Hybrid autoencoder/Galerkin-based approach for nonlinear reduced order modelling

Séminaire EDP - Strasbourg, 9/12/25

Camilla Fiorini, Nicolas Lepage, Iraj Mortazavi - M2N CNAM

Samir Beneddine, Denis Sipp - ONERA

Nicolas Thome - Sorbonne Université

Reduced order models

The aim of reduced order models (ROMs) is to **reduce the computational cost** while maintaining an acceptable **level of accuracy**, by reducing the dimensionality of the problem.

We consider a partial differential equation (PDE):

$$\mathcal{L}(u;\vartheta) = 0$$

differential operator

state variable

parameters

We suppose to have data from previous simulations, the so-called **snapshots**, $u_1, u_2, \ldots, u_{N_S}$

such that:

$$\mathcal{L}_N(u_i;\vartheta_i) = 0 \quad i = 1, \ldots, N_S$$

discrete version of the differential operator

and we want to leverage these to **predict** the solution for a **new value of the parameter**.

Reduced order models

We are interested in reconstructing the solution manifold $\mathcal{M} = \{u(\,\cdot\,;\vartheta),\ \vartheta \in \Theta\}$.

If the manifold is *nice enough* (i. e. the dependence of the solution on the parameter is smooth), **linear methods** can be used: we look for a **finite-dimensional linear space** $\mathcal{V}_{N_R}$ that approximates the manifold $\mathcal{M}$.

A classical technique is the proper orthogonal decomposition (POD), whose ansatz is

$$u(\mathbf{x};\vartheta) \simeq \sum_{k=1}^{N_R} \alpha_k(\vartheta)\varphi_k(\mathbf{x})$$

$\mathcal{V}_{N_R}$ basis functions $\longleftrightarrow$ **offline phase**

online phase $\longleftrightarrow$ POD coefficients

Offline phase

In order to build the basis of the linear space $\mathcal{V}_{N_R}$, we collect the snapshots $u_1, u_2, \ldots, u_{N_S}$ into the following matrix:

$$S = \begin{bmatrix} \boxed{\begin{matrix} u(x_1;\vartheta_1) \\ u(x_2;\vartheta_1) \\ \vdots \\ u(x_N;\vartheta_1) \end{matrix}} & \boxed{\begin{matrix} u(x_1;\vartheta_2) \\ u(x_2;\vartheta_2) \\ \vdots \\ u(x_N;\vartheta_2) \end{matrix}} & \begin{matrix} \cdots \\ \cdots \\ \\ \cdots \end{matrix} & \boxed{\begin{matrix} u(x_1;\vartheta_{N_s}) \\ u(x_2;\vartheta_{N_s}) \\ \vdots \\ u(x_N;\vartheta_{N_s}) \end{matrix}} \end{bmatrix} \updownarrow \text{degrees of freedom}$$

snapshot 1 snapshot 2 snapshot N_s

and we perform an SVD decomposition of this matrix. The singular-vectors obtained are the basis φ_i of the *best* linear approximation space. Then one can select the first N_R, with

$$N_R \leq N_S \ll N$$

Online phase

Multiple techniques exist in order to compute the coefficients α_k.

The *real* coefficients can be computed using orthonormality of the basis functions

$$u(\mathbf{x};\vartheta) \simeq \sum_{k=1}^{N_R} \alpha_k(\vartheta)\varphi_k(\mathbf{x})$$

Online phase

Multiple techniques exist in order to compute the coefficients α_k.

The *real* coefficients can be computed using orthonormality of the basis functions

$$u(\mathbf{x};\vartheta)\cdot\varphi_j(\mathbf{x}) \simeq \sum_{k=1}^{N_R}\alpha_k(\vartheta)\varphi_k(\mathbf{x})\cdot\varphi_j(\mathbf{x})$$

Online phase

Multiple techniques exist in order to compute the coefficients α_k.

The *real* coefficients can be computed using orthonormality of the basis functions

$$u(\mathbf{x};\vartheta)\cdot\varphi_j(\mathbf{x}) \simeq \alpha_j(\vartheta)$$

Online phase

Multiple techniques exist in order to compute the coefficients α_k.

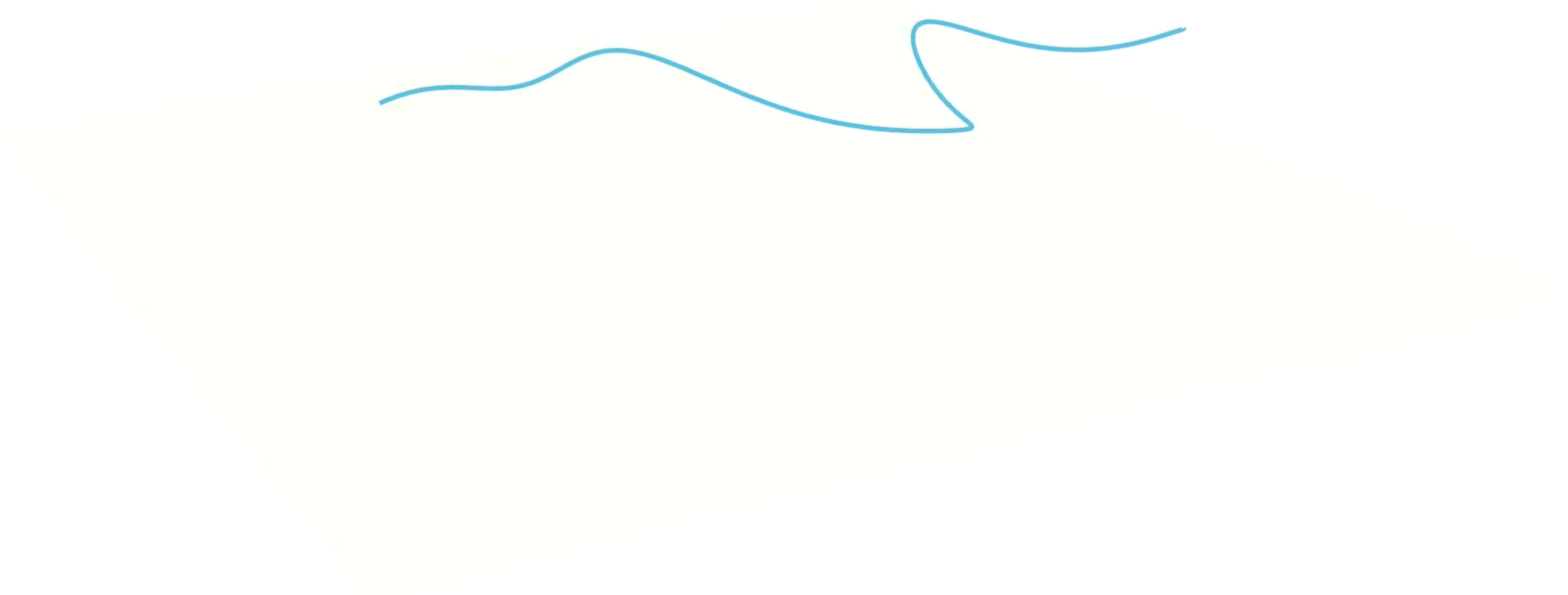
The *real* coefficients can be computed using orthonormality of the basis functions

$$u(\mathbf{x};\vartheta) \cdot \varphi_j(\mathbf{x}) \simeq \alpha_j(\vartheta)$$

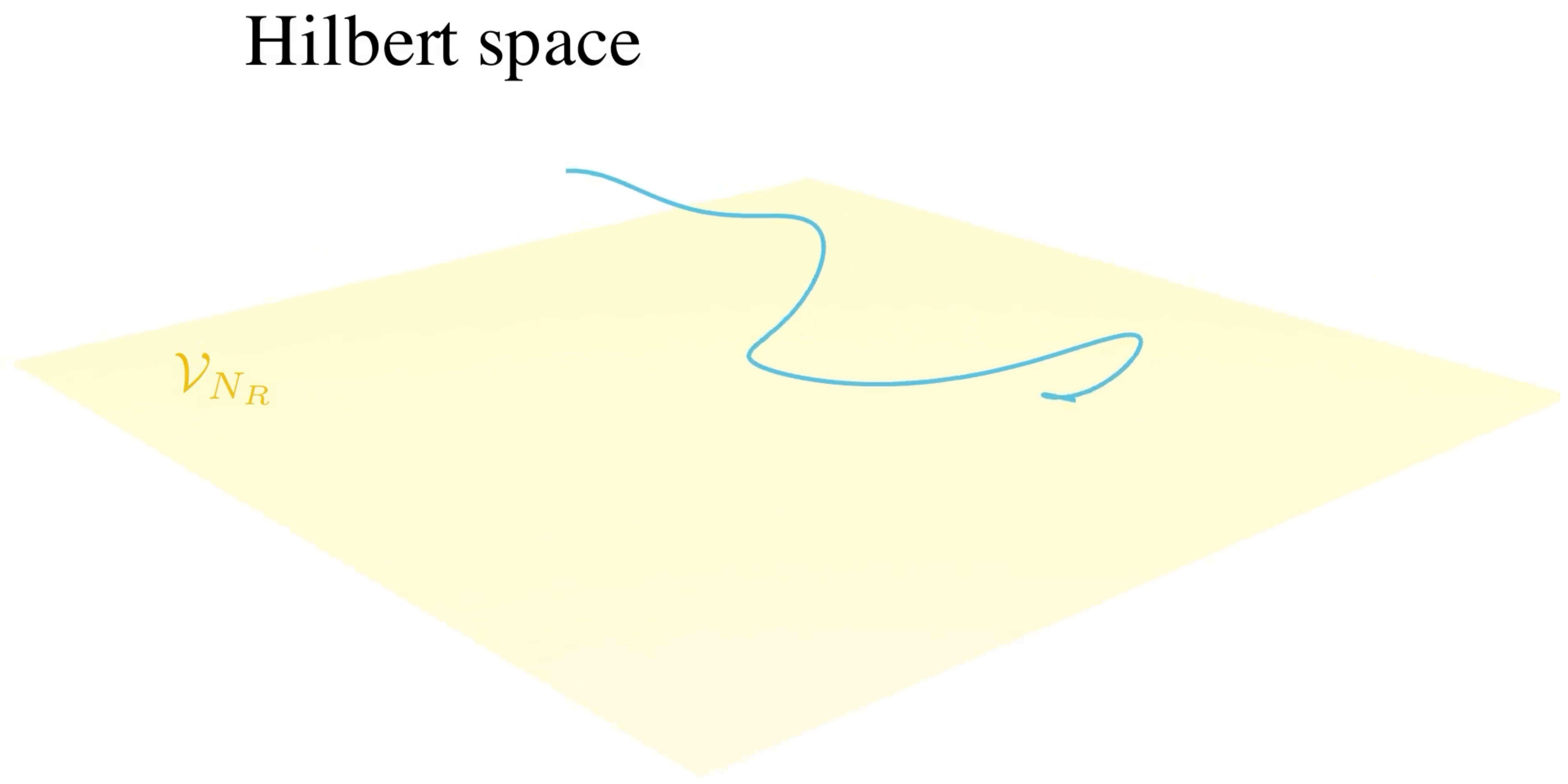
This is of course not an operational definition, since one would need the exact solution to compute the coefficients. However, it gives an intuition as to why one way to compute the coefficient is to **project the PDE onto the reduced space** $\mathcal{V}_{N_R}$ and then solve the obtained reduced PDE. This introduces an additional error.

