Data-driven reduced order model for advection-dominated problems using neural-network shifted-proper orthogonal decomposition

Harshith Gowrachari[1], Nicola Demo[1], Giovanni Stabile[2] and Gianluigi Rozza[1]

[1]Mathematics Area, mathLab, SISSA, International School of Advanced Studies, Trieste, Italy
[2]Department of Pure and Applied Sciences, Informatics and Mathematics Section, University of Urbino Carlo Bo, Urbino, Italy

Introduction

Advection-dominated problems are commonly noticed in nature, in engineering systems, and in wide range of industrial processes. For these problems, linear compression methods (proper orthogonal decomposition and reduced basis method) are not suitable, as the Kolmogorov N-width decay is slow, which leads to inefficient and inaccurate reduced order models. **To accelerate the Kolmogorov N-width decay** there are a few recent pre-processing techniques that can be used to transform the full-order solutions [3, 2, 1]. Here, we use a **neural-network based pre-processsing technique** [2] that **automatically detects the optimal non-linear transformation of the full-order solutions** by exploiting a deep-learning architecture. In this work, we use this pre-processing technique to develop purely data-driven reduced order models for 1D traveling waves and a 2D two-phase flows.

Keywords: Neural Network shifted Proper Orthogonal Decomposition (NNsPOD), Reduced Order Model (ROM).

Neural Network shifted based pre-processing [2]

We define the **shift operator** $\mathcal{T}_b$ which acts on the field $u(x,t)$ as in (1), where $\mathbf{b} : \Omega \times T \to \Omega$, is the shifting quantity that is not a fixed value linearly depending on time as in [3], but here it depends on both, the space and the time coordinates. We use a neural network to learn this shift operator for a given problem. Another important step in this pre-possessing technique is the reconstruction of the field values in each shifted-space.

$$\tilde{u}(x,t) = \mathcal{T}_b u(x,t) = u(x - \mathbf{b}(x,t), t), \qquad x, \mathbf{b} \in \Omega, t \in T \tag{1}$$

$$\{\mathbb{V} \ni \mathbf{u}(t_i)\}_{i=1}^{N_s} = \{u(\mathbb{V}, t_i)\}_{i=1}^{N_s} \tag{2}$$

The learning process is built on the set of snapshots as a discrete field, as defined in (2), where $\mathbb{V}$ is the finite dimensional functional space defined on the discretized domain, and t_i are the N_s time steps. The automatic-shift detection and the reconstruction of field values workloads are divided between two separate neural networks, as mentioned below. The training of ShiftNet and InterpNet separated, at first InterpNet is trained and later the ShiftNet.

- ***ShiftNet*** quantifies the optimal-shift $\mathbf{b}(x,t)$, to obtain the shifted-space for each snapshot in the FOM solution manifold and the loss function employed for this process is,

$$\mathcal{L}_{shiftNet} = \frac{1}{N_s} \sum_{i=1}^{N_s} \| u_{\text{ref}}(x, t_i) - u(x - \mathbf{b}(x, t_i), t_i) \|_2, \quad x \in \mathbb{V} \tag{3}$$

