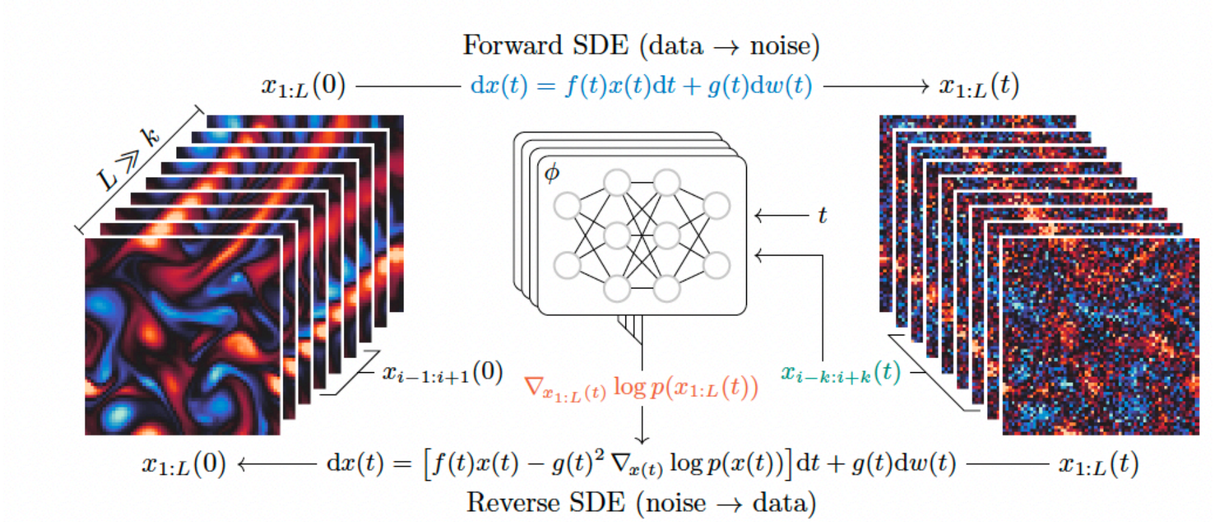
25 March 2025

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Score-based Data Assimilation

Authors: François ROZET, Gilles LOUPPE

Teachers: Ronan Fablet, François Rousseau and Mai Quyen Pham



Students: ARTHAUD Lilian, NGUYEN Nhan and SENEQUIER Fabien

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

TABLE OF CONTENTS

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

1 - Method Recap

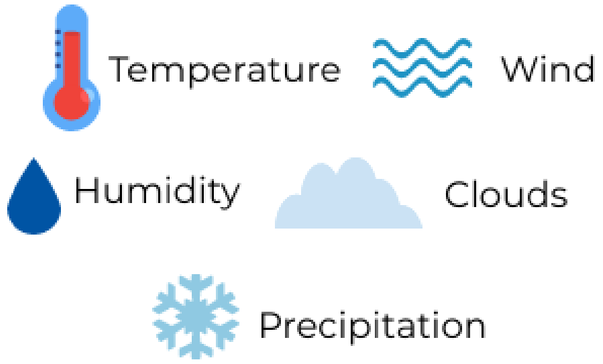
1.1. Data Assimilation

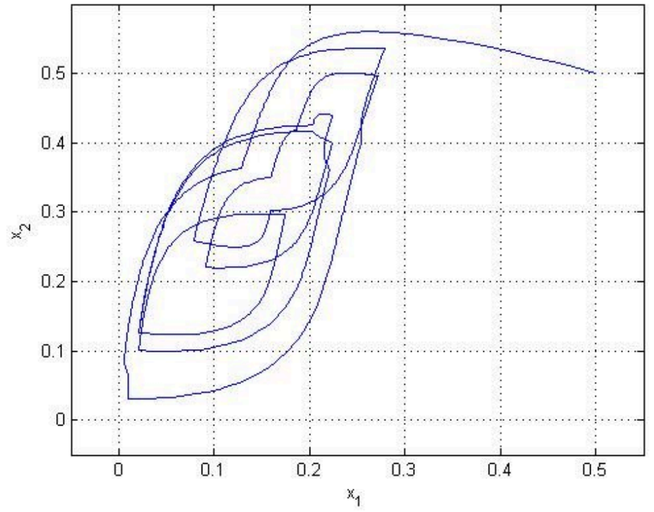
Imperfect information

- **Sparse observation**
- **Intermittent observation**
- **Noisy observation**

State of a system evolving over time

(State trajectories)





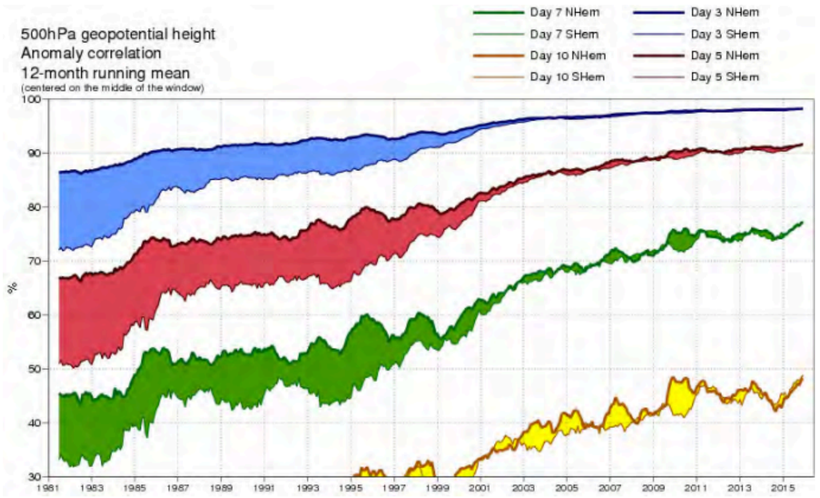
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

1 - Method Recap

1.1. Data Assimilation

PARTICLE-BASED AND VARIATIONAL METHODS

NUMERCIAL WEATHER PREDICTION (NWP)