Different kinds of errors

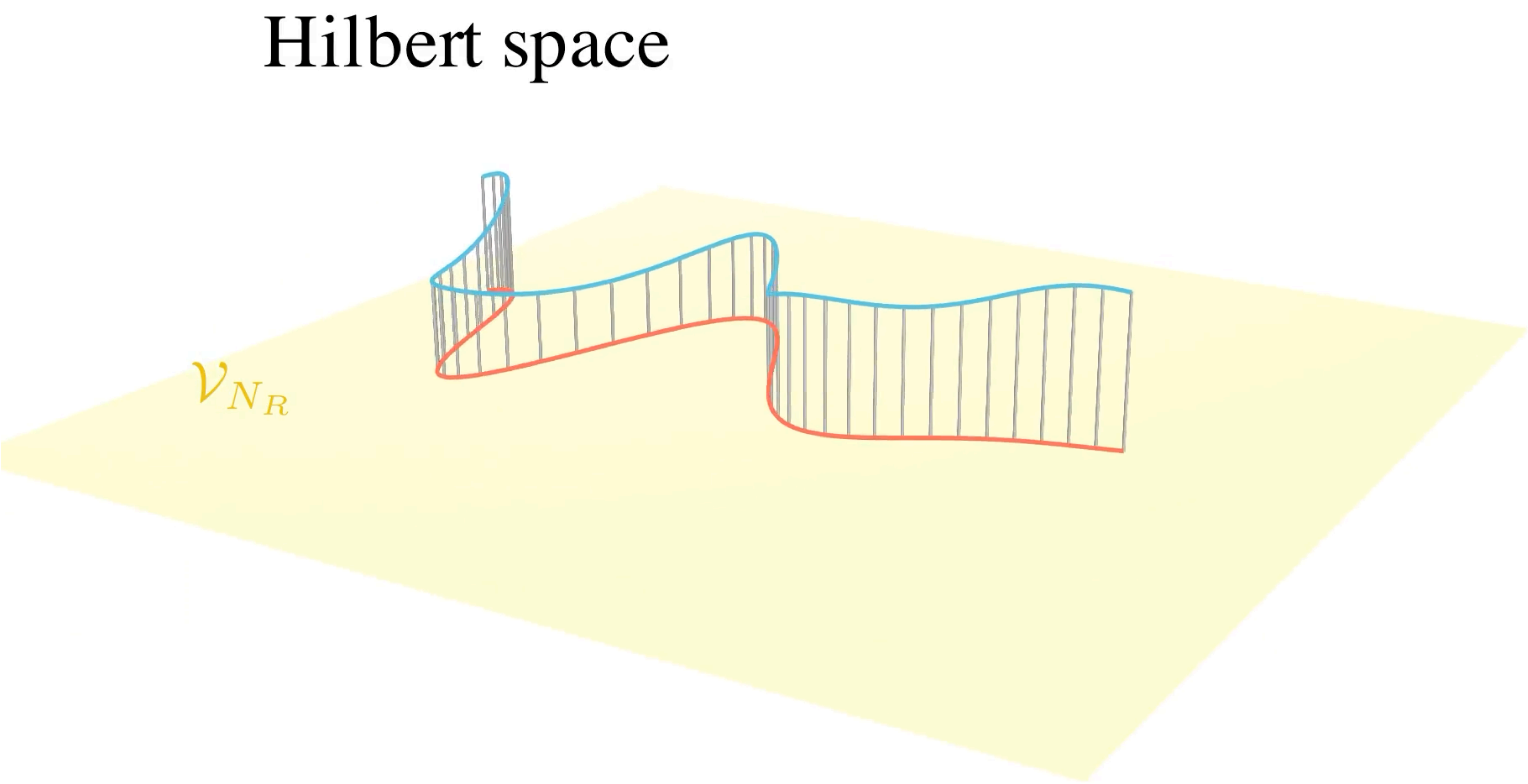
Hilbert space



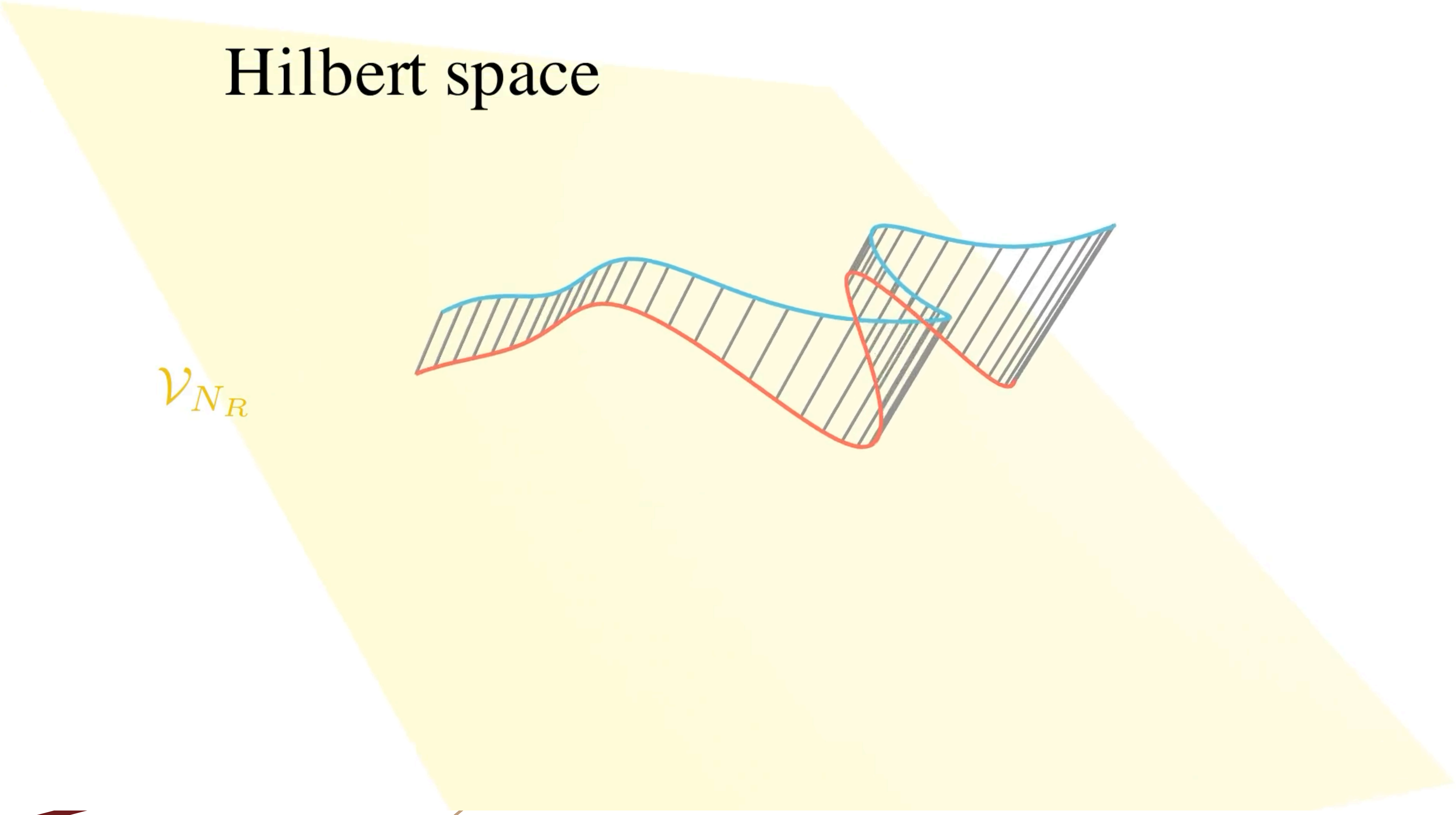
Different kinds of errors



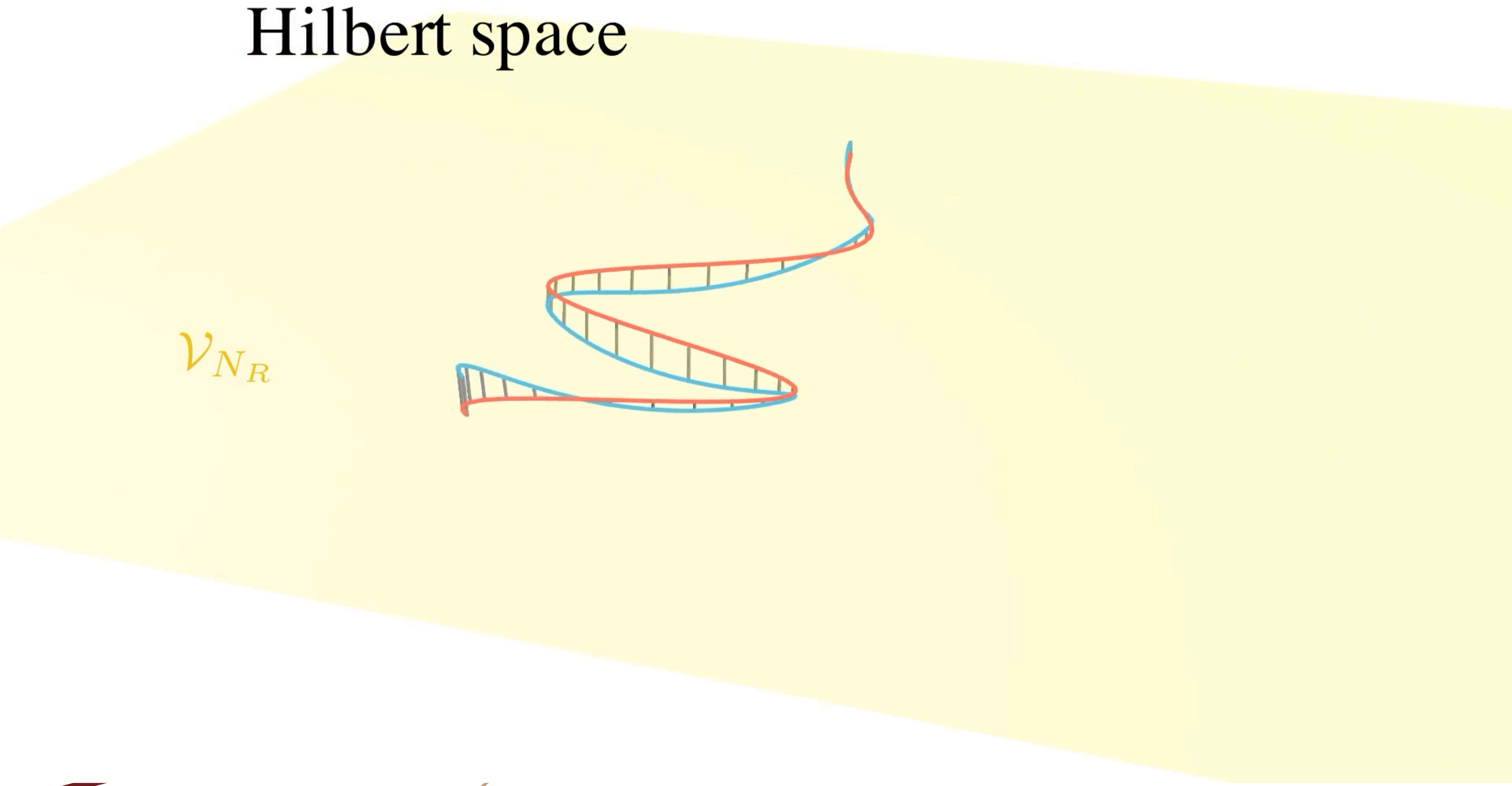
Different kinds of errors



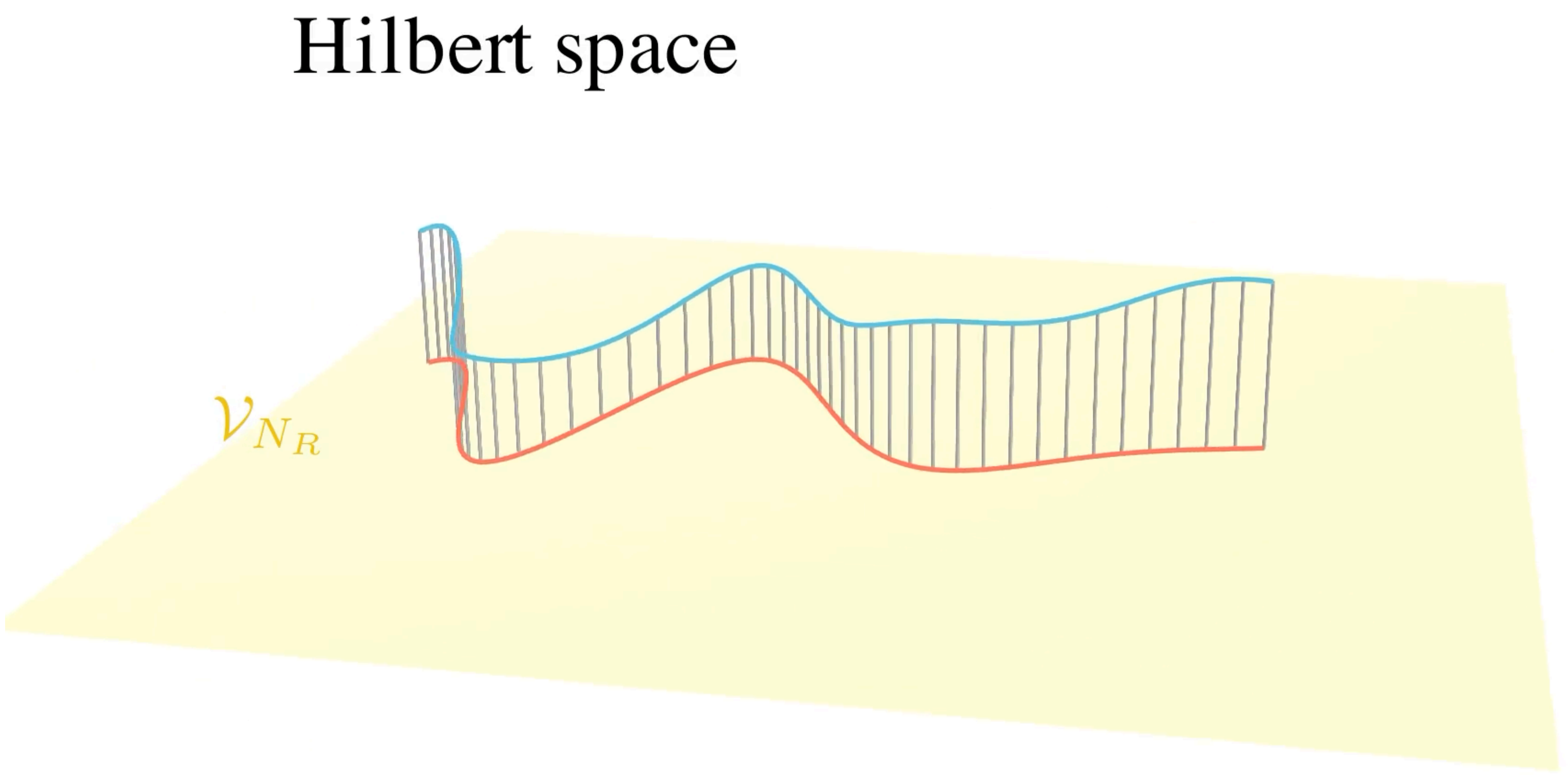
Different kinds of errors



Different kinds of errors

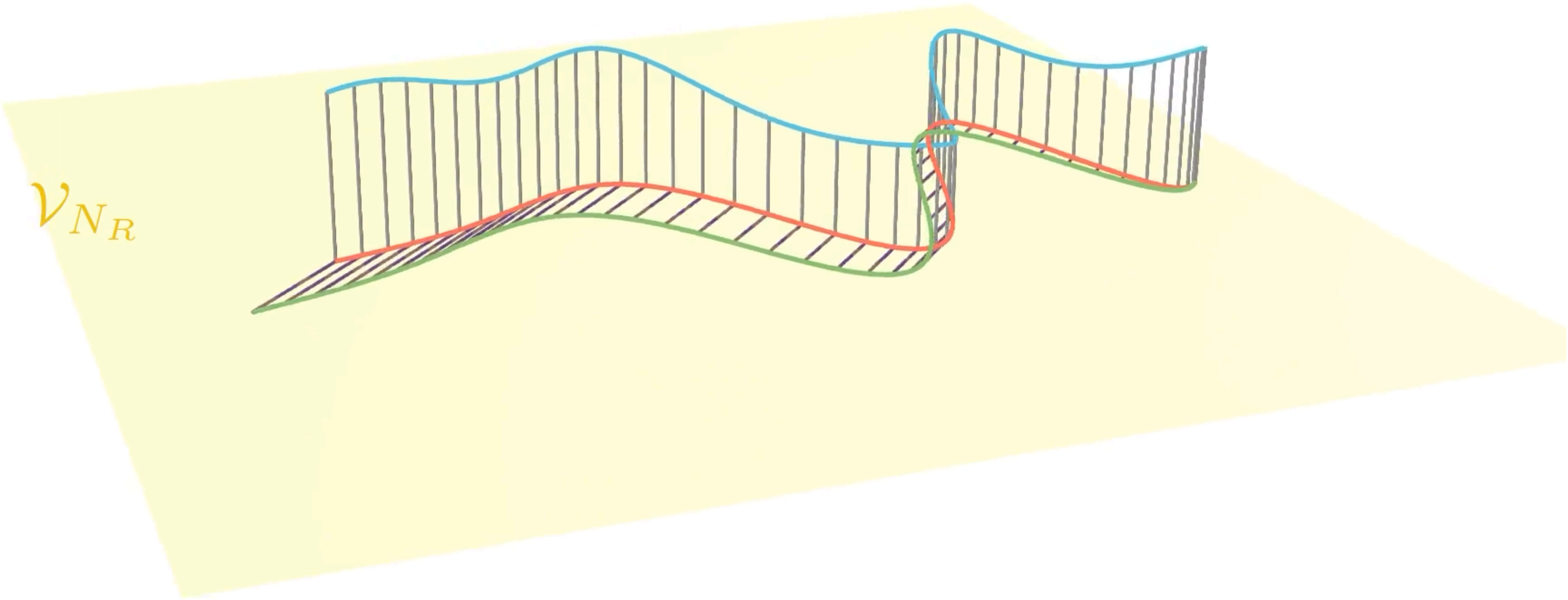


Different kinds of errors

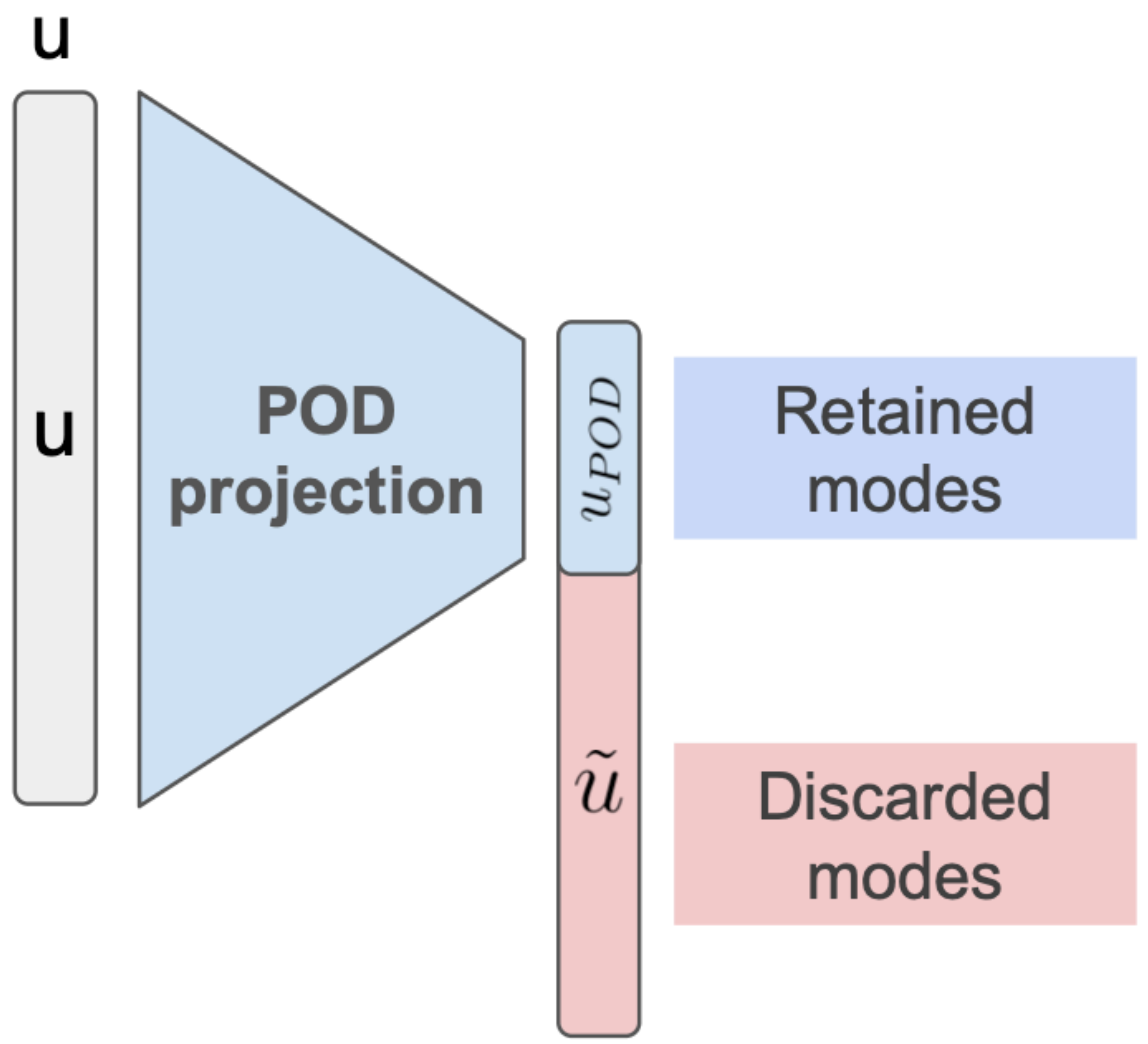


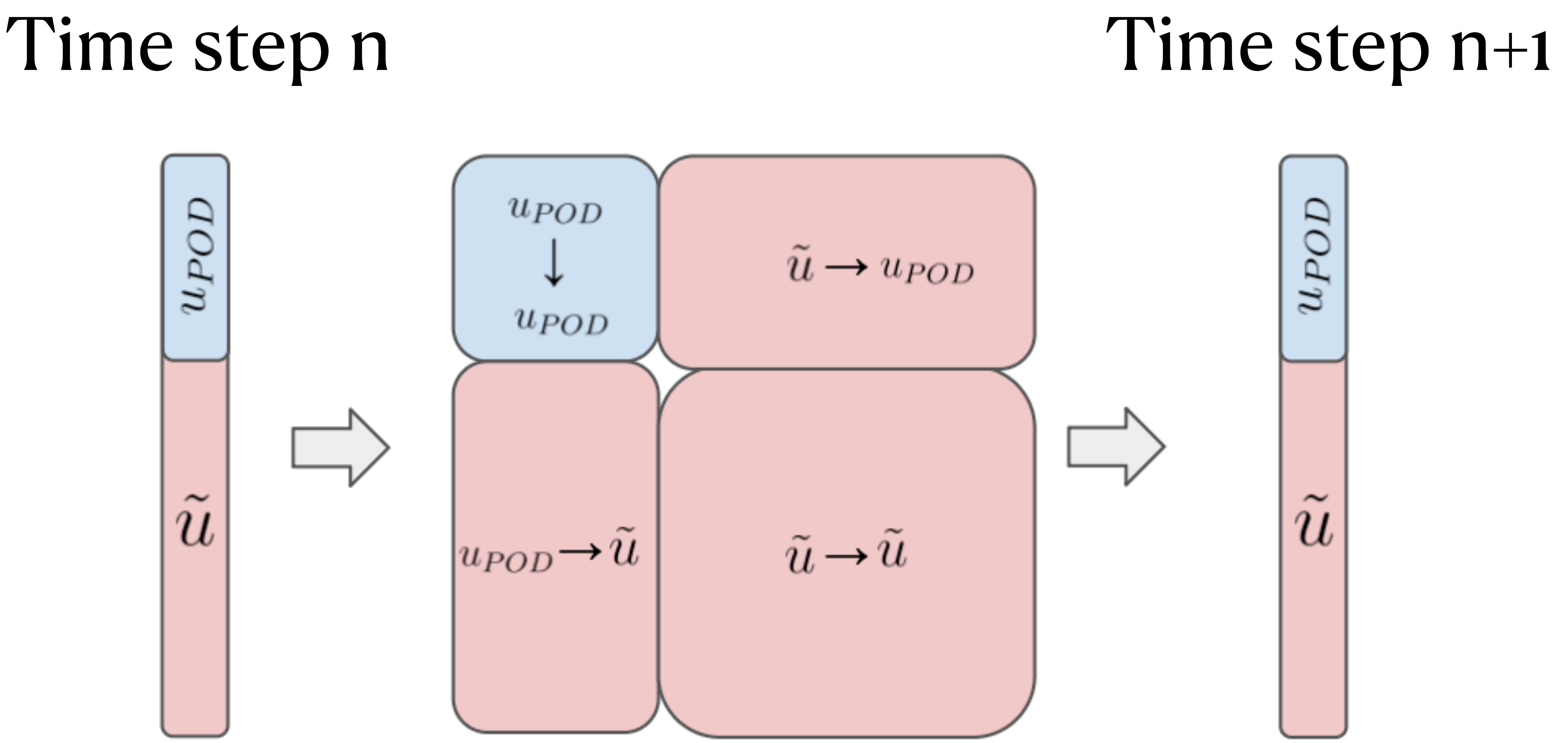
Different kinds of errors

Hilbert space

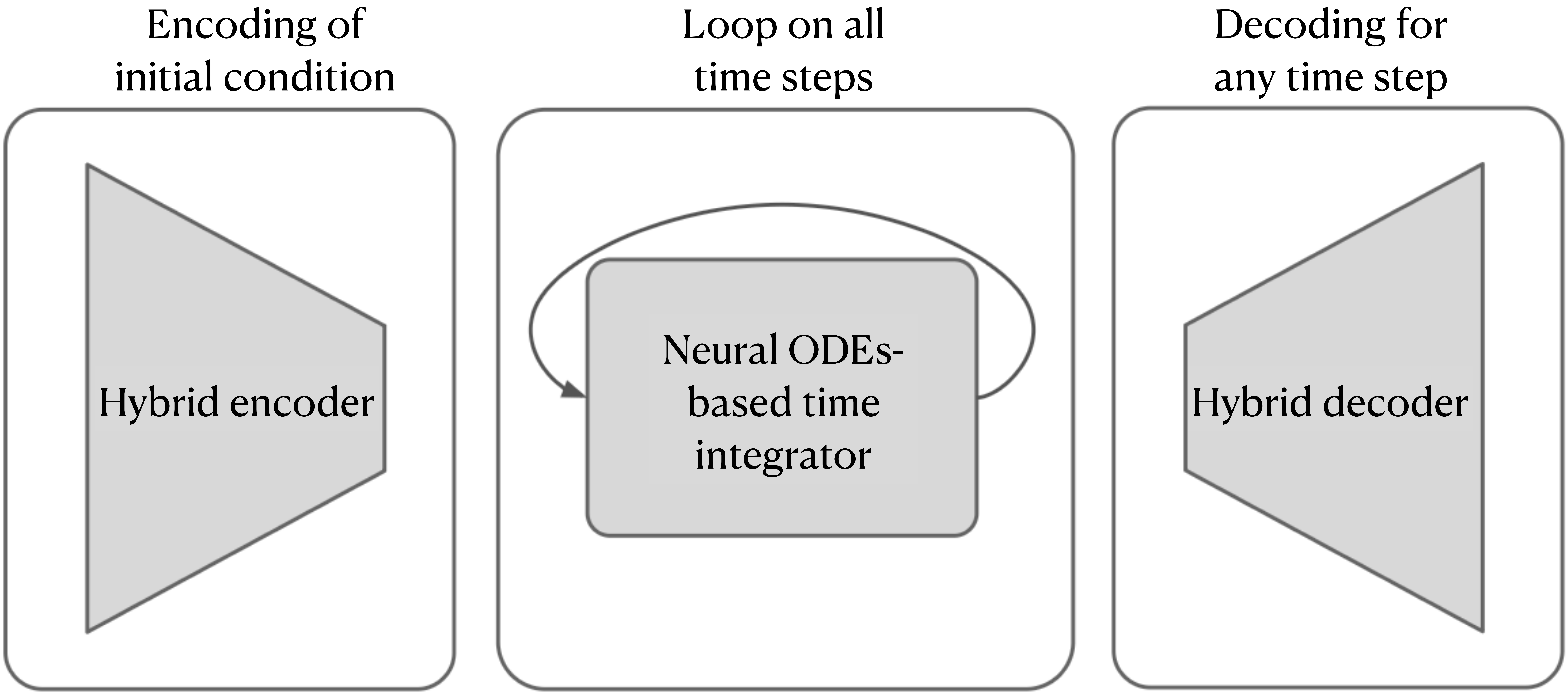


Different kinds of errors



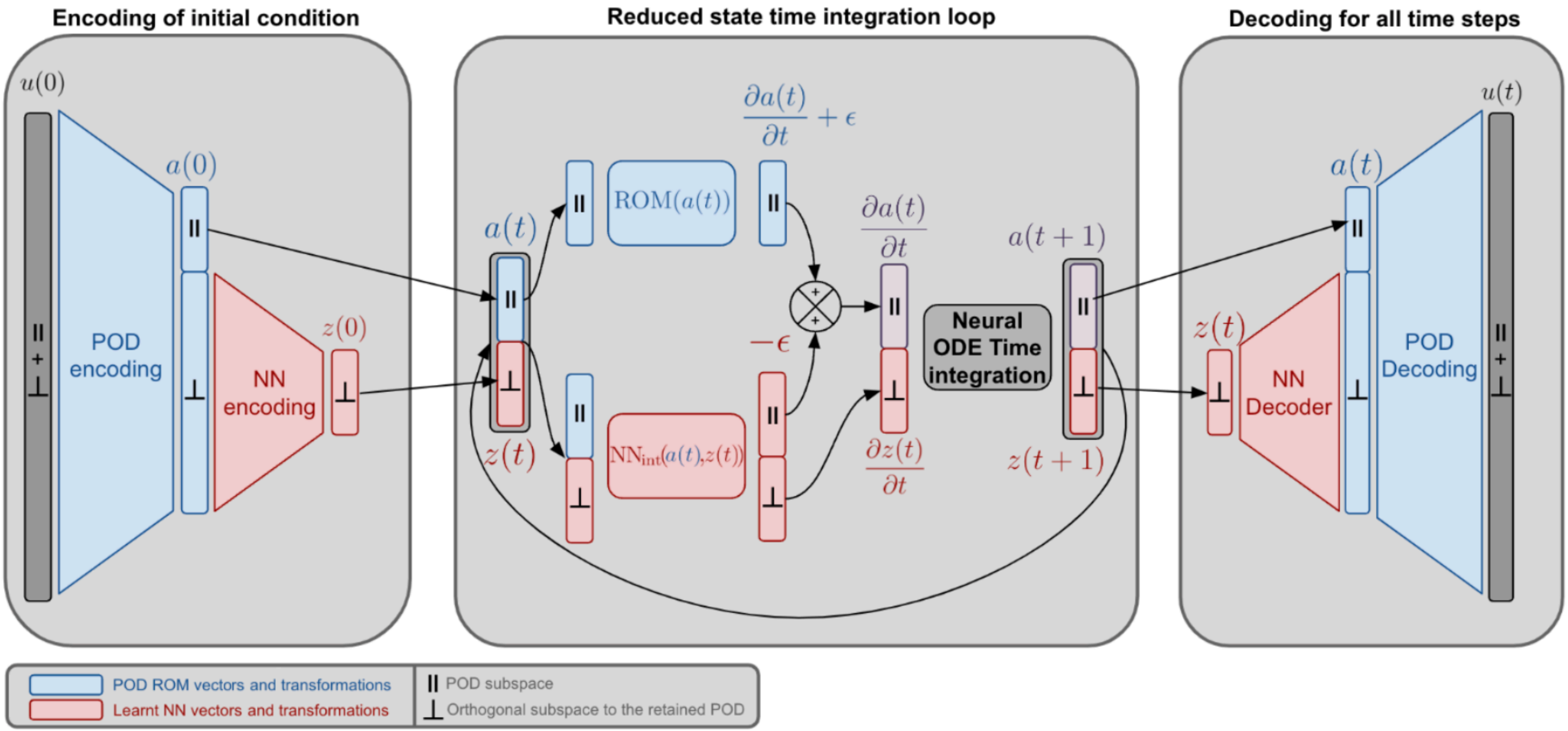


