# Stabilization and filtering approaches in computational fluid dynamics

Anna Ivagnes

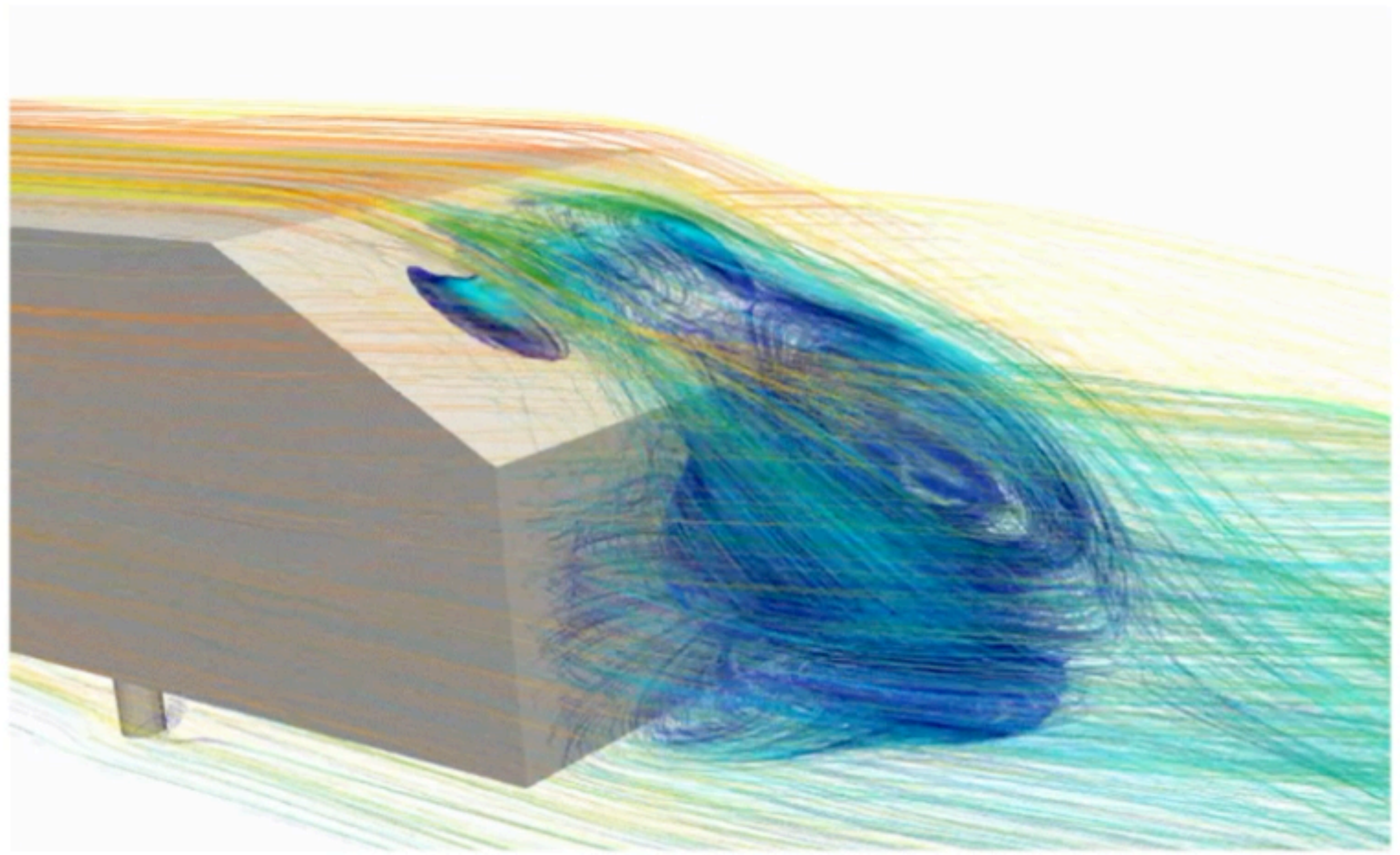
*17 December 2025*

SISSA mathLab

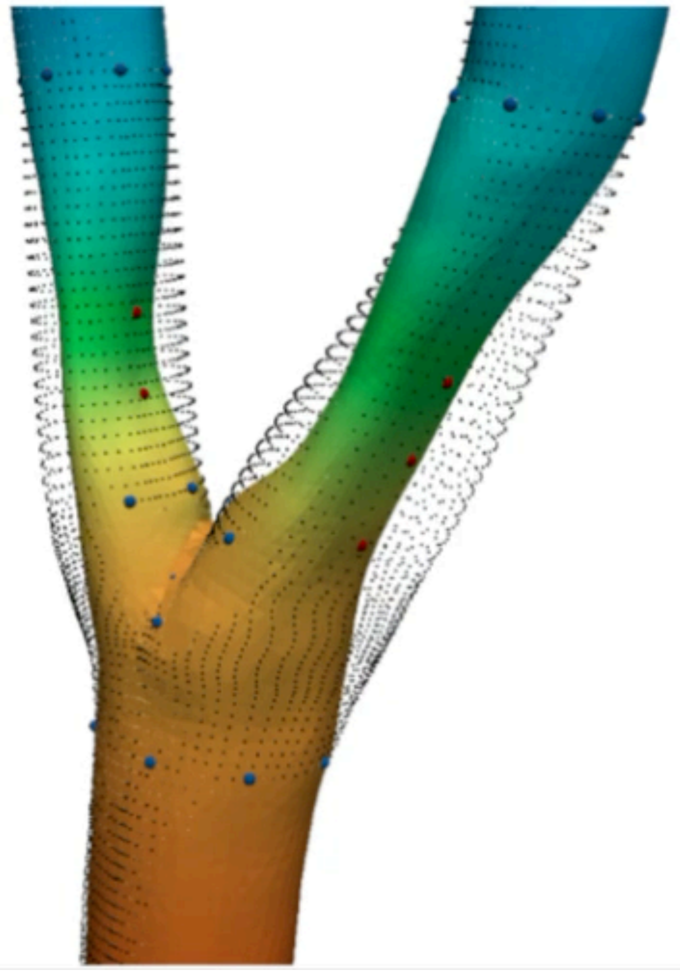
SISSA

# Mathematical problem

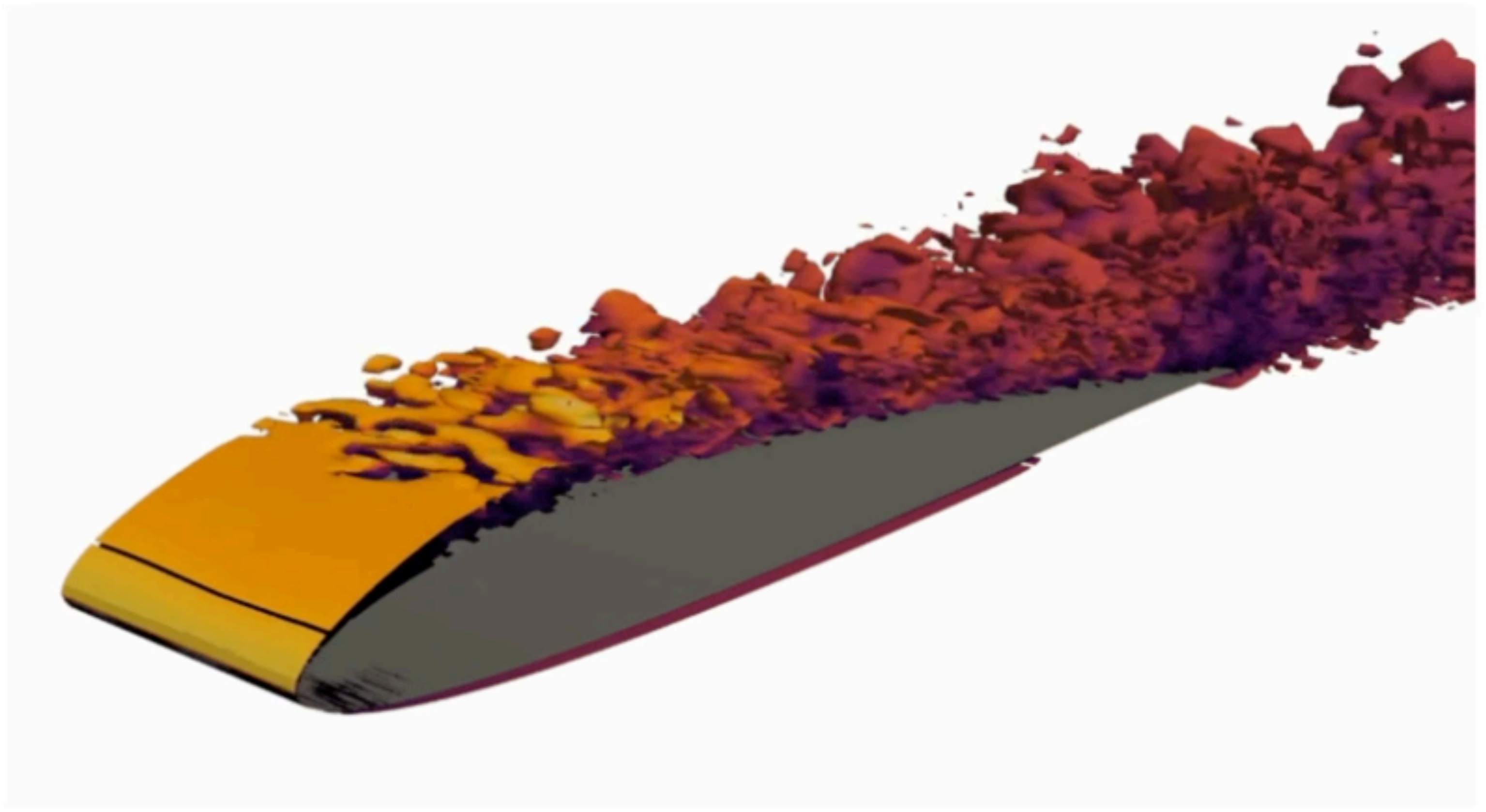
Partial Differential Equations (PDEs) are ubiquitous in many fields of Natural and Applied Sciences.



***Automotive***



***Biomedical***

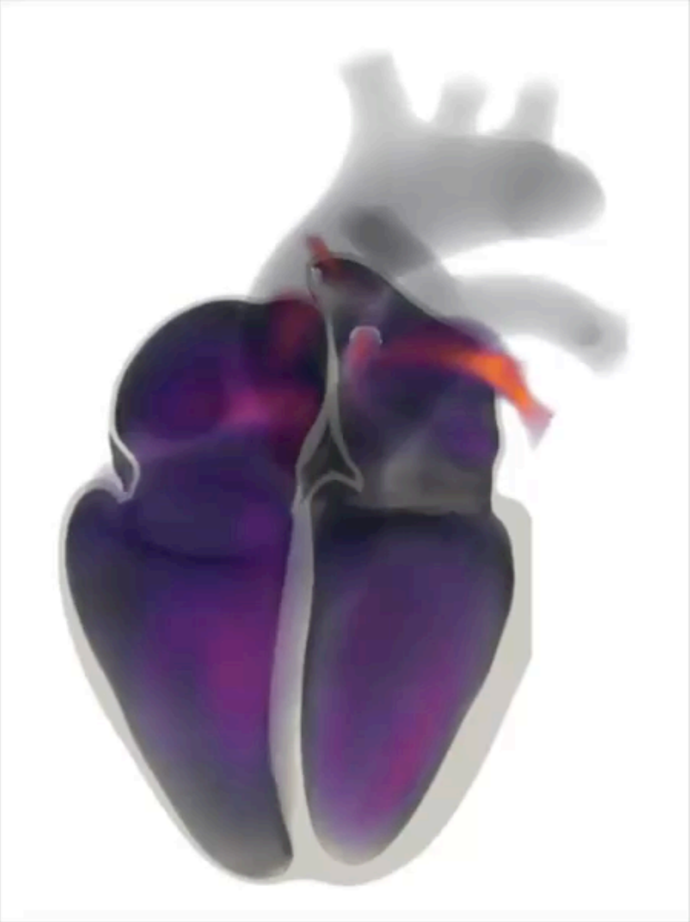


***Aeronautics***


- Zancanaro, Matteo, et al. "Hybrid neural network reduced order modelling for turbulent flows with geometric parameters." *Fluids* 6.8 (2021): 296.
- Ballarin, Francesco, et al. "Fast simulations of patient-specific haemodynamics of coronary artery bypass grafts based on a POD–Galerkin method and a vascular shape parametrization." *Journal of Computational Physics* 315 (2016): 609-628.
- Tonicello, Niccolò, et al. "Analysis of high-order explicit LES dynamic modeling applied to airfoil flows." *Flow, Turbulence and Combustion* 108.1 (2022): 77-104.

# LIMITATIONS OF CLASSICAL PDE SOLVERS

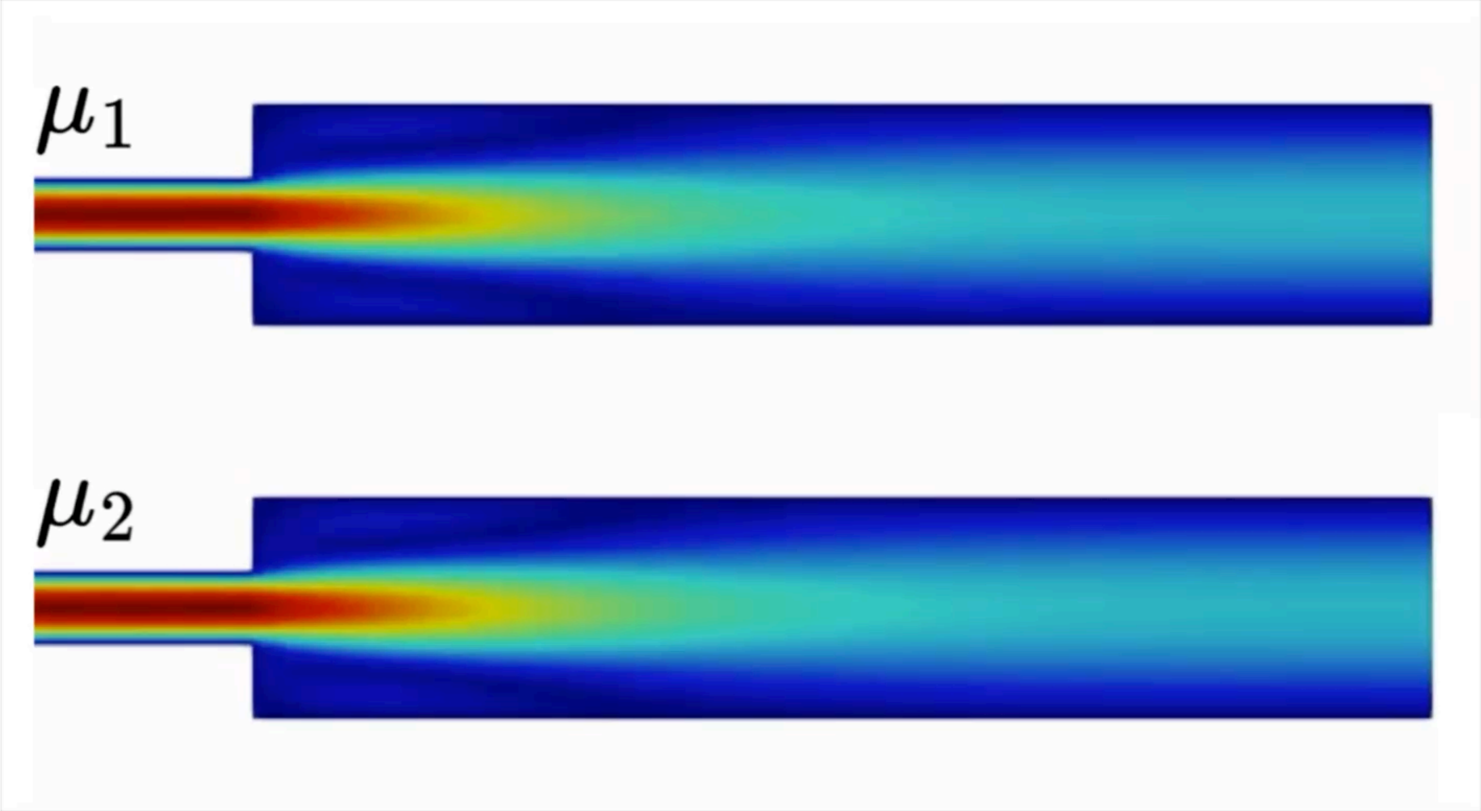
## COMPUTATIONAL POWER



One HeartBeat simulation takes ***1000 CPU cores*** for ***One day Compute***

Africa, Pasquale Claudio, et al. "lifex-cfd: An open-source computational fluid dynamics solver for cardiovascular applications." Computer Physics Communications 296 (2024): 109039.