From www.ecmwf.int

OCEAN DATA ASSIMILATION

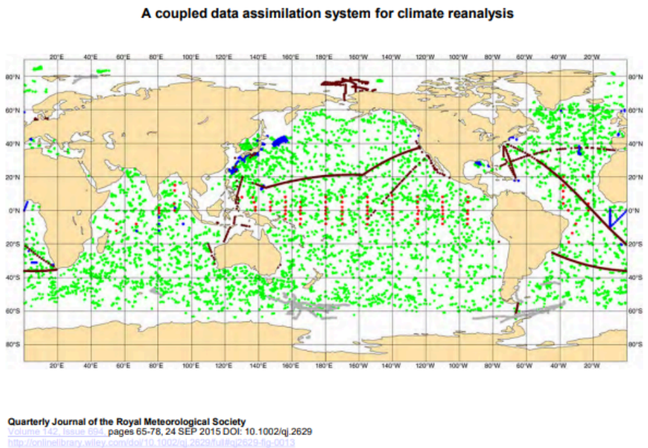


Figure from Lalayoux et al (2016)

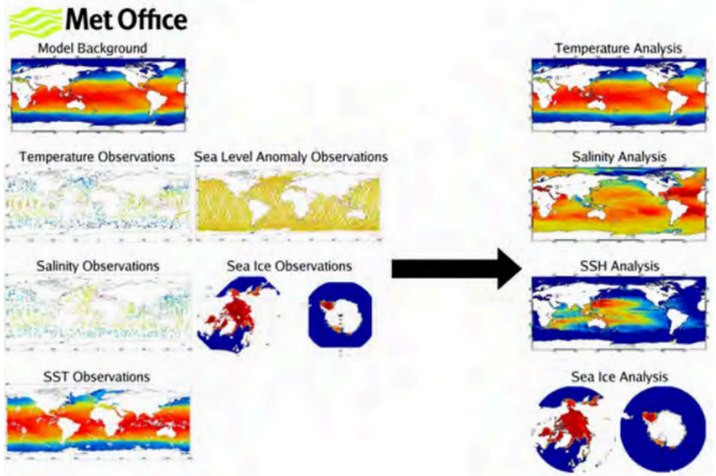


Figure from www.metoffice.gov.uk

(-) Depend on the transition dynamics for inference, not suitable for long time horizons or high-dimensional, nonlinear systems with complex dynamics.

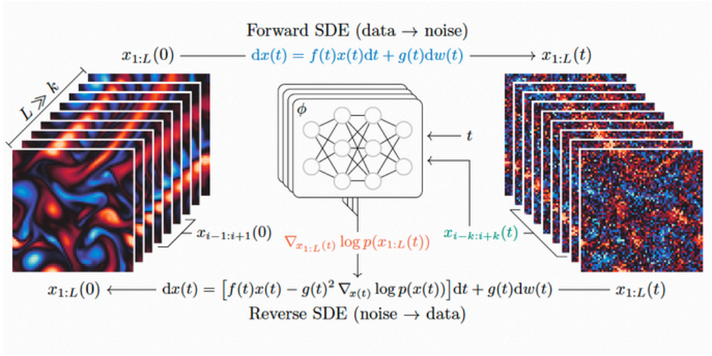
SCORE-BASED DATA ASSIMILATION FOR TRAJECTORY INFERENCE

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

1 - Method Recap

1.2. Score-Based Methods

Forward SDE (Diffusion): Adds noise gradually to the data sample (training phase).



Idea: Start from clean data and gradually turn it into pure noise via an SDE (Stochastic Differential Equation).

$$\mathrm{d}x(t) = f(t)\,x(t)\,\mathrm{d}t + g(t)\,\mathrm{d}w(t)$$

with : $f(t)$: Drift function, giving a deterministic direction of the noise.

$w(t)$: Wiener process (Brownian motion), the random component of the SDE

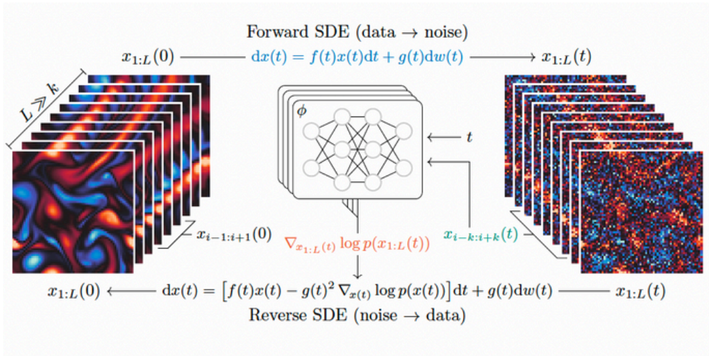
$g(t)$: Weighting function on the Wiener process

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

1 - Method Recap

1.2. Score-Based Methods

Reverse SDE (Sampling): Reverse the noise process to reconstruct data from random noise using learned score (sampling phase).



Using the learned score, we simulate the reverse SDE:

$$\mathrm{d}x(t) = \left[f(t)\,x(t) - g(t)^2\,\nabla_{x(t)} \log p(x(t))\right]\mathrm{d}t + g(t)\,\mathrm{d}w(t)$$

We generate data by going backward through noise, guided by the score.

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

1 - Method Recap

1.3. Diffusion Posterior Sampling for Inverse Problems

Key idea:

- **Observation :** $y = \mathcal{A}(x_{1:L}) + \eta$ where $\mathcal{A}$: measurement function (non-linear)
  η : observational error (stochastic additive term)

- **Score correction:** modifies the reverse SDE to match posterior

$$\underbrace{\nabla_{x(t)} \log p(x(t) \mid y)}_{\text{posterior score}} = \underbrace{\nabla_{x(t)} \log p(x(t))}_{\substack{\text{prior score} \\ \text{(how likely x is before seeing y)}}} + \underbrace{\nabla_{x(t)} \log p(y \mid x(t))}_{\substack{\text{likelihood score} \\ \text{(how well x explains} \\ \text{the observation y)}}}$$

1 - Method Recap

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Our Objective : apply Score-based methods for denoising

