Advances in Computational Mathematics (2024) 50:33
https://doi.org/10.1007/s10444-024-10110-1

Error estimates for POD-DL-ROMs: a deep learning framework for reduced order modeling of nonlinear parametrized PDEs enhanced by proper orthogonal decomposition

Simone Brivio[1] · Stefania Fresca[1] · Nicola Rares Franco[1] · Andrea Manzoni[1]

Received: 17 May 2023 / Accepted: 15 February 2024 / Published online: 24 April 2024

Abstract

POD-DL-ROMs have been recently proposed as an extremely versatile strategy to build accurate and reliable reduced order models (ROMs) for nonlinear parametrized partial differential equations, combining *(i)* a preliminary dimensionality reduction obtained through proper orthogonal decomposition (POD) for the sake of efficiency, *(ii)* an autoencoder architecture that further reduces the dimensionality of the POD space to a handful of latent coordinates, and *(iii)* a dense neural network to learn the map that describes the dynamics of the latent coordinates as a function of the input parameters and the time variable. Within this work, we aim at justifying the outstanding approximation capabilities of POD-DL-ROMs by means of a thorough error analysis, showing how the sampling required to generate training data, the dimension of the POD space, and the complexity of the underlying neural networks, impact on the solutions us to formulate practical criteria to control the relative error in the approximation of the solution field of interest, and derive general error estimates. Furthermore, we show that, from a theoretical point of view, POD-DL-ROMs outperform several deep learning-based techniques in terms of model complexity. Finally, we validate our findings by means of suitable numerical experiments, ranging from parameter-dependent operators analytically defined to several parametrized PDEs.

Keywords Operator learning · Neural networks · Approximation bounds · Reduced order modeling · Parametrized PDEs · Deep learning-based reduced order modeling

Mathematics Subject Classification (2010) 65M99 · 65P99 · 68U01

Communicated by: Tobias Breiten

✉ Andrea Manzoni
andrea1.manzoni@polimi.it

Extended author information available on the last page of the article

1 Introduction

Solutions to partial differential equations (PDEs) are not usually available in analytic form and need to be approximated by suitable *high-fidelity* methods, such as the Finite Element Method (FEM) [35, 37]. The latter usually entails a suitable spatial discretization of the (bounded, compact) computational domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$, regulated by the step size $h > 0$ and yielding a set of N_h degrees of freedom, that in some cases might correspond to the vertices of the elements providing the domain discretization. *High-fidelity* methods are usually referred to as full order models (FOMs) as they provide very accurate solutions, however, resulting in computationally demanding strategies in terms of either time or resources. Within this work, we focus on a parametric setting, where in general the PDE solution u depends not only on the spatial coordinate $x \in \Omega$ and the time variable $t \in \mathcal{T} = [0, T]$, but also on a parameter vector $\boldsymbol{\mu} \in \mathcal{P}$ — being the parameter space $\mathcal{P} \subset \mathbb{R}^p$ a compact set — namely $u = u(x, \boldsymbol{\mu}, t)$. Once the problem has been discretized in space, we aim at exploring the solution manifold $\mathcal{S}_{N_h} = \{\mathbf{u}(\boldsymbol{\mu}, t) = [u(x_i, \boldsymbol{\mu}, t)]_{i=1}^{N_h} \in \mathbb{R}^{N_h} : (\boldsymbol{\mu}, t) \in \mathcal{P} \times \mathcal{T}\}$, evaluating the problem solution in multiple scenarios, for different parameter values. To carry out this task efficiently, as well as to tackle other *multi-query* tasks such as those involving uncertainty quantification and to perform real-time numerical simulations, FOMs must be replaced by efficient and reliable reduced order models (ROMs), a wide class of strategies providing very efficient results yet retaining an adequate representation of the solution manifold $\mathcal{S}_{N_h}$.

Linear projection-based ROMs, such as the reduced basis (RB) method relying on either greedy algorithms or the Proper Orthogonal Decomposition (POD) to build a low-dimensional linear trial subspace, are widely used in the context of parametrized PDEs. Usually relying on a (Petrov-)Galerkin projection to generate the corresponding ROM by enforcing at the reduced order level, the physical constraints expressed by the FOM, these strategies feature, however, several drawbacks, especially when dealing with time-dependent, nonlinear, and nonaffine problems, ultimately requiring suitable *hyper-reduction* strategies such as the empirical interpolation method (EIM) [1, 9, 36] or the Discrete EIM (DEIM, [4]). Despite being very general, and widely applied, hyper-reduction techniques usually feature an intrusive nature, required to handle algebraic arrays extracted from the FOM, ultimately resulting in overwhelming computational costs when dealing with nonlinear time-dependent parametrized PDEs.

To overcome these limitations, data-driven deep learning-based ROMs (DL-ROMs) were recently proposed in [10, 12] and similar works [27, 30, 34, 45] to exploit the power of DNNs to both perform dimensionality reduction of a set of high-dimensional snapshots data (obtained by sampling the solution manifold) and learn parameter-to-solution maps nonintrusively. Unfortunately, these techniques require to train complex architectures and might become unfeasible to train as soon as the FOM dimension N_h increases, suffering from the *curse of dimensionality* in their *vanilla* version. To counter this issue, POD-DL-ROMs were then introduced in [16], leveraging on the power of DL-ROMs and the physically consistent dimensionality reduction achieved through POD, and then training a DL-ROM network using FOM data projected on a (possibly, large dimensional) POD space: overall, POD-DL-ROMs are capable of lower training efforts in terms of both memory storage and computational time. The POD-DL-ROM paradigm has been tested against several problems, showing remarkable approximation capabilities in the numerical simulation of, e.g., fluid flows and fluid-structure interaction problems [15, 16], cardiac electrophysiology [18], and micro-electromechanical systems [14] among others.

However, a thorough numerical analysis of the POD-DL-ROM technique — connecting, e.g., the complexity of the NN architectures involved in a POD-DL-ROM, the sampling error entailed by the selection of training data, the POD error generated while projecting those data onto a POD space, with the overall accuracy of the computed solution — is still lacking. Within this work, we aim at addressing these questions in light of a solid theoretical analysis, providing general error estimates for the POD-DL-ROM technique, assessing their validity in a series of numerical experiments involving different parametrized problems.

1.1 Literature review and existing results

Thanks to the flourishing and rapidly evolving literature of Approximation Theory, many deep learning-based approaches to reduced order modeling are now being justified with rigorous theoretical results and error estimates. The majority of these are grounded on a notorious result by Yarotski (2017) [42], which we report below. In what follows, we use the acronym “ReLU” for the rectified linear unit activation, i.e., the scalar map $x \to \max\{x, 0\}$.

Theorem 1 (Yarotski [42]) *Let $b \in \mathbb{N}$, $b \geq 1$ and $0 < \varepsilon < 1/2$. Any $f \in W^{s,+\infty}([0, 1]^b)$ can be approximated uniformly with an error of at most ε by a ReLU Deep Neural Network (DNN) having at most $c \log(1/\varepsilon)$ layers and $c\varepsilon^{-b/s} \log(1/\varepsilon)$ weights, where $c = c(s, b, f)$ is a constant.*

Indeed, this result and its subsequent generalizations, see, e.g., [21, 43], constitute the foundation of many recent works, for instance:

(i) In [10], the authors exploited these results to formulate an error analysis for general DL-ROMs. However, their analysis is limited to the time-independent case and does not resolve the *curse of dimensionality*, as it binds the complexity of DL-ROMs linearly with the FOM dimension N_h;

(ii) Yarotski’s Theorem was also considered in [11], where the authors investigated the approximation capabilities of Convolutional Neural Networks (CNNs), suggesting a strong connection between these architectures and the Fourier transform;

(iii) Similarly, the results in [42] are fundamental for the derivation of the approximation bounds reported in [26], which, instead, concern the DeepONet paradigm, an approach first proposed by Lu et al. in [28];

(iv) Finally, Yarotski’s Theorem and its generalizations were also employed to derive approximation bounds for deep learning-based ROM strategies that couple POD and feedforward neural networks, see, e.g., [2].

Here, we aim at proposing a similar analysis for POD-DL-ROMs, emphasizing the main differences between this approach and the existing literature.

1.2 Overall idea and paper structure

We analyze the overall approximation error entailed by the use of POD-DL-ROMs when dealing with the solution of both linear and nonlinear time-dependent parametrized PDEs by highlighting two separate error contributions: one, coming from the preliminary dimensionality reduction obtained through POD, and one entailed by the use of neural networks.

In brief, the idea goes as follows. First, we show that in the finite data regime, the overall error of a POD-DL-ROM, $\mathcal{E}_R$, can be decomposed as

$$\mathcal{E}_R \leq \mathcal{E}_S + \mathcal{E}_{POD} + \mathcal{E}_{NN},$$

where $\mathcal{E}_S$ is the sampling error, $\mathcal{E}_{POD}$ is the POD projection error, and $\mathcal{E}_{NN}$ is the approximation error of the neural network model in the DL-ROM pipeline. Then, we address each of the three contributions separately.

For the first two, we rely on classical arguments that bind together the discrete and the continuous formulation of POD, see, e.g., [9, 36], ultimately showing that the sampling error vanishes as a function of the sample size, while $\mathcal{E}_{POD}$ is uniquely characterized by the eigenvalue decay of the data correlation matrix. In this sense, our analysis is strictly related to the one proposed in [26]. To study the neural network error, instead, we consider a specific construction that reflects the general philosophy of DL-ROM techniques. More precisely, we emphasize the fact that POD-DL-ROMs use a neural network architecture that is obtained through the combination of two networks: a feature map, ϕ, which captures the roughness in the parameter-to-solution operator, and a smoother decoder Ψ. In particular, we base our proof on a generalization of Yarotski's Theorem, due to Gühring et al. [21], which, during the composition step, allows us to keep the approximation error under control. For the sake of better readability, we report the latter result below.

Theorem 2 (Gühring et al. [20]) *Let* $b, s \in \mathbb{N}$, *with* $b \geq 1$, $s \geq 2$ *and* $n \in \{0, 1\}$. *For any tolerance* $0 < \varepsilon < 1/2$ *and any* $f \in W^{s,+\infty}([0,1]^b)$, *there exists and a ReLU DNN* Ψ *having at most* $c \log(1/\varepsilon)$ *layers and* $c\varepsilon^{-b/(n-s)} \log(1/\varepsilon)$ *weights, where* $c = c(s, b, f, n)$ *is a constant, such that*

$$\|\Psi - f\|_{W^{n,+\infty}([0,1]^b)} < \varepsilon.$$

All of this ultimately allows us to characterize the accuracy of POD-DL-ROMs in terms of their complexity, providing explicit error bounds that we later compare with the existing literature and verify numerically.

The paper is organized as follows: in Section 2, we formulate the problem, describing rigorously the POD-DL-ROM approach and the reducibility measures for the framework at hand; Section 3 contains the main results of this work, namely the *error* *decomposition* formula, a *lower bound* result and an *upper bound* result for the approximation error. Section 4 then demonstrates the advantages of POD-DL-ROMs when compared to similar deep learning-based frameworks, such as, e.g., POD+DNN and DeepONets. Finally, a series of numerical experiments that validate the theoretical analysis is shown in Section 5, while the last section draws some conclusions and summarizes possible further developments.

2 An overview of the POD-DL-ROM technique

POD-DL-ROMs provide a general-purpose ROM approach combining a data dimensionality reduction obtained through POD with the DL-ROM approach. After introducing the general class of problems we deal with, we overview the main building blocks of the POD-DL-ROM technique. For further details regarding, e.g., detailed algorithms for the offline (or training) and the online query (or testing) stages, the interested reader can refer to, e.g., [16]. An extension of the POD-DL-ROM technique in view of time forecasts of the problem solution out of the training time window has been proposed in [13].

2.1 Problem formulation

Within this work, we consider time-dependent parametric PDEs of the following type

$$\begin{cases} \dfrac{\partial u}{\partial t} + \mathcal{L}(\boldsymbol{\mu})u(\boldsymbol{\mu},t) + \mathcal{N}(u(\boldsymbol{\mu},t),\boldsymbol{\mu}) = f(\boldsymbol{\mu},t), & \text{in } \Omega \times (0,T] \\ \mathcal{B}(\boldsymbol{\mu})u(\boldsymbol{\mu},t) = g(\boldsymbol{\mu},t), & \text{on } \partial\Omega \times (0,T] \\ u(\boldsymbol{\mu},0) = u_0(\boldsymbol{\mu}), & \text{in } \Omega, \end{cases} \tag{1}$$

where

- $u = u(x,\boldsymbol{\mu},t)$ is the PDE solution $\forall x \in \Omega$. Here, we highlight the explicit dependence of u on the time variable $t \in \mathcal{T} = [0,T]$ (for some $T > 0$) and the input parameter vector $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^p$, $\mathcal{P}$ compact;
- $\mathcal{L}$ is a linear operator, whereas $\mathcal{N}$ is a nonlinear operator and $\mathcal{B}$ is the boundary operator; virtually, all these operators might be $\boldsymbol{\mu}$-dependent
- $u_0 = u_0(\boldsymbol{\mu})$ is the initial condition;
- Ω is the (bounded) spatial domain where the problem is set.

Depending on the nature of the problem, the input parameter can refer to either the physical or geometrical properties of the problem at hand. We considered the formulation (2) as a general framework since it describes a wide variety of problems ranging in the fields of engineering, physics, and life sciences, just to make a few examples. Introducing a computational mesh over Ω with mesh size $h > 0$ and a corresponding space discretization of the problem (1) having N_h degrees of freedom (dofs) obtained through, e.g., the finite element method, the finite-dimensional counterpart of problem

(1) provides our FOM and reads as follows:

$$\begin{cases} \mathbf{M}(\boldsymbol{\mu})\dfrac{\partial \mathbf{u}}{\partial t}(\boldsymbol{\mu}, t) + \mathbf{A}(\boldsymbol{\mu})\mathbf{u}(\boldsymbol{\mu}, t) + \mathbf{N}(\mathbf{u}(\boldsymbol{\mu}, t), \boldsymbol{\mu}) = \mathbf{f}(\boldsymbol{\mu}, t), \quad t \in (0, T] \\ \mathbf{u}(\boldsymbol{\mu}, 0) = \mathbf{u}_0(\boldsymbol{\mu}), \end{cases} \tag{2}$$

where $\mathbf{u}(\boldsymbol{\mu}, t) \in \mathbb{R}^{N_h}$ denotes the vector of the N_h dofs of the FOM solution, $\mathbf{M}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ the mass matrix, $\mathbf{A}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ the stiffness matrix, $\mathbf{N}(\cdot, \boldsymbol{\mu}) : \mathbb{R}^{N_h} \to \mathbb{R}^{N_h}$ a nonlinear map, $\mathbf{f}(\boldsymbol{\mu}, t) \in \mathbb{R}^{N_h}$ the source term and $\mathbf{u}_0(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ the initial data. The FOM (2) is then discretized in time, introducing a suitable time-advancing scheme over a partition of $\mathcal{T}$ made by N_t time steps $\{t_k\}_{k=1}^{N_t}$.

To explore efficiently the solution manifold $\mathcal{S}_{N_h} = \{\mathbf{u}(\boldsymbol{\mu}, t) : (\boldsymbol{\mu}, t) \in \mathcal{P} \times \mathcal{T}\}$, we employ the POD-DL-ROM technique, performing a two-step dimensionality reduction: first, POD (realized through randomized SVD) is applied on a set of FOM snapshots; then, a DL-ROM is built to approximate the map between $(\boldsymbol{\mu}, t)$ and the POD generalized coordinates. This latter task can be achieved by relying on two neural network architectures, *(i)* a deep autoencoder — possibly involving convolutional layers — that extracts a set of few, latent coordinates, ultimately representing the reduced order coordinates of the ROM, and *(ii)* a deep feedforward neural network, to learn the map between $(\boldsymbol{\mu}, t)$ and these latent coordinates. Below, we report the main building blocks of a POD-DL-ROM, originally proposed in [16]:

(i) The snapshot matrix for the parameter vectors $\boldsymbol{\mu}_j$, $j = 1, \ldots, N_s$ is collected, thus obtaining $\mathbf{U}_j = [\mathbf{u}(\boldsymbol{\mu}_j, t_k)]_{k=1}^{N_t} \in \mathbb{R}^{N_h \times N_t}$;

(ii) The whole snapshot matrix is obtained stacking $\mathbf{U}_j$, $j = 1, \ldots, N_s$, namely $\mathbf{U} = [\mathbf{U}_j]_{j=1}^{N_s} \in \mathbb{R}^{N_h \times N_{data}}$, where $N_{data} = N_s N_t$;

(iii) A singular value decomposition (SVD) is performed on the snapshot matrix $\mathbf{U}$, and the first N left singular vectors are retained, thus yielding $\mathbf{U} \approx \mathbf{V}\boldsymbol{\Sigma}\mathbf{W}^T$, where $\mathbf{V} \in \mathbb{R}^{N_h \times N}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{N \times N}$ and $\mathbf{W} \in \mathbb{R}^{N \times N_{data}}$. Then, projecting $\mathbf{U}$ on the reduced linear subspace $\mathbf{V} \in \mathbb{R}^{N_h \times N}$, we obtain a snapshot matrix for the POD coefficients $\mathbf{Q} = \mathbf{V}^T\mathbf{U}$;

(iv) The POD coefficient vectors $\boldsymbol{q}(\boldsymbol{\mu}_j, t_k)$, $j = 1, \ldots, N_s$, $k = 1, \ldots, N_t$, obtained from the columns of $\mathbf{Q}$, along with the parameters vector $\boldsymbol{\mu}_j$ and the time instants t_k, are used to train a DL-ROM. This latter consists of a deep autoencoder $\Psi \circ \Psi'$ and a deep feedforward neural network (to which we refer to as *reduced network*) ϕ, defined as follows:

$$\begin{cases} z^{DYN} = \phi(\boldsymbol{\theta}_{DYN}; \boldsymbol{\mu}_j, t_k) \\ z^{ENC} = \Psi'(\boldsymbol{\theta}_{ENC}; \boldsymbol{q}(\boldsymbol{\mu}_j, t_k)) \\ \hat{\boldsymbol{q}} = \Psi(\boldsymbol{\theta}_{DEC}; z^{DYN}(\boldsymbol{\theta}_{DYN}, \boldsymbol{\mu}_j, t_k)), \end{cases}$$

where ϕ, Ψ', Ψ are the reduced network, the encoder and the decoder, respectively, while $\boldsymbol{\theta}_{DYN}, \boldsymbol{\theta}_{ENC}, and \boldsymbol{\theta}_{DEC}$ are their corresponding neural network weights and biases (they are omitted, hereon, for the sake of readability). The three networks are trained according to the *per-example* loss function below,

$$\mathcal{L}_{supervised} = \omega_N \mathcal{L}_N + \omega_n \mathcal{L}_n,$$

where n denotes the latent dimension of the architecture,

$$\mathcal{L}_N = \sum_{j=1}^{N_s} \sum_{k=1}^{N_t} \|\hat{\boldsymbol{q}}(\boldsymbol{\mu}_j, t_k) - \boldsymbol{q}(\boldsymbol{\mu}_j, t_k)\|^2,$$
$$\mathcal{L}_n = \sum_{j=1}^{N_s} \sum_{k=1}^{N_t} \|\boldsymbol{z}^{ENC}(\boldsymbol{\mu}_j, t_k) - \boldsymbol{z}^{DYN}(\boldsymbol{\mu}_j, t_k)\|^2,$$

while $\omega_N, \omega_n > 0$ are suitable hyperparameters having the purpose to properly balance the two contributions. As a matter of notation, from hereon, we equip any finite-dimensional space $\mathbb{R}^b$ (for some $b \in \mathbb{N}$) with the ℓ^2 norm: thus, unless otherwise stated, we define $\|\cdot\| := \|\cdot\|_2$. It is worth to remark that $\mathcal{L}_N$ penalizes high reconstruction errors and $\mathcal{L}_n$ ensures a good representation in the latent space.