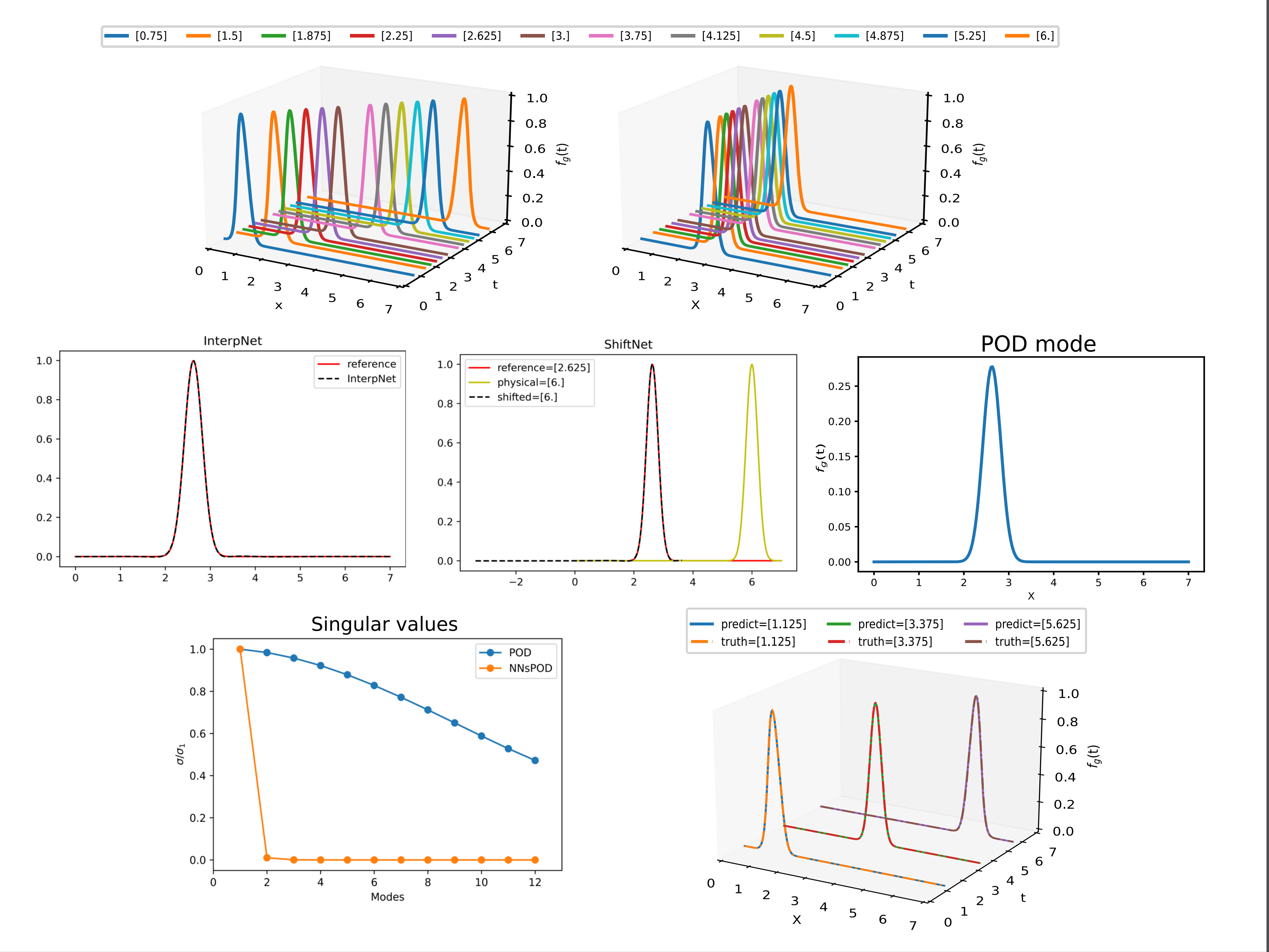
- ***InterpNet*** will learn reference configuration, such that it reconstructs the same configuration w.r.t each shifted space to compute the distance and employed loss function is,

$$\mathcal{L}_{interpNet} = \frac{1}{n} \sum_{i=1}^{n} (\tilde{u} - u_{\text{ref}}(x_i, t))^2, \quad x \in \mathbb{V} \tag{4}$$

