an estimate $\hat{\mu}$ of μ (possibly up to a normalizing constant).

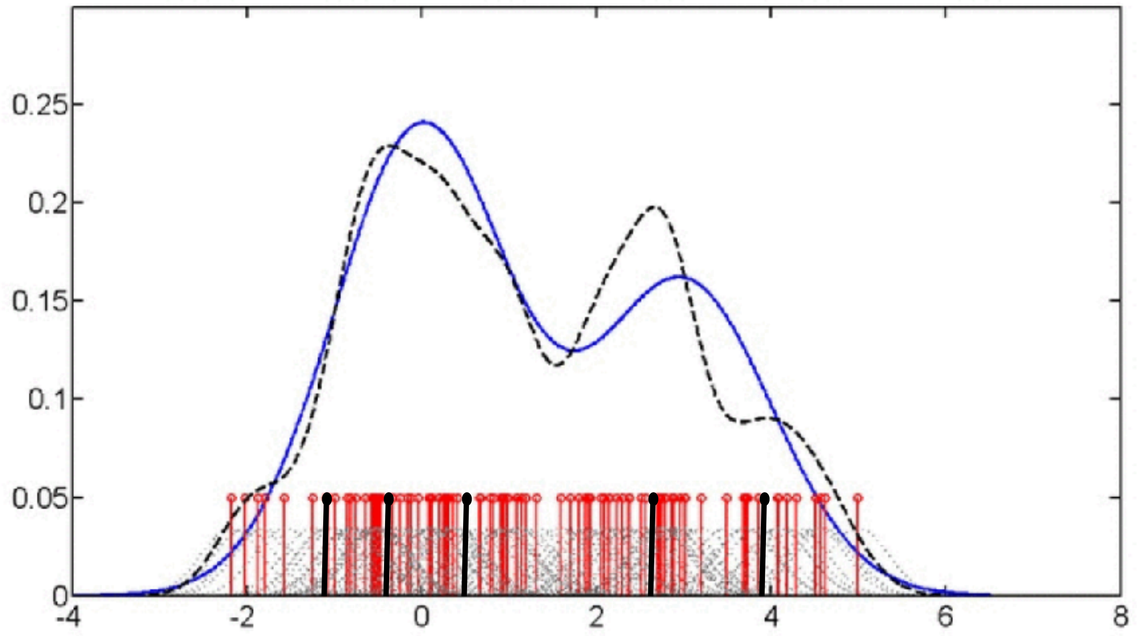
Density Estimation and Sampling

- **Sampling**: Given a probability measure $\mu \in \mathscr{P}(\Omega)$, generate a sample $\mathbf{x} \sim \mathscr{P}$.
- **Density Estimation**: Given data $\{\mathbf{x}_i\}_{i=1}^n$from the unknown probability measure $\mu \in \mathscr{P}(\Omega)$ calculate an estimate $\hat{\mu}$ of μ (possibly up to a normalizing constant).



Generative modeling: combine density estimation and sampling

- (i) **Density Estimation of** $\hat{\mu}$ (Given data $\{\mathbf{x}_i\}_{i=1}^n$)
- (ii) Then **Sampling** from $\hat{\mu}$ to generate new data $\mathbf{x}_{\text{new}}$.



Generative modeling: combine density estimation and sampling

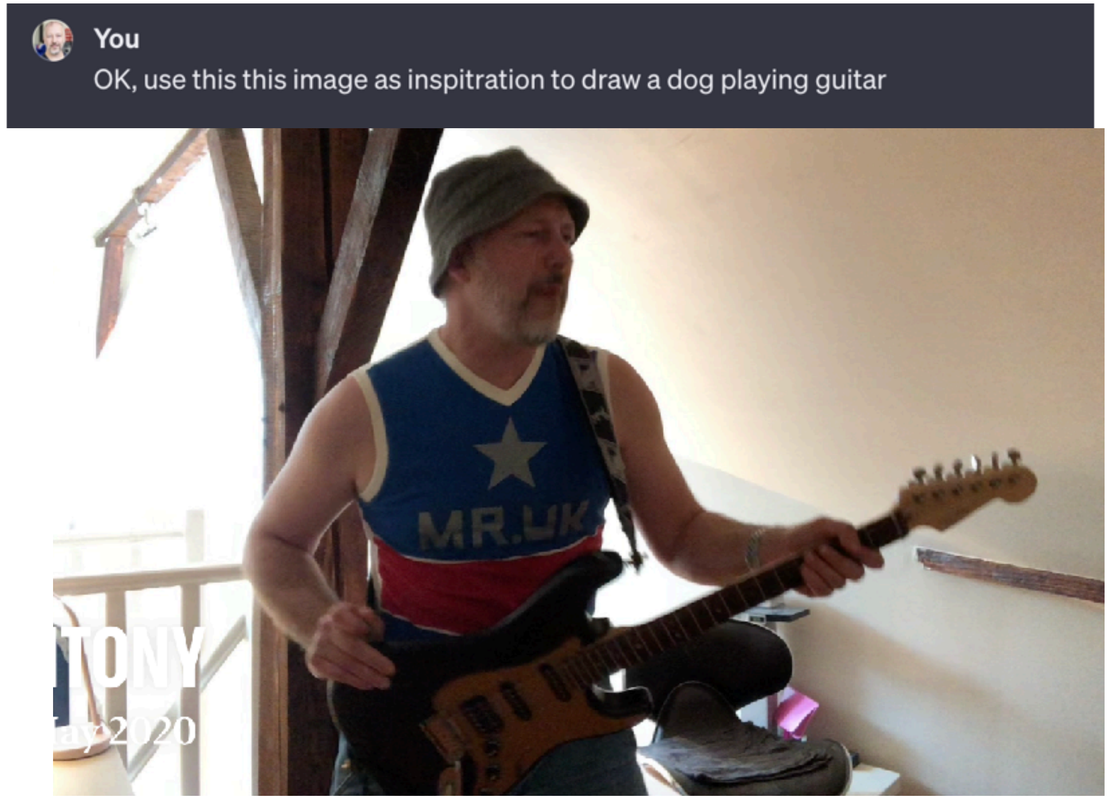
- (i) **Density Estimation of** $\hat{\mu}$ (Given data $\{\mathbf{x}_i\}_{i=1}^n$)
- (ii) Then **Sampling** from $\hat{\mu}$ to generate new data $\mathbf{x}_{\text{new}}$.

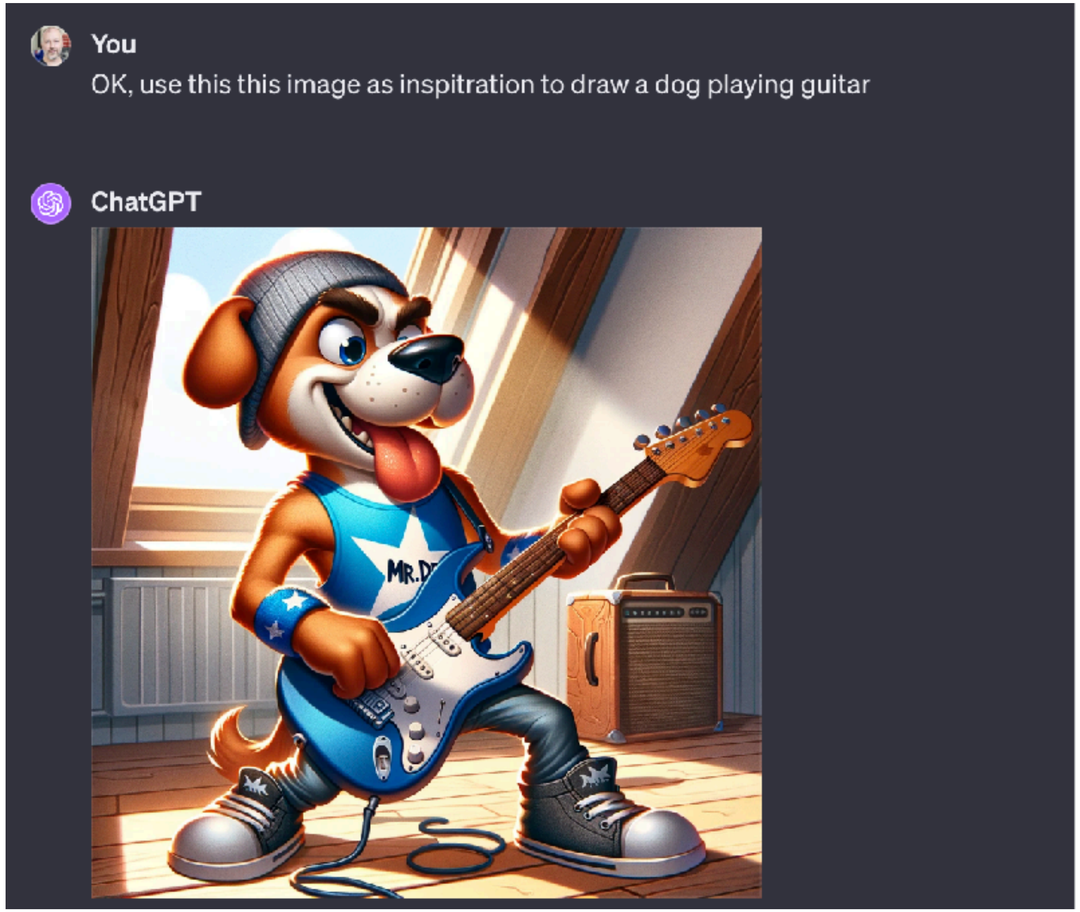
(NVIDIA group, ICLR 2018)

