Predictive Modeling of Fluid Flows Using Conditional Score-Based Diffusion Models

Wilfried Genuist[1,2], **Éric Savin**[1], **Filippo Gatti**[2], **Didier Clouteau**[2]

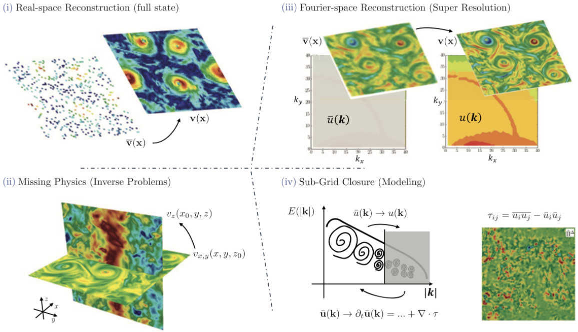
[1]Université Paris-Saclay, ONERA, 91120 Palaiseau, France
[2]Université Paris-Saclay, CentraleSupélec, ENS Paris-Saclay, CNRS, LMPS-Laboratoire de Mécanique Paris-Saclay, 91190 Gif-sur-Yvette, France

`wilfried.genuist@{centralesupelec,onera}.fr`

Sorbonne Université, Paris, France, December 19th, 2024

Using Machine Learning in CFD

- Accelerating direct numerical simulations: improving discretization schemes, correcting coarse-resolution simulations, reducing the computational domain...
- Improving turbulence closure models (RANS), subgrid-scale models (LES), wall models, ROMs, DMD;
- Reconstructing complex flows: filling space-wave number and/or time-frequency gaps (*e.g.* super-resolution), inverse problems in optimization or control...



From M. Buzzicotti, *Europhys. Lett.* **142**, 23001 (2023)

Machine learning layout

- Supervised learning: $\hat{\mathbf{y}} \simeq \arg\max_{\theta} p_{\theta}(\mathbf{y}|\mathbf{X} = \hat{\mathbf{x}})$ where $p_{\theta}(\mathbf{y}|\mathbf{x})$ is trained from a dataset $\{\mathbf{X}_i, \mathbf{Y}_i\}_{i=1}^{N}$;
- Unsupervised learning: $\hat{\mathbf{x}} \simeq \arg\max_{\theta} p_{\theta}(\mathbf{x})$ where $p_{\theta}(\mathbf{x})$ is trained from a dataset $\{\mathbf{X}_i\}_{i=1}^{N}$;
- Unsupervised learning can be made supervised by conditioning the inference $\hat{\mathbf{x}} \simeq \arg\max_{\theta} p_{\theta}(\mathbf{x}|\mathbf{Y} = \hat{\mathbf{y}})$ whenever $p_{\theta}(\mathbf{x}, \mathbf{y})$ is trained from a dataset $\{\mathbf{X}_i, \mathbf{Y}_i\}_{i=1}^{N}$;
- Generative adversarial networks (GANs), variational auto-encoders (VAEs), normalizing flows, generative diffusion models... aim at computing and sampling complex **conditional distributions** on (very) high-dimensional sets;
- **This talk**: predicting missing data in fluid flows by conditional inference using generative diffusion models.

Generative modeling

- **Step #1**: modeling $\{p_\theta(\mathbf{x})\}_\theta$ or $\{p_\theta(\mathbf{x}|\mathbf{y})\}_\theta$–typically Gibbs distributions $p_\theta(\mathbf{x}) = \mathrm{e}^{F_\theta - U_\theta(\mathbf{x})}$ with internal energy $U_\theta(\mathbf{x})$ and free energy F_θ, or stationary distributions of Markov processes, beyond Gaussian distributions $U_\theta(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{C}_\theta^{-1}\mathbf{x}$ which are usually not able to capture high-dimensional complex, structured data;
- **Step #2**: optimizing $\hat{\theta} = \arg\min_\theta D(p, p_\theta)$ for some relevant "distance" D between the true distribution p and the parametric model p_θ–typically the Kullback-Leibler divergence D_{KL}:

$$D_{\mathrm{KL}}(p, p_\theta) = \mathbb{E}_p\{\log p - \log p_\theta\} \geq 0\,;$$

- **Step #3**: inferring from $p_{\hat{\theta}}$–typically transforming it into a simple distribution, *i.e.* Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$, and reversing the transformation.

Objectives and dataset

- **Fast Inference**: diffusion models enable significantly faster inference compared to traditional PDE solvers.
- **Multi-scale and multi-domain turbulence**: ability to capture turbulent features across various spatial and temporal scales, adaptable to different application contexts.
- **Extrapolation capabilities**: diffusion models provide improved generalization to unseen turbulent configurations or extrapolation to spatial regions not covered by training data.
- **Prediction stability**: enhanced numerical stability of predictions compared to traditional solvers constrained by time-step and resolution limitations.
- **Reduced computational cost**: Reduction in computation costs for complex predictions, especially for high-dimensional datasets.
- **Uncertainty modeling**: Natural integration of uncertainty estimation through the stochastic process inherent in diffusion models.

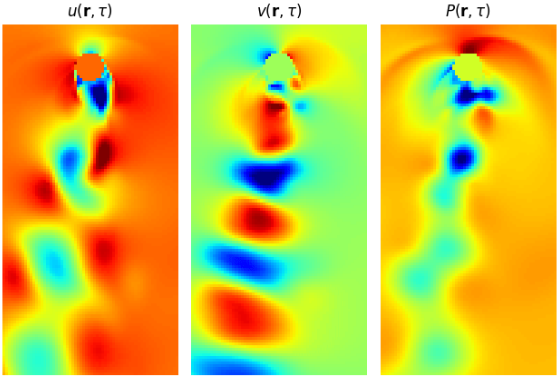
The training and test data (Kohl et al. [6]) address the case of von Kármán vortex around a 2D cylinder (compressible + transonic) for Reynolds number $Re = 10^4$ and a Mach number $M \in [0.53, 0.63] \cup [0.69, 0.90]$.

- Each image has a resolution of $R = 128 \times 64$ pixels for velocity fields $\mathbf{u} = (u, v)$ and pressure P over a spatial domain Ω at time steps $\tau \in \{0, 1, \ldots, \mathcal{T}\}$;

Objectives and dataset

The training and test data (Kohl et al. [6]) address the case of von Kármán vortex around a 2D cylinder (compressible + transonic) for Reynolds number $Re = 10^4$ and a Mach number $M \in [0.53, 0.63] \cup [0.69, 0.90]$.

- Each image has a resolution of $R = 128 \times 64$ pixels for velocity fields $\mathbf{u} = (u, v)$ and pressure P over a spatial domain Ω at time steps $\tau \in \{0, 1, \dots, \mathcal{T}\}$;



Objectives and dataset

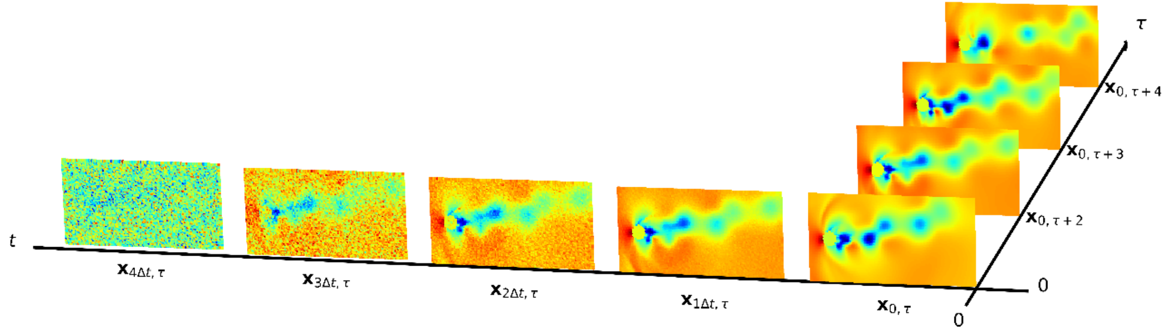
The training and test data (Kohl et al. [6]) address the case of von Kármán vortex around a 2D cylinder (compressible + transonic) for Reynolds number $Re = 10^4$ and a Mach number $M \in [0.53, 0.63] \cup [0.69, 0.90]$.