2 - Numerical Implementation

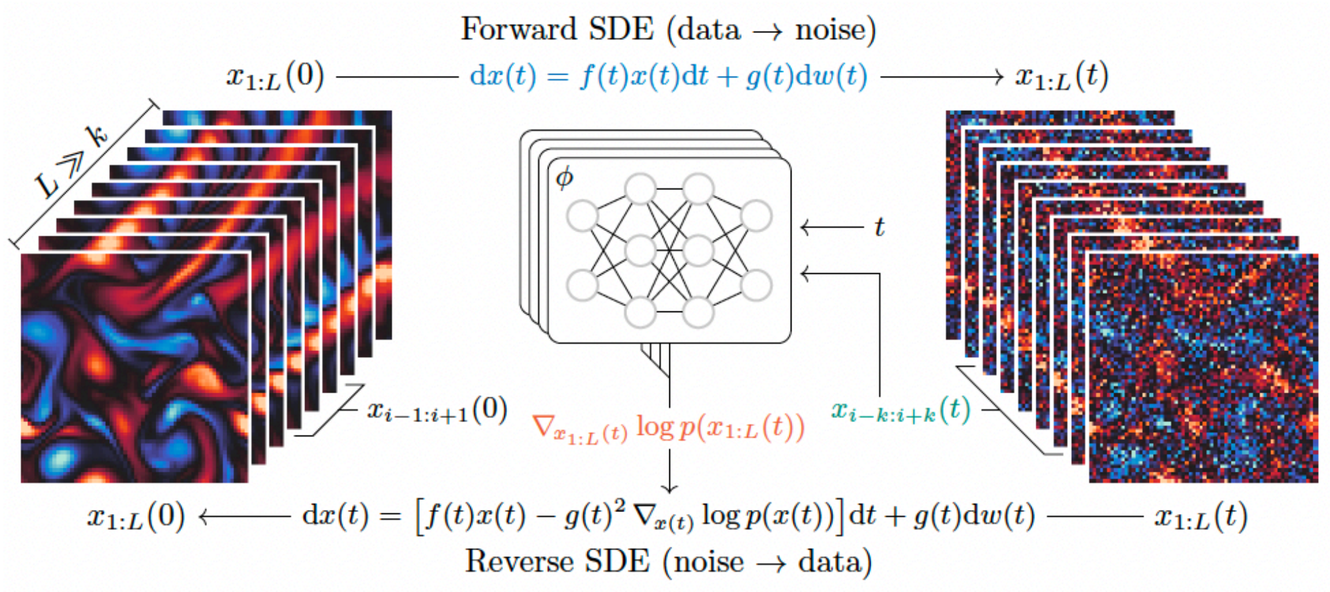
```
├── sda
│   ├── mcs.py
│   ├── score.py
│   └── utils.py
└── experiments
    └── kolmogorov
        ├── generate.py
        ├── train.py
        ├── utils.py
        ├── sandwich.ipynb
        ├── figures.ipynb
        ├── figures_bis.ipynb
        ├── runs/
        │   └── ... (trained models)
        └── data/
            ├── x_000001.npy ... x_00xxxx.npy
            ├── train.h5
            ├── valid.h5
            └── test.h5
```

https://github.com/FabienSenequier/Projet-CompI

2 - Numerical Implementation

Data Generation
generate.py

Score Model Training
train.py



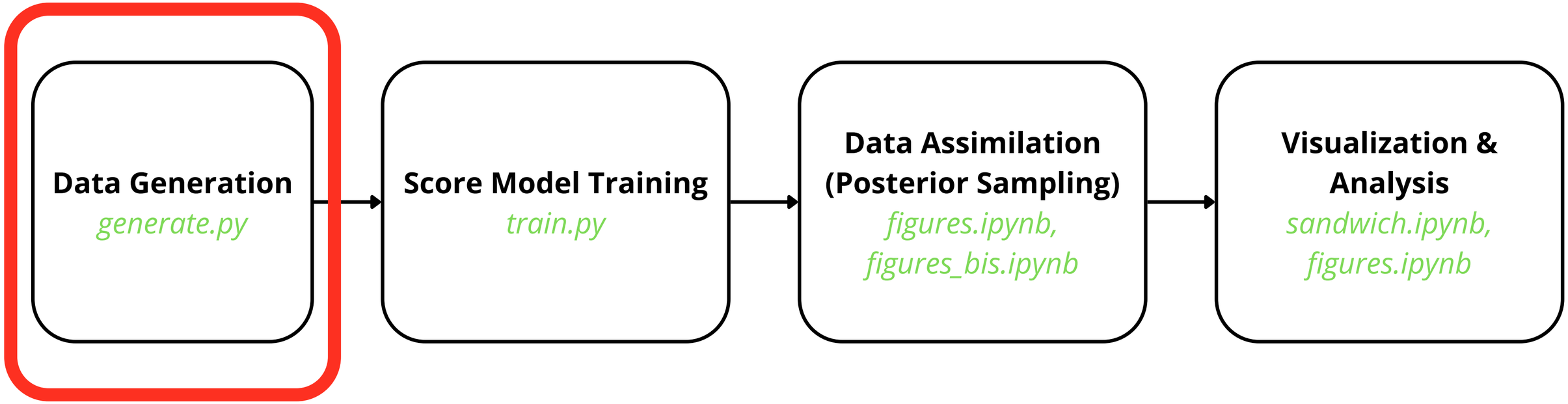
Visualization & Analysis
sandwich.ipynb, figures.ipynb

Data Assimilation (Posterior Sampling)
figures.ipynb, figures_bis.ipynb

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

2 - Numerical Implementation

2.1. Data Generation



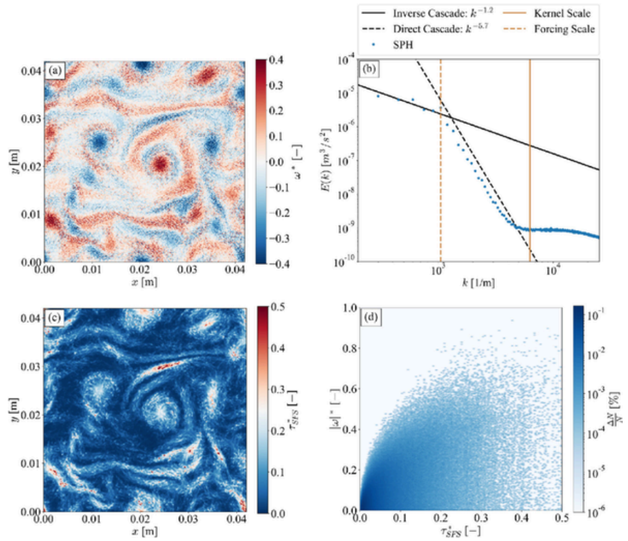
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

2 - Numerical Implementation

2.1. Data Generation - *generate.py*

- **Simulate** 1024 fluid dynamics using **Kolmogorov flow** equation.
- **Save** as a NumPy file.
- **Aggregate** into:
  - 80%: train.h5.
  - 10%: valid.h5.
  - 10%: test.h5.

```
data/
├── x_000001.npy ... x_00xxxx.npy
├── train.h5
├── valid.h5
└── test.h5
```

