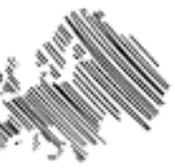
Generative AI and Diffusion Models a Statistical Physics Analysis

Giulio Biroli
&
Marc Mézard

Arxiv & JSTAT 2023



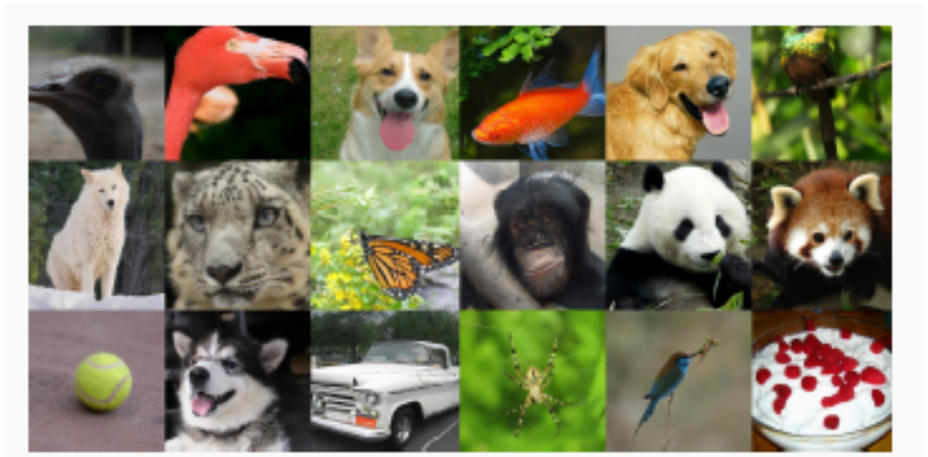
ellis
European Laboratory for Learning and Intelligent Systems

PR[AI]RIE
PaRis Artificial Intelligence Research InstitutE

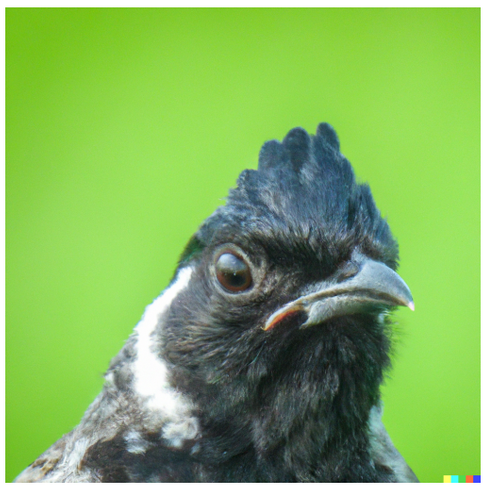
SIMONS FOUNDATION

Generative AI & Diffusion Models

- Impressive results in image and text generation (GPT-4; Dall-E, …)
- Diffusion models are the state of the art in image generation from 2020



- Wide range of applications (text-to-image): Dall-E, Imagen,..