- Each image has a resolution of $R = 128 \times 64$ pixels for velocity fields $\mathbf{u} = (u, v)$ and pressure P over a spatial domain Ω at time steps $\tau \in \{0, 1, \ldots, \mathcal{T}\}$;
- We write $\mathbf{x}_{0,\tau} = \{\mathbf{u}(\mathbf{r}_j, \tau), P(\mathbf{r}_j, \tau) \mid \mathbf{r}_j \in \Omega, 1 \leq j \leq R, 0 \leq \tau \leq \mathcal{T}\}$ (sample at time τ);

Objectives and dataset

The training and test data (Kohl et al. [6]) address the case of von Kármán vortex around a 2D cylinder (compressible + transonic) for Reynolds number $Re = 10^4$ and a Mach number $M \in [0.53, 0.63] \cup [0.69, 0.90]$.

- Each image has a resolution of $R = 128 \times 64$ pixels for velocity fields $\mathbf{u} = (u, v)$ and pressure P over a spatial domain Ω at time steps $\tau \in \{0, 1, \ldots, \mathcal{T}\}$;
- We write $\mathbf{x}_{0,\tau} = \{\mathbf{u}(\mathbf{r}_j, \tau), P(\mathbf{r}_j, \tau) \mid \mathbf{r}_j \in \Omega, 1 \leq j \leq R, 0 \leq \tau \leq \mathcal{T}\}$ (sample at time τ);



Objectives and dataset

The training and test data (Kohl et al. [6]) address the case of von Kármán vortex around a 2D cylinder (compressible + transonic) for Reynolds number $Re = 10^4$ and a Mach number $M \in [0.53, 0.63] \cup [0.69, 0.90]$.

- Each image has a resolution of $R = 128 \times 64$ pixels for velocity fields $\mathbf{u} = (u, v)$ and pressure P over a spatial domain Ω at time steps $\tau \in \{0, 1, \ldots, \mathcal{T}\}$;
- We write $\mathbf{x}_{0,\tau} = \{\mathbf{u}(\mathbf{r}_j, \tau), P(\mathbf{r}_j, \tau) \mid \mathbf{r}_j \in \Omega, 1 \leq j \leq R, 0 \leq \tau \leq \mathcal{T}\}$ (sample at time τ);
- $\mathcal{D} = \{\mathbf{x}_{0,0}, \mathbf{x}_{0,1}, \ldots, \mathbf{x}_{0,\mathcal{T}}\}$ with $\mathcal{T} = 1000$, denotes the dataset consisting of regularly sampled images for a given Mach number M, initial conditions $\mathcal{I}$ and boundary conditions $\mathcal{B}$;

Objectives and dataset

The training and test data (Kohl et al. [6]) address the case of von Kármán vortex around a 2D cylinder (compressible + transonic) for Reynolds number $Re = 10^4$ and a Mach number $M \in [0.53, 0.63] \cup [0.69, 0.90]$.

- Each image has a resolution of $R = 128 \times 64$ pixels for velocity fields $\mathbf{u} = (u, v)$ and pressure P over a spatial domain Ω at time steps $\tau \in \{0, 1, \dots, \mathcal{T}\}$;
- We write $\mathbf{x}_{0,\tau} = \{\mathbf{u}(\mathbf{r}_j, \tau), P(\mathbf{r}_j, \tau) \mid \mathbf{r}_j \in \Omega, 1 \leq j \leq R, 0 \leq \tau \leq \mathcal{T}\}$ (sample at time τ);
- $\mathcal{D} = \{\mathbf{x}_{0,0}, \mathbf{x}_{0,1}, \dots, \mathbf{x}_{0,\mathcal{T}}\}$ with $\mathcal{T} = 1000$, denotes the dataset consisting of regularly sampled images for a given Mach number M, initial conditions $\mathcal{I}$ and boundary conditions $\mathcal{B}$;

Generative modeling assumption: $\mathbf{x}_{0,\tau} \sim p_{\text{data}} \equiv p_0$

Objectives and dataset

The training and test data (Kohl et al. [6]) address the case of von Kármán vortex around a 2D cylinder (compressible + transonic) for Reynolds number $Re = 10^4$ and a Mach number $M \in [0.53, 0.63] \cup [0.69, 0.90]$.

- Each image has a resolution of $R = 128 \times 64$ pixels for velocity fields $\mathbf{u} = (u, v)$ and pressure P over a spatial domain Ω at time steps $\tau \in \{0, 1, \ldots, \mathcal{T}\}$;
- We write $\mathbf{x}_{0,\tau} = \{\mathbf{u}(\mathbf{r}_j, \tau), P(\mathbf{r}_j, \tau) \mid \mathbf{r}_j \in \Omega, 1 \leq j \leq R, 0 \leq \tau \leq \mathcal{T}\}$ (sample at time τ);
- $\mathcal{D} = \{\mathbf{x}_{0,0}, \mathbf{x}_{0,1}, \ldots, \mathbf{x}_{0,\mathcal{T}}\}$ with $\mathcal{T} = 1000$, denotes the dataset consisting of regularly sampled images for a given Mach number M, initial conditions $\mathcal{I}$ and boundary conditions $\mathcal{B}$;

Generative modeling assumption: $\mathbf{x}_{0,\tau} \sim p_{\text{data}} \equiv p_0$

Objective:

$$\hat{\mathbf{x}}_{0,\tau} = \arg\max_{\mathbf{x}_{0,\tau}} p_0(\mathbf{x}_{0,\tau} | \tau, \mathcal{I}(\mathbf{x}_{0,0}), \mathcal{B}(\mathbf{x}_{0,\tau}))\,.$$

- Based on VAE concept and ancestral sampling techniques;

- Based on VAE concept and ancestral sampling techniques;
- Denoising diffusion probabilistic models (DDPMs, Ho et al. [5]) are composed of two Markov chains:

- Based on VAE concept and ancestral sampling techniques;
- Denoising diffusion probabilistic models (DDPMs, Ho et al. [5]) are composed of two Markov chains:
  - Forward chain (forward process): perturb data with noise,
  - Backward chain (reverse process): denoising chain to go back to data;

- Based on VAE concept and ancestral sampling techniques;
- Denoising diffusion probabilistic models (DDPMs, Ho et al. [5]) are composed of two Markov chains:
  - Forward chain (forward process): perturb data with noise,
  - Backward chain (reverse process): denoising chain to go back to data;
- The forward process is designed to transform data samples to a chosen (simple) prior distribution, typically standard normal Gaussian distribution, *i.e.* $p_{\text{ref}} = \mathcal{N}(\mathbf{0}, \mathbf{I})$;