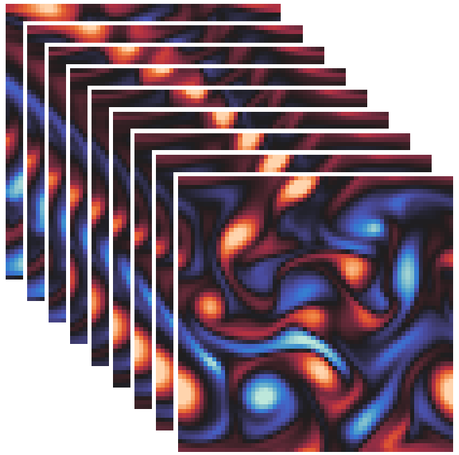


IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

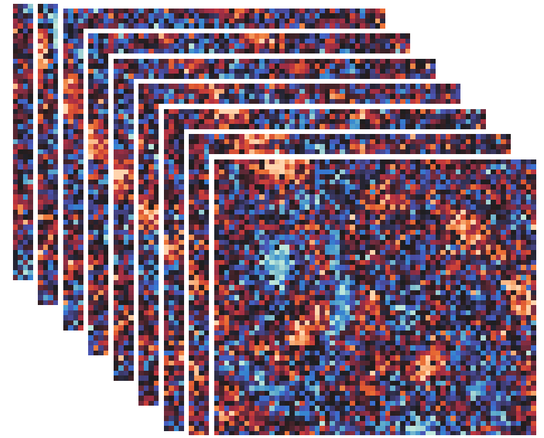
2 - Numerical Implementation

2.1. Data Generation - *generate.py*

```
!python "/content/drive/MyDrive/FISE 3A/UE_F_Computational Imaging/Project/sda-master/experiments/kolmogorov/generate.py"
```



Clean sequence of vorticity fields from the data w
(Pure data)

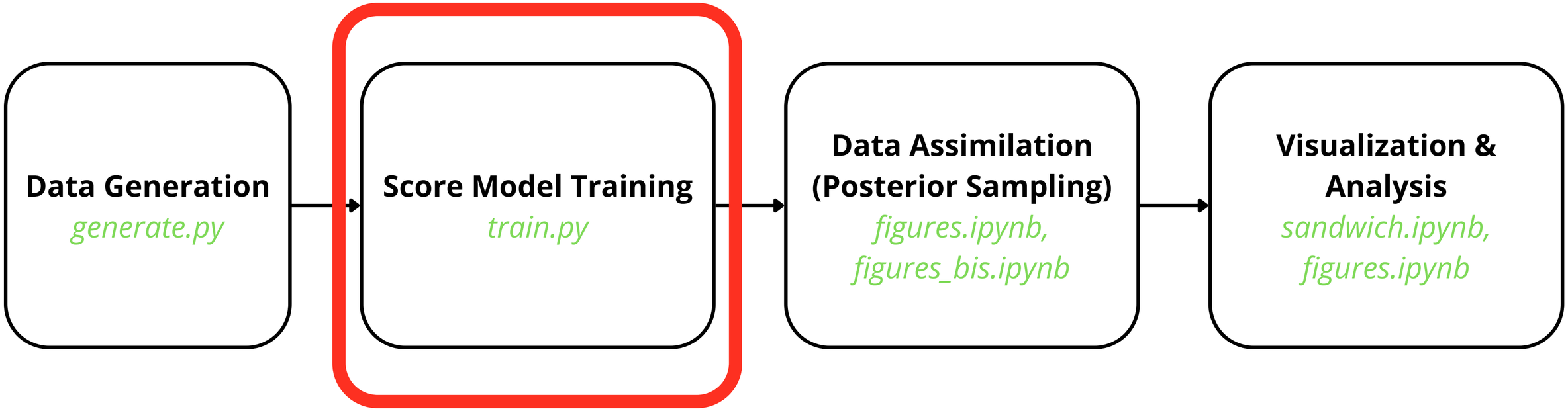


Noisy version of the vorticity fields
(70% original signal + 40% randam noise)

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

2 - Numerical Implementation

2.2. Score Model Training



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

2 - Numerical Implementation

2.2. Score Model Training - *train.py*

- **Train score-based model** (Markovian conditional score network) by using **variance-preserving SDE (VPSDE)** framework.
- Experiment Settings:

```
!python "/content/drive/MyDrive/FISE 3A/UE_F_Computational Imaging/Project/sda-master/experiments/kolmogorov/train.py"
```

```
CONFIG = {
    # Architecture
    'window': 5,
    'embedding': 64,
    'hidden_channels': (96, 192, 384),
    'hidden_blocks': (3, 3, 3),
    'kernel_size': 3,
    'activation': 'SiLU',
    # Training
    'epochs': 4096,
    'batch_size': 32,
    'optimizer': 'AdamW',
    'learning_rate': 2e-4,
    'weight_decay': 1e-3,
    'scheduler': 'linear',
}
```

Limitation of Google Colab

Train with CPU

```
CONFIG = {
    'window': 3,
    'embedding': 16,
    'hidden_channels': (32, 64, 128),
    'hidden_blocks': (2, 2, 2),
    'kernel_size': 3,
    'activation': 'SiLU',
    'epochs': 100,
    'batch_size': 2,
    'optimizer': 'AdamW',
    'learning_rate': 2e-4,
    'weight_decay': 1e-3,
    'scheduler': 'linear',
}
```

2 - Numerical Implementation

2.2. Score Model Training - *train.py*

```
runs/
└── ... (trained models)
```

- **Final output:** state.pth - the trained score model weights.