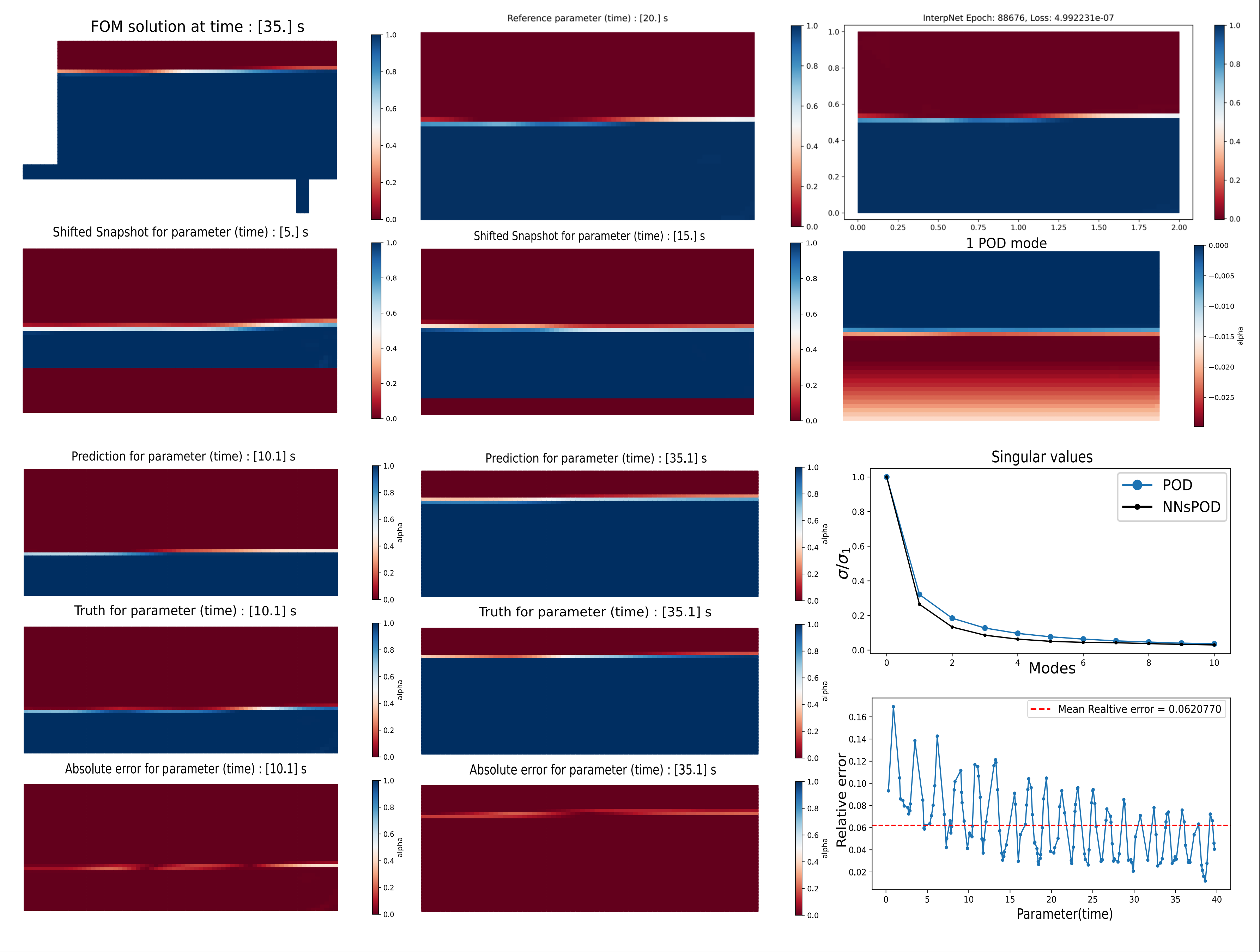
NNsPOD-ROM algorithm

```
FOM pre-processing : offline stage
while $\epsilon' > \epsilon_{interp}$ do
    InterpNet.forward ;
    compute $\epsilon'$ ;
    InterpNet.backward ;
end
$\mathcal{T}_{\text{interp}}$ = InterpNet.forward;          # reconstructs reference snapshot
for $\mu_i \in \mathcal{P}, \mathbf{u}_i \in \mathbf{X}$, where $i = 0$ to $N_{train}$ do
    while $\epsilon'' > \epsilon_{shift}$ do
        $\tilde{\mathbf{x}} = \mathbf{x}$ - ShiftNet.forward ;
        $\tilde{\mathbf{u}}_i = \mathcal{T}_{\text{interp}} \circ \tilde{\mathbf{x}}$;      # reconstruct snapshot in each shifted space
        compute $\epsilon''$ ;
        ShiftNet.backward ;
    end
    $\mathcal{T}_{\text{shift}}$=ShiftNet.forward;        # optimal shift for given parameter
    $\tilde{\mathbf{x}} = \mathbf{x} - \mathcal{T}_{\text{shift}}$ ;                              # shifted space
    $\mathcal{L}(\tilde{\mathbf{x}}, \mathbf{u}_i)$ ;                                 # linear interpolator
    $\tilde{\mathbf{u}}_i = \mathcal{L} \circ \mathbf{x}$;              # reconstruct snapshot in physical space
end
POD = POD(rank=N);
ROM = ROM($\mathcal{X}$, POD, $\bullet$)
s = POD.singular_values ;
FOM post-processing : online stage ;
for $\mu_i \in \mathcal{P}$, where $i = 0$ to $N_{test}$ do
    $\tilde{\mathbf{x}} = \mathbf{x} + \mathcal{T}_{\text{shift}}$ ;                              # shifted space
    $\mathcal{L}(\tilde{\mathbf{x}}, \mathbf{u}_{\text{ref}})$ ;                                 # linear interpolator
    $\tilde{\mathbf{u}}_i = \mathcal{L} \circ \mathbf{x}$;              # reconstruct snapshot in physical space
end
$R_i$= ROM.predict($\mu_{\text{new}}$);        # predicts snapshots for given parameters
```

1D Travelling Wave



Two-Phase Flow



Conclusion

- **1D Travelling Wave** : We notice sharp singular values decay and predict accurate results just by considering 1 mode.
- **Two-Phase Flow** : Qualitatively NNsPOD-ROM results (considering just 1 mode) outperforms standard POD-ROM results (considering 10 modes), though we do not achieve a sharp singular values decay.

References and Acknowledgements

[1] T. Long, R. Barnett, R. Jefferson-Loveday, G. Stabile, and M. Icardi. A novel reduced-order model for advection-dominated problems based on radon-cumulative-distribution transform. *arXiv preprint arXiv:2304.14883*, 2023.

[2] D. Papapicco, N. Demo, M. Girfoglio, G. Stabile, and G. Rozza. The neural network shifted-proper orthogonal decomposition: A machine learning approach for non-linear reduction of hyperbolic equations. *Computer Methods in Applied Mechanics and Engineering*, 2022.

[3] J. Reiss, P. Schulze, J. Sesterhenn, and V. Mehrmann. The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena. *SIAM Journal on Scientific Computing*, 40(3):A1322–A1344, 2018.

This work was partially supported by an industrial Ph.D. grant sponsored by Danieli S.p.A., and Programma Operativo Nazionale Ricerca e Innovazione 2014-2020, P.I. Gianluigi Rozza.

Opensource software

OpenFOAM®

EZyRB

EZyRB is a Python package that performs a data-driven model order reduction for parametrized problems.

github.com/mathLab/EZyRB   mathlab.github.io/EZyRB