(Song et al, 2021)

Generative modeling: combine density estimation and sampling

- (i) **Density Estimation of** $\hat{\mu}$ (Given data $\{\mathbf{x}_i\}_{i=1}^n$)
- (ii) Then **Sampling** from $\hat{\mu}$ to generate new data $\mathbf{x}_{\text{new}}$.





Generative modeling: combine density estimation and sampling

- (i) **Density Estimation of** $\hat{\mu}$ (Given data $\{\mathbf{x}_i\}_{i=1}^n$)
- (ii) Then **Sampling** from $\hat{\mu}$ to generate new data $\mathbf{x}_{\text{new}}$.

Many methods over the years:

- Kernel density estimation
- Boltzmann machines
- Variational auto-encoder
- Generative adversarial networks (GANs)
- Energy-based models
- Normalizing flows
- Diffusion models
- Stochastic localisation

GOAL: Construct an ODE or a SDE

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

such that: *if* $X_{t=t_1} \sim \mu_{\text{simple}}$ = *simple base measure, then* $X_{t=t_2} \sim \mu_{\text{target}}$= *target measure.*

Well-suited formalism for ***sampling****:*

- draw a sample from the simple base measure;

- propagate it through the S/ODE;

- get a sample from the target

Stochastic Interpolants:
A Unifying Framework for Flows and Diffusions

Michael S. Albergo*[1], Nicholas M. Boffi*[2], and Eric Vanden-Eijnden[2]

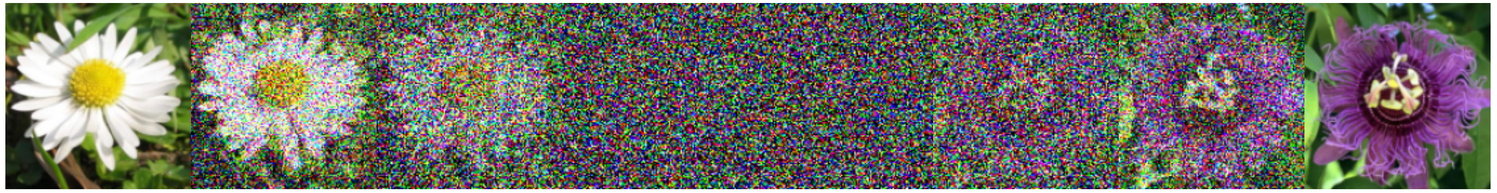
[1]Center for Cosmology and Particle Physics, New York University
[2]Courant Institute of Mathematical Sciences, New York University

Well-suited formalism for ***density estimation****:*

- learn the function $\mathbf{b}(\mathbf{X}_t, t)$ *and/or* $\sigma(\mathbf{X}_t, t)$ *such that we can "fit" a target distribution to the data*

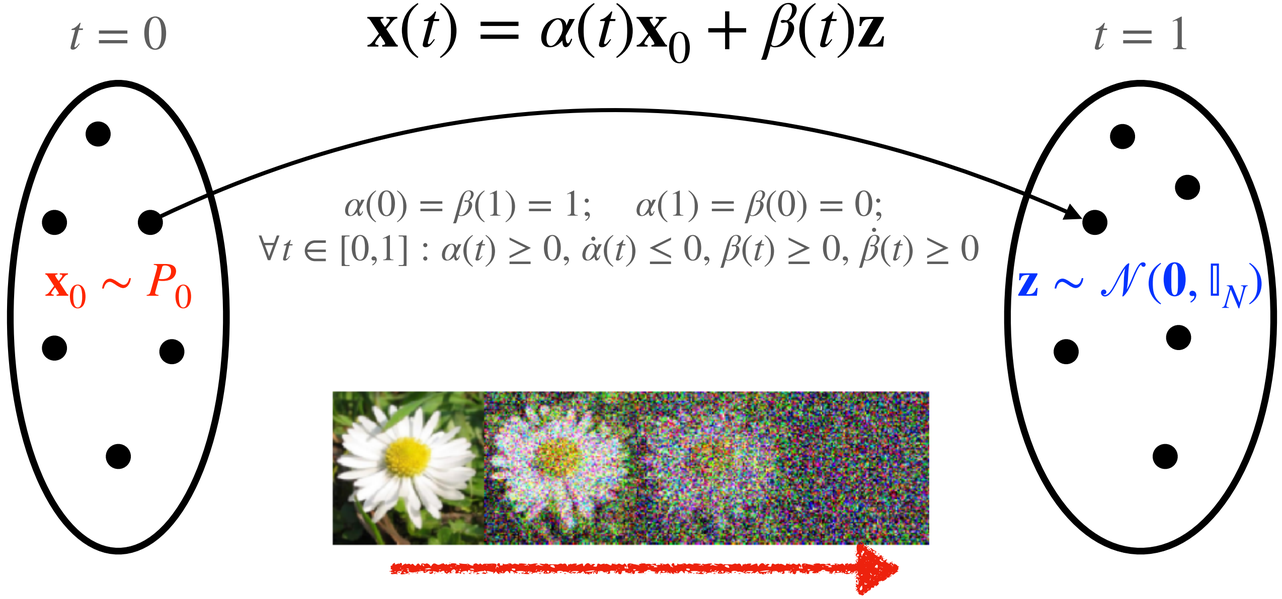
I

A short introduction to stochastic interpolants



[Albergo, Boffi, Vanden-Eijnden ’23]

One-sided linear interpolant process



Can show: At any $t \in [0,1]$, density $\rho(\mathbf{x}(t))$ solves the *transport equation*

$$\partial_t \rho + \nabla \cdot (\mathbf{b}\rho) = 0$$

$$\mathbf{b}(\mathbf{x}, t) = \mathbb{E}[\partial_t \mathbf{x}(t) \,|\, \mathbf{x}(t) = \mathbf{x}] = \mathbb{E}[\dot{\alpha}(t)\mathbf{x_0} + \dot{\beta}(t)\mathbf{z} \,|\, \mathbf{x}(t) = \mathbf{x}] \,.$$

[Albergo, Boffi, Vanden-Eijnden ’23]

One-sided linear interpolant process

$$\mathbf{x}(t) = \alpha(t)\mathbf{x}_0 + \beta(t)\mathbf{z}$$

$$\rho(\mathbf{x}, t) = \int d\mathbf{x}_0 d\mathbf{z} P(\mathbf{x}_o) P(\mathbf{z}) \delta(\mathbf{x} - \mathbf{x}(t))$$

$$\rho(\mathbf{x}, t) = \mathbb{E}_{\mathbf{x}_0, \mathbf{z}}[\delta(\mathbf{x} - \mathbf{x}(t))]$$

$$\partial_t \rho(\mathbf{x}, t) = -\mathbb{E}_{\mathbf{x}_0, \mathbf{z}}[\dot{\mathbf{x}}(t)\delta'(\mathbf{x} - \mathbf{x}(t))]$$

$$\mathbf{b}(\mathbf{x}, t) = \mathbb{E}[\partial_t \mathbf{x}(t) \,|\, \mathbf{x}(t) = \mathbf{x}]$$

$$\mathbf{b}(\mathbf{x}, t) = \frac{\int d\mathbf{x}_0 d\mathbf{z} P(\mathbf{x}_0) P(\mathbf{z}) \dot{\mathbf{x}}(t) \delta(\mathbf{x} - \mathbf{x}(t))}{\int d\mathbf{x}_0 d\mathbf{z} P(\mathbf{x}_0) P(\mathbf{z}) \delta(\mathbf{x} - \mathbf{x}(t))}$$

$$\rho(\mathbf{x}, t)\mathbf{b}(\mathbf{x}, t) = \mathbb{E}_{\mathbf{x}_0, \mathbf{z}}[\dot{x}(t)\delta(\mathbf{x} - \mathbf{x}(t))]$$

$$\nabla(\rho(\mathbf{x}, t)\mathbf{b}(\mathbf{x}, t)) = \mathbb{E}_{\mathbf{x}_0, \mathbf{z}}[\dot{\mathbf{x}}(t)\delta'(\mathbf{x} - \mathbf{x}(t))]$$

