Ablation study of a non-intrusive data-driven surrogate model for the Rayleigh-Bénard convection problem

Adriano M. A. Cortes[1], Roberto M. Velho[1], Gabriel F. Barros[2], Fernando A. Rochinha[3], Alvaro L. G. A. Coutinho[2]

[1] *High Performance Computing Center and Systems Engineering and Computer Science, COPPE/Federal University of Rio de Janeiro*
Av. Horácio Macedo 2030 CT, Bloco H - 319, 21941-914, Rio de Janeiro, Brazil
adricortes@cos.ufrj.br, robertovelho@cos.ufrj.br
[2]*High Performance Computing Center and Civil Engineering, COPPE/Federal University of Rio de Janeiro*
Av. Athos da Silveira Ramos, 149 CT, Bloco B - 101, 21941-909, Rio de Janeiro, Brazil
gabriel.barros@coc.ufrj.br, alvaro@coc.ufrj.br
[3]*Mechanical Engineering, Federal University of Rio de Janeiro*
Av. Horácio de Macedo, 2030 CT, Bloco G - 204, 21941-914, Rio de Janeiro, Brazil
faro@mecanica.ufrj.br

Abstract.

We present a non-intrusive data-driven parameter-dependent surrogate model, based on a neural network, for the Rayleigh-Bénard problem. Despite the problem being time-dependent, we employ an machine learning technique that does not necessitate a time structure such as an auto-regressive one or a recurrent network - for instance of Long short-term memory (LSTM) type. This approach significantly reduces the amount of data required for training. We underline that the problem is a coupled system of equations with complex dynamics. The model's performance is rigorously assessed on data both within and outside the interval of parameters used for training.

Keywords: Model order reduction, data-driven models, Rayleigh-Bénard, machine learning, coupled systems.

1 Introduction

Numerical models based on mathematical modeling can expand our understanding of complex physical systems. The Rayleigh-Bénard (RB) problem is a classical problem characterized by complex dynamics, corresponding to a flow between two horizontal plates where the top plate is held at a constant cool temperature and the bottom one is held at a higher temperature. At critical Rayleigh number the heated fluid near the bottom plate rises, while the cool fluid near the top is more dense and descends. This leads to circular convection cells in two dimensions. Understanding this phenomenon, and being able to reproduce it, is crucial, as it models multiple physical phenomena in nature. One of the challenges is the high computational cost of evaluating a high-fidelity model (HFM), or full-order model (FOM), names used interchangeably along this text. The present work aims to build a surrogate model to mitigate this drawback. This surrogate predictive model, or reduced order model (ROM), consists of a neural network that infers the buoyancy in the RB problem Fresca and Manzoni [1], Cracco et al. [2]. Data for training comes from FOM simulations for a set of values in the parameters space, time and Rayleigh number. We train such a model, perform predictions, and assess the error for values of Rayleigh numbers inside and outside the interval of the ones used for training.

2 Methodology

In this section, we describe the problem formulation, the process of generating data for training and testing the model, and the details on how the ROM is constructed.

2.1 Problem Setup

The Rayleigh-Bénard convection problem is defined here as the coupling of the incompressible Navier-Stokes and the energy (heat) equations. Assuming the dimensionless variables,

$$\mathbf{x}^* = \frac{\mathbf{x}}{L}, \quad t^* = \frac{t}{L/U_0}, \quad \mathbf{u}^* = \frac{\mathbf{u}}{U_0}, \quad p^* = \frac{p}{\rho\, {U_0}^2}, \quad \theta^* = \frac{g\beta\Delta T L^3}{\nu^2}, \tag{1}$$

with $\mathbf{x}$ the Cartesian coordinates, t time, $\mathbf{u}$ the velocity, p the pressure, $\Delta T = T - T_0$ the temperature difference (assuming T_0 as the reference temperature the temperature in the top of the domain), U_0 the reference velocity, L the reference length, g the gravitational acceleration, ρ the density, β the thermal expansion coefficient, and ν is the kinematic viscosity. From now on we will drop the superscript $*$ in the equations. Therefore, the system of equations is written in dimensionless form as