- Based on VAE concept and ancestral sampling techniques;
- Denoising diffusion probabilistic models (DDPMs, Ho et al. [5]) are composed of two Markov chains:
  - Forward chain (forward process): perturb data with noise,
  - Backward chain (reverse process): denoising chain to go back to data;
- The forward process is designed to transform data samples to a chosen (simple) prior distribution, typically standard normal Gaussian distribution, *i.e.* $p_{\text{ref}} = \mathcal{N}(\mathbf{0}, \mathbf{I})$;
- The reverse process is built by learning a transition kernel;

Diffusion models: the discrete setting

Introduction

- Based on VAE concept and ancestral sampling techniques;
- Denoising diffusion probabilistic models (DDPMs, Ho et al. [5]) are composed of two Markov chains:
  - Forward chain (forward process): perturb data with noise,
  - Backward chain (reverse process): denoising chain to go back to data;
- The forward process is designed to transform data samples to a chosen (simple) prior distribution, typically standard normal Gaussian distribution, *i.e.* $p_{\text{ref}} = \mathcal{N}(\mathbf{0}, \mathbf{I})$;
- The reverse process is built by learning a transition kernel;
- Samples are generated by sampling the prior distribution and follow the backward Markov chain:

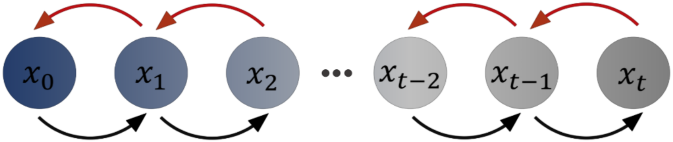
Diffusion models: the discrete setting

Introduction

- Based on VAE concept and ancestral sampling techniques;
- Denoising diffusion probabilistic models (DDPMs, Ho et al. [5]) are composed of two Markov chains:
  - ☐ Forward chain (forward process): perturb data with noise,
  - ☐ Backward chain (reverse process): denoising chain to go back to data;
- The forward process is designed to transform data samples to a chosen (simple) prior distribution, typically standard normal Gaussian distribution, *i.e.* $p_{\text{ref}} = \mathcal{N}(\mathbf{0}, \mathbf{I})$;
- The reverse process is built by learning a transition kernel;
- Samples are generated by sampling the prior distribution and follow the backward Markov chain:

$$\overbrace{p^{\theta}_{t-1|t}(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}^{\theta}_t(\mathbf{x}_t), \boldsymbol{\Sigma}^{\theta}_t(\mathbf{x}_t))}$$

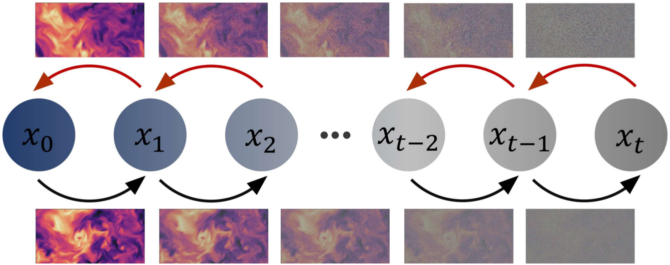
$\mathbf{x}_0 \sim p_{\text{data}}$



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

$$\underbrace{q_{t|0}(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha}_t)\mathbf{I})}$$

- Based on VAE concept and ancestral sampling techniques;
- Denoising diffusion probabilistic models (DDPMs, Ho et al. [5]) are composed of two Markov chains:
  - Forward chain (forward process): perturb data with noise,
  - Backward chain (reverse process): denoising chain to go back to data;
- The forward process is designed to transform data samples to a chosen (simple) prior distribution, typically standard normal Gaussian distribution, *i.e.* $p_{\text{ref}} = \mathcal{N}(\mathbf{0}, \mathbf{I})$;
- The reverse process is built by learning a transition kernel;
- Samples are generated by sampling the prior distribution and follow the backward Markov chain:



Let $\{\mathbf{x}_t;\ 0 \leq t < \infty\}$ be a random process defined on $\mathbb{R}^p$ and indexed on $t \in \mathbb{R}_+$.

- $\mathbf{x}_t$ verifies the Itō Stochastic Differential Equation (SDE) (Oksendal [7]) describing the evolution of $\mathbf{x}_0$ to a state $\mathbf{x}_T$:

$$\mathrm{d}\mathbf{x}_t = \boldsymbol{\mu}(\mathbf{x}_t, t)\mathrm{d}t + \boldsymbol{\sigma}(\mathbf{x}_t, t)\mathrm{d}\mathbf{W}_t\,, \quad \mathbf{x}_0 \sim p_0\,, \tag{1}$$

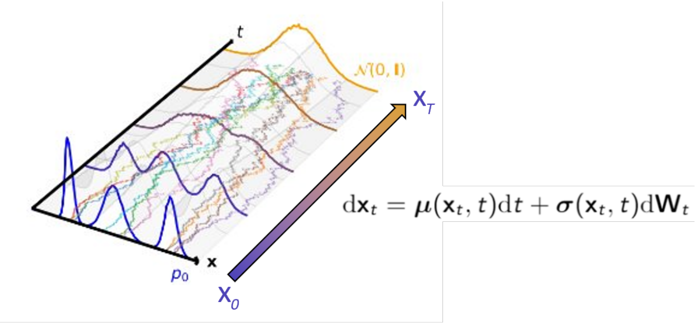
  where $\boldsymbol{\mu}(\cdot, t) : \mathbb{R}^p \to \mathbb{R}^p$ is the drift coefficient, $\boldsymbol{\Sigma}(\cdot, t) : \mathbb{R}^p \to \mathbb{R}^{p\times p}$ is the diffusion coefficient and $\{\mathbf{W}_t;\ t \geq 0\}$ is a p-dimensional Brownian motion (or Wiener process).

Let $\{\mathbf{x}_t;\, 0 \leq t < \infty\}$ be a random process defined on $\mathbb{R}^p$ and indexed on $t \in \mathbb{R}_+$.

- $\mathbf{x}_t$ verifies the Itō Stochastic Differential Equation (SDE) (Oksendal [7]) describing the evolution of $\mathbf{x}_0$ to a state $\mathbf{x}_T$:

$$\mathrm{d}\mathbf{x}_t = \boldsymbol{\mu}(\mathbf{x}_t, t)\mathrm{d}t + \boldsymbol{\sigma}(\mathbf{x}_t, t)\mathrm{d}\mathbf{W}_t\,, \quad \mathbf{x}_0 \sim p_0\,, \tag{1}$$

where $\boldsymbol{\mu}(\cdot, t) : \mathbb{R}^p \to \mathbb{R}^p$ is the drift coefficient, $\boldsymbol{\Sigma}(\cdot, t) : \mathbb{R}^p \to \mathbb{R}^{p\times p}$ is the diffusion coefficient and $\{\mathbf{W}_t;\, t \geq 0\}$ is a p-dimensional Brownian motion (or Wiener process).



Let $\{\mathbf{x}_t;\, 0 \leq t < \infty\}$ be a random process defined on $\mathbb{R}^p$ and indexed on $t \in \mathbb{R}_+$.