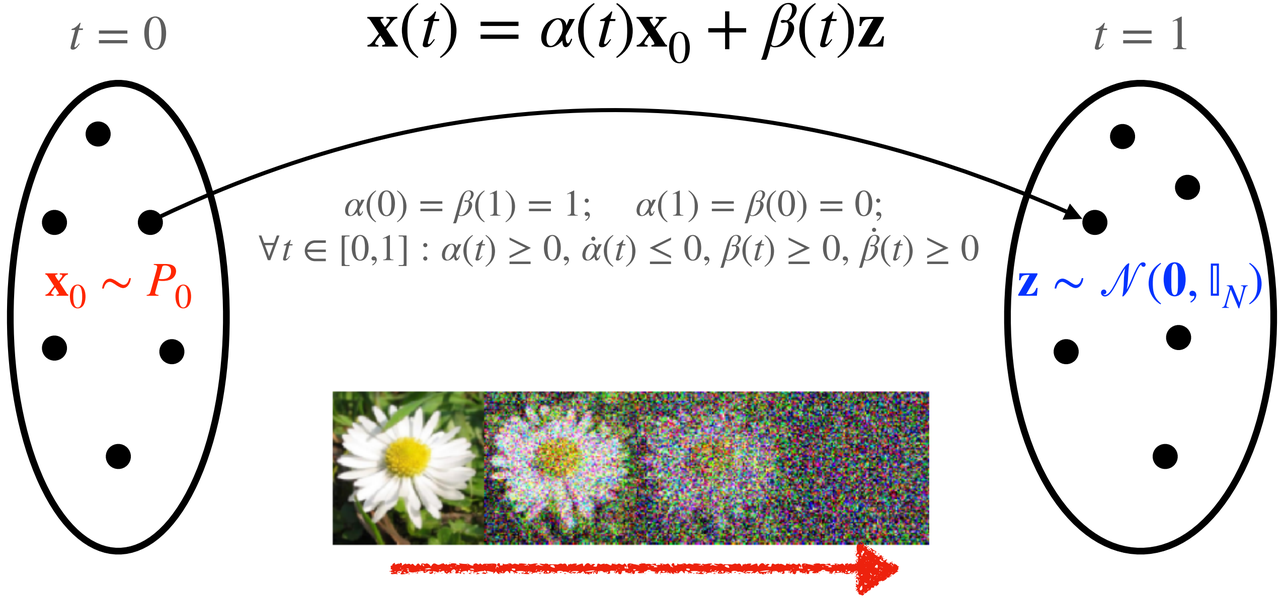
Can show: At any $t \in [0,1]$, density $\rho(\mathbf{x}(t))$ solves the *transport equation*

$$\boxed{\partial_t \rho + \nabla \cdot (\mathbf{b}\rho) = 0}$$

$$\mathbf{b}(\mathbf{x}, t) = \mathbb{E}[\partial_t \mathbf{x}(t) \,|\, \mathbf{x}(t) = \mathbf{x}] = \mathbb{E}[\dot{\alpha}(t)\mathbf{x_0} + \dot{\beta}(t)\mathbf{z} \,|\, \mathbf{x}(t) = \mathbf{x}]\,.$$

[Albergo, Boffi, Vanden-Eijnden ’23]

One-sided linear interpolant process



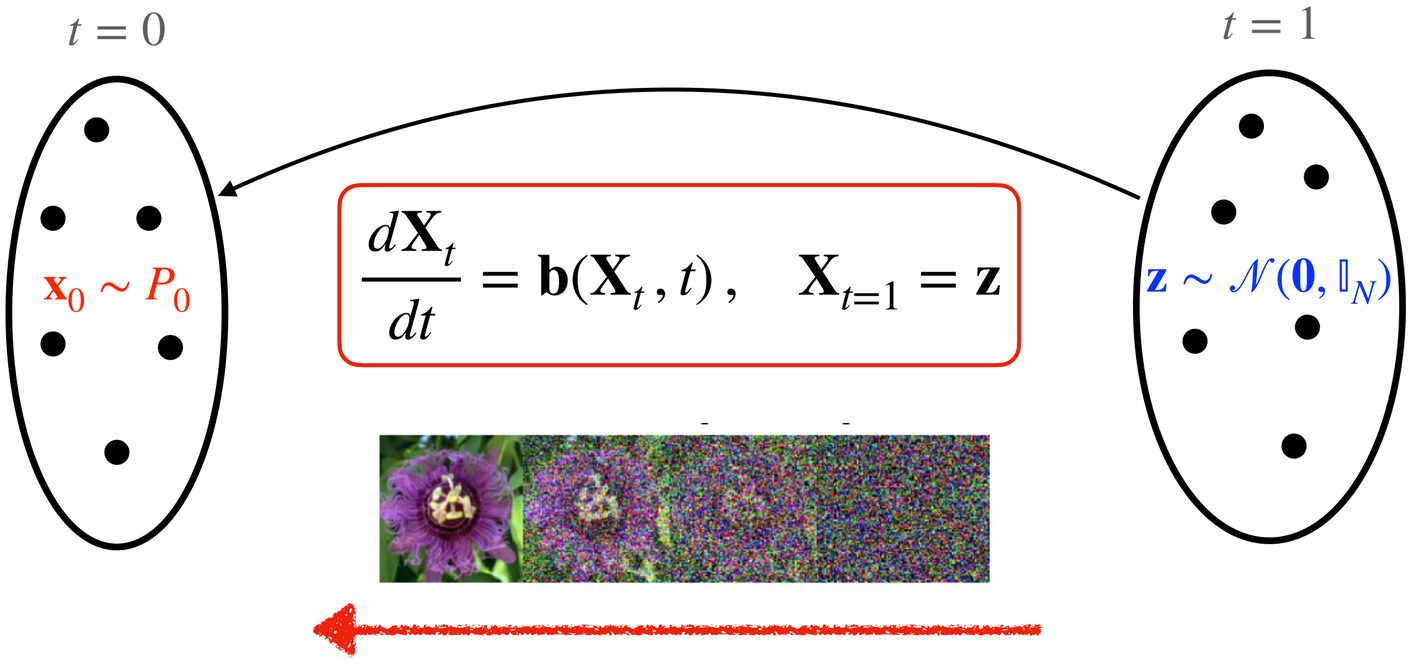
Can show: At any $t \in [0,1]$, density $\rho(\mathbf{x}(t))$ solves the *transport equation*

$$\partial_t \rho + \nabla \cdot (\mathbf{b}\rho) = 0$$

$$\mathbf{b}(\mathbf{x}, t) = \mathbb{E}[\partial_t \mathbf{x}(t) \,|\, \mathbf{x}(t) = \mathbf{x}] = \mathbb{E}[\dot{\alpha}(t)\mathbf{x_0} + \dot{\beta}(t)\mathbf{z} \,|\, \mathbf{x}(t) = \mathbf{x}] \,.$$

[Albergo, Boffi, Vanden-Eijnden ’23]

