MOX-Report No. 47/2024

A practical existence theorem for reduced order models based on convolutional autoencoders

Franco, N.R.; Brugiapaglia, S.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it https://mox.polimi.it

A practical existence theorem for reduced order models based on convolutional autoencoders

Nicola Rares Franco[1] and Simone Brugiapaglia[2]

[1]MOX, Department of Mathematics, Politecnico di Milano, Italy
[2]Department of Mathematics and Statistics, Concordia University, Montreal, QC, Canada

Abstract

In recent years, deep learning has gained increasing popularity in the fields of Partial Differential Equations (PDEs) and Reduced Order Modeling (ROM), providing domain practitioners with new powerful data-driven techniques such as Physics-Informed Neural Networks (PINNs), Neural Operators, Deep Operator Networks (DeepONets) and Deep-Learning based ROMs (DL-ROMs). In this context, deep autoencoders based on Convolutional Neural Networks (CNNs) have proven extremely effective, outperforming established techniques, such as the reduced basis method, when dealing with complex nonlinear problems. However, despite the empirical success of CNN-based autoencoders, there are only a few theoretical results supporting these architectures, usually stated in the form of universal approximation theorems. In particular, although the existing literature provides users with guidelines for designing convolutional autoencoders, the subsequent challenge of learning the latent features has been barely investigated. Furthermore, many practical questions remain unanswered, e.g., the number of snapshots needed for convergence or the neural network training strategy. In this work, using recent techniques from sparse high-dimensional function approximation, we fill some of these gaps by providing a new *practical existence theorem* for CNN-based autoencoders when the parameter-to-solution map is holomorphic. This regularity assumption arises in many relevant classes of parametric PDEs, such as the parametric diffusion equation, for which we discuss an explicit application of our general theory.

1 Introduction

Scientists and engineers rely on Partial Differential Equations (PDEs) to model and describe physical phenomena characterizing the behavior of systems, materials, and processes. In tandem with efficient numerical solvers, PDE modeling allows engineers to generate robust simulations of physical systems, effectively providing them with reliable tools for forecasting, design, and optimization.

In practical applications, PDE models often involve multiple parameters, which here we denote as $\boldsymbol{\mu} \in \mathbb{R}^p$, that describe the physical properties of the system and/or specify the scenario under consideration. We can think of, e.g., the viscosity coefficient in a fluid flow simulation [48], the morphology of a vascular network

in a biophysical model [53], or the permeability coefficient in a heat-transfer simulation [9]. When these parameter values remain constant, traditional numerical solvers based on, e.g., finite elements, finite differences, or finite volumes, can provide precise and reliable approximations at a computationally feasible expense. However, there are also applications where the model parameters are allowed to change and thus necessitate multiple—fast—simulations. Examples include optimal control (i.e., find the $\boldsymbol{\mu}$ minimizing a given cost functional), inverse problems (i.e., retrieve $\boldsymbol{\mu}$ from sensor measurements), and uncertainty quantification (i.e., $\boldsymbol{\mu}$ is uncertain). For all such *many-query* scenarios, the computational cost entailed by classical solvers becomes prohibitive and constitutes a major limitation.

A popular solution to these issues is provided by Reduced Order Models (ROMs). They are suitable model surrogates that seek to alleviate the computational burden associated with the aforementioned tasks by constructing low-dimensional representations that are rich enough to capture the essential features of the system. By learning from high-quality samples generated by classical solvers, ROMs can offer precise and efficient predictions, effectively restoring the feasibility of real-time simulations and their applicability to many-query scenarios. However, depending on the problem at hand, constructing accurate and reliable ROMs can be a challenging task. In fact, complex model features such as high-dimensional parameter spaces, strong nonlinearities and singular behaviors, pose significant challenges. This is well illustrated by all those problems facing the so-called *Kolmogorov barrier* [6, 7, 45].

Recently, motivated by the impressive success of deep learning in a variety of fields including image recognition, natural language processing, and scientific computing, researchers have attempted to leverage this methodology to construct effective ROMs, leading to the development of Deep Learning-based ROMs (DL-ROMs) [21, 25, 47]. Experimentally, these techniques have obtained quite remarkable results. If provided with enough data and properly trained, DL-ROMs can accurately simulate fluid flows [24], as well as complex biological [22, 53] and mechanical phenomena [50]. Furthermore, if complemented with suitable *ad hoc* strategies, they can incorporate some key physical properties of the underlying system such as local mass conservation [8].

In general, this research line is part of a broader trend concerning the development of deep learning algorithms for operator learning in high-dimensional spaces: in fact, a surrogate model can be interpreted as an approximation of the parameter-to-solution map of a given PDE. In this sense, there is a close connection between DL-ROMs and techniques such as DeepONets [39] and (Fourier) Neural Operators [31, 37]. For instance, the DeepONet algorithm can be regarded as a space-continuous version of the so-called POD-NN ROM [28], a predecessor of the POD-DL-ROM [24]. Similarly, the architectures implemented in some DL-ROMs can be traced back to discrete equivalents of certain Neural Operators [19, 22]. Here, however, we shall limit our attention to the case of DL-ROMs, adopting a perspective commonly accepted in the ROM literature.

At first, the success of deep learning in reduced order modeling was mostly empirical (see, e.g., [25, 28, 36]). However, with new mathematical insights on neural

network approximation theory, such as the seminal paper [54] and subsequent developments (see, e.g., [16] and references therein), DL-ROMs are now starting to develop theoretical foundations. Relevant contributions in this direction include [32, 40, 51], which are theoretical works characterized by a major focus on approximation theory (practical details concerning networks type or training strategies are not addressed), research on DeepONets [33] and Neural Operators [30], and recent results on autoencoder-based ROMs (see, e.g., [9, 19, 21, 38]).

Here, we shall focus on the latter class of DL-ROMs, i.e., deep learning-based surrogate models that reduce the problem complexity by leveraging deep convolutional autoencoders [10]. This choice is motivated by the fact that, despite being extremely popular among researchers, ROMs based on convolutional autoencoders are still lacking a comprehensive theoretical foundation. While issues like the role of convolutional blocks [20] or the choice of the latent dimension [19, 21] are relatively well-understood, some key practical questions are still open, especially when it comes to the actual training of these architectures. Our purpose for this work is to take a step further and extend the existing literature by offering additional insights on DL-ROM training, with a particular focus on the challenge of learning convolutional features. To this end, we propose a new analysis of these architectures based on the framework of *practical existence theorems*. This new paradigm was recently introduced in [1, 5] for scalar- and Hilbert-valued approximation and further extended to Banach-valued functions in [2] (see also the recent review paper [3]). It leverages recent advances in sparse high-dimensional polynomial approximation theory [4] and complements existence results for neural networks (commonly referred to as *universal approximation theorems*) with more practical insights on model training, regularization and sampling.

1.1 Main contributions

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain, and let

$$\Theta \ni \boldsymbol{\mu} \mapsto u_{\boldsymbol{\mu}} \in H^s(\Omega)$$

be the parameter-to-solution map of a parametrized PDE, where $\Theta \subset \mathbb{R}^p$ is the parameter space and $H^s(\Omega)$ denotes a suitable Sobolev space with smoothness index $s \in \mathbb{N}$. A classical numerical solver based on, e.g., finite elements or finite differences, provides access to pointwise approximations of the PDE solution over a collection of nodes $\mathbf{x}_1, \dots, \mathbf{x}_{N_h} \in \overline{\Omega}$, with N_h being the total number of vertices constituting the spatial grid.

Therefore, we can think of the numerical solver, also referred to as Full Order Model (FOM) in the reduced order modeling literature, as a map

$$\Theta \ni \boldsymbol{\mu} \mapsto [u_{\boldsymbol{\mu}}(\mathbf{x}_1), \dots, u_{\boldsymbol{\mu}}(\mathbf{x}_{N_h})]^\top \in \mathbb{R}^{N_h}.$$

The purpose of DL-ROMs is to construct a Deep Neural Network (DNN) model $\Phi : \mathbb{R}^p \to \mathbb{R}^{N_h}$ such that $\Phi_j(\boldsymbol{\mu}) \approx u_{\boldsymbol{\mu}}(\mathbf{x}_j)$, where Φ_j denotes the jth output neuron of Φ. In the case of autoencoder-based approaches, the construction of Φ relies on three neural network models, namely,

$$\Psi' : \mathbb{R}^{N_h} \to \mathbb{R}^m, \qquad \Psi : \mathbb{R}^m \to \mathbb{R}^{N_h},$$

$$\phi : \mathbb{R}^p \to \mathbb{R}^m.$$

The first two models, the encoder and the decoder, respectively, are trained such that

$$\Psi(\Psi'([u_{\boldsymbol{\mu}}(\mathbf{x}_1), \dots, u_{\boldsymbol{\mu}}(\mathbf{x}_{N_h})]^\top)) \approx [u_{\boldsymbol{\mu}}(\mathbf{x}_1), \dots, u_{\boldsymbol{\mu}}(\mathbf{x}_{N_h})]^\top.$$

In this way, by leveraging the autoencoder $\Psi \circ \Psi'$, the spatial features characterizing the solutions to the PDE can be synthesized using a smaller number of degrees of freedom, known as the "latent" variables. In fact, each discrete vector $\mathbf{u}_{\boldsymbol{\mu}} := [u_{\boldsymbol{\mu}}(\mathbf{x}_1), \dots, u_{\boldsymbol{\mu}}(\mathbf{x}_{N_h})]^\top$ can now be represented as $\Psi'(\mathbf{u}_{\boldsymbol{\mu}}) \in \mathbb{R}^m$, with a substantial reduction in complexity whenever $m \ll N_h$.

Conversely, the third network, ϕ, sometimes also referred to as *reduced network*, is trained to learn the parameter-to-latent-variables map,

$$\phi(\boldsymbol{\mu}) \approx \Psi'([u_{\boldsymbol{\mu}}(\mathbf{x}_1), \dots, u_{\boldsymbol{\mu}}(\mathbf{x}_{N_h})]^\top).$$

Once all DNN modules have been trained, the encoder block Ψ' can be discarded and the DL-ROM constructed by composition, i.e.,

$$\Phi := \Psi \circ \phi.$$

Given a new parametric instance $\boldsymbol{\mu} \in \Theta$, the reduced network computes the corresponding latent solution, namely $\phi(\boldsymbol{\mu})$, which is then expanded by the decoder to retrieve the final output. In other words, methods based on autoencoders are grounded on the idea of splitting the complexity of the problem into two components. On the one hand, we have the spatial complexity of PDE solutions, tackled by Ψ' and Ψ. On the other hand, there is the inherent complexity associated with the parameter dependence of PDE solutions, addressed by ϕ.