Recalling that $(\boldsymbol{\mu}, t) \to \mathbf{V}\hat{\boldsymbol{q}}(\boldsymbol{\mu}, t) \approx \mathbf{u}(\boldsymbol{\mu}, t)$ provides the POD-DL-ROM approximation, the objective of the present work is to characterize the relative error

$$\mathcal{E}_R := \left(\int_{\mathcal{P}\times\mathcal{T}} \frac{\|\mathbf{u}(\boldsymbol{\mu}, t) - \mathbf{V}\hat{\boldsymbol{q}}(\boldsymbol{\mu}, t)\|^2}{\|\mathbf{u}(\boldsymbol{\mu}, t)\|^2} d(\boldsymbol{\mu}, t) \right)^{1/2}$$

in terms of the POD-DL-ROMs complexity. Here, we choose to focus on analyzing $\mathcal{E}_R$ since it is a common measure for the accuracy in the ROM literature. Moreover, we highlight that the entire workflow yielding the error estimate, we propose in this work is only based on the approximation error. Indeed, in the wake of analogous works, e.g., [7, 26, 31], our analysis disregards the contribution of the optimization error, which is an additional source of error stemming from the stochastic nature of the neural network training, related to, e.g., the stochastic gradient descent algorithms and random initializations of the networks weights and biases. We remark that a thorough analysis of the training stage and the optimization algorithm employed to compute the optimal parameters is beyond the purpose of the present work but it may constitute a promising direction for future research. Moreover, we mention that the extension to more general vector energy norms including the contribution of symmetric positive definite mass matrices to define the counterpart of norms in functional spaces like, e.g., $L^2(\Omega)$ or $H^1(\Omega)$, is also straightforward and is not considered here for the sake of simplicity.

2.2 POD: from the discrete to the continuous formulation

Before proceeding towards the thorough analysis of $\mathcal{E}_R$, we have to appropriately define the working setting, which depends on the linear dimensionality reduction. First, we notice that even though within the POD-DL-ROM pipeline, we computed the POD matrix $\mathbf{V}$ through the (randomized) SVD algorithm, thus using a fully *data-driven* procedure that employs a set of training data, the relative error $\mathcal{E}_R$ aims at measuring the *approximation* capabilities over the entire time-parameter space $\mathcal{P} \times \mathcal{T}$, taking advantage of a continuous formulation. Within this section, we aim at filling the gap between the discrete and the continuous formulation of POD, highlighting links and bounds, focusing initially only on the source of error coming from the projection phase, rather than directly considering $\mathcal{E}_R$: this allows us to set the ground upon which the more complex approximation results of POD-DL-ROM are based.

We start by considering the $(\mathcal{P} \times \mathcal{T})$-discrete setting, and the fact that $\mathbf{V}$ results from the solution of a minimization problem; indeed, denoting by

$$\mathbf{K} = \frac{|\mathcal{P} \times \mathcal{T}|}{N_{data}} \mathbf{U}\mathbf{U}^T \in \mathbb{R}^{N_h \times N_h}$$

the (discrete) correlation matrix and by σ_k^2 its eigenvalues, it holds that [36]

$$\begin{aligned}\sum_{k>N} \sigma_k^2 &= \frac{|\mathcal{P} \times \mathcal{T}|}{N_{data}} \sum_{j=1}^{N_{data}} \|\mathbf{u}_j - \mathbf{V}\mathbf{V}^T \mathbf{u}_j\|^2 \\ &= \min_{\mathbf{W} \in \mathbb{R}^{N_h \times N} : \mathbf{W}^T\mathbf{W} = \boldsymbol{I}} \frac{|\mathcal{P} \times \mathcal{T}|}{N_{data}} \sum_{j=1}^{N_{data}} \|\mathbf{u}_j - \mathbf{W}\mathbf{W}^T \mathbf{u}_j\|^2,\end{aligned}$$

where N is the chosen POD dimension and $\mathbf{u}_j$ is the solution vector that corresponds to the tuple $(\boldsymbol{\mu}, t)_j$. We can proceed analogously for the $(\mathcal{P} \times \mathcal{T})$-continuous setting, by considering

$$\mathbf{K}_\infty = \int_{\mathcal{P} \times \mathcal{T}} \mathbf{u}(\boldsymbol{\mu}, t)\mathbf{u}(\boldsymbol{\mu}, t)^T d(\boldsymbol{\mu}, t) \in \mathbb{R}^{N_h \times N_h} \tag{3}$$

as the (continuous) correlation matrix and denoting by $\sigma_{k,\infty}^2$ its eigenvalues; similarly, we can prove that there exists an optimal rank-N matrix $\mathbf{V}_\infty \in \mathbb{R}^{N_h \times N}$ such that

$$\begin{aligned}\sum_{k>N} \sigma_{k,\infty}^2 &= \int_{\mathcal{P} \times \mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu}, t) - \mathbf{V}_\infty \mathbf{V}_\infty^T \mathbf{u}(\boldsymbol{\mu}, t)\|^2 d(\boldsymbol{\mu}, t) \\ &= \min_{\mathbf{W} \in \mathbb{R}^{N_h \times N} : \mathbf{W}^T\mathbf{W} = \boldsymbol{I}} \int_{\mathcal{P} \times \mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu}, t) - \mathbf{W}\mathbf{W}^T \mathbf{u}(\boldsymbol{\mu}, t)\|^2 d(\boldsymbol{\mu}, t)\end{aligned}$$

From the considerations above, we can infer that

$$\sum_{k>N} \sigma_{k,\infty}^2 = \int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu}, t) - \mathbf{V}_\infty \mathbf{V}_\infty^T \mathbf{u}(\boldsymbol{\mu}, t)\|^2 d(\boldsymbol{\mu}, t)$$
$$\leq \int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu}, t) - \mathbf{V}\mathbf{V}^T \mathbf{u}(\boldsymbol{\mu}, t)\|^2 d(\boldsymbol{\mu}, t);$$

from the inequality above, we can remark that the *data-driven* POD matrix $\mathbf{V}$ is not optimal for the continuous formulation, which stems from the hypothesis of having infinite data samples while being the best orthogonal matrix in terms of explained variability with respect the training data at hand. In other words, even though $\mathbf{V}$ is optimal for the training data, we have no guarantee that it is optimal for the test data, too; however, since in practice, we are not able to obtain the matrix $\mathbf{V}_\infty$, we must necessarily rely on $\mathbf{V}$ also in the online testing phase.

Finally, we show how the discrete and the continuous POD formulations are related: indeed, denoting by $[\cdot]_i$ the i-th entry of a vector, and extending this notation to matrices, we have that $\forall k, l = 1, \ldots, N_h$

$$[\mathbf{K}_\infty - \mathbf{K}]_{kl} = \int_{\mathcal{P}\times\mathcal{T}} [\mathbf{u}]_k [\mathbf{u}]_l d(\boldsymbol{\mu}, t) - \frac{|\mathcal{P}\times\mathcal{T}|}{N_{data}} \sum_{j=1}^{N_{data}} [\mathbf{u}_j]_k [\mathbf{u}_j]_l,$$

recalling that $\mathbf{u}_j$ is the solution vector that corresponds to the tuple $(\boldsymbol{\mu}, t)_j$. Upon requiring integrability (easily verified for non-trivial bounded solutions), we can use the strong law of large numbers [23] and obtain $[\mathbf{K} - \mathbf{K}_\infty]_{kl} \xrightarrow{a.s.} 0$ as $N_s, N_t \to \infty$, $\forall k, l = 1, \ldots, N_h$, which implies that $\|\mathbf{K} - \mathbf{K}_\infty\|_1 \xrightarrow{a.s.} 0$, being $\|\mathbf{Z}\|_1$ any 1-norm of the squared matrix $\mathbf{Z}$. By employing Bauer-Fike's theorem [37] with the 1-norm, we can state that, upon ordering, for any $\sigma_{k,\infty}^2$, there exists σ_k^2 belonging to the spectrum of $\mathbf{K}$ such that

$$|\sigma_k^2 - \sigma_{k,\infty}^2| \leq K_1(\mathbf{X}) \|\mathbf{K} - \mathbf{K}_\infty\|_1, \qquad \forall k = 1, \ldots, N_h$$

where $\mathbf{X}$ is the matrix collecting the right eigenvectors of $\mathbf{K}$, and $K_1(\mathbf{X})$ denotes its condition number. Thus, we can conclude that, setting N as the POD dimension, it holds that

$$\sum_{k>N} \sigma_k^2 \xrightarrow{a.s.} \sum_{k>N} \sigma_{k,\infty}^2, \qquad N_s, N_t \to \infty.$$

2.3 An overlook over the reducibility measures for POD-DL-ROMs

POD-DL-ROMs couple POD, for the sake of a preliminary dimensionality reduction, with an autoencoder-based architecture to reconstruct the parameter-to-POD-coefficients map. Thus, at first, it is evident that the projection-based nature of the paradigm invokes the definition of a linear reducibility measure to account for the *FOM-to-POD* dimensionality reduction task.

Definition 1 Let $\mathcal{S}_{N_h} = \{\mathbf{u}(\boldsymbol{\mu}, t) \in \mathbb{R}^{N_h} : (\boldsymbol{\mu}, t) \in \mathcal{P} \times \mathcal{T}\}$ be the solution manifold. The linear Kolmogorov N-width of $\mathcal{S}_{N_h}$ is defined as

$$d_N(\mathcal{S}_{N_h}) = \inf_{V_N \subset \mathbb{R}^{N_h} : \dim(V_N) = N} \sup_{\mathbf{u} \in \mathcal{S}_{N_h}} \inf_{\mathbf{v} \in V_N} \|\mathbf{u} - \mathbf{v}\|.$$

It is worth to notice that the linear Kolmogorov N-width is strictly related to the eigenvalues decay of the correlation matrix $\mathbf{K}_\infty \in \mathbb{R}^{N_h \times N_h}$. In fact, following the same notation of Section 2.2, we have that:

$$\begin{aligned}
\sqrt{\sum_{k>N} \sigma_{k,\infty}^2} &= \left(\int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu}, t) - \mathbf{V}_\infty \mathbf{V}_\infty^T \mathbf{u}(\boldsymbol{\mu}, t)\|^2 d(\boldsymbol{\mu}, t) \right)^{1/2} \\
&\leq \left(\int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu}, t) - \mathbf{W}\mathbf{W}^T \mathbf{u}(\boldsymbol{\mu}, t)\|^2 d(\boldsymbol{\mu}, t) \right)^{1/2} \\
&\leq |\mathcal{P} \times \mathcal{T}|^{1/2} \sup_{(\boldsymbol{\mu}, t) \in \mathcal{P}\times\mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu}, t) - \mathbf{W}\mathbf{W}^T \mathbf{u}(\boldsymbol{\mu}, t)\|^2
\end{aligned}$$

for any $\mathbf{W} \in \mathbb{R}^{N_h \times N}$; thus,

$$\sqrt{\sum_{k>N} \sigma_{k,\infty}^2} \leq |\mathcal{P} \times \mathcal{T}|^{1/2} d_N(\mathcal{S}_{N_h}).$$

The above relationship shows that the eigenvalue decay is an alternative (and more practical) measure of reducibility, with respect to a weaker norm. However, notice that in practice, we can only approximate the quantity $\sum_{k>N} \sigma_{k,\infty}^2 \approx \sum_{k>N} \sigma_k^2$, which is consistent with the theory thanks to the convergence result presented in Section 2.2.

The autoencoder-based architecture of a POD-DL-ROM introduces a second level of dimensionality reduction, which operates a further compression of the information coming from the parameter-to-POD-coefficients map $\mathcal{Q} : (\boldsymbol{\mu}, t) \to \mathbf{V}^T \mathbf{u}(\boldsymbol{\mu}, t)$. The nonlinear nature of the dimensionality reduction performed through the autoencoder $\Psi \circ \Psi'$ (being Ψ', Ψ the *encoder* and the *decoder*, respectively) induces a nonlinear analog of the Kolmogorov n-width [8].

Definition 2 The nonlinear Kolmogorov n-width of the reduced manifold $\mathcal{S}_N = \{\boldsymbol{q}(\boldsymbol{\mu}, t) = \mathbf{V}^T \mathbf{u}(\boldsymbol{\mu}, t) \in \mathbb{R}^N : (\boldsymbol{\mu}, t) \in \mathcal{P} \times \mathcal{T}\}$ is defined as

$$\delta_n(\mathcal{S}_N) = \inf_{\substack{\Psi \in C(\mathbb{R}^N, \mathbb{R}^n) \\ \Psi' \in C(\mathbb{R}^n, \mathbb{R}^N)}} \sup_{\mathbf{u} \in \mathcal{S}_{N_h}} \|\mathbf{u} - \Psi(\Psi'(\mathbf{u}))\|.$$

Now, to deal with nonlinear approximation methods, we state another fundamental definition upon which the main results of this work are based.

Definition 3 The reduced manifold $\mathcal{S}_N = \{\boldsymbol{q}(\boldsymbol{\mu}, t) = \mathbf{V}^T \mathbf{u}(\boldsymbol{\mu}, t) \in \mathbb{R}^N : (\boldsymbol{\mu}, t) \in \mathcal{P} \times \mathcal{T}\}$ enjoys the perfect embedding assumption with regularity s, s' if the infimum in Definition 2 is attained, namely there exist $\Psi_* \in C^s(\mathbb{R}^N, \mathbb{R}^n)$, $\Psi'_* \in C^{s'}(\mathbb{R}^n, \mathbb{R}^N)$ such that

$$\Psi_*(\Psi'_*(\boldsymbol{q}(\boldsymbol{\mu}, t)) = \boldsymbol{q}(\boldsymbol{\mu}, t) \qquad \forall (\boldsymbol{\mu}, t) \in \mathcal{P} \times \mathcal{T}.$$

In conclusion, as we did with the POD dimension N, we need to characterize the latent dimension n with a practical criterion. To do that, an extension of *Theorem 3* provided in [10] shows that if the parameter-to-solution map $\mathcal{G} : (\boldsymbol{\mu}, t) \to \mathbf{u}(\boldsymbol{\mu}, t)$ and thus the parameter-to-POD-coefficients map $\mathcal{Q} : (\boldsymbol{\mu}, t) \to \mathbf{V}^T\mathbf{u}(\boldsymbol{\mu}, t)$ are Lipschitz-continuous, there exists $n \leq 2p + 3$ such that $\delta_n(\mathcal{S}_N) = 0$.

3 Main results

Before stating the main result of this work, namely an *upper bound* result, that concerns only POD-DL-ROMs, we make some preliminary reasoning that, instead, applies to any POD+DNN approach, i.e., we do not constrain the neural network $\hat{\boldsymbol{q}}$, that approximates the parameter-to-POD-coefficients map, to be a DL-ROM. For this purpose, we briefly recall that the POD+DNN technique involves the reconstruction of the parameter-to-solution map through the approximation $(\boldsymbol{\mu}, t) \mapsto \mathbf{V}\hat{\boldsymbol{q}}(\boldsymbol{\mu}, t) \approx \mathbf{u}(\boldsymbol{\mu}, t)$, where $\hat{\boldsymbol{q}}$ is a generic (possibly dense) neural network.

In particular, we start by characterizing $\mathcal{E}_R$ through an error decomposition formula, that enables us to describe the various error contributions and formulate possible strategies to control them. Secondly, we state a *lower bound* result, that highlights how, regardless of the architecture of neural network $\hat{\boldsymbol{q}}$, the relative error $\mathcal{E}_R$ can be bounded from below by a quantity depending on the POD projection. Then, we move to our *upper bound* result, where we quantify how complex a POD-DL-ROM should be in order to achieve a specific bound on the relative error $\mathcal{E}_R$.

We initially remark that the computation of the error $\mathcal{E}_R$ and other related quantities hinges upon the evaluation of complex integrals, possibly in high-dimensional spaces, which can be effectively handled through Monte Carlo methods. In this respect, we shall make the following assumptions, which we assume to hold true hereon.

Assumption 1 (Sampling criterion) *Let $p > 0$, assume that $\mathcal{P} \subset \mathbb{R}^p$ is compact and denote $\mathcal{T} = [0, T]$ for some $T > 0$. We assume that the training (and testing) snapshots are sampled uniformly and iid in the parameter space, $\boldsymbol{\mu} \sim \mathcal{U}(\mathcal{P})$, while a uniform grid is employed for the time variable, $t \in \{\Delta t, 2\Delta t, ..., N_t\Delta t\}$, where $N_t \in \mathbb{N}_{\geq 2}$ and $\Delta t = T/N_t$.*

Assumption 2 (Parameter-to-solution map) *Let $\mathcal{G} : \mathcal{P} \times \mathcal{T} \to \mathbb{R}^{N_h}$ be the parameter-to-solution map, mapping $(\boldsymbol{\mu}, t) \mapsto \mathbf{u}(\boldsymbol{\mu}, t)$. We assume that*

i) $m = \operatorname{ess\,inf}_{(\boldsymbol{\mu},t)\in\mathcal{P}\times\mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu}, t)\| > 0$, $M = \operatorname{ess\,sup}_{(\boldsymbol{\mu},t)\in\mathcal{P}\times\mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu}, t)\| < \infty$;
ii) $\mathcal{G}$ is Lipschitz-continuous with constant $L > 0$.

From these assumptions, one can easily derive a couple of auxiliary results, which will be of practical interest in the remainder, and are reported below; for the sake of brevity, their proofs are postponed to Appendix A.

Proposition 1 *Let* $f \in L^2(\mathcal{P} \times \mathcal{T})$*. Under* Assumption 1*, one has*

$$\mathbb{E}\left|\int_{\mathcal{P}\times\mathcal{T}} f(\boldsymbol{\mu}, t)d(\boldsymbol{\mu}, t) - \frac{|\mathcal{P} \times \mathcal{T}|}{N_{data}} \sum_{i=1}^{N_t} \sum_{j=1}^{N_s} f(\boldsymbol{\mu}_j, t_i)\right| \leq O(N_s^{-1/2} + N_t^{-1}).$$

where the expectation is taken across all the possible realizations of the data sampling procedure.

Proposition 2 *Under the* Assumption 2*, define* $w(\boldsymbol{\mu}, t) = \|\mathbf{u}(\boldsymbol{\mu}, t)\|^{-2}$*. Then,*

$$\|\cdot\|_{L^2_w} = \left(\int_{\mathcal{P}\times\mathcal{T}} \|\cdot\|^2 w(\boldsymbol{\mu}, t)d(\boldsymbol{\mu}, t)\right)^{1/2}$$

is a norm in $L^2(\mathcal{P} \times \mathcal{T}; \mathbb{R}^{N_h})$*.*

3.1 The error decomposition formula

In the following, we state an error decomposition formula that is valid for any POD+DNN approach — and, in particular, for our POD-DL-ROM strategy. Given the more general nature of this result, its formulation is therefore not restricted to the technique at hand.