Backward ODE: Sampling through denoising



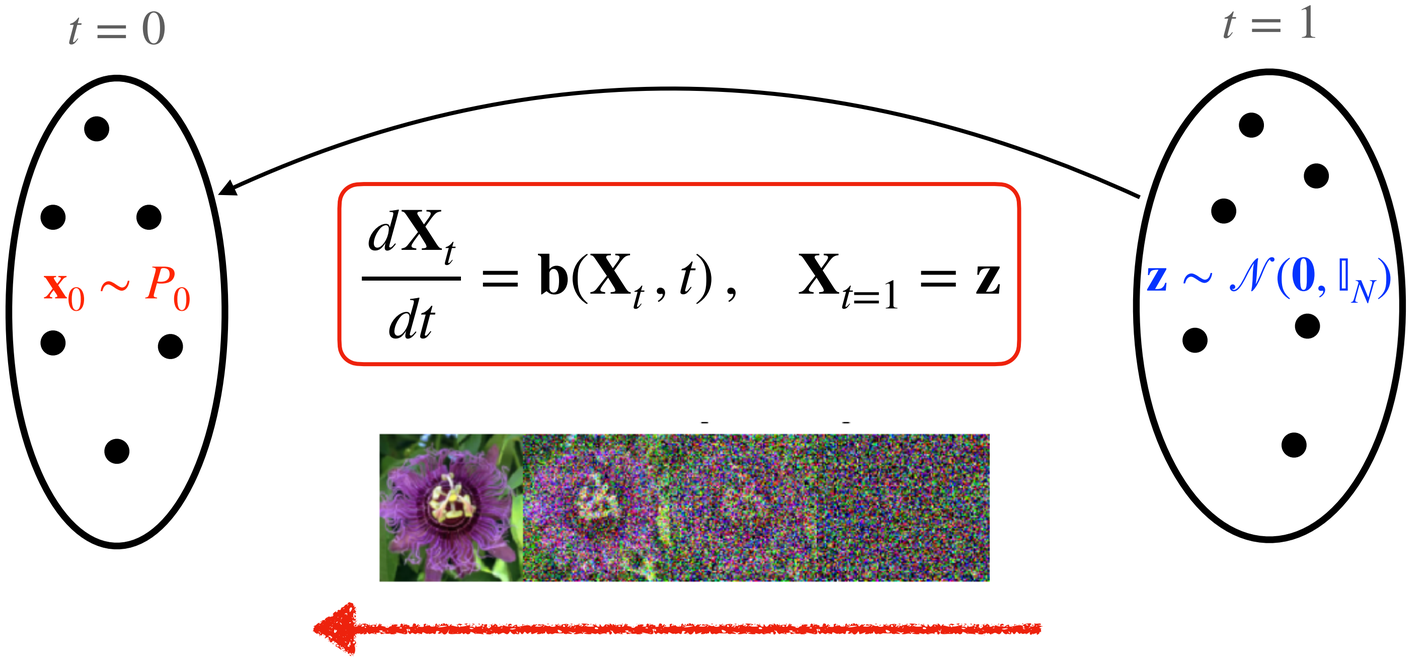
1. Mechanical analogy: Each "particles" at position $\mathbf{X}_t$ move with velocity $\mathbf{b}(\mathbf{X}_t, t)$

2. The **current** is given by $J = \rho(\mathbf{X}_t, t)\mathbf{b}(\mathbf{X}_t, t)$. Conservation of mass imposes $\nabla \cdot J = \frac{\partial \rho_t}{\partial t}$, or equivalently $\partial_t \rho + \nabla \cdot (\mathbf{b}\rho) = 0$

3. This is the same equation as before! Hence, in law, we have *at all times* $\mathbf{X}_t = \alpha(t)\mathbf{X}_0 + \beta(t)\mathbf{Z}$

4. In particular, $\mathbf{X}_0$ is distributed as P_0

Backward ODE: Sampling through denoising



How to compute $\mathbf{b}(\mathbf{X}_t, t)$?

$$\mathbf{b}(\mathbf{x}, t) = \mathbb{E}[\partial_t \mathbf{x}(t) \,|\, \mathbf{x}(t) = \mathbf{x}] = \mathbb{E}[\dot{\alpha}(t)\mathbf{x_0} + \dot{\beta}(t)\mathbf{z} \,|\, \mathbf{x}(t) = \mathbf{x}] .$$

You are given $\mathbf{X}_t$, so the only problem is to estimate $\mathbb{E}\left[\mathbf{X}_0 \,|\, \mathbf{X}_t\right]$

$$\mathbf{b}(\mathbf{X}_t, t) = \frac{\dot{\beta}(t)}{\beta(t)}\mathbf{X}_t + \left(\dot{\alpha}(t) - \frac{\dot{\beta}(t)\alpha(t)}{\beta(t)}\right)\mathbb{E}\left[\mathbf{X}_0 \,|\, \mathbf{X}_t\right]$$

Gaussian denoising

Unknown signal

AWGN Channel Observation

Optimal denoising

$$\mathbf{X}_0 \longrightarrow \frac{\mathbf{X}(t)}{\alpha_t} = \mathbf{X}_0 + \frac{\beta(t)}{\alpha(t)}\mathbf{Z} \longrightarrow \hat{\mathbf{x}}_0 = ?$$

$$\mathbf{b}(\mathbf{X}_t, t) = \frac{\dot{\beta}(t)}{\beta(t)}\mathbf{X}_t + \left(\dot{\alpha}(t) - \frac{\dot{\beta}(t)\alpha(t)}{\beta(t)}\right)\mathbb{E}\left[\mathbf{X}_0 \,|\, \mathbf{X}_t\right]$$

Gaussian denoising

Unknown signal

AWGN Channel Observation

Optimal denoising

$$\mathbf{X}_0 \longrightarrow \frac{\mathbf{X}(t)}{\alpha_t} = \mathbf{X}_0 + \frac{\beta(t)}{\alpha(t)}\mathbf{Z} \longrightarrow \hat{\mathbf{x}}_0 = ?$$

$$\mathbf{b}(\mathbf{X}_t, t) = \frac{\dot{\beta}(t)}{\beta(t)}\mathbf{X}_t + \left(\dot{\alpha}(t) - \frac{\dot{\beta}(t)\alpha(t)}{\beta(t)}\right)\mathbb{E}\left[\mathbf{X}_0 \,|\, \mathbf{X}_t\right]$$

Bayes Theorem

$$P(\mathbf{X}_0 \,|\, \mathbf{X}(t)) = \frac{1}{P(\mathbf{X}(\mathbf{t}))} P(\mathbf{X}(t) \,|\, \mathbf{X}_0) P_0(\mathbf{X}_0) \longrightarrow$$

MMSE estimator

$$\widehat{\mathbf{X}}_0 = \mathbb{E}\left[\mathbf{X}_0 \,|\, \mathbf{X}(t)\right]$$

$\mathbf{X}(t) = \alpha(t)\mathbf{X}_0 + \beta(t)\mathbf{Z}$, find the MMSE estimator $\widehat{\mathbf{X}}_0 = \mathbb{E}\left[\mathbf{X} \,|\, \mathbf{X}(t)\right]$

Solving the backward ODE in a nutshell

$$\frac{\mathrm{d}\mathbf{X}_t}{\mathrm{d}t} = \frac{\dot{\beta}(t)}{\beta(t)}\mathbf{X}_t + \left(\dot{\alpha}(t) - \frac{\dot{\beta}(t)\alpha(t)}{\beta(t)}\right)\mathbb{E}\left[\mathbf{x}_0 \,|\, \mathbf{x}(t) = \mathbf{X}_t\right]$$

1) **Discretisation** (Backward Euler's method):

$$\mathbf{X}_{t-\delta} = \left(1 - \delta\frac{\dot{\beta}(t)}{\beta(t)}\right)\mathbf{X}_t - \delta\left(\dot{\alpha}(t) - \frac{\dot{\beta}(t)\alpha(t)}{\beta(t)}\right)\mathbb{E}\left[\mathbf{x}_0 \,|\, \mathbf{x}(t) = \mathbf{X}_t\right]$$

2) **Posterior average estimation**: It is just a **Gaussian denoising** problem!

Given $\mathbf{x}(t) = \alpha(t)\mathbf{x}_0 + \beta(t)\mathbf{z}$, find the MMSE estimator $\hat{\mathbf{x}}_0 = \mathbb{E}\left[\mathbf{x} \,|\, \mathbf{x}(t)\right]$

$$P(\mathbf{x} \,|\, \mathbf{x}(t) = \mathbf{X}_t) = \frac{1}{Z(\mathbf{Y}_t)}\exp\left(\frac{\alpha(t)}{\beta(t)^2}\langle \mathbf{X}_t, \mathbf{x}\rangle - \frac{\alpha(t)^2}{2\beta(t)^2}\|\mathbf{x}\|^2\right)P_0(\mathbf{x})$$

Tilting field

Target measure

Solving the backward ODE in a nutshell

$$\frac{\mathrm{d}\mathbf{X}_t}{\mathrm{d}t} = \frac{\dot{\beta}(t)}{\beta(t)}\mathbf{X}_t + \left(\dot{\alpha}(t) - \frac{\dot{\beta}(t)\alpha(t)}{\beta(t)}\right)\mathbb{E}\left[\mathbf{x}_0 \,|\, \mathbf{x}(t) = \mathbf{X}_t\right]$$

1) **Discretisation** (Backward Euler's method):

$$\mathbf{X}_{t-\delta} = \left(1 - \delta\frac{\dot{\beta}(t)}{\beta(t)}\right)\mathbf{X}_t - \delta\left(\dot{\alpha}(t) - \frac{\dot{\beta}(t)\alpha(t)}{\beta(t)}\right)\mathbb{E}\left[\mathbf{x}_0 \,|\, \mathbf{x}(t) = \mathbf{X}_t\right]$$

2) **Posterior average estimation**: It is just a **Gaussian denoising** problem!