Our model



Encoding of
initial condition
Hybrid encoder
Loop on all
time steps
Neural ODEs-
based time
integrator
Decoding for
any time step
Hybrid decoder

Our model



Encoding of initial condition
Reduced state time integration loop
Decoding for all time steps
$u(0)$
$a(0)$
POD encoding
NN encoding
$z(0)$
$a(t)$
$z(t)$
ROM$(a(t))$
NN$_{\mathrm{int}}(a(t), z(t))$
$\frac{\partial a(t)}{\partial t} + \epsilon$
$-\epsilon$
$\frac{\partial a(t)}{\partial t}$
$\frac{\partial z(t)}{\partial t}$
Neural ODE Time integration
$a(t+1)$
$z(t+1)$
$u(t)$
NN Decoder
POD Decoding
POD ROM vectors and transformations
Learnt NN vectors and transformations
POD subspace
Orthogonal subspace to the retained POD

Numerical results—Burgers' equation

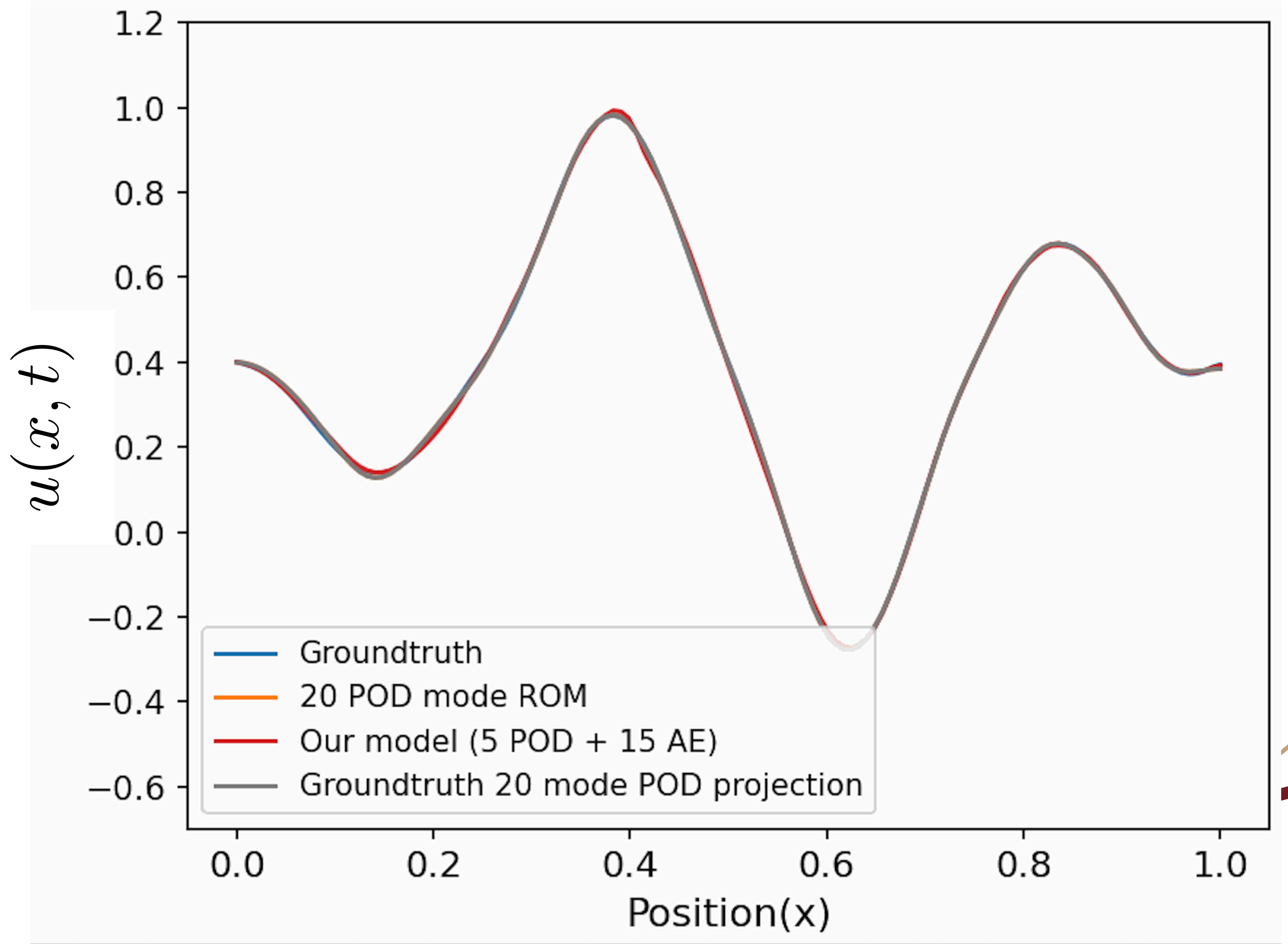
We first tested our model on the viscous Burgers' equation:

$$\begin{cases} \partial_t u - \nu \partial_{xx} u + u \partial_x u = 0 & t > 0, x \in (0,1) \\ u(0,t) = c & t > 0, x = 0 \\ \partial_x u(1,t) = 0 & t > 0, x = 1 \\ u(x,0) = g(x) & t > 0, x \in (0,1) \end{cases}$$

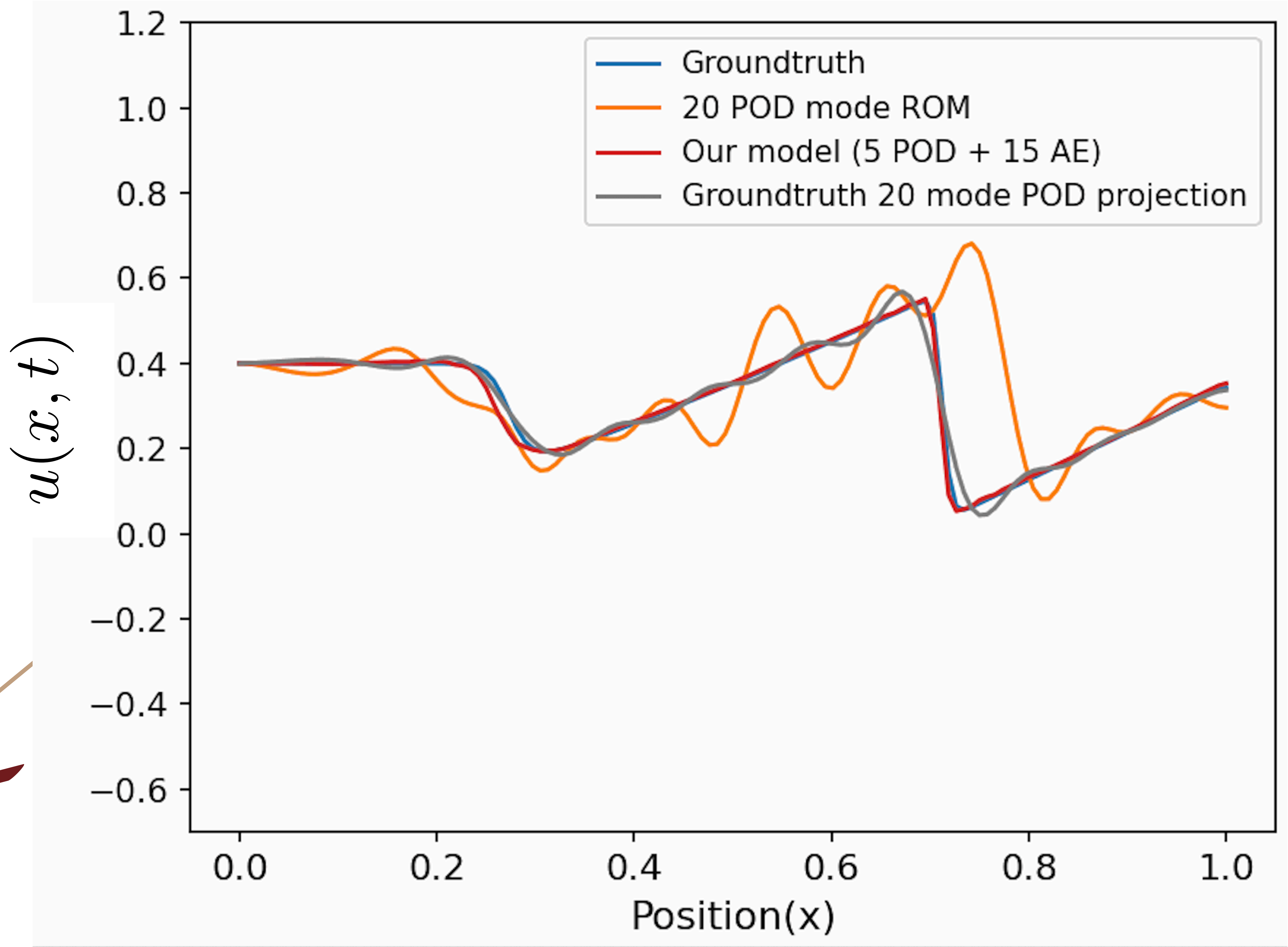
This test case offers low numerical cost while displaying highly convective and nonlinear behaviour when ν is small.