- Time reversal of Eq. (1) $t \to T - t$ (Anderson [1], Haussmann and Pardoux [4]), leading to the reverse-time SDE of the diffusion process:

$$\mathrm{d}\mathbf{x}_t = \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, t)\mathrm{d}t + \boldsymbol{\sigma}(\mathbf{x}_t, t)\mathrm{d}\overline{\mathbf{W}}_t\,, \quad \mathbf{x}_T \sim p_T\,,$$

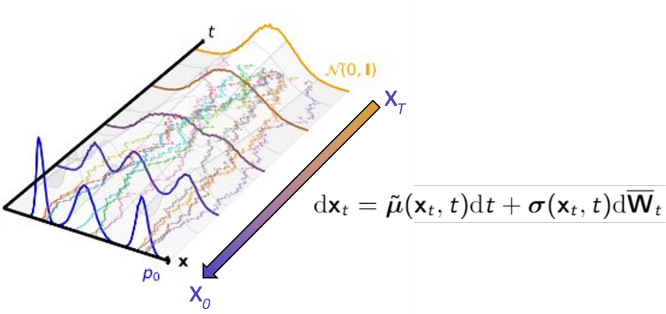
where $\tilde{\boldsymbol{\mu}}(\cdot, t) = \boldsymbol{\mu}(\cdot, t) - \boldsymbol{\nabla}_{\mathbf{x}} \cdot \boldsymbol{\Sigma}(\cdot, t) - \boldsymbol{\Sigma}(\cdot, t)\boldsymbol{\nabla}_{\mathbf{x}} \log p_t(\cdot)$, $\boldsymbol{\Sigma}(\cdot, t) = \boldsymbol{\sigma}(\cdot, t)\boldsymbol{\sigma}(\cdot, t)^{\mathsf{T}}$, p_t is the marginal probability density function of $\mathbf{x}_t$, and $(\overline{\mathbf{W}}_t)_{t\geq 0}$ is a time reversed Brownian motion.

Let $\{\mathbf{x}_t;\, 0 \leq t < \infty\}$ be a random process defined on $\mathbb{R}^p$ and indexed on $t \in \mathbb{R}_+$.

- Time reversal of Eq. (1) $t \to T - t$ (Anderson [1], Haussmann and Pardoux [4]), leading to the reverse-time SDE of the diffusion process:

$$\mathrm{d}\mathbf{x}_t = \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, t)\mathrm{d}t + \boldsymbol{\sigma}(\mathbf{x}_t, t)\mathrm{d}\overline{\mathbf{W}}_t\,, \quad \mathbf{x}_T \sim p_T\,,$$

where $\tilde{\boldsymbol{\mu}}(\cdot, t) = \boldsymbol{\mu}(\cdot, t) - \nabla_{\mathbf{x}} \cdot \boldsymbol{\Sigma}(\cdot, t) - \boldsymbol{\Sigma}(\cdot, t)\nabla_{\mathbf{x}} \log p_t(\cdot)$, $\boldsymbol{\Sigma}(\cdot, t) = \boldsymbol{\sigma}(\cdot, t)\boldsymbol{\sigma}(\cdot, t)^{\mathsf{T}}$, p_t is the marginal probability density function of $\mathbf{x}_t$, and $(\overline{\mathbf{W}}_t)_{t \geq 0}$ is a time reversed Brownian motion.



Diffusion models: the continuous setting

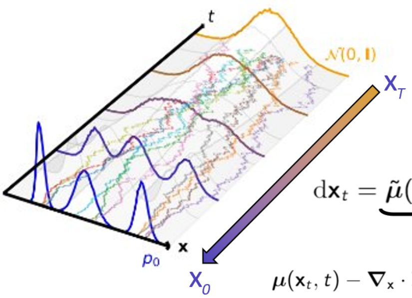
Forward-backward system and SDEs

Let $\{\mathbf{x}_t;\ 0 \leq t < \infty\}$ be a random process defined on $\mathbb{R}^p$ and indexed on $t \in \mathbb{R}_+$.

- Time reversal of Eq. (1) $t \to T - t$ (Anderson [1], Haussmann and Pardoux [4]), leading to the reverse-time SDE of the diffusion process:

$$\mathrm{d}\mathbf{x}_t = \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, t)\mathrm{d}t + \boldsymbol{\sigma}(\mathbf{x}_t, t)\mathrm{d}\overline{\mathbf{W}}_t\,, \quad \mathbf{x}_T \sim p_T\,,$$

where $\tilde{\boldsymbol{\mu}}(\cdot, t) = \boldsymbol{\mu}(\cdot, t) - \boldsymbol{\nabla}_{\mathbf{x}} \cdot \boldsymbol{\Sigma}(\cdot, t) - \boldsymbol{\Sigma}(\cdot, t)\boldsymbol{\nabla}_{\mathbf{x}} \log p_t(\cdot)$, $\boldsymbol{\Sigma}(\cdot, t) = \boldsymbol{\sigma}(\cdot, t)\boldsymbol{\sigma}(\cdot, t)^{\mathsf{T}}$, p_t is the marginal probability density function of $\mathbf{x}_t$, and $(\overline{\mathbf{W}}_t)_{t\geq 0}$ is a time reversed Brownian motion.



$$\mathrm{d}\mathbf{x}_t = \underbrace{\tilde{\boldsymbol{\mu}}(\mathbf{x}_t, t)}_{\boldsymbol{\mu}(\mathbf{x}_t, t) - \boldsymbol{\nabla}_{\mathbf{x}} \cdot \boldsymbol{\Sigma}(\mathbf{x}_t, t) - \Sigma(\mathbf{x}_t, t)\overset{?}{\boxed{\boldsymbol{\nabla}_{\mathbf{x}} \log p_t(\mathbf{x}_t)}}}\mathrm{d}t + \boldsymbol{\sigma}(\mathbf{x}_t, t)\mathrm{d}\overline{\mathbf{W}}_t$$

- The score function $\mathbf{s}(\mathbf{x}_t, t) = \boldsymbol{\nabla}_{\mathbf{x}} \log p_t(\mathbf{x}_t)$ is usually intractable, so $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$ is learned instead, using denoising score-matching (Song et al. [8], Vincent [9]):

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x}_t \sim p_t(\cdot)} \left\{ \left\| \boldsymbol{\nabla}_{\mathbf{x}} \log p_t\left(\mathbf{x}_t\right) - \mathbf{s}_{\boldsymbol{\theta}}\left(\mathbf{x}_t, t\right) \right\|^2 \right\} \\ &= \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x}_0 \sim p_0(\cdot)} \mathbb{E}_{\mathbf{x}_t \sim p_{t|0}(\cdot|\mathbf{x}_0)} \left\{ \left\| \boldsymbol{\nabla}_{\mathbf{x}} \log p_{t|0}\left(\mathbf{x}_t | \mathbf{x}_0\right) - \mathbf{s}_{\boldsymbol{\theta}}\left(\mathbf{x}_t, t\right) \right\|^2 \right\} ;\end{aligned}$$

- The score function $\mathbf{s}(\mathbf{x}_t, t) = \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)$ is usually intractable, so $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$ is learned instead, using denoising score-matching (Song et al. [8], Vincent [9]):

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x}_t \sim p_t(\cdot)} \left\{ \left\| \nabla_{\mathbf{x}} \log p_t\left(\mathbf{x}_t\right) - \mathbf{s}_{\boldsymbol{\theta}}\left(\mathbf{x}_t, t\right) \right\|^2 \right\} \\ &= \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x}_0 \sim p_0(\cdot)} \mathbb{E}_{\mathbf{x}_t \sim p_{t|0}(\cdot|\mathbf{x}_0)} \left\{ \left\| \nabla_{\mathbf{x}} \log p_{t|0}\left(\mathbf{x}_t | \mathbf{x}_0\right) - \mathbf{s}_{\boldsymbol{\theta}}\left(\mathbf{x}_t, t\right) \right\|^2 \right\} ;\end{aligned}$$

- Choose an easy (solvable) and reversible SDE: usually linear coefficients, *e.g.* Ornstein-Uhlenbeck process;