Given $\mathbf{x}(t) = \alpha(t)\mathbf{x}_0 + \beta(t)\mathbf{z}$, find the MMSE estimator $\hat{\mathbf{x}}_0 = \mathbb{E}\left[\mathbf{x} \,|\, \mathbf{x}(t)\right]$

$$P(\mathbf{x} \,|\, \mathbf{x}(t) = \mathbf{x}_t) \propto \exp\left(-\mathscr{H}_0 + \frac{\alpha(t)}{\beta(t)^2}\langle \mathbf{X}_t, \mathbf{x}\rangle - \frac{\alpha(t)^2}{2\beta(t)^2}\|\mathbf{x}\|^2\right)$$

Target measure

Tilting field

Solving the backward ODE in a nutshell

Algorithm 1 Flow-based sampling algorithm

Input: Denoiser, parameters: $\delta, \alpha(t), \beta(t)$

$X_{t=1} = \mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}_N)$, $N_{\text{steps}} \equiv \lfloor 1/\delta \rfloor$

for $l = 0, 1, \ldots, N_{\text{steps}} - 1$ **do**

$t = 1 - l\delta$; $\alpha = \alpha(t)$; $\beta = \beta(t)$; $\gamma = \alpha/\beta$

Compute $\hat{\mathbf{x}}_0 = \mathbb{E}[\mathbf{x}_0 | X_t]$ using the denoiser

(assuming a channel $X_t = \alpha \mathbf{X}_0 + \beta \mathbf{Z}$)

Update the field $X_{t-\delta}$ via $\mathbf{X}_{t-\delta} = \left(1 - \delta \frac{\dot{\beta}(t)}{\beta(t)}\right) \mathbf{X}_t - \delta \left(\dot{\alpha}(t) - \frac{\dot{\beta}(t)\alpha(t)}{\beta(t)}\right) \hat{\mathbf{x}}_0$

end for

Return $\hat{x}_0$

The crux of the problem: computing the denoiser

$$P(\mathbf{x} \mid \mathbf{x}(t) = \mathbf{X}_t) = \frac{1}{Z(\mathbf{X}_t)} \exp\left(\frac{\alpha(t)}{\beta(t)^2} \langle \mathbf{X}_t, \mathbf{x} \rangle - \frac{\alpha(t)^2}{2\beta(t)^2} \|\mathbf{x}\|^2 \right) P_0(\mathbf{x}) \qquad \hat{\mathbf{x}}_0 = \mathbb{E}\left[\mathbf{X} \mid \mathbf{X}(t) \right]$$

1

If $P_0(\mathbf{x})$ is known …

… "*just*" estimate the integral…

… $\hat{\mathbf{x}}_0 = \mathbb{E}\left[\mathbf{X} \mid \mathbf{X}(t) \right]$

High-dimensional integral that can be hard to compute depending on the problem (Optimal Bayesian Inference)

2

Often $P_0(\mathbf{x})$ is not known …

…. rather we are given many examples …

… in the form of a Dataset $\mathscr{D} = \{\mathbf{x}_i\}_{i=1}^n$

The strategy is to learn a (sequence of) denoiser(s) by training a neural network on the dataset $\mathscr{D}$

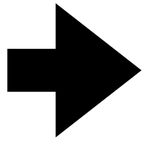
Learning from from data: many options

Learn the velocity field:

$b(\mathbf{x}, t)$ is the unique minimiser of

$$\mathscr{R}[\hat{b}] = \int_0^1 dt \mathbb{E}_{X_0,Z}\left[\hat{b}(\mathbf{x}_t, t)^2 - 2\dot{\mathbf{x}}_t \hat{b}(\mathbf{x}_t, t)\right]$$

Represent $b(\mathbf{x}, t)$ as a neural net, and minimise instead the empirical loss over the training sample



$$\mathscr{R}^{\text{emp}}[\hat{b}] = \int_0^1 dt \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_Z\left[\hat{b}(\mathbf{x}_t^{(i)}, t)^2 - 2\dot{\mathbf{x}}_t^{(i)} \hat{b}(\mathbf{x}_t^{(i)}, t)\right]$$

Learn the optimal denoiser:

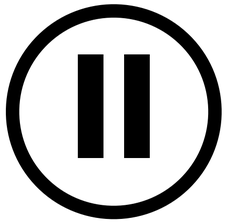
$\eta(\mathbf{x}_t, \alpha, \beta)$ is the unique minimiser of

$$\mathscr{R}[\hat{\eta}_D] = \int_0^1 dt \mathbb{E}_{X_0,Z}\left[\|\eta(\mathbf{x}_t, t) - \mathbf{x_0}\|_2^2\right]$$

Represent $\eta(\mathbf{x}_t, \alpha, \beta)$ as a neural net, and minimise instead the empirical loss over the training sample

$$\mathscr{R}^{\text{epm}}[\hat{\eta}_D] = \int_0^1 dt \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_Z\left[\|\eta(\mathbf{x}_t, t) - \mathbf{x_0}\|_2^2\right]$$

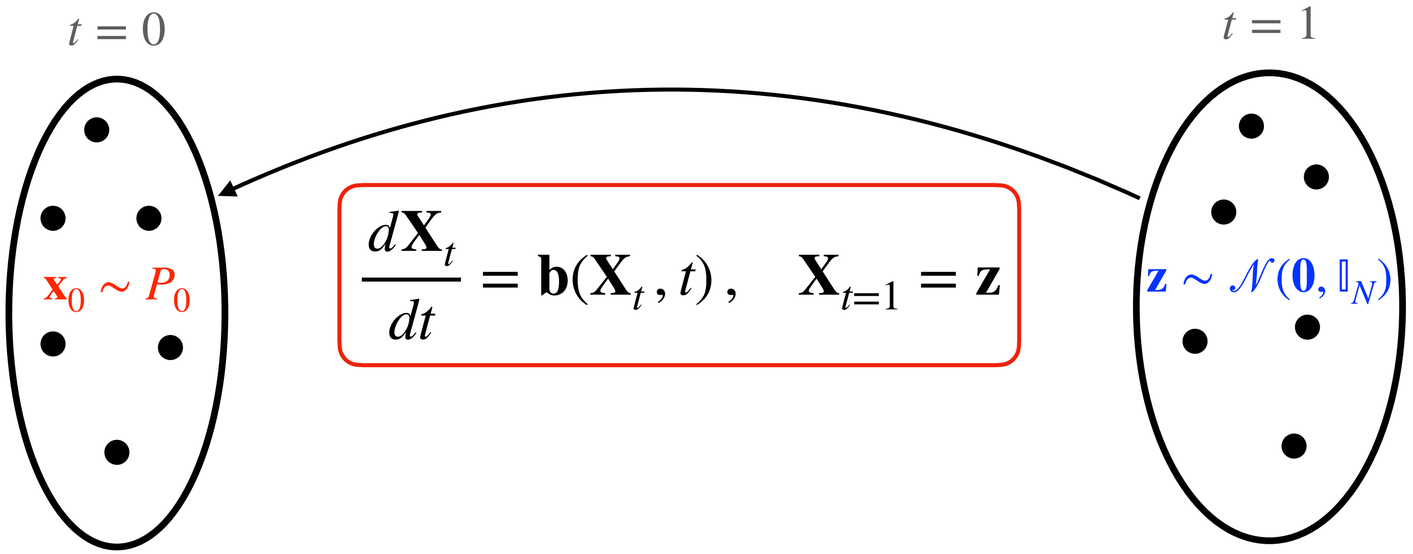
Careful : do not just memorise the dataset! The choice of NN yield an implicit regularisation



But what about “score”, “SDE”, “diffusion” etc etc?

The score

Backward ODE: Sampling through denoising



$$\mathbf{b}(\mathbf{X}_t, t) = \frac{\dot{\beta}(t)}{\beta(t)}\mathbf{X}_t + \left(\dot{\alpha}(t) - \frac{\dot{\beta}(t)\alpha(t)}{\beta(t)}\right)\mathbb{E}\left[\mathbf{X}_0 \,|\, \mathbf{X}_t\right]$$

How to compute $\mathbf{b}(\mathbf{X}_t, t)$?