Numerical results—Burgers' equation

Initial condition

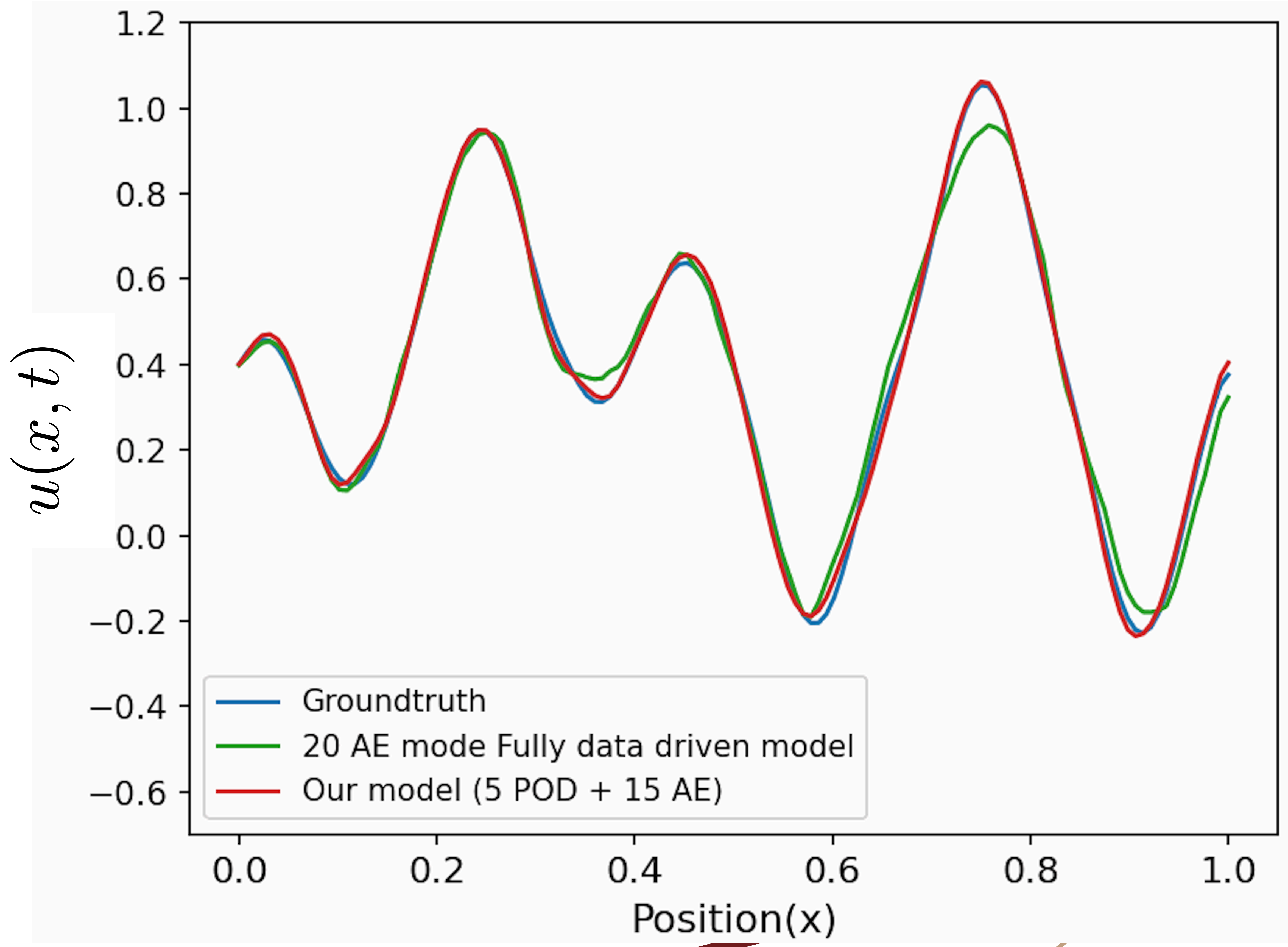


Final time

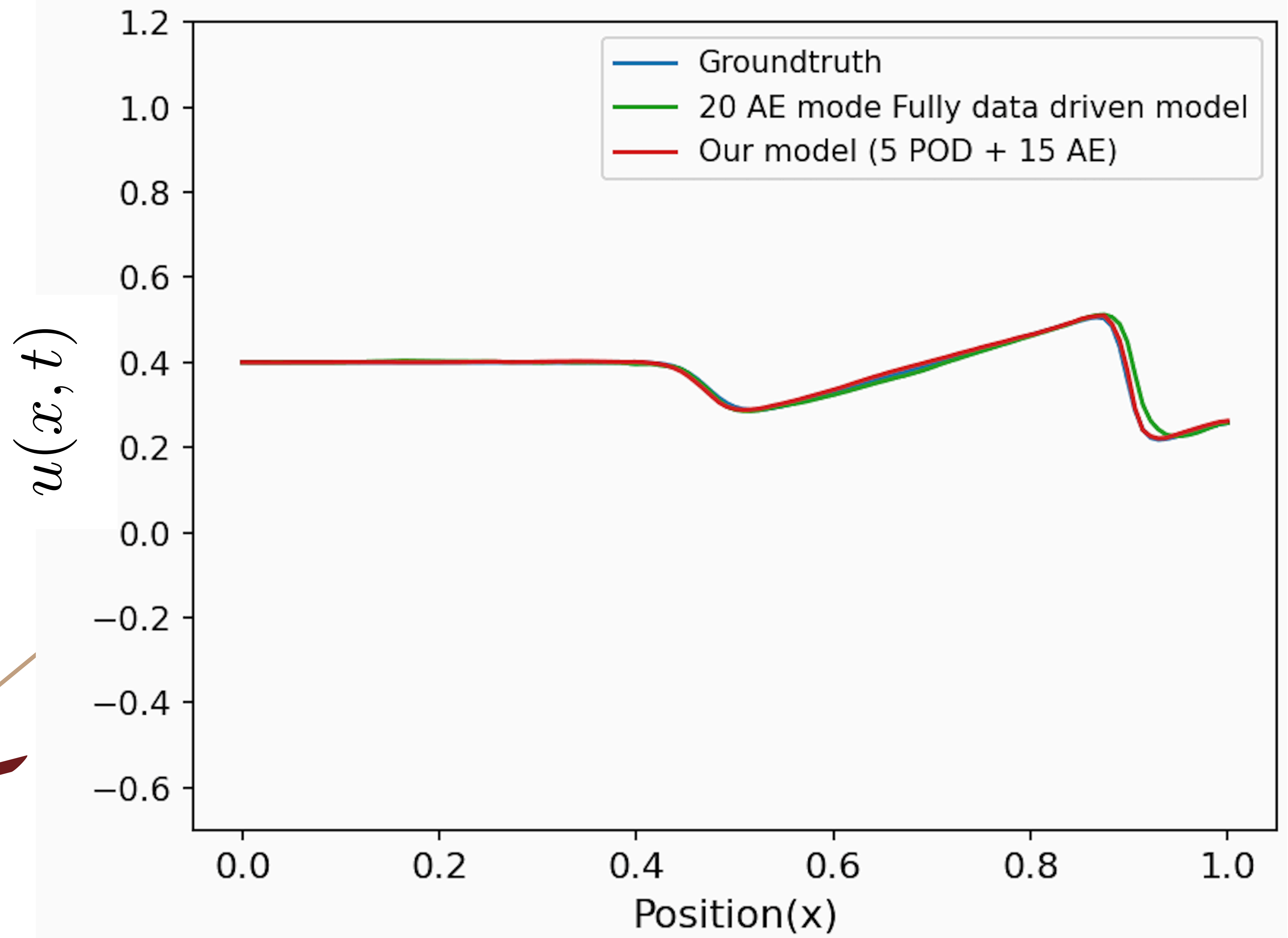


Numerical results—Burgers' equation

Initial condition



Final time



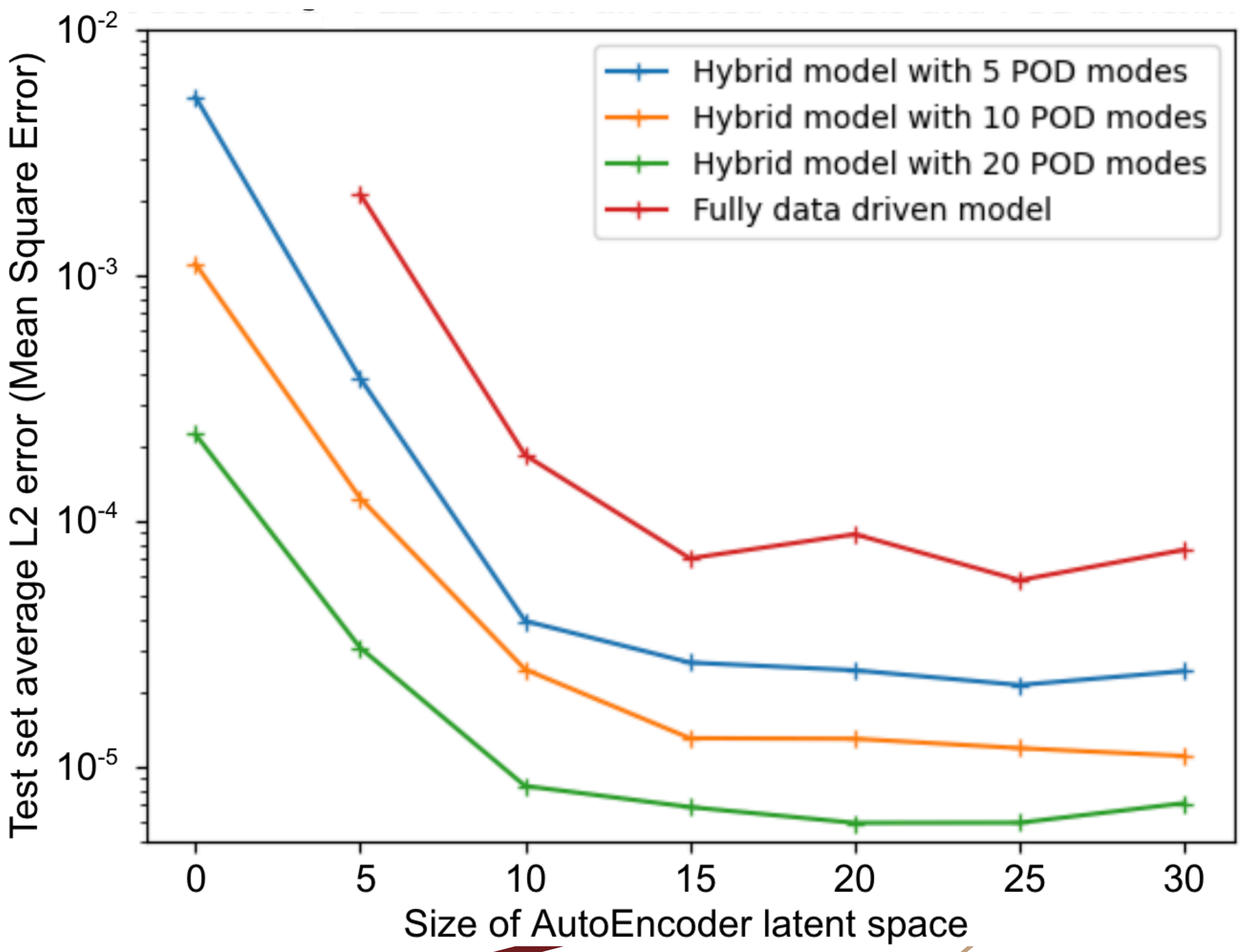
Numerical results—Burgers' equation

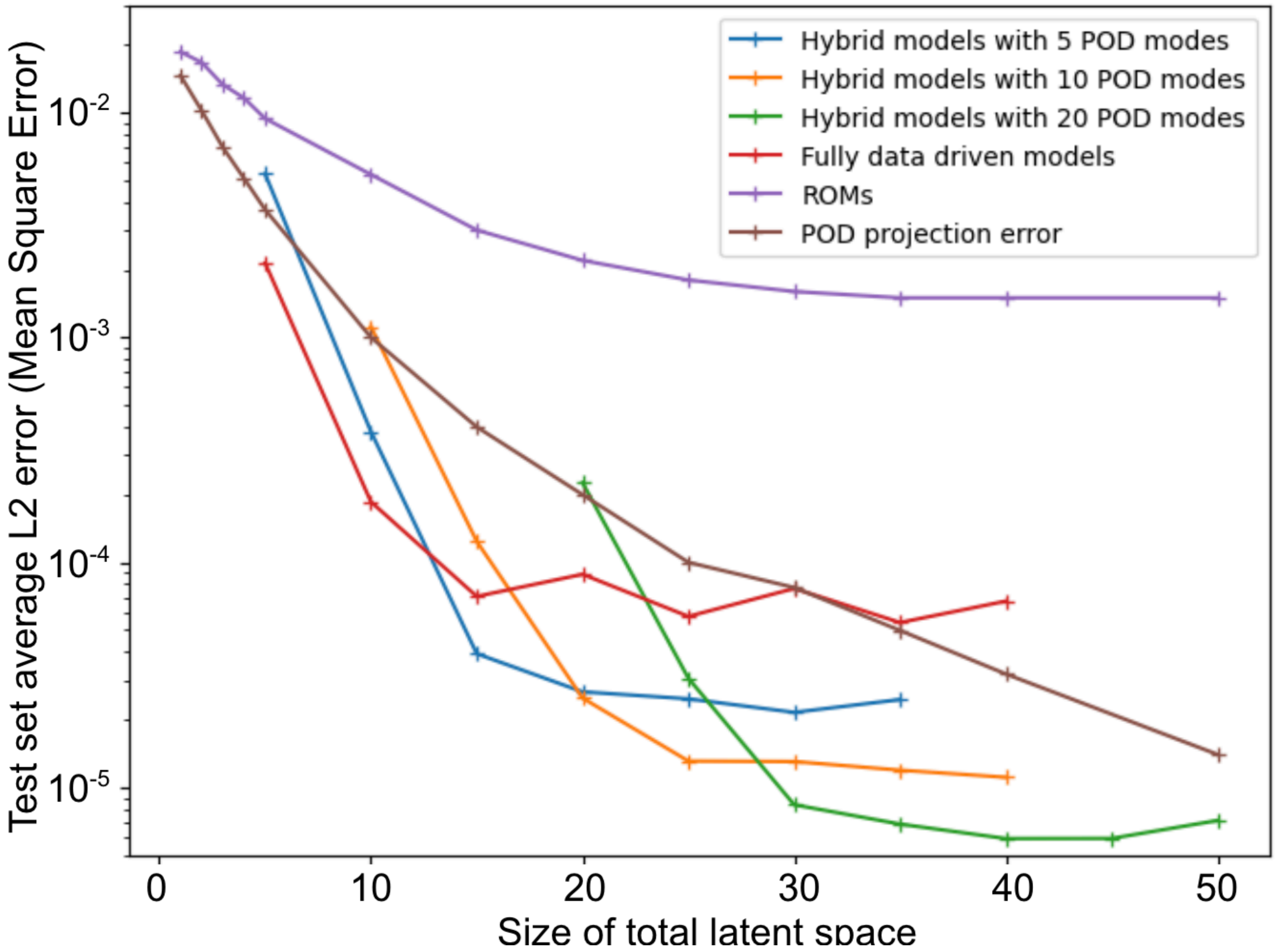
Average of the L^2 error over the whole test set.