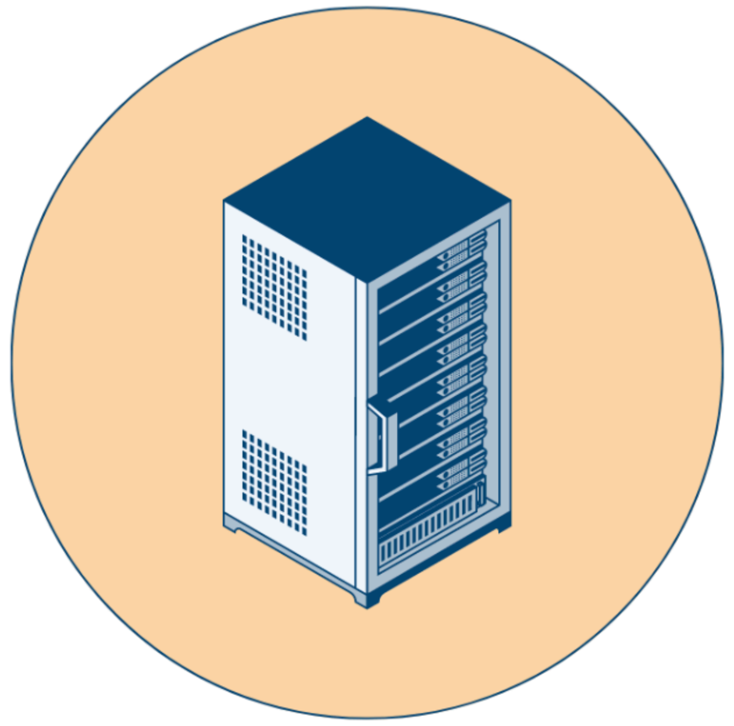
## PARAMETER DEPENDENCY




Classical solvers ***do not generalize solutions for multiple parameters***

Pichi, Federico, et al. "Driving bifurcating parametrized nonlinear PDEs by optimal control strategies: application to Navier–Stokes equations with model order reduction." ESAIM: Mathematical Modelling and Numerical Analysis 56.4 (2022): 1361-1400.

# Surrogate modeling



## Offline stage

### Collect high-fidelity data

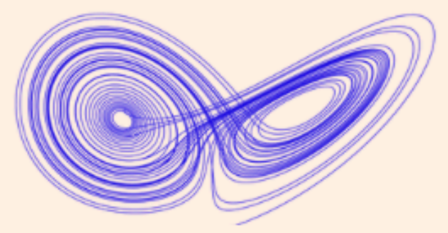
- Needs a *supercomputer*
- Computationally expensive
- Slow

Problem

$$\begin{cases} \dot{x} &= \sigma(y - x) \\ \dot{y} &= x(\rho - z) - y \\ \dot{z} &= xy - \beta z \end{cases}$$

$\xrightarrow{\text{slow}}$

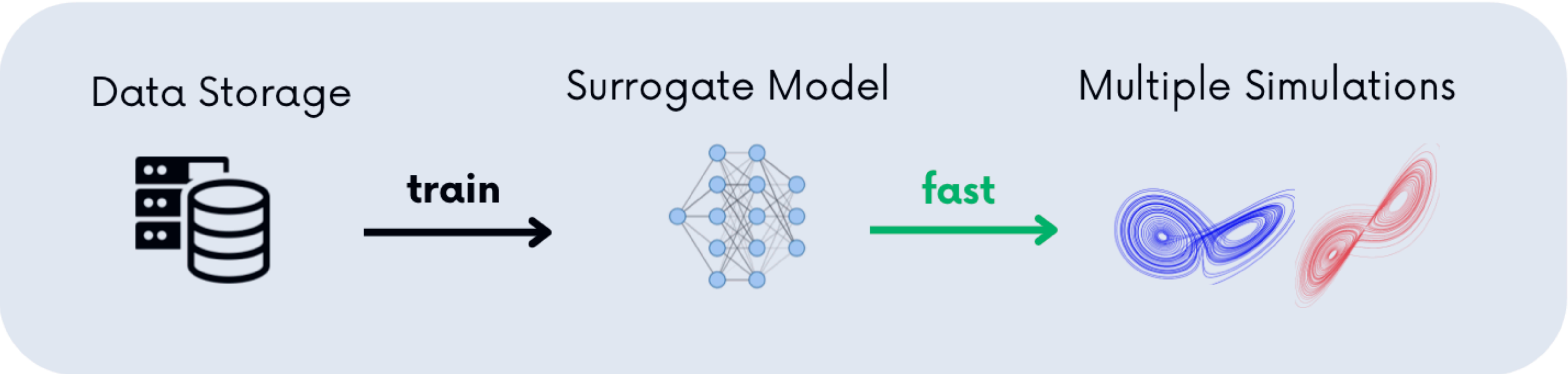
Single Simulation



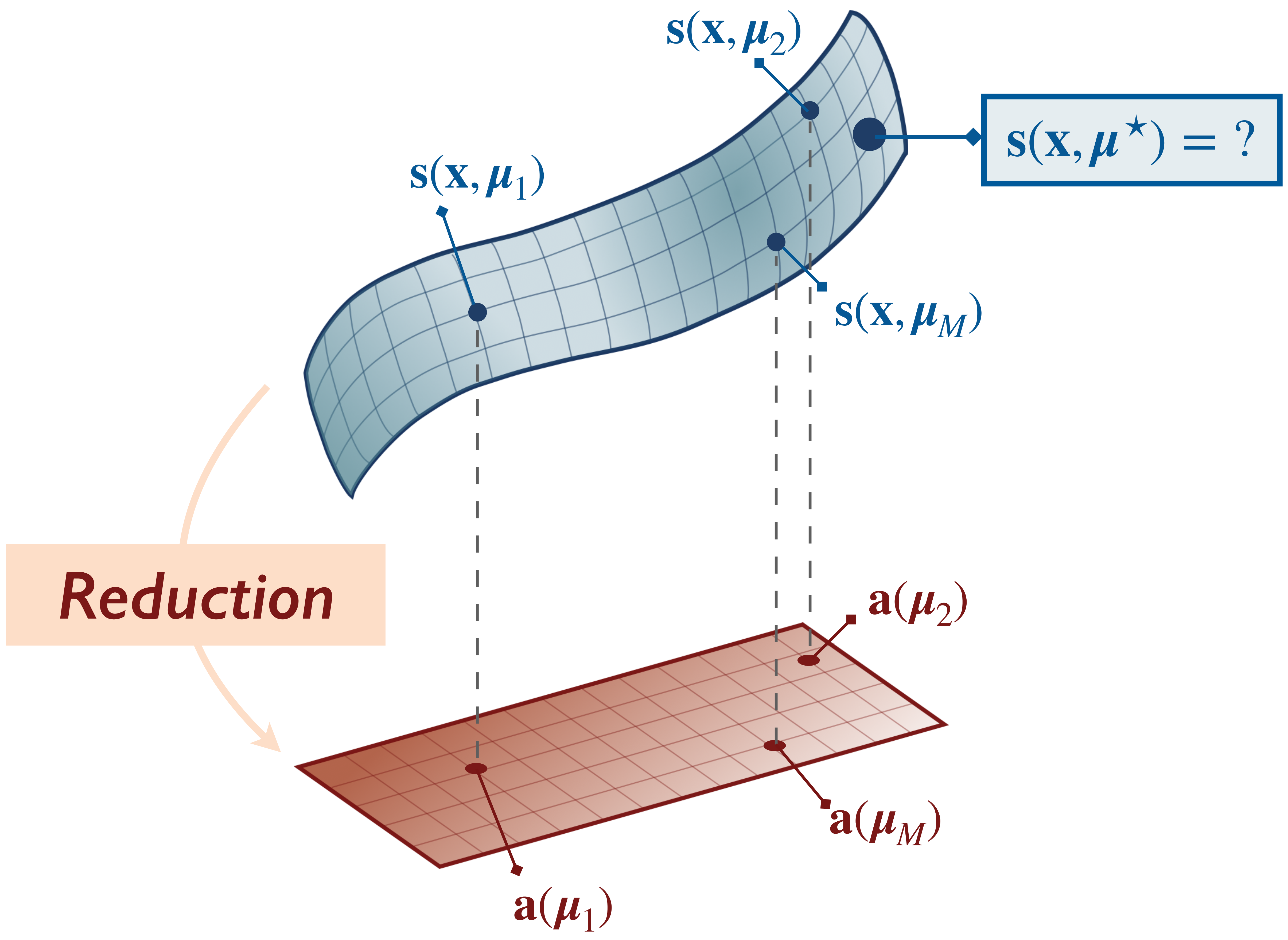
## Online stage

### The (fast) surrogate model gives the prediction for new parameters

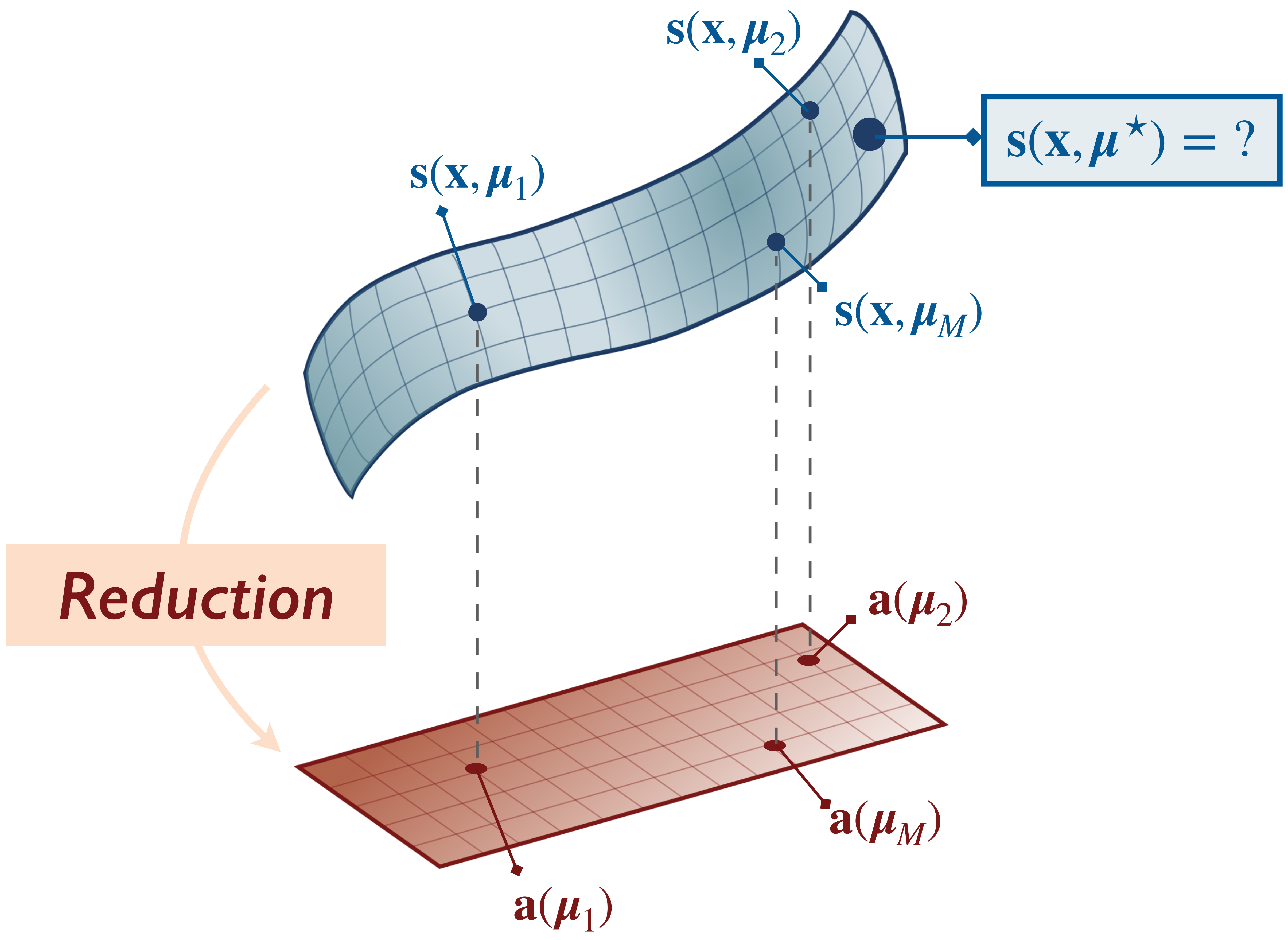
- Needs only a *laptop*
- Small computational resources
- Fast



# Reduced order models



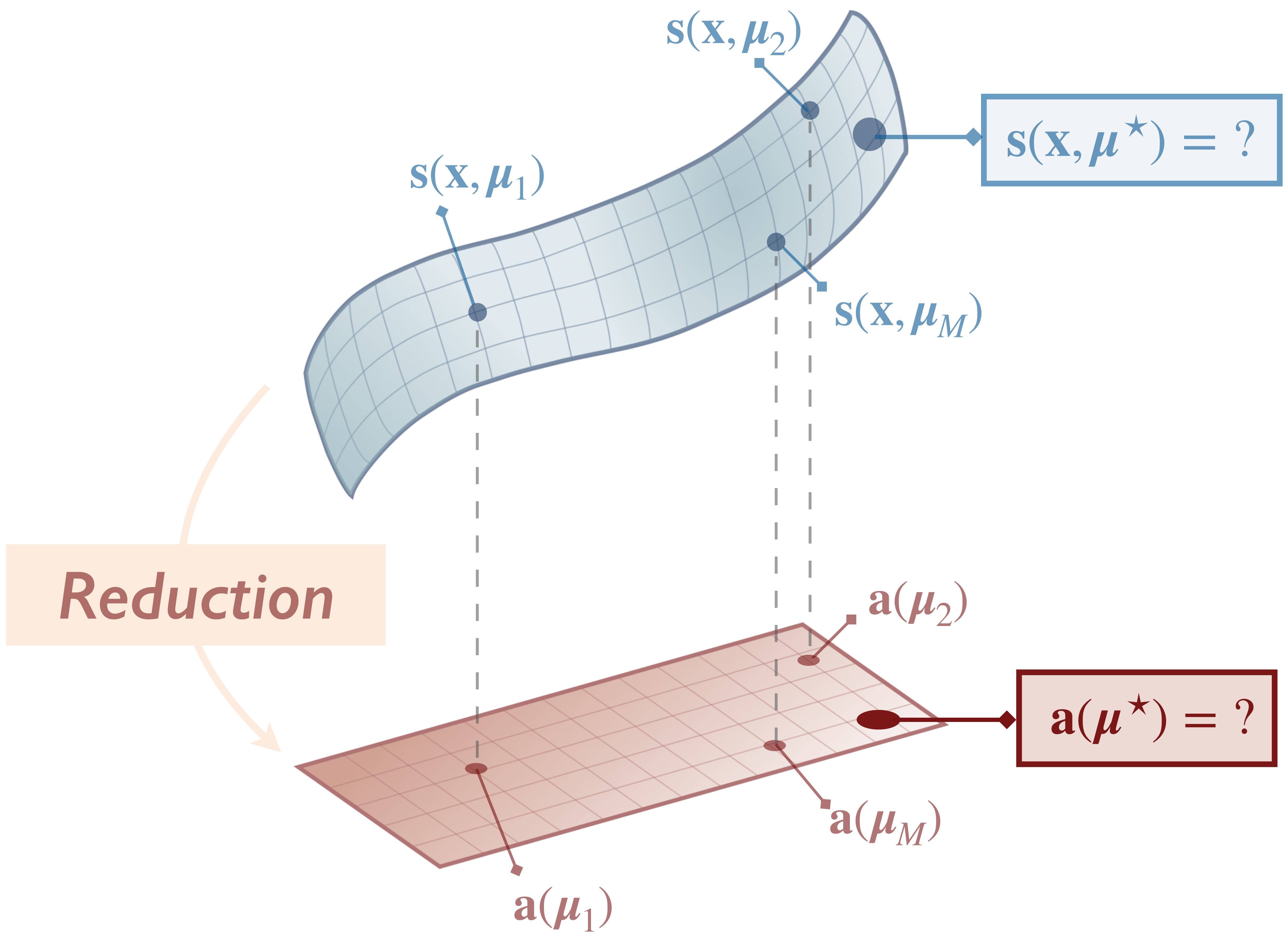
# Reduced order models




*In POD-based ROMs:* $\mathrm{s}(\mathbf{x}, \boldsymbol{\mu}^*) \simeq \sum_{i=1}^{r} a_i(\boldsymbol{\mu}^*)\, \boxed{\boldsymbol{\phi}_i(\mathbf{x})}$