A trivial function of the optimal denoiser $\mathbb{E}\left[\mathbf{X}_0 \,|\, \mathbf{X}_t\right] = \eta_{\alpha,\beta}(X_t)$

Alternative view point with the score (via Tweedie's formula)

$$P(\mathbf{X}_t) = \int d\mathbf{X}_0 P(\mathbf{X}_t \mid \mathbf{X}_0) P(\mathbf{X}_0) \quad \Rightarrow \quad \nabla P(\mathbf{X}_t) = \int d\mathbf{X}_0 \nabla P(\mathbf{X}_t \mid \mathbf{X}_0) P(\mathbf{X}_0)$$

$$P(\mathbf{X}_t \mid \mathbf{X}_0) = \frac{1}{(2\pi\beta^2)^{d/2}} e^{-\frac{\|\mathbf{X}_t - \alpha_t \mathbf{X}_0\|_2^2}{2\beta_t^2}} \quad \Rightarrow \quad \nabla P(\mathbf{X}_t) = -\frac{1}{\beta_t^2} \int d\mathbf{X}_0 (\mathbf{X}_t - \alpha_t \mathbf{X}_0) P(\mathbf{X}_t \mid \mathbf{X}_0) P(\mathbf{X}_0)$$

$$\nabla \log P(\mathbf{X}_t) = -\frac{1}{\beta^2} \mathbf{X}_t + \frac{\alpha_t}{\beta_t^2} \mathbb{E}[\mathbf{X_0} \mid \mathbf{X}_t]$$

$$\alpha_t \mathbb{E}[\mathbf{X}_0 \mid \mathbf{X}_t] = \mathbf{X}_t + \beta_t^2 \nabla \log P(\mathbf{X}_t)$$

Choose your camp: Denoiser vs Score !

$$\mathbf{b}(\mathbf{X}_t, t) = \frac{\dot{\beta}(t)}{\beta(t)}\mathbf{X}_t + \left(\dot{\alpha}(t) - \frac{\dot{\beta}(t)\alpha(t)}{\beta(t)} \right) \mathbb{E}\left[\mathbf{X}_0 \,|\, \mathbf{X}_t\right]$$

$$\mathbf{b}(\mathbf{X}_t, t) = \frac{\dot{\alpha}(t)}{\alpha(t)}\mathbf{X}_t + \left(\frac{\dot{\alpha}(t)}{\alpha(t)}\beta(t)^2 - \dot{\beta}(t)\beta(t)^2 \right) \frac{\beta^2}{\alpha(t)} \nabla \log P(\mathbf{X}_t)$$

Interesting interpretation as a force: if $P(\mathbf{X}_t) \propto e^{-\mathscr{H}(x)}$, then $\nabla \log P(\mathbf{X}_t) = -\nabla \mathscr{H} = F$

(With additional diffusion noise, leads to Langevin-type sampling)

Choose your camp: Denoiser vs Score !

$$\mathbf{b}(\mathbf{X}_t, t) = \frac{\dot{\beta}(t)}{\beta(t)}\mathbf{X}_t + \left(\dot{\alpha}(t) - \frac{\dot{\beta}(t)\alpha(t)}{\beta(t)}\right)\mathbb{E}\left[\mathbf{X}_0 \,|\, \mathbf{X}_t\right]$$

Thm: The score $s(t,x) = \nabla \log \rho(t,x)$ of the PDF of $x(t)$ is given for all $t \in (0,1)$ by

$$s(t,x) = -\gamma^{-1}(t)\mathbb{E}[z \,|\, x(t) = x]$$ *Stein's identity*

In addition, it is the unique minimizer of

$$L_s(\hat{s}) = \int_0^1 \mathbb{E}\left[\,|\hat{s}(t,x(t))|^2 + 2\gamma^{-1}(t) z \cdot \hat{s}(t,x(t))\right] dt$$

$$\mathbf{b}(\mathbf{X}_t, t) = \frac{\dot{\alpha}(t)}{\alpha(t)}\mathbf{X}_t + \left(\frac{\dot{\alpha}(t)}{\alpha(t)}\beta(t)^2 - \dot{\beta}(t)\beta(t)^2\right)\frac{\beta^2}{\alpha(t)}\nabla \log P(\mathbf{X}_t)$$

Interesting interpretation as a force: if $P(\mathbf{X}_t) \propto e^{-\mathscr{H}(x)}$, then $\nabla \log P(\mathbf{X}_t) = -\nabla \mathscr{H} = F$

Learning from from data: many options

Learn the velocity field:

$b(\mathbf{x}, t)$ is the unique minimiser of

$$\mathscr{R}[\hat{b}] = \int_0^1 dt \mathbb{E}_{X_0,Z}\left[\hat{b}(\mathbf{x}_t, t)^2 - 2\dot{\mathbf{x}}_t \hat{b}(\mathbf{x}_t, t)\right]$$

Learn the optimal denoiser:

$\eta(\mathbf{x}_t, \alpha, \beta)$ is the unique minimiser of

$$\mathscr{R}[\hat{\eta}_D] = \int_0^1 dt \mathbb{E}_{X_0,Z}\left[\|\eta(\mathbf{x}_t, t) - \mathbf{x_0}\|_\mathbf{2}^\mathbf{2}\right]$$

Learn the optimal score:

$S(\mathbf{x}_t)$ is the unique minimiser of

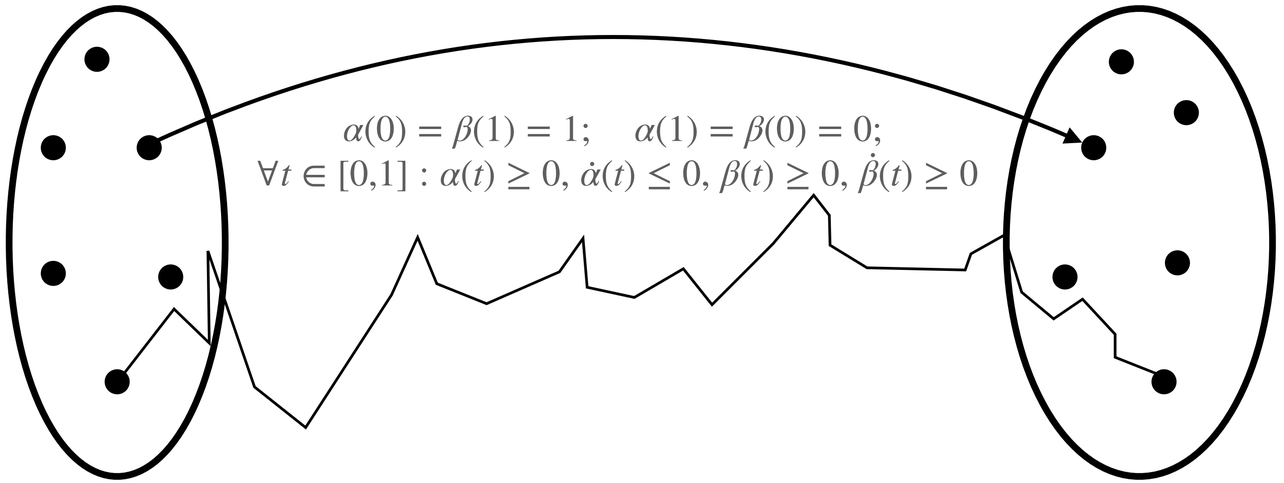
$$\mathscr{R}[\hat{S}] = \int_0^1 dt \mathbb{E}_{X_0,Z}\left[\hat{S}(\mathbf{x}_t, t)^2 - 2\dot{\mathbf{Z}}_t \hat{S}(\mathbf{x}_t, t)\right]$$