- The score function $\mathbf{s}(\mathbf{x}_t, t) = \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)$ is usually intractable, so $\mathbf{s}_{\theta}(\mathbf{x}_t, t)$ is learned instead, using denoising score-matching (Song et al. [8], Vincent [9]):

$$\begin{aligned}\hat{\theta} &= \arg\min_{\theta} \mathbb{E}_{\mathbf{x}_t \sim p_t(\cdot)} \left\{ \left\| \nabla_{\mathbf{x}} \log p_t \left(\mathbf{x}_t\right) - \mathbf{s}_{\theta} \left(\mathbf{x}_t, t\right) \right\|^2 \right\} \\ &= \arg\min_{\theta} \mathbb{E}_{\mathbf{x}_0 \sim p_0(\cdot)} \mathbb{E}_{\mathbf{x}_t \sim p_{t|0}(\cdot|\mathbf{x}_0)} \left\{ \left\| \nabla_{\mathbf{x}} \log p_{t|0} \left(\mathbf{x}_t | \mathbf{x}_0\right) - \mathbf{s}_{\theta} \left(\mathbf{x}_t, t\right) \right\|^2 \right\} ;\end{aligned}$$

- Choose an easy (solvable) and reversible SDE: usually linear coefficients, *e.g.* Ornstein-Uhlenbeck process;
- Knowing the transition kernel, one can sample on the diffusion path: $\mathbf{x}_t \sim p_{t|0} \left(\cdot | \mathbf{x}_0\right)$;

- The score function $\mathbf{s}(\mathbf{x}_t, t) = \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)$ is usually intractable, so $\mathbf{s}_{\theta}(\mathbf{x}_t, t)$ is learned instead, using denoising score-matching (Song et al. [8], Vincent [9]):

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x}_t \sim p_t(\cdot)} \left\{ \left\| \nabla_{\mathbf{x}} \log p_t\left(\mathbf{x}_t\right) - \mathbf{s}_{\boldsymbol{\theta}}\left(\mathbf{x}_t, t\right) \right\|^2 \right\} \\ &= \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x}_0 \sim p_0(\cdot)} \mathbb{E}_{\mathbf{x}_t \sim p_{t|0}(\cdot|\mathbf{x}_0)} \left\{ \left\| \nabla_{\mathbf{x}} \log p_{t|0}\left(\mathbf{x}_t|\mathbf{x}_0\right) - \mathbf{s}_{\boldsymbol{\theta}}\left(\mathbf{x}_t, t\right) \right\|^2 \right\} ;\end{aligned}$$

- Choose an easy (solvable) and reversible SDE: usually linear coefficients, *e.g.* Ornstein-Uhlenbeck process;
- Knowing the transition kernel, one can sample on the diffusion path: $\mathbf{x}_t \sim p_{t|0}\left(\cdot|\mathbf{x}_0\right)$;
- Compute the score function on different noise scales with weighting function $\lambda(t)$ on the Fisher divergence:

$$\mathcal{L}_{\mathrm{DSM}}(\boldsymbol{\theta}) = \int_0^T \lambda(t) \mathbb{E}_{\mathbf{x}_0 \sim p_0(\cdot)} \mathbb{E}_{\mathbf{x}_t \sim p_{t|0}(\cdot|\mathbf{x}_0)} \left\{ \left\| \nabla_{\mathbf{x}} \log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0) - \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \right\|^2 \right\} \mathrm{d}t \, .$$

Score-matching technique

Autoregressive conditional objective

- Recall the minimization objective:

$$\hat{\mathbf{x}}_{0,\tau} = \arg\max_{\mathbf{x}_{0,\tau}} p_0\left(\mathbf{x}_{0,\tau} \middle| \tau, \mathcal{I}(\mathbf{x}_{0,0}), \mathcal{B}(\mathbf{x}_{0,\tau})\right);$$

- Recall the minimization objective:

$$\hat{\mathbf{x}}_{0,\tau} = \arg\max_{\mathbf{x}_{0,\tau}} p_0(\mathbf{x}_{0,\tau} | \tau, \mathcal{I}(\mathbf{x}_{0,0}), \mathcal{B}(\mathbf{x}_{0,\tau})) \,;$$

- Instead, consider an autoregressive formulation (Kohl et al. [6]), conditioning on the last n_f physical states ($\mathbf{c}(\tau, n_\mathrm{f})$ conditioning):

$$\hat{\mathbf{x}}_{0,\tau} = \arg\max_{\mathbf{x}_{0,\tau}} \; p_0\left(\mathbf{x}_{0,\tau} | \mathbf{c}(\tau, n_\mathrm{f})\right) \;;$$

- Recall the minimization objective:

$$\hat{\mathbf{x}}_{0,\tau} = \arg\max_{\mathbf{x}_{0,\tau}} p_0(\mathbf{x}_{0,\tau}|\tau, \mathcal{I}(\mathbf{x}_{0,0}), \mathcal{B}(\mathbf{x}_{0,\tau}))\,;$$

- Instead, consider an autoregressive formulation (Kohl et al. [6]), conditioning on the last n_f physical states ($\mathbf{c}(\tau, n_\mathrm{f})$ conditioning):

$$\hat{\mathbf{x}}_{0,\tau} = \arg\max_{\mathbf{x}_{0,\tau}} \; p_0\left(\mathbf{x}_{0,\tau}|\mathbf{c}(\tau, n_\mathrm{f})\right)\;;$$

- We build a conditional denoiser, taking both time and physical state as external inputs: $\mathbf{s}_\theta(\mathbf{x}_{t,\tau}, \mathbf{c}(\tau, n_\mathrm{f}), t)$;

- Recall the minimization objective:

$$\hat{\mathbf{x}}_{0,\tau} = \arg\max_{\mathbf{x}_{0,\tau}} p_0(\mathbf{x}_{0,\tau}|\tau, \mathcal{I}(\mathbf{x}_{0,0}), \mathcal{B}(\mathbf{x}_{0,\tau}))\,;$$

- Instead, consider an autoregressive formulation (Kohl et al. [6]), conditioning on the last n_f physical states ($\mathbf{c}(\tau, n_\mathrm{f})$ conditioning):

$$\hat{\mathbf{x}}_{0,\tau} = \arg\max_{\mathbf{x}_{0,\tau}} \; p_0\left(\mathbf{x}_{0,\tau}|\mathbf{c}(\tau, n_\mathrm{f})\right)\;;$$

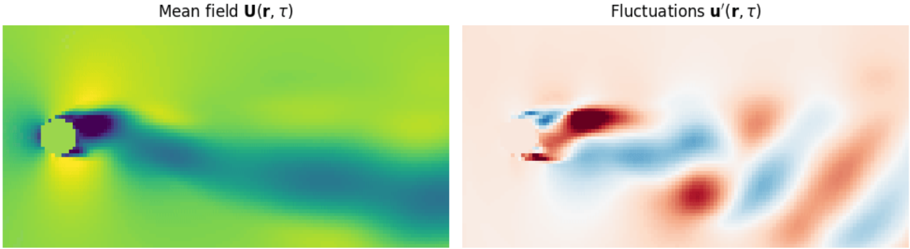
- We build a conditional denoiser, taking both time and physical state as external inputs: $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_{t,\tau}, \mathbf{c}(\tau, n_\mathrm{f}), t)$;
- With Bayes' theorem (+ Batzolis et al. [2]), we can rewrite the conditional objective and train the conditional score function:

$$\mathcal{L}_{\mathrm{DSM}}(\boldsymbol{\theta}) = \int_0^T \lambda(t)\mathbb{E}_{\substack{\mathbf{x}_{0,\tau},\,\mathbf{c}(\tau,n_\mathrm{f})\sim p_0(\cdot,\mathbf{c})\\ \mathbf{x}_{t,\tau}\sim p_{t|0}(\cdot|\mathbf{x}_{0,\tau})}} \left\{\left\|\nabla_{\mathbf{x}} \log p_{t|0}(\mathbf{x}_{t,\tau}|\mathbf{x}_{0,\tau}) - \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_{t,\tau}, \mathbf{c}(\tau, n_\mathrm{f}), t)\right\|^2\right\} \mathrm{d}t\,.$$