An angry bird

A cat-looking dog

Monkey drinking a cocktail in a speak-easy bar

Diffusion Models

Training set: a set of images $\vec{a}^{\mu} \in \mathbb{R}^N$ $\mu = 1, ..., P$

N is the dimension of the data, P their number

Langevin equation for an Ornstein-Uhlenbeck process

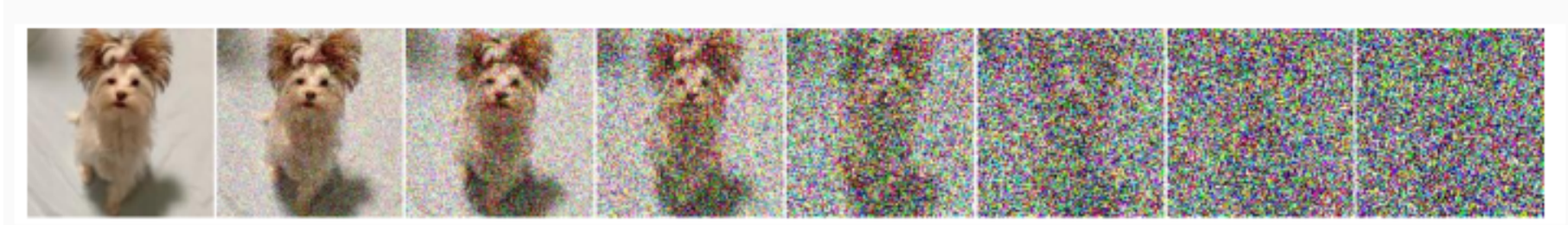
$$\frac{d\vec{x}}{dt} = -\vec{x} + \vec{\eta}(t) \qquad \langle \eta_i(t)\eta_j(t') \rangle = 2T\delta_{ij}\delta(t-t')$$

$$\vec{x}^{\mu}(t=0) = \vec{a}^{\mu}$$

It transforms the data in iid Gaussian $\mathcal{N}(0,1)$ at t>>1

$$P_t(\vec{x}) = \int d\vec{a}\; P_0(\vec{a}) \frac{1}{\sqrt{2\pi\Delta_t}^N} \exp\left(-\frac{1}{2}\frac{(\vec{x}-\vec{a}e^{-t})^2}{\Delta_t}\right) = \int d\vec{a}\; P_t(\vec{a},\vec{x})$$

$$\Delta_t = T(1-e^{-2t})$$



Diffusion models learn how to go backward in time

Generating new images from white noise

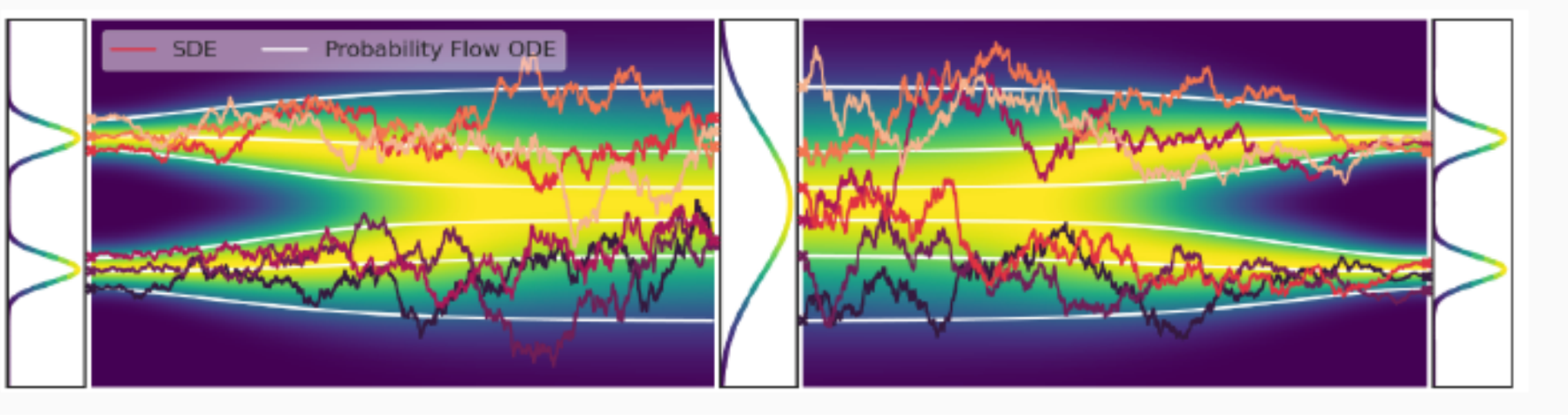
Time-reversal

Score function provides the force field to go back in time

$$\mathcal{F}_i(\vec{x},t) = \frac{\partial \log P_t(\vec{x})}{\partial x_i} \qquad\qquad -\frac{dy_i}{dt} = y_i + 2T\mathcal{F}_i(y,t) + \eta_i(t)$$