Theorem 1 *Let* $\mathcal{G} : (\boldsymbol{\mu}, t) \mapsto \mathbf{u}(\boldsymbol{\mu}, t)$ *for any* $(\boldsymbol{\mu}, t) \in \mathcal{P} \times \mathcal{T}$ *be the parameter-to-solution map. Consider a POD+DNN approximation of* $\mathcal{G}$ *as* $\mathcal{G}(\boldsymbol{\mu}, t) \approx \mathbf{V}\hat{\boldsymbol{q}}$*, where* $\hat{\boldsymbol{q}} : \mathbb{R}^{p+1} \to \mathbb{R}^N$ *is a neural network trained over a given training set made by a collection of input parameters* $(\boldsymbol{\mu}_i, t_i)_{i=1}^{N_{data}}$ *and the corresponding snapshot matrix* $\mathbf{U} \in \mathbb{R}^{N_h \times N_{data}}$*, while* $\mathbf{V} \in \mathbb{R}^{N_h \times N}$ *is the POD projection matrix. Then, under the* Assumptions 1 and 2*, we have*

$$\mathcal{E}_R \leq \mathcal{E}_S + \mathcal{E}_{POD} + \mathcal{E}_{NN}, \tag{4}$$

where

- $\mathcal{E}_S = \mathcal{E}_S(\mathcal{G}, \{(\boldsymbol{\mu}_i, t_i)_{i=1}^{N_{data}}\}, N)$ *is the sampling error, that satisfies* $\mathcal{E}_S \xrightarrow{a.s.} 0$ *as* $N_s, N_t \to \infty$ *and* $\mathbb{E}[\mathcal{E}_S] = O(N_s^{-1/4} + N_t^{1/2})$*;*
- $\mathcal{E}_{POD} = \mathcal{E}_{POD}(\mathcal{G}, \{(\boldsymbol{\mu}_i, t_i)_{i=1}^{N_{data}}\}, N)$ *is the POD projection error, that satisfies* $\mathcal{E}_{POD} \xrightarrow{a.s.} \mathcal{E}_{POD,\infty}$ *as* $N_s, N_t \to \infty$*, where* $\mathcal{E}_{POD,\infty} = \mathcal{E}_{POD,\infty}(\mathcal{G}, N)$ *is independent of the sampling criterion;*
- $\mathcal{E}_{NN} = \mathcal{E}_{NN}(\mathcal{G}, N, \hat{\boldsymbol{q}})$ *is the approximation error of the neural network, which is arbitrarily low depending of the approximation capabilities of the network* $\hat{\boldsymbol{q}}$*.*

Proof By means of the triangular inequality, we obtain

$$\mathcal{E}_R = \left(\int_{\mathcal{P}\times\mathcal{T}} \frac{\|\mathbf{u}(\boldsymbol{\mu},t) - \mathbf{V}\hat{\boldsymbol{q}}(\boldsymbol{\mu},t)\|^2}{\|\mathbf{u}(\boldsymbol{\mu},t)\|^2} d(\boldsymbol{\mu},t)\right)^{1/2} = \|\mathbf{u}(\boldsymbol{\mu},t) - \mathbf{V}\hat{\boldsymbol{q}}(\boldsymbol{\mu},t)\|_{L^2_w}$$
$$\leq \|\mathbf{u}(\boldsymbol{\mu},t) - \mathbf{V}\mathbf{V}^T\mathbf{u}(\boldsymbol{\mu},t)\|_{L^2_w} + \|\mathbf{V}\mathbf{V}^T\mathbf{u}(\boldsymbol{\mu},t) - \mathbf{V}\hat{\boldsymbol{q}}(\boldsymbol{\mu},t)\|_{L^2_w}. \quad (5)$$

According to the notation of Section 2, let $\boldsymbol{q}(\boldsymbol{\mu},t) := \mathbf{V}^T\mathbf{u}(\boldsymbol{\mu},t)$. We define

$$\mathcal{E}_{NN} := \left(\int_{\mathcal{P}\times\mathcal{T}} \frac{\|\mathbf{V}\boldsymbol{q}(\boldsymbol{\mu},t) - \mathbf{V}\hat{\boldsymbol{q}}(\boldsymbol{\mu},t)\|^2}{\|\mathbf{u}(\boldsymbol{\mu},t)\|^2} d(\boldsymbol{\mu},t)\right)^{1/2}.$$

Notice that, of the two terms at the right-hand-side of Eq. (5), $\mathcal{E}_{NN}$ corresponds to the second one; in particular, it corresponds to the only error component actually depending on the neural network approximation. Moreover, we can bound the remaining term in (5) as

$$\|\mathbf{u}(\boldsymbol{\mu},t) - \mathbf{V}\mathbf{V}^T\mathbf{u}(\boldsymbol{\mu},t)\|_{L^2_w} \leq m^{-1}\left(\int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu},t) - \mathbf{V}\mathbf{V}^T\mathbf{u}(\boldsymbol{\mu},t)\|^2 d(\boldsymbol{\mu},t)\right)^{1/2}.$$

Let now $\mathbf{K} = |\mathcal{P}\times\mathcal{T}|N_{data}^{-1}\mathbf{U}\mathbf{U}^T \in \mathbb{R}^{N_h\times N_h}$ be the discrete correlation matrix and let σ_k^2 be its eigenvalues. By employing the triangular inequality and the trivial inequality $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for $a, b \geq 0$,

$$\begin{aligned} &m^{-1}\left(\int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu},t) - \mathbf{V}\mathbf{V}^T\mathbf{u}(\boldsymbol{\mu},t)\|^2 d(\boldsymbol{\mu},t)\right)^{1/2} \leq \\ \leq &m^{-1}\left(\int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu},t) - \mathbf{V}\mathbf{V}^T\mathbf{u}(\boldsymbol{\mu},t)\|^2 d(\boldsymbol{\mu},t) - \sum_{k>N}\sigma_k^2 + \sum_{k>N}\sigma_k^2\right)^{1/2} \leq \\ \leq &m^{-1}\left(\left|\int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu},t) - \mathbf{V}\mathbf{V}^T\mathbf{u}(\boldsymbol{\mu},t)\|^2 d(\boldsymbol{\mu},t) - \sum_{k>N}\sigma_k^2\right| + \sum_{k>N}\sigma_k^2\right)^{1/2} \leq \\ \leq &m^{-1}\left|\int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu},t) - \mathbf{V}\mathbf{V}^T\mathbf{u}(\boldsymbol{\mu},t)\|^2 d(\boldsymbol{\mu},t) - \sum_{k>N}\sigma_k^2\right|^{1/2} + m^{-1}\sqrt{\sum_{k>N}\sigma_k^2}. \end{aligned}$$

In light of this, we define the sampling error as

$$\mathcal{E}_S := m^{-1}\left|\int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu},t) - \mathbf{V}\mathbf{V}^T\mathbf{u}(\boldsymbol{\mu},t)\|^2 d(\boldsymbol{\mu},t) - \sum_{k>N}\sigma_k^2\right|^{1/2},$$

and the POD error as

$$\mathcal{E}_{POD} := m^{-1}\sqrt{\sum_{k>N}\sigma_k^2}.$$

Thus, we obtain the inequality in (4)

$$\mathcal{E}_R \leq \mathcal{E}_S + \mathcal{E}_{POD} + \mathcal{E}_{NN}.$$

In the last part of the proof, we aim at showing the characteristic properties of $\mathcal{E}_S$ and $\mathcal{E}_{POD}$; recalling that

$$\sum_{k>N} \sigma_k^2 = \frac{|\mathcal{P} \times \mathcal{T}|}{N_{data}} \sum_{j=1}^{N_{data}} \|\mathbf{u}_j - \mathbf{V}\mathbf{V}^T \mathbf{u}_j\|^2,$$

we can write the sampling error in a slightly different form

$$\mathcal{E}_S = m^{-1} \left| \int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu}, t) - \mathbf{V}\mathbf{V}^T \mathbf{u}(\boldsymbol{\mu}, t)\|^2 d(\boldsymbol{\mu}, t) - \right. \\ \left. \frac{|\mathcal{P} \times \mathcal{T}|}{N_{data}} \sum_{j=1}^{N_{data}} \|\mathbf{u}_j - \mathbf{V}\mathbf{V}^T \mathbf{u}_j\|^2 \right|^{1/2}.$$

Moreover, thanks to the compactness hypothesis of Assumption 1 and the boundedness hypothesis of Assumption 2, we have that

$$f(\boldsymbol{\mu}, t) = \|\mathbf{u}(\boldsymbol{\mu}, t) - \mathbf{V}\mathbf{V}^T \mathbf{u}(\boldsymbol{\mu}, t)\|^2 \leq M^2 \|\mathbf{I} - \mathbf{V}\mathbf{V}^T\|^2 < +\infty,$$

so that $f \in L^2(\mathcal{P} \times \mathcal{T})$. Thus, by means of Proposition 1, we conclude that $\mathbb{E}[\mathcal{E}_S] = O(N_s^{-1/4} + N_t^{-1/2})$.

Finally, since $\mathcal{E}_S$ and $\mathcal{E}_{POD}$ depend on the number of samples and snapshots in the training set, it is natural to verify their behavior in the *infinite data* limit. Thanks to Assumption 1, by the strong law of large numbers, it is evident that $\mathcal{E}_S \xrightarrow{a.s.} 0$ as $N_s, N_t \to \infty$ and, by means of the results in Section 2,

$$\mathcal{E}_{POD} \xrightarrow{a.s.} \mathcal{E}_{POD,\infty} := m^{-1} \sqrt{\sum_{k>N} \sigma_{k,\infty}^2}, \qquad N_s, N_t \to \infty.$$

□

Remark 1 The convergence rate for $\mathcal{E}_S$ can be improved by modifying Assumption 1. Indeed, Monte Carlo sampling could be replaced by other strategies: for instance, using quasi-Monte Carlo techniques [3, 32], and under suitable regularity assumptions, one has $\mathbb{E}[\mathcal{E}_S] = O((\log(N_s))^{\frac{p+1}{2}} N_s^{-1/2} + N_t^{-1/2})$.

3.2 Lower bound for the relative error

POD-DL-ROMs couple classical projection-based methods such as the POD with deep learning-based techniques that allow to correctly reproduce the nonlinearity of the parameter-to-POD-coefficient map $\mathcal{Q}$. This means that we still need to rely on the linear transformation represented by the POD matrix $\mathbf{V} \in \mathbb{R}^{N_h \times N}$ (or $\mathbf{V}_\infty$ in the *infinite data* limit) to expand the neural network approximation of the POD coefficients.

This last consideration is crucial: indeed, the fact that the POD-DL-ROM technique hinges upon a linear decomposition forces the relative error to still depend on the eigenvalues decay of the correlation matrix; the mentioned dependence is highlighted in the *lower bound* result provided below.

Theorem 2 *Under the same assumptions of* Theorem 1*, we have that*

$$\mathcal{E}_R \geq \frac{m}{M}\mathcal{E}_{POD,\infty},$$

where $\mathcal{E}_{POD,\infty} := m^{-1}\sqrt{\sum_{k>N} \sigma_{k,\infty}^2}$.

Proof We immediately notice that, by optimality of projection coefficients,

$$\mathcal{E}_R = \int_{\mathcal{P}\times\mathcal{T}} \frac{\|\mathbf{u}(\boldsymbol{\mu}, t) - \mathbf{V}\hat{\boldsymbol{q}}(\boldsymbol{\mu}, t)\|}{\|\mathbf{u}(\boldsymbol{\mu}, t)\|} d(\boldsymbol{\mu}, t) \geq \int_{\mathcal{P}\times\mathcal{T}} \frac{\|\mathbf{u}(\boldsymbol{\mu}, t) - \mathbf{V}\mathbf{V}^T\mathbf{u}(\boldsymbol{\mu}, t)\|}{\|\mathbf{u}(\boldsymbol{\mu}, t)\|} d(\boldsymbol{\mu}, t),$$

where we recall that $\mathbf{V}$ is the POD matrix computed via SVD using the discrete formulation and $\mathbf{V}_\infty$ is relative to the continuous formulation. Then,

$$\begin{aligned}
(\mathcal{E}_{POD,\infty})^2 &= m^{-2}\sum_{k>N}\sigma_{k,\infty}^2 \\
&= m^{-2}\int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu}, t) - \mathbf{V}_\infty\mathbf{V}_\infty^T\mathbf{u}(\boldsymbol{\mu}, t)\| d(\boldsymbol{\mu}, t) \\
&\leq m^{-2}\int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu}, t) - \mathbf{V}\mathbf{V}^T\mathbf{u}(\boldsymbol{\mu}, t)\|^2 d(\boldsymbol{\mu}, t) \\
&= m^{-2}\int_{\mathcal{P}\times\mathcal{T}} \frac{\|\mathbf{u}(\boldsymbol{\mu}, t) - \mathbf{V}\mathbf{V}^T\mathbf{u}(\boldsymbol{\mu}, t)\|^2}{\|\mathbf{u}(\boldsymbol{\mu}, t)\|^2}\|\mathbf{u}(\boldsymbol{\mu}, t)\|^2 d(\boldsymbol{\mu}, t) \\
&\leq \frac{M^2}{m^2}\int_{\mathcal{P}\times\mathcal{T}} \frac{\|\mathbf{u}(\boldsymbol{\mu}, t) - \mathbf{V}\mathbf{V}^T\mathbf{u}(\boldsymbol{\mu}, t)\|^2}{\|\mathbf{u}(\boldsymbol{\mu}, t)\|^2} d(\boldsymbol{\mu}, t) \\
&\leq \frac{M^2}{m^2}(\mathcal{E}_R)^2,
\end{aligned}$$

from which the thesis follows. □

Remark 2 The quantity $\mathcal{E}_{POD,\infty}$ reflects the expressivity of the ideal POD basis, that is, the one obtained with an infinite amount of data. As such, it is actually related to the contribute $\mathcal{E}_S + \mathcal{E}_{POD}$ appearing in the error decomposition formula, Theorem 1.

To see this, note that

$$\begin{aligned}
&\mathcal{E}_S + \mathcal{E}_{POD} \geq \\
&\geq m^{-1}\left(\int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu}, t) - \mathbf{V}\mathbf{V}^T\mathbf{u}(\boldsymbol{\mu}, t)\|^2 d(\boldsymbol{\mu}, t)\right)^{1/2} \geq \\
&\geq m^{-1}\left(\min_{\mathbf{W}\in\mathbb{R}^{N_h\times N}:\mathbf{W}^T\mathbf{W}=\boldsymbol{I}} \int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu}, t) - \mathbf{W}\mathbf{W}^T\mathbf{u}(\boldsymbol{\mu}, t)\|^2 d(\boldsymbol{\mu}, t)\right)^{1/2} = \\
&= m^{-1}\left(\int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu}, t) - \mathbf{V}_\infty\mathbf{V}_\infty^T\mathbf{u}(\boldsymbol{\mu}, t)\|^2 d(\boldsymbol{\mu}, t)\right)^{1/2} = \\
&= m^{-1}\sqrt{\sum_{k>N}\sigma_{k,\infty}^2} = \mathcal{E}_{POD,\infty},
\end{aligned}$$

by very definition of $\mathcal{E}_S$, $\mathcal{E}_{POD}$, and $\mathbf{V}_\infty$. It is worth to remark that $\mathcal{E}_{POD,\infty}$ only depends on the eigenstructure of the continuous correlation matrix $\mathbf{K}_\infty$, while it is independent of the data sampling.

Remark 3 Since $\mathbf{V}_\infty$ is not available in practice, we cannot compute $\mathcal{E}_{POD,\infty}$ exactly. In practice, we can use a stricter bound: leveraging on quantities emerging from the proof, we actually employ

$$\tilde{\mathcal{E}}_{POD} := m^{-1}\left(\int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu}, t) - \mathbf{V}\mathbf{V}^T\mathbf{u}(\boldsymbol{\mu}, t)\|^2 d(\boldsymbol{\mu}, t)\right)^{1/2}$$

which we either compute analytically (if possible) or estimate via Monte Carlo.

Theorem 2 states that, no matter how accurate the neural networks approximation is, the relative error $\mathcal{E}_R$ is still bounded from below by the variance neglected by the POD projection. Additionally, the lower bound does not depend on how much data we gather for the supervised training phase. Of note, this is in agreement with the results provided in the analysis of other linear decomposition-based techniques, such as DeepONets [26].

3.3 Upper bound for the relative error

On the basis of the error decomposition and the perfect embedding hypothesis, we aim at providing the main result of this work, which is contained in the Theorem 3 and is endowed with a *constructive* proof founded on the approximation results of [42]. We remark that the present result is only valid for POD-DL-ROMs.

Theorem 3 *Let $\mathcal{G} : (\boldsymbol{\mu}, t) \mapsto \mathbf{u}(\boldsymbol{\mu}, t)$ for any $(\boldsymbol{\mu}, t) \in \mathcal{P} \times \mathcal{T}$ be the parameter-to-solution map and suppose valid* Assumptions 1 and 2. *Let $\delta > 0$ and $0 < \varepsilon < 1$; suppose to have collected $N_{data} = N_{data}(\delta, \varepsilon)$ data samples into the snapshot matrix $\mathbf{U} \in \mathbb{R}^{N_h\times N_{data}}$. Consider the $(\mathcal{P} \times \mathcal{T})$-discrete correlation matrix $\mathbf{K} = |\mathcal{P} \times$*

$\mathcal{T}|N_{data}^{-1}\mathbf{U}\mathbf{U}^T \in \mathbb{R}^{N_h \times N_h}$ and let σ_k^2 be its eigenvalues. Moreover, choose

$$N = \arg\min\left\{ j \in \mathbb{N} : \sum_{k>j} \sigma_k^2 \le \frac{m^2}{9}\varepsilon^2 \right\}.$$

We define the parameter-to-POD-coefficients map $\mathcal{Q} : (\boldsymbol{\mu}, t) \mapsto \boldsymbol{q}(\boldsymbol{\mu}, t)$ as $\mathcal{Q}(\boldsymbol{\mu}, t) = \mathbf{V}^T \mathcal{G}(\boldsymbol{\mu}, t)$ for any $(\boldsymbol{\mu}, t) \in \mathcal{P} \times \mathcal{T}$, where $\mathbf{V} \in \mathbb{R}^{N_h \times N}$ is the reduced rank-N POD matrix computed via SVD. We assume that there exists $n > 0$, $\Psi_* : \mathbb{R}^n \to \mathbb{R}^N$, $\Psi_*' : \mathbb{R}^N \to \mathbb{R}^n$ that are respectively s-times and s'-times differentiable (with $s \gg s' \ge 2$), such that they enjoy the perfect embedding assumption stated in Definition 3, *namely*

$$\Psi_*(\Psi_*'(\boldsymbol{q}(\boldsymbol{\mu}, t)) = \boldsymbol{q}(\boldsymbol{\mu}, t) \qquad \forall (\boldsymbol{\mu}, t) \in \mathcal{P} \times \mathcal{T}.$$

We let

$$C_1 = \sup_{|\boldsymbol{\alpha}| \le s'} \sup_{\mathbf{v} \in \mathbb{R}^N} |D^{\boldsymbol{\alpha}} \Psi_*'(\mathbf{v})| \qquad C_2 = \sup_{|\boldsymbol{\alpha}| \le s} \sup_{\mathbf{w} \in \mathbb{R}^n} |D^{\boldsymbol{\alpha}} \Psi_*(\mathbf{w})|.$$

Then, there exists a constant $c = c(\mathcal{P}, \mathcal{T}, L, C_1, C_2, p, n, s, s')$ and a POD-DL-ROM architecture $\mathbf{V}\hat{\boldsymbol{q}} = \mathbf{V}\psi \circ \phi : \mathbb{R}^{p+1} \to \mathbb{R}^N$ composed of a decoder $\psi : \mathbb{R}^n \to \mathbb{R}^N$ having at most:

- *$L_{n \to N} = c \log(\varepsilon^{-1})$ layers,*
- *$w_{n \to N} = cN\varepsilon^{-n/(s-1)} \log(\varepsilon^{-1})$ active weights,*

and a reduced map $\phi : \mathbb{R}^{p+1} \to \mathbb{R}^n$ having at most:

- *$L_{(p+1) \to n} = c \log(\varepsilon^{-1})$ layers,*
- *$w_{(p+1) \to n} = cn\varepsilon^{-(p+1)} \log(\varepsilon^{-1})$ active weights,*

such that $\mathbb{P}\{\mathcal{E}_R < \varepsilon\} > 1 - \delta$.