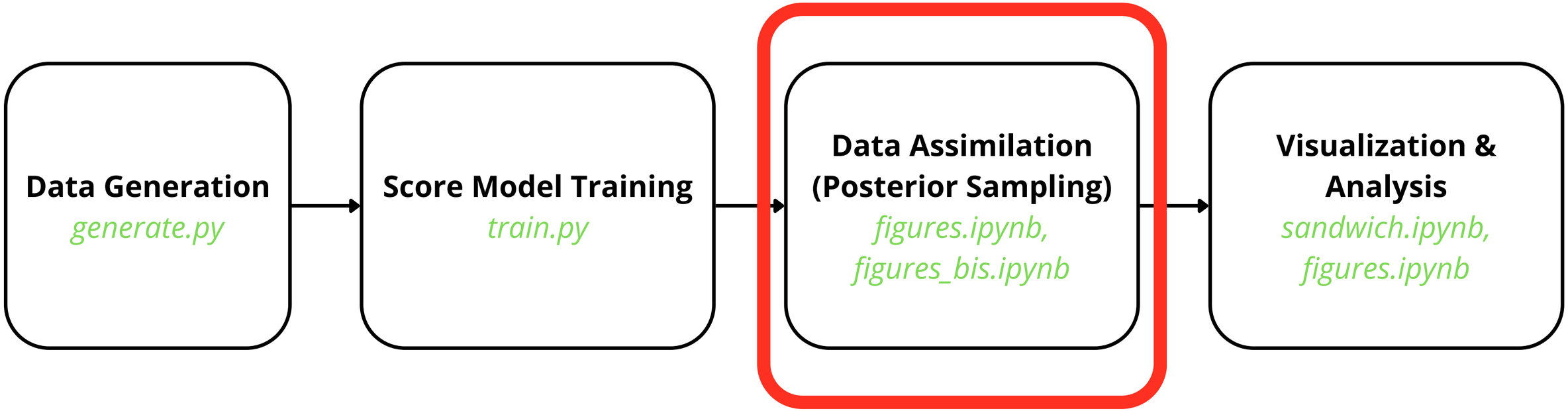
- This model learned the gradient of the log-density (the "score") of turbulent fluid trajectories (Kolmogorov flow).

- Once trained, this model can be used in reverse-time SDEs to simulate, assimilate, or reconstruct dynamical states from noisy/partial data.

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

2 - Numerical Implementation

2.2. Score Model Training



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

2 - Numerical Implementation

2.3. Data Assimilation (Posterior Sampling) - *figures.ipynb, figures_bis.ipynb*

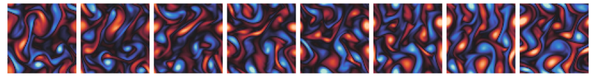
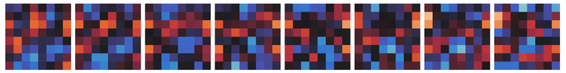
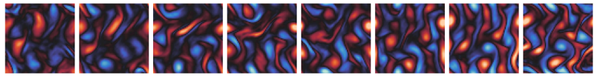
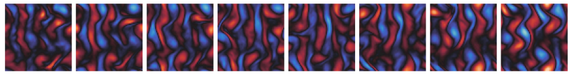
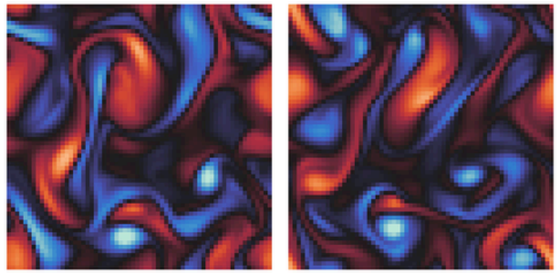
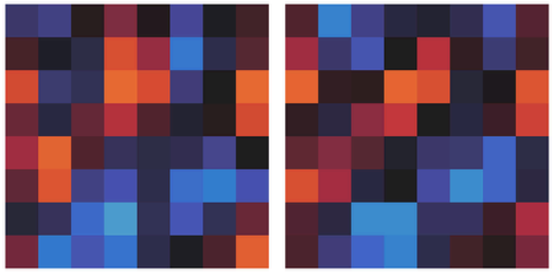
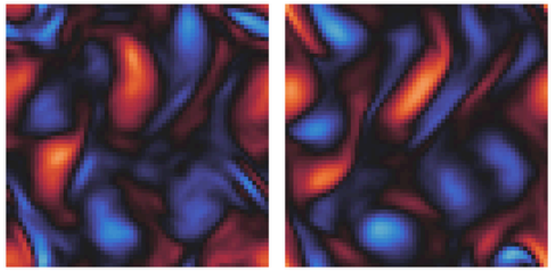
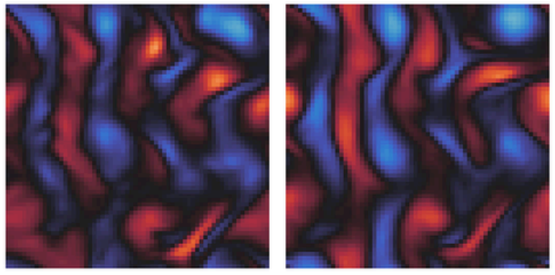
- **Generate visual results of the data assimilation** process using a trained score model.
- Include: Loading Model & Data, Performing Reverse SDE (Noise -> Data) and Visualization.
  - ***Step 1:*** Simulate synthetic trajectories
  - ***Step 2:*** Assimilation (given partial observations)
  - ***Step 3:*** Extrapolation (predict the future)
  - ***Step 4:*** Evaluation (compare with ground truth)

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

2 - Numerical Implementation

2.3. Data Assimilation (Posterior Sampling) - *figures.ipynb, figures_bis.ipynb*

- ***Step 1:*** Simulate synthetic trajectories
- ***Step 2:*** Assimilation (given partial observations)
  - **SDA:** Score-based Data Assimilation
  - **DPS:** Diffusion Posterior Sampling