*reduced basis*

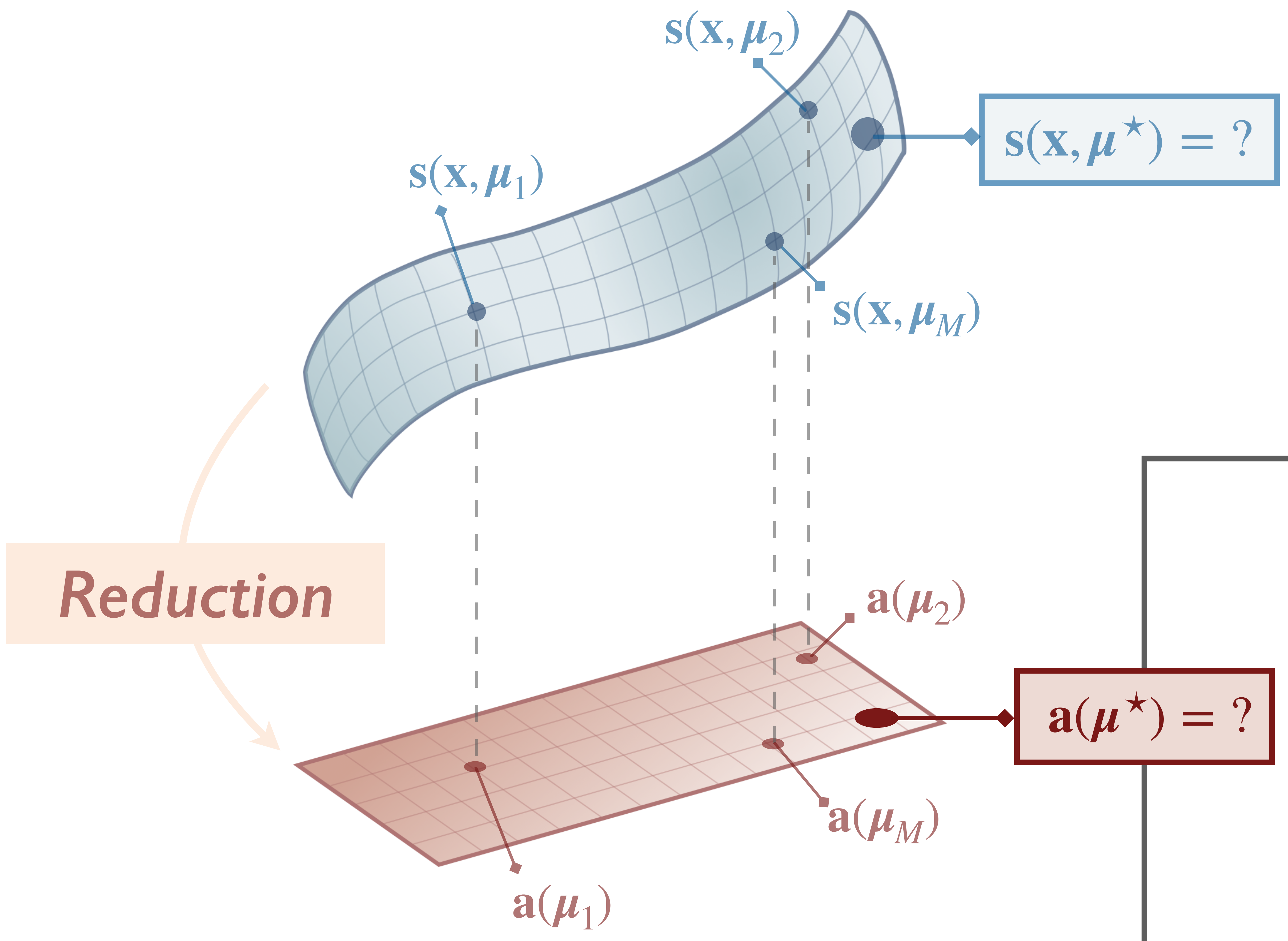
# Reduced order models

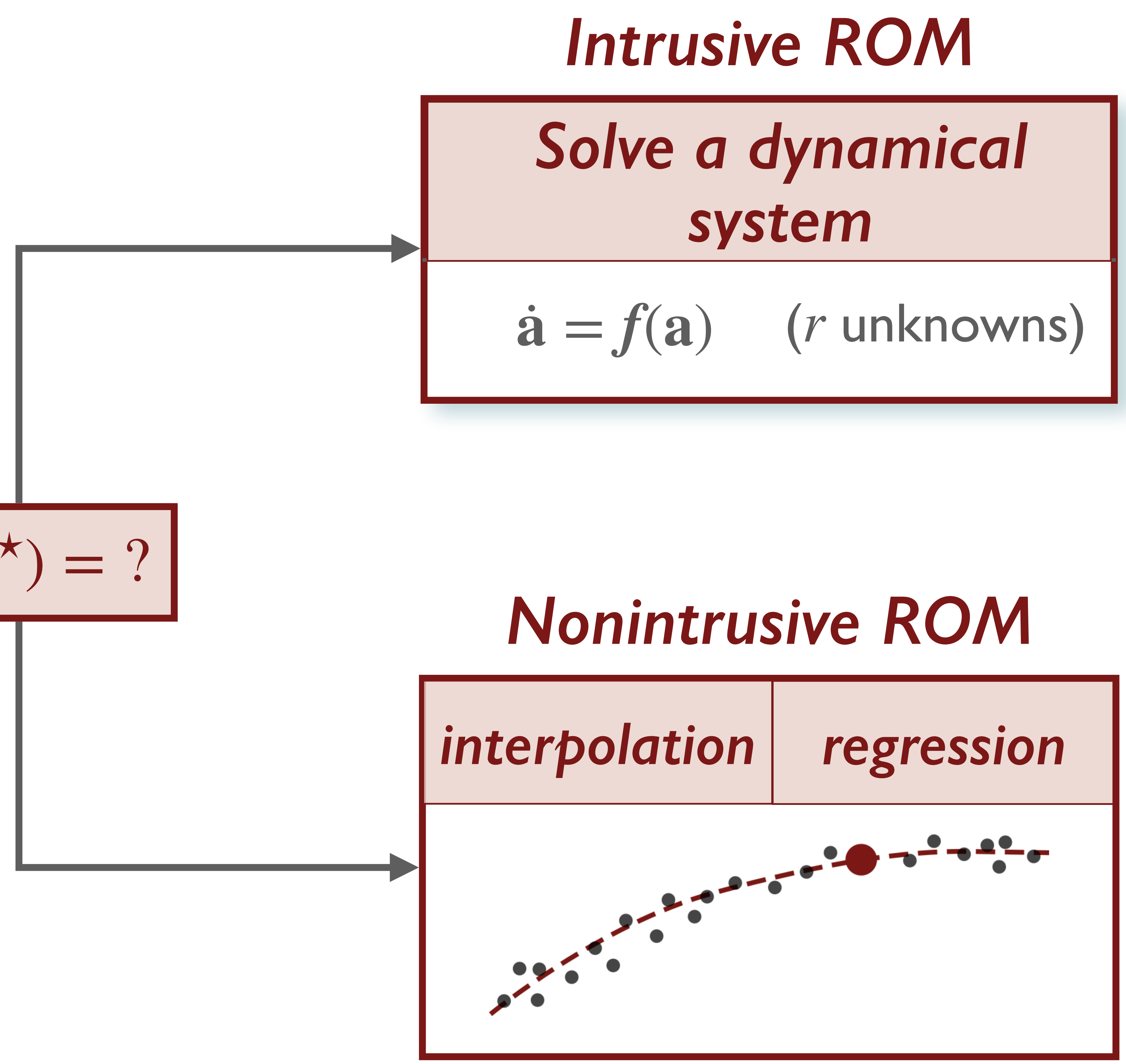



*In POD-based ROMs:* $\mathrm{s}(\mathbf{x}, \boldsymbol{\mu}^*) \simeq \sum_{i=1}^{r} a_i(\boldsymbol{\mu}^*)\,\boldsymbol{\phi}_i(\mathbf{x})$

*reduced basis*

# Reduced order models




*In POD-based ROMs:* $\mathbf{s}(\mathbf{x}, \boldsymbol{\mu}^*) \simeq \sum_{i=1}^{r} a_i(\boldsymbol{\mu}^*) \, \boldsymbol{\phi}_i(\mathbf{x})$

*reduced basis*

# Recent activities

## 1. *Nonlinear approaches for reduced order methods*

*with*
***Giovanni Stabile** (Sant'Anna), **Gianluigi Rozza** (SISSA)*

## 2. *Reinforcement learning for filters*

*with*
***Maria Strazzullo** (PoliTo), **Gianluigi Rozza** (SISSA)*

## 3. *Data-driven structure-preserving filtering approach*

*with*
***Toby van Gastelen, Syver Agdestein, Benjamin Sanderse** (CWI),*
***Giovanni Stabile** (Sant'Anna), **Gianluigi Rozza** (SISSA)*

# Recent activities

## 1. *Nonlinear approaches for reduced order methods*

*with*

*Giovanni Stabile (Sant'Anna), Gianluigi Rozza (SISSA)*

## 2. *Reinforcement learning for filters*

*with*

*Maria Strazzullo (PoliTo), Gianluigi Rozza (SISSA)*

## 3. *Data-driven structure-preserving filtering approach*

*with*

*Toby van Gastelen, Syver Agdestein, Benjamin Sanderse (CWI),*
*Giovanni Stabile (Sant'Anna), Gianluigi Rozza (SISSA)*

- *High* number of degrees of freedom: $N_{dof}$
- Modeling complex *PDEs (discretized Navier-Stokes)*

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} = \boldsymbol{F}(\mathbf{u}, p; \boldsymbol{\mu}^*), \\ \boldsymbol{C}(\mathbf{u}; \boldsymbol{\mu}^*) = \mathbf{0}\,. \end{cases}$$

# *Nonlinear approaches for reduced order methods*

- $N_u\,, N_p$:

  *reduced dimensions*

- *Velocity field:*

$$\mathbf{u}(\mathbf{x},\boldsymbol{\mu}) \simeq \sum_{i=1}^{N_u} a_i(\boldsymbol{\mu})\,\boldsymbol{\phi}_i(\mathbf{x})$$

- *Pressure field:*

$$p(\mathbf{x},\boldsymbol{\mu}) \simeq \sum_{i=1}^{N_p} b_i(\boldsymbol{\mu})\,\chi_i(\mathbf{x})$$

❖ *High* number of degrees of freedom: $N_{dof}$

❖ Modeling complex *PDEs (discretized Navier-Stokes)*

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} = \boldsymbol{F}(\mathbf{u}, p; \boldsymbol{\mu}^*), \\ \boldsymbol{C}(\mathbf{u}; \boldsymbol{\mu}^*) = \mathbf{0}\,. \end{cases}$$

# *Nonlinear approaches for reduced order methods*

- $N_u\,, N_p$:

  *reduced dimensions*

- *Velocity field:*

$$\mathbf{u}(\mathbf{x},\boldsymbol{\mu}) \simeq \sum_{i=1}^{N_u} a_i(\boldsymbol{\mu})\,\boldsymbol{\phi}_i(\mathbf{x})$$

- *Pressure field:*

$$p(\mathbf{x},\boldsymbol{\mu}) \simeq \sum_{i=1}^{N_p} b_i(\boldsymbol{\mu})\,\chi_i(\mathbf{x})$$

❖ *High* number of degrees of freedom: $N_{dof}$

❖ Modeling complex *PDEs (discretized Navier-Stokes)*

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} = \boldsymbol{F}(\mathbf{u}, p; \boldsymbol{\mu}^*), \\ \boldsymbol{C}(\mathbf{u}; \boldsymbol{\mu}^*) = \mathbf{0}\,. \end{cases}$$

***Projection***
*of full order equation(s)*
*into the reduced space(s)*

❖ *Low* number of degrees of freedom: $N_u + N_p$

❖ Modeling *ODEs*

$$\begin{cases} \dot{\mathbf{a}} = \boldsymbol{f}(\mathbf{a}, \mathbf{b}), \\ \boldsymbol{c}(\mathbf{a}) = \mathbf{0}\,. \end{cases}$$

# *Nonlinear approaches for reduced order methods*

- $N_u\,, N_p\,, N_{\nu_t}$ :
  *reduced dimensions*

- *Velocity field:*
  $$\mathbf{u}(\mathbf{x},\boldsymbol{\mu}) \simeq \sum_{i=1}^{N_u} a_i(\boldsymbol{\mu})\,\boldsymbol{\phi}_i(\mathbf{x})$$

- *Pressure field:*
  $$p(\mathbf{x},\boldsymbol{\mu}) \simeq \sum_{i=1}^{N_p} b_i(\boldsymbol{\mu})\,\chi_i(\mathbf{x})$$

- *Eddy viscosity field:*
  $$\nu_t \sim \sum_{i=1}^{N_{\nu_t}} g_i(\boldsymbol{\mu})\,\eta_i(\mathbf{x})$$

*State-of-the-art:* ***EV-ROM***

$$\begin{cases} \dot{\mathbf{a}} = \boldsymbol{f}(\mathbf{a}, \mathbf{b}, \mathbf{g}) \\ \boldsymbol{c}(\mathbf{a}, \mathbf{b}, \mathbf{g}) = \mathbf{0} \end{cases}$$

***Turbulence map:***

$$\begin{bmatrix} \mathbf{a} \\ \boldsymbol{\mu} \end{bmatrix} \longrightarrow \boxed{\mathcal{G}} \longrightarrow \boxed{\mathbf{g}}$$

*Goal:* reintroducing the turbulent viscosity (RANS modeling)

# Nonlinear approaches for reduced order methods

- $N_u, N_p, N_{\nu_t}$:
  reduced dimensions

- Velocity field:
$$\mathbf{u}(\mathbf{x},\boldsymbol{\mu}) \simeq \sum_{i=1}^{N_u} a_i(\boldsymbol{\mu})\,\boldsymbol{\phi}_i(\mathbf{x})$$

- Pressure field:
$$p(\mathbf{x},\boldsymbol{\mu}) \simeq \sum_{i=1}^{N_p} b_i(\boldsymbol{\mu})\,\chi_i(\mathbf{x})$$

- Eddy viscosity field:
$$\nu_t \sim \sum_{i=1}^{N_{\nu_t}} g_i(\boldsymbol{\mu})\,\eta_i(\mathbf{x})$$

**DD-EV-ROM**

$$\begin{cases} \dot{\mathbf{a}} = f(\mathbf{a}, \mathbf{b}, \mathbf{g}) + \boldsymbol{\tau}_u \\ c(\mathbf{a}, \mathbf{b}, \mathbf{g}) + \boldsymbol{\tau}_p = 0 \end{cases}$$

- $N_u\,, N_p\,, N_{\nu_t}$ :
  reduced dimensions

- *Velocity field:*
  $$\mathbf{u}(\mathbf{x},\boldsymbol{\mu}) \simeq \sum_{i=1}^{N_u} a_i(\boldsymbol{\mu})\,\boldsymbol{\phi}_i(\mathbf{x})$$

- *Pressure field:*
  $$p(\mathbf{x},\boldsymbol{\mu}) \simeq \sum_{i=1}^{N_p} b_i(\boldsymbol{\mu})\,\chi_i(\mathbf{x})$$

- *Eddy viscosity field:*
  $$\nu_t \sim \sum_{i=1}^{N_{\nu_t}} g_i(\boldsymbol{\mu})\,\eta_i(\mathbf{x})$$

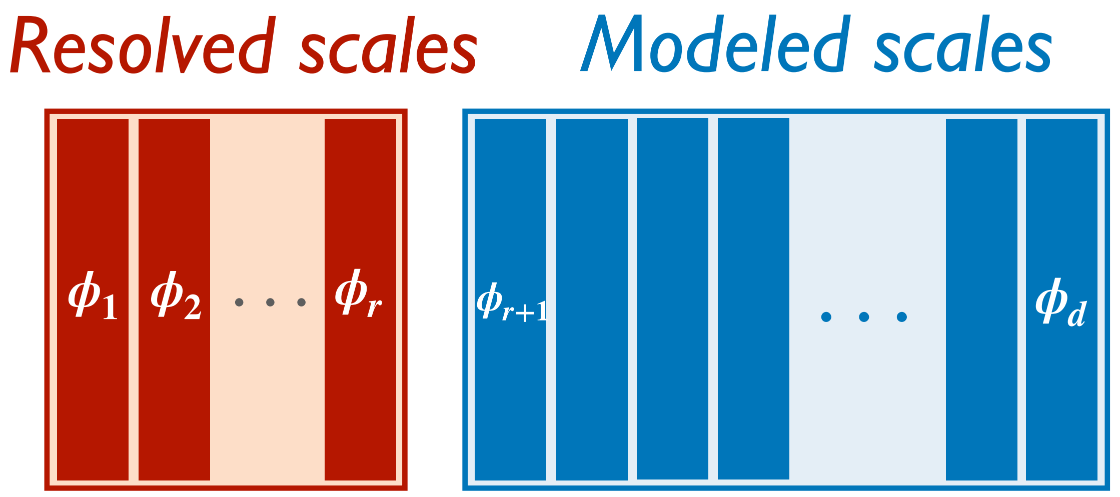
**DD-EV-ROM**

$$\begin{cases} \dot{\mathbf{a}} = f(\mathbf{a},\mathbf{b},\mathbf{g}) + \boldsymbol{\tau}_u \\ c(\mathbf{a},\mathbf{b},\mathbf{g}) + \boldsymbol{\tau}_p = 0 \end{cases}$$