Proof We immediately aim at bounding $\mathcal{E}_S = \mathcal{E}_S(N_s, N_t)$; independently of $N \le N_h$, under the Assumption 1, by the Weak Law of Large Numbers [23] we can infer the following statement:

$$\forall \delta > 0, \quad \forall 0 < \varepsilon < 1, \quad \exists N_s, N_t : \mathbb{P}\{\mathcal{E}_S(N_s, N_t) < \varepsilon/3\} > 1 - \delta.$$

Then, we notice that, choosing N as in the theorem statement, we derive

$$\mathcal{E}_{POD} = m^{-1} \sqrt{\sum_{k>N} \sigma_k^2} \le \frac{\varepsilon}{3}.$$

Then, we are left to bound $\mathcal{E}_{NN}$: by means of the Cauchy-Schwarz and the Hölder inequalities, considering that $\|\mathbf{V}\|^2 = 1$, it is trivial that

$$\begin{aligned}
\mathcal{E}_{NN} &= \left(\int_{\mathcal{P}\times\mathcal{T}} \frac{\|\mathbf{V}\boldsymbol{q}(\boldsymbol{\mu},t) - \mathbf{V}\hat{\boldsymbol{q}}(\boldsymbol{\mu},t)\|^2}{\|\mathbf{u}(\boldsymbol{\mu},t)\|^2} d(\boldsymbol{\mu},t) \right)^{1/2} \\
&\leq m^{-1} \left(\int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{V}\boldsymbol{q}(\boldsymbol{\mu},t) - \mathbf{V}\hat{\boldsymbol{q}}(\boldsymbol{\mu},t)\|^2 d(\boldsymbol{\mu},t) \right)^{1/2} \\
&\leq m^{-1} \left(|\mathcal{P}\times\mathcal{T}| \sup_{(\boldsymbol{\mu},t)\in\mathcal{P}\times\mathcal{T}} \|\mathbf{V}\boldsymbol{q}(\boldsymbol{\mu},t) - \mathbf{V}\hat{\boldsymbol{q}}(\boldsymbol{\mu},t)\|^2 \right)^{1/2} \\
&\leq m^{-1} \left(|\mathcal{P}\times\mathcal{T}| \|\mathbf{V}\|^2 \sup_{(\boldsymbol{\mu},t)\in\mathcal{P}\times\mathcal{T}} \|\boldsymbol{q}(\boldsymbol{\mu},t) - \hat{\boldsymbol{q}}(\boldsymbol{\mu},t)\|^2 \right)^{1/2} \\
&= m^{-1} |\mathcal{P}\times\mathcal{T}|^{1/2} \sup_{(\boldsymbol{\mu},t)\in\mathcal{P}\times\mathcal{T}} \|\boldsymbol{q}(\boldsymbol{\mu},t) - \hat{\boldsymbol{q}}(\boldsymbol{\mu},t)\|,
\end{aligned} \tag{6}$$

Therefore, we are left to bound the error due to the neural network approximation of the map $\mathcal{Q}$, namely

$$\sup_{(\boldsymbol{\mu},t)\in\mathcal{P}\times\mathcal{T}} \|\boldsymbol{q}(\boldsymbol{\mu},t) - \hat{\boldsymbol{q}}(\boldsymbol{\mu},t)\|.$$

Firsly, we notice that we can take $n \leq 2p+3$, since $\mathcal{G}$ (and consequently $\mathcal{Q}$) is Lipschitz-continuous (see, e.g., [10, Theorem 3]). Then, we proceed as in [10], according to the following steps:

- We consider the reduced manifold $\mathcal{S}_N := \{\boldsymbol{q} = \mathcal{Q}(\boldsymbol{\mu},t) : (\boldsymbol{\mu},t) \in \mathcal{P}\times\mathcal{T}\}$; then $\mathcal{V}_n = \Psi'_*(S_N)$ is such that $\mathrm{diam}(\mathcal{V}_N) \leq LC_1\mathrm{diam}(\mathcal{P}\times\mathcal{T})$, thanks to the Lipschitz-continuity hypothesis provided by Assumption 2. Thus, by Theorem due to Gühring et al. [20] recalled in Section 1.2, there exists a ReLU DNN $\psi : \mathbb{R}^n \to \mathbb{R}^N$ such that

$$\begin{aligned}
&\sup_{\mathbf{v}\in\mathcal{V}_n} \|\psi(\mathbf{v}) - \Psi_*(\mathbf{v})\| < \frac{m}{6}|\mathcal{P}\times\mathcal{T}|^{-1/2}\varepsilon \\
&\operatorname{ess}\sup_{\mathbf{v},\mathbf{v}'\in\mathcal{V}_n} \frac{|(\psi-\Psi_*)(\mathbf{v}) - (\psi-\Psi_*)(\mathbf{v}')|}{|\mathbf{v}-\mathbf{v}'|} < \frac{m}{6}|\mathcal{P}\times\mathcal{T}|^{-1/2}\varepsilon,
\end{aligned} \tag{7}$$

  with $L_{n\to N} = c\log(\varepsilon^{-1})$ layers and $w_{n\to N} = cN\varepsilon^{-n/(s-1)}\log(\varepsilon^{-1})$ active weights. Notice that the Lipschitz constant of ψ is bounded by the quantity $C_3 = C_2 + \frac{m}{6}|\mathcal{P}\times\mathcal{T}|^{-1/2}$;
- setting $\phi_*(\boldsymbol{\mu},t) = \Psi'_*(\boldsymbol{q}(\boldsymbol{\mu},t)) \quad \forall(\boldsymbol{\mu},t) \in \mathcal{P}\times\mathcal{T}$, we notice that it is Lipschitz-continuous, with constant bounded by LC_1, and thus, by the Theorem due to Yarotski [42] recalled in Section 1.1, there exists a ReLU DNN $\phi : \mathbb{R}^{p+1} \to \mathbb{R}^n$ such that

$$\sup_{(\boldsymbol{\mu},t)\in\mathcal{P}\times\mathcal{T}} \|\phi(\boldsymbol{\mu},t) - \phi_*(\boldsymbol{\mu},t)\| < \frac{m}{6C_3}|\mathcal{P}\times\mathcal{T}|^{-1/2}\varepsilon, \tag{8}$$

  with $L_{(p+1)\to n} = c\log(\varepsilon^{-1})$ layers and $w_{(p+1)\to n} = cn\varepsilon^{-(p+1)}\log(\varepsilon^{-1})$ active weights.

Moreover, let $\hat{\boldsymbol{q}} = \psi \circ \phi : \mathbb{R}^{p+1} \to \mathbb{R}^N$ be the underlying neural network of the POD-DL-ROM, mapping parameters and time to POD coefficients. Then, by means of the triangular inequality, the perfect embedding assumption, the definition of ϕ_*, and the Lipschitz-continuity of ψ, we derive:

$$\begin{aligned}
&\sup_{(\boldsymbol{\mu},t)\in\mathcal{P}\times\mathcal{T}} \|\boldsymbol{q}(\boldsymbol{\mu},t) - \hat{\boldsymbol{q}}(\boldsymbol{\mu},t)\| \\
\leq &\sup_{(\boldsymbol{\mu},t)\in\mathcal{P}\times\mathcal{T}} (\|\Psi_*(\Psi_*'(\boldsymbol{q}(\boldsymbol{\mu},t)) - \psi(\phi_*(\boldsymbol{\mu},t))\| + \|\psi(\phi_*(\boldsymbol{\mu},t)) - \psi(\phi(\boldsymbol{\mu},t))\|) \\
\leq &\sup_{\mathbf{v}\in\mathcal{V}_N} \|\psi(\mathbf{v}) - \Psi_*(\mathbf{v})\| + C_3 \sup_{(\boldsymbol{\mu},t)\in\mathcal{P}\times\mathcal{T}} \|\phi(\boldsymbol{\mu},t) - \phi_*(\boldsymbol{\mu},t)\| < \frac{m}{3}|\mathcal{P}\times\mathcal{T}|^{-1/2}\varepsilon,
\end{aligned}$$

employing the bounds (7) and (8). Then, plugging the last inequality in (6), we can state that $\mathcal{E}_{NN} < \frac{\varepsilon}{3}$. Finally, by means of the error decomposition formula, we derive the desired bound

$$\mathcal{E}_R \leq \mathcal{E}_{POD} + \mathcal{E}_S + \mathcal{E}_{NN} < \frac{\varepsilon}{3} + \frac{\varepsilon}{3} + \frac{\varepsilon}{3} = \varepsilon,$$

with probability greater than $1 - \delta$.

Remark 4 The DL-ROM paradigm proposed in [12] and applied to cardiac electrophysiology in [17], has been theoretically analyzed in [10], providing approximation bounds and a complexity analysis, which shows that in general DL-ROMs suffer from *curse of dimensionality* with respect the number of high-fidelity dofs N_h. Relying on the present Theorem 3, we demonstrate how the preliminary dimensionality reduction through POD affects both the complexity of the POD-DL-ROM and its approximation capabilities. Indeed, POD-DL-ROMs avoid the curse of dimensionality of the DL-ROMs at the cost of discarding the *small scales* contribution, which might be, however, relevant when considering, e.g., highly nonlinear problems showing a slow eigenvalue decay. On the other hand, POD-DL-ROMs provide a neural network architecture with a lower number of *trainable* weights, thus yielding a lighter training procedure in practice. Finally, we can highlight that the *a priori* choice of employing DL-ROMs or POD-DL-ROMs must be based exclusively on the linear reducibility of the problem and the availability of computational resources.

4 Comparative analysis with deep learning-based existing strategies

On the basis of the results of the previous section, we comment the advantages of POD-DL-ROMs when compared with other deep learning-based existing strategies present in the literature, namely:

- Simple DNNs to approximate the POD (or Kernel-POD) coefficients, that results in the widely used POD+DNN approach [6, 22, 38, 41];
- The POD-DeepONets architecture, which was proposed in [29] and based on the classical DeepONets approach [28];

- The technique presented in [33], which aims at reconstructing the parameter-to-solution map by coupling linear projection methods and residual networks and which we will hereon refer to as lin+ResNets;
- The CNNs architecture for operator learning proposed in [11], whose analysis is based on the Fourier decomposition.

4.1 POD-DL-ROMs vs POD+DNNs: a matter of regularity

The purpose of this subsection is to highlight how the POD-DL-ROM approach provides a suitable setting to establish tighter bounds on the model complexity when compared to generic POD+DNNs, especially when the parameter-to-solution map is not regular.

It is worth to remark that, under the hypothesis of Theorem 3, the number of layers of the POD-DL-ROM network architecture is expected to scale as

$$L_{POD-DL-ROM} = O(\log(\varepsilon^{-1})),$$

while the total number of active weights behaves as

$$\begin{aligned} w_{POD-DL-ROM} &= w_{n\to N} + w_{(p+1)\to n} \\ &= O(N\varepsilon^{-n/(s-1)}\log(\varepsilon^{-1})) + O(n\varepsilon^{-(p+1)}\log(\varepsilon^{-1})). \end{aligned}$$

We expect that, in general, $n \ll N = N(\varepsilon)$; moreover, $n \leq 2p+3$ since the parameter-to-POD-coefficients map is Lipschitz-continuous, due to Assumption 2. Thus, it is evident that the majority of the neural network complexity amounts to the decoder, which has to perform the most difficult task, namely, decoding the information provided by the latent coordinates. Instead, the reduced network only aims at providing an alternative representation of the time-parameters vector $(\boldsymbol{\mu}, t)$ such that it makes as easy as possible for the decoder to reconstruct the POD coefficients. Noting that $w_{POD-DL-ROM}$ depends exponentially on s, we can control the complexity of the POD-DL-ROM by choosing s as large as possible, namely, $s \gg s' \geq 2$.

Essentially, we aim at finding a representation of POD coefficients of the form

$$\Psi_*(\Psi'_*(\boldsymbol{q}(\boldsymbol{\mu}, t)) = \boldsymbol{q}(\boldsymbol{\mu}, t) \qquad \forall(\boldsymbol{\mu}, t) \in \mathcal{P} \times \mathcal{T}, \tag{9}$$

through the composition of an encoder Ψ'_* that absorbs all the irregularity of the identity map $\mathcal{I} = \Psi_* \circ \Psi'_*$, and a decoder Ψ_* that is extremely regular. We highlight that the perfect embedding assumption stated in Definition 3 is critical; indeed, under the hypothesis of Theorem 3, leaving out only the perfect embedding assumption, we may be tempted to trivially use Yarotski's Theorem [42] to construct a ReLU DNN which has $L_{POD+DNN}$ layers and $w_{POD+DNN}$ active weights, where

$$\begin{aligned} L_{POD+DNN} &= O(\log(\varepsilon^{-1})) \\ w_{POD+DNN} &= O(N\varepsilon^{-(p+1)}\log(\varepsilon^{-1})), \end{aligned}$$

in order to control the relative error with $\mathcal{E}_R < \varepsilon$. Notice that:

- The number of layers $L_{POD+DNN}$ is of the same order as $L_{POD-DL-ROM}$;
- The estimate of the number of active weights $w_{POD+DNN}$ can only take advantage of mild regularity assumptions on $\mathcal{G}$ (and $\mathcal{Q}$), that is only Lipschitz-continuous.

However, it is evident that Theorem 3 only provides a theoretical result offering a different perspective in order to enhance the complexity estimate of POD+DNN. Indeed, within the framework stated by Theorem 3, given an accuracy level ε one could take advantage of the POD-DL-ROM theoretical setting, and thus the perfect embedding assumption, to construct a proper architecture that approximates the parameter-to-solution map keeping $\mathcal{E}_R < \varepsilon$ — and, then, notice that the resulting architecture is indeed in general a POD+DNN. The difference in practice is represented by the training procedure. Indeed, notice that training a network like the one involved in a POD+DNN with the classical supervised loss formulation, by letting $\omega_n = 0$ in (2.1) and thus without taking advantage of the encoder, does not ensure to recover an adequate representation in the latent space. Instead, if we train the network relying on the POD-DL-ROM paradigm, namely taking $\omega_n > 0$ in (2.1), we actually employ the encoder to implicitly enforce the architecture to satisfy the perfect embedding assumption, and then discard the encoder in the online testing phase.

Suppose now that $N \gg n$: trivially, we have that $w_{DNN} \gtrsim w_{(p+1)\to n}$; moreover, $w_{POD+DNN} \gtrsim w_{n\to N}$, upon requiring that $\frac{n}{(s-1)} < p+1$, that provides an estimate for the regularity of the decoder in the representation (9), that is $s > \frac{n}{p+1} + 1$. In practice, given that $n \leq 2p+3$, we can safely assume that $s \gtrsim 3 + \frac{1}{p+1}$ and finally $s \gtrsim 4$. Thus, if the parameter-to-solution map $\mathcal{G}$ is only Lipschitz-continuous, if the perfect embedding assumption is satisfied for $s \geq 4$, POD-DL-ROMs achieve a tighter bound on the model complexity when compared to general POD+DNN approaches: this is due to the fact that there exists a better representation (in terms of regularity) $\phi_*(\boldsymbol{\mu}, t)$ for the time-parameters vector $(\boldsymbol{\mu}, t)$ that can be recovered by the reduced network.

Until now, we considered the case where the parameter-to-solution map is only Lipschitz-continuous; however, it is interesting to consider cases where we can verify that the map $\mathcal{G}$ shows higher regularity, and see how this increased regularity affects the complexity of both POD-DL-ROMs and POD+DNNs in terms of number of active weights. Indeed, by means of similar arguments employed previously, and thanks to the Theorem due to Yarotski [42] recalled in Section 1.1, assuming that $\mathcal{G} \in W^{r,+\infty}(\mathcal{P} \times \mathcal{T}; \mathbb{R}^{N_h})$, we obtain that

$$w_{POD+DNN} = O(N\varepsilon^{-(p+1)/r}\log(\varepsilon^{-1})).$$

Thanks to the fact that the *exact* reduced map ϕ_* of Theorem 3 now would be $\min\{r, s'\}$-times differentiable,

$$\begin{aligned} w_{POD-DL-ROM} &= w_{n\to N} + w_{(p+1)\to n} \\ &= O(N\varepsilon^{-n/(s-1)}\log(\varepsilon^{-1})) + O(n\varepsilon^{-(p+1)/\min\{r,s'\}}\log(\varepsilon^{-1})) \end{aligned}$$

Assuming that $N \gg n$, it is trivial to verify that $w_{POD+DNN} \gtrsim w_{(p+1)\to n}$; furthermore, it is valid that $w_{POD+DNN} \gtrsim w_{n\to N}$ if $\frac{n}{(s-1)} > \frac{p+1}{r}$, that is $s > \frac{nr}{p+1} + 1$, which gives the estimate $s \gtrsim (2 + \frac{1}{p+1})r + 1$ and finally $s \gtrsim 3r + 1$. The meaning of the last estimate is that, if the parameter-to-solution map is extremely regular (namely, $r \to \infty$), it becomes more and more difficult for the POD-DL-ROMs to guarantee lower complexity than simple POD+DNNs, since the perfect embedding assumption should be verified for $s \to \infty$. This is rather intuitive: indeed, if the parameter-to-solution map is extremely regular, we do not need to recover a better representation $\phi_*(\boldsymbol{\mu}, t)$ for the time-parameter vector $(\boldsymbol{\mu}, t)$ in order to make it easier for the underlying neural network to learn the solution manifold.

4.2 POD-DeepONets and POD-DL-ROMs: a comparison

In this subsection, we aim at analyzing the POD-DeepONet architecture from a theoretical standpoint, showing the close relationship with POD-DL-ROMs when dealing with problems whose general formulation can be reduced to (2). We let X be a Banach space and we consider two compact subsets, $K_1 \subset X$ and $K_2 \subset \mathbb{R}^d$, where d denotes the number of spatial (or spatio-temporal) dimensions of the problem at hand. Defining $W \subset C(K_1)$ as a compact subset, we suppose that we aim at learning the operator $\mathcal{G}_{\infty\to\infty} : W \to C(K_2)$, where the subscript highlights that the considered operator is a map between infinite-dimensional spaces. We first consider a DeepONet architecture [28] employed to reconstruct $\mathcal{G}_{\infty\to\infty}$, which in its *unstacked* formulation consists in the combination of the output of two different neural networks through the scalar product. In particular, we define the *branch net* $\boldsymbol{b} : W \to \mathbb{R}^N$ as the neural network that processes information about the input function $\phi \in W$, and the *trunk net* $\boldsymbol{\tau} : \mathbb{R}^d \to \mathbb{R}^N$, which aims at encoding the coordinate input $y \in \mathbb{R}^d$ in a set of basis functions. Then, we can define the DeepONet approximation as

$$\mathcal{G}_{\infty\to\infty}(\phi)(y) \approx \hat{G}(\phi)(y) = \boldsymbol{b}(\phi) \cdot \boldsymbol{\tau}(y). \tag{10}$$

and note that N describes the number of basis functions employed in the decomposition (10); thus, N plays the same role as the POD dimension in the POD-DL-ROM architecture. Based on the analysis proposed in [26], we can split the DeepONet operator $\hat{G} : W \to C(K_2)$ into $\hat{G} = \mathcal{R}_\tau \circ \mathcal{A}_{m\to N} \circ \mathcal{E}_m$, where $\mathcal{E}_m$, $\mathcal{A}_{m\to N}$ and $\mathcal{R}_\tau$ are defined as follows:

- the *encoder* operator is defined as the map $\mathcal{E}_m : C(K) \to \mathbb{R}^m$, such that, given $x_i \in K_1, \forall i = 1, \dots, m$:

  $$\mathcal{E}_m(\psi) = [\psi(x_1), \psi(x_2), ..., \psi(x_m)]^T \qquad \forall \psi \in C(K_1).$$

  It is worth noticing that $\mathcal{E}_m$ is well defined since any continuous function can be evaluated pointwise;
- $\mathcal{A}_{m\to N} : \mathbb{R}^m \to \mathbb{R}^N$ is the *approximation* operator; thus, we can decompose the branch net of the DeepONet operator as $\boldsymbol{b} = \mathcal{A}_{m\to N} \circ \mathcal{E}_m$;

- recalling that $\boldsymbol{\tau} : \mathbb{R}^d \to \mathbb{R}^N$ is the trunk net, we define the τ-induced *reconstructor* operator $\mathcal{R}_\tau : \mathbb{R}^N \to C(K_2)$ as

$$\mathcal{R}_\tau(\boldsymbol{\xi}) = \boldsymbol{\xi} \cdot \boldsymbol{\tau} \qquad \forall \boldsymbol{\xi} \in \mathbb{R}^N.$$

In a more compact formulation, we retrieve the classical architecture of the DeepONets, namely:

$$\hat{G}(\phi) = \mathcal{R}_\tau \circ (\mathcal{A}_{m\to N} \circ \mathcal{E}_m(\phi)) = \mathcal{R}_\tau \circ \boldsymbol{b}(\phi) = \boldsymbol{b}(\phi) \cdot \boldsymbol{\tau}.$$

POD-DeepONets were recently introduced in [29] and the test cases considered within the paper confirm better approximation accuracy when compared with classical DeepONets: the methodology consists in substituting the *trunk net* with the corresponding row of the POD matrix. The drawback is that POD-DeepONets can only approximate operators defined as $\mathcal{G}_{\infty\to N_h} : W \to \mathbb{R}^{N_h}$, losing the capability of mapping between infinite-dimensional spaces.