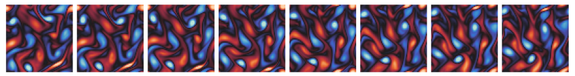
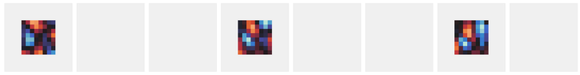
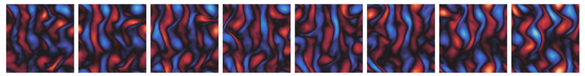
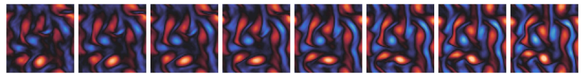
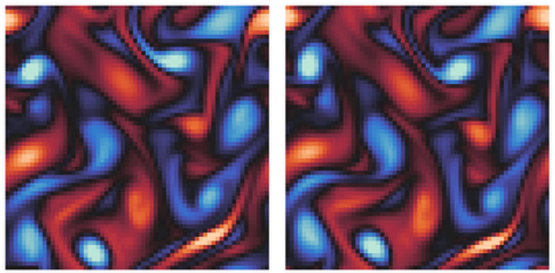
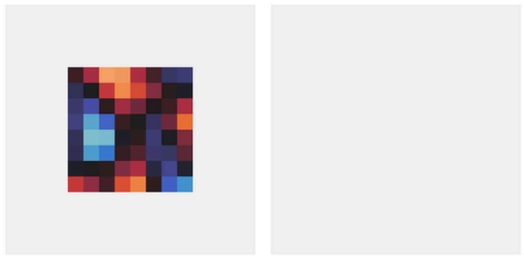
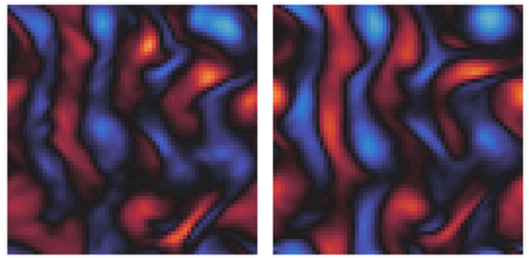
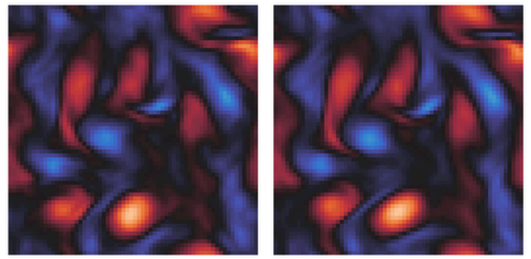
Ground truth	Observation	SDA Results	DPS Results
			
x_star_assim.png	y_star_assim.png	x_sda_assim.png	x_dps_assim.png
			
		tensor(0.1212)	tensor(0.8142)

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

2 - Numerical Implementation

2.3. Data Assimilation (Posterior Sampling) - *figures.ipynb, figures_bis.ipynb*

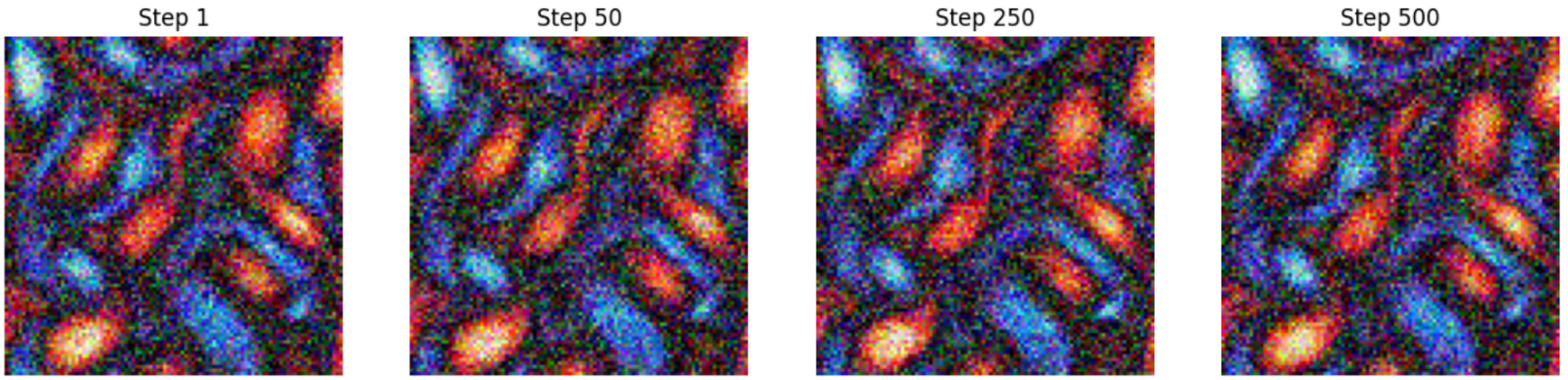
- ***Step 3:*** Extrapolation (forecasting)
- ***Step 4:*** Evaluation (compare with ground truth)

Ground truth	Observation	SDA Results	DPS Results
			
x_star_extra.png	y_star_extra.png	x_sda_extra.png	x_dps_extra.png
			
		tensor(0.0128)	tensor(0.6871)

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

3 - Developed Denoising Algorithm

3.1. Noising Process and Training



- We iteratively add **noise** to our input image
- We calculate the **score** thanks to our model
- We compute the **loss**
- We do **backpropagation**

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

3 - Developed Denoising Algorithm

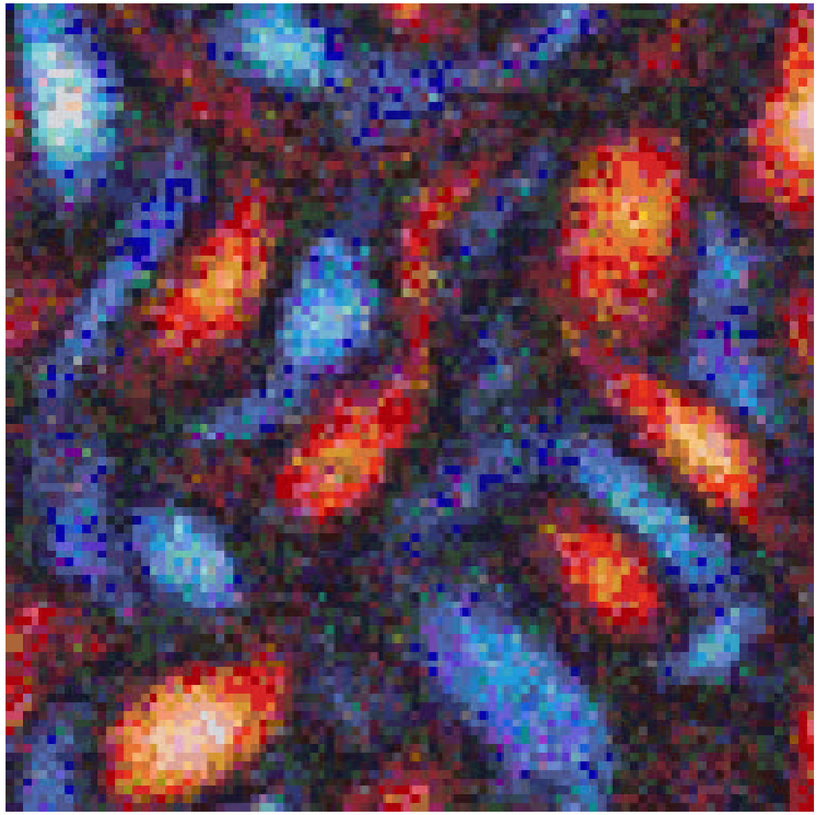
3.1. Denoising algorithm

We iteratively refine the image by **moving in the direction of the score** and adding noise.