Model	Test-set MSE
Hybrid model 20 POD + 10 AE modes	$\mathbf{8.364 \times 10^{-6}}$
Hybrid model 10 POD + 20 AE modes	1.302×10^{-5}
Hybrid model 5 POD + 25 AE modes	2.154×10^{-5}
Linear encoding hybrid model, 30 POD + 0 AE modes	8.265×10^{-5}
Fully data-driven model 30 modes	7.655×10^{-5}
POD-Galerkin ROM 30 modes	1.600×10^{-3}
30 modes POD projection error (best case scenario for POD-based models)	7.73×10^{-5}

Numerical results—Burgers' equation

Exploration of optimal size of the reduced space

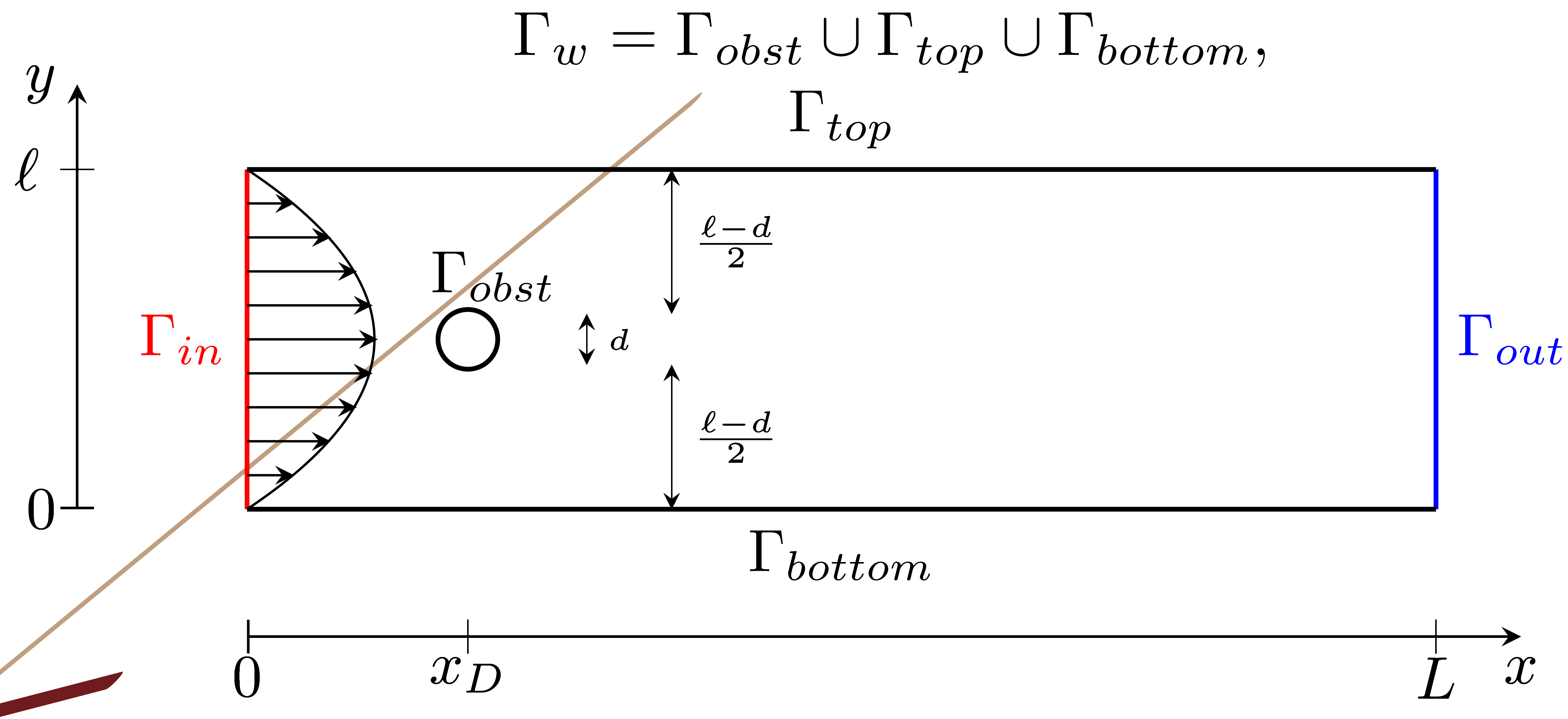




Numerical results—Navier-Stokes

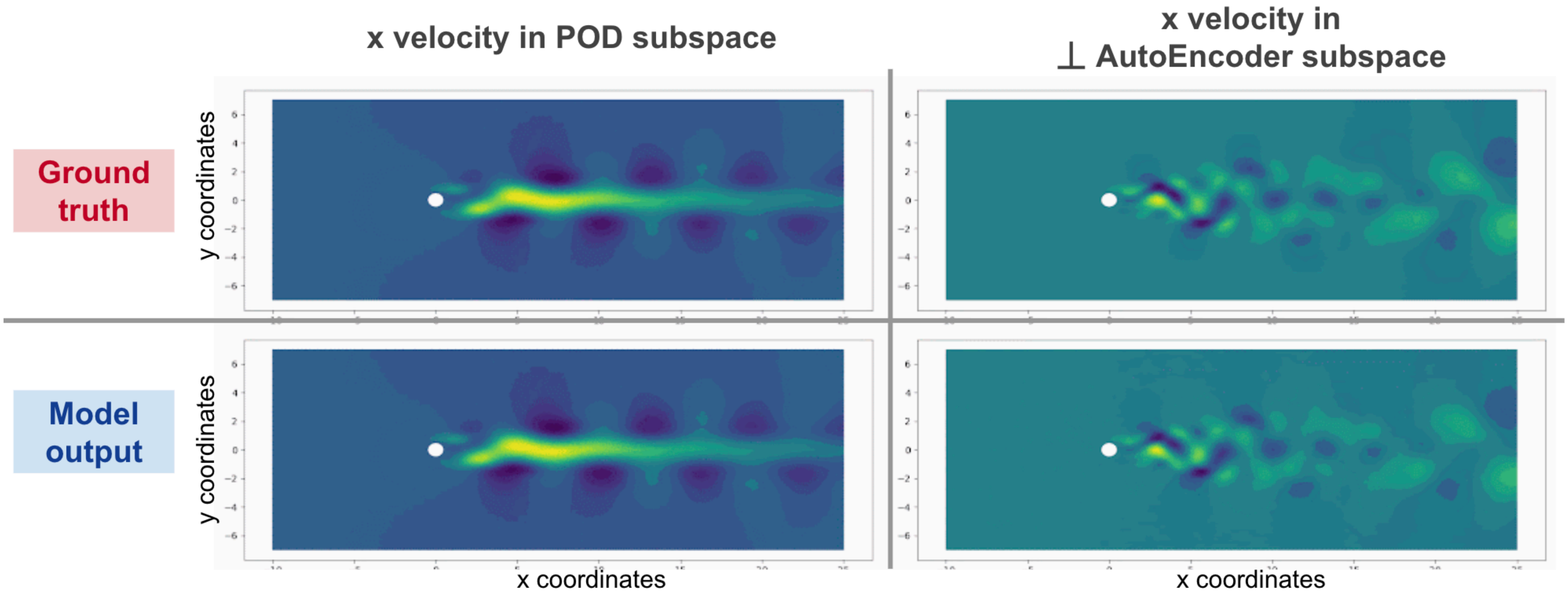
We now consider the Navier-Stokes equations in a channel with a cylindrical obstacle:

$$\begin{cases} \partial_t \mathbf{u} - \nu \Delta \mathbf{u} - (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p = 0 & \Omega, t > 0 \\ \nabla \cdot \mathbf{u} = 0 & \Omega, t > 0 \\ \mathbf{u}(\mathbf{x}, 0) = 0 & \Omega, t = 0, \\ \mathbf{u} = -g(y)\mathbf{n} & \text{on } \Gamma_{in}, \\ \mathbf{u} = 0 & \text{on } \Gamma_w \\ (\nu \nabla \mathbf{u} - pI)\mathbf{n} = 0 & \text{on } \Gamma_{out}, \end{cases}$$



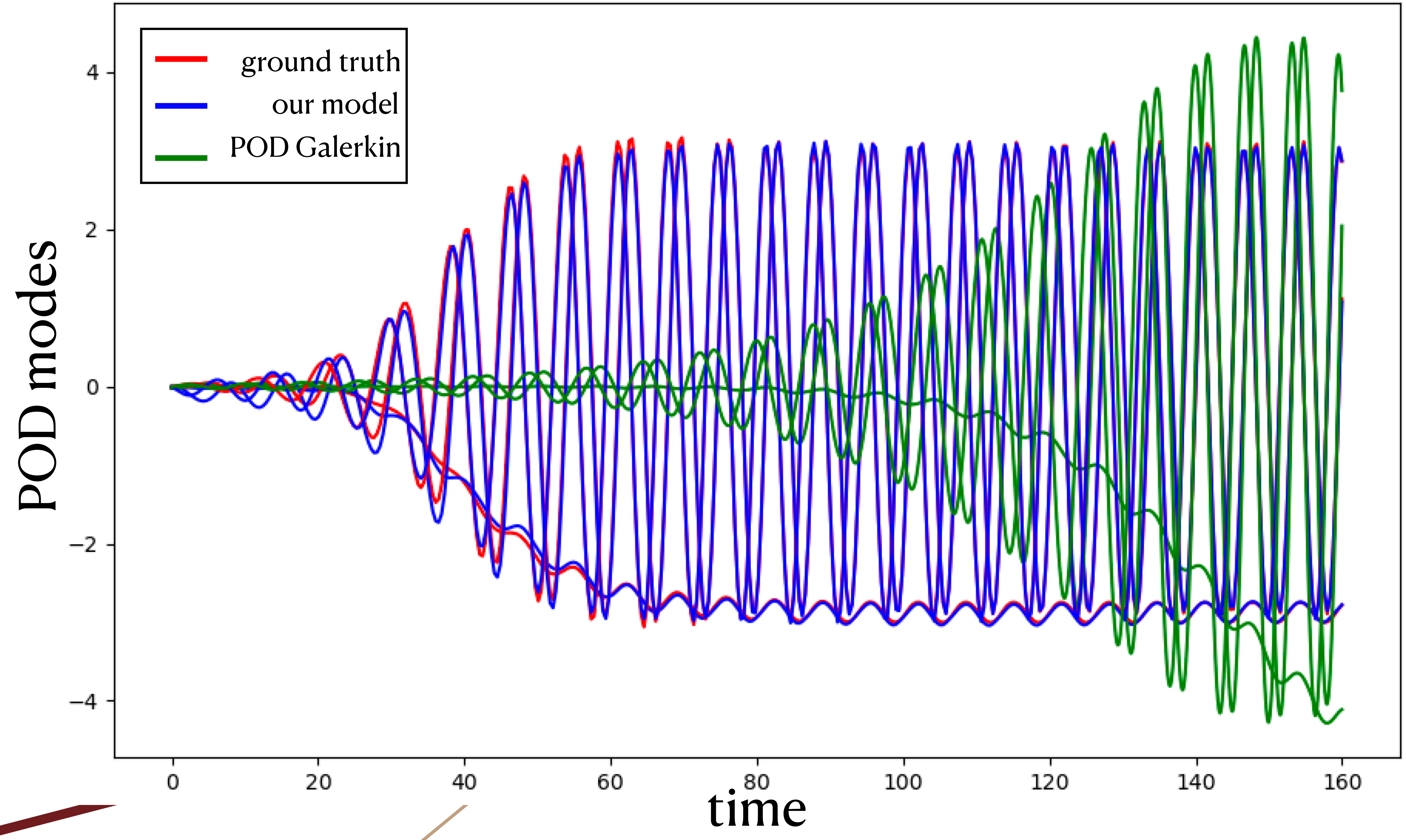
Numerical results—Navier-Stokes

Non parametric test case, fixed Reynolds (= 100). Results with 3 POD modes and 27 AE modes.



Numerical results—Navier-Stokes

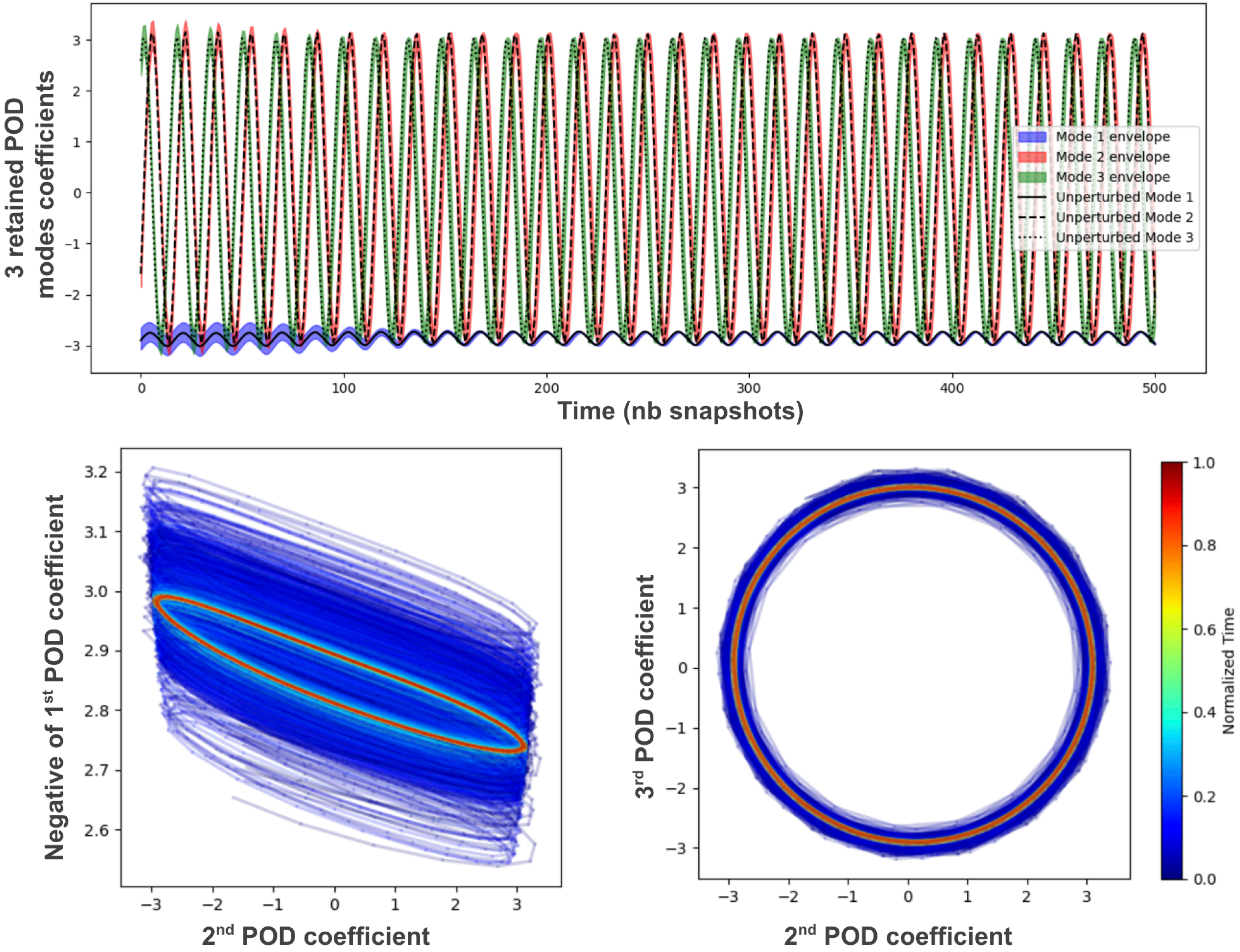
Non parametric test case, fixed Reynolds (= 100). Results with 3 POD modes and 27 AE modes.