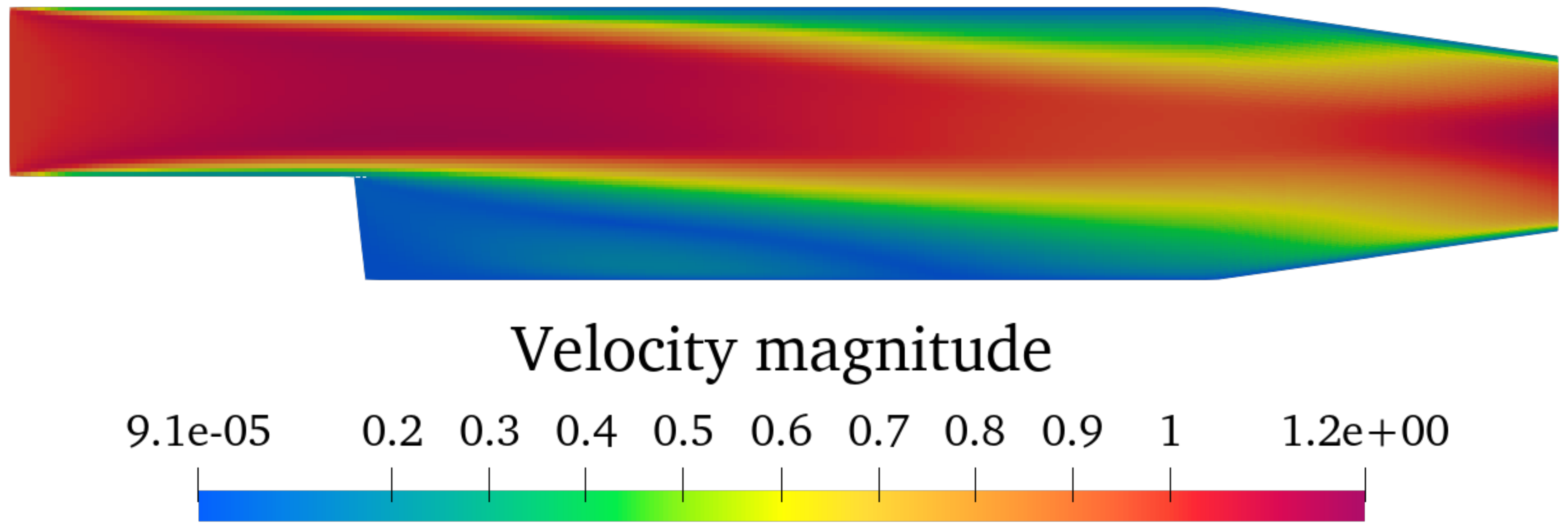
*Correction map:*

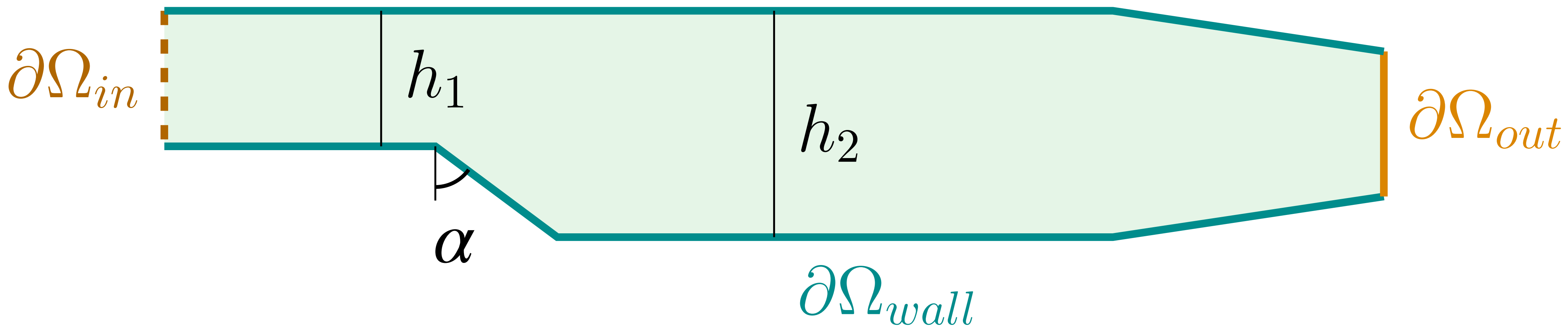
$$\begin{bmatrix} \mathbf{a} \\ \mathbf{g} \\ \boldsymbol{\mu} \end{bmatrix} \longrightarrow \mathcal{M} \longrightarrow \begin{bmatrix} \boldsymbol{\tau}_u \\ \boldsymbol{\tau}_p \end{bmatrix}$$

***Goal:*** reintroducing the contribution of the neglected modes



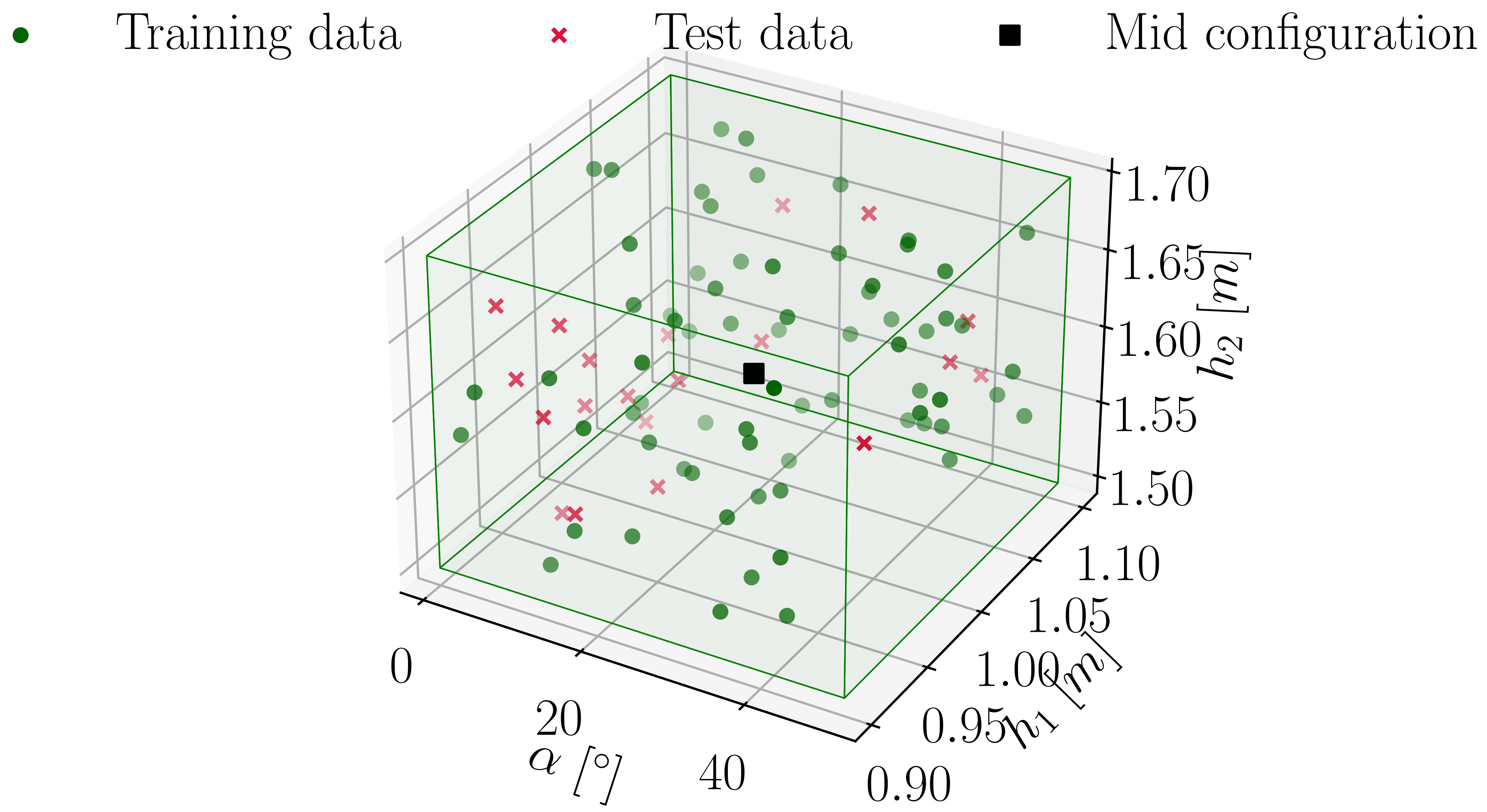
***Test case:*** 2D steady backward facing step




Geometrical parameterization:

$$\boldsymbol{\mu} = (\alpha, h_1, h_2)$$




Training and test parameters in the backstep test case.

A. Ivagnes, G. Stabile and G. Rozza. *Data-driven Closure Strategies for Parametrized Reduced Order Models via Deep Operator Networks*, arXiv:2505.1730, under review.

# *Nonlinear approaches for reduced order methods*

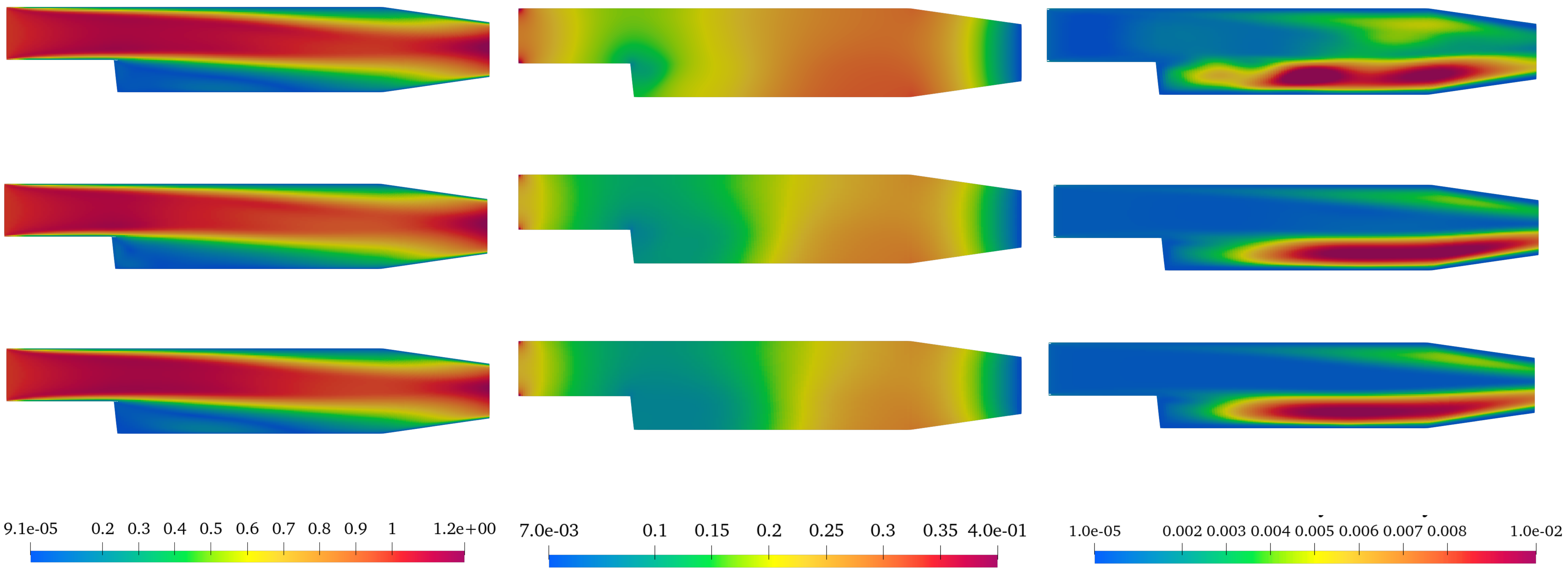
Graphical results for a ***test parameter*** at $N_u = 4, N_p = 5, N_{\nu_t} = 20$

| | *Velocity magnitude* | *Pressure* | *Eddy viscosity* |
|---|---|---|---|

***EV-ROM***

***DD-EV-ROM***

***FOM (reference)***




A. Ivagnes, G. Stabile and G. Rozza. *Data-driven Closure Strategies for Parametrized Reduced Order Models via Deep Operator Networks*, arXiv:2505.1730, under review.

# Recent activities

## 1. Nonlinear approaches for reduced order methods

*with*
***Giovanni Stabile*** *(Sant'Anna),* ***Gianluigi Rozza*** *(SISSA)*

Sant'Anna
Scuola Universitaria Superiore Pisa

## 2. Reinforcement learning for filters

*with*
***Maria Strazzullo*** *(PoliTo),* ***Gianluigi Rozza*** *(SISSA)*

Politecnico di Torino
1859

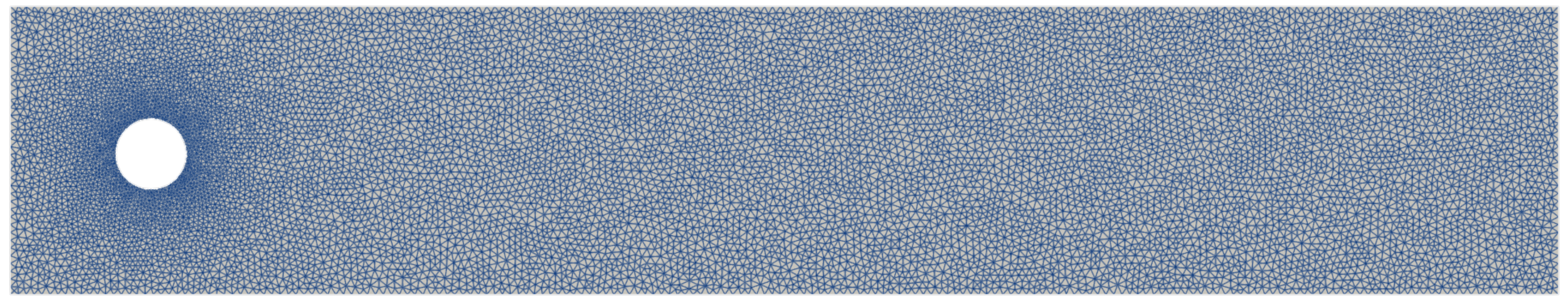
## 3. Data-driven structure-preserving filtering approach

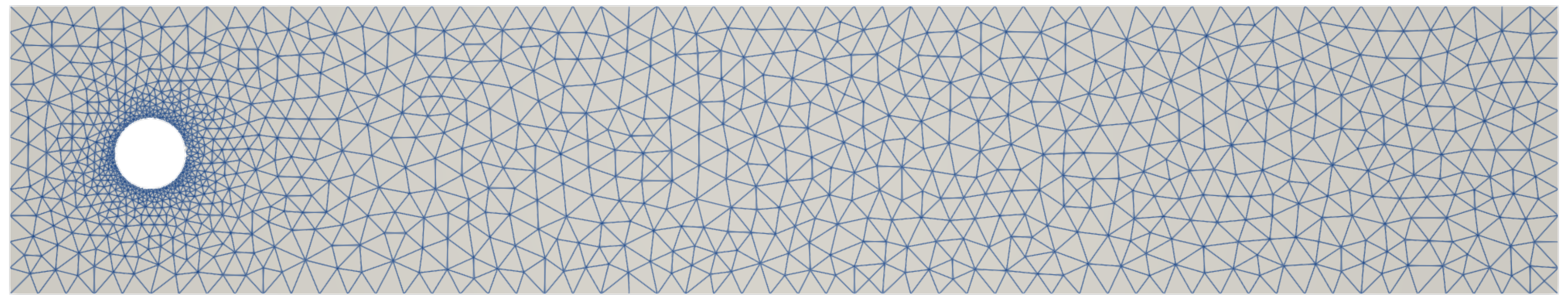
*with*
***Toby van Gastelen, Syver Agdestein, Benjamin Sanderse*** *(CWI),*
***Giovanni Stabile*** *(Sant'Anna),* ***Gianluigi Rozza*** *(SISSA)*