$$x_0 = \text{noisy\_image}$$

$$x_{t+1} = x_t + \eta \cdot s(x_t) + \xi \cdot \sqrt{2\eta}$$

with
- η : step size
- $s(x_t)$: score
- $\xi \sim \mathcal{N}(0, \sigma^2 I)$

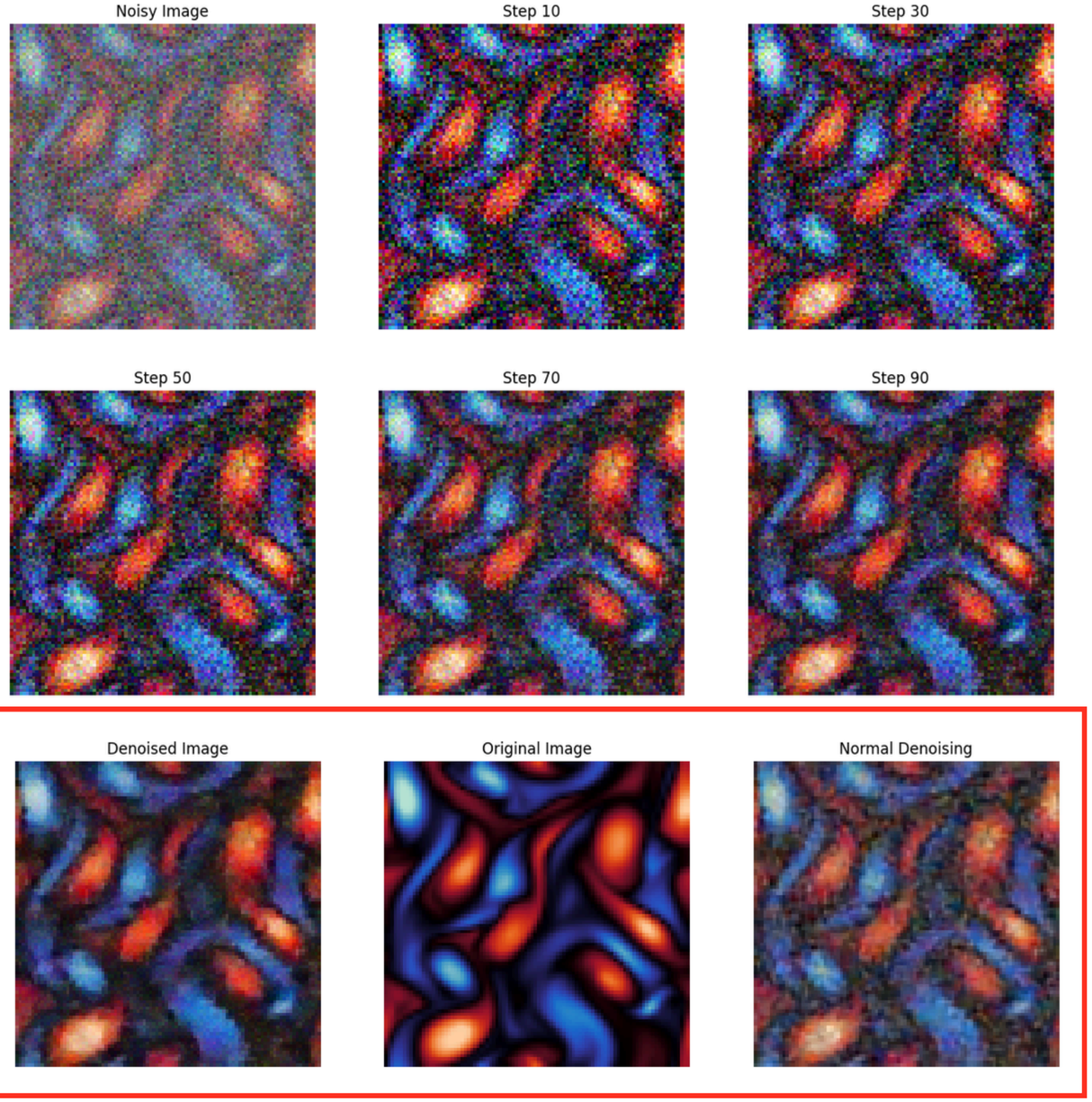


IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

3 - Developed Denoising Algorithm

3.1. Denoising results

	Score-based denoising method	Baseline denoising method
MSE	0.0042	0.0069
PSNR	23.7	21.6



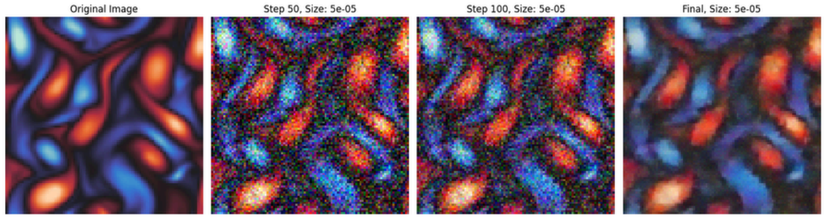
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

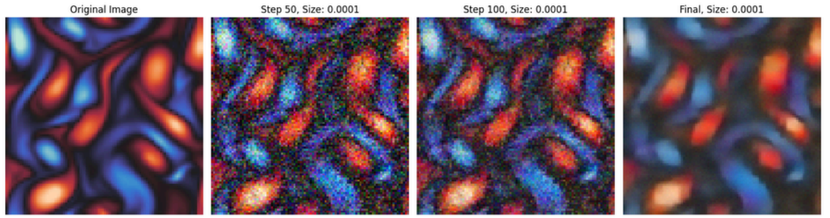
3 - Developed Denoising Algorithm

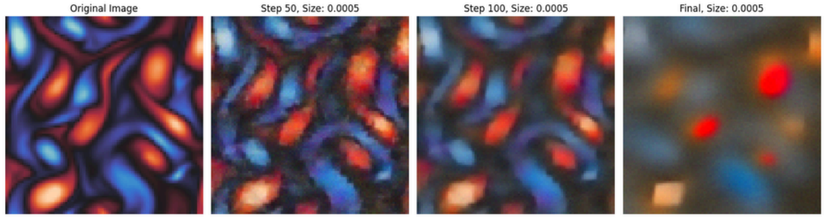
3.1. Denoising results - Step size impact