Time-reversed Langevin equation transforms iid Gaussians $\mathcal{N}(0,1)$ in new data

$$P_{Gauss}(\vec{x}) \rightarrow P_{data}(\vec{x})$$



Sohl-Dickstein et al 2015 (ideas from out of equilibrium thermodynamics!)
Yang & Ermon 2019, Ho et al 2020, …
For a review see Yang et al “Diffusion Models: A Comprehensive Survey of Methods and Applications

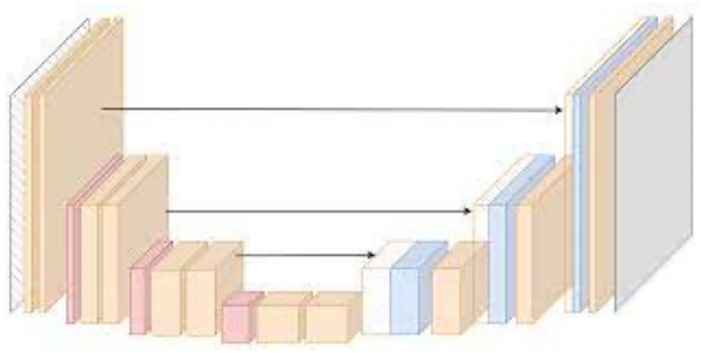
Estimating the score function

Regression problem $\mathcal{F}_i(\vec{x},t) = \dfrac{\partial \log P_t(\vec{x})}{\partial x_i}$

$$P_t(\vec{x}) = \int d\vec{a}\; P_0(\vec{a}) \frac{1}{\sqrt{2\pi\Delta_t}^N} \exp\left(-\frac{1}{2}\frac{(\vec{x}-\vec{a}e^{-t})^2}{\Delta_t}\right) = \int d\vec{a}\; P_t(\vec{a},\vec{x})$$

$$\mathcal{L}(\theta) = \int d\vec{x} P_t(\vec{x}) \left\| \vec{\mathcal{S}}^\theta(\vec{x}) - \vec{\mathcal{F}}(\vec{x},t) \right\|^2$$

Deepnet (UNet)



$$\mathcal{L}(\theta) = \int d\vec{x} P_t(\vec{x}) \left[\vec{\mathcal{S}}^\theta(\vec{x}) \cdot \vec{\mathcal{S}}^\theta(\vec{x}) - 2\vec{\mathcal{S}}^\theta(\vec{x}) \cdot \vec{\mathcal{F}}(\vec{x},t) \right] + C$$

$$\mathcal{L}(\theta) = \int d\vec{x} P_t(\vec{x}) \vec{\mathcal{S}}^\theta(\vec{x}) \cdot \vec{\mathcal{S}}^\theta(\vec{x}) - 2\int d\vec{x} P_t(\vec{x}) \vec{\mathcal{S}}^\theta(\vec{x}) \cdot \frac{\vec{\nabla} P_t(\vec{x})}{P_t(\vec{x})} + C$$

$$\mathcal{L}(\theta) = \mathbb{E}_{x,a} \left\| \vec{\mathcal{S}}^\theta(\vec{x}) + \frac{\vec{x}-\vec{a}e^{-t}}{\Delta_t} \right\|^2 + C$$

Empirical minimisation

$$\mathcal{L}(\theta(t)) = \sum_{x_\mu(t), a_\mu} \left\| \vec{S}^{\theta(t)}(\vec{x}_\mu(t)) + \frac{\vec{x}_\mu(t) - \vec{a}_\mu e^{-t}}{\Delta_t} \right\|^2$$

Theoretical (math) Results

- Several results by De Bortoli et al. (2012, 2022)

Informally

If $|\vec{\mathcal{S}}^{\theta}(\vec{x}) - \vec{\mathcal{F}}(\vec{x}, t)| < \epsilon$ and the distribution of data, π, is regular enough then there exists positive constants B,C,D such that

$$\|\vec{\mathcal{P}}^{\theta}(\vec{x}_B(T)) - \pi\|_{TV} < Be^{-T} + C(\epsilon + \Delta t^{1/2})e^{DT}$$

Generated distribution

Discretisation step

A power law in T if data lie on a compact manifold

Open theoretical questions

What is the role of dimension of the data ?