CWI

Sant'Anna
Scuola Universitaria Superiore Pisa

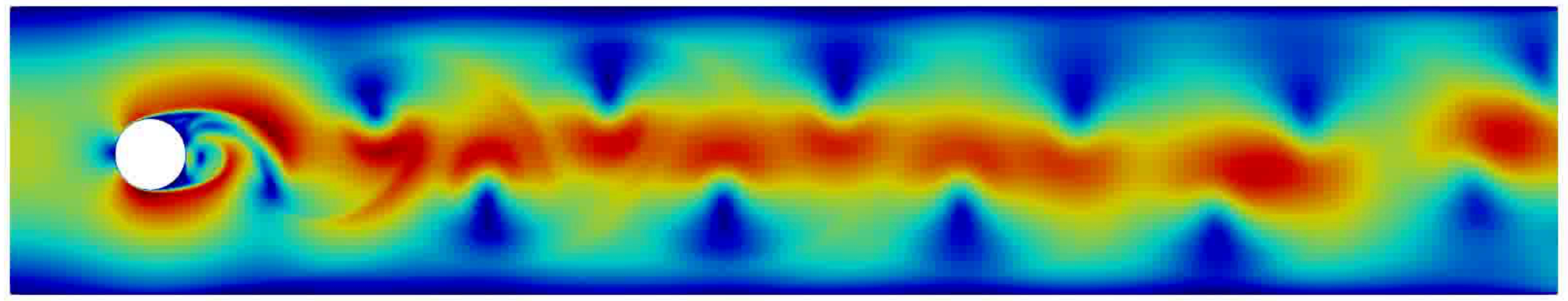
***Test case:*** *2D flow past a cylinder with parabolic inlet velocity,* ***Re=1000***

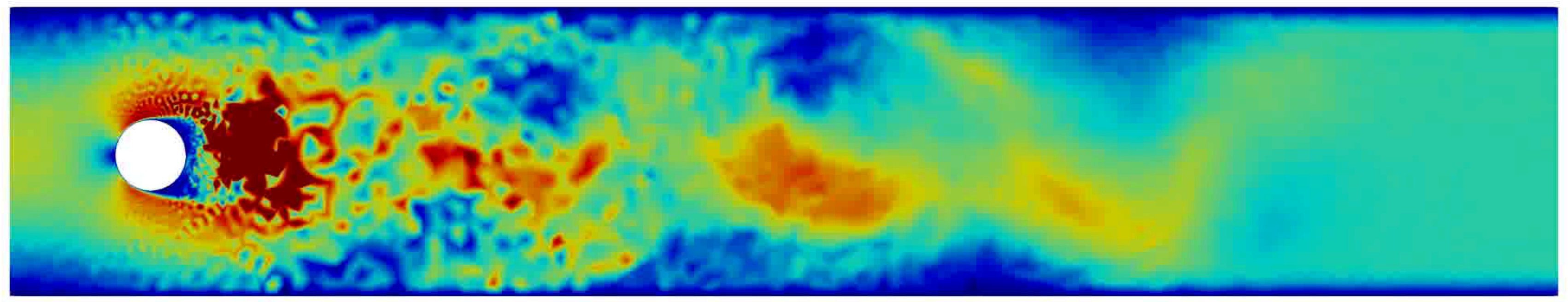




*FE simulation on a* ***fine*** *mesh*

*FE simulation on a* ***coarse*** *mesh*






***Problem:*** *regularize/improve the simulation on a coarse mesh*

# *Evolve-Filter(-Relax)*

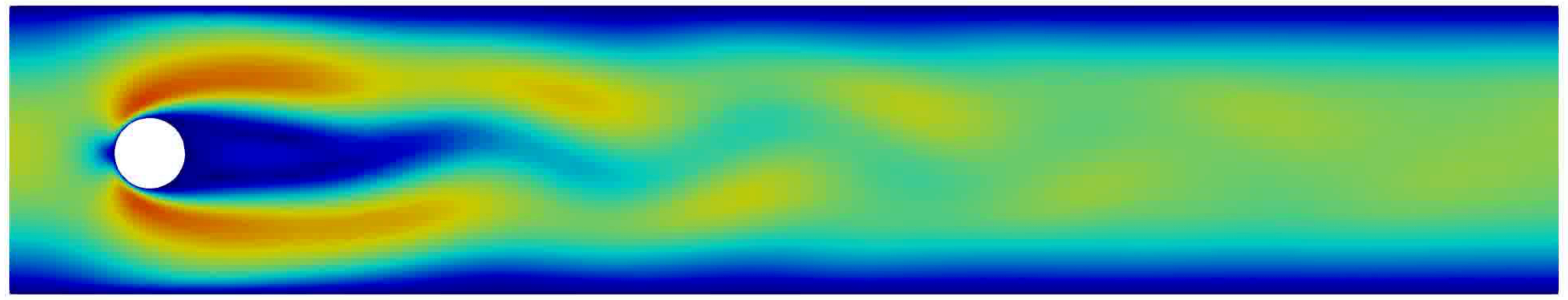
At each time step $t_n$:

$$
\begin{array}{lll}
\textbf{(I)} & \textit{Evolve}: & \begin{cases} \dfrac{\boldsymbol{w}^{n+1}-\boldsymbol{u}^{n}}{\Delta t} + (\boldsymbol{w}^{n+1}\cdot\nabla)\boldsymbol{w}^{n+1} - \nu\Delta\boldsymbol{w}^{n+1} + \nabla p^{n+1} = 0 & \text{in } \Omega\times\{t_{n+1}\}, \\ \nabla\cdot\boldsymbol{w}^{n+1} = 0 & \text{in } \Omega\times\{t_{n+1}\}, \end{cases} \\
\textbf{(II)} & \textit{Filter:} & \begin{cases} -2\delta^2\,\Delta\overline{\boldsymbol{w}}^{n+1} + \overline{\boldsymbol{w}}^{n+1} = \boldsymbol{w}^{n+1} & \text{in } \Omega\times\{t_{n+1}\}, \\ \overline{\boldsymbol{w}}^{n+1} = \boldsymbol{u}_D^{n+1} & \text{on } \partial\Omega_D\times\{t_{n+1}\}, \\ \dfrac{\partial\overline{\boldsymbol{w}}^{n+1}}{\partial\boldsymbol{n}} = 0 & \text{on } \partial\Omega_N\times\{t_{n+1}\}. \end{cases} \\
\textbf{(III)} & \textit{Relax:} & \boldsymbol{u}^{n+1} = (1-\chi)\,\boldsymbol{w}^{n+1} + \chi\,\overline{\boldsymbol{w}}^{n+1}\,,
\end{array}
$$

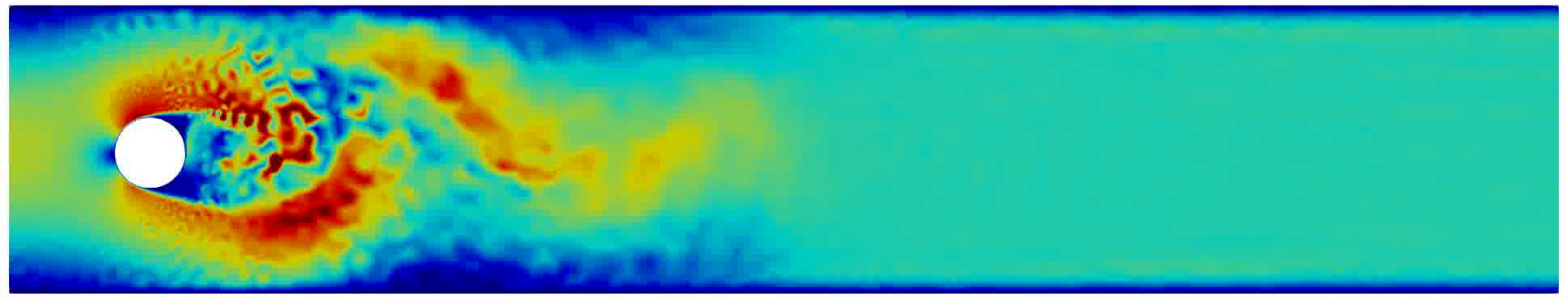
- ✳ $\delta$: filter radius
- ✳ $\chi$: relax parameter
- ✳ A bit of notation:
  - noEFR (or DNS): $\chi = 0$
  - EF: $\chi = 1$

***Standard choices:*** $\delta \sim h_{min}$ or $\delta \sim \eta$ (Kolmogorov), $\chi \sim \mathcal{O}(\Delta t)$

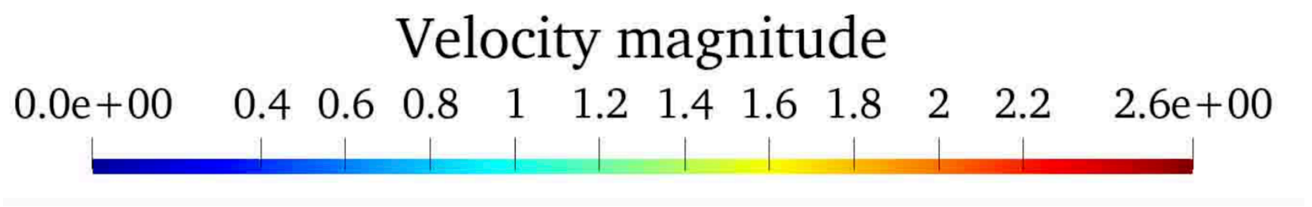
# Evolve-Filter(-Relax)



EF ($\delta = \eta$)



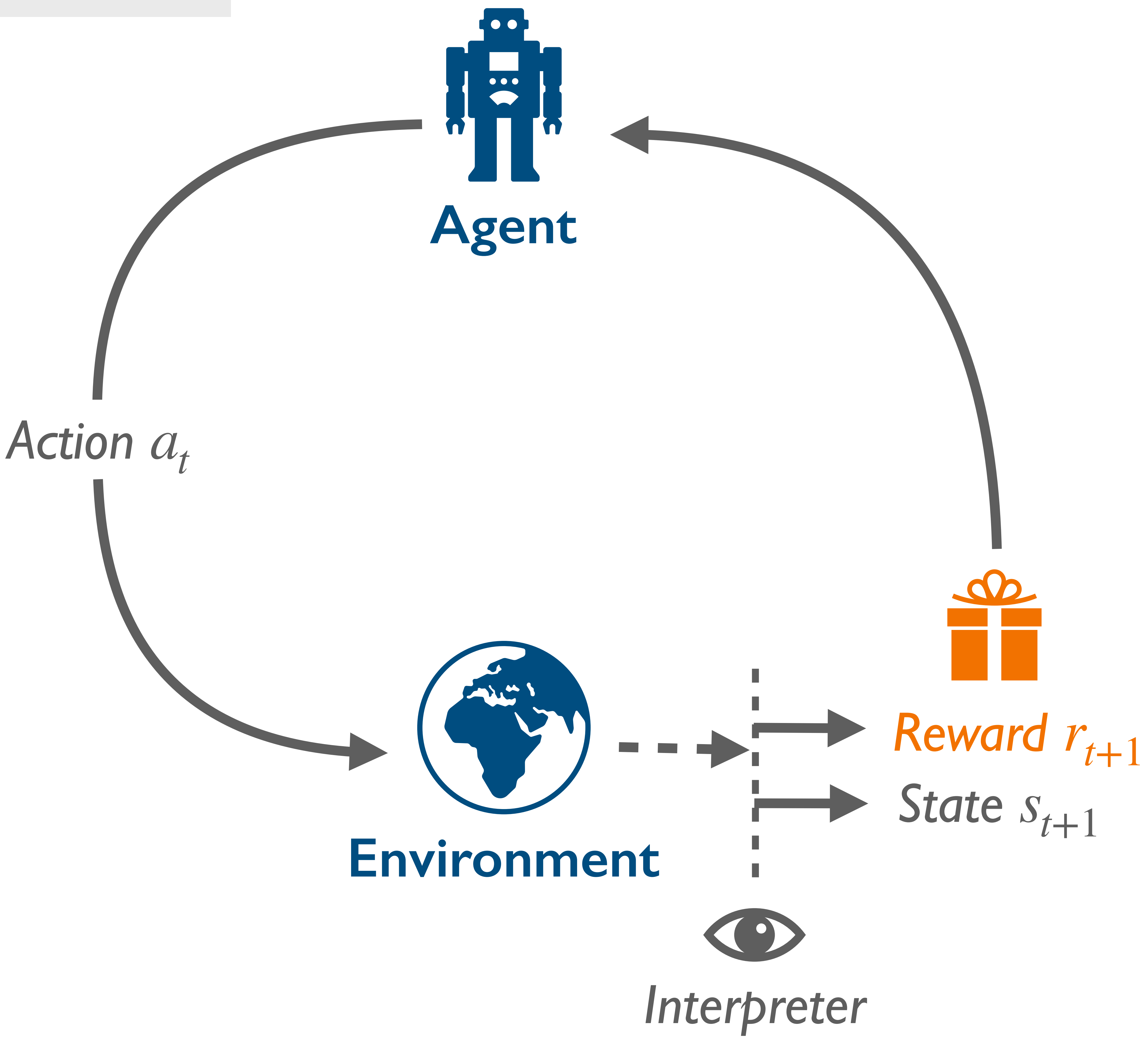
EFR ($\delta = \eta, \chi = 5\Delta t$)




*Standard choices* lead either to *overdiffusive* or to *noisy* results.

*We need to do something more:* $\delta(t)$ *and* $\chi(t)$

## How does RL work

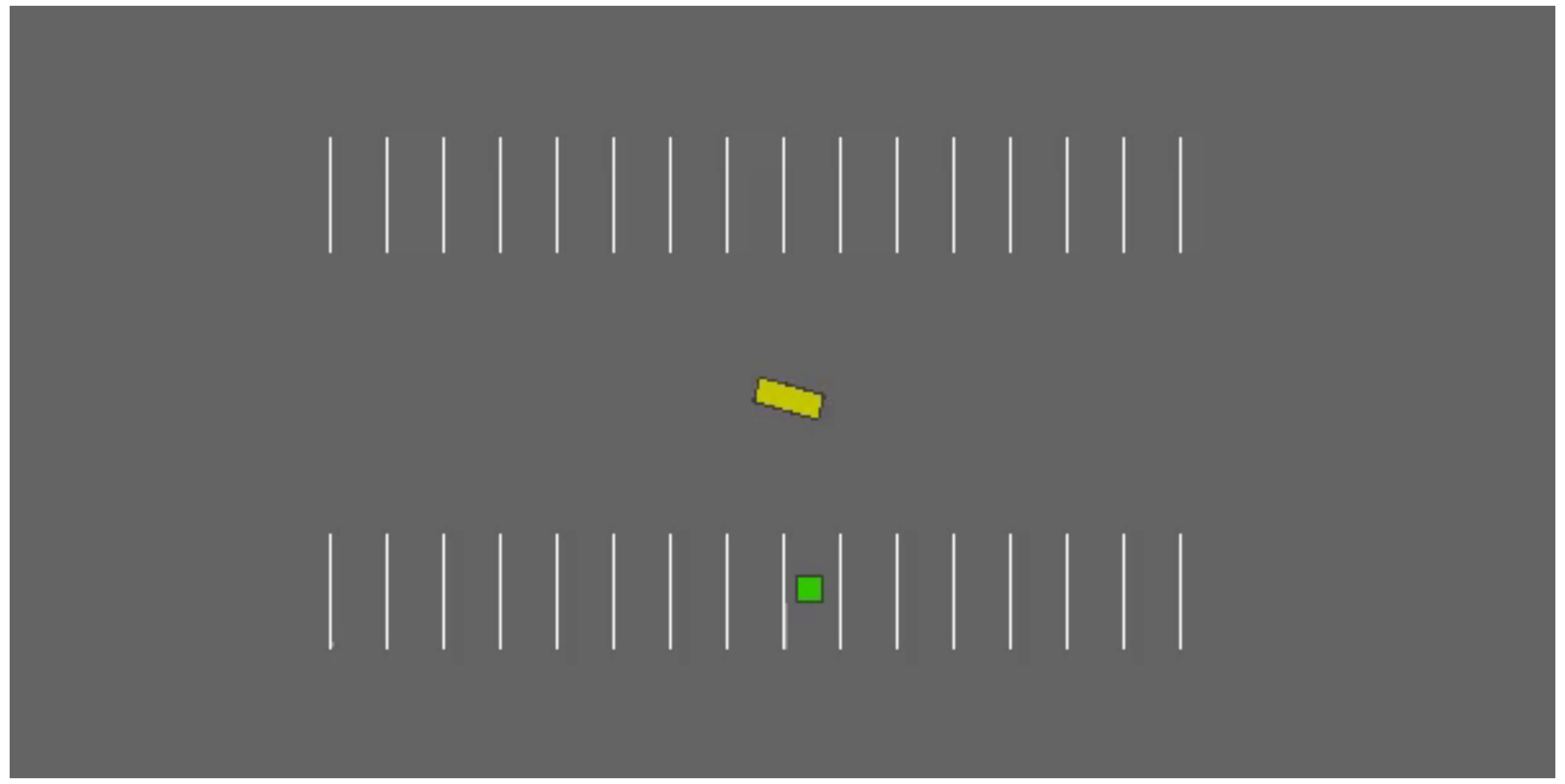



A set of multiple "*time steps*" is one RL ***episode***

The reward may be given:

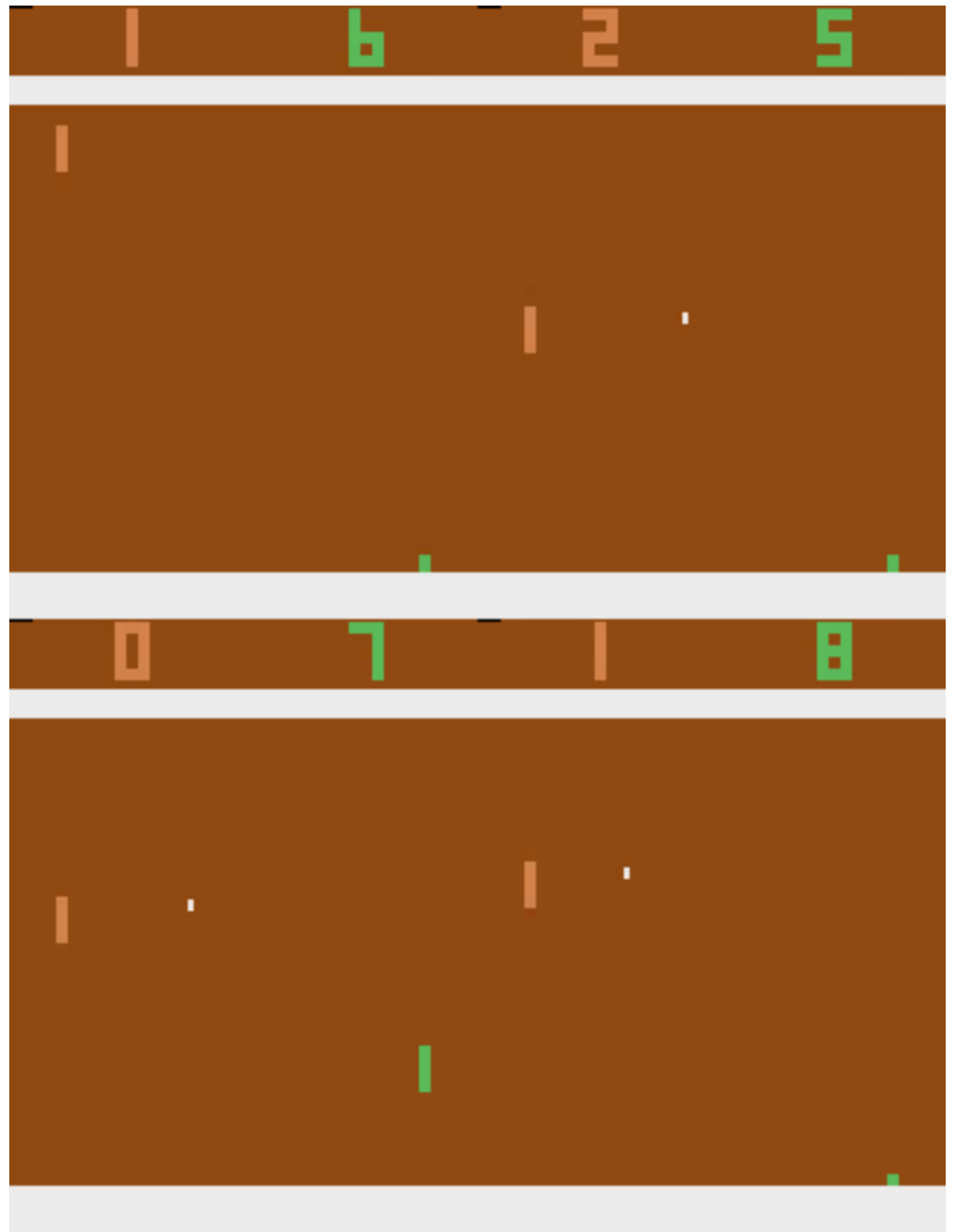
- at each time step (cumulative): $r_1, r_2, \ldots, r_t, \ldots, r_T$
- only at the end: $r_T$

## **How does RL work:** some examples



*Parking environment*


*Breakout*




*Pong environment*

Stable baselines3

# *Reinforcement learning for EF*

**How does RL work:** our case

**Agent**

*Action* $a_t$*:*

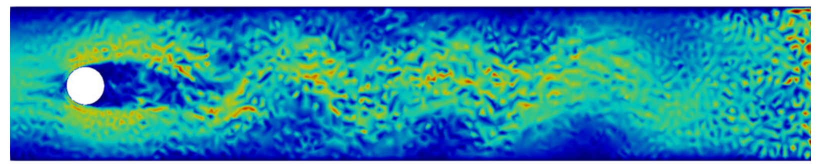
How much should I filter?
($\delta$ value)

**Environment**
*(EF solver)*

FEniCS project

*Reward* $r_{t+1}$*:*
$1 - \text{error}(u_{\text{ref}}(t+1), u_{\text{EF}}(t+1))$

*State* $s_{t+1}$*:*



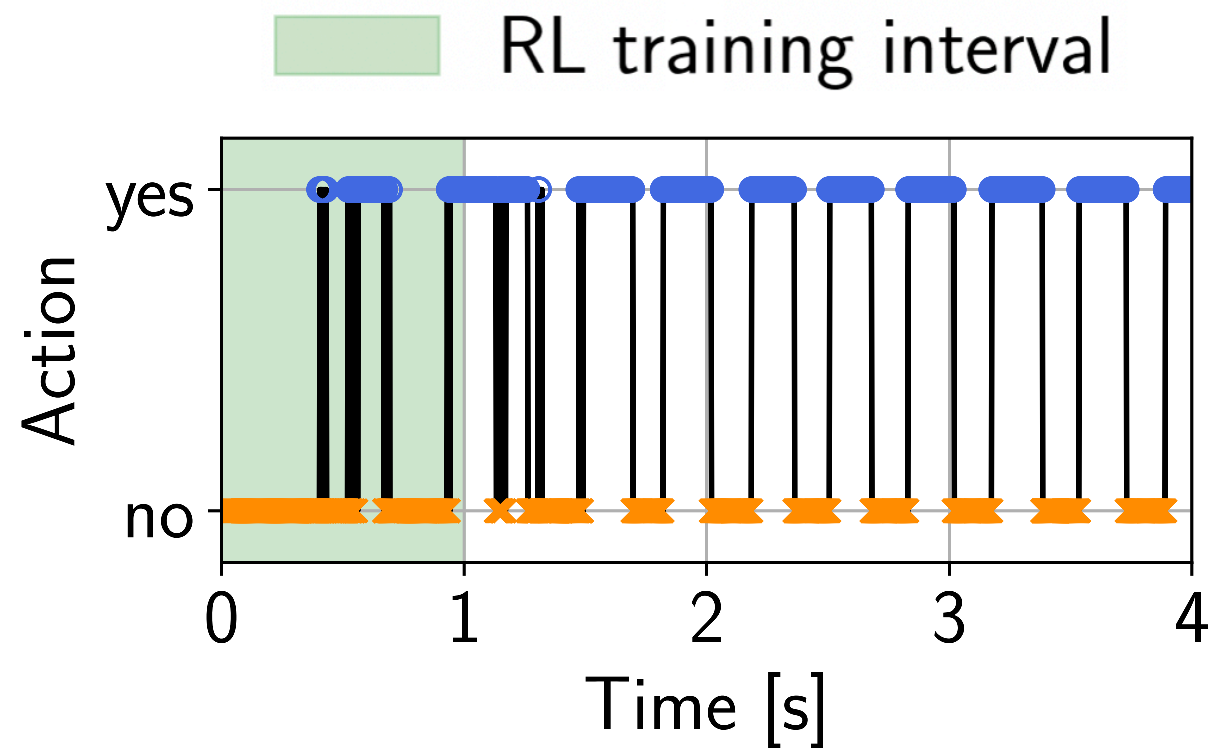
$u_{\text{EF}}(t+1)$

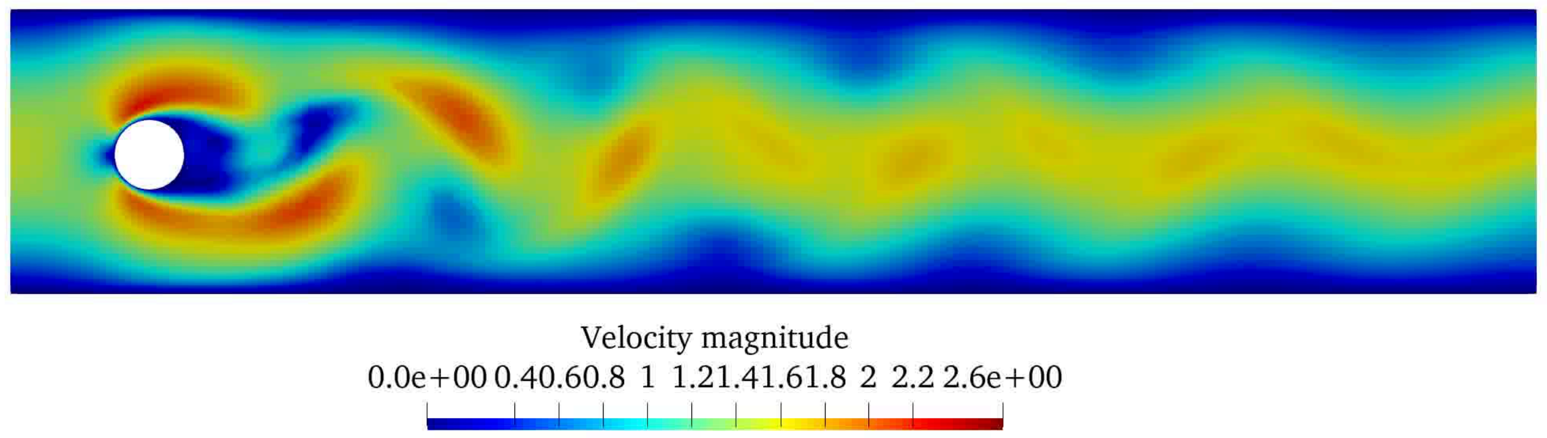
# *Reinforcement learning for EF*

## Binary action

✴ ***Action space*** $\mathscr{A} = \{a_1, a_2\}$

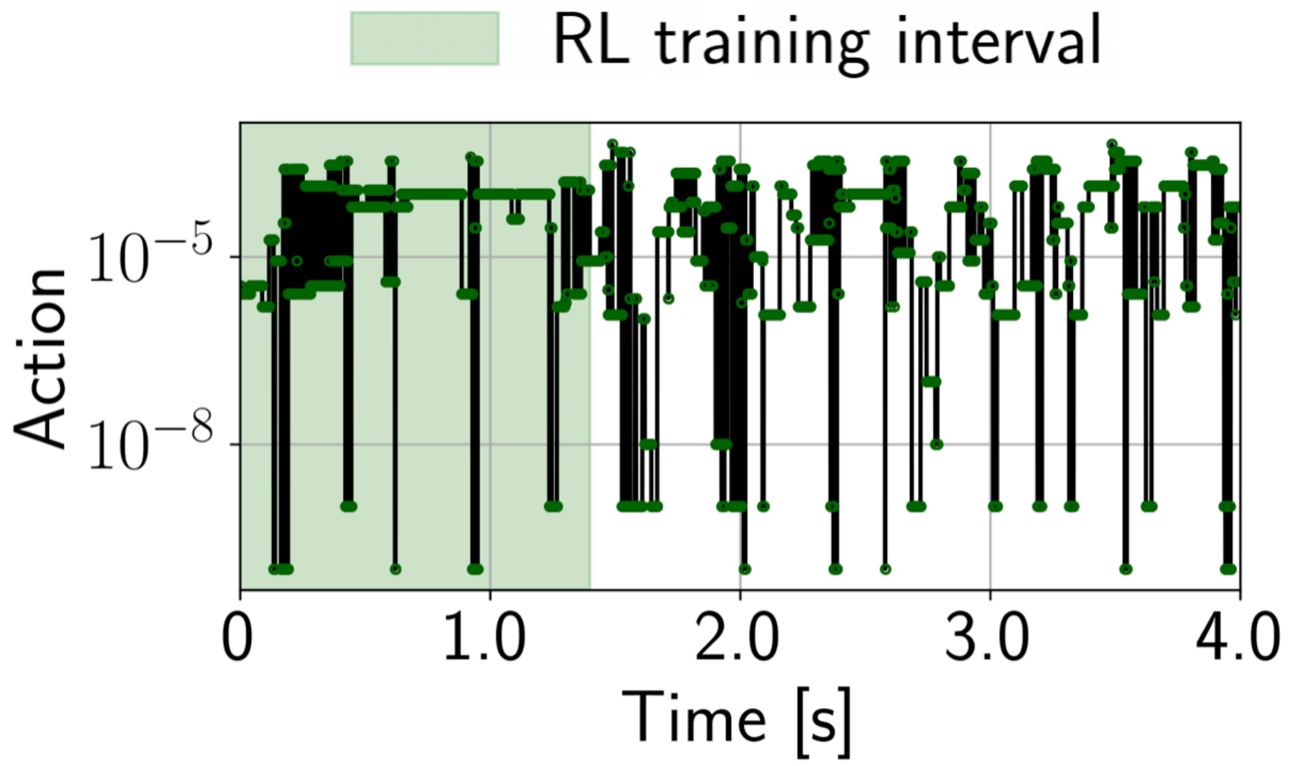
- $a_1 = 0 \rightarrow$ noEF
- $a_2 = 1 \rightarrow$ EF($\delta = \eta$)