The existing literature provides insights on the choice of the latent dimension m (see, e.g., [21]) and on the type of architectures, favoring the use of Convolutional Neural Networks (CNNs)—whenever possible—for the decoder module (see, e.g., [20]). It is worth mentioning that, while the works [20, 21] are purely theoretical, their conclusions are perfectly aligned with empirical evidence. In fact, researchers had long conjectured that autoencoders could compress solutions to their intrinsic dimension, dictated by the number of parameters [25, 36]. Similarly, by leveraging the heuristic observation that discrete signals defined over hypercubic domains are roughly equivalent to RGB images, several authors had suggested the use of CNNs for the autoencoder module [25, 43]. Nonetheless, little is know about the training of these architectures, in terms of, e.g., sample size and choice of the loss function. Our main contribution, which is fully detailed in Theorem 1, goes precisely in this direction.

Given a probability distribution ϱ over the parameter space, for each of the three architectures, ϕ, Ψ and Ψ', we identify a specific class of neural network models, with the decoder Ψ being convolutional, such that, with high probability, the trained DL-ROM satisfies an error bound of the form

$$\mathbb{E}^{1/2}_{\boldsymbol{\mu}\sim\varrho}\left[\sup_{j=1,\dots,N_h} |u_{\boldsymbol{\mu}}(\mathbf{x}_j) - \Psi_j(\phi(\boldsymbol{\mu}))|^2\right] \leq C\left(\sqrt{m}e^{-\frac{1}{\sqrt{2}}\gamma\tilde{N}^{1/(2p)}} + \sqrt{\frac{2m^{1-2s}}{2s-1}}\right),$$

where m is (proportional to) the latent dimension, $\tilde{N}$ is the sample size (up to log factors), whereas $\gamma > 0$ and $C > 0$ are two constants related to the regularity and the magnitude of the solution operator, respectively; finally, we recall, that p and s are the number of parameters and the smoothness of the PDE solutions, respectively. In doing so, we also specify the sampling and the optimization procedure associated with the reduced network ϕ, identifying a specific loss function and a corresponding regularization criterion, thus making our existence result *practical*. Note that, although based on the same ideas presented in [1], our result is somewhat stronger. In fact, with respect to the space variable $\mathbf{x}$, it provides a uniform error bound, as opposed to a space-averaged one.

At its core, our derivation leverages the theory of sparse polynomial approximation of high-dimensional, holomorphic maps, and thus relies on the assumption that the parameter-to-solution map admits a suitable holomorphic extension. However, as we will explore later, this assumption is not overly restrictive. In fact, there are numerous practical cases where this condition holds true, such as the parametric diffusion equation, for which an application of Theorem 1 is explicitly discussed in Section 3.1. Finally, we mention that our analysis is limited to the one-dimensional case, $d = 1$. Generalizations to higher-dimensional domains are in order but out of the scope of this work.

1.2 Outline

The paper is organized as follows. First, in Section 2, we set the notation and introduce some of the basic mathematical concepts upon which our analysis in constructed, such as holomorphic extensions and neural network models. Then, in Section 3 we present our main result, Theorem 1, and its application to the parametric diffusion equation. The proof of the theorem, which is comprised of multiple steps, is postponed to Section 4, together with some auxiliary results that are necessary for our construction (only some of them: the most technical ones are deferred to Appendix A and B). Lastly, Section 5 is dedicated to a final discussion of our findings and potential avenues for future research.

2 Preliminaries and notation

In this section we introduce the main notions and definitions needed to carry out our analysis. In Section 2.1, we introduce the concepts of holomorphic extension and of hidden anisotropy. Section 2.2, instead, provides the essential background on feedforward and convolutional neural networks.

2.1 Holomorphic regularity assumption

One of the key ingredients of our study is the notion of holomorphic extension. Let $\Theta' \subseteq \mathbb{C}^p$ be an open set and let $\mathscr{V}$ be a Hilbert space. We denote by $\mathrm{Hol}(\Theta', \mathscr{V})$ the set of holomorphic maps from Θ' to $\mathscr{V}$. More precisely, $f \in \mathrm{Hol}(\Theta', \mathscr{V})$ if and

only if the following limit exists for all $\mathbf{z} \in \Theta'$ and all directions $j = 1, \dots, p$:

$$\lim_{\substack{h \in \mathbb{C} \\ h \to 0}} \frac{f(\mathbf{z} + h\mathbf{e}_j) - f(\mathbf{z})}{h} \in \mathscr{V},$$

where $\mathbf{e}_j = (\delta_{i,j})_{i=1}^p$ and $\delta_{i,j}$ denotes the Kronecker delta.

Definition 1. (Holomorphic extension) *Let $(\mathscr{V}, \|\cdot\|)$ be a Hilbert space and let $\Theta \subseteq \mathbb{R}^p$ be a set. Let $K \subseteq \mathbb{C}^p$ be a closed set such that $\Theta \subseteq K$. We say that a map $f : \Theta \to \mathscr{V}$ admits a holomorphic extension to K if there exists an open set Θ', $K \subseteq \Theta' \subseteq \mathbb{C}^p$, and a holomorphic map $\tilde{f} \in \mathrm{Hol}(\Theta', \mathscr{V})$ such that $\tilde{f}_{|\Theta} = f$. In this case we also set*

$$\|f\|_{L^\infty(K,\mathscr{V})} := \inf \Big\{ \|\tilde{f}\|_{L^\infty(\Theta',\mathscr{V})} \ \text{ s.t. } \ \Theta' \text{ open}, \ K \subseteq \Theta' \subseteq \mathbb{C}^p, \ \tilde{f} \in \mathrm{Hol}(\Theta', \mathscr{V}), \ \tilde{f}_{|\Theta} = f \Big\}. \tag{1}$$

Specifically, we are interested in maps that admit holomorphic extensions to so-called Bernstein polyellipses.

Definition 2. (Bernstein polyellipse) *Let $\boldsymbol{\rho} = (\rho_i)_{i=1}^p \in (1, +\infty)^p$. We call the set*

$$\mathcal{E}_{\boldsymbol{\rho}} := \mathcal{E}_{\rho_1} \times \cdots \times \mathcal{E}_{\rho_p} \subset \mathbb{C}^p$$

a Bernstein polyellipse *of parameter $\boldsymbol{\rho}$, where $\mathcal{E}_\rho := \{\frac{z+z^{-1}}{2} : z \in \mathbb{C}, \ 1 \leq |z| \leq \rho\}$.*

This setting is justified by the fact that several families of parametric models based on differential equations have parameter-to-solution maps admitting holomorphic extensions to Bernstein polyellipses. These include parametric diffusion problems, parametric parabolic problems, PDEs over parametrized domains, and parametric initial-value problems. For further discussion, we refer to, e.g., [4, Chapter 4] and [11].

If a map f admits a holomorphic extension to a polyellipse $\mathcal{E}_{\boldsymbol{\rho}}$, the parameter $\boldsymbol{\rho}$ acts as a measure of its *anisotropy*, i.e., it gauges the smoothness of f with respect to each input variable, and it may or may not be known *a priori*. Here, we focus on the more realistic case of unknown or *hidden* anisotropy (cf. [1, Definition 4]).

Definition 3. (Hidden anisotropy) *Let $\mathscr{V}$ be a Hilbert space, $\Theta = [-1, 1]^p \subset \mathbb{R}^p$, $\gamma > 0$ and $\epsilon > 0$. We write $\mathcal{HA}_{\gamma,\epsilon}(\Theta; \mathscr{V})$ for the set of Hilbert-valued maps $f : \Theta \to \mathscr{V}$ which admit a holomorphic extension to a Bernstein polyellipse $\mathcal{E}_{\boldsymbol{\rho}} \supset \Theta$ whose parameter $\boldsymbol{\rho} = (\rho_j)_{j=1}^d$ satisfies*

$$p! \prod_{j=1}^{p} \log(\rho_j) \geq \gamma^p (p+1)^p (1+\epsilon)^{-1}. \tag{2}$$

Although the definition of hidden anisotropy might seem obscure or somewhat arbitrary, there is a clear rationale for it. If a map $f : \Theta = [-1, 1]^p \to \mathscr{V}$

admits a holomorphic extension to a Bernstein polyellipse $\mathcal{E}_{\boldsymbol{\rho}}$, then its *best n-term approximation* f_n with respect to Legendre orthogonal polynomials on $L^2(\Theta;\mathscr{V})$ satisfies the following exponential decay rate for any $\epsilon > 0$ (see, e.g., [4, Theorem 3.15]):

$$\|f - f_n\|_{L^2(\Theta;\mathscr{V})} \leq \exp\left(-C_{\epsilon,p,\boldsymbol{\rho}} \cdot n^{1/p}\right),$$

for n large enough (more precisely, for $n \geq \bar{n}$ where $\bar{n} = \bar{n}(\epsilon, p, \boldsymbol{\rho})$) and where

$$C_{\epsilon,p,\boldsymbol{\rho}} = \frac{1}{p+1}\left(\frac{p!\prod_{j=1}^{p}\log(\rho_j)}{1+\epsilon}\right)^{1/p}.$$

Condition (2) of Definition 3 simply ensures a uniform control of the constant $C_{\epsilon,p,\boldsymbol{\rho}}$ via the inequality $C_{\epsilon,p,\boldsymbol{\rho}} \geq \gamma$. In other words, all functions $f \in \mathcal{HA}_{\gamma,\epsilon}(\Theta;\mathscr{V})$ satisfy the same best n-term exponential decay rate $\|f - f_n\|_{L^2(\Theta;\mathscr{V})} \leq \exp\left(-\gamma \cdot n^{1/p}\right)$, for n large enough. For further details we refer to [1, Section 3] and references therein.

Since our main focus will be on solution operators to parametrized PDEs, we find it convenient to introduce a short-cut notation for maps taking values in Sobolev spaces. We report it below.

Definition 4. (Hidden anisotropy for Sobolev-valued maps) *For $\Omega = (0,1)$, $\Theta = [-1,1]^p \subset \mathbb{R}^p$, $\gamma > 0$, $\epsilon > 0$ and $s \in \mathbb{N}$, we set*

$$\mathcal{HA}_{\gamma,\epsilon,s}(\Theta) = \mathcal{HA}_{\gamma,\epsilon}(\Theta; H^s(\Omega)).$$

In practice, Definition 4 simply sets $\mathscr{V} := H^s(\Omega)$. Here, following usual conventions, we equip $H^s(\Omega)$ with the energy norm

$$\|u\|_{H^s(\Omega)} := \sqrt{\int_0^1 |u|^2 dx + \sum_{k=1}^{s}\int_0^1 \left|\frac{d^k u}{dx^k}\right|^2 dx}.$$

We point out that, while Definition 4 allows for $s = 0$, our attention will be devoted to smoother scenarios, namely $s \geq 1$.