- Since we choose an autoregressive formulation, we impose consistency in between predictions via a regularization term of the type: $\mathcal{R}(f(\mathbf{x}_\tau, \mathbf{x}_{\tau - n_\mathrm{f}}), f(\hat{\mathbf{x}}_\tau, \hat{\mathbf{x}}_{\tau - n_\mathrm{f}}));$

Score-matching technique
Autoregressive conditional objective

- Since we choose an autoregressive formulation, we impose consistency in between predictions via a regularization term of the type: $\mathcal{R}(f(\mathbf{x}_\tau, \mathbf{x}_{\tau-n_f}), f(\hat{\mathbf{x}}_\tau, \hat{\mathbf{x}}_{\tau-n_f}))$;
- Covariance penalization to conserve fluid properties (Dryden [3]) $\to$ mean field decomposition: mean part $\mathbf{U}$ + random fluctuations $\mathbf{u}'$ (s.t. $\mathbf{u}'(\mathbf{r},\tau) = \mathbf{U}(\mathbf{r},\tau) - \mathbf{u}(\mathbf{r},\tau)$ & zero mean $\mathbb{E}\{\mathbf{u}'(\mathbf{r},\tau)\} = \mathbf{0}$);



- Since we choose an autoregressive formulation, we impose consistency in between predictions via a regularization term of the type: $\mathcal{R}(f(\mathbf{x}_\tau, \mathbf{x}_{\tau-n_\mathrm{f}}), f(\hat{\mathbf{x}}_\tau, \hat{\mathbf{x}}_{\tau-n_\mathrm{f}}))$;
- Covariance penalization to conserve fluid properties (Dryden [3]) $\rightarrow$ mean field decomposition: mean part $\mathbf{U}$ + random fluctuations $\mathbf{u}'$ (s.t. $\mathbf{u}'(\mathbf{r},\tau) = \mathbf{U}(\mathbf{r},\tau) - \mathbf{u}(\mathbf{r},\tau)$ & zero mean $\mathbb{E}\{\mathbf{u}'(\mathbf{r},\tau)\} = \mathbf{0}$);
  - Reconstruct the estimated field $\hat{\mathbf{u}}$ from: $\hat{\mathbf{x}}_{0,\tau} \simeq \mathbf{x}_{t,\tau} + \boldsymbol{\Sigma}_t \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_{t,\tau}, \mathbf{c}(\tau, n_\mathrm{f}), t)$;

- Since we choose an autoregressive formulation, we impose consistency in between predictions via a regularization term of the type: $\mathcal{R}(f(\mathbf{x}_{\tau}, \mathbf{x}_{\tau - n_{\mathrm{f}}}), f(\hat{\mathbf{x}}_{\tau}, \hat{\mathbf{x}}_{\tau - n_{\mathrm{f}}}))$;
- Covariance penalization to conserve fluid properties (Dryden [3]) $\rightarrow$ mean field decomposition: mean part $\mathbf{U}$ + random fluctuations $\mathbf{u}'$ (s.t. $\mathbf{u}'(\mathbf{r}, \tau) = \mathbf{U}(\mathbf{r}, \tau) - \mathbf{u}(\mathbf{r}, \tau)$ & zero mean $\mathbb{E}\{\mathbf{u}'(\mathbf{r}, \tau)\} = \mathbf{0}$);
  - Reconstruct the estimated field $\hat{\mathbf{u}}$ from: $\hat{\mathbf{x}}_{0,\tau} \simeq \mathbf{x}_{t,\tau} + \boldsymbol{\Sigma}_t \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_{t,\tau}, \mathbf{c}(\tau, n_{\mathrm{f}}), t)$;
  - Compute the 2×2 autocorrelation matrix of the velocity field fluctuations between consecutive predictions:

$$\mathbf{R}_{\mathbf{u}}(\mathbf{r}_1, \mathbf{r}_2, \tau_1, \tau_2) = \mathbb{E}\{\mathbf{u}'(\mathbf{r}_1, \tau_1) \otimes \mathbf{u}'(\mathbf{r}_2, \tau_2)\}\,;$$

- Since we choose an autoregressive formulation, we impose consistency in between predictions via a regularization term of the type: $\mathcal{R}(f(\mathbf{x}_\tau, \mathbf{x}_{\tau-n_\mathrm{f}}), f(\hat{\mathbf{x}}_\tau, \hat{\mathbf{x}}_{\tau-n_\mathrm{f}}))$;
- Covariance penalization to conserve fluid properties (Dryden [3]) $\rightarrow$ mean field decomposition: mean part $\mathbf{U}$ + random fluctuations $\mathbf{u}'$ (s.t. $\mathbf{u}'(\mathbf{r},\tau) = \mathbf{U}(\mathbf{r},\tau) - \mathbf{u}(\mathbf{r},\tau)$ & zero mean $\mathbb{E}\{\mathbf{u}'(\mathbf{r},\tau)\} = \mathbf{0}$);
  - ☐ Reconstruct the estimated field $\hat{\mathbf{u}}$ from: $\hat{\mathbf{x}}_{0,\tau} \simeq \mathbf{x}_{t,\tau} + \mathbf{\Sigma}_t \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_{t,\tau}, \mathbf{c}(\tau, n_\mathrm{f}), t)$;
  - ☐ Compute the 2×2 autocorrelation matrix of the velocity field fluctuations between consecutive predictions:

$$\mathbf{R}_\mathbf{u}(\mathbf{r}_1, \mathbf{r}_2, \tau_1, \tau_2) = \mathbb{E}\{\mathbf{u}'(\mathbf{r}_1, \tau_1) \otimes \mathbf{u}'(\mathbf{r}_2, \tau_2)\} ;$$

  - ☐ Compute the L_2 error on the covariance matrices on the space distribution:

$$\mathcal{L}_\varepsilon(\boldsymbol{\theta}) = \lambda_\varepsilon \mathbb{E}_{\mathbf{r}_1, \mathbf{r}_2 \in \Omega} \mathbb{E}_{\mathbf{u} \in \mathcal{D}} \left\{ \left\| \mathbf{R}_\mathbf{u}(\mathbf{r}_1, \mathbf{r}_2, \tau, \tau - n_\mathrm{f}) - \mathbf{R}_{\hat{\mathbf{u}}}(\mathbf{r}_1, \mathbf{r}_2, \tau, \tau - n_\mathrm{f}) \right\|_2 \right\} ;$$