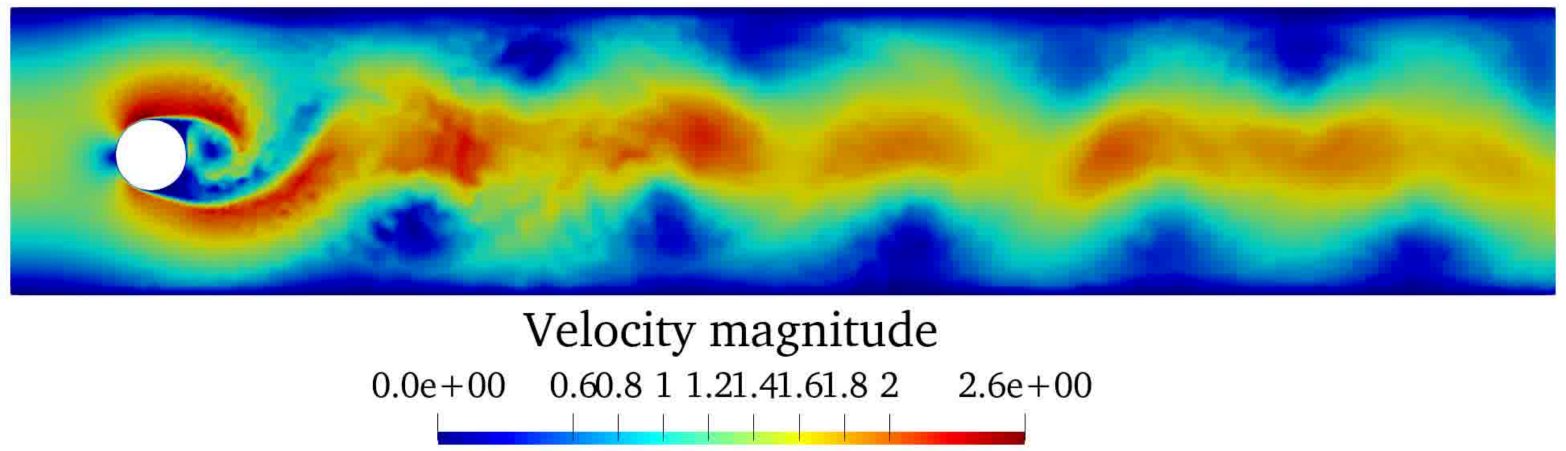


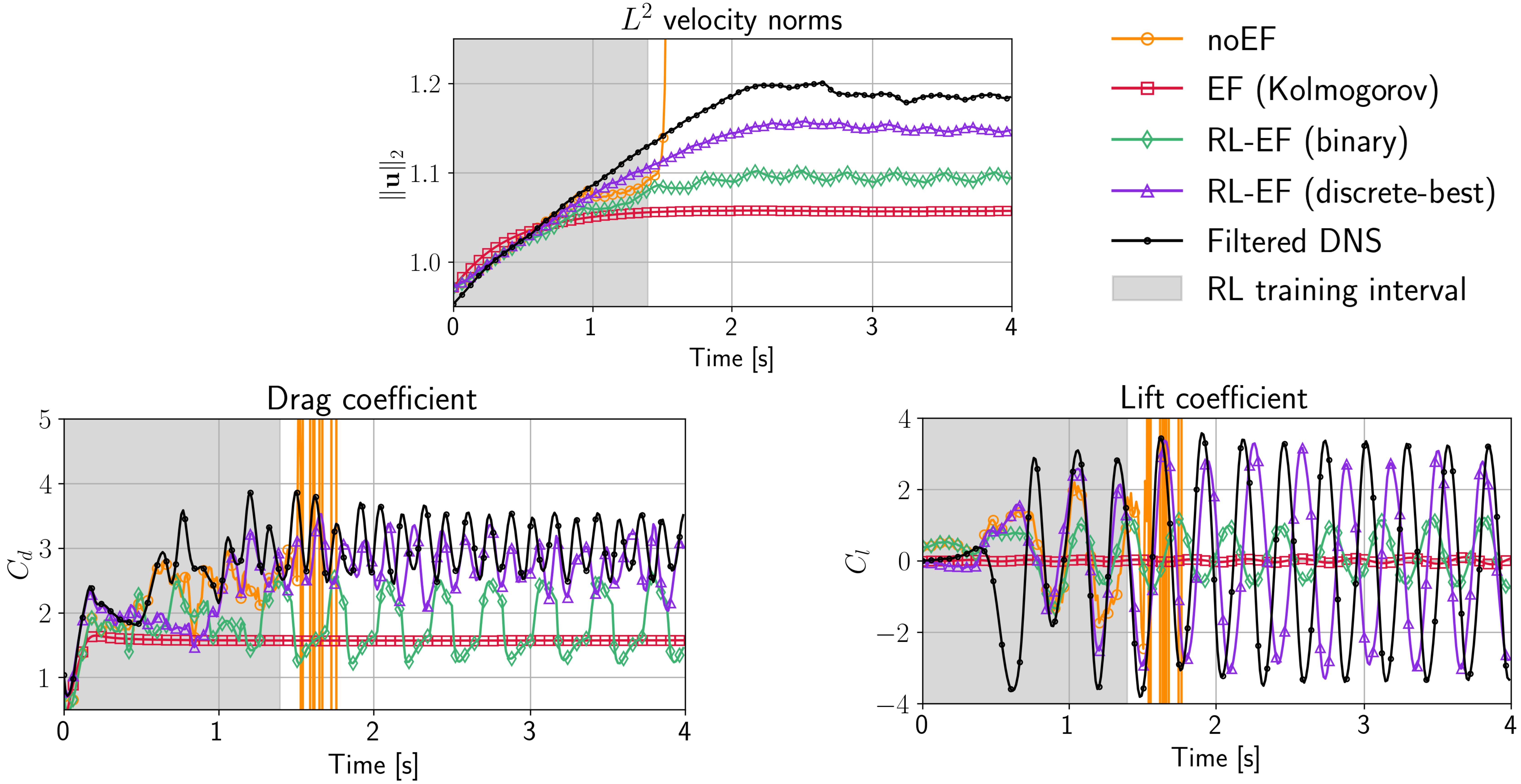

## Discrete (multiple) action

✴ ***Action space*** $\mathscr{A} = \{10^{-10}, 10^{-9}, \ldots, 10^{-3}\}$

## Numerical results




A. Ivagnes, M. Strazzullo and G. Rozza. *Reinforcement learning for the Evolve-Filter approach*, in preparation.

# Recent activities

1. **Nonlinear approaches for reduced order methods**

   *with*
   ***Giovanni Stabile*** *(Sant'Anna),* ***Gianluigi Rozza*** *(SISSA)*

2. **Reinforcement learning for filters**

   *with*
   ***Maria Strazzullo*** *(PoliTo),* ***Gianluigi Rozza*** *(SISSA)*

3. **Data-driven structure-preserving filtering approach**

   *with*
   ***Toby van Gastelen, Syver Agdestein, Benjamin Sanderse*** *(CWI),*
   ***Giovanni Stabile*** *(Sant'Anna),* ***Gianluigi Rozza*** *(SISSA)*

## What if we modify the filter?

***DD-EFR:***

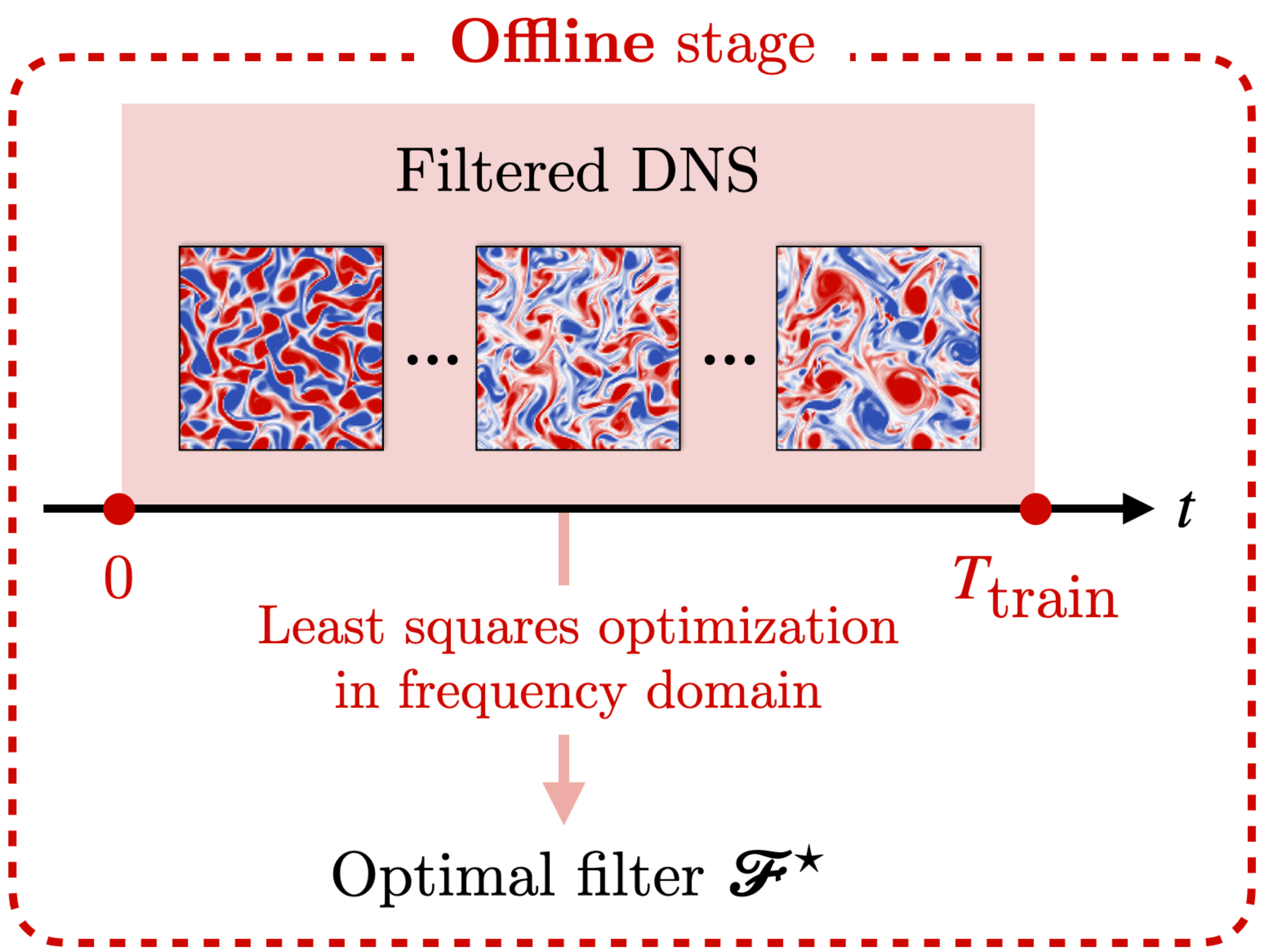
**(I)** *Evolve* :
$$\begin{cases} \frac{\boldsymbol{w}^{n+1}-\boldsymbol{u}^n}{\Delta t} + (\boldsymbol{w}^{n+1}\cdot\nabla)\boldsymbol{w}^{n+1} - \nu\Delta\boldsymbol{w}^{n+1} + \nabla p^{n+1} = 0 & \text{in } \Omega\times\{t_{n+1}\}, \\ \nabla\cdot\boldsymbol{w}^{n+1} = 0 & \text{in } \Omega\times\{t_{n+1}\}, \end{cases}$$

**(II)** *Filter:*
$$\begin{cases} \boxed{\overline{\boldsymbol{w}}^n = \mathscr{F}\boldsymbol{w}^n} & \text{in } \Omega\times\{t_{n+1}\}, \\ \overline{\boldsymbol{w}}^{n+1} = \boldsymbol{u}_D^{n+1} & \text{on } \partial\Omega_D\times\{t_{n+1}\}, \\ \frac{\partial\overline{\boldsymbol{w}}^{n+1}}{\partial\boldsymbol{n}} = 0 & \text{on } \partial\Omega_N\times\{t_{n+1}\}. \end{cases}$$

**(III)** *Relax:*
$$\boldsymbol{u}^{n+1} = (1-\chi)\,\boldsymbol{w}^{n+1} + \chi\,\overline{\boldsymbol{w}}^{n+1}\ ,$$

# *Data-driven structure-preserving filtering approach*

**Pipeline of the project**





A. Ivagnes, T. V. Gastelen, S. Agdestein, B. Sanderse, G. Stabile, G. Rozza, *A new data-driven energy-stable Evolve-Filter-Relax model for turbulent flow simulation*, arXiv: 2507.17423

# *Data-driven structure-preserving filtering approach*

## The data-driven filter

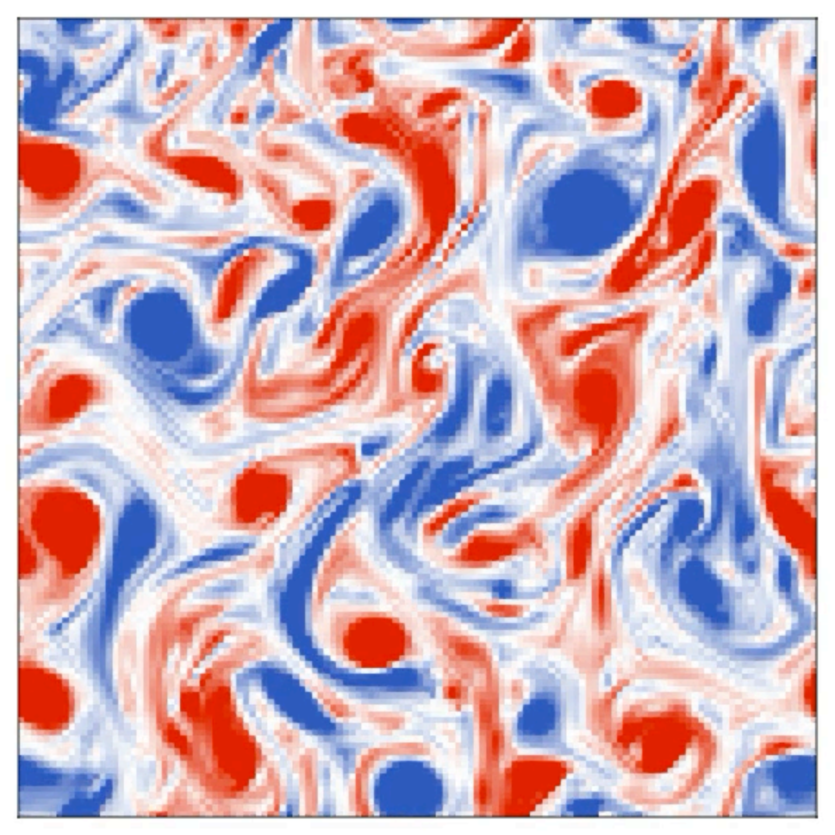
$\boldsymbol{u}_{\text{ref}}(t_i)$ : reference velocity at time step $t_i$ (filtered DNS)

$$\begin{bmatrix} | & | & & | \\ \boldsymbol{u}_{\text{ref}}(t_1) & \boldsymbol{u}_{\text{ref}}(t_2) & \cdots & \boldsymbol{u}_{\text{ref}}(T_{\text{train}}) \\ | & | & & | \end{bmatrix}$$

*Snapshots matrix*

DNS grid: $512^2$

EFR/Filtered DNS grid: $128^2$



*Filtered DNS - vorticity*

*2D decaying homogeneous turbulence test case at* $Re = 40{,}000$

Agdestein, Syver Døving, and Benjamin Sanderse. "Discretize first, filter next: Learning divergence-consistent closure models for large-eddy simulation." *Journal of Computational Physics* 522 (2025): 113577.

IncompressibleNavierStokes.jl

# *Data-driven structure-preserving filtering approach*

## The data-driven filter

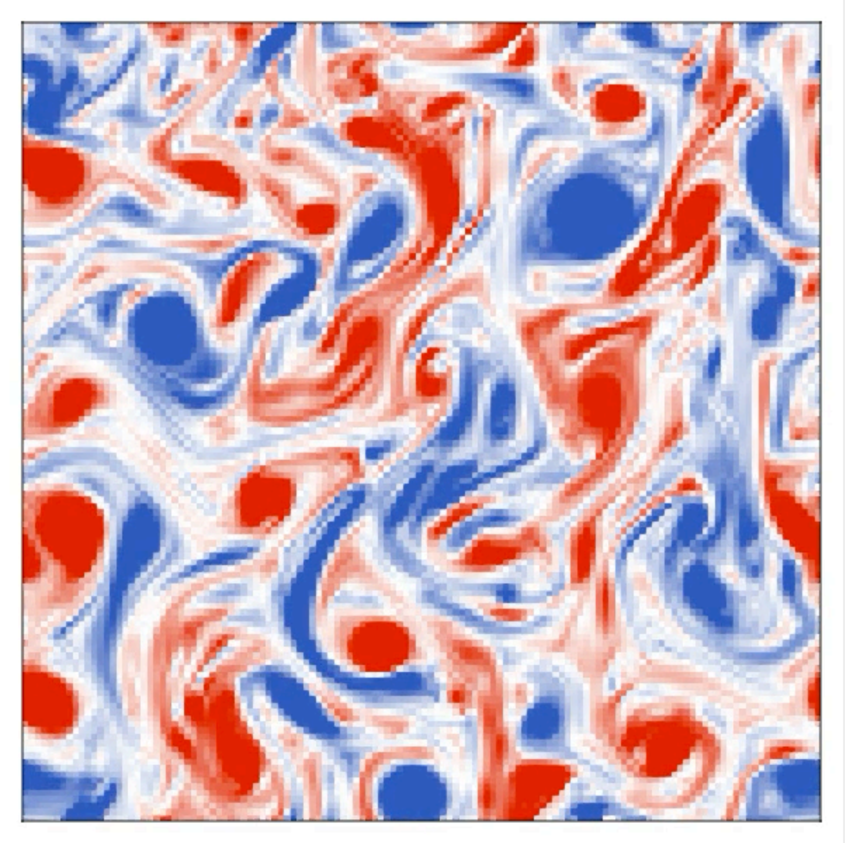
$u_{\text{ref}}(t_i)$ : reference velocity at time step $t_i$ (filtered DNS)

$$\begin{bmatrix} | & | & & | \\ u_{\text{ref}}(t_1) & u_{\text{ref}}(t_2) & \cdots & u_{\text{ref}}(T_{\text{train}}) \\ | & | & & | \end{bmatrix}$$

*Snapshots matrix*

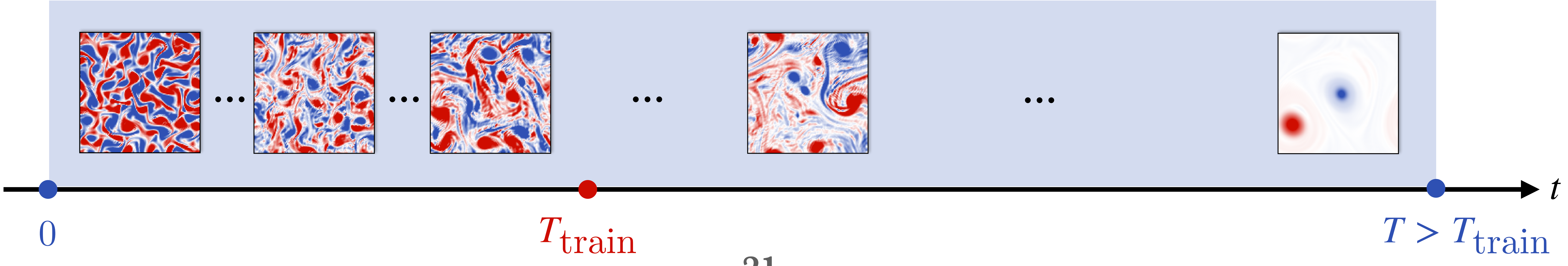
DNS grid: $512^2$

EFR/Filtered DNS grid: $128^2$



*Filtered DNS - vorticity*

*2D decaying homogeneous turbulence test case at $Re = 40{,}000$*



# *Data-driven structure-preserving filtering approach*

## The data-driven filter

$u_{\text{ref}}(t_i)$ : reference velocity at time step $t_i$ (filtered DNS)

$$\begin{bmatrix} | & | & & | \\ u_{\text{ref}}(t_1) & u_{\text{ref}}(t_2) & \dots & u_{\text{ref}}(T_{\text{train}}) \\ | & | & & | \end{bmatrix}$$