Supposing to initially deal with stationary, time-independent problems and denoting by $\mathbf{v}_j \in \mathbb{R}^N$ the j-th row of the POD matrix $\mathbf{V} \in \mathbb{R}^{N_h \times N}$, we define the *expansion* operator $L_{\mathbf{v}_j} : \mathbb{R}^N \to \mathbb{R}^{N_h}$ as

$$L_{\mathbf{v}_j}(\boldsymbol{\xi}) = \boldsymbol{\xi} \cdot \mathbf{v}_j \qquad \forall \boldsymbol{\xi} \in \mathbb{R}^N,$$

and the POD-DeepONet operator as

$$[\mathcal{G}_{\infty\to N_h}(\phi)]_j \approx [\hat{G}_{POD-DeepONet}(\phi)]_j = \hat{\boldsymbol{q}}(\phi) \cdot \mathbf{v}_j = L_{\mathbf{v}_j} \circ \mathcal{A}_{m\to N} \circ \mathcal{E}_m(\phi),$$

$\forall j = 1, \ldots, N_h$, where $\hat{\boldsymbol{q}}$ is the corresponding *branch net*, which now approximates the underlying POD coefficients. It is worth to notice that, by employing the vector formulation, we can write:

$$\mathcal{G}_{\infty\to N_h}(\phi) \approx \hat{G}_{POD-DeepONet}(\phi) = \mathbf{V}\hat{\boldsymbol{q}}(\phi).$$

Then, we need to adapt the POD-DeepONet framework to the problem considered within this work (2), where even the input parameter space is finite-dimensional, thus eliminating the need of the *encoder* operator $\mathcal{E}_m$. Indeed, POD-DeepONets for finite-dimensional-input problems involving the reconstruction of the map $\mathcal{G}_{p\to N_h} : \mathbb{R}^p \to \mathbb{R}^{N_h}$ take the form

$$[\mathcal{G}_{p\to N_h}(\boldsymbol{\mu})]_j \approx [\hat{G}_{POD-DeepONet}(\boldsymbol{\mu})]_j = \hat{\boldsymbol{q}}(\boldsymbol{\mu}) \cdot \mathbf{v}_j = L_{\mathbf{v}_j} \circ \mathcal{A}_{p\to N},$$

$\forall j = 1, \ldots, N_h$, or in a more compact way

$$\mathcal{G}_{p\to N_h}(\boldsymbol{\mu}) \approx \hat{G}_{POD-DeepONet}(\boldsymbol{\mu}) = \mathbf{V}\hat{\boldsymbol{q}}(\boldsymbol{\mu}),$$

where $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^p$, $\mathcal{P}$ compact. It is worth noticing that in this case, the *branch net* coincides with the *approximation* operator $\mathcal{A}_{m\to N}$.

Finally, in order to include also the time-dependence, we could adopt two different strategies:

- We could treat the time t as a spatial coordinate in a DeepONet-like way, leading to a POD matrix of dimension $N_h N_t \times N$, that however increases the possible impact of the *curse of dimensionality*, however offering the opportunity to deal with time-dependent basis functions;
- Alternatively, we may consider the time t as an additional parameter, a choice which reduces the computational requirements and is consistent with the POD-DL-ROM approach, leading to the construction of time-independent global spatial basis functions.

Within this comparison, for the sake of consistency, we choose to employ this latter approach. Thus, aiming at reconstructing the map $(\boldsymbol{\mu}, t) \mapsto \mathbf{u}(\boldsymbol{\mu}, t)$, we could employ different neural network architectures; for instance, if we choose to employ a DL-ROM architecture as the *branch net* of the POD-DeepONets, we retrieve the POD-DL-ROM approach, while employing a *vanilla* DNN as the *branch net* results in the POD+DNN approach. The comparison between POD-DL-ROM and POD+DNNs is extensively treated in the previous subsection.

Finally, inspired by the DeepONet approach, we notice that extending the content of the present paper to the case of infinite-dimensional input parameters is straightforward and introduces an additional source of error, namely the *encoding* error, that ultimately depends on the variability of the input parameters and their spatial discretization; for a thorough discussion on the topic, we refer the reader to, e.g., [26].

4.3 Learning POD coefficients with ResNets

The ResNets-based approach proposed in [33] couples linear decompositions and residual networks (ResNets) to reconstruct field-to-solution maps, an approach which is inherently close to POD-DL-ROMs. In this case, we start our analysis of the technique by examining the proposed architecture, and by adapting it to the problem formulation considered within the present work.

Indeed, we immediately notice that the lin+ResNet architecture needs that every residual layer has input dimension equal to the output dimension layer output dimension: iterating, for a fully residual network, we must require that the input of the network has the same dimension of the network output. Such a constraint in the architecture is managed in [33] by projecting both the input fields and the output targets onto two linear subspaces of equal dimension $N \ll N_h$, where N_h is the FOM dimension. Then, the output targets are numerically approximated on the same mesh and projected onto a subspace of dimension N, too. The approach results in the sequence of maps:

$$\mathbb{R}^{N_h} \xrightarrow{lin.proj.} \mathbb{R}^N \xrightarrow{residual} \mathbb{R}^N \xrightarrow{residual} \dots \xrightarrow{residual} \mathbb{R}^N \xrightarrow{lin.lift.} \mathbb{R}^{N_h},$$

where the linear projection is usually carried out by employing POD, Karhunen-Loève expansions [39] or active subspaces [44]. However, when dealing with finite-dimensional parameter inputs instead of fields (for instance $(\boldsymbol{\mu}, t) \in \mathbb{R}^{p+1}$ with $p+1 < N$), it may occur that the ResNet input dimension $(p + 1)$ is different from the output dimension N; to fill the gap, it is necessary to employ for instance a dense layer

$\mathbb{R}^{p+1} \to \mathbb{R}^N$ as the first layer of the architecture. Thus, we will consider the sequence of maps:

$$\mathbb{R}^{p+1} \xrightarrow{dense} \mathbb{R}^N \xrightarrow{residual} \mathbb{R}^N \xrightarrow{residual} \dots \xrightarrow{residual} \mathbb{R}^N \xrightarrow{lin.lift.} \mathbb{R}^{N_h}.$$

The lin+ResNets approach ultimately aims at providing a constructive way to build a neural network in terms of *breadth* and *depth*.

The *breadth*, which may be intuitively defined as the maximum number of neurons per layer in the network, coincides with N, the characteristic dimension of the preliminary dimensionality reduction. In order to favor compressed representations, the authors of [33] suggest keeping as low as possible the *latent* dimension k of the ResNet, which can be identified with the dimension of the nonlinearity added at each layer. Indeed, the residual map between the layer $\mathbf{z}_l \in \mathbb{R}^N$ and $\mathbf{z}_{l+1} \in \mathbb{R}^N$ can be identified with

$$\mathbf{z}_{l+1} = \mathbf{z}_l + \mathbf{W}_{1l}\sigma(\mathbf{W}_{0l}\mathbf{z}_l + \mathbf{b}_l),$$

where $\mathbf{W}_{0l} \in \mathbb{R}^{N\times k}$, $\mathbf{W}_{1l} \in \mathbb{R}^{k\times N}$, $\mathbf{b}_l \in \mathbb{R}^k$ and σ is the activation function; the total number of weights per layer is then $O(Nk)$. However, in contrast to our approach, they did not propose a way to identify k: we remark that the discussion on the latent dimension n of the POD-DL-ROM architecture is fundamental because it allows to set a tighter bound on the complexity of the decoder network in terms of active weights.

Furthermore, the authors developed approximation bounds on the underlying ResNet complexity in terms of its *depth*, employing the connection between ResNets, Neural ODE and control flows [5]. The bound on the ResNet *depth* enable the user to control the ℓ^2 error on the solution (and by extension the relative error too) with a suitable bound ε by employing $O(\varepsilon^{-1})$ layers. Thus, we can straightforwardly state that, on the basis of the complexity analysis, POD-DL-ROMs outperform the ResNets-based approach in terms of number of layers:

$$L_{lin+ResNets} = O(\varepsilon^{-1}) \gtrsim O(\log(\varepsilon^{-1})) = L_{POD-DL-ROM}$$

and number of active weights:

$$w_{lin+ResNets} = O(Nk\varepsilon^{-1}) \gtrsim O(N\varepsilon^{-n/(s-1)}\log(\varepsilon^{-1}))+ \\ + O(n\varepsilon^{-(p+1)}\log(\varepsilon^{-1}))) = w_{POD-DL-ROM},$$

supposing for instance $N \gg n$ and $s \geq n+1$, which are reasonable assumptions. Indeed, $N \gg n$ is satisfied when the nonlinear Kolmogorov n-width decays much faster than the eigenvalue decay of the correlation matrix, a phenomenon that is usually encountered in applications; the condition $s \geq n+1$ is valid by ensuring $s \geq 2p+4$, that is the decoder map must be sufficiently regular.

Despite the disadvantage on the complexity front, we remark that ResNets constitute one of the most suitable paradigms to implement adaptive-depth architectures since adding a layer to an already trained architecture can produce an arbitrary small perturbation on the network output; for a more detailed analysis on the lin+ResNets training, we refer the reader to [33].

4.4 The effect of the POD basis optimality on the network complexity

Within this subsection, our purpose is finally to show how choosing the POD basis as global spatial basis function in the linear decomposition leads to a reduced complexity of the underlying neural network, comparing in detail CNNs for operator learning and POD-DL-ROMs. In particular, we notice that, within the POD-DL-ROM approach, the reconstruction of the approximated solution at the high-fidelity level depends on the decomposition assumption $\mathbf{u}(\boldsymbol{\mu}, t) \approx \sum_{j<N} \hat{q}_j(\boldsymbol{\mu}, t)\mathbf{v}_j$, where N denotes the POD dimension. Analogously, the recent work on the approximation bounds for CNNs proposed in [11] strives to reconstruct a decomposition between global spatial basis functions that are strictly related to the Fourier modes, and a set of coefficients, that is, $\mathbf{u}(\boldsymbol{\mu}, t) \approx \sum_{j<C} \hat{a}_j(\boldsymbol{\mu}, t)\boldsymbol{f}_j$, where the sum is over C terms (the number of channels in the input and output is $O(C)$).

In the following, we assume that $u(\cdot, \boldsymbol{\mu}, t) \in C^{\alpha}(\Omega)$ for any $(\boldsymbol{\mu}, t) \in \mathcal{P} \times \mathcal{T}$, being $\alpha \geq 1$ the spatial regularity, and $\varepsilon > 0$ is the desired accuracy level; we then describe the three main differences between the CNN-based approach and the POD-DL-ROM technique:

- The convolutional block is limited to uniformly spaced mesh points (h is the spacing parameter) in square domains, while POD-DL-ROMs are more versatile both in terms of the domain shape and the mesh properties.
- The architecture proposed in [11] consists of two different blocks: the dense block is devoted to the parameter-dependent coefficient approximation, while the convolutional block strives to reconstruct the spatial basis function. Instead, POD-DL-ROMs compute the spatial basis before the training of neural networks by means of SVD [36] or randomized SVD [40] through an unsupervised learning criterion: in principle, this means that POD-DL-ROMs do not need any active weights to reconstruct the spatial basis functions, while the CNN approach needs $O(\varepsilon^{-\frac{2}{2\alpha-1}} \log(h^{-1}))$ weights to *learn* them (we refer the reader to *Theorem 2* in [11]).
- In the decomposition employed in [11], C plays the role of the reduced dimension: it is an analog of the POD dimension N employed within the POD-DL-ROM technique. In the following, we exploit an optimality result fulfilled by the POD basis to show that the complexity of the neural network in the parameter-to-coefficient map approximation is lower in the case of POD-DL-ROM when compared to the approach proposed in [11].

The quasi-optimality of the POD decomposition in its discrete formulation confirms that with a N-terms truncation, provided a sufficient amount of data have been suitably sampled, no linear decomposition captures as much variance as the discrete formulation of the POD decomposition so that the reduced dimension C of [11] satisfies the inequality $C > N$ with probability $1 - \delta$ (see Section 2.2 and Appendix 3). Furthermore, we assume that:

(i) $N \gg n$ as usual, since we expect that the nonlinear Kolmogorov n-width decays (much) faster than the linear reduced dimension N;
(ii) $u(\cdot, \boldsymbol{\mu}, t) \in C^{\alpha}(\Omega)$ for any $(\boldsymbol{\mu}, t) \in \mathcal{P} \times \mathcal{T}$ for some $\alpha \geq 1$ to comply with the hypotheses of *Theorem 2* of [11];

(iii) The parameter-to-solution map has regularity r, i.e., $\mathcal{G} \in W^{r,\infty}(\mathcal{P} \times \mathcal{T}; \mathbb{R}^{N_h})$;
(iv) The decoder map is adequately regular, namely $\frac{n}{s-1} > \frac{p+1}{r}$ ($s \geq 3r + 1$ is sufficient, as in 4.1).

We recall that *Theorem 2* in [11] provides the estimate $C = O(\varepsilon^{-\frac{2}{2\alpha-1}})$. Therefore, in the worst case scenario $N = O(\varepsilon^{-\frac{2}{2\alpha-1}})$; however, depending on the singular values decay that in some cases might be even exponential (e.g., stationary elliptic PDEs, analytic parameter-to-solution maps, see [36]), we actually obtain improved estimates. We then derive:

$$\begin{aligned}
w_{POD-DL-ROM} &= O(N\varepsilon^{-n/(s-1)}\log(\varepsilon^{-1})) + O(n\varepsilon^{-(p+1)}\log(\varepsilon^{-1})) \\
&\approx O(N\varepsilon^{-n/(s-1)}\log(\varepsilon^{-1})) \\
&\lesssim O(C\varepsilon^{-n/(s-1)}\log(\varepsilon^{-1})) \\
&= O(\varepsilon^{-\frac{2}{2\alpha-1}-\frac{n}{(s-1)}}\log(\varepsilon^{-1})) \\
&\lesssim O(\varepsilon^{-\frac{2}{2\alpha-1}}[\varepsilon^{-\frac{n}{(s-1)}}(\log(\varepsilon^{-1}) + \log(h^{-1})]) \\
&\lesssim O(\varepsilon^{-\frac{2}{2\alpha-1}}[\varepsilon^{-\frac{p+1}{r}}(\log(\varepsilon^{-1}) + \log(h^{-1})]) \\
&= w_{CNN}.
\end{aligned}$$

Thus, we can conclude that, if the hypotheses setting is verified, the overall complexity of the POD-DL-ROMs in terms of active weights is lower (or equal) than the complexity of the CNN architecture proposed in [11].

5 Numerical experiments

Within this section, we present different numerical tests, aiming at validating the theoretical analysis proposed in the previous Sections. In particular, we focus on *(i)* the error bounds of Theorems 2–3 and the error decomposition formula, as well as on *(ii)* the role of the reduced dimension N and the total number of snapshots N_{data} and on *(iii)* the comparison against recent approaches proposed in the literature, in light of the theoretical results of Sects. 3 and 4. In particular, the numerical experiments involve:

a) A benchmark test case with an analytically defined operator that allows us to know *a priori* the properties of the parametric operator (like, e.g., the regularity of the parameter-to-solution map) in order to validate the theoretical estimates on the network complexity;
b) A linear 1D initial boundary value problem (IBVP), to show how to select N_{data} and N in order to minimize the *a priori* error (given by the sum of $\mathcal{E}_S$ and $\mathcal{E}_{POD}$), then validating *a posteriori* the network complexity as a function of the relative error;
c) A nonlinear 2D time-dependent IBVP in a non-conventional domain, to show the effectiveness of the POD-DL-ROM approach when dealing with more complex

problems, validating also the *lower bound* and the *upper bound* on the relative error $\mathcal{E}_R$, which stem from the theoretical analysis.

We remark that the complexity analysis of POD-DL-ROM and related approaches is discussed from a theoretical point of view only in terms of the approximation error; however, when numerical experiments are addressed, we also have to take into account the training error, which plays a major role especially when the network is sufficiently deep or wide, or data are limited. For the same reasons, in our numerical experiments, we mainly address the complexity study in terms of number of active weights w, since the latter is a quantity which is less sensitive (when compared to the depth L) to the training error. Thus, the experimental complexity analysis presented here may not reflect exactly the estimates provided in the previous sections, but they validate qualitatively the theory. However, within the present section, aiming at mitigating the effect of the training error on the error estimates, we employ several *ad hoc* strategies, like, e.g.,

– We employ early stopping to prevent overfitting;
– The approximation results in terms of network complexity are achieved in an error range $[\varepsilon_1, \varepsilon_2]$ that is deemed appropriate for the chosen number of samples N_{data}: in practice, the training error depends on data availability;
– For a fixed number of active weights, we regulate the network architecture trying to randomly achieve the configuration that minimizes the training error; we keep the depth of the network as low as possible in order to ensure convergence to a suitable minimum and avoid expensive training loops;
– Starting from educated guesses, we look for the best training hyperparamenters (which are the learning rate and the learning rate decay).

Finally, we remark that, in order to comply with the hypotheses of the Theorems of Section 3, we limit the numerical experiments to generic dense layers equipped with the α-LeakyReLU activation function,

$$\text{LeakyReLU}_\alpha(x) = \begin{cases} x, & x \geq 0 \\ \alpha x, & x < 0. \end{cases}$$

Unless otherwise stated, we set $\alpha = 0.1$. The optimization procedure is carried out by employing the Adam algorithm [25]. Note that although our Theorems are stated for ReLU activations, we adopt the α-LeakyReLU variation to enhance model training. It is well known, indeed, that ReLU networks are harder to train because of vanishing gradients [19]; still, this modification is consistent with our theory as, when it comes to model expressivity and architectural complexity, α-LeakyReLU and ReLU networks are known to be mathematically equivalent [19].