How many data one needs to get a good diffusion model ?

What is the role of the approximation class (number of parameters, form of the network,..) ?

A statphys analysis of diffusion models in
the high dimension - large number of data limit

High-dimensional Gaussian Data

N is the dimension of the data, P their number and we consider the limit $N, P \to \infty$

The vector of data is multivariate Gaussian $\vec{a} \sim \mathcal{N}(0, C^0)$ where the density of eigenvalues of C^0 converges to a function $\rho(\lambda)$

$$\frac{d\vec{x}}{dt} = -\vec{x} + \vec{\eta}(t) \qquad \langle \eta_i(t)\eta_j(t')\rangle = 2T\delta_{ij}\delta(t-t')$$

$$P_t(\vec{x}) = \int d\vec{a}\; P_0(\vec{a}) \frac{1}{\sqrt{2\pi\Delta_t}^N} \exp\left(-\frac{1}{2}\frac{(\vec{x}-\vec{a}e^{-t})^2}{\Delta_t}\right) = \int d\vec{a}\; P_t(\vec{a},\vec{x})$$

Gaussian

$$\mathcal{F}_i(\vec{x},t) = \frac{\partial \log P_t(\vec{x})}{\partial x_i}$$

Linear score

$$\mathcal{F}_i(x,t) = -\sum_j W_{ij}x_j \qquad W_{true} = \left(T(1-e^{-2t})\mathbb{I} + e^{-2t}C^0\right)^{-1}$$

Empirical estimation of the score

$$\mathcal{S}_i(x,t) = -\sum_j W_{ij} x_j \longrightarrow$$

$$W_{emp} = \frac{1}{T(1-e^{-2t})}\left(\mathbb{I} - e^{-t}(D^t)^{-1}M^t\right)$$

$$D^t_{ij} = \frac{1}{P}\sum_\mu x^\mu_i(t)x^\mu_j(t)$$

$$M^t_{ij} = \frac{1}{2P}\sum_\mu \left(x^\mu_i(t)x^\mu_j(0) + x^\mu_i(0)x^\mu_j(t)\right)$$

Minimize the empirical loss

$$\mathcal{L}(\theta(t)) = \sum_{x_\mu(t), a_\mu} \left\| \vec{S}^{\theta(t)}(\vec{x}_\mu(t)) + \frac{\vec{x}_\mu(t) - \vec{a}_\mu e^{-t}}{\Delta_t} \right\|^2$$

What is the role of dimension of the data ?

How many data one needs to get a good diffusion model ?

A random matrix theory problem

How much data is needed for a good score estimation depends on the dimension

Key idea $$D^t_{ij} = \frac{1}{P}\sum_{\mu} x^{\mu}_i(t)x^{\mu}_j(t) \qquad D^t_{ij} = C^t_{ij} + \delta C^t_{ij}$$

For large P $\longrightarrow$ $$\delta D^t_{ij} \sim O\left(\frac{1}{\sqrt{P}}\right) = \sqrt{\frac{N}{P}}R_{ij}$$

$$D^t = C^t + \sqrt{\frac{N}{P}}R$$

A generalized Rosenzweig-Porter random matrix model

Kratsov et al 2015, Facoetti, GB, Vivo, 2016, Benigni 2018,…

For $P \gg N$ the density of eigenvalues (but not eigenvectors) is correctly estimated

For $P \gg N^2$ eigenvalues and eigenvectors are correctly estimated

Number of Data vs Dimension

The diffusion model generate Gaussian data $\vec{x}_{DM} \sim \mathcal{N}(0, C_{DM})$

First regime: $P \ll N$

C_{DM} is different from C^0: wrong estimation and bad generative model.