*Evolve for 1 time step*

$$\longrightarrow \quad W^{\text{true}} = \begin{bmatrix} | & | & & | \\ w(t_2) & w(t_3) & \dots & w(T_{\text{train}}) \\ | & | & & | \end{bmatrix}$$

# *Data-driven structure-preserving filtering approach*

## The data-driven filter

$\boldsymbol{u}_{\text{ref}}(t_i)$ : reference velocity at time step $t_i$ (filtered DNS)

$$\begin{bmatrix} | & | & & | \\ \boldsymbol{u}_{\text{ref}}(t_1) & \boldsymbol{u}_{\text{ref}}(t_2) & \dots & \boldsymbol{u}_{\text{ref}}(T_{\text{train}}) \\ | & | & & | \end{bmatrix}$$

*Evolve for 1 time step*

$$W^{\text{true}} = \begin{bmatrix} | & | & & | \\ \boldsymbol{w}(t_2) & \boldsymbol{w}(t_3) & \dots & \boldsymbol{w}(T_{\text{train}}) \\ | & | & & | \end{bmatrix}$$

$$U^{\text{true}} = \begin{bmatrix} | & | & & | \\ \boldsymbol{u}_{\text{ref}}(t_2) & \boldsymbol{u}_{\text{ref}}(t_3) & \dots & \boldsymbol{u}_{\text{ref}}(T_{\text{train}}) \\ | & | & & | \end{bmatrix}$$

# *Data-driven structure-preserving filtering approach*

## The data-driven filter

$$W^{\text{true}} = \begin{bmatrix} | & | & & | \\ w(t_2) & w(t_3) & \dots & w(T_{\text{train}}) \\ | & | & & | \end{bmatrix} \qquad U^{\text{true}} = \begin{bmatrix} | & | & & | \\ u_{\text{ref}}(t_2) & u_{\text{ref}}(t_3) & \dots & u_{\text{ref}}(T_{\text{train}}) \\ | & | & & | \end{bmatrix}$$

***Goal***: find the filter $\mathscr{F}$ that minimizes $\mathscr{L} = \|\mathscr{F} W^{\text{true}} - U^{\text{true}}\|_F^2$

# *Data-driven structure-preserving filtering approach*

## The data-driven filter

$$W^{\text{true}} = \begin{bmatrix} | & | & & | \\ w(t_2) & w(t_3) & \dots & w(T_{\text{train}}) \\ | & | & & | \end{bmatrix} \qquad U^{\text{true}} = \begin{bmatrix} | & | & & | \\ u_{\text{ref}}(t_2) & u_{\text{ref}}(t_3) & \dots & u_{\text{ref}}(T_{\text{train}}) \\ | & | & & | \end{bmatrix}$$

***Goal***: find the filter $\mathscr{F}$ that minimizes $\mathscr{L} = \|\mathscr{F} W^{\text{true}} - U^{\text{true}}\|_F^2$

$$\Big\downarrow \textit{FFT}$$

$$\mathscr{L} = \|\hat{\mathscr{F}} \hat{W}^{\text{true}} - \hat{U}^{\text{true}}\|_F^2$$

# *Data-driven structure-preserving filtering approach*

## The data-driven filter

$$W^{\text{true}} = \begin{bmatrix} | & | & & | \\ w(t_2) & w(t_3) & \dots & w(T_{\text{train}}) \\ | & | & & | \end{bmatrix} \qquad U^{\text{true}} = \begin{bmatrix} | & | & & | \\ u_{\text{ref}}(t_2) & u_{\text{ref}}(t_3) & \dots & u_{\text{ref}}(T_{\text{train}}) \\ | & | & & | \end{bmatrix}$$

***Goal***: find the filter $\mathscr{F}$ that minimizes $\mathscr{L} = \|\mathscr{F} W^{\text{true}} - U^{\text{true}}\|_F^2$

$$\Big\downarrow \textit{FFT}$$

$$\mathscr{L} = \|\hat{\mathscr{F}} \hat{W}^{\text{true}} - \hat{U}^{\text{true}}\|_F^2$$

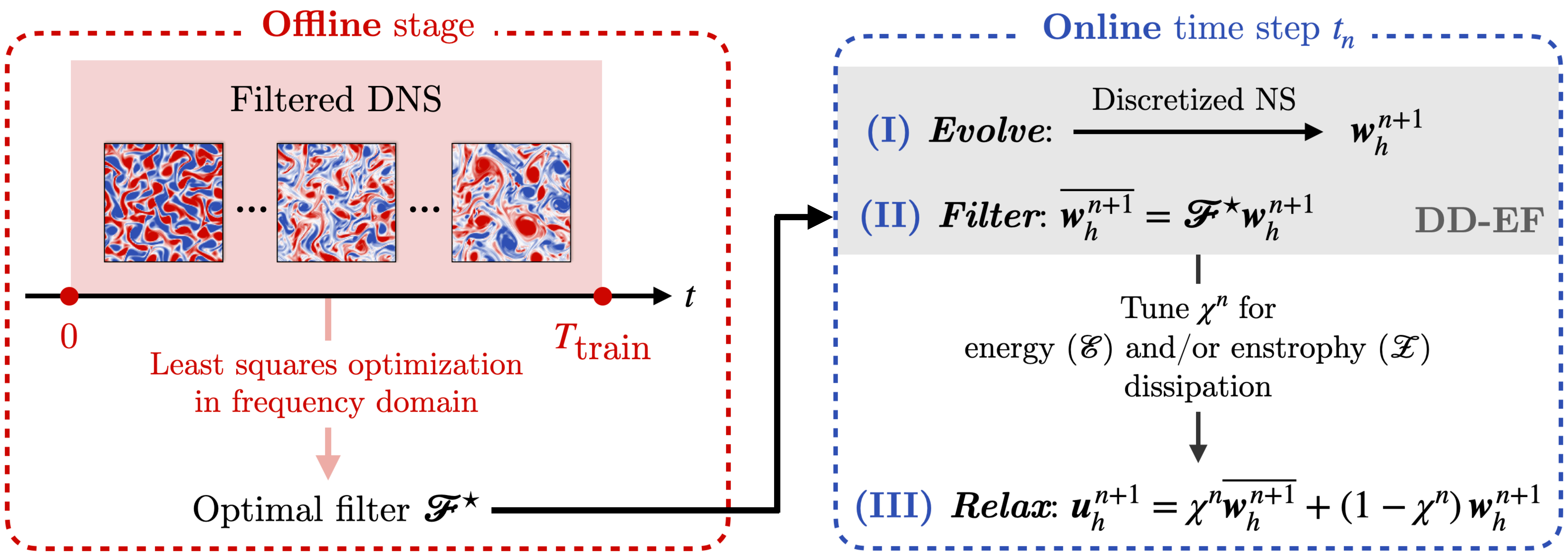
$$\Big\downarrow \; \frac{\partial \mathscr{L}}{\partial \hat{\mathscr{F}}} = 0 \textit{ with periodic BCs}$$

$$\hat{f}_i^{\star} = \frac{(\hat{W}_i^{\text{true}})^{\dagger} \hat{U}_i^{\text{true}}}{(\hat{W}_i^{\text{true}})^{\dagger} \hat{W}_i^{\text{true}}}$$

In case of periodic BCs we can compute a *diagonal filter matrix* via least-squares method.

# *Data-driven structure-preserving filtering approach*

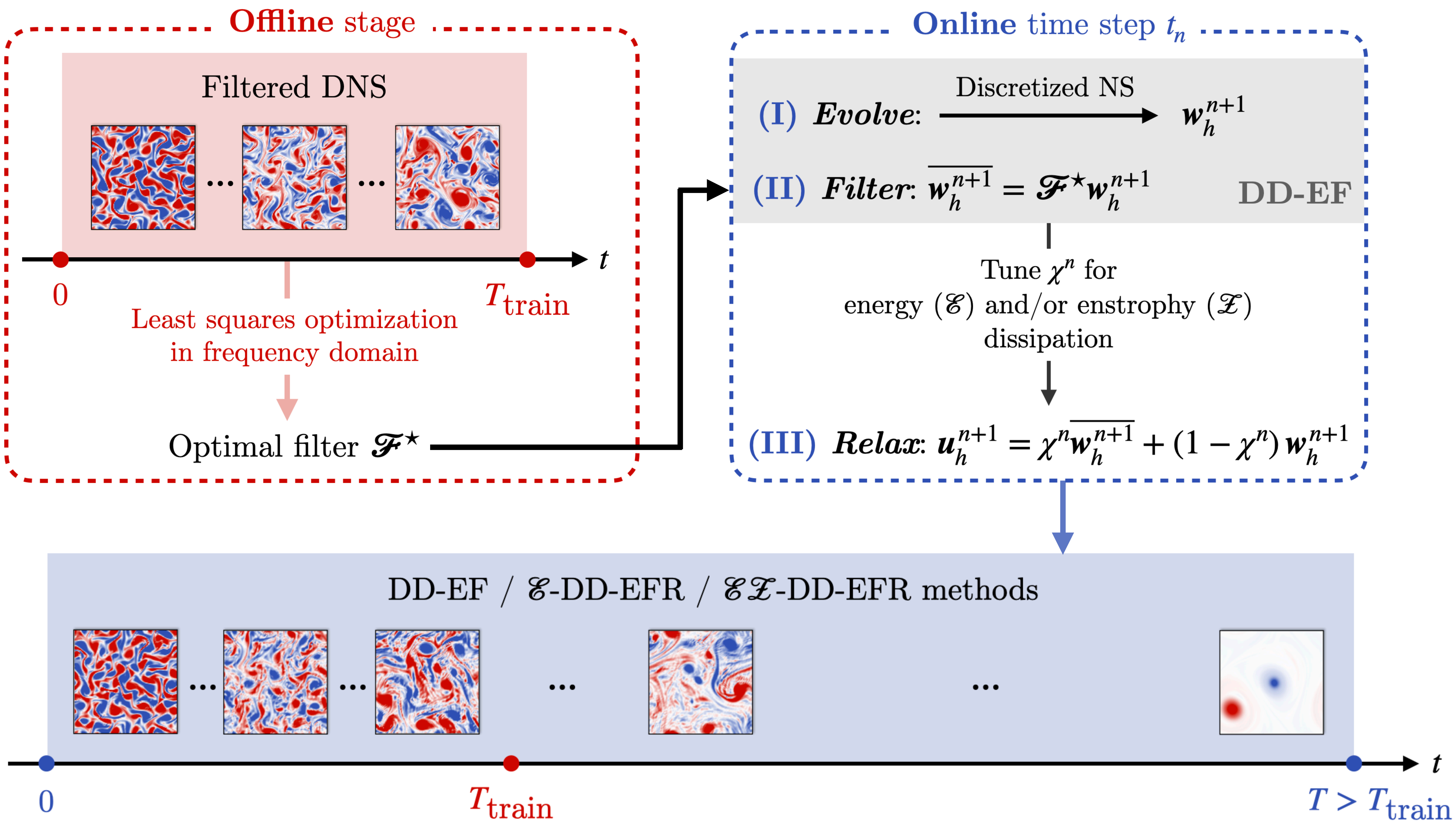
**Pipeline of the project**




A. Ivagnes, T. V. Gastelen, S. Agdestein, B. Sanderse, G. Stabile, G. Rozza, *A new data-driven energy-stable Evolve-Filter-Relax model for turbulent flow simulation*, arXiv: 2507.17423

# *Data-driven structure-preserving filtering approach*

**Pipeline of the project**

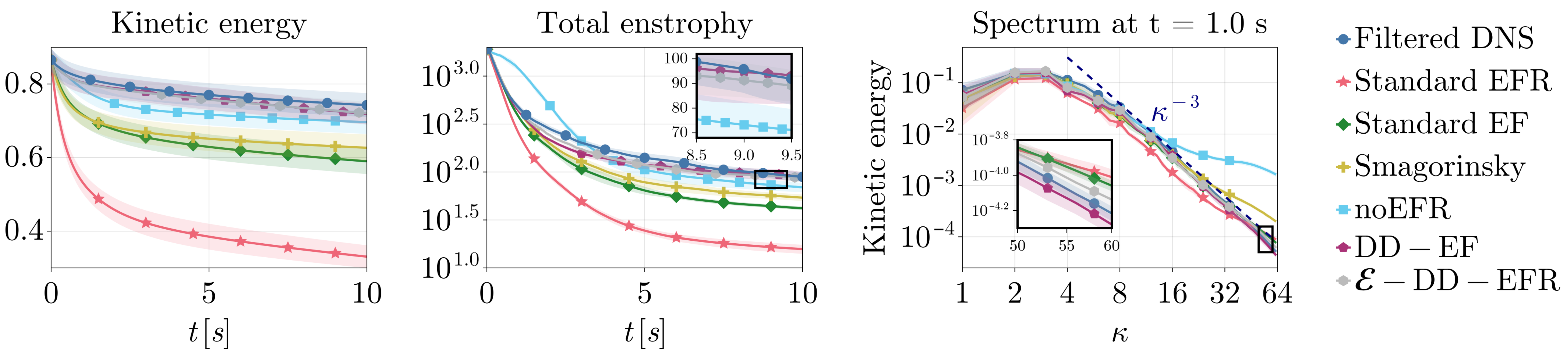



A. Ivagnes, T. V. Gastelen, S. Agdestein, B. Sanderse, G. Stabile, G. Rozza, *A new data-driven energy-stable Evolve-Filter-Relax model for turbulent flow simulation,* arXiv: 2507.17423

# Data-driven structure-preserving filtering approach

## Numerical results

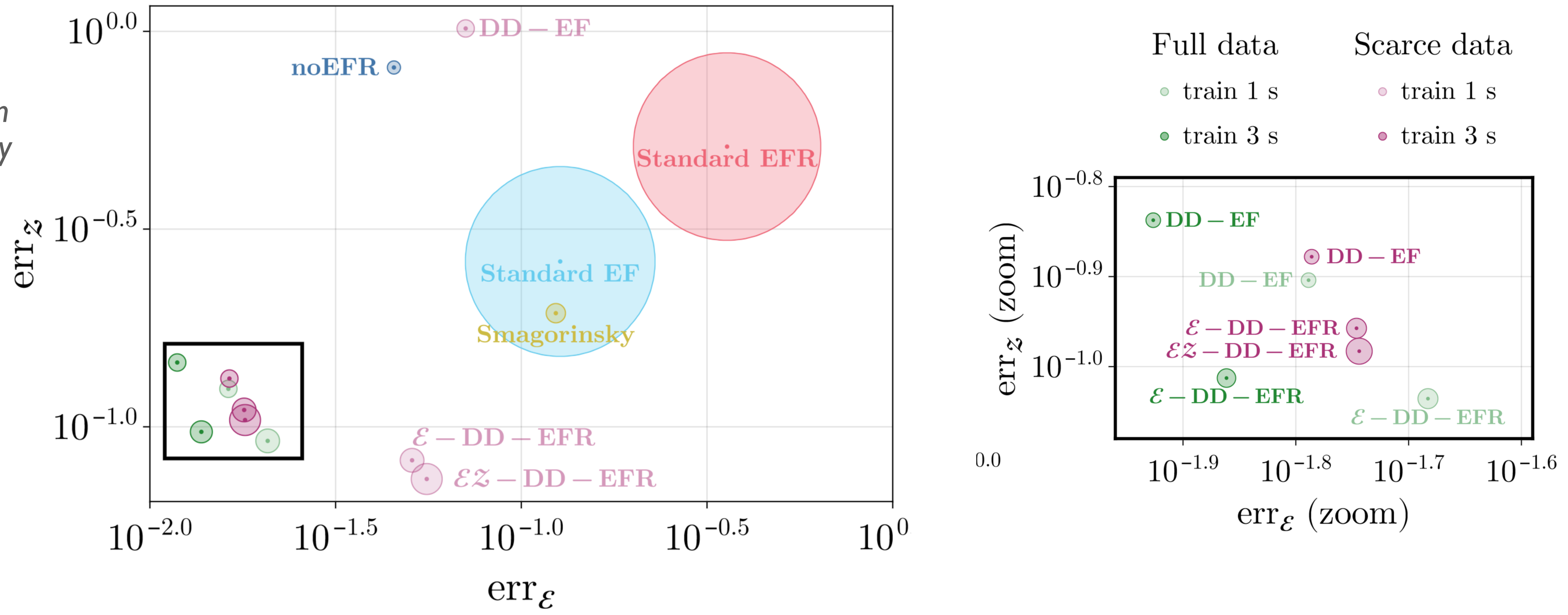
*Results* for $T_{\text{train}} = 3$ seconds




A. Ivagnes, T. V. Gastelen, S. Agdestein, B. Sanderse, G. Stabile, G. Rozza, *A new data-driven energy-stable Evolve-Filter-Relax model for turbulent flow simulation,* arXiv: 2507.17423

# About computational time

Relative error in global enstrophy




Relative error in global energy

The dimension of the circles is proportional to the online CPU time of simulations: ***DD-EF(R) wins***!

# Thank you for the attention!