$$\nabla \cdot \mathbf{u} = 0, \tag{2a}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \sqrt{\frac{Pr}{Ra}}\nabla^2 \mathbf{u} + \theta\, \mathbf{e}_2, \tag{2b}$$

$$\frac{\partial \theta}{\partial t} + (\mathbf{u} \cdot \nabla)\theta = \frac{1}{\sqrt{Ra \cdot Pr}}\nabla^2 \theta, \tag{2c}$$

where $\mathbf{e}_2$ is the canonical unitary vector in y direction, eq. (2a) is the continuity equation, eq. (2b) is the momentum equation, and eq. (2c) describes the energy equation in terms of the thermal buoyancy. All the fields are functions of $(t, \mathbf{x})$, with $\mathbf{x} \in \Omega \subset \mathbb{R}^2$. The dimensionless numbers

$$Pr = \frac{\nu}{\alpha}, \quad Ra = \frac{g\beta\Delta T L^3}{\nu\alpha} \tag{3}$$

represent the Prandtl and Rayleigh numbers, respectively, with α being the thermal diffusivity. Regarding the boundary conditions, we have no-slip conditions for $\mathbf{u}$ on the entire boundary of Ω, a rectangular domain, and prescribed θ on the top and bottom, with no-flux on both sides. The initial condition is zero velocity throughout the domain, with a constant gradient buoyancy profile pointing downward, augmented with a small Gaussian random disturbance to trigger the instability.

2.2 Dataset Generation and Processing

The equations are discretized by a spectral method using the open-source library Dedalus, Burns et al. [3]. The problem is non-dimensionalized using the rectangle height, such that the rectangular domain Ω has dimensions 4 by 1, with 256 by 64 grid points in each direction, respectively, and Chebyshev basis in both direction. All simulations are performed from time 0 to 100 with a fixed time step of 1×10^{-3} with a 2nd-order, 2-stage, Diagonally Implicit Runge–Kutta method. Details on how to solve the RB problem numerically are found in Griebel et al. [4]. A snapshot of the dynamics was taken every 0.25 time units, thus generating 401 snapshots per simulation. Both buoyancy and vorticity are calculated as outputs, but we selected only the buoyancy snapshots to perform the parametric model order reduction technique.

Note that, although we have chosen a fixed time step for the FOM simulation, the machine learning method explained in section 2.3 could be constructed with a non-equally spaced time intervals. The choice for the equally spaced one in here comes from the fact that, for different Rayleigh numbers, the time instants when the convection cells formation starts and the system reaches its steady-state is different. We thus opt out for giving simulations of different Rayleigh values the same time resolution for the different time instants. If we were constructing a surrogate model for a fixed Rayleigh number, and for simplicity having only time as its variable, we could choose the collection of data at time instants of our interest, for instance giving more attention to a specific time spam of the dynamics. The construction of the surrogate model described ahead would follow accordingly.

Based on the phase-diagram in Kevrekidis et al. [5], and to guarantee the presence of steady states, we fixed $Pr = 0.15$ and varied Ra with the following values: $\{10, 12, 14, 16, 17, 18, 20, 22, 24, 25\} \times 10^3$. The simulations

corresponding to the values of 17×10^3 and 25×10^3 are used as test data for the machine learning model on which the predictive reduced model is based. The simulations for the remaining values, totaling 8, are used as training data. Out of the 401 time instants collected for each Rayleigh number, we have selected 252, where we can observe the characteristic convection cell formation in the RB problem evolving to a steady state. In terms of the time evolution of the simulation, this corresponds to the interval from 24 to 87, out of the FOM simulated one from 0 to 100.

2.3 Description of the Technique

Each snapshot generated for the FOM with dimensions 256 by 64 is reshaped into a single vector of dimension 16384. All the selected 252 snapshots (now vectors) are concatenated in time-ascending order, generating a matrix of dimensions 16384 by 252 for each Rayleigh number. Given the high dimensionality of the simulation data, directly fitting a predictive model could be prohibitive. We start the ROM technique by assembling a large matrix, concatenating all the 8 matrices, thus generating a matrix of dimensions 16384 by 2016, which we call S. The following step is to perform the Singular Value Decomposition (SVD) of S.