Diffusion, Noise, and SDE

More general stochastic interpolant process, and SDE

$$\mathbf{x}(t) = \alpha(t)\mathbf{x}_0 + \beta(t)\mathbf{x}_1 + \gamma(t)\xi$$

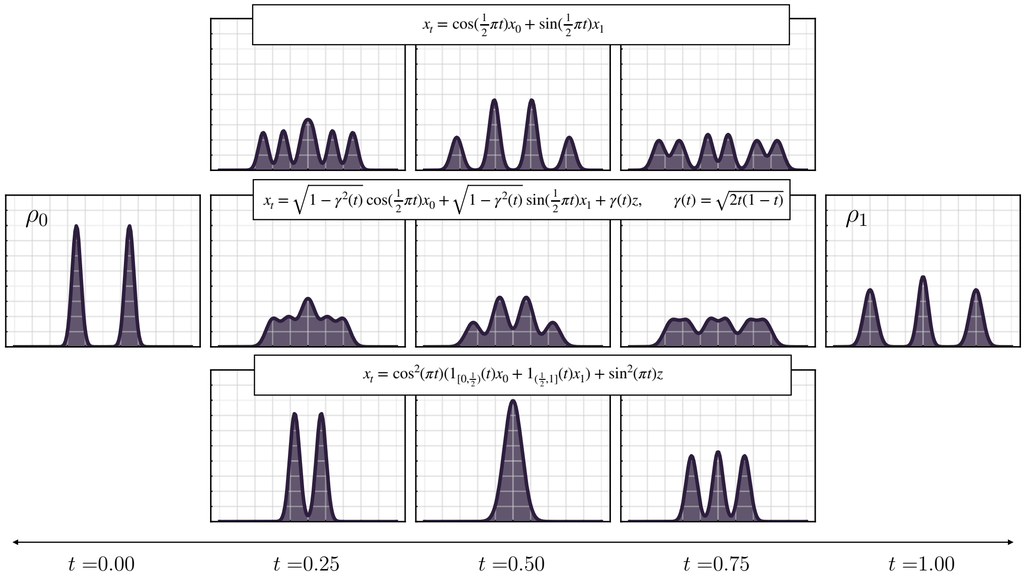
$$\alpha(0) = \beta(1) = 1 \qquad \alpha(1) = \beta(0) = \gamma(0) = \gamma(1) = 0$$



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

More general stochastic interpolant process, and SDE

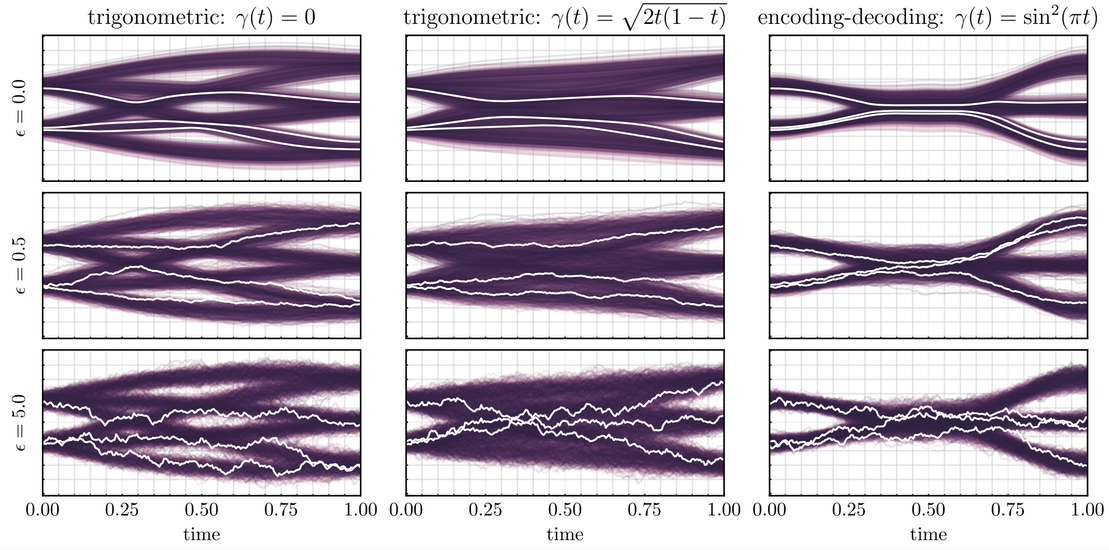
Slide credit: Eric Vanden-eijden



Choice of stochastic interpolant fixes the connecting PDF $\rho(t)$

*ODE and SDE sample the **same** $\rho(t)$ in different ways*

ODE gives one-to-one map; SDE samples ρ_1 more broadly from any $x_0 \sim \rho_0$



Flow vs Diffusion vs Stochastic localization

Flow-Based sampling

ODE-based

$$\frac{d\mathbf{X}_t}{dt} = \mathbf{b}(\mathbf{X}_t, t), \quad \mathbf{X}_{t=1} = \mathbf{z}$$

$$\mathbf{X}_{t=1} \sim \mathbf{Z} \Leftrightarrow \mathbf{X}_{t=0} \sim P_0$$

[Rezende and Shakir Mohamed '15; Dinh, Sohl-Dickstein, and Bengio '16; Chen, Rubanova, Bettencourt, and Duvenaud '18; Albergo and Vanden-Eijnden '23; Lipman, Chen, Ben-Hamu, Nickel, and Le '23; …]

Diffusion-based sampling

SDE

$$d\mathbf{X}_t = \mathbf{b}(\mathbf{X}_t, t)dt + \sqrt{2\epsilon(t)}d\mathbf{Z}_t$$

$$\mathbf{X}_{t=1} \sim \mathbf{Z} \Leftrightarrow \mathbf{X}_{t=0} \sim P_0$$

[Sohl-Dickstein, Weiss, Maheswaranathan, Ganguli '15; Ho, Jain, and Abbeel '20; Song, Ermon '19; Song, Sohl-Dickstein, Kingma, Kumar, Ermon, and Poole '19 …]

Stochastic localisation sampling

SDE

$$\mathbf{X}_{t+\delta t} = \mathbf{X}_t + \mathbb{E}[\mathbf{X}_0 \,|\, \mathbf{X}_t]\delta t + \mathbf{Z}\sqrt{\delta t}$$

$$\mathbf{X}_{t=0} = \delta(\mathbf{X}) \Leftrightarrow \mathbf{X}_{t=\infty} \sim P_0$$

[Eldan '13; Chen and Eldan '22; El Alaoui, Montanari, Sellke '22; Montanari Wu '03….]

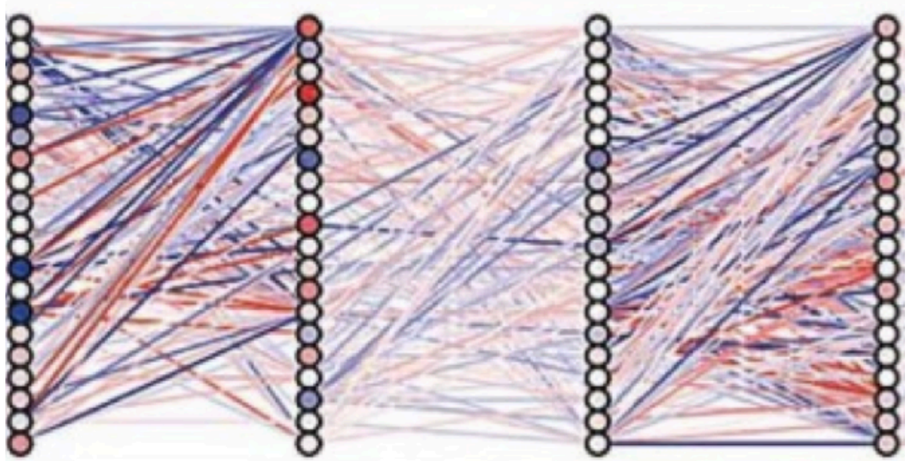
All leads to the same denoising problem

$$P(\mathbf{x} \,|\, \mathbf{x}(t) = \mathbf{X}_t) = \frac{1}{Z(\mathbf{Y}_t)} \exp\left(\frac{\alpha(t)}{\beta(t)^2}\langle \mathbf{X}_t, \mathbf{x}\rangle - \frac{\alpha(t)^2}{2\beta(t)^2}\|\mathbf{x}\|^2\right) P_0(\mathbf{x})$$