2.2 Background on neural networks

We now recall the mathematical definition of some of the most classical neural network architectures. We start with (feedforward) Deep Neural Networks (DNNs) implementing "standard" layers. Hereon, we adopt the usual convention according to which scalar functions are allowed to operate on vectors by acting componentwise on their entries. That is, given $\sigma : \mathbb{R} \to \mathbb{R}$, we let

$$\sigma([\mathrm{v}_1, \ldots, \mathrm{v}_l]) := [\sigma(\mathrm{v}_1), \ldots, \sigma(\mathrm{v}_l)].$$

Definition 5. (Standard layer) *Let n, m be positive integers. A standard layer with activation function $\sigma : \mathbb{R} \to \mathbb{R}$ is a map $L : \mathbb{R}^m \to \mathbb{R}^n$ of the form*

$$L(\mathbf{v}) = \sigma\left(\mathbf{W}\mathbf{v} + \mathbf{b}\right),$$

where $\mathbf{W} \in \mathbb{R}^{m \times n}$ *and* $\mathbf{b} \in \mathbb{R}^n$ *are the layer parameters, referred to as the weight matrix and the bias vector, respectively. The layer is said to be affine if* σ *is the identity map.*

A classical choice for the activation function σ is the Rectified Linear Unit (ReLU). Given a scalar input $a \in \mathbb{R}$, the latter acts as

$$\sigma(a) := \max\{0, a\}.$$

Architectures based on ReLU activations are very popular, as they reproduce the same expressivity of free-knot splines [13]. As specified below, ReLU networks are just compositions of multiple layers with ReLU activation.

Definition 6. (ReLU network) *Let* m, n *be positive integers. We say that a map* $\Phi : \mathbb{R}^m \to \mathbb{R}^n$ *is a ReLU network if it can be written as*

$$\Phi = L_{\ell+1} \circ L_\ell \circ \cdots \circ L_1$$

for some $\ell \geq 0$ *and some* $L_1, \ldots, L_{\ell+1}$*, where* $L_i : \mathbb{R}^{n_{i-1}} \to \mathbb{R}^{n_i}$ *are standard layers with ReLU activation for* $i = 1, \ldots, \ell$*,* $n_0 := m$*, and* $L_{\ell+1} : \mathbb{R}^{n_\ell} \to \mathbb{R}^n$ *is an affine layer. The layers* $L_1, \ldots, L_\ell$ *are called hidden layers, whereas* $L_{\ell+1}$ *is referred to as the output layer.*

In general, a tuple of layers $(L_{\ell+1}, L_\ell, \ldots, L_1)$ naturally defines a composite architecture of depth ℓ and size

$$\text{size} := \sum_{j=1}^{\ell+1} \left(\|\mathbf{W}_j\|_0 + \|\mathbf{b}_j\|_0 \right),$$

where $\mathbf{W}_j$ and $\mathbf{b}_j$ are the weight matrix and the bias vector of L_j, while $\|\mathbf{A}\|_0$ denotes the number of nonzero entries in the tensor $\mathbf{A}$.

We note that, in principle, a ReLU network $\Phi : \mathbb{R}^m \to \mathbb{R}^n$ may admit multiple representations, possibly referring to layer tuples of different depth and size. For instance, the map $\Phi : \mathbb{R}^1 \to \mathbb{R}^1$ defined as $\Phi(a) := \sigma(a)$ can be equivalently re-written as $\Phi(a) = \sigma(\sigma(a))$. The first representation has depth 1 and size 2, whereas the second one has depth 2 and size 3. This ambiguity comes from the fact that *depth* and *size* are properties of layer tuples, rather than properties of their composition. These considerations bring us to the following.

Definition 7. (Depth and size) *We say that a ReLU network has depth* $\leq \ell$ *and size* $\leq S$ *if it can be realized through a tuple of layers with depth* $\leq \ell$ *and size* $\leq S$*.*

With this clarification, we can now continue our summary by moving to convolutional layers and, thus, Convolutional Neural Networks (CNNs). Historically, convolutional architectures were first introduced to handle time series and RGB images [35], which were commonly stored in data structures orgnanized into *channels*. For instance, in the case of 1D convolutions, CNNs are designed to accept

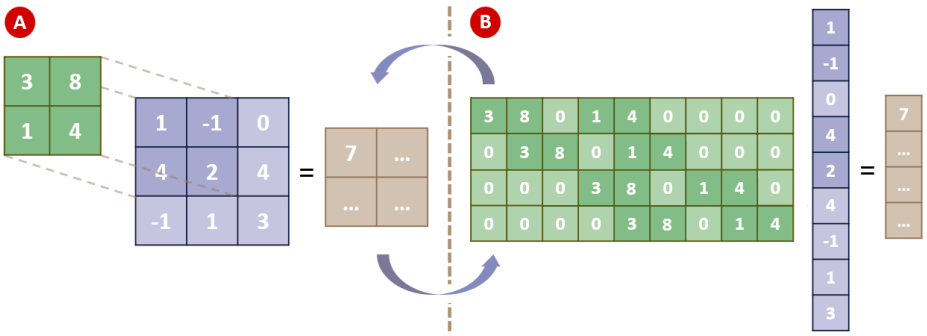

Figure 1: A 2D convolutional layer acting on a given input (simplified setting: 1 channel at input/output, no activation nor bias). The action of the convolutional layer can be visualized either in terms of a moving filter (A) or using the equivalent matrix representation (B): in both cases, despite mapping from $\mathbb{R}^9$ onto $\mathbb{R}^4$, the layer only comes with 4 learnable parameters, instead of $9 \cdot 4 = 36$.

inputs of dimension $\mathbb{R}^{m \times n}$ and return outputs of dimension $\mathbb{R}^{m' \times n'}$. Thus, they can only be connected to standard DNNs up to introducing suitable *reshape* operations.

Compared to standard architectures, CNNs are more effective in handling high dimensional data, as, by leveraging their spatial structure, they can carry out complex computations with few degrees of freedom. Indeed, it can be shown that a convolutional layer operating from $\mathbb{R}^{m \times n}$ to $\mathbb{R}^{m' \times n'}$ is formally equivalent to a standard layer $\mathbb{R}^{mn} \to \mathbb{R}^{m'n'}$ whose weight matrix is sparse and contains shared entries [46] (see Figure 1 for an illustration).

We provide a more rigorous definition of these architectures below. In what follows, we make use of the following notation, which is rather helpful when dealing with tensor objects. Given $\mathbf{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, we write $\mathbf{A}_{i_1,\ldots,i_p}$ for the $n_{p+1} \times \cdots \times n_d$ subtensor obtained by fixing the first p dimensions along the specified axis, where $1 \leq i_j \leq n_j$.

Definition 8. (1D Convolutional layer) *Let m, m', s, t, d be positive integers and let g be a common divisor of m and m'. For any input size $n \in \mathbb{N}$, $n > 0$, let*

$$n_{out} := \left\lfloor \frac{n - d(s-1) - 1}{t} + 1 \right\rfloor$$

A 1D Convolutional layer with m input channels, m' output channels, grouping number g, kernel size s, stride t, dilation factor d and activation function $\sigma : \mathbb{R} \to \mathbb{R}$, is a map of the form

$$L : \mathbb{R}^{m \times n} \to \mathbb{R}^{m' \times n_{out}}$$

whose action on a given input $\mathbf{V} \in \mathbb{R}^{m \times n}$ is defined as

$$L(\mathbf{V})_{k'} = \sigma\left(\sum_{k \in \mathcal{K}} \mathbf{W}_{k',k} \otimes_{t,d} \mathbf{V}_k + \mathbf{B}_{k'} \right),$$

where $1 \leq k' \leq m'$*, while*

$$\mathcal{K} = \{\lfloor g(k'-1)/m \rfloor m/g + 1, \ldots, (\lfloor g(k'-1)/m \rfloor + 1)\, m/g\}\,.$$

Here, $\mathbf{W} \in \mathbb{R}^{m' \times (m/g) \times s}$ *and* $\mathbf{B} \in \mathbb{R}^{m' \times n_{out}}$ *are the weight tensor and the bias matrix, respectively, whereas* $\otimes_{t,d}$ *is the cross-correlation operator with stride* t *and dilation* d*. The latter is defined so that, for any* $\mathbf{w} \in \mathbb{R}^s$ *and* $\mathbf{v} \in \mathbb{R}^n$*, one has* $\mathbf{w} \otimes_{t,d} \mathbf{v} \in \mathbb{R}^{n_{out}}$*, where*

$$(\mathbf{w} \otimes_{t,d} \mathbf{x})_j := \sum_{i=1}^{s} w_i v_{(j-1)t+(i-1)d+1}.$$

The default values for the stride and the dilation factor are $t = 1$ and $d = 1$, respectively. For this reason, with a slight abuse of notation, one says that Φ has no stride and no dilation when $t = d = 1$. Similarly, we assume $g = 1$ whenever the grouping number is not declared explicitly.

Definition 9. (1D transposed convolutional layer) *Let* m, m', s, t, d *be positive integers and let* g *be a common divisor of* m *and* m'*. For any input size* $n \in \mathbb{N}$*,* $n > 0$*, let*

$$n_{out} := (n-1)t + d(s-1) + 1.$$

A 1D transposed convolutional layer with m *input channels,* m' *output channels, grouping number* g*, kernel size* s*, stride* t*, dilation factor* d *and activation function* $\sigma : \mathbb{R} \to \mathbb{R}$*, is a map of the form*

$$L : \mathbb{R}^{m \times n} \to \mathbb{R}^{m' \times n_{out}}$$

whose action on a given input $\mathbf{V} \in \mathbb{R}^{m \times n}$ *is defined as*

$$L(\mathbf{V})_{k'} = \sigma\left(\sum_{k \in \mathcal{K}} \mathbf{W}_{k,k'} \otimes_{t,d}^{\top} \mathbf{V}_k + \mathbf{B}_{k'}\right),$$

where $1 \leq k' \leq m'$*, while*

$$\mathcal{K} = \{\lfloor g(k'-1)/m \rfloor m/g + 1, \ldots, (\lfloor g(k'-1)/m \rfloor + 1)\, m/g\}\,.$$

Here, $\mathbf{W} \in \mathbb{R}^{(m/g) \times m' \times s}$ *and* $\mathbf{B} \in \mathbb{R}^{m' \times n_{out}}$ *are the weight tensor and the bias matrix, respectively, whereas* $\otimes_{t,d}^{\top}$ *is the transposed cross-correlation operator with stride* t *and dilation* d*. The latter is defined so that, for any* $\mathbf{w} \in \mathbb{R}^s$ *and* $\mathbf{v} \in \mathbb{R}^n$*, one has* $\mathbf{w} \otimes_{t,d}^{\top} \mathbf{v} \in \mathbb{R}^{n_{out}}$*, where*

$$\left(\mathbf{w} \otimes_{t,d}^{\top} \mathbf{v}\right)_j := \sum_{i \in \mathcal{I}} w_{\lfloor (i-1)t/d + (1-j)/d \rfloor + 1} v_i,$$

with $\mathcal{I} = \left\{\left\lfloor \frac{j-1}{t} + 1 \right\rfloor, \ldots, \left\lfloor \frac{(s-1)d+j-1}{t} + 1 \right\rfloor\right\}$*.*

As we mentioned, it is also useful to define reshaping operations. We provide a rigorous definition below.