Numerical results—Navier-Stokes

Non parametric test case, fixed Reynolds (= 100).

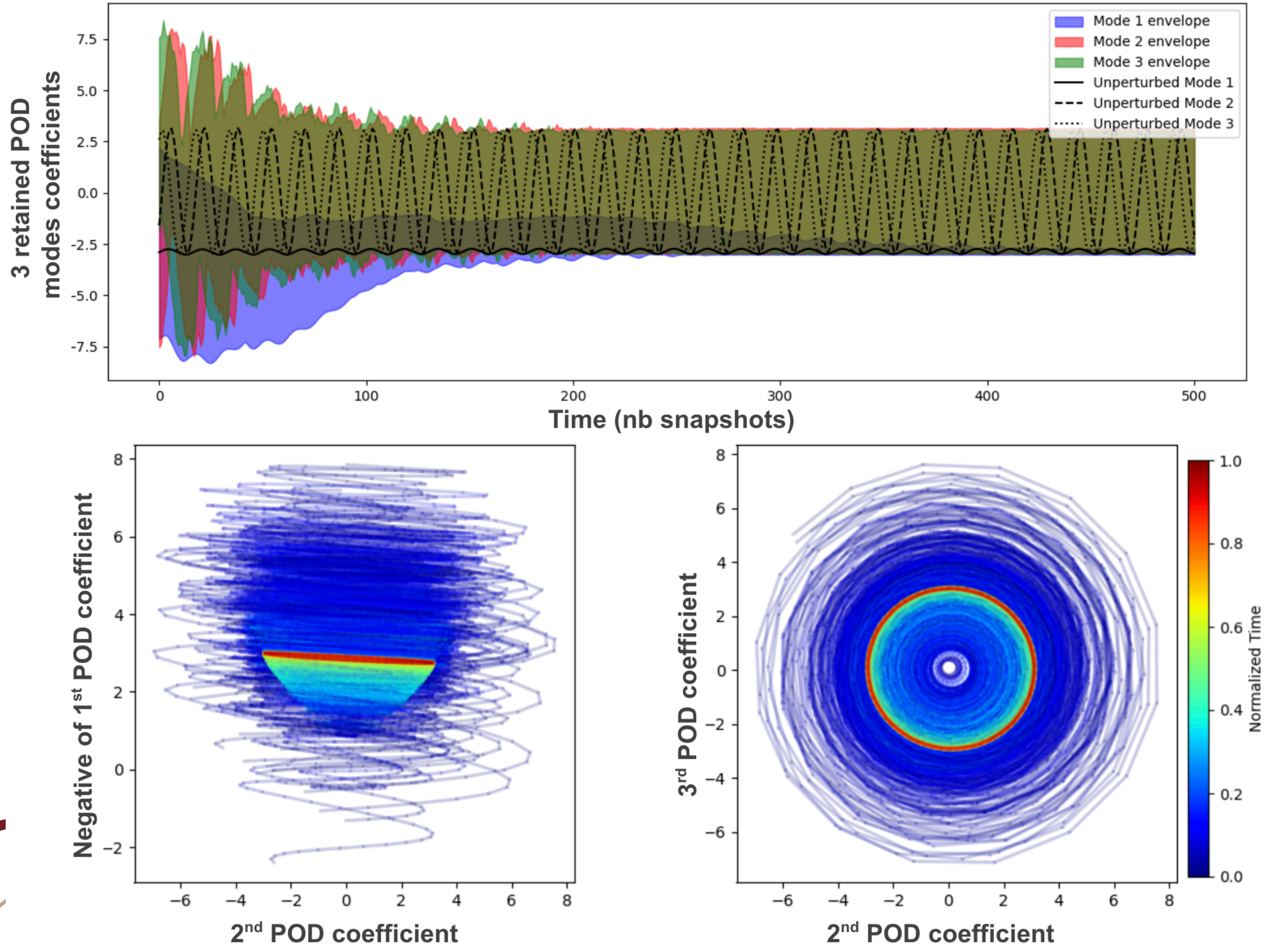
Robustness to **small perturbations**.



Numerical results—Navier-Stokes

Non parametric test case, fixed Reynolds (= 100).

Robustness to **large perturbations**.

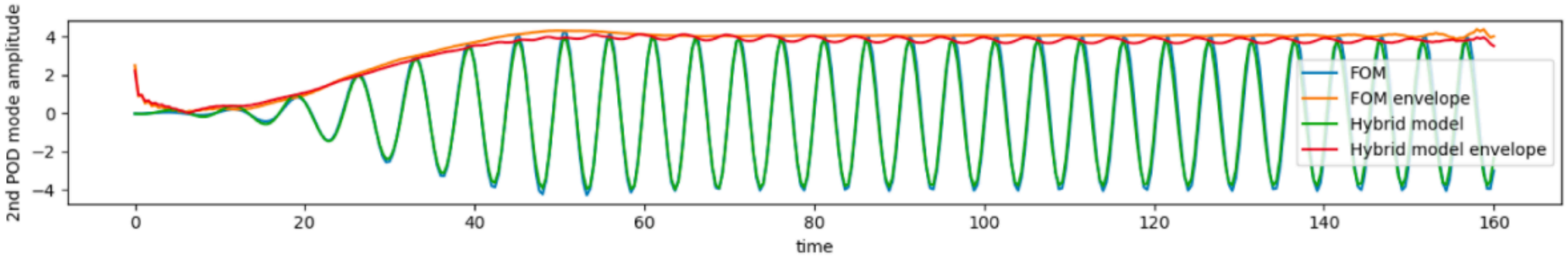


Numerical results—Navier-Stokes

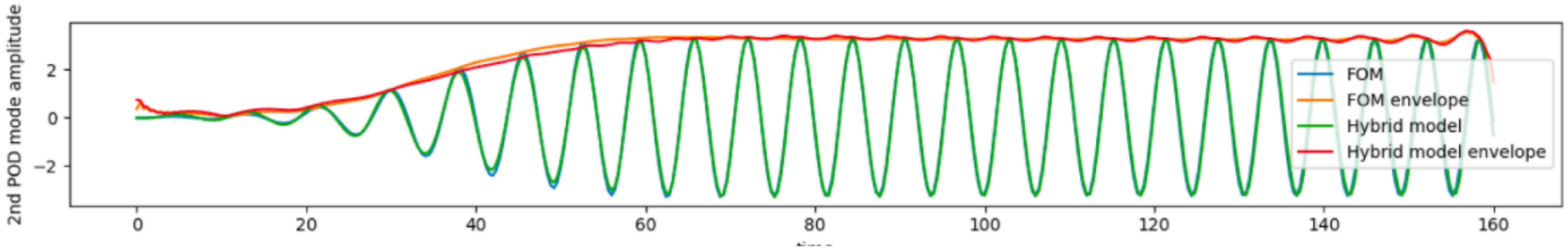
We now show a **parametric test case**.

Training set: Re = 60, 70, ..., 110, 120. Test set: Re = 50, 55, 65, 75, ..., 105, 115, 125

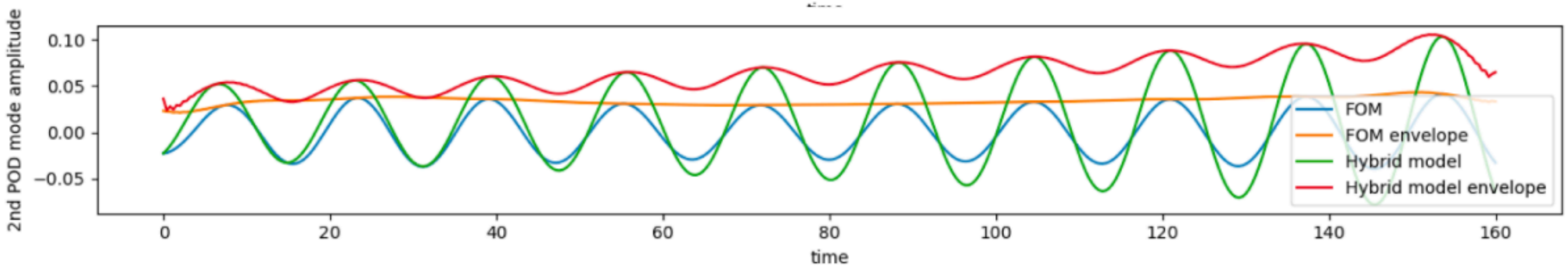
Re = 125



Re = 105



Re = 50



Numerical results—Navier-Stokes

Model	test-set MSE
Hybrid model 3 POD + 27 AE modes	$\mathbf{1.13 \times 10^{-4}}$
POD-Galerkin ROM 3 modes	1.42×10^{-2}
POD-Galerkin ROM 30 modes	6.53×10^{-3}
3 modes POD projection error	1.16×10^{-3}

Figure 18: Mean square error (MSE) on the complete time integration of the cylinder flow averaged for all reynolds for hybrid models and comparative models with 30 modes total latent size.

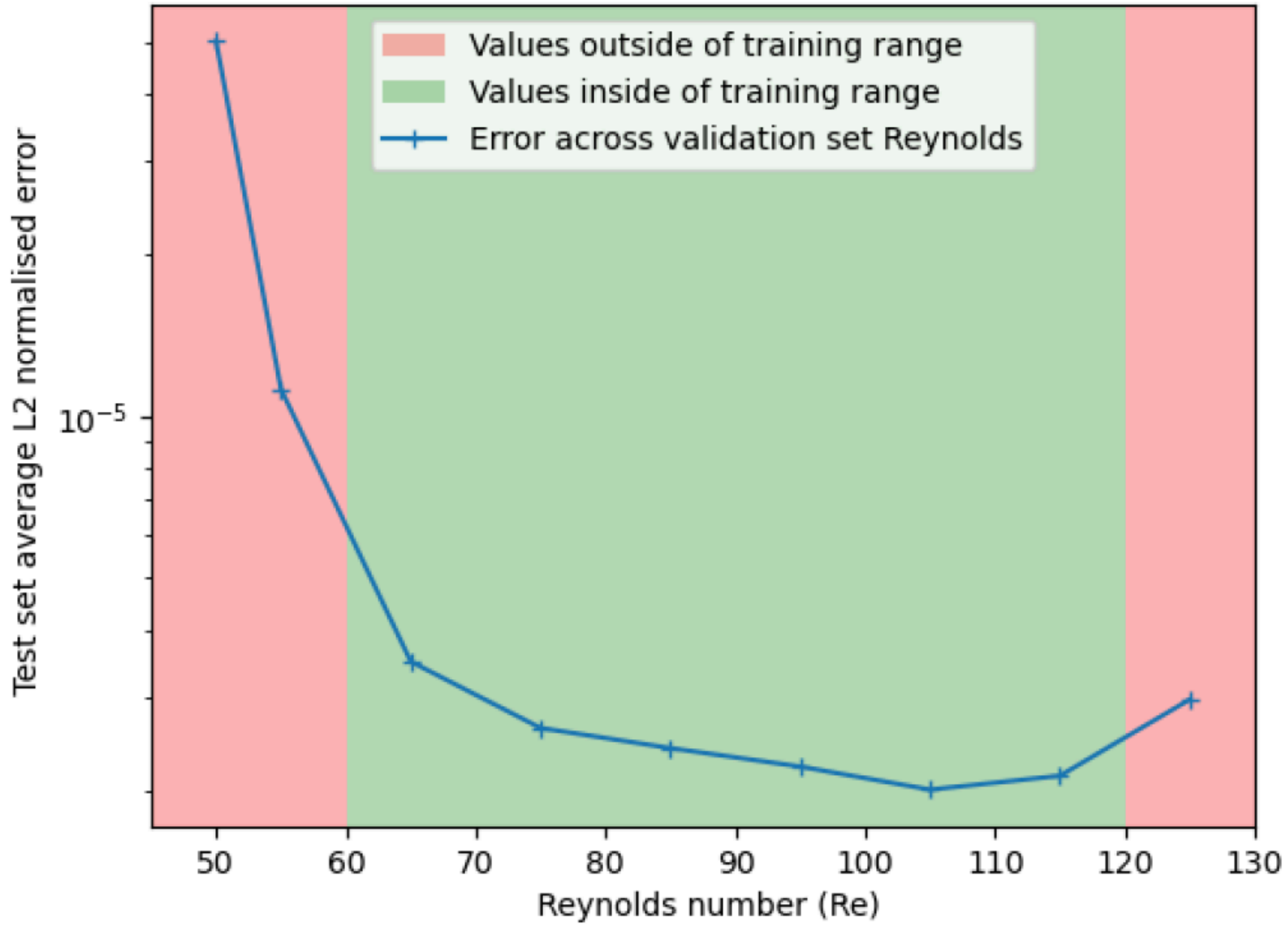
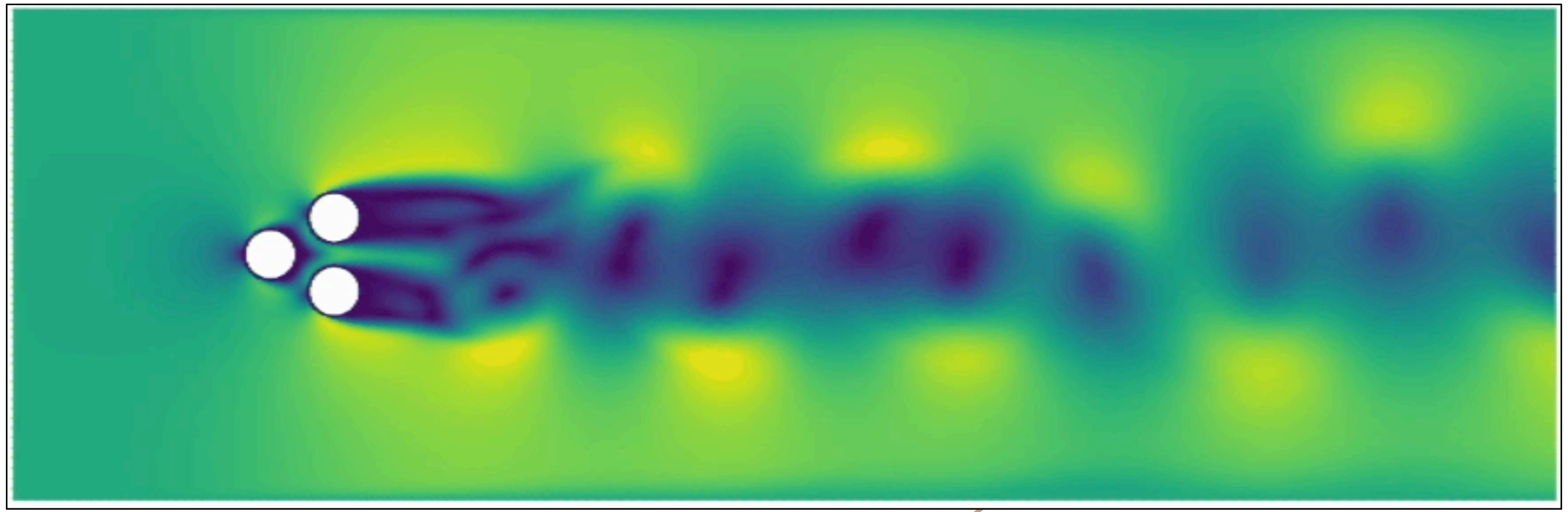


Figure 17: Plot of the relative MSE of the model for different Reynolds number test examples

Ongoing work—Fluidic pinball

We now consider a more complex test case for the Navier-Stokes equations.