We choose the first N modes (the choice for the concrete value of N is discussed ahead), corresponding to the largest singular values, and project the spatial information from dimension 16384 into a reduced basis of dimension N. We construct a feature matrix with 2 columns, Rayleigh value and time instant. Each entry (row) of such matrix maps these two values to the corresponding spatial data in the reduced basis of dimension N for the corresponding Rayleigh value and time.

A supervised learning problem using the previous feature matrix (having dimensions 2016 by 2) and the set of target vectors of dimension N is trained using a neural network (NN). The parameters associated to this NN and the training process are presented below. All the machine learning code was implemented using the *PyTorch* framework. Other authors have performed close investigations to the one here presented, see Fresca and Manzoni [1] - where a convolutional autoencoder is coupled to a neural network and Guo and Hesthaven [6] - where the neural network is substituted by a Gaussian Process.

The NN in our setup is composed by a certain number of hidden layers each with several neurons followed by a Sigmoid activation function. The weight initialization uses the Kaiming Normal (with mode "fan in" and Leaky Relu as nonlinearity) technique, and the chosen loss function is the mean squared error. This predictive supervised learning model is tested as follows: we construct one test matrix for each value of Ra selected for this purpose, 17×10^3 and 25×10^3, and for all the 252 times (the same used for training), and the trained model is applied to each of these test matrices. We thus obtain the predicted target vectors of dimension N in the reduced basis for all times for each of these Ra values. By using the SVD, we can recover this information in the original physical basis. We are then able to compare these predictions against the FOM data for these values of Ra and assess the predictive model's performance. The predictions and results for our model are presented in section 3.

Note that, if we wished to give attention to specific moments of the dynamics, or construct surrogates for different time spams of the collected FOM data, we could perform a time split over the interval of collected time instants and assemble different surrogates for each split, as in Cracco et al. [2]. This may help the surrogate to learn data specific to one characteristic moment of the dynamics, as the convection cell formation or the steady-state. It also reduces the hardware requirements and the training time for each surrogate, as less data is used. Another possible benefit is training the model in a parallel way (distributed computing).

3 Results and Ablation Studies

Concerning the reduced basis, observe that, by using a low-dimensional one, we lose more data via the SVD than with a higher-dimensional one, and the predictive model may not even be able to capture the necessary information to properly learn. In contrast, using a high-dimensional one, we may encounter the problem of training a model whose input is 2-dimensional (Ra and time) with a high-dimensional target vector. Note that the number of examples (rows in the feature matrix) is invariant once the FOM data is produced and selected. The use of a high-dimensional basis may again affect the learning process of the predictive model, due to a lack of enough examples. This suggests there is a certain interval within which to search for a good value of the dimension N of the reduced basis. Regarding the architecture of the NN, the typical issues with such models arise. A network with few neurons may not be able to properly learn all the training data, while one with too many neurons may suffer from overfitting and not being able to generalize.

We performed ablation studies to understand the influence of each parameter on the model's performance. In our experiments, we analyzed N between 4 and 16, choosing 12 as the best value. We tested networks with 2 to 5 hidden layers and 20 to 60 neurons in each one, selecting 3 as the number of layers and 40 for the number of

neurons in each layer. Other parameters related to the optimization process of training were also analyzed, such as the batch size, the learning rate (LR), the number of epochs, etc. The chosen values were: 1500 epochs, an initial LR of 5×10^{-3} (using an adaptive scheme - scheduler - with a multiplicative factor of 0.9 every 25 epochs), Adam optimizer (with *PyTorch* default parameters), and a 6 as batch size. The predictive model was then trained with those chosen optimal values. The final average loss for the training was 1×10^{-7} while using a LR of 1×10^{-5} for the last epochs (regard the usage of the scheduler).

We analyze the error in the buoyancy values between the ground truth and the predicted ones for the Rayleigh numbers selected for test. For each time instant, we measure the absolute error using the l_∞-norm and the ratio of the l_2-norm of the absolute error divided by the l_2-norm of the ground truth. Note that the norms are calculated over the entire set of grid points. We present the norm values in Fig. 1.