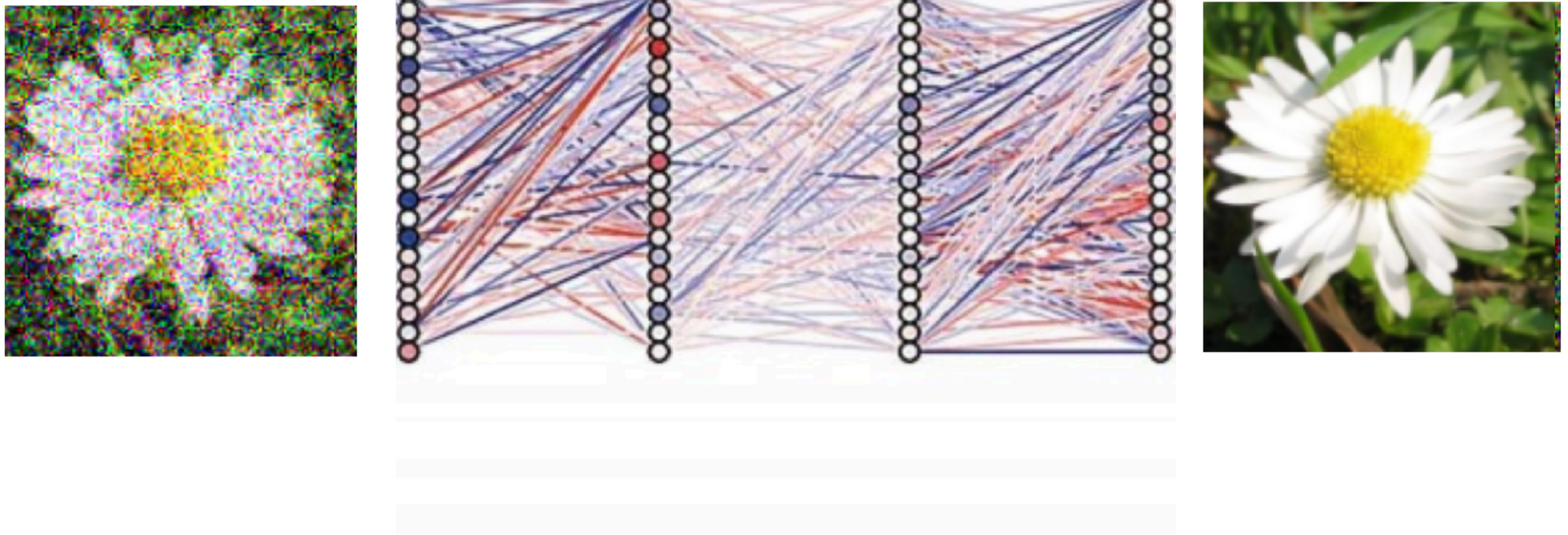
Tilting field

Target measure

In practice:
Successive denoising!

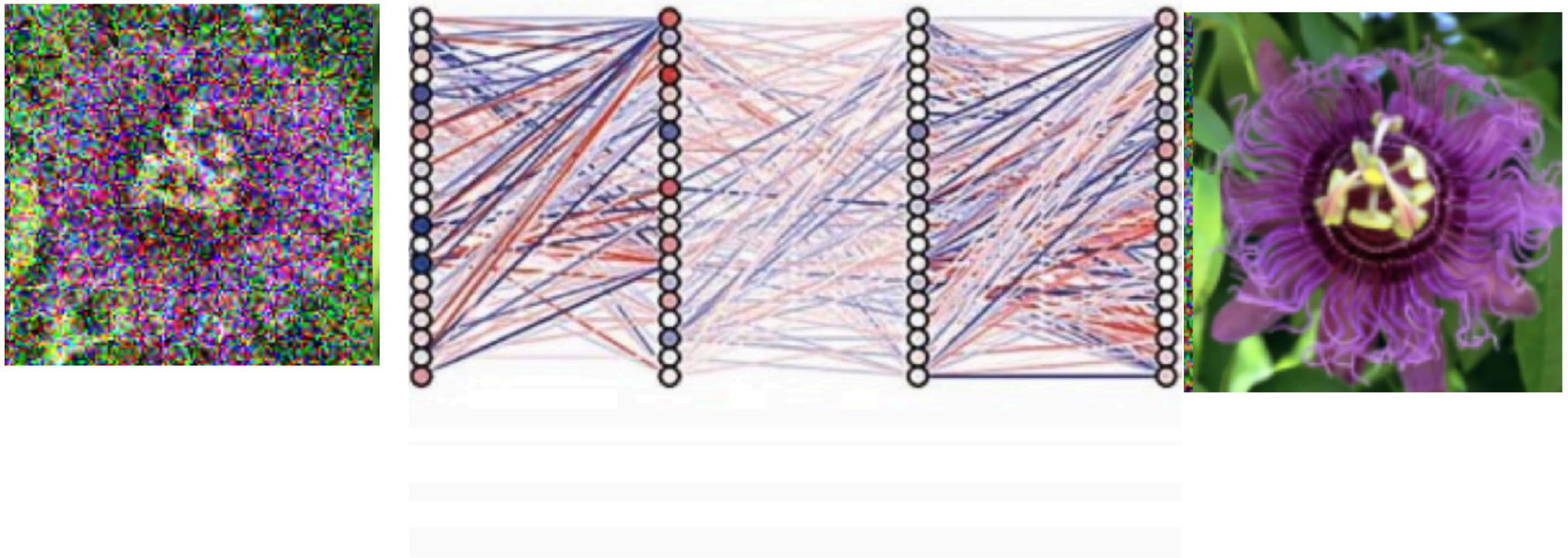


Denoising



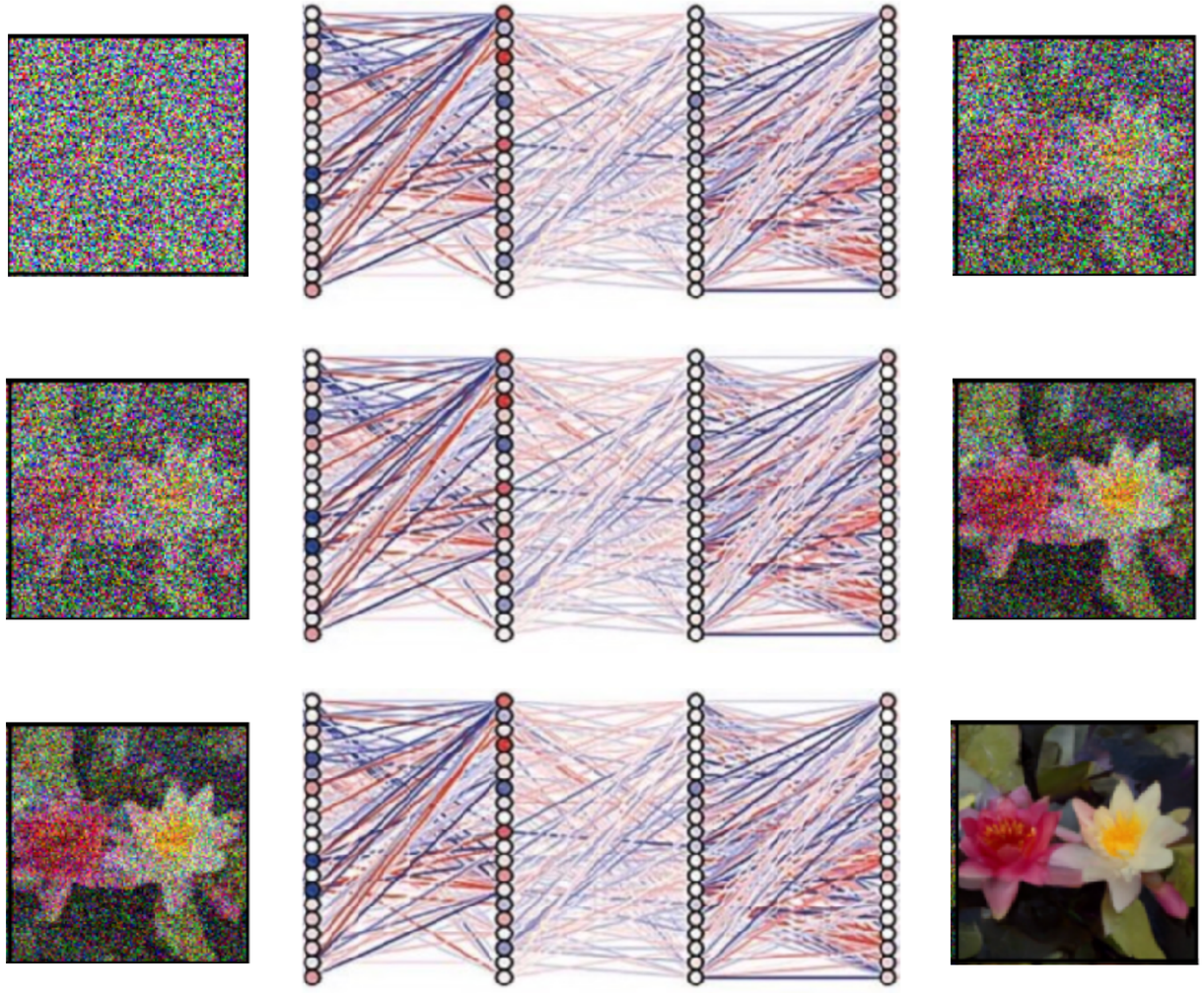
Denoising auto-encoders

Denoising



Denoising auto-encoders

Generating images by Iterative denoising



Generating images by Iterative denoising

Generation d’image par “Nettoyage successif”