All the code was implemented in Python 3, mainly relying on the TensorFlow and FEniCS libraries, and it is freely available on Github.[1]

[1] https://github.com/briviosimone

5.1 Benchmark test case

We begin our experimental analysis by considering a benchmark test case similar to the one described in [11], and involving the reconstruction of an analytically defined operator, namely

$$u_\beta(x, \boldsymbol{\mu}) = \mu_3 |x - \mu_1|^\beta e^{-\mu_2 x}, \qquad x \in [0, 1],$$

where $\boldsymbol{\mu} = [\mu_1, \mu_2, \mu_3] \in \mathcal{P} = [0, 1] \times [0, 1] \times [1, 2]$. Within this numerical test we vary $\beta \in \{3/2, 7/3, 3\}$ and we analyze the three resulting cases independently. Notice that the hyperparameter $\beta > 0$ controls the regularity of the parameter-to-solution map. Indeed,

$$\begin{aligned} u_{3/2}(x, \cdot) &\in W^{1,+\infty}(\mathcal{P}) \setminus W^{2,+\infty}(\mathcal{P}) \\ u_{7/3}(x, \cdot) &\in W^{2,+\infty}(\mathcal{P}) \setminus W^{3,+\infty}(\mathcal{P}) \\ u_3(x, \cdot) &\in W^{3,+\infty}(\mathcal{P}) \setminus W^{4,+\infty}(\mathcal{P}); \end{aligned}$$

thus $\beta = 3/2, 7/3, 3$ correspond to $r = 1, 2, 3$ respectively, where r is defined as the regularity of the parameter-to-solution map in agreement with this paper notation. Furthermore, the problem does not depend on the time variable, thus we set $N_t = 1$, $N_{data} = N_s$ and $p = 2$ (instead of $p = 3$) to comply with the theoretical framework of the present work. Moreover, we discretize the problem in space by means of a uniform discretization with $N_h = 1000$. Selecting $n = 5 \leq 2p + 3 = 7$ to ensure both a suitable compression and an adequate representation in the latent space, $N_s = 500$, and

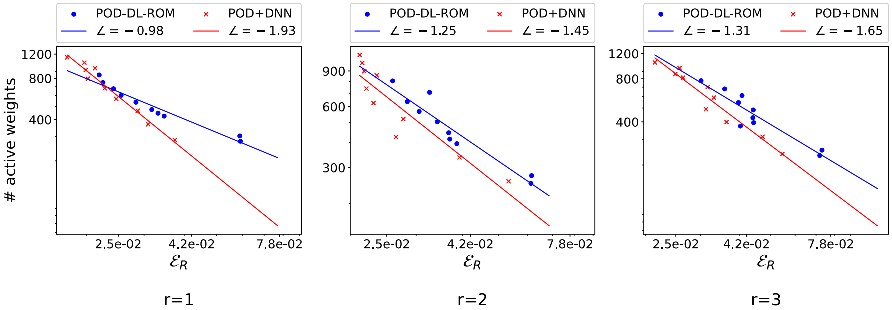
$$N = N(r) = \begin{cases} 20, & r = 1 \\ 17, & r = 2 \\ 15, & r = 3, \end{cases}$$

to control the variability retained by the preliminary linear dimensionality reduction. We then proceed towards a complexity analysis, showing a comparison of the results against the CNN approach considered in [11], the POD+DNN framework, and the lin+ResNets technique. We remark that for the sake of fairness and consistency, we keep the batch size during training equal to $B = 20$ for every comparison considered in the benchmark test case. Then, for any $r \in \{1, 2, 3\}$, we estimate the approximation error $\mathcal{E}_R$ on the respective test set consisting of $N_s^{test} = 10^4$ samples.

From a theoretical standpoint, we immediately notice $\mathcal{G} : \boldsymbol{\mu} \mapsto \mathbf{u}(\boldsymbol{\mu}) \in W^{r,+\infty}(\mathcal{P}; \mathbb{R}^{N_h})$; then, from the findings of Section 4, since $n \ll N$, we can infer that

$$\begin{aligned} w_{POD+DNN} &= O(N\varepsilon^{-3/r} \log(\varepsilon^{-1})) \\ w_{POD-DL-ROM} &= O(N\varepsilon^{-5/(s-1)} \log(\varepsilon^{-1})). \end{aligned}$$

Thus, owing to the fact that in the POD-DL-ROMs approach the perfect embedding assumption with coefficients s, s' is enforced thanks to their peculiar loss formulation, we expect them yielding a less steep increase (when compared to POD+DNNs) in the model complexity as the accuracy level decreases whenever the decoder map is



r=1 r=2 r=3

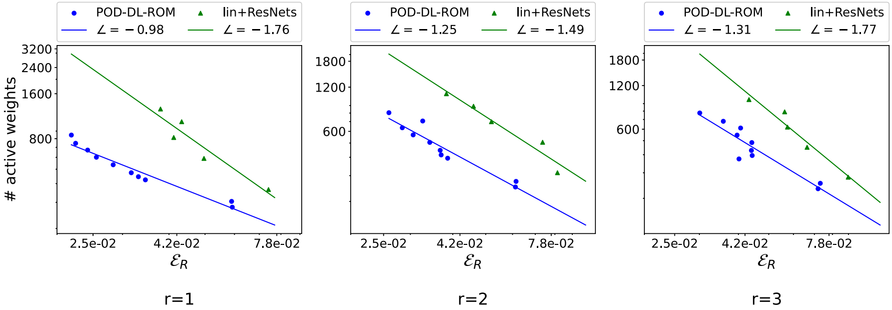
Fig. 1 Benchmark test case: model complexity comparison between POD-DL-ROMs and POD+DNNs as the parameter-to-solution regularity r varies in $\{1, 2, 3\}$. The trends are displayed through solid lines, which fit the collected results in the least squares sense

suitably regular, which is equivalent to require $s > \frac{5}{3}r + 1$. Figure 1 demonstrates that the latter behavior is more likely to happen as the regularity of the parameter-to-solution map r decreases.

We then compare POD-DL-ROMs against the lin+ResNets approach; for the latter, we limit the analysis to the case where the basis functions are yielded by POD for the sake of consistency. We thus fix the latent space dimension of the residual layers as $k = 5$ and, from the estimates obtained in Section 4, we recall that the complexity bound of lin+ResNets in terms of number of active weights is in general independent of the regularity of the parameter-to-solution map, namely:

$$w_{lin+ResNets} = O(Nk\varepsilon^{-1}).$$

We thus remark that the lin+ResNets approach does not take advantage of any regularity assumption on the parameter-to-solution map: we then expect a similar trend as r varies in $\{1, 2, 3\}$. Nonetheless, if the trained POD-DL-ROM architecture is able to find an adequate representation in the latent space which induces a very regular decoder, that is $s > 6$, we can ensure that the POD-DL-ROM outperform the lin+ResNets approach in terms of complexity: this behavior is indeed observed in Fig. 2.



r=1 r=2 r=3

Fig. 2 Benchmark test case: model complexity trend of POD-DL-ROMs and the lin+ResNets approach for different values regularity of the parameter solution map r

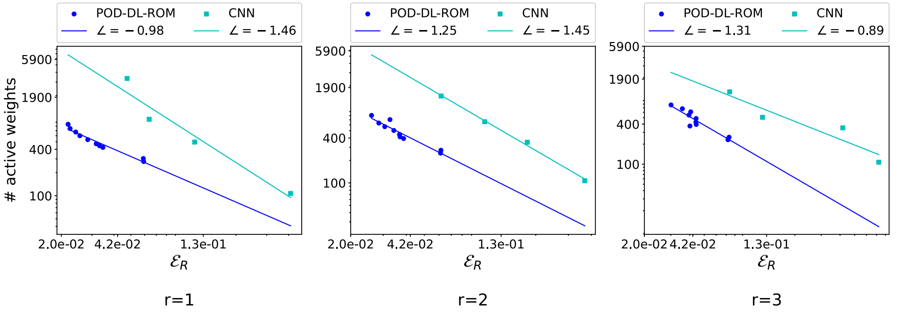


Fig. 3 Benchmark test case: comparison between POD-DL-ROMs and CNNs in terms of number of active weights, varying the regularity $r \in \{1, 2, 3\}$

Finally, we consider the comparison against the CNN approach considered in [11]: if the decoder map is sufficiently regular (from the theoretical analysis we derive the condition $s \geq \frac{5}{3}r + 1$), POD-DL-ROMs take advantage of the basis optimality to achieve a less steep increase of complexity as the error bound $\mathcal{E}_R < \varepsilon$ decreases: the behavior is indeed observed in Fig. 3, in the cases when the regularity of the parameter-to-solution map is low ($r = 1, 2$). Moreover, differently from the CNN-based technique, we remark that the POD-DL-ROMs' algorithm does not require to learn the basis functions, thus not affecting the overall complexity of the underlying network.

5.2 1D initial boundary value problem

The present test case is designed to highlight the advantages of POD-DL-ROMs when compared to other considered approaches even when dealing with time-dependent parametrized problems. Moreover, before starting the training process, we show *a priori* how to choose the hyperparameters N, N_s, N_t, based on the analysis of $\mathcal{E}_S$ and

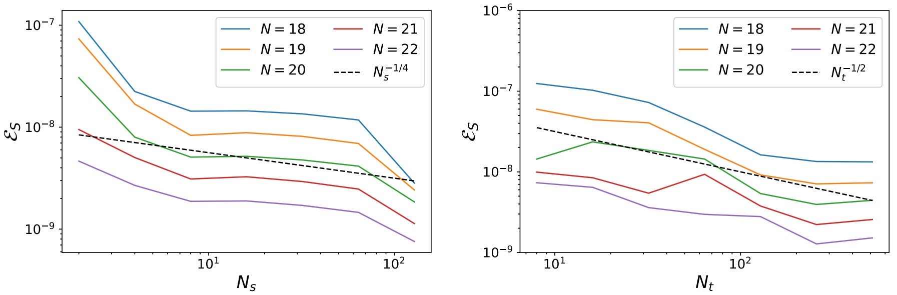


Fig. 4 1D IBVP test case: decay of the sampling error $\mathcal{E}_S$ with respect to N_s, N_t, and N

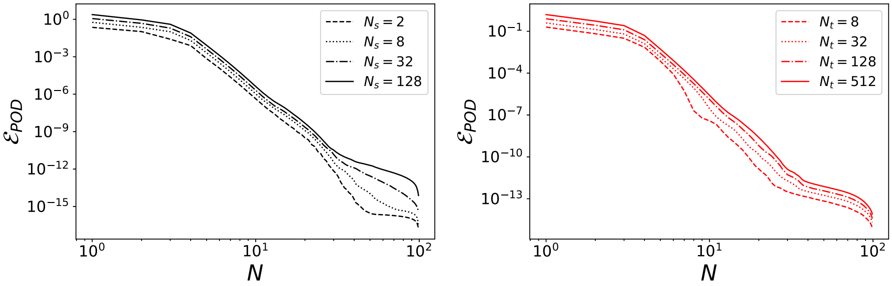


Fig. 5 1D IBVP test case: decay of the projection error $\mathcal{E}_{POD}$ varying N_s, N_t, and N

$\mathcal{E}_{POD}$. In particular, we consider the following IBVP:

$$\begin{cases} \dfrac{\partial u}{\partial t} - \dfrac{\partial^2 u}{\partial x^2} = u + 10\cos(x)\sin(2\pi t), & \text{in } (0,\pi) \times (0,T] \\ u = 10(2\mu^3 - 3\mu^2 + \mu), & \text{at } \{x = 0\} \times (0,T] \\ \dfrac{\partial u}{\partial x} = 2|1 - 2\mu| - 1, & \text{at } \{x = \pi\} \times (0,T] \\ u(x,0) = u_0(\mu), & \text{in } (0,\pi), \end{cases}$$

where the initial condition is

$$u_0 = u_0(\mu) = 10(2\mu^3 - 3\mu^2 + \mu)\cos(x) + (2|1 - 2\mu| - 1)\sin(x),$$

while $\mu \in \mathcal{P} = [0,1]$ and $T = 1$. Thus, $p = 1$ and we can fix $n = 5 = 2p + 3$ to ensure an adequate representation in the latent space, according to the framework presented in the present paper. We collected synthetic data generated with a high-fidelity model solved on a uniform grid of $N_h = 100$ points: we generate a test set of $N_s^{test} = 100$ samples of $N_t^{test} = 200$ snapshots each with a MATLAB-based PDE solver, sampling $\boldsymbol{\mu} \sim \mathcal{U}(\mathcal{P})$ iid and t from a uniform grid of step $\Delta t^{test} = T/N_t^{test}$.

We start by analyzing the dependence of $\mathcal{E}_S$ on N_s, N_t and N; for the sake of clarity, we specify that the sampling criterion employed in the *a priori* analysis below is based on the theoretical analysis of the entire work: thus, we assume $\boldsymbol{\mu} \sim \mathcal{U}(\mathcal{P})$ iid and that t is sampled from a uniform grid of step $\Delta t = 1/N_t$. To analyze the effect of N_s on the sampling error, we fix $N_t = 1000$ and we generate a group of datasets depending on $N_s \in \{l = 2^k : k = 1, \ldots, 7\}$: as shown in Fig. 4, the decay has slope $-1/4$ and it is independent of the chosen value of N. Conversely, we fix $N_s = 100$ and vary $N_t \in \{l = 2^k : k = 3, \ldots, 9\}$, validating experimentally in Fig. 4 that $\mathcal{E}_S \sim N_t^{-1/2}$, independently of N. We then move to the analysis of the projection error, showing in Fig. 5 how $\mathcal{E}_{POD}$ decays with N and is mostly independent of N_s and N_t respectively. We notice that the present analysis is done before the training of the underlying neural network and allows us to know *a priori* how much variance is not accounted for due to the sampling ($\mathcal{E}_S$) and the initial dimensionality reduction ($\mathcal{E}_{POD}$), allowing us to calibrate the values N, N_s, N_t before we start the expensive training procedure. The

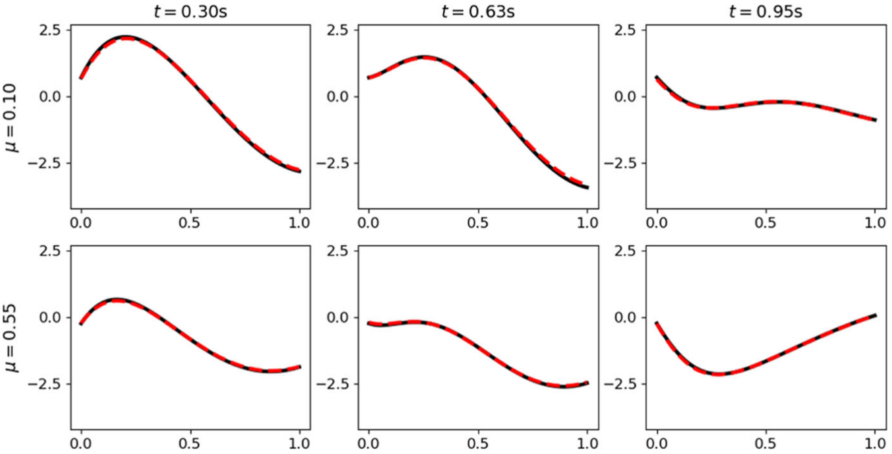


Fig. 6 1D IBVP test case: comparison between the “true” solution (solid black line) and the most accurate POD-DL-ROM prediction (dashed red line) to demonstrate that the variability of the solution manifold is correctly reproduced

idea is to choose N, N_s, N_t to guarantee that $\mathcal{E}_{POD}$ and $\mathcal{E}_S$ are suitably small, so that we can control the relative error $\mathcal{E}_R$ with a strict bound, which is provided by the error decomposition of Theorem 1. Thus, based on the results of the present *a priori* analysis, we choose $N_s = 50$, $N_t = 20$, $N = 20$ (Fig. 6).

We then move our focus to the comparison of the POD-DL-ROM technique against other approaches in terms of complexity, showing the relation between the relative error $\mathcal{E}_R$ and the number of active weights employed in the underlying neural network. Notice that, since the analytical solution of the IBVP is not available, here we are not provided with any information on the regularity of the parameter-to-solution map. Anyway, experimental results on the complexity analysis confirm our theoretical expectations: when dealing with parameter-to-solution maps arising from parametric PDEs, POD-DL-ROMs' complexity increases slower than POD+DNNs' one as the relative error decreases. Indeed, the latent representation of the POD-DL-ROM approach induces a decoder that is extremely regular, that is $s \gg 2$, which enables a slow increase in network complexity, as suggested by the theoretical approximation bounds of Theorem 3 and validated in Fig. 7. Similarly, we notice that the results relative to the comparison between POD-DL-ROMs and lin+ResNets are in agreement with the theory, demonstrating again how, lin+ResNets are outperformed in terms of complexity by POD-DL-ROMs, when it is possible for the latter to achieve an extremely regular decoder map due to an adequate latent representation. Finally, when compared to the Fourier-inspired CNN technique POD-DL-ROMs' number of active weights show a slower increase as the relative error $\mathcal{E}_R$ decreases, as shown in Fig. 7; as proved theoretically in Section 4, the magnitude of the slope is strongly linked to the optimality of the basis functions. Moreover we validate how the burden of learning the set of basis function impacts heavily on the underlying CNN complexity, which shows a remarkable difference when compared to the POD-DL-ROM approach in terms of number of active weights, not only regarding the slope magnitude but also in the absolute sense. The observed behavior highlights how crucial it is in terms of

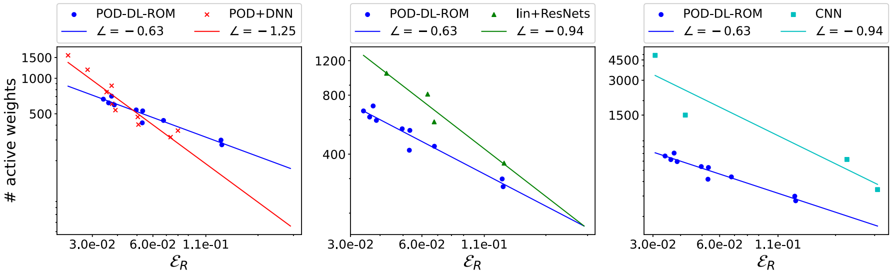


Fig. 7 1D IBVP test case: comparison between POD-DL-ROMs and other techniques in terms of number of active weights. The solid line represents the least squares fitting of the *log-log* data

complexity to consider a *fixed* set of optimal basis functions instead of a *learnable* set of non-optimal ones.

Thus, this validates the theoretical considerations and concludes our comparison based on model complexity, demonstrating how POD-DL-ROMs outperform any of the considered techniques when tackling more complex problems, for which the regularity of the parameter-to-solution map is low or unknown *a priori*.

5.3 2D nonlinear initial boundary value problem

This numerical experiment involves a nonlinear version of a time-dependent parametrized diffusion equation with a non-affine source term in an unconventional domain, cf. Fig. 8; the strong formulation of the problem at hand takes the form

$$\begin{cases} \dfrac{\partial u}{\partial t} - \nabla \cdot \left(0.001(1+u^2)\nabla u\right) = 0, & \text{in } \Omega \times (0, T] \\ u = 1 - e^{-100t} + h(x, y, \mu)e^{-100t}, & \text{on } \Gamma_D \times (0, T] \\ \dfrac{\partial u}{\partial n} = 0, & \text{on } \Gamma_N \times (0, T] \\ u_0 = h(x, y, \mu), & \text{in } \Omega, \end{cases}$$

where $T = 0.05$ and

- $h(x, y, \mu) = 0.1 + 10y\sin(\mu\pi x)$ represents a non-affine term, being $\mu \in \mathcal{P} = [5, 7]$ the parameter that regulates the spatial frequency of $h = h(x, y, \mu)$;
- letting $E_{a,b}(x, y)$ be the ellipse of axes a and b and center (x, y), we set $D_1 = E_{0.2,0.2}(0.5, 0.4)$ and $D_2 = E_{0.3,0.1}(1.0, 0.2)$; then, we can define the domain as $\Omega = (0, 1) \times (0, 0.4) \setminus (D_1 \cup D_2)$;
- the Dirichlet and the Neumann boundary are $\Gamma_D = \partial D_1 \cup \partial D_2$ and $\Gamma_N = \partial\Omega \setminus (\partial D_1 \cup \partial D_2)$, respectively.