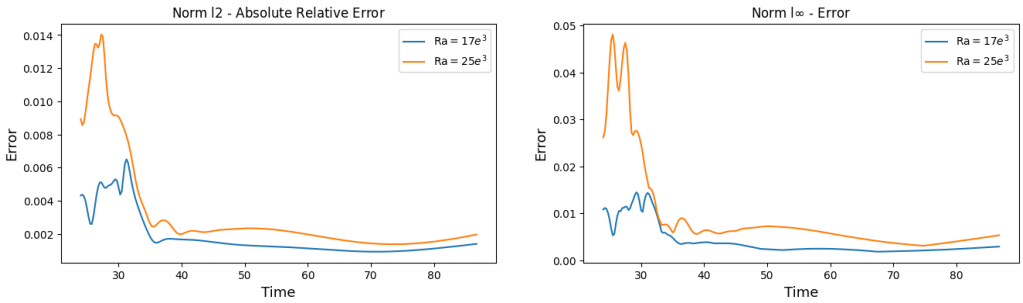

Figure 1. Relative errors in the l_2 and l_∞ norms for $Ra = 17 \times 10^3$ and $Ra = 25 \times 10^3$

Based on the error norms, we pick three different time instants from each Rayleigh test value in order to observe the error at each grid point. We chose the time 31 for the smaller Rayleigh test value and 27 for the larger one, as those are the instants where the l_2-norm reaches the maximum value for each of those, respectively. We also pick two later time instants, 50 and 84, in order to observe different moments of the dynamics. We compare the ground truth and predictions for the chosen times for $Ra = 17 \times 10^3$ in Fig. 2. Additionally, We conducted a similar analysis for $Ra = 25 \times 10^3$ and the results are presented in Fig. 3.

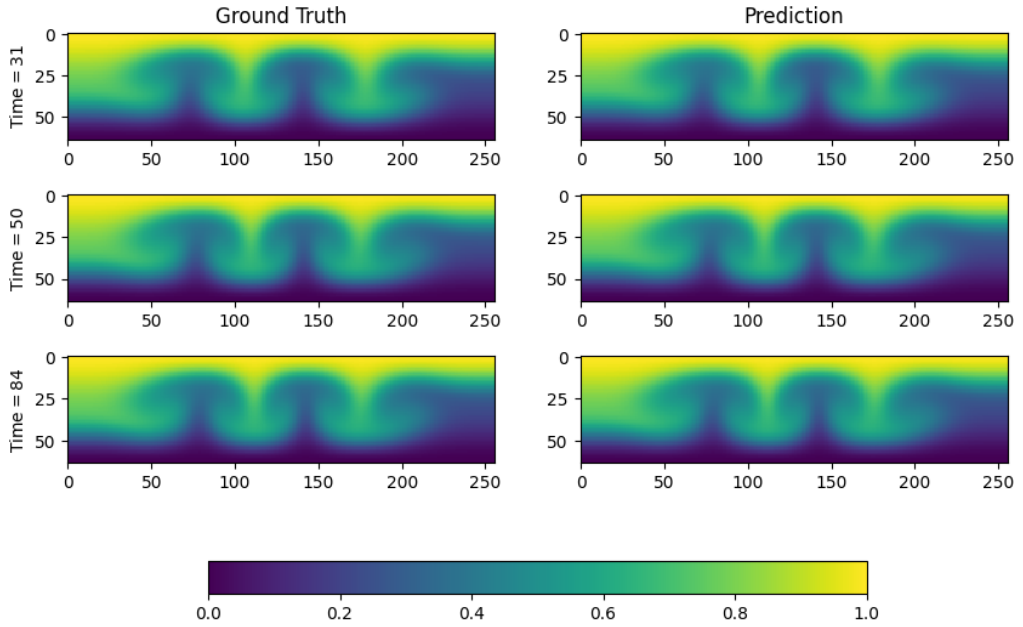

Figure 2. Ground Truth versus Predictions for $Ra = 17 \times 10^3$ at multiple time instances.