Steady symmetric	Periodic symmetric	Periodic asymmetric	Quasi-periodic asymmetric	Chaotic symmetric

$Re_1 \simeq 18$ $\quad Re_2 \simeq 68$ $\quad Re_3 \simeq 104$ $\quad Re_4 \simeq 115$

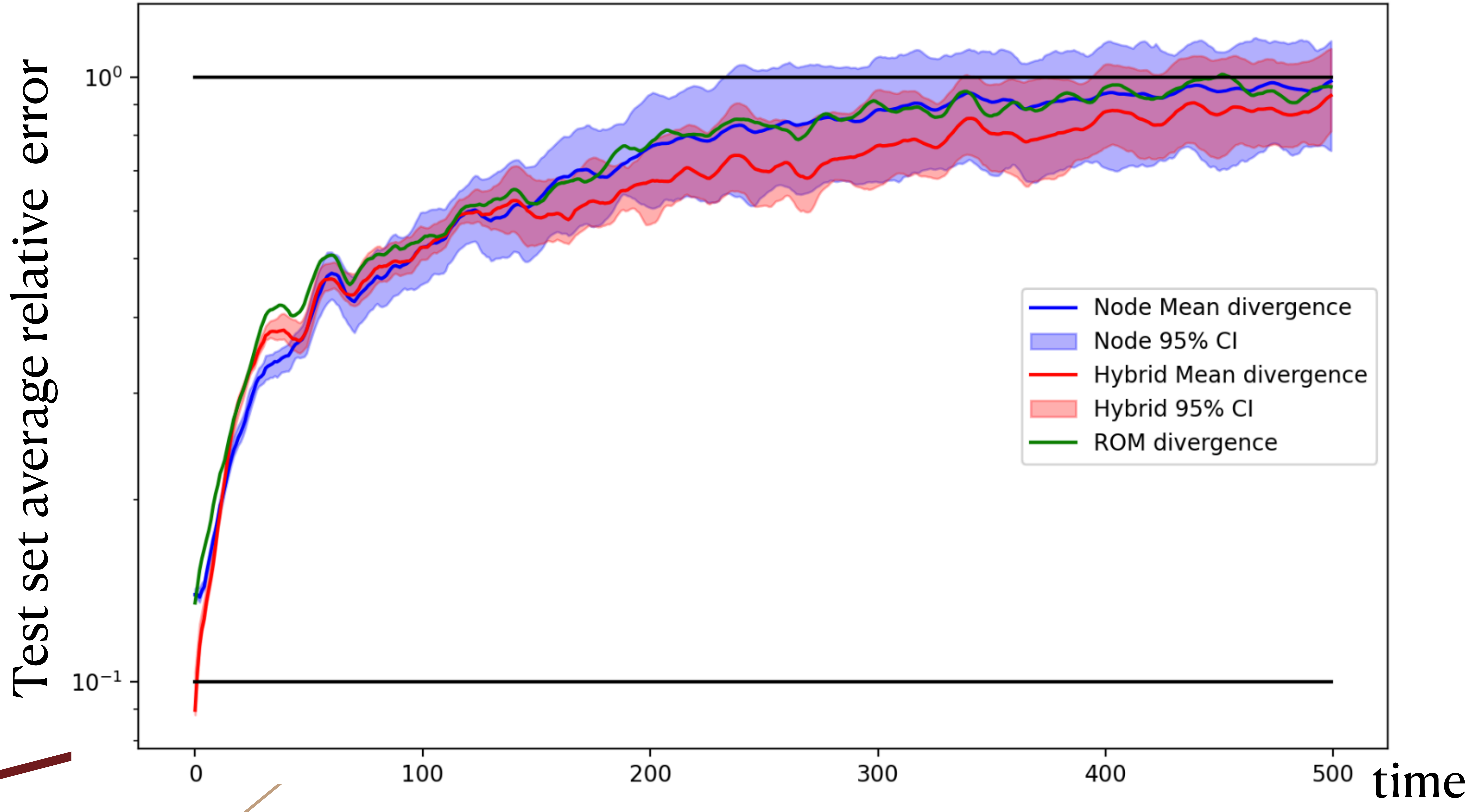
Ongoing work—Fluidic pinball

Chaotic test case (Re = 120)

Fully data driven:
40 AE modes

Hybrid model: 20 AE
+ 20 POD modes

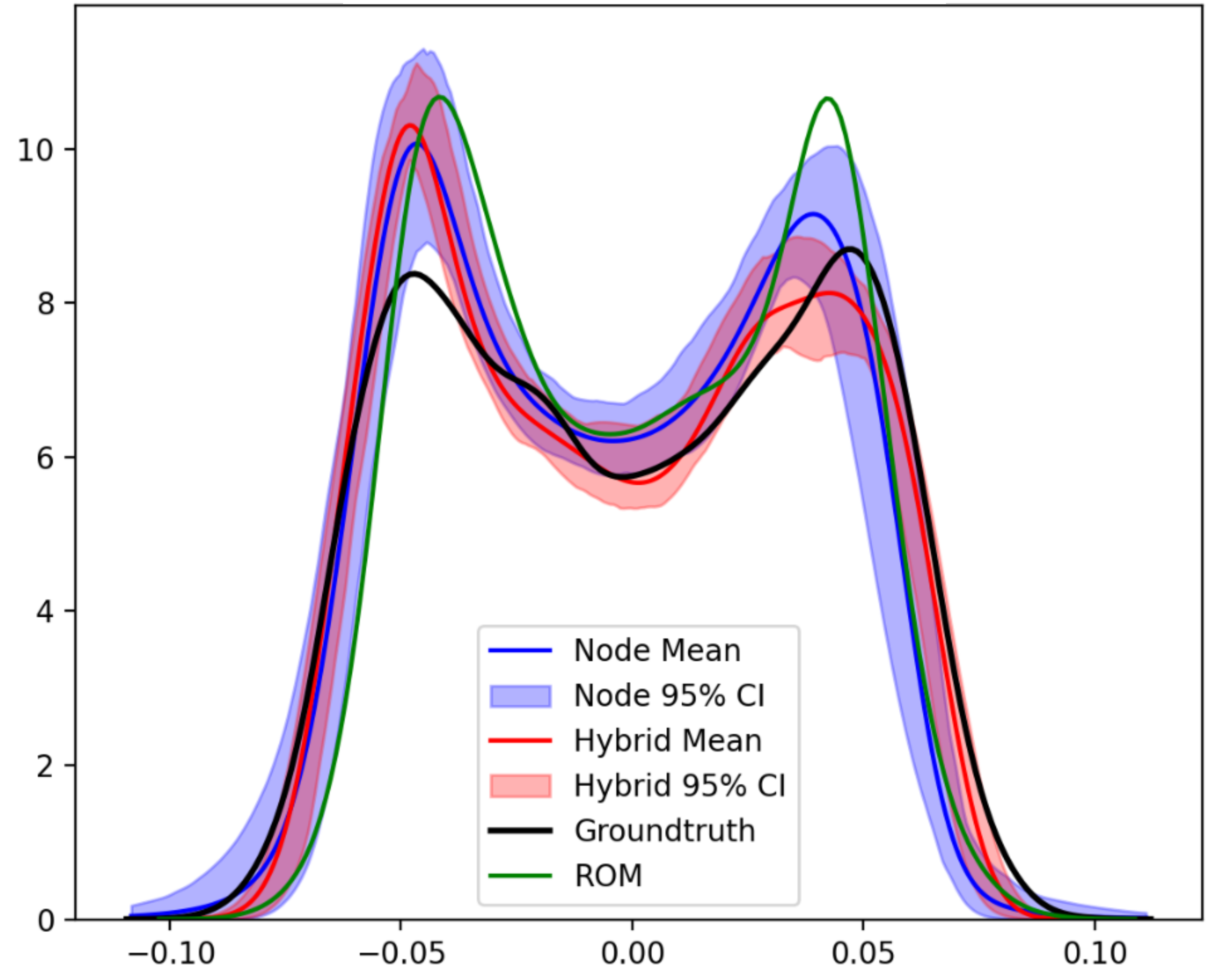
ROM: 20 POD modes



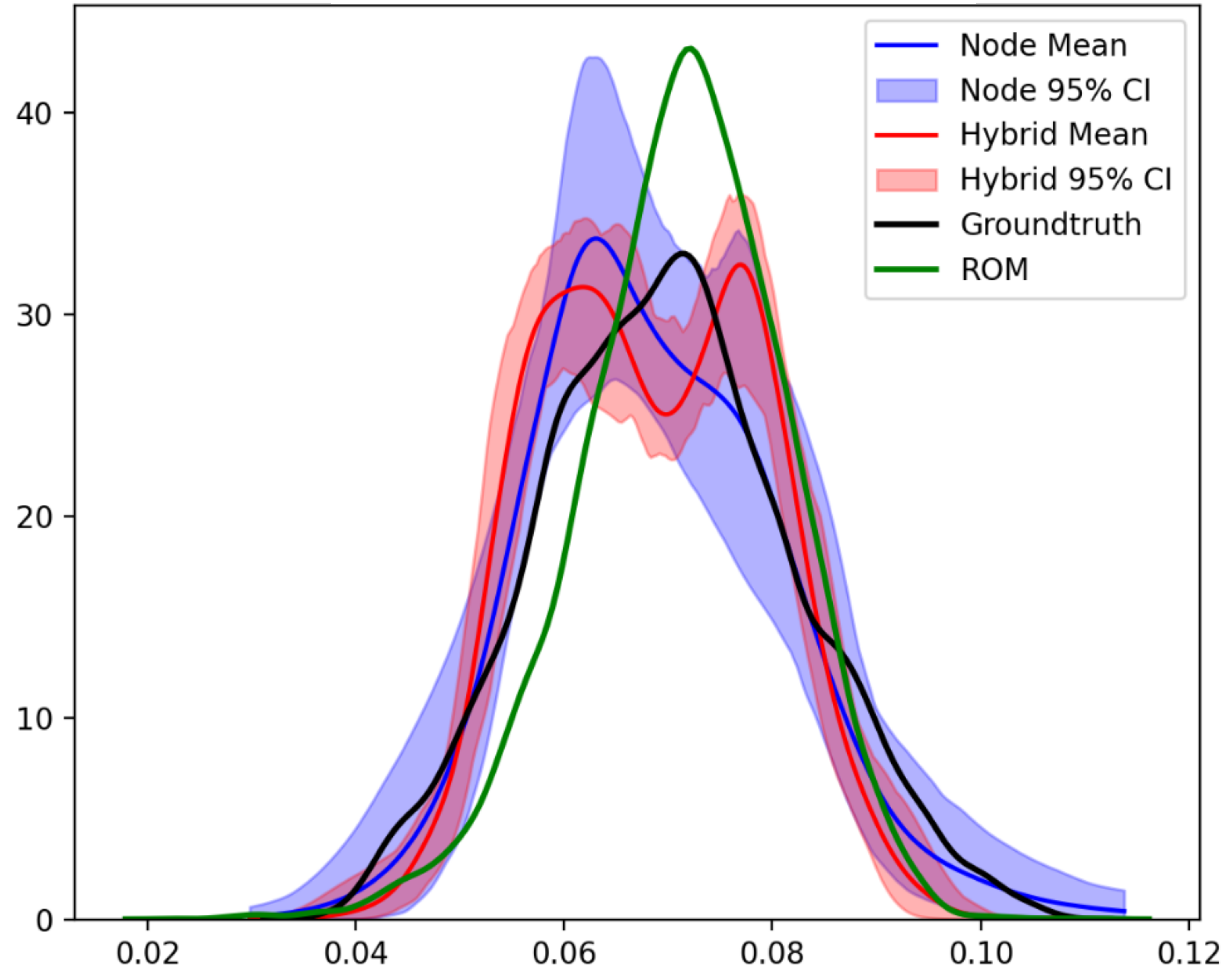
Ongoing work—Fluidic pinball

Chaotic test case (Re = 120)

Lift coefficient PDF



Drag coefficient PDF



Conclusion

- Hybrid approach outperforms standard ROMs and data driven methods in non chaotic cases
- Flexible framework
- Dynamics captured both in POD and orthogonal space

Thank you for your attention

SPARCL WORKSHOP

Structure-Preserving Approach for ROMs of Conservation Laws

April, 8th - 10th 2026

CNAM, Paris

Time dynamics of AE modes