Through this numerical experiment, we aim at verifying the *upper bound* and *lower bound* results presented in Section 3. To do so, we generate the training set and the test set input-output pairs through the numerical solution of the discretized problem

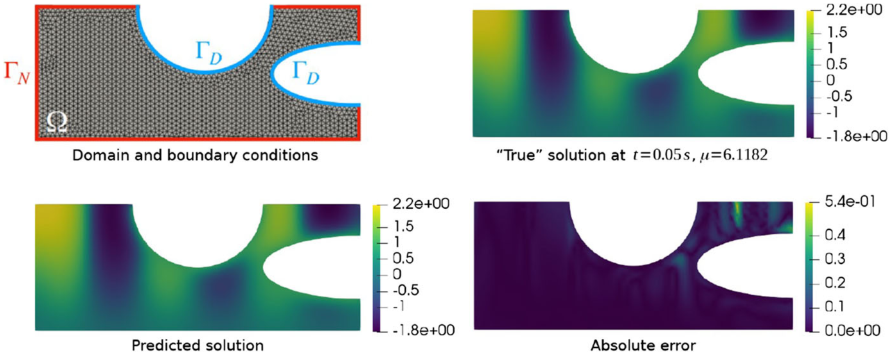


Fig. 8 2D IBVP test case: domain and boundary specifics (upper left), comparison between "true" solution (upper right) and POD-DL-ROM's predicted solution (lower left), and visualization of the absolute error (lower right), in the case of $N = 16$, which achieves the best accuracy with respect to the relative error metric

on a mesh of $N_h = 1666$ dofs by means of P1-FEM, employing a Forward Euler time-advancing scheme and the Newton method to handle nonlinearities. The training set is made by $N_s = 20$ samples relative to $\boldsymbol{\mu} \sim \mathcal{U}(\mathcal{P})$ iid of $N_t = 30$ snapshots each, sampling t from a uniformly space-time grid of step T/N_t. The test set data consist of $N_s^{test} = 30$ samples, evaluated on the same time grid employed in the training set.

Then, for each $N \in \{2^k \mid k = 0, \dots, 7\}$ we train a POD-DL-ROM of latent dimension $n = 2p + 1 = 5$, which is composed of:

- A reduced network of 3 hidden dense layers of 10 units each;
- An encoder and a decoder with 5 hidden dense layers of 25 units each.

We then evaluate the lower bound $\frac{m}{M}\tilde{\mathcal{E}}_{POD}$, the upper bound due to the error decomposition formula $\mathcal{E}_{NN} + \mathcal{E}_S + \mathcal{E}_{POD}$, the value relative error $\mathcal{E}_R$, according to the theoretical framework of Section 3.

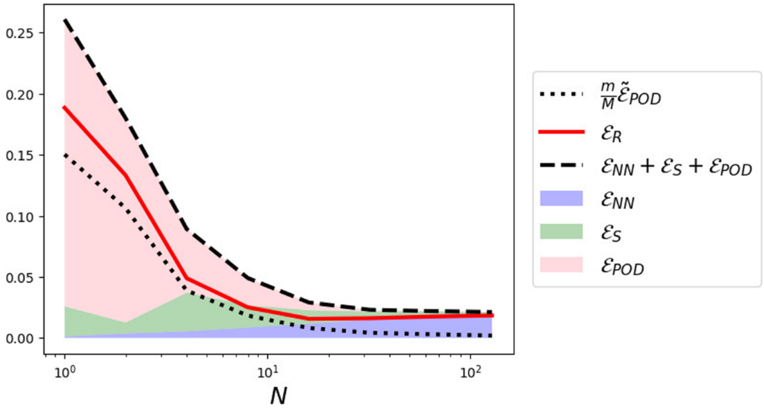


Fig. 9 2D IBVP test case: analysis of the error bounds varying the POD dimension N

We show both the lower bound and the upper bound results in Fig. 9, displaying as well the error contributions $\mathcal{E}_{NN}$, $\mathcal{E}_S$, $\mathcal{E}_{POD}$ to assess the way they affect the relative error $\mathcal{E}_R$. We then remark again that it is crucial for POD-DL-ROMs to provide both an adequate neural network approximation of the parameter-to-solution map and a suitably large POD dimension. Indeed, we notice that in the present test case, especially for low values of N, $\mathcal{E}_{NN}$ shows a marginal contribution to the upper bound value when compared to the sampling error $\mathcal{E}_S$ and the projection error $\mathcal{E}_{POD}$. On the other hand, as the POD dimension increases, learning higher-dimensional parameter-to-POD-coefficients maps becomes more burdensome: indeed, for larger values of N the majority of the upper bound value is explained by the contribution of $\mathcal{E}_{NN}$. Furthermore, as expected, we observe the strong dependence of the lower bound $\frac{m}{M}\tilde{\mathcal{E}}_{POD}$ on the POD dimension, demonstrating again the importance of choosing an adequate value for N. Finally, we assess *a posteriori* that the number of samples in the training set is suitable since the sampling error $\mathcal{E}_S$ does not heavily influence the upper bound of the relative error.

5.4 3D large-scale differential problem

As a final test case, we consider a real application concerning a heat exchanger device featuring: a complex 3D geometry (cf. Fig. 10), discontinuous boundary conditions, and a six-dimensional parameter space. In particular, we aim at reconstructing the temperature u of a laminar fluid flow, modeled as the solution to the following time-dependent advection-diffusion equation

$$\begin{cases} \dfrac{\partial u}{\partial t} - D\Delta u + \boldsymbol{v} \cdot \nabla u = 0, & \text{in } \Omega \times (0, T] \\ u = \displaystyle\sum_{j=1}^{3} g_j 1_{\Gamma_j}, & \text{on } \cup_{j=1}^{3} \Gamma_j \times (0, T] \\ u = 0, & \text{on } (\Gamma_{IN} \cup \Gamma_W) \times (0, T] \\ \nabla u \cdot n = 0, & \text{on } \Gamma_{OUT} \times (0, T] \\ u_0 = 0, & \text{in } \Omega, \end{cases} \tag{11}$$

where $T = 2$, whereas Γ_1, Γ_2, Γ_3 are the boundaries of the three baffles (cf. Fig. 10). Here, $D \in [0.01, 0.1]$ is the thermal diffusivity, while, for $i = 1, 2, 3$, $g_i \in [1, 11]$ is the temperature at Γ_i. Note that, due to $u \equiv 0$ on $\Gamma_{IN} \cup \Gamma_W$, the latter results in a discontinuous boundary condition. The transport field $\boldsymbol{v} \in \mathbb{R}^3$, instead, is obtained as solution of the steady incompressible Navier-Stokes equations in a low-*Re* regime

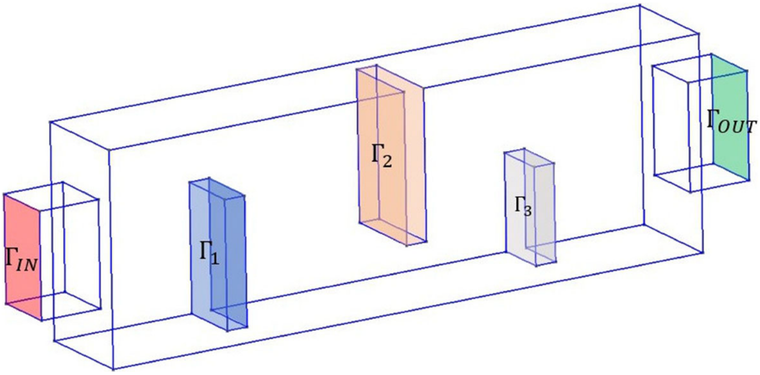


Fig. 10 3D large-scale test case: we highlight the domain boundaries of interest. Notice that Γ_W can be retrieved as set difference, namely, $\Gamma_W = \partial\Omega \setminus (\cup_{j=1}^{3}\Gamma_j \cup \Gamma_{IN} \cup \Gamma_{OUT})$

($Re \approx 50 \div 200$), namely

$$\begin{cases} -\nu\Delta\boldsymbol{v} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v} + \nabla p = \boldsymbol{0}, & \text{in } \Omega \\ \nabla\cdot\boldsymbol{v} = 0, & \text{in } \Omega \\ \boldsymbol{v} = [h(y,z;A),0,0], & \text{on } \Gamma_{IN} \\ \boldsymbol{v} = \boldsymbol{0}, & \text{on } \cup_{j=1}^{3}\Gamma_j \cup \Gamma_W \\ -p\boldsymbol{n} + \nu(\nabla\boldsymbol{v})\boldsymbol{n} = \boldsymbol{0}, & \text{on } \Gamma_{OUT}, \end{cases} \tag{12}$$

where $\nu \in [0.01, 0.02]$ is the fluid viscosity, while $A \in [1, 2]$ is a model parameter regulating the amplitude of the inlet profile

$$h(y, z; A) = 0.15^{-2} \cdot 16A(0.75 - y)(y - 0.25)(0.4 - z)(z - 0.1).$$

We collect $N_s = 40$ training samples, each consisting of $N_t = 60$ uniformly spaced timesteps, via the following four-step discretization procedure:

- First, in order to initialize the nonlinear solver for the Navier-Stokes equation, we solve the linearized version of (12) (namely, the Stokes equation obtained by removing the nonlinear term), via finite elements, using the $\mathbb{P}^1\text{b}-\mathbb{P}^1$ inf-sup stable pair;
- We then solve the Navier-Stokes equations with Newton iterations, obtaining an approximated velocity field $\boldsymbol{v}$ in the $\mathbb{P}^1$b space;
- We interpolate the Navier-Stokes velocity field onto the $\mathbb{P}^2$ space;
- We solve the time-dependent advection-diffusion problem using Forward Euler in time and $\mathbb{P}^2$ elements in space, thus yielding $N_h = 111942$ dofs.

Following the same procedure, we also collect $N_s^{test} = 10$ additional samples, which we use for testing purposes.

Since the fluid temperature depends on a 6 parameters, hereby represented by the vector $\boldsymbol{\mu} = [\nu, A, D, g_1, g_2, g_3] \in \mathbb{R}^6$, and (11) is time-dependent, we set the latent dimension to $n = 2 \cdot 6 + 3 = 15$, whereas we vary the reduced dimension $N \in \{2^i, \text{for } i = 1, \dots, 8\}$ in order to validate the lower and the upper bounds for different values of N. The reduced network of the POD-DL-ROM architecture entails 3 hidden layers of 50 units each, while both the encoder and the decoder feature 5 hidden layers of 30 neurons each.

Results are shown in Figs. 11 and 12. Once again, we see that the obtained approximations are in perfect agreement with our theory, with POD-DL-ROM always reporting errors within our theoretical error bounds. Concerning the different error contributions, we see a trend similar to the one observed for the other test cases. For small values of N, most of the error is caused by the POD block, as the complexity of our model problem cannot be replicated with few basis functions. On the other hand, if N is large, learning the POD coefficients becomes more challenging, and the error produced by the neural networks eventually dominates. Nonetheless, these results show that the POD-DL-ROM paradigm can provide an extremely accurate surrogate model, even for problems featuring discontinuous boundary conditions and complex 3D geometries.

Conclusions

The main goal of this work is to suggest effective and practical strategies to set a POD-DL-ROM stemming from a rigorous analysis of the technique, to control the approximation accuracy, measured in terms of the relative error $\mathcal{E}_R$, which is linked to relevant features and hyperparameters that can be effectively regulated. To accomplish the task, we analyze the error $\mathcal{E}_R$, providing a *lower bound* that depends only on the projection-based nature of the method. Then, by the *error decomposition* formula and the *upper bound* result, we highlight the contribution of sampling, POD projection, and neural network approximation; in particular:

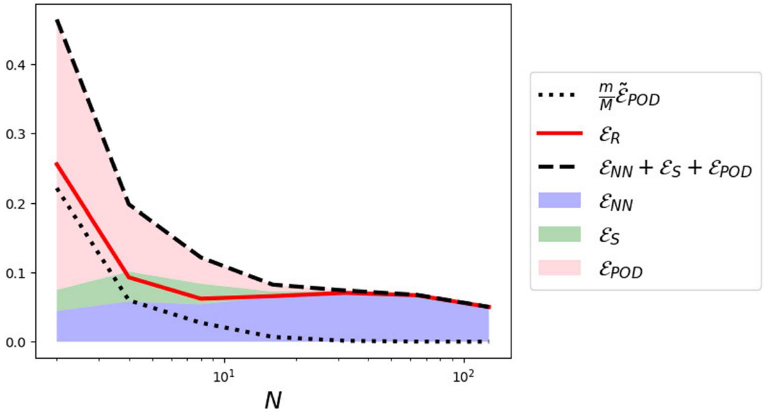


Fig. 11 3D large-scale test case: analysis of the error bounds for varying POD dimension N

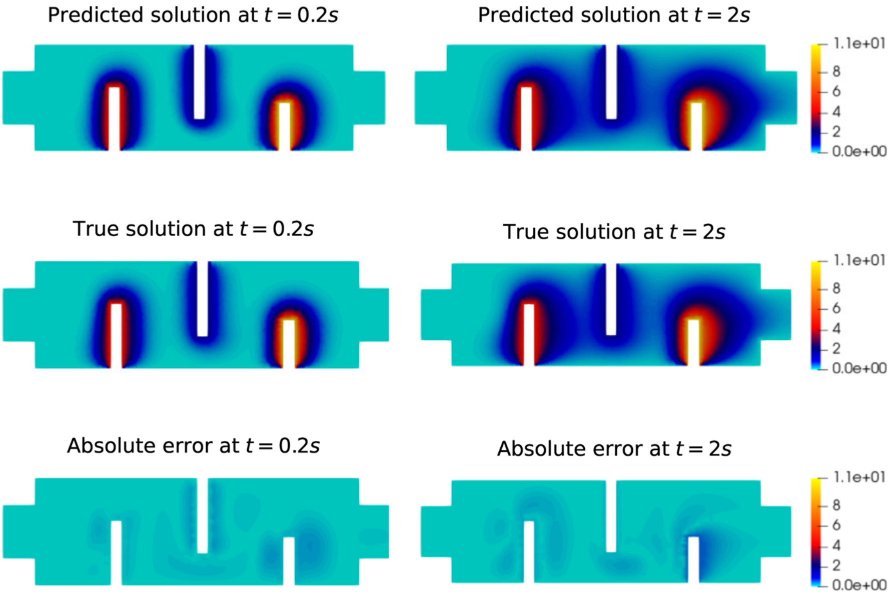


Fig. 12 3D large-scale test case: comparison between the high-fidelity solution and the POD-DL-ROM simulation for a test instance on the plane $z = 0.25$. We choose $N = 128$, which corresponds to the best model according to the relative error metric

(i) On the basis of the analysis of the sampling error $\mathcal{E}_S$ we propose a family of strategies to adopt in the data collection phase in order to ensure the convergence of $\mathcal{E}_S \to 0$ in the limit of infinite data, providing also a decay estimate through Monte Carlo analysis in terms of the number of sampled snapshots N_{data};
(ii) We determine a practical criterion based on the eigenvalue decay to control $\mathcal{E}_{POD}$ in terms of the reduced dimension N;
(iii) Starting from the approximation results proposed in [42], we estimate the complexity of the underlying neural network that is required to reach a given accuracy.

Then, relying on the aforementioned findings, we compare the POD-DL-ROM paradigm to other architectures that are widely used in the literature, namely DL-ROMs [10, 12, 17], POD+DNNs [6, 22, 38], POD-DeepONets [29], lin+ResNets [33] as well as CNNs [11], showing the strengths of the POD-DL-ROM strategy, especially when dealing with low-regularity maps. Ultimately, we demonstrate the outstanding approximation properties of POD-DL-ROMs, which motivate the excellent performance already encountered in a variety of test cases analyzed in the recent literature [15, 16] and in the present work. Several working directions could stem from the present paper; for instance, more efficient sampling criteria arising from Monte Carlo analysis could be implemented: we mention variance reduction techniques and quasi-Monte Carlo methods [3], among others. On the other hand, one could consider *ad hoc* layers to be employed in the reconstruction of parameter-to-POD-coefficients maps instead of relying purely on dense layers; however, this latter option would require novel and precise approximation results for the considered layers. Moreover, an alternative formulation could split the time- and the parameter-dependence, avoiding to treat time as an additional parameter, similarly to what has been proposed in [24], in order to further enhance the approximation bounds proposed in this paper.

Appendix A. Additional proofs

A.1 Proof of Proposition 1

Thanks to Assumption 1, trivially we obtain $\Delta t = TN_t^{-1} = O(N_t^{-1})$ and we set $t_i = i\Delta t$. Letting $f = f(\boldsymbol{\mu}, t)$ be the (sufficiently regular) integrand of the integral that we want to approximate, we obtain

$$\mathbb{E}\left|\int_{\mathcal{P}\times\mathcal{T}} f(\boldsymbol{\mu}, t)d(\boldsymbol{\mu}, t) - \frac{\Delta t|\mathcal{P}|}{N_s}\sum_{i=1}^{N_t}\sum_{j=1}^{N_s} f(\boldsymbol{\mu}_j, t_i)\right| \le I_1 + I_2,$$

where

$$I_1 = \left|\int_{\mathcal{P}\times\mathcal{T}} f(\boldsymbol{\mu}, t)d(\boldsymbol{\mu}, t) - \Delta t\sum_{i=1}^{N_t}\int_{\mathcal{P}} f(\boldsymbol{\mu}, t_i)d\boldsymbol{\mu}\right| = O(N_t^{-1})$$

and

$$\begin{aligned}
I_2 &= \Delta t\sum_{i=1}^{N_t}\mathbb{E}\left|\int_{\mathcal{P}} f(\boldsymbol{\mu}, t_i)d\boldsymbol{\mu} - \frac{|\mathcal{P}|}{N_s}\sum_{j=1}^{N_s} f(\boldsymbol{\mu}_j, t_i)\right| \\
&= O\left(N_s^{-1/2}\Delta t\sum_{i=1}^{N_t}(\mathrm{Var}(f(\mu, t_i)))^{1/2}\right) \\
&= O\left(N_s^{-1/2}\left(O(N_t^{-1}) + \int_{\mathcal{T}}\mathrm{Var}(f(\boldsymbol{\mu}, t))dt\right)\right) = O(N_s^{1/2})
\end{aligned}$$

Notice that

$$\int_{\mathcal{T}}\mathrm{Var}(f(\boldsymbol{\mu}, t)) < +\infty$$

because

$$\int_{\mathcal{T}}\left(\int_{\mathcal{P}} f(\boldsymbol{\mu}, t)^2 d\boldsymbol{\mu}\right)^{1/2} dt \le T^{1/2}\left(\int_{\mathcal{T}\times\mathcal{P}} f(\boldsymbol{\mu}, t)^2 d(\boldsymbol{\mu}, t)\right)^{1/2} < +\infty,$$

since $f \in L^2(\mathcal{P}\times\mathcal{T})$. Thus, the error we commit in approximating the integral goes to zero upon requiring $N_s, N_t \to \infty$. Finally, notice that

$$\frac{\Delta t|\mathcal{P}|}{N_s} = \frac{T|\mathcal{P}|}{N_{data}} = \frac{|\mathcal{P}\times\mathcal{T}|}{N_{data}},$$

which allows us to write

$$\mathbb{E}\left|\int_{\mathcal{P}\times\mathcal{T}} f(\boldsymbol{\mu},t)d(\boldsymbol{\mu},t) - \frac{|\mathcal{P}\times\mathcal{T}|}{N_{data}}\sum_{i=1}^{N_t}\sum_{j=1}^{N_s} f(\boldsymbol{\mu}_j,t_i)\right| \leq O(N_s^{-1/2}+N_t^{-1}).$$

A.2 Proof of Proposition 2

We notice immediately that the integral is well defined $\forall \mathbf{v} \in L^2(\mathcal{P}\times\mathcal{T};\mathbb{R}^{N_h})$ thanks to the boundedness assumptions on the solution $\mathbf{u} \in L^2(\mathcal{P}\times\mathcal{T};\mathbb{R}^{N_h})$. We also remark that the boundedness hypotheses may be relaxed: our choice was aimed at consistency with the other theoretical results of the present work. In order to prove that $\|\cdot\|_{L^2_w}$ is a norm, we have to show that:

(i) It satisfies the triangle inequality. Given $\mathbf{v}, \mathbf{z} \in L^2(\mathcal{P}\times\mathcal{T};\mathbb{R}^{N_h})$, by means of the triangular inequality, it is trivial to show that

$$\begin{aligned}
&\|\mathbf{v}+\mathbf{z}\|^2_{L^2_w} = \\
&= \int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{v}(\boldsymbol{\mu},t)+\mathbf{z}(\boldsymbol{\mu},t)\|^2 w(\boldsymbol{\mu},t)d(\boldsymbol{\mu},t) \leq \\
&\leq \int_{\mathcal{P}\times\mathcal{T}} (\|\mathbf{v}(\boldsymbol{\mu},t)\| + \|\mathbf{z}(\boldsymbol{\mu},t)\|)^2 w(\boldsymbol{\mu},t)d(\boldsymbol{\mu},t) = \\
&= \int_{\mathcal{P}\times\mathcal{T}} (\|\mathbf{v}(\boldsymbol{\mu},t)\|^2 + \|\mathbf{z}(\boldsymbol{\mu},t)\|^2 + 2\|\mathbf{v}(\boldsymbol{\mu},t)\|\|\mathbf{z}(\boldsymbol{\mu},t)\|) w(\boldsymbol{\mu},t)d(\boldsymbol{\mu},t).
\end{aligned}$$

Moreover, by the Cauchy-Schwarz inequality, the following inequality holds,

$$\begin{aligned}
&\int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{v}(\boldsymbol{\mu},t)\|\|\mathbf{z}(\boldsymbol{\mu},t)\| w(\boldsymbol{\mu},t)d(\boldsymbol{\mu},t) \leq \\
&\leq \sqrt{\int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{v}(\boldsymbol{\mu},t)\|^2 w(\boldsymbol{\mu},t)d(\boldsymbol{\mu},t) \int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{z}(\boldsymbol{\mu},t)\|^2 w(\boldsymbol{\mu},t)d(\boldsymbol{\mu},t)}.
\end{aligned}$$

Thus, we can infer

$$\|\mathbf{v}+\mathbf{z}\|^2_{L^2_w} \leq \|\mathbf{v}\|^2_{L^2_w} + \|\mathbf{z}\|^2_{L^2_w} + 2\|\mathbf{v}\|_{L^2_w}\|\mathbf{z}\|_{L^2_w} = (\|\mathbf{v}\|_{L^2_w} + \|\mathbf{z}\|_{L^2_w})^2$$

and derive the thesis;

(ii) $\|\cdot\|_{L^2_w}$ is homogeneous thanks to the linearity of the integral;

(iii) If $\mathbf{v} \in L^2(\mathcal{P}\times\mathcal{T};\mathbb{R}^{N_h})$, $\|\mathbf{v}\|_{L^2_w} = 0$ implies that $\mathbf{v} = \mathbf{0}$ a.e. by trivial arguments.