Definition 10. (Reshape) *Let m and n be positive integers. Let $R_{m,n} : \mathbb{R}^{mn} \to \mathbb{R}^{m\times n}$ be the bijective linear map defined as*

$$R_{m,n} : x \mapsto \begin{bmatrix} x_1 & \dots & x_n \\ x_{n+1} & \dots & x_{2n} \\ \dots & \dots & \dots \\ x_{(m-1)n+1} & \dots & x_{mn} \end{bmatrix}.$$

The map $R_{m,n}$ and its inverse, $R_{m,n}^{-1}$, are called reshape *operations.*

Definition 11. (Convolutional Neural Network) *We say that a map Ψ is a* Convolutional Neural Network (CNN) *if it can be realized as the composition of (transposed) convolutional layers and reshape operations.*

It is important to note that, by including reshape operations, CNNs can accept both vectors and matrices. In fact, any CNN $\tilde{\Psi} : \mathbb{R}^{m\times n} \to \mathbb{R}^{m'\times n'}$ comes with its vectorized counterpart $\Psi := R_{m',n'}^{-1} \circ \tilde{\Psi} \circ R_{m,n}$. Note in fact that, although Ψ operates on vectors, it can be considered a CNN according to Definition 11.

The concepts of depth and size can be easily generalized to CNNs in the natural way. We mention that, in doing so, reshape modules are typically ignored. Finally, in what follows, we will say that a CNN has at most q channels per layer if it can be realized without relying on convolutional layers that have more than q channels (either at input or output). Similarly, when stating that a CNN has depth $\leq \ell$, size $\leq S$, number of channels per layer $\leq q$, and kernel size per layer $\leq K$, we intend that there exists a representation of such architecture satisfying all those requirements simultaneously.

3 Main result

We are now ready to present our main result. However, before coming to the actual statement of the Theorem 1, it is worth recapping the general context. Let $p \in \mathbb{N}$, $p \geq 1$, $\Omega = (0,1)$ and $\Theta = [-1,1]^p$. Let $\{x_1 < x_2 < \dots < x_{N_h}\} \subset \overline{\Omega}$ be a fixed spatial grid, and let ϱ be the uniform probability distribution over Θ. DL-ROMs aim at approximating the map

$$\Theta \ni \boldsymbol{\mu} \mapsto [u_{\boldsymbol{\mu}}(x_1), \dots, u_{\boldsymbol{\mu}}(x_{N_h})]^\top \in \mathbb{R}^{N_h},$$

where $u_{\boldsymbol{\mu}} := \mathcal{G}(\boldsymbol{\mu})$ is the solution to some parametrized PDE, with $\mathcal{G} : \Theta \to H^s(\Omega)$ taking values in a suitable Sobolev space, $s \geq 1$.

In the DL-ROM paradigm, such approximation is provided by a neural network architecture

$$\Phi : \mathbb{R}^p \to \mathbb{R}^{N_h},$$

obtained via the composition of a reduced module, $\phi : \mathbb{R}^p \to \mathbb{R}^m$, and a decoder module, $\Psi : \mathbb{R}^m \to \mathbb{R}^{N_h}$. During training, these architectures are supplemented by an auxiliary encoder module, $\Psi' : \mathbb{R}^{N_h} \to \mathbb{R}^m$, which effectively turns the intermediate state space, $\mathbb{R}^m$, onto a *latent* space. In fact, the idea is that

$$\Phi(\boldsymbol{\mu}) := \Psi(\phi(\boldsymbol{\mu})) \approx [u_{\boldsymbol{\mu}}(x_1), \dots, u_{\boldsymbol{\mu}}(x_{N_h})]^\top,$$

while, at the same time, $\phi(\boldsymbol{\mu}) \approx \Psi'([u_{\boldsymbol{\mu}}(x_1), \dots, u_{\boldsymbol{\mu}}(x_{N_h})]^\top)$.

Recently, the theory underlying DL-ROMs has evolved significantly, see, e.g. [9, 19, 20, 21]. However, practical insights on the implementation and training of these architectures are far from being exhaustive. For this reason, domain practitioners often rely on suitable rules of thumb, deduced from empirical evidence. Some of these include:

i) in practice, althought certain studies suggest otherwise [42], ReLU networks are particularly expressive compared to other architectures that rely on different nonlinearities, such as the sigmoidal activation [44]. Thus, the ReLU activation, together with its smooth variants (softplus, SeLU, GeLU, etc.), can be a good choice when constructing DL-ROMs. Here, for the sake of simplicity, we shall focus on the "standard" ReLU;

ii) convolutional layers can significantly enhance the performances of the decoder module. In fact, given the high dimensionality of the output, classical layers would be prohibitive to train [36, 43];

iii) the number of convolutional layers in the decoder module should be proportional to the resolution of the spatial grid [20];

iv) the encoder block does not need to be as complex as the decoder, nor it benefits as much from the use of convolutional layers [21];

v) ideally, it should be possible to use the same autoencoder for different problems simultaneously [41, 52], without the need for re-training (a practice also known as *transfer learning*). In fact, classical compression techniques based on, e.g., Fourier transform, or wavelets, are somewhat universal. Similarly, we should be able to find problem-agnostic autoencoders that do not require a specific training routine;

vi) the reduced map should be trained by taking the encoder outputs as a ground truth reference, i.e., by minimizing

$$\frac{1}{N}\sum_{i=1}^{N} \|\phi(\boldsymbol{\mu}_i) - \Psi'([u_{\boldsymbol{\mu}_i}(x_1), \dots, u_{\boldsymbol{\mu}_i}(x_{N_h})]^\top)\|_2^2,$$

where $\{\boldsymbol{\mu}_i\}_{i=1}^N \subset \Theta$ is an independent identically distributed (i.i.d.) random sample, generated according to ϱ;

vii) to avoid overfitting and ensure a proper generalization, DL-ROMs can benefit from suitable regularization strategies [25], especially at the latent level [49];

As we shall see in a moment, by embedding convolutional neural networks within the novel framework of *practical existence theorems*, we can finally derive a comprehensive theory supporting these heuristics. We report our main result, Theorem 1, right below. For the sake of better readability, the proof is postponed to Section 4. In what follows, given random variable X, we shall write $\mathbb{E}^{1/2}[X]$ as a short-hand notation for $(\mathbb{E}[X])^{1/2}$.

Theorem 1. *There are universal constants $c_0, c_1, c_2, c_3, c_4 > 0$ such that the following holds. Let $p \in \mathbb{N}$, $p \geq 1$, and $\epsilon, \gamma > 0$. Let ϱ be the uniform probability distribution over $\Theta := [-1,1]^p$. Let $\Omega = (0,1)$ and*

$$\mathcal{G} : \Theta \ni \boldsymbol{\mu} \to u_{\boldsymbol{\mu}} \in H^s(\Omega)$$

be a (nonlinear) map belonging to $\mathcal{HA}_{\gamma,\epsilon,s}(\Theta)$, where $s \geq 1$ (see Definition 4). Fix a training size $N \geq 1$ and a probability of failure $0 < \varepsilon < 1$. Define

$$\tilde{N} := N \cdot (c_0 \log(2N)(\log(2N)\min\{\log(2N)+p,\ \log(2N)\log(2p)\}) + \log(1/\varepsilon))^{-1}$$

$$\Delta := \min\left\{2^{p/2+1}\tilde{N}^{3/2},\ e^2(\tilde{N}/2^p)^{1+\frac{1}{2}\log_2 p},\ \frac{\tilde{N}^{1/2}(\log\tilde{N}+(p+1)\log 2)^{p-1}}{2^{p/2-1}(p-1)!}\right\}.$$

Let $\{x_j\}_{j=1}^{N_h} \subset \overline{\Omega}$ be an equispaced grid of stepsize $h = 2^{-k}$ for some $k \in \mathbb{N}$. Fix a latent dimension $m \geq 1$ and let $\tilde{m} := 2m+1$. Then, there exist

a) a class of ReLU networks $\mathscr{F}$ from $\mathbb{R}^p \to \mathbb{R}^{\tilde{m}}$ with

$$\begin{aligned}\text{depth} &\leq c_1(1+p\log p)(1+\log\tilde{N})\left((\tilde{N}/2^p)^{1/2}+\log(\Delta)+\gamma\tilde{N}^{1/(2p)}\right),\\ \text{size} &\leq c_2 p\left(p\tilde{N}/2^p+\left((\tilde{N}/2^p)^{1/2}+p\Delta\right)\left(\log(\tilde{N}\Delta)+\gamma\tilde{N}^{1/(2p)}\right)\right)+\tilde{m}\Delta,\end{aligned}$$

b) a latent regularization function $\mathcal{R} : \mathscr{F} \to [0,+\infty)$, equivalent to a certain norm of the trainable parameters, and a regularization parameter $\lambda = \lambda(\tilde{N}, p)$,

c) a ReLU convolutional neural network, $\Psi : \mathbb{R}^{\tilde{m}} \to \mathbb{R}^{N_h}$, whose architecture only depends on $\mathcal{G}$ through s, with

$$\text{depth} \leq c_3\log(1/h), \qquad \text{size} \leq c_3 m\log(1/h),$$

$$\text{channels per layer} \leq 8m, \qquad \text{kernel size per layer} \leq 2,$$

acting as a decoder,

d) a ReLU network, $\Psi' : \mathbb{R}^{N_h} \to \mathbb{R}^{\tilde{m}}$, whose architecture only depends on $\mathcal{G}$ through s and $\|\mathcal{G}\|_{L^\infty(\mathcal{E}_\varrho, H^s(\Omega))}$, operating as an encoder,

such that the following holds with probability $1-\varepsilon$. Let $\{\boldsymbol{\mu}_i\}_{i=1}^N$ be an i.i.d. random sample, uniformly drawn from the parameter space Θ. Denote by P the function-to-grid operator, $P : u \mapsto [u(x_1), \ldots, u(x_{N_h})]$. Every minimizer $\hat{\phi} \in \mathscr{F}$ of

$$\min_{\phi\in\mathscr{F}} \sqrt{\frac{1}{N}\sum_{i=1}^N \|\phi(\boldsymbol{\mu}_i) - \Psi'(Pu_{\boldsymbol{\mu}_i})\|_2^2} + \lambda\mathcal{R}(\phi) \tag{3}$$

satisfies

$$\mathbb{E}^{1/2}_{\boldsymbol{\mu}\sim\varrho}\left[\sup_{j=1,\dots,N_h}|u_{\boldsymbol{\mu}}(x_j)-\Psi_j(\hat{\phi}(\boldsymbol{\mu}))|^2\right]\leq$$
$$\leq c_4\left(\sqrt{m}e^{-\frac{1}{\sqrt{2}}\gamma\tilde{N}^{1/(2p)}}+\sqrt{\frac{2m^{1-2s}}{2s-1}}\right)\|\mathcal{G}\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}},H^s(\Omega))},\quad(4)$$

for all $\tilde{N}\geq N_0$*, where* $N_0=N_0(\epsilon,\mathcal{G},m,p,s)$ *and* $\mathcal{E}_{\boldsymbol{\rho}}$ *is as in Definition 3.*

Theorem 1 has multiple implications. First of all, it shows clearly how the different properties of the problem affect the design of the neural network architectures. For instance, the complexity of the decoder, both in terms of depth and size, scales logarithmically with the grid resolution, h. In contrast, the reduced network, ϕ, does not depend on h, but on p. Notably, thanks to the regularity of the parameter-to-solution map, the reduced network is only mildly affected by the curse of dimensionality: its size grows at most quadratically in p (up to logarithmic factors).