The absolute error at each grid point for the selected times can be seen in Fig. 4 and Fig. 5. While the relative errors are shown in Fig. 6. We draw attention to the fact such models may be hard to training. Concerning

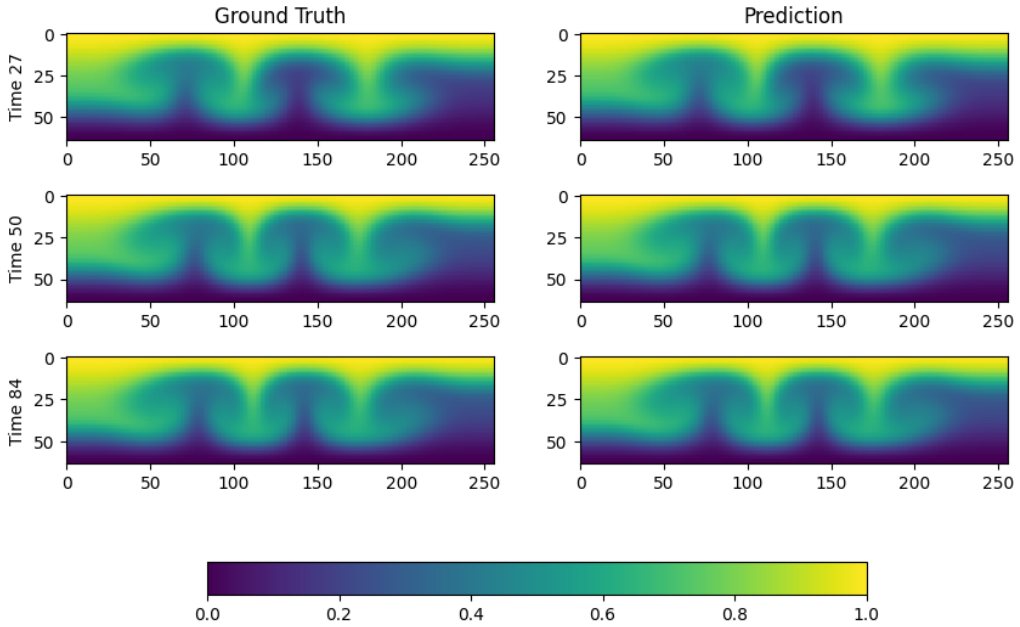

Figure 3. Ground Truth versus Predictions for $Ra = 25 \times 10^3$ at multiple time instances.

hyperparameters, an increment of 500 epochs, from the 1500 used, can generate a model closer to over-fitting, where we observe better accuracy for the lower Ra value (in the interior of the set of trained Rayleigh values) and a degradation of the accuracy for the higher one. This shows a reduction in the model's generalization capacity. Compare the relative errors in Fig. 6 and in Fig. 7. Now, if we changed the architecture from 3 layers with 40 neurons each to 2 layers with 50 neurons each, thus a reduction of about 16% in the number of neurons, while keeping the number of epochs, we may obtain models with a huge difference in performance. Observe Fig. 8, in which we can observe relative errors for the smaller test Rayleigh value reaching up to 15%, while for the higher one the error reaches higher than 50%.

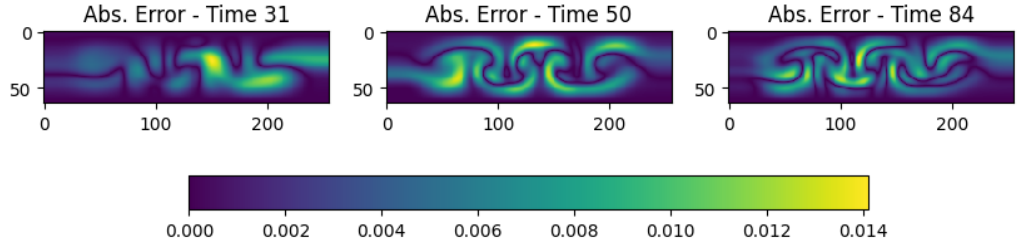

Figure 4. Absolute pointwise error for multiple times for $Ra = 17 \times 10^3$.

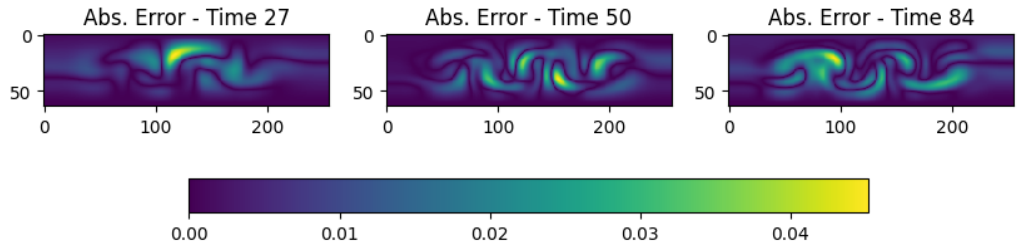

Figure 5. Absolute pointwise error for multiple times for $Ra = 25 \times 10^3$.