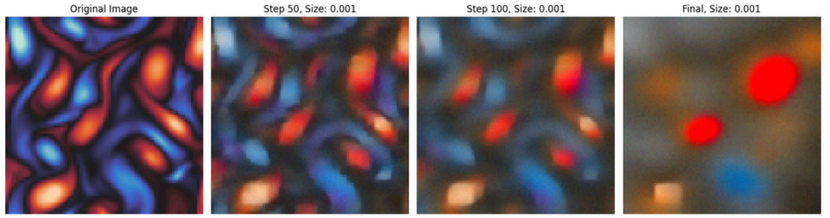
$$x_0 = \text{noisy\_image}$$

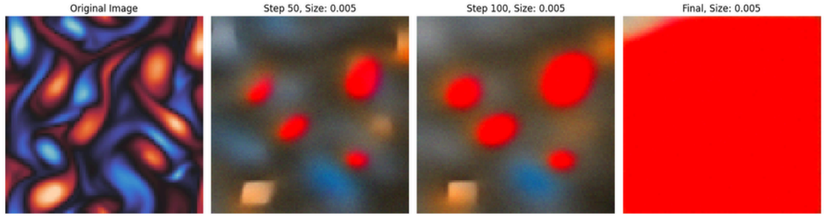
$$x_{t+1} = x_t + \eta \cdot s(x_t) + \xi \cdot \sqrt{2\eta}$$







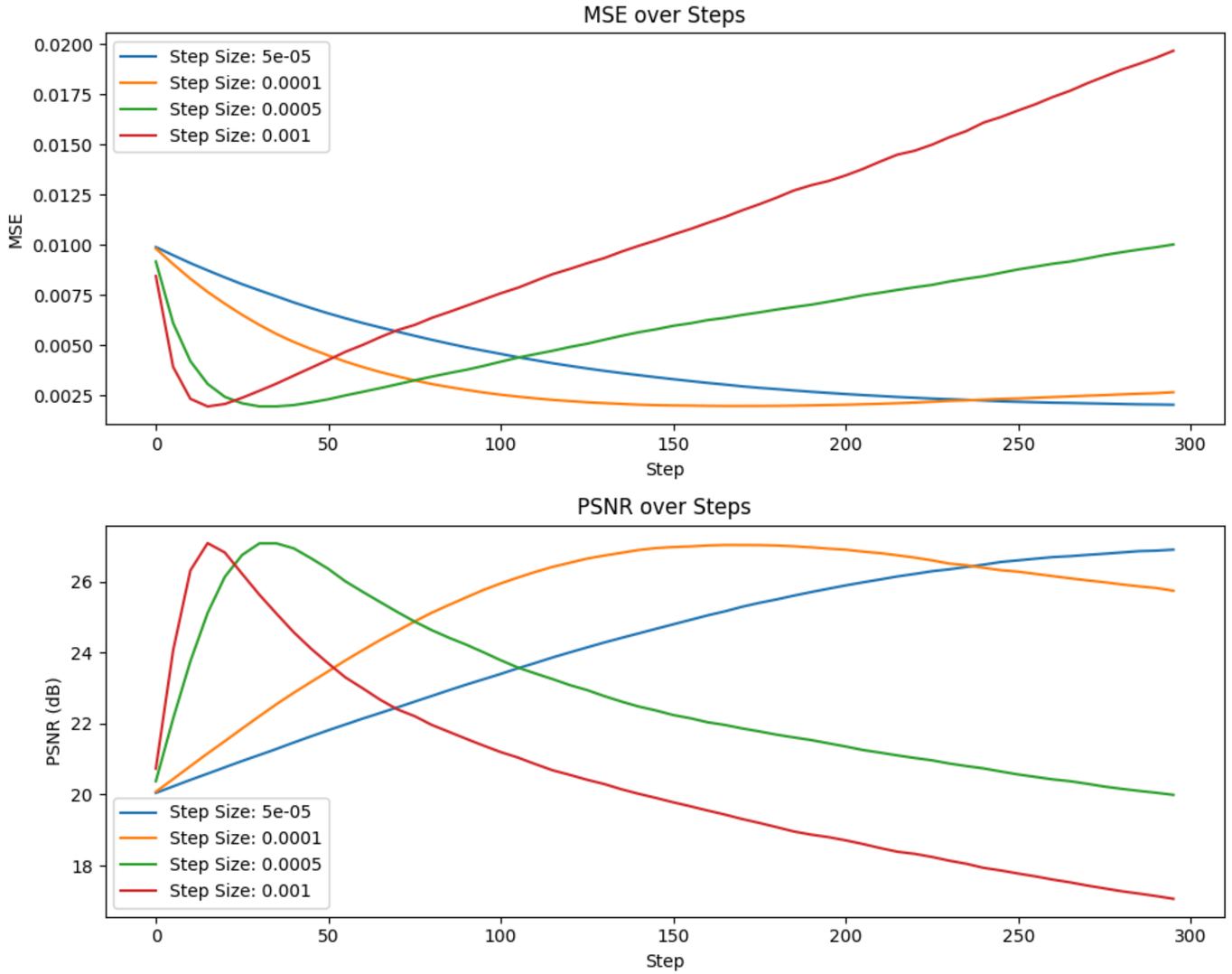




IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

3 - Developed Denoising Algorithm

3.1. Denoising results - Number of steps



IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

4 - Conclusion & Perspectives

Our Objective : apply Score-based methods for denoising

- Successfully implemented and demonstrated how **SDA** approach can **denoise, reconstruct** and **predict** turbulent fluid flows from partial or noisy observations.
- Visualizations show **strong recovery performance** with high fidelity to ground truth.
- Perspectives:

⚡ **Hardware Acceleration:** leverage GPU acceleration to reduce training and sampling time.

🌍 **Real-World Applications:** apply to realistic geophysical data such as atmospheric, oceanic or climate models.

⚙️ **Scalability:** extend the framwork to support larger domains or higher-resolution simulations

5 - Q & A

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Do you have any question to ask us ?

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

THANK YOU FOR YOUR ATTENTION !