Another interesting fact concerns the latent dimension, m, which directly appears in the error bound (4). On the hand, increasing m can improve the accuracy of the model (at a rate that depends on the smoothness of the PDE solutions, s). However, in order to generalize properly, DL-ROMs with a larger latent space necessitate of more training data, as clearly depicted by the term $\sqrt{m}\exp(-\tilde{N}^{1/(2p)}/\sqrt{2})$. More specifically, in order to achieve a prescribed target accuracy level $\tau>0$ it is sufficient for m to scale polynomially in $1/\tau$ and for $\tilde{N}$ to scale polynomially in $\log(1/\tau)$ (this can be seen by bounding the two main terms in the right-hand side of (4) from above with τ and rearranging the corresponding inequalities).

In general, Theorem 1 shows that the error of a trained DL-ROM can be bounded by two terms: a *sampling error*, which—asymptotically—decays exponentially with respect to the training set size, and an *approximation error*, driven by the architecture design and the output smoothness. Interestingly, our result confirms most of the heuristics adopted by domain practitioners: from the use of ReLU networks and convolutional autoencoders, to the introduction of latent regularization techniques. On this note, we also observe that Theorem 1 implicitly supports the use of *transfer learning*. In fact, looking back at the proof, our result suggests that, for a fixed degree of smoothness, there exists a universal autoencoder performing equivalently well for all operators $\mathcal{G}$ (up to a norm factor). In practice, such autoencoder could be initialized and trained *a priori*, by relying on synthetic data.

The regularizer $\mathcal{R}$ is an important component of Theorem 1. Recalling the notation introduced in Definition 6, the latter can be explicitly characterized using the matrix $\ell^{2,1}$-norm as

$$\mathcal{R}(\phi)=\|\mathbf{W}_{\ell+1}\|_{2,1}:=\sum_{j=1}^{n_\ell}\|\mathbf{W}_{\ell+1}\mathbf{e}_j\|_2,$$

where $\mathbf{W}_{\ell+1}\in\mathbb{R}^{\tilde{m}\times n_\ell}$ is the weight matrix associated with the last (linear) layer of ϕ, while $\mathbf{e}_j\in\mathbb{R}^{n_l}$ is the jth vector of the canonical basis. The presence of

this regularization term is essential to prove the practical existence theorem in [1] (upon which Theorem 1 relies). In fact, it allows one to rigorously connect deep neural network training with sparse polynomial approximation via compressed sensing. For a more detailed discussion, we refer the reader to [3].

In relation to this, an inspection of the proof of Theorem 1 reveals that $\hat{\phi}$ consists mostly of sparsely connected layers. This sparsity is further promoted by the regularizer $\mathcal{R}$, whose action naturally favors *compressibility* (i.e., approximate sparsity) of the last layer's weights; keeping only the absolute largest weights of this layer would yield improved network size bounds (see [3, Section 9.4] for further details). These observations suggests that adopting *network pruning* in this context might be an effective strategy (see [23]); notably, this is coherent with recent empirical evidence in the reduced order modeling literature, see, e.g., [22, 26].

Clearly, despite offering several insights, Theorem 1 comes with its own limitations: we provide a detailed discussion on the matter in Section 5.

3.1 Application to a parametric diffusion model

To showcase the applicability of Theorem 1, we consider a parametric diffusion equation with affine parametric dependence on the diffusion term. This model is often used as a case study in the parametric PDE literature (see, e.g., [4, Chapter 4], [11, 40], and references therein).

We consider the physical domain $\Omega = (0,1)$, a forcing term $F \in H^{-1}(\Omega)$, and functions $a_0 \in L^\infty(\Omega)$, $\{\psi_j\}_{j=1}^p \subset L^\infty(\Omega)$ defining an affine parametric diffusion term

$$a_{\boldsymbol{\mu}}(x) = a_0(x) + \sum_{j=1}^{p} \mu_j \psi_j(x), \quad x \in \Omega,\ \boldsymbol{\mu} \in \Theta. \tag{5}$$

Then, we consider the following parametric weak problem: for any $\boldsymbol{\mu} \in \Theta$, find $u_{\boldsymbol{\mu}} \in H_0^1(\Omega)$ such that

$$\int_\Omega a_{\boldsymbol{\mu}} \frac{du_{\boldsymbol{\mu}}}{dx} \frac{dv}{dx}\, dx = \int_\Omega F v\, dx, \quad \forall v \in H_0^1(\Omega). \tag{6}$$

In order to apply Theorem 1, we need to (i) find sufficient conditions ensuring that the map $\mathcal{G} : \boldsymbol{\mu} \mapsto u_{\boldsymbol{\mu}}$ belongs to $\mathcal{HA}_{\gamma,\epsilon,1}(\Theta)$ and (ii) estimate $\|\mathcal{G}\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}}, H^1(\Omega))}$.

First, we assume the parametric problem (6) to be *uniformly elliptic*, i.e., such that

$$\sum_{k=1}^{p} |\psi_k(x)| \leq a_0(x) - r, \tag{7}$$

for some $r > 0$. This implies, in particular, that $\operatorname{essinf}_{x\in\Omega} a_{\boldsymbol{\mu}}(x) \geq r$ for every $\boldsymbol{\mu} \in \Omega$ (and, hence, that (6) is elliptic for every fixed $\boldsymbol{\mu} \in \Theta$). In addition, for some fixed $\gamma, \epsilon > 0$ and $\xi > 0$ we assume the functions $\{\psi_k\}_{k=1}^p$ to be such that

$$\sum_{k=1}^{p} \left(\frac{\rho_k + \rho_k^{-1}}{2} - 1 \right) \|\psi_k\|_{L^\infty} \leq \xi, \tag{8}$$

for every $\boldsymbol{\rho}$ satisfying condition (2). In this setting, [4, Proposition 4.9] immediately implies that $\mathcal{G} \in \mathcal{HA}_{\gamma,\epsilon,1}$. Note that a holomorphic extension of $\mathcal{G}$ to $\mathcal{E}_{\boldsymbol{\rho}}$ is the map $\boldsymbol{\zeta} \mapsto u_{\boldsymbol{\zeta}}$, where $u_{\boldsymbol{\zeta}}$ is the (thanks to uniform ellipticity, unique) solution to the weak problem associated with the complex-valued diffusion coefficient

$$a_{\boldsymbol{\zeta}}(x) = a_0(x) + \sum_{j=1}^{p} \zeta_j \psi_j(x), \quad x \in \Omega,\ \boldsymbol{\zeta} \in \mathcal{E}_{\boldsymbol{\rho}}.$$

In addition, we see that

$$\|\mathcal{G}\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}}, H^1(\Omega))} \leq \sqrt{1 + \frac{1}{\pi^2}} \cdot \|\mathcal{G}\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}}, H_0^1(\Omega))} \leq \sqrt{1 + \frac{1}{\pi^2}} \cdot \frac{\|F\|_{H^{-1}(\Omega)}}{r - \xi},$$

where the first inequality hinges on the Poincaré inequality, while the second one is a consequence of [4, Proposition 4.9]. Here, $H_0^1(\Omega)$ is equipped with its classical energy (semi)norm

$$\|u\|_{H_0^1} := \sqrt{\int_\Omega \left|\frac{du}{dx}\right|^2 dx}.$$

We are then allowed to apply Theorem 1 to problem (6), in which case the error bound (4) reads

$$\mathbb{E}^{1/2}_{\boldsymbol{\mu} \sim \varrho}\left[\sup_{j=1,\dots,N_h} |u_{\boldsymbol{\mu}}(x_j) - \Psi_j(\hat{\phi}(\boldsymbol{\mu}))|^2\right] \leq \\ \leq c\left(\sqrt{m} e^{-\frac{1}{\sqrt{2}}\gamma \tilde{N}^{1/(2p)}} + \sqrt{\frac{2}{m}}\right) \frac{\|F\|_{H^{-1}(\Omega)}}{r - \xi},$$

for some universal constant $c > 0$.

We conclude by noting that Theorem 1 could also be applied to problem (6) for $s > 1$. Indeed, if $F \in H^{s-2}(\Omega)$ and $a_{\boldsymbol{\mu}} \in C^{s-1}(\Omega)$, then standard regularity theory results for PDEs imply that $u_{\boldsymbol{\mu}} \in H^s(\Omega)$ (see, e.g., [17, Section 6.3, Theorem 2]). However, finding precise sufficient conditions on the parametric coefficient $a_{\boldsymbol{\mu}}$ able to ensure that $\mathcal{G} \in \mathcal{HA}_{\gamma,\epsilon,s}$ and bounding $\|\mathcal{G}\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}}, H^s(\Omega))}$ requires an extension of [4, Proposition 4.9] and a careful analysis that is outside the scope of this paper.

4 Proof of Theorem 1

We subdivide the proof into several steps. In particular, we shall state, and prove, a few claims that eventually lead to the full proof. Before diving into the details, we recall the definition of *operator norm*. Given a linear map $T : (\mathscr{A}, \|\cdot\|_{\mathscr{A}}) \to (\mathscr{B}, \|\cdot\|_{\mathscr{B}})$ between two normed spaces, we set

$$|||T||| := \sup_{\substack{a \in \mathscr{A} \\ \|a\|_{\mathscr{A}} = 1}} \|Ta\|_{\mathscr{B}}.$$

Equivalently, due to linearity, $|||T|||$ is nothing but the (best) Lipschitz constant of T.