Second regime: $P \gg N$

C_{DM} has the same density of eigenvalues of C^0 but different eigenvectors.

$\frac{1}{N}\langle \vec{x} \cdot \vec{x} \rangle$ is correct but directions of fluctuations are not.

2-Wasserstein distance between the true process and the generated one vanishes.

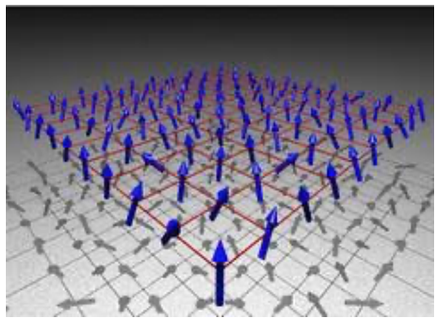
Third regime: $P \gg N^2$

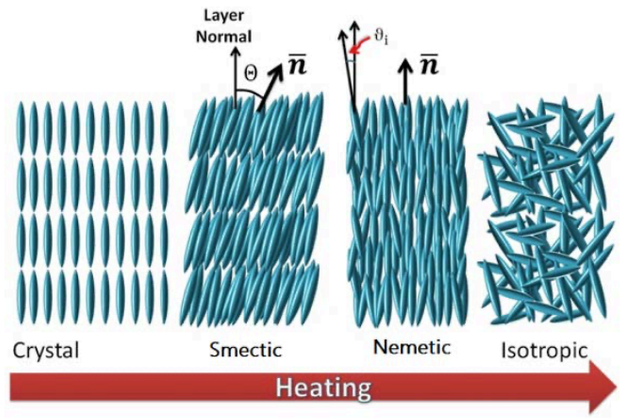
C_{DM} has the same eigenvalues and eigenvectors of C^0.

Total variation distance between the true process and the generated one vanishes.

High-dimensional Data from Statistical Physics

The data comes from the low temperature phase of a statphys model



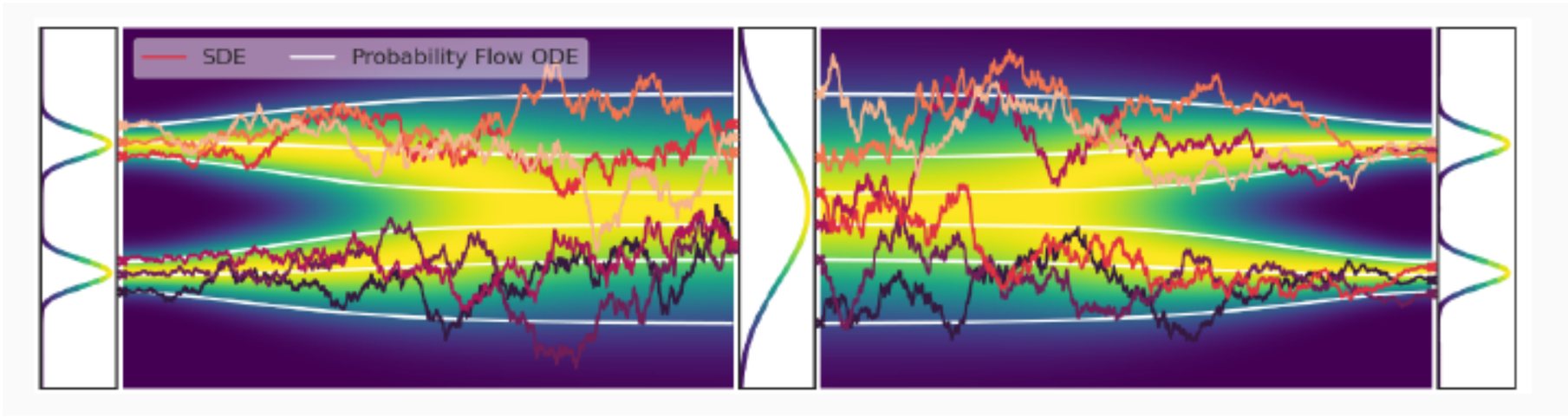


$$P(\vec{a}) \sim e^{-\beta H(\vec{a})}$$

We consider low-temperature cases in which the symmetry is broken and there are multiple pure states