- Since we choose an autoregressive formulation, we impose consistency in between predictions via a regularization term of the type: $\mathcal{R}(f(\mathbf{x}_\tau, \mathbf{x}_{\tau-n_\mathrm{f}}), f(\hat{\mathbf{x}}_\tau, \hat{\mathbf{x}}_{\tau-n_\mathrm{f}}))$;
- Covariance penalization to conserve fluid properties (Dryden [3]) $\rightarrow$ mean field decomposition: mean part $\mathbf{U}$ + random fluctuations $\mathbf{u}'$ (s.t. $\mathbf{u}'(\mathbf{r},\tau) = \mathbf{U}(\mathbf{r},\tau) - \mathbf{u}(\mathbf{r},\tau)$ & zero mean $\mathbb{E}\{\mathbf{u}'(\mathbf{r},\tau)\} = \mathbf{0}$);
  - ☐ Reconstruct the estimated field $\hat{\mathbf{u}}$ from: $\hat{\mathbf{x}}_{0,\tau} \simeq \mathbf{x}_{t,\tau} + \mathbf{\Sigma}_t \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_{t,\tau}, \mathbf{c}(\tau, n_\mathrm{f}), t)$;
  - ☐ Compute the 2×2 autocorrelation matrix of the velocity field fluctuations between consecutive predictions:
  $$\mathbf{R}_\mathbf{u}(\mathbf{r}_1, \mathbf{r}_2, \tau_1, \tau_2) = \mathbb{E}\{\mathbf{u}'(\mathbf{r}_1, \tau_1) \otimes \mathbf{u}'(\mathbf{r}_2, \tau_2)\}\,;$$
  - ☐ Compute the L_2 error on the covariance matrices on the space distribution:
  $$\mathcal{L}_\varepsilon(\boldsymbol{\theta}) = \lambda_\varepsilon \mathbb{E}_{\mathbf{r}_1,\mathbf{r}_2\in\Omega} \mathbb{E}_{\mathbf{u}\in\mathcal{D}} \left\{ \|\mathbf{R}_\mathbf{u}(\mathbf{r}_1, \mathbf{r}_2, \tau, \tau - n_\mathrm{f}) - \mathbf{R}_{\hat{\mathbf{u}}}(\mathbf{r}_1, \mathbf{r}_2, \tau, \tau - n_\mathrm{f})\|_2 \right\}\;;$$
- By averaging over the physical state in time, the resulting loss function reads:

$$\mathcal{L}_\mathrm{total}(\boldsymbol{\theta}) = \int_{n_\mathrm{f}}^{\mathcal{T}} \underbrace{\mathcal{L}_\mathrm{DSM}(\boldsymbol{\theta})}_{\text{conditional score-matching}} + \underbrace{\mathcal{L}_{\boldsymbol{\varepsilon}}(\boldsymbol{\theta})}_{\text{covariance regularization}} \, \mathrm{d}\tau\,.$$

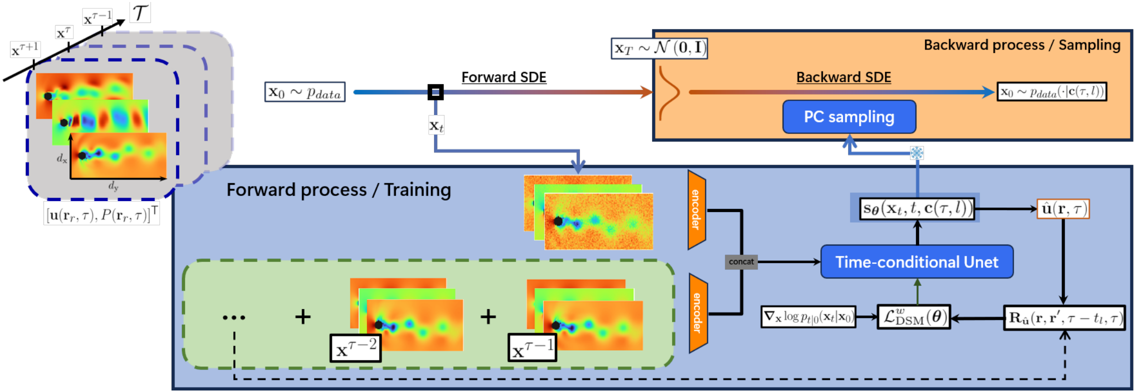


FIGURE – Architecture[1] of the entire diffusion model (+ SDE and sampling procedure).

[1]Backbone of the model: `https://huggingface.co/blog/annotated-diffusion`

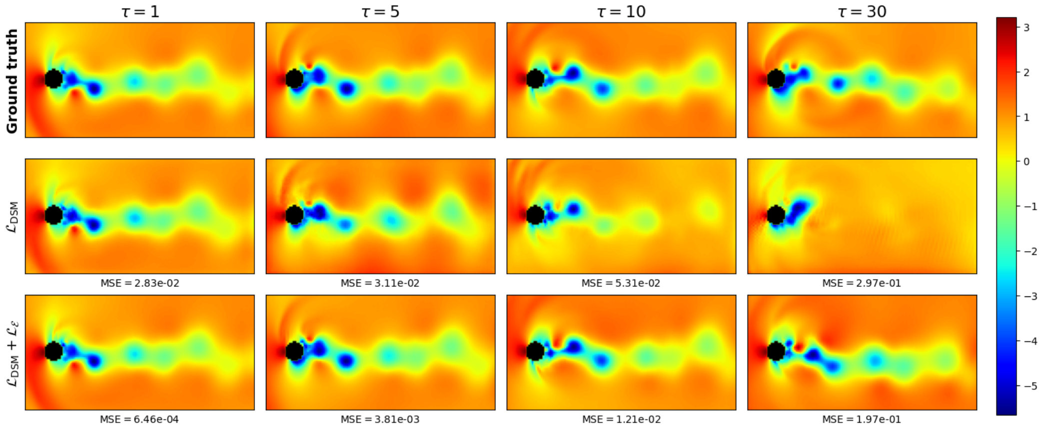


FIGURE – predictions of the pressure field at Mach 0.5.

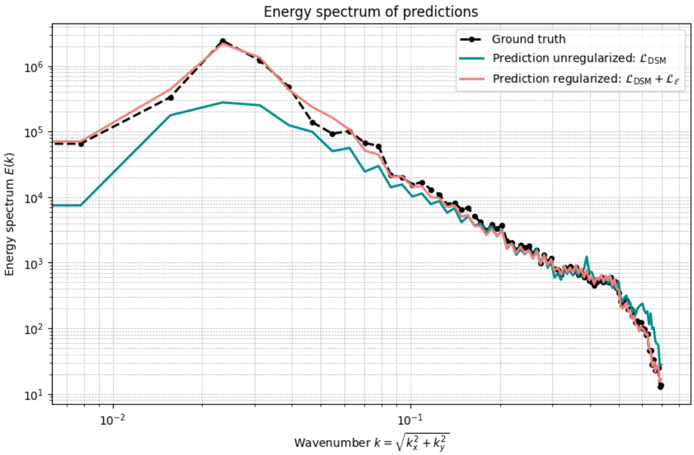


FIGURE – Energy density spectrum at Mach 0.5.

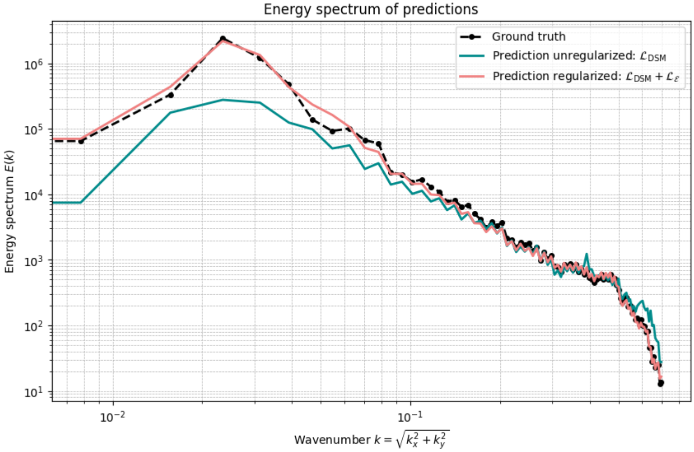


FIGURE – Energy density spectrum at Mach 0.5.