Step 1. *Without loss of generality, the function-to-grid operator, P, can be assumed to be injective over $\mathcal{G}(\Theta) \subset H^s(\Omega)$.*

Proof. Assume that we are able to prove Theorem 1 whenever P is injective over $\mathcal{G}(\Theta) \subset H^s(\Omega)$. Let us now consider an operator $\tilde{\mathcal{G}}$, satisfying all the hypotheses of the Theorem, but for which P is not injective over the image set $\tilde{\mathcal{G}}(\Theta)$. Then, the idea is to exploit the following Lemma, which, essentially, is just a re-writing of [14, Theorem 5.1].

Lemma 1. *Let Ω, s and $x_1, \dots, x_{N_h}$, be as in Theorem 1. There exists a bounded linear operator $Q : H^s(\Omega) \to H^s(\Omega)$ such that*

i) $(Qf)(x_j) = f(x_j)$ for $f \in H^s(\Omega)$ and all $j = 1, \dots, N_h$;

ii) if $f, g \in H^s(\Omega)$ and $f(x_j) = g(x_j)$ for all $j = 1, \dots, N_h$, then

$$\|Qf\|_{H^s(\Omega)} \leq \|g\|_{H^s(\Omega)}.$$

In practice, Q is a projection operator that maps $H^s(\Omega)$ onto a suitable subspace of smooth splines. Most importantly, Q acts as an interpolator with minimum norm: see (i) and (ii), respectively. Furthermore, it is straightforward to see that P is injective over $Q(H^s(\Omega))$. In fact, by letting $g \equiv 0$ in (ii), and by exploiting (i), we see that

$$P(Qf) = 0 \implies Pf = 0 = Pg \implies \|Qf\|_{H^s(\Omega)} \leq \|g\|_{H^s(\Omega)} = 0 \implies Qf \equiv 0.$$

With this in mind, let Q be as in Lemma 1, and let $\tilde{\mathcal{G}}_Q := Q \circ \tilde{\mathcal{G}}$. Since Q is both linear and continuous, it is holomorphic, and, furthermore,

$$\tilde{\mathcal{G}} \in \mathcal{HA}_{\gamma,\epsilon,s}(\Theta) \implies \tilde{\mathcal{G}}_Q \in \mathcal{HA}_{\gamma,\epsilon,s}(\Theta).$$

In particular, since $\tilde{\mathcal{G}}_Q$ satisfies all the properties in Theorem 1 and P is injective over $\tilde{\mathcal{G}}_Q(\Theta) \subseteq Q(H^s(\Omega))$, we are allowed to invoke Theorem 1 with $\mathcal{G} := \tilde{\mathcal{G}}_Q$, thus obtaining the error bound in Eq. (4) (recall that we assumed the Theorem to hold true whenever the additional hypothesis of injectivity is satisfied). However, since $(\tilde{\mathcal{G}}_Q \boldsymbol{\mu})(x_j) = (\tilde{\mathcal{G}}\boldsymbol{\mu})(x_j) = u_{\boldsymbol{\mu}}(x_j)$ for all $j = 1, \dots, N_h$, and

$$\|\tilde{\mathcal{G}}_Q\|_{L^\infty(\mathcal{E}, H^s(\Omega))} \leq \|\tilde{\mathcal{G}}\|_{L^\infty(\mathcal{E}, H^s(\Omega))}$$

due to (ii), it is evident that (4) also holds for $\mathcal{G} := \tilde{\mathcal{G}}$, thus proving our claim. □

Step 2. *There exists a linear operator $T : H^s(\Omega) \to \mathbb{R}^{\tilde{m}}$ and a CNN $\Psi : \mathbb{R}^{\tilde{m}} \to \mathbb{R}$ satisfying (d), such that*

$$\sup_{j=1,\dots,N_h} |u(x_j) - \Psi_j(Tu)| \leq \sqrt{\frac{2}{2s-1}} m^{1/2-s} \|u\|_{H^s(\Omega)} \quad \forall u \in \mathcal{G}(\Theta) \subset H^s(\Omega), \tag{9}$$

and $|||T||| \leq 2$.

Proof. By [20, Theorem 1] there exists a continuous linear operator $T : H^s(\Omega) \to \mathbb{C}^{2m+1}$ and a linear CNN (no activations nor biases at any level) $\Psi : \mathbb{C}^{2m+1} \to \mathbb{R}^{N_h}$, whose depth, size and number of channels satisfy the complexity bounds in (d), such that (9) holds[1]. The operator T only depends on s, and its operator norm can be bounded as $|||T||| \leq 2$: we refer the reader to the Appendix, Lemma A.2, for a rigorous description of T and its properties.

We note, however, that we cannot readily use such T and Ψ, as they take values (respectively, inputs) in $\mathbb{C}^{\tilde{m}} \cong \mathbb{R}^{2\tilde{m}}$. To fix this, for any $k \in \mathbb{N}$, let $B : \mathbb{C}^{\tilde{m}} \to \mathbb{R}^{\tilde{m}}$ be the linear map

$$B\left([a_{-m} + ib_{-m}, \dots, a_0 + ib_0, \dots, a_m + ib_m]\right) = [a_0, a_1, b_1, \dots, a_m, b_m]$$

(recall that $\tilde{m} = 2m + 1$), and let $B^\dagger : \mathbb{R}^k \to \mathbb{C}^k$ be its pseudo-inverse, acting as

$$B^\dagger\left([a_0, a_1, b_1, \dots, a_m, b_m]\right) = [a_m - ib_m, \dots, a_0, \dots, a_m + ib_m].$$

By diving deeper into the definition of T, cf. Eq. (27) in the Appendix, we see that for all $u \in H^s(\Omega)$ the image vector

$$Tu = [z_{-m}, \dots, z_0, \dots, z_m] \in \mathbb{C}^{\tilde{m}}$$

satisfies $z_0 \in \mathbb{R}$ and $z_k = \overline{z_{-k}}$ for all $k \in \{1, \dots, m\}$. Consequently, it is straightforward to see that

$$\Psi(B^\dagger B T u) = \Psi(Tu)$$

for all $u \in H^s(\Omega)$. In light of this, we are allowed to replace T with $B \circ T$ and Ψ with $\Psi \circ B^\dagger$, so that the two maps operate on the right spaces (i.e., $\mathbb{R}^{\tilde{m}}$ and not $\mathbb{C}^{\tilde{m}}$). In this concern, note also that $|||B||| \leq 1$: in particular, the bound on the operator norm is preserved. To keep the notation lighter, the presence of B and $B^\dagger$ will be omitted.

Note: with this construction, Ψ is linear. However, since $T(\mathcal{G}(\Theta))$ is compact, we can easily turn Ψ onto a ReLU CNN (without changing its outputs) by including suitable biases within the layers of the architecture. We refer to Lemma B.1 and Corollary B.1 in the Appendix for a detailed explanation. Once again, in order to simply the notation, we shall directly assume Ψ to be a ReLU CNN and avoid the introduction of auxiliary architectures. □

Step 3. *For every $\delta > 0$, there exists a ReLU encoder $\Psi' : \mathbb{R}^{N_h} \to \mathbb{R}^{\tilde{m}}$ such that*

$$\sup_{\boldsymbol{\mu} \in \Theta} \|T u_{\boldsymbol{\mu}} - \Psi'(P u_{\boldsymbol{\mu}})\|_2 < \delta. \tag{10}$$

Proof. In light of Step 1, we assume P to be injective over $\mathcal{G}(\Theta)$. Since the latter is compact (recall that $\mathcal{G}$ is continuous) and P is continuous, this suffices to show that P admits a continuous inverse

$$P^{-1} : P(\mathcal{G}(\Theta)) \to \mathcal{G}(\Theta),$$

[1] [20, Theorem 1] does not mention the bound on the kernel size explicitly; however, this is a direct consequence of [20, Lemma 3], upon which the previous Theorem is built.

which we may readily extend to a broader map from $\mathbb{R}^{N_h}$ onto $H^s(\Omega)$ (see, e.g., Dugundji's extension Theorem [15]): with little abuse of notation, we shall still denote this extension by P^{-1}. Let $E := T \circ P^{-1}$, so that $E : \mathbb{R}^{N_h} \to \mathbb{R}^{\tilde{m}}$, and fix any tolerance $\delta > 0$. Then, there exists a ReLU network $\Psi' : \mathbb{R}^{N_h} \to \mathbb{R}^{\tilde{m}}$ such that

$$\sup_{\mathbf{v} \in P(\mathcal{G}(\Theta))} \|E(\mathbf{v}) - \Psi'(\mathbf{v})\|_2 < \delta. \tag{11}$$

The existence of such Ψ' is guaranteed by the compactness of $P(\mathcal{G}(\Theta))$ and by the continuity of E, as ReLU networks are known to be dense in the space of continuous maps over compact subsets [29]. Since, by definition, we also have

$$E(Pu_{\boldsymbol{\mu}}) = TP^{-1}P(u_{\boldsymbol{\mu}}) = Tu_{\boldsymbol{\mu}}, \tag{12}$$

for all $\boldsymbol{\mu} \in \Theta$, it is clear that (11) is nothing but (10). □

Remark 2. *In what follows, we let $\mathcal{R} : \mathscr{F} \to [0, +\infty)$ be the regularization functional in [1, Theorem 5], so that, for any $\phi \in \mathscr{F}$, the penalty term $\mathcal{R}(\phi)$ corresponds to the ℓ^1 norm of the weights in the output layer of the network ϕ.*

Step 4. *Having fixed any $\delta > 0$ and Ψ' as in Step 3, for every rescaling factor $\eta > 0$ one has*

$$\begin{aligned}\operatorname*{argmin}_{\phi \in \mathscr{F}} \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\phi(\boldsymbol{\mu}_i) - \Psi'(Pu_{\boldsymbol{\mu}_i})\|^2 + \lambda \mathcal{R}(\phi)} = \\ = \frac{1}{\eta} \cdot \operatorname*{argmin}_{\phi \in \mathscr{F}} \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\phi(\boldsymbol{\mu}_i) - \eta\Psi'(Pu_{\boldsymbol{\mu}_i})\|^2 + \lambda \mathcal{R}(\phi)}.\end{aligned} \tag{13}$$