Different phases ~ different classes

How the diffusion model breaks the symmetry and generate typical configurations



An exactly solvable model

Curie-Weiss Model of ferromagnetism $P(\vec{a}) = \frac{1}{Z} \exp\left(\frac{\beta}{2N} \sum_{i \neq j} a_i a_j + \frac{h}{N} \sum_i a_i \right)$

Ferromagnetic phase transition: 2 states with magnetisation $\pm m^*(\beta)$

The weights of the two states are $W_\pm = \frac{e^{\pm h m^*}}{2\cosh(h m^*)}$

How the diffusion model generates the symmetry breaking ?

How the weights of the states are reconstructed ?

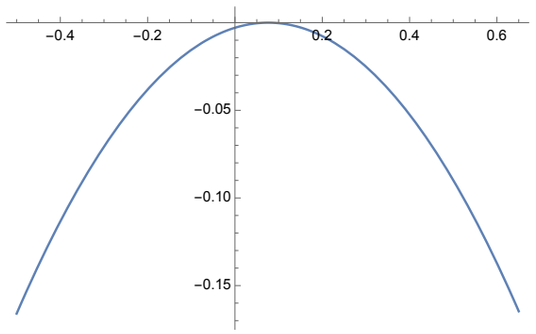
An exactly solvable model

Exact computation of the score and analysis of the stochastic process

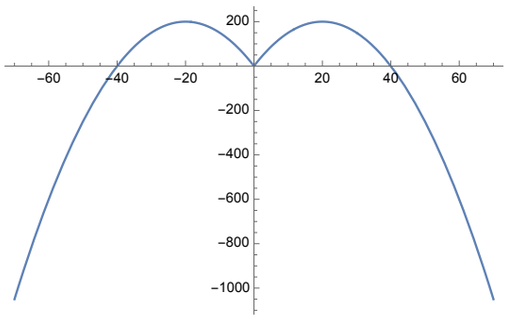
Beginning of the backward process $\mu(t) = (1/\sqrt{N}) \sum_i x_i \sim O(1)$

$$\frac{d\mu}{dt} = -\frac{dV}{d\mu} + \eta(t)$$

$\sqrt{N}e^{-t} \ll 1$



$\sqrt{N}e^{-t} \gg 1$



$$V(\mu) = -\frac{1}{2}\left[\mu - m^* \tanh(hm^*)\frac{\sqrt{N}e^{-t}}{T}\right]^2 \qquad V(\mu) = -\frac{1}{2}\mu^2 + 2m^*\sqrt{N}e^{-t}|\mu|$$

Symmetry breaking of the backward process at $\sqrt{N}e^{-t} \sim O(1)$

Weights of the two states reconstructed in the first time regime

General argument

Backward process

$$-\frac{dy_i}{dt} = y_i + 2T\mathcal{F}_i(y,t) + \eta_i(t)$$

The score can be related to the free-energy in an external field z

$$\mathcal{F}(x)_i = -\frac{x_i}{\Delta_t} - \frac{e^{-t}}{\Delta_t}\left.\frac{\partial F}{\partial z_i}\right|_{z_i=\frac{x_i e^{-t}}{\beta\Delta_t}}$$

In presence of multiple pure states with different weights at the beginning of the B.P.

$$e^{-\beta F} = \sum_\alpha e^{-\beta F_\alpha + \beta\sum_i z_i m_i^\alpha} \longrightarrow \mathcal{F}(x)_i = -\frac{x_i}{\Delta_t} + \frac{\partial}{\partial x_i}\ln\left(\sum_\alpha w_\alpha e^{\mu_\alpha(x)\sqrt{N}\frac{e^{-t}}{\Delta_t}}\right)$$