A.3 Quasi-optimality of the discrete formulation of the POD decomposition

We base the following analysis on the results of the $(\mathcal{P}\times\mathcal{T})$-continuous problem proposed in [36]. We first recall that by definition $\mathbf{V}_\infty \in \mathbb{R}^{N_h \times N}$ (where N is the

POD dimension) is optimal for the $(\mathcal{P} \times \mathcal{T})$-continuous formulation, that is with respect to the $L^2(\mathcal{P} \times \mathcal{T}; \mathbb{R}^{N_h})$ norm. Formally, we set $\delta, \varepsilon > 0$ and, by assuming $\mathbf{u}(\boldsymbol{\mu}, t) \in L^2(\mathcal{P} \times \mathcal{T}, \mathbb{R}^{N_h})$, we define $T : L^2(\mathcal{P} \times \mathcal{T}) \to \mathbb{R}^{N_h}$ as

$$Tg := \int_{\mathcal{P}\times\mathcal{T}} \mathbf{u}(\boldsymbol{\mu}, t) g(\boldsymbol{\mu}, t) d(\boldsymbol{\mu}, t) \quad \forall g \in L^2(\mathcal{P} \times \mathcal{T}).$$

The adjoint operator of T, namely T^*, enjoys the property

$$T^*\mathbf{w} = (\mathbf{u}(\boldsymbol{\mu}, t), \mathbf{w})_2 \qquad \forall \mathbf{w} \in \mathbb{R}^{N_h}.$$

Moreover, recall the definition of the (continuous) correlation matrix (3) and denote by $(\sigma_{k,\infty}^2, \boldsymbol{\zeta}_k)$ its eigenpairs (where $\{\boldsymbol{\zeta}_k\}_k$ denotes an orthonormal basis). We thus define the HS-norm of T as

$$\|T\|_{HS} = \sqrt{\sum_{k \leq \text{rank}(T)} \sigma_{k,\infty}^2}.$$

Setting

$$\boldsymbol{\xi}_k = \frac{1}{\sigma_{k,\infty}} T^* \boldsymbol{\zeta}_k \qquad \forall k = 1, \ldots, N_h,$$

we denote by $T_{N,\infty}$ the rank-N Schmidt approximation, with

$$T_{N,\infty} = \sum_{k=1}^{N} \sigma_{k,\infty} \boldsymbol{\zeta}_k (\boldsymbol{\xi}_k(\boldsymbol{\mu}, t), \cdot)_{L^2(\mathcal{P}\times\mathcal{T})} = \mathbf{V}_\infty \mathbf{V}_\infty^T T.$$

and by $T_N = \mathbf{V}\mathbf{V}^T T$ its approximation by means of the discrete POD formulation. *Theorem 6.2* and *Proposition 6.3* in [36] show that the rank-N Schmidt operator and therefore the set of basis $\mathbf{V}_\infty$ are optimal with respect to the HS-norm, namely they retain the most variability. Formally:

$$\begin{aligned} \|T_{N,\infty} - T\|_{HS} &= \min_{B \in \mathcal{B}_N} \|B - T\|_{HS} \\ &= \min_{\mathbf{W} \in \mathbb{R}^{N_h \times N} : \mathbf{W}^T\mathbf{W} = \boldsymbol{I}} \left(\int_{\mathcal{P}\times\mathcal{T}} \|\mathbf{u}(\boldsymbol{\mu}, t) - \mathbf{W}\mathbf{W}^T \boldsymbol{q}(\boldsymbol{\mu}, t)\|^2 d(\boldsymbol{\mu}, t) \right)^{1/2} \\ &= \sqrt{\sum_{k>N} \sigma_{k,\infty}^2} \\ &= m\mathcal{E}_{POD,\infty}, \end{aligned} \tag{13}$$

where $\mathcal{B}_N = \{B \in \mathcal{L}(L^2(\mathcal{P} \times \mathcal{T}); \mathbb{R}^{N_h})) : \text{rank}(B) \leq N \wedge \|B\|_{HS} < +\infty\}$, being $\mathcal{L}(U)$ the space of linear continuous operators from U to U, for U Banach. Now, suppose to define $B_N \in \mathcal{B}_N$ which does not attain the minimum in (13), thus

$$0 < 2\varepsilon_{\max} := \|B_N - T\|_{HS} - \|T_{N,\infty} - T\|_{HS}. \tag{14}$$

By means of the results of Theorem 1, with the same hypotheses, we have that

$$\begin{aligned}\|T_{N,\infty} - T\|_{HS} &\leq \|T_N - T\|_{HS} \\ &\leq m(\mathcal{E}_S + \mathcal{E}_{POD}) \xrightarrow[N_s, N_t \to \infty]{a.s.} m\mathcal{E}_{POD,\infty} = \|T_{N,\infty} - T\|_{HS}.\end{aligned}$$

Thus, since a.s. convergence implies convergence in probability, we derive that

$$\begin{aligned}&\forall \delta > 0, \quad \forall 0 < \varepsilon < \varepsilon_{\max}, \quad \exists N_s, N_t : \\ &\qquad \mathbb{P}\left\{ \|T_N - T\|_{HS} - \|T_{N,\infty} - T\|_{HS} < \varepsilon \right\} > 1 - \delta.\end{aligned}$$

Finally, thanks to (14), we have

$$\forall \delta > 0, \quad \exists N_s, N_t : \mathbb{P}\left\{ \|B_N - T\|_{HS} - \|T_N - T\|_{HS} > \varepsilon_{\max} \right\} > 1 - \delta.$$

Acknowledgements The authors are members of the Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica (GNCS-INdAM) and acknowledge the project "Dipartimento di Eccellenza" 2023-2027, funded by MUR, as well as the support of Fondazione Cariplo, Italy, Grant n. 2019-4608. AM acknowledges the PRIN 2022 Project "Numerical approximation of un- certainty quantification problems for PDEs by multi-fidelity methods (UQ-FLY)" (No. 202222PACR), funded by the European Union - NextGenerationEU. AM and SF acknowledge the project FAIR (Future Artificial Intelligence Research), funded by the NextGenerationEU program within the PNRR-PE-AI scheme (M4C2, Investment 1.3, Line on Artificial Intelligence). SF also acknowledges the Isaac Newton Institute for Mathematical Sciences, Cambridge, UK, for support and hospitality during the program "The mathematical and statistical foundation of future data-driven engineering," EPSRC grant no EP/R014604, where part of this work was undertaken.

Funding Open access funding provided by Politecnico di Milano within the CRUI-CARE Agreement.

Data Availability The source code implementation of the method described in the paper is made available from the GitHub repository: https://github.com/DLROM-hub/poddlrom-error-estimates together with a sample numerical experiment to showcase a possible use.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Barrault, M., Maday, Y., Nguyen, N.C., Patera, A.T.: An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. C.R. Math. **339**(9), 667–672 (2004). https://doi.org/10.1016/j.crma.2004.08.006
2. Bhattacharya, K., Hosseini, B., Kovachki, N.B., Stuart, A.M.: Model reduction and neural networks for parametric PDEs. The SMAI J. Comput. Math. **7**, 121–157 (2021). https://doi.org/10.5802/smai-jcm.74
3. Caflisch, R.E.: Monte Carlo and quasi-Monte Carlo methods. Acta Numer. **7**, 1–49 (1998). https://doi.org/10.1017/S0962492900002804
4. Chaturantabut, S., Sorensen, D.C.: Nonlinear model reduction via discrete empirical interpolation. SIAM J. Sci. Comput. **32**(5), 2737–2764 (2010). https://doi.org/10.1137/090766498
5. Chen, R.T.Q., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. In: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (eds.) Advances in Neural Information Processing Systems, vol. 31. Curran Associates, Inc. (2018)
6. Chen, W., Wang, Q., Hesthaven, J.S., Zhang, C.: Physics-informed machine learning for reduced-order modeling of nonlinear problems. J. Comput. Phys. **446**, 110666 (2021). https://doi.org/10.1016/j.jcp.2021.110666
7. Deng, B., Shin, Y., Lu, L., Zhang, Z., Karniadakis, G.E.: Approximation rates of DeepONets for learning operators arising from advection-diffusion equations. Neural Netw. **153**, 411–426 (2022). https://doi.org/10.1016/j.neunet.2022.06.019
8. DeVore, R.A., Howard, R., Micchelli, C.: Optimal nonlinear approximation. Manuscripta Math. **63**, 469–478 (1989). https://doi.org/10.1007/BF01171759
9. Farhat, C., Grimberg, S., Manzoni, A., Quarteroni, A.: Computational bottlenecks for proms: precomputation and hyperreduction. In: P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, L.M. Silveira (eds.) Volume 2: Snapshot-Based Methods and Algorithms, pp. 181–244. De Gruyter, Berlin, Boston (2020). https://doi.org/10.1515/9783110671490-005
10. Franco, N., Manzoni, A., Zunino, P.: A deep learning approach to reduced order modelling of parameter dependent partial differential equations. Math. Comput. **92**(340), 483–524 (2023). https://doi.org/10.1090/mcom/3781
11. Franco, N.R., Fresca, S., Manzoni, A., Zunino, P.: Approximation bounds for convolutional neural networks in operator learning. Neural Netw. **161**, 129–141 (2023). https://doi.org/10.1016/j.neunet.2023.01.029
12. Fresca, S., Dedè, L., Manzoni, A.: A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized PDEs. Journal of Scientific Computing **87**(61) (2021). https://doi.org/10.1007/s10915-021-01462-7
13. Fresca, S., Fatone, F., Manzoni, A.: Long-time prediction of nonlinear parametrized dynamical systems by deep learning-based ROMs. In: NIPS Workshop The Symbiosis of Deep Learning and Differential Equations (2021)
14. Fresca, S., Gobat, G., Fedeli, P., Frangi, A., Manzoni, A.: Deep learning-based reduced order models for the real-time simulation of the nonlinear dynamics of microstructures. Int. J. Numer. Meth. Eng. **123**(20), 4749–4777 (2022). https://doi.org/10.1002/nme.7054
15. Fresca, S., Manzoni, A.: Real-time simulation of parameter-dependent fluid flows through deep learning-based reduced order models. Fluids **6**(7) (2021). https://doi.org/10.3390/fluids6070259
16. Fresca, S., Manzoni, A.: POD-DL-ROM: enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition. Comput. Methods Appl. Mech. Eng. **388**, 114181 (2022). https://doi.org/10.1016/j.cma.2021.114181
17. Fresca, S., Manzoni, A., Dedè, L., Quarteroni, A.: Deep learning-based reduced order models in cardiac electrophysiology. PLOS One **15**(10) (2020). https://doi.org/10.1371/journal.pone.0239416
18. Fresca, S., Manzoni, A., Dedè, L., Quarteroni, A.: Pod-enhanced deep learning-based reduced order models for the real-time simulation of cardiac electrophysiology in the left atrium. Frontiers in physiology p. 1431 (2021)
19. Geist, M., Petersen, P., Raslan, M., Schneider, R., Kutyniok, G.: Numerical solution of the parametric diffusion equation by deep neural networks. J. Sci. Comput. **88**(1), 22 (2021)
20. Gühring, I., Kutyniok, G., Petersen, P.: Error bounds for approximations with deep ReLU neural networks in Ws, p norms. Anal. Appl. **18**(05), 803–859 (2020). https://doi.org/10.1142/S0219530519410021

21. Gühring, I., Raslan, M.: Approximation rates for neural networks with encodable weights in smoothness spaces. Neural Netw. **134**, 107–130 (2021). https://doi.org/10.1016/j.neunet.2020.11.010
22. Hesthaven, J., Ubbiali, S.: Non-intrusive reduced order modeling of nonlinear problems using neural networks. J. Comput. Phys. **363**, 55–78 (2018). https://doi.org/10.1016/j.jcp.2018.02.037
23. Jacod, J., Protter, P.: Probability essentials. Springer Berlin, Heidelberg (2004). https://doi.org/10.1007/978-3-642-55682-1
24. Jin, P., Meng, S., Lu, L.: MIONet: learning multiple-input operators via tensor product. SIAM J. Sci. Comput. **44**, A3490–A3514 (2022). https://doi.org/10.1137/22M1477751
25. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Y. Bengio, Y. LeCun (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015). arXiv:1412.6980
26. Lanthaler, S., Mishra, S., Karniadakis, G.E.: Error estimates for DeepONets: a deep learning framework in infinite dimensions. Transactions of Mathematics and Its Applications **6**(1), tnac001 (2022). https://doi.org/10.1093/imatrm/tnac001
27. Lee, K., Carlberg, K.T.: Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. J. Comput. Phys. **404**, 108973 (2020). https://doi.org/10.1016/j.jcp.2019.108973
28. Lu, L., Jin, P., Pang, G., Zhang, Z., Karniadakis, G.E.: Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. Nature Machine Intelligence **3**(3), 218–229 (2021). https://doi.org/10.1038/s42256-021-00302-5
29. Lu, L., Meng, X., Cai, S., Mao, Z., Goswami, S., Zhang, Z., Karniadakis, G.E.: A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data. Comput. Methods Appl. Mech. Eng. **393**, 114778 (2022). https://doi.org/10.1016/j.cma.2022.114778
30. Mücke, N.T., Bohté, S.M., Oosterlee, C.W.: Reduced order modeling for parameterized time-dependent PDEs using spatially and memory aware deep learning. J. Comput. Sci. **53**, 101408 (2021). https://doi.org/10.1016/j.jocs.2021.101408
31. Mishra, S., Molinaro, R.: Estimates on the generalization error of physics-informed neural networks for approximating PDEs. IMA J. Numer. Anal. **43**(1), 1–43 (2022). https://doi.org/10.1093/imanum/drab093
32. Niederreiter, H.: Random number generation and quasi-Monte Carlo methods. SIAM (1992)
33. O'Leary-Roseberry, T., Du, X., Chaudhuri, A., Martins, J.R., Willcox, K., Ghattas, O.: Learning high-dimensional parametric maps via reduced basis adaptive residual networks. Comput. Methods Appl. Mech. Eng. **402**, 115730 (2022). https://doi.org/10.1016/j.cma.2022.115730
34. Pant, P., Doshi, R., Bahl, P., Farimani, A.B.: Deep learning for reduced order modelling and efficient temporal evolution of fluid simulations. Phys. Fluids **33**(10), 107101 (2021). https://doi.org/10.1063/5.0062546
35. Quarteroni, A.: Numerical models for differential problems. Springer Cham (2017). https://doi.org/10.1007/978-3-319-49316-9
36. Quarteroni, A., Manzoni, A., Negri, F.: Reduced basis methods for partial differential equations. Springer Cham (2016). https://doi.org/10.1007/978-3-319-15431-2
37. Quarteroni, A., Sacco, R., Saleri, F., Gervasio, P.: Matematica numerica. Springer Milano (2014). https://doi.org/10.1007/978-88-470-5644-2
38. Salvador, M., Dede, L., Manzoni, A.: Non intrusive reduced order modeling of parametrized PDEs by kernel pod and neural networks. Comput. Math. Appl. **104**, 1–13 (2021). https://doi.org/10.1016/j.camwa.2021.11.001
39. Schwab, C., Todor, R.A.: Karhunen-Loéve approximation of random fields by generalized fast multipole methods. J. Comput. Phys. **217**(1), 100–122 (2006). https://doi.org/10.1016/j.jcp.2006.01.048
40. Szlam, A., Kluger, Y., Tygert, M.: An implementation of a randomized algorithm for principal component analysis (2014). arXiv:1412.3510v1
41. Wang, Q., Hesthaven, J.S., Ray, D.: Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem. J. Comput. Phys. **384**, 289–307 (2019). https://doi.org/10.1016/j.jcp.2019.01.031
42. Yarotsky, D.: Error bounds for approximations with deep ReLU networks. Neural Netw. **94**, 103–114 (2017). https://doi.org/10.1016/j.neunet.2017.07.002
43. Yarotsky, D.: Optimal approximation of continuous functions by very deep ReLU networks. In: S. Bubeck, V. Perchet, P. Rigollet (eds.) Proceedings of the 31st Conference On Learning Theory,

Proceedings of Machine Learning Research, vol. 75, pp. 639–649. PMLR (2018). https://proceedings.mlr.press/v75/yarotsky18a.html

44. Zahm, O., Constantine, P.G., Prieur, C., Marzouk, Y.M.: Gradient-based dimension reduction of multivariate vector-valued functions. SIAM J. Sci. Comput. **42**(1), A534–A558 (2020). https://doi.org/10.1137/18M1221837
45. Zhu, Y., Zabaras, N.: Bayesian deep convolutional encoder-decoder networks for surrogate modeling and uncertainty quantification. J. Comput. Phys. **366**, 415–447 (2018). https://doi.org/10.1016/j.jcp.2018.04.018

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Simone Brivio[1] · **Stefania Fresca**[1] · **Nicola Rares Franco**[1] · **Andrea Manzoni**[1]

Simone Brivio
simone.brivio@polimi.it

Stefania Fresca
stefania.fresca@polimi.it

Nicola Rares Franco
nicolarares.franco@polimi.it

[1] MOX - Dipartimento di Matematica, Politecnico di Milano, P.zza Leonardo da Vinci 32, I-20133 Milano, Italy