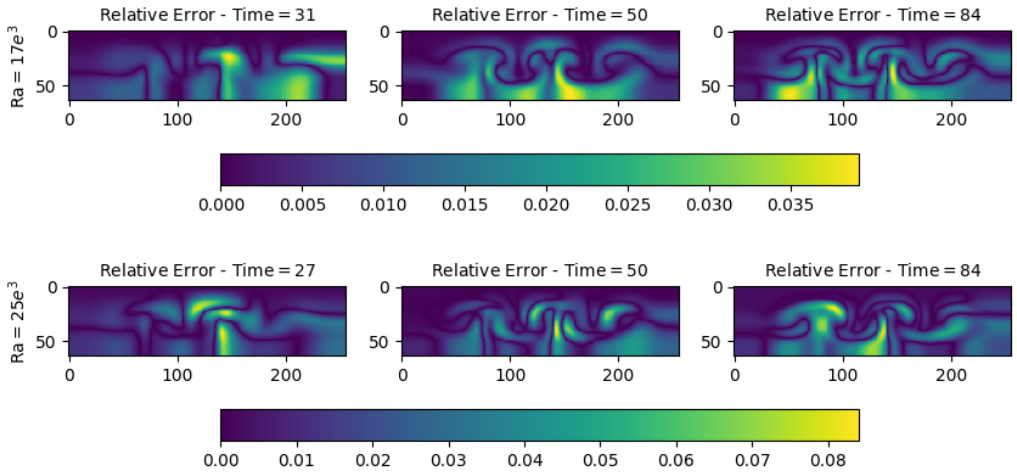

Figure 6. Pointwise relative errors for multiple times for $Ra = 17 \times 10^3$ (top) and $Ra = 25 \times 10^3$ (bottom) for the best configuration - 1500 epochs.

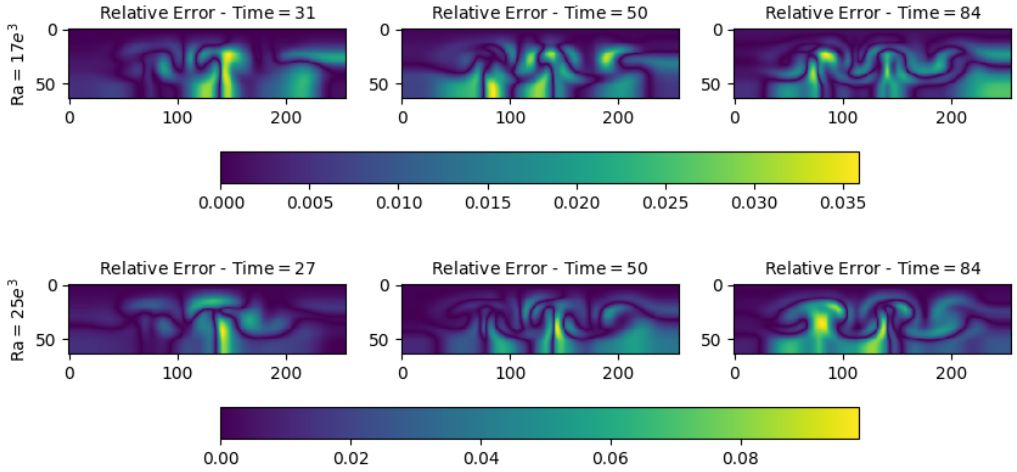

Figure 7. Pointwise relative errors for multiple times for $Ra = 17 \times 10^3$ (top) and $Ra = 25 \times 10^3$ (bottom) - using 2000 epochs.