Symmetry breaking of the backward process at $\sqrt{N}e^{-t} \sim O(1)$

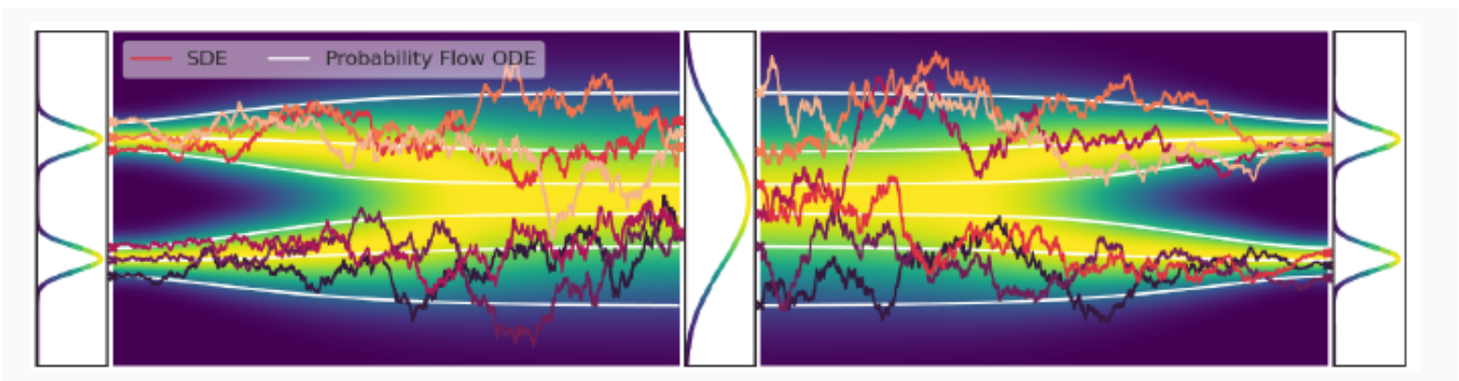


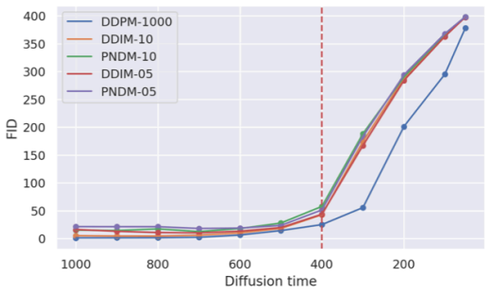
Image generation

- Numerical evidence that the weights of the classes are generated at the beginning

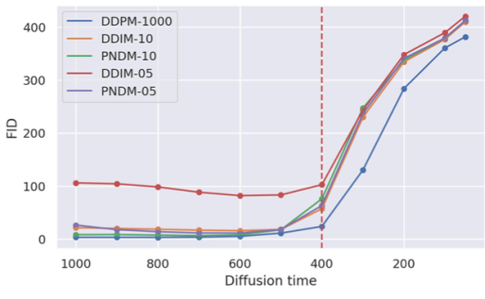
  De Bortoli (2022)

- Numerical evidence of two time regimes and “symmetry breaking”