Proof. By definition, $\mathcal{R}(\eta\phi) = |\eta|\mathcal{R}(\phi)$ for all $\eta \in \mathbb{R}$. In fact, $\phi \in \mathscr{F} \implies \eta\phi \in \mathscr{F}$, as the latter is easily obtained by multiplying all the terminal weights in ϕ by the scalar value η. Then, it is straightforward to see that, for all $\eta > 0$,

$$\begin{aligned}\operatorname*{argmin}_{\phi \in \mathscr{F}} \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\phi(\boldsymbol{\mu}_i) - \Psi'(Pu_{\boldsymbol{\mu}_i})\|^2 + \lambda \mathcal{R}(\phi)} = \\ = \operatorname*{argmin}_{\phi \in \mathscr{F}} \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\eta\phi(\boldsymbol{\mu}_i) - \eta\Psi'(Pu_{\boldsymbol{\mu}_i})\|^2 + \lambda \mathcal{R}(\eta\phi)} = \\ = \frac{1}{\eta} \cdot \operatorname*{argmin}_{\phi \in \mathscr{F}} \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\phi(\boldsymbol{\mu}_i) - \eta\Psi'(Pu_{\boldsymbol{\mu}_i})\|^2 + \lambda \mathcal{R}(\phi)}.\end{aligned} \tag{14}$$

□

Step 5. *Let $\eta^* := (4\|\mathcal{G}\|_{L^\infty(\mathcal{E}, H^s(\Omega))})^{-1}$. For every $\delta > 0$, and a corresponding choice of Ψ', one has*

$$\mathbb{E}^{1/2}_{\boldsymbol{\mu} \sim \varrho} \|Tu_{\boldsymbol{\mu}} - \hat{\phi}(\boldsymbol{\mu})\|_2^2 \leq \eta_*^{-1} c_4 \exp\left(-\frac{1}{\sqrt{2}} \gamma \tilde{N}^{1/(2p)}\right) + c_4 \delta, \tag{15}$$

where $\hat{\phi} \in \mathscr{F}$ is any minimizer of (3).

Proof. For any $\eta > 0$, let us consider the rescaled minimization problem in Step 4, and let

$$\hat{\phi}_\eta := \operatorname*{argmin}_{\phi \in \mathscr{F}} \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\phi(\boldsymbol{\mu}_i) - \eta\Psi'(Pu_{\boldsymbol{\mu}_i})\|^2} + \lambda\mathcal{R}(\phi). \tag{16}$$

Let $f_\eta : \Theta \to \mathbb{R}^{\tilde{m}}$ be defined as $f_\eta(\boldsymbol{\mu}) := \eta T u_{\boldsymbol{\mu}} = \eta T\mathcal{G}(\boldsymbol{\mu})$. Let $\mathcal{E}_{\boldsymbol{\rho}}$ be the Bernstein polyellipse in Definition 3 corresponding to $\mathcal{G} \in \mathcal{HA}_{\gamma,\epsilon,s}(\Theta)$. Since T is linear, and thus entire, it is clear that f_η admits a holomorphic extension to $\mathcal{E}_{\boldsymbol{\rho}}$. Furthermore, by composition,

$$\|f_\eta\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}},\mathbb{R}^{\tilde{m}})} \leq \eta|||T||| \cdot \|\mathcal{G}\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}},H^s(\Omega))} \leq 2\eta\|\mathcal{G}\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}},H^s(\Omega))}.$$

In light of this, hereon we shall fix the rescaling parameter to

$$\eta_* := \frac{1}{4\|\mathcal{G}\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}},H^s(\Omega))}},$$

so that $f := f_{\eta_*}$ satisfies $\|f\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}},\mathbb{R}^{\tilde{m}})} \leq 1/2$. We now recall that, thanks to (10), we also have

$$\sup_{\boldsymbol{\mu}\in\Theta} \|f(\boldsymbol{\mu}) - \eta_*\Psi'(Pu_{\boldsymbol{\mu}})\|_2 < \eta_*\delta.$$

This allows us to interpret $\eta_*\Psi'(Pu_{\boldsymbol{\mu}_i})$ as perturbations of $f(\boldsymbol{\mu}_i)$, and thus consider Problem (16) as the training of a neural network model with ground truth f and noisy samples $\eta_*\Psi'(Pu_{\boldsymbol{\mu}_i}) \approx f(\boldsymbol{\mu}_i)$. In particular, by applying [1, Theorem 5] to f and (16) with $\eta = \eta_*$, we see that the loss minimizer $\hat{\phi}_{\eta_*}$ satisfies

$$\mathbb{E}^{1/2}_{\boldsymbol{\mu}\sim\varrho}\|f(\boldsymbol{\mu}) - \hat{\phi}_{\eta_*}(\boldsymbol{\mu})\|_2^2 \leq c_4 \exp\left(-\frac{1}{\sqrt{2}}\gamma\tilde{N}^{1/(2p)}\right) + c_4\eta_*\delta, \tag{17}$$

with probability $1-\varepsilon$, for all $\tilde{N} \geq N_0$, where $N_0 = N_0(\gamma, p, f)$ is a lower bound on the size of the training set. Let now ϕ be (any of) the original minimizer in the Theorem. As noted in (14), we have $\hat{\phi} = \hat{\phi}_{\eta_*} \cdot \eta_*^{-1}$ for some minimizer ϕ_{η_*} of the rescaled problem (16). Thus,

$$\begin{aligned}\mathbb{E}^{1/2}_{\boldsymbol{\mu}\sim\varrho}\|Tu_{\boldsymbol{\mu}} - \hat{\phi}(\boldsymbol{\mu})\|_2^2 = \\ = \eta_*^{-1}\mathbb{E}^{1/2}_{\boldsymbol{\mu}\sim\varrho}\|\eta_* Tu_{\boldsymbol{\mu}} - \hat{\phi}_{\eta_*}(\boldsymbol{\mu})\|_2^2 \leq \\ \leq \eta_*^{-1}c_4 \exp\left(-\frac{1}{\sqrt{2}}\gamma\tilde{N}^{1/(2p)}\right) + c_4\delta,\end{aligned} \tag{18}$$

as $f(\boldsymbol{\mu}) = \eta_* Tu_{\boldsymbol{\mu}}$. □

Step 6. *The error bound in (4) holds true.*

Proof. Let $\|\cdot\|_1$ denote the 1-norm on $\mathbb{R}^{\tilde{m}}$, so that $\|\mathbf{a}\|_1 := \sum_{i=1}^{\tilde{m}} |a_j|$. We recall that the following hold

$$\|\mathbf{a}-\mathbf{b}\|_1 \leq \sqrt{\tilde{m}}\|\mathbf{a}-\mathbf{b}\|_2, \qquad |\Psi_j(\mathbf{a}) - \Psi_j(\mathbf{b})| \leq \|\mathbf{a}-\mathbf{b}\|_1. \tag{19}$$

For the interested reader, we refer to [20], Pag. 7, for a detailed proof of the second inequality. We also note that, by definition,

$$\|u_{\boldsymbol{\mu}}\|_{H^s(\Omega)} \leq \|\mathcal{G}\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}}, H^s(\Omega))}. \tag{20}$$

To ease notation, let

$$E := \mathbb{E}^{1/2}_{\boldsymbol{\mu}\sim\varrho}\left[\sup_j |u_{\boldsymbol{\mu}}(x_j) - \Psi_j(\hat{\phi}(\boldsymbol{\mu}))|^2\right]$$

Since

$$E \leq \mathbb{E}^{1/2}_{\boldsymbol{\mu}\sim\varrho}\left[\sup_j |u_{\boldsymbol{\mu}}(x_j) - \Psi_j(Tu_{\boldsymbol{\mu}})|^2\right] + \mathbb{E}^{1/2}_{\boldsymbol{\mu}\sim\varrho}\left[\sup_j |\Psi_j(Tu_{\boldsymbol{\mu}}) - \Psi_j(\hat{\phi}(\boldsymbol{\mu}))|^2\right],$$

combining (9), (18), (19) and (20), ultimately yields

$$\begin{aligned} E &\leq \sqrt{\frac{2m^{1-2s}}{2s-1}}\|\mathcal{G}\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}}, H^s(\Omega))} + \mathbb{E}^{1/2}_{\boldsymbol{\mu}\sim\varrho}\|Tu_{\boldsymbol{\mu}} - \hat{\phi}(\boldsymbol{\mu})\|_1^2 \leq \\ &\leq \sqrt{\frac{2m^{1-2s}}{2s-1}}\|\mathcal{G}\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}}, H^s(\Omega))} + \sqrt{\tilde{m}}\mathbb{E}^{1/2}_{\boldsymbol{\mu}\sim\varrho}\|Tu_{\boldsymbol{\mu}} - \hat{\phi}(\boldsymbol{\mu})\|_2^2, \end{aligned}$$

and thus,

$$E \leq \left(\sqrt{\frac{2m^{1-2s}}{2s-1}} + 4\sqrt{\tilde{m}}c_4 \exp\left(-\frac{1}{\sqrt{2}}\gamma\tilde{N}^{1/(2p)}\right) + c_4\sqrt{\tilde{m}}g^{-1}\delta\right) g \tag{21}$$

where, for better readability, we have set $g := \|\mathcal{G}\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}}, H^s(\Omega))}$. Let us now fix the value of $\delta > 0$ such that

$$\delta \leq \frac{g}{c_4}\sqrt{\frac{2m^{1-2s}}{(2s-1)\tilde{m}}}. \tag{22}$$

Then, (21) can be simplified to

$$E \leq \left(2\sqrt{\frac{2m^{1-2s}}{2s-1}} + 4\sqrt{\tilde{m}}c_4 \exp\left(-\frac{1}{\sqrt{2}}\gamma\tilde{N}^{1/(2p)}\right)\right) g.$$

Since $\sqrt{\tilde{m}} \leq \sqrt{4m}$, it follows that,

$$E \leq \max\{2, 8c_4\}\left(\sqrt{\frac{2m^{1-2s}}{2s-1}} + \sqrt{m}\exp\left(-\frac{1}{\sqrt{2}}\gamma\tilde{N}^{1/(2p)}\right)\right) g. \tag{23}$$

In particular, up to re-naming the universal constant as $c_4' := \max\{2, 8c_4\}$, Eq. (23) immediately yields the desired conclusion. □

5 Conclusion

Motivated by the empirical success of deep-learning-based reduced order models for parametric PDEs, we proposed a new *practical existence theorem* (Theorem 1). Our analysis focuses on models relying on deep autoencoders, where two networks, Ψ' and Ψ are used to compress the output, whereas a third network, ϕ, is used to learn the parameter-to-latent-variables map; the parameter-to-solution operator $\mathcal{G}$ is then approximated via composition, $\mathcal{G} \approx \Psi \circ \phi$. Focusing on the case of deep convolutional autoencoders, our theorem provides an explicit error bound for trained models in which the decoder Ψ is constructed explicitly and the reduced network ϕ is trained via regularized empirical loss minimization. In doing so, the theorem also provides a list of sufficient conditions on the overall complexity of the reduced order model, $\Psi \circ \phi$, as well as detailed information concerning the training phase of the reduced network ϕ (sample size, sampling strategy, choice of the loss function). Notably, our theorem validates several heuristic observations from previous numerical studies, hence reducing the gap between theory and practice in this fast-growing area.