- Energy density spectrum: $E(k_x, k_y) = |\hat{u}(k_x, k_y)|^2 + |\hat{v}(k_x, k_y)|^2$
- Small scale turbulence $\rightarrow$ large k (high frequency)
- Large scale turbulence $\rightarrow$ small k (low frequency)
- Size of flow structure/length scale of structures: $\lambda \sim 2\pi/k$

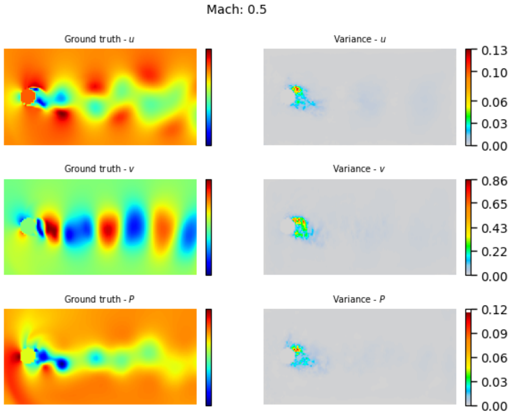


FIGURE – Variance of 10 predictions **without** covariance regularization.

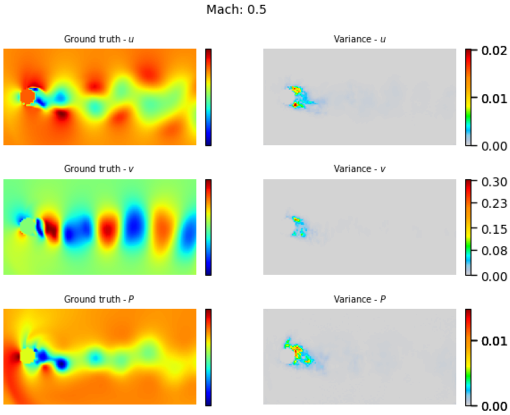


FIGURE – Variance on 10 predictions **with** covariance regularization.

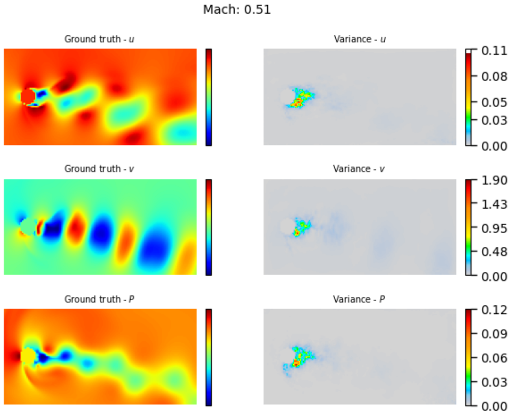


FIGURE – Variance of 10 predictions **without** covariance regularization.

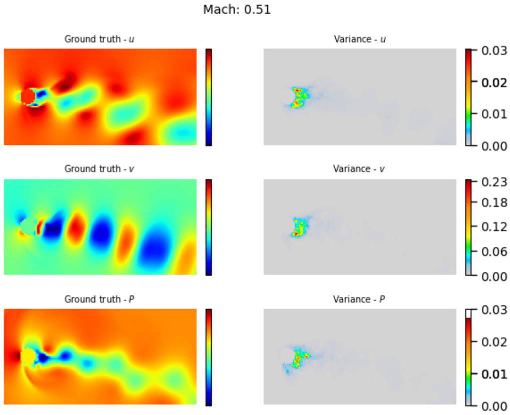


FIGURE – Variance of 10 predictions **with** covariance regularization.

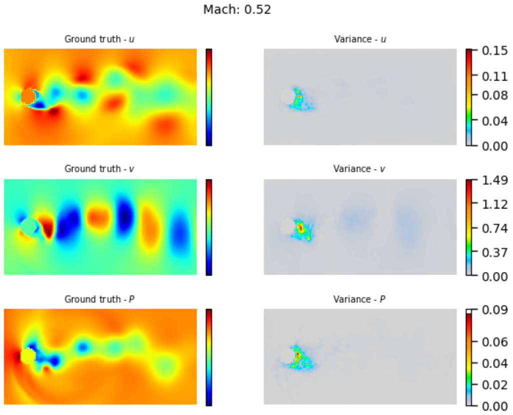


FIGURE – Variance of 10 predictions **without** covariance regularization.

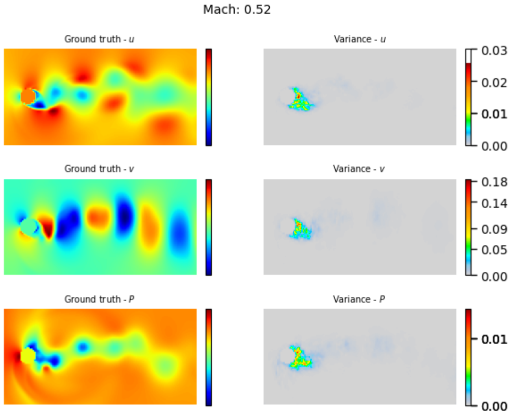


FIGURE – Variance of 10 predictions **with** covariance regularization.

Perspectives

Pros:

- **Robust and reliable sampling procedure:** ensures stability in generated fields through well-designed diffusion steps, minimizing deviations during early inference,
- **Simultaneous multi-field generation:** concurrent generation of u, v, and P, preserving correlations and multi-scale resolution,
- **Accurate long-term predictions:** reliable predictions sustained up to $\tau \simeq 15 \times T_{\mathrm{Lya}}$, where T_{Lya} represents the Lyapunov timescale, ensuring coherence in turbulent flow dynamics.

Cons:

- **Slow inference speed:** the iterative nature of the sampling process, often requiring a high number of diffusion steps,
- **Error accumulation:** due to autoregressive propagation, errors can propagate easily,
- **Training instability:** handling very large datasets $\mathcal{D}$ and high dimensionality leads to a very slow convergence.

Perspectives:

- **Latent space diffusion:** dimensionality reduction by learning a latent prior,
- **Efficient conditioning:** develop lighter conditioning methods to reduce the computational overhead of conditioning operations,
- **Density constraints:** introduce physical constraints on the density function $\rho(\mathbf{x})$ to improve the model's fidelity to conservation laws.

Thank you for listening! Questions?

Animated results are also available here:

Wilfried GENUIST

wilfried.genuist@onera.fr

wilfried.genuist@centralesupelec.fr

References

[1] Brian D. O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[2] Georgios Batzolis, Jan Stanczuk, Carola-Bibiane Schönlieb, and Christian Etmann. Conditional image generation with score-based diffusion models. *arXiv:2111.13606*, 2021.

[3] Hugh L. Dryden. A review of the statistical theory of turbulence. *Quarterly of Applied Mathematics*, 1(1): 7–42, 1943.

[4] Ulrich G. Haussmann and Étienne Pardoux. Time reversal of diffusions. *The Annals of Probability*, 14(4): 1188–1205, 1986.

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[6] Georg Kohl, Li-Wei Chen, and Nils Thuerey. Benchmarking autoregressive conditional diffusion models for turbulent flow simulation. *arXiv:2309.01745*, 2023.

[7] Bernt Oksendal. *Stochastic Differential Equations: an Introduction with Applications*. Springer, 2013.

[8] Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence*, pages 574–584. PMLR, 2020.

[9] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23 (7):1661–1674, 2011.