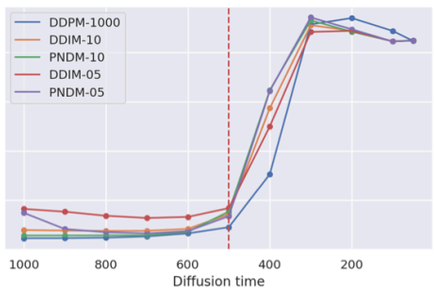
  Raya, Ambrogioni 2023



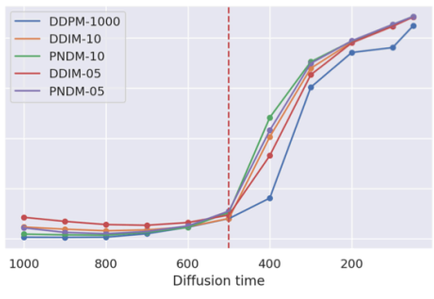
(a) MNIST



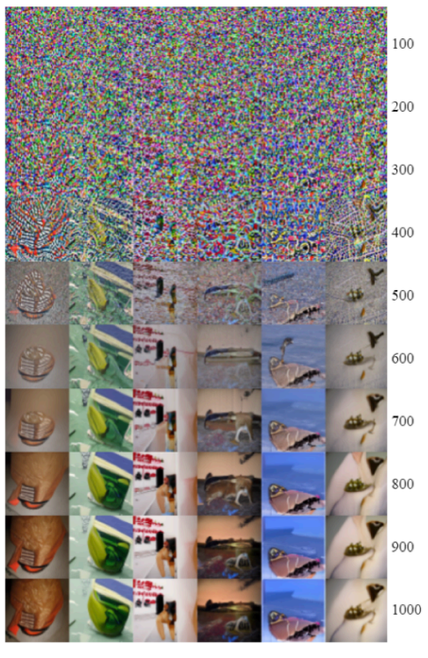
(b) CIFAR10



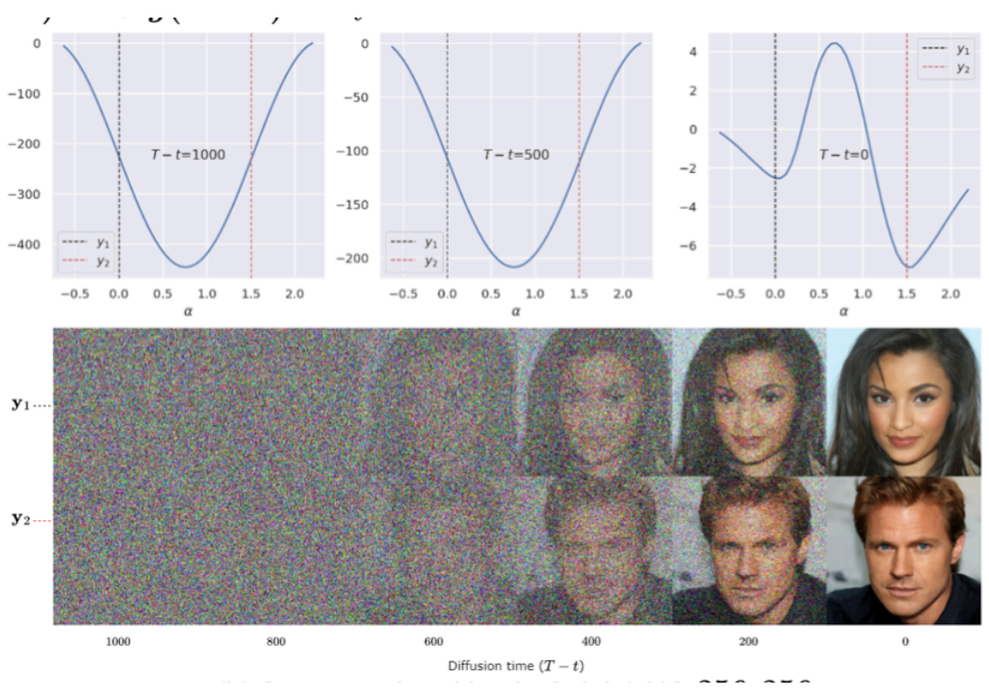
(c) Imagenet64



(d) CelebA64



(e) Imagenet late start generation



Conclusion: a first study of high-dimensional DM

- Two different time-regimes and symmetry breaking during the generation process
- Competition between number of data and dimension of data to get an efficient diffusion model
- Different regimes in number of data vs number of dimension & accuracy in high-D

Perspectives

- Study the role of the approximation class, e.g. a MLP for the score
- Study the competition between the number of data and dimension of data in more complex cases (CW, mixture of Gaussians,..)
- Study realistic applications and problems (inpainting, “copyright problem”,…)