4 Conclusions

We observe the success of the predictive model in reconstructing the dynamics, both in an earlier stage (associated with the convection cell formation in the RB problem) and as it moves to a steady-state condition. For the test value with a Rayleigh number in the interior of the interval of trained values (although not seen during the training step), $Ra = 17 \times 10^3$, we observe better performance, with relative errors smaller than 4%, than for the larger test value, $Ra = 25 \times 10^3$, with relative errors reaching values slightly larger than 8%. We also observed higher errors via the l_2- and l_∞-norms at the initial times. Those times are associated to the fast dynamics of the formation of the convection cells. Also observe locally higher values at the end of the simulation times. Despite the possible inability of the model to learn these stages, such behavior is expected as no examples are provided in the feature matrix for times lower than 24 or higher than 87. Concerning the computational speed-up using this surrogate predictive model, while the FOM data took about two days to be generated, the online stage of reconstructing the dynamics for all times (the same used in training) takes less than one second. The offline stage associated to the SVD and the training of the NN was about 30 minutes for each network configuration.

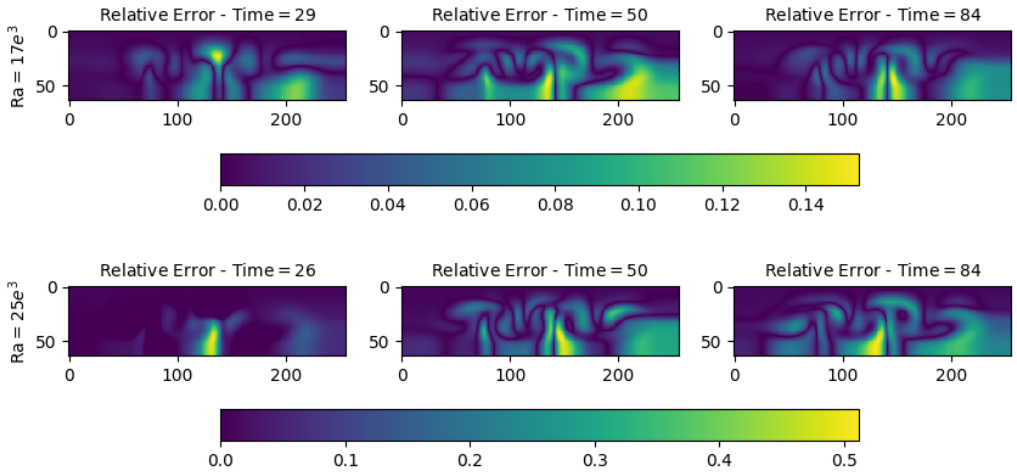

Figure 8. Pointwise relative errors for multiple times for $Ra = 17 \times 10^3$ (top) and $Ra = 25 \times 10^3$ (bottom) - smaller network.

Acknowledgements. This study was financed in part by CAPES, Brazil - Finance Code 001. This work is also partially supported by FAPERJ, CNPq, and Petrobras.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] S. Fresca and A. Manzoni. Pod-dl-rom: Enhancing deep learning-based reduced order models for nonlinear parametrized pdes by proper orthogonal decomposition. *Computer Methods in Applied Mechanics and Engineering*, vol. 388, 2022.

[2] M. Cracco, G. Stabile, A. Lario, A. Sheidani, M. Larcher, F. Casadei, G. Valsamos, and G. Rozza. Deep learning-based reduced-order methods for fast transient dynamics. *arXiv:2212.07737*, 2022.

[3] K. J. Burns, G. M. Vasil, J. S. Oishi, D. Lecoanet, and B. P. Brown. Dedalus: A flexible framework for numerical simulations with spectral methods. *Physical Review Research*, vol. 2, n. 2, pp. 023068, 2020.

[4] M. Griebel, T. Dornseifer, and T. Neunhoeffer. *Numerical simulation in fluid dynamics: a practical introduction*. Society for Industrial and Applied Mathematics, USA, 1998.

[5] I. Kevrekidis, R. Rico-Martínez, R. Ecke, R. Farber, and A. Lapedes. Global bifurcations in rayleigh-bénard convection. experiments, empirical maps and numerical bifurcation analysis. *Physica D: Nonlinear Phenomena*, vol. 71, n. 3, pp. 342–362, 1994.

[6] M. Guo and J. S. Hesthaven. Data-driven reduced order modeling for time-dependent problems. *Computer Methods in Applied Mechanics and Engineering*, vol. 345, 2018.