We conclude by mentioning some limitations of our theory, whose study is left to future work. First, our theory only covers the case of one-dimensional physical domains. Generalizing the theory to higher dimensions is an important open question. In this regard, there are two main obstacles that hinder the extensibility of our analysis to $d > 1$. The first one is a technicality regarding the operator T in the proof of Theorem 1 (see also Lemma A.2). Simply put, the latter consists of a periodicization operator, mapping arbitrary functions onto smooth periodic signals, composed with the truncated Fourier transform for data compression. Adapting this idea for $d > 1$ is nontrivial since domains can have arbitrary shapes. One idea could be to embed Ω onto a suitable hypercube $[-1, 1]^d$ and, depending on the smoothness of $\partial\Omega$, leverage well-known estimates of Sobolev extension operators, see, e.g., [18]. The second issue, instead, is merely practical. Our construction relies on a convolutional architecture that can replicate the performances of the Fourier transform. In higher-dimensions, this requires a very careful adaptation of [20, Lemma 1-4].

Aside from this, another interesting line of future work is the study of further parametric PDEs with holomorphic parametric dependence, such as elliptic problems with higher regularity ($s > 1$), parabolic problems, or PDEs over parametrized domains (see, e.g., [12]). Here, in fact, we only discussed the application of our theory to the parametric diffusion equation (see Section 3.1).

As we mentioned previously, it is worth remarking that Theorem 1 only addresses the training of the reduced network ϕ. The encoder Ψ and the decoder Ψ', instead, are constructed either using universal approximation theorems (encoder), or explicitly (decoder), that is, by hard-coding all weights and biases in the architecture. In addition, the fact that ϕ has standard as opposed to convolutional is inherited from [1, Theorem 5], upon which Theorem 1 relies.

We conclude with some comments on the curse of dimensionality. First, we observe that the sample complexity is affected by the curse in Theorem 1. In fact, considering the term involving $\tilde{N}$ in (4), for a given target accuracy $\tau > 0$, one has $\sqrt{m}\exp(-\gamma\tilde{N}^{1/(2p)}/\sqrt{2}) \leq \tau$ if anly only if $\tilde{N} \geq (\sqrt{2}\log(\sqrt{m}/\tau)/\gamma)^{2p}$, which

leads to an exponential dependence of $\tilde{N}$ on p. Moreover, our results lose significance if $p \approx N_h$, mainly due to the curse of dimensionality affecting the reduced network $\phi : \mathbb{R}^p \to \mathbb{R}^{\tilde{m}}$. In fact, our complexity bounds include the exponential term $\tilde{N}^{1/(2p)}$. In principle, this issue could be addressed in—at least—three ways. The first one could be to focus on a smaller class of operators, that is, analytic parameter-to-solution maps enjoying suitable summability properties in their power series expansion, as in [33, 51]. Second, one could consider proving a practical existence theorem using algebraic as opposed to exponential best s-term decay rates (see [4, Chapter 3]) in the spirit of [3, Theorem 8.1]. This would also have to be combined (similarly to the first strategy) with higher regularity assumptions involving infinite-dimensional analyticity and would require an adaptation of the argument in [3] to the Hilbert- or, at least, vector-valued setting. A third approach could rely on incorporating an additional compression phase at input, either through autoencoders or linear projections; see, e.g., [19, 27, 34]. Nevertheless, adapting these ideas to our context is challenging due to the need for a theory describing the implementation and training of a (convolutional) encoder; as we mentioned previously, this is, in general, highly nontrivial.

Acknowledgments

SB acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) through grant RGPIN-2020-06766 and the Fonds de Recherche du Québec Nature et Technologies (FRQNT) through grant 313276. NF is member of the Gruppo Nazionale per il Calcolo Scientifico (GNCS) of the Istituto Nazionale di Alta Matematica (INdAM). The present research is part of the activities of project Dipartimento di Eccellenza 2023-2027, Department of Mathematics, Politecnico di Milano, funded by MUR, and of project Cal.Hub.Ria (Piano Operativo Salute, traiettoria 4), funded by MSAL.

The authors would also like to thank Prof. Paolo Zunino (Politecnico di Milano) for promoting the development and publication of this work and Prof. Ben Adcock (Simon Fraser University) for providing helpful comments on a earlier version of this manuscript.

A Hermite polynomials and signal periodicization

This Appendix presents two supplementary results, both of which are essential for our construction. In particular, we expand on the definition of the operator T appearing in the proof of Theorem 1, while simultaneously deriving some useful inequalities.

Following [20], our approach involves employing a periodicization operator that leverages on Hermite interpolation: see Fig. 2 for a visual representation. Thus, we first derive some preliminary results related to Hermite polynomials (Lemma A.1), and then proceed with a synthetic discussion about the definition and the analytical properties of the operator T (Lemma A.2).

Lemma A.1. *Let $s \in \mathbb{N}$, $s \geq 1$. For any $0 \leq j \leq s-1$, let $p_{s,j}$ and $q_{s,j}$ be the unique polynomials of degree $2s-1$ for which the following hold true*

$$p_{s,j}^{(k)}(0) = \delta_{j,k} \qquad p_{s,j}^{(k)}(1) = 0$$

$$q_{s,j}^{(k)}(0) = 0 \qquad q_{s,j}^{(k)}(1) = \delta_{j,k},$$

Then, $\|q_{s,j}\|_{L^2(0,1)} = \|p_{s,j}\|_{L^2(0,1)}$. Furthermore, $\|p_{s,j}\|_{L^2(0,1)} \leq (\sqrt{1/2})^{j+1}$.

Proof. Since $q_{s,j}(x) = (-1)^j p_{s,j}(1-x)$, the first statement is obvious. As for the second one, we shall proceed in three steps.

Step 1. *We prove that $p_{s,j}(x) \geq 0 \; \forall x \in [0,1]$.*

We note that, since $p_{s,j}$ has a zero of order s at 1, we have

$$p_{s,j}(x) = g(x)(1-x)^s$$

for some polynomial g of degree $s-1$, which depends on s and j. We now notice that, since $g(x) = p_{s,j}(x)(1-x)^{-s}$, one has

$$g^{(k)}(x) = \sum_{l=0}^{k} \binom{k}{l} p_{s,j}^{(k-l)}(x)(1-x)^{-s-l} \frac{(s+l)!}{s!}.$$

Let a_k be the kth coefficient in the polynomial expansion of g. Then, the above implies

$$a_k = \frac{1}{k!} g^{(k)}(0) = \frac{1}{k!} \sum_{l=0}^{k} \binom{k}{l} p_{s,j}^{(k-l)}(0) \frac{(s+l)!}{s!} \geq 0,$$

since $p_{s,j}^{(l)}(0) \geq 0$ for all $0 \leq l \leq k \leq s-1$. In particular, all the coefficients in g are positive, implying $g \geq 0$ on $[0, +\infty)$, and thus $p_{s,j} \geq 0$ on $[0,1]$, as claimed.

Step 2. *We prove that $\|p_{s,0}\|_{L^2(0,1)} \leq \sqrt{1/2}$.*

Using the definition, it is straightforward to verify that the polynomial $p_{s,0}$ can be written in closed form as

$$p_{s,0}(x) = 1 - \frac{\int_0^x y^{s-1}(1-y)^{s-1} dy}{\int_0^1 y^{s-1}(1-y)^{s-1} dy}. \tag{24}$$

In fact, the right-hand-side of (24): i) is a polynomial of degree $(s-1)+(s-1)+1 = 2s-1$; ii) vanishes at $x=1$, while it equals 1 at $x=0$; iii) its derivative is proportional to $x^{s-1}(1-x)^{s-1}$, which vanishes at $x = 0, 1$ with all its higher order derivatives (up to degree $s-2$). Since the polynomial $p_{s,0}$ is uniquely characterized by such conditions, this proves that the identity in (24) holds true.

We now note that, since the integrand $y \mapsto y^{s-1}(1-y)^{s-1}$ is positive, the polynomial $p_{s,0}$ happens to be monotone nonincreasing in $[0,1]$. Consequently,

$$0 \leq p_{s,0}(x) \leq p_{s,0}(0) = 1,$$

for all $x \in [0,1]$, and thus

$$\|p_{s,0}\|^2_{L^2(0,1)} = \int_0^1 p^2_{s,0}(x)dx \leq \int_0^1 p_{s,0}(x)dx \tag{25}$$

Furthermore, due symmetry, it is straightforward to see that

$$p_{s,0}(x) = 1 - p_{s,0}(1-x),$$

from which, up to a simple change of variables, it follows that

$$\int_0^1 p_{s,0}(x)dx = 1 - \int_0^1 p_{s,0}(1-x)dx = 1 + \int_1^0 p_{s,0}(z)dz = 1 - \int_0^1 p_{s,0}(z)dz$$

$$\implies \int_0^1 p_{s,0}(x)dx = \frac{1}{2},$$

which in turn implies $\|p_{s,0}\|_{L^2(0,1)} \leq \sqrt{1/2}$ due to (25).

Step 3. *We prove that* $\|p_{s,j}\|_{L^2(0,1)} \leq (\sqrt{1/2})^{j+1}$.

To prove the remaining cases, we shall exploit the following recursive formula,

$$p_{s,j}(x) = \int_0^x p_{s-1,j-1}(y)dy + (p_{s,0}(x) - 1)\int_0^1 p_{s-1,j-1}(y)dy,$$

which can be easily verified by hand. We re-write the above as

$$p_{s,j}(x) = p_{s,0}(x)\int_0^1 p_{s-1,j-1}(y)dy - \int_x^1 p_{s-1,j-1}(y)dy. \tag{26}$$

Since all polynomials in the form $p_{\tilde{s},\tilde{j}}$ are positive (cf. Step 1), we have

$$0 \leq p_{s,j}(x) \leq p_{s,0}(x)\int_0^1 p_{s-1,j-1}(y)dy,$$

implying that,

$$\|p_{s,j}\|_{L^2(0,1)} \leq \|p_{s,0}\|_{L^2(0,1)}\int_0^1 p_{s-1,j-1}(y)dy \leq \|p_{s,0}\|_{L^2(0,1)}\|p_{s-1,j-1}\|_{L^2(0,1)}.$$

Finally, iterating the above and applying the result at Step 2, yields

$$\|p_{s,j}\|_{L^2(0,1)} \leq \|p_{s,0}\|_{L^2(0,1)} \cdot \|p_{s-1,0}\|_{L^2(0,1)} \cdot \ldots \cdot \|p_{s-j,0}\|_{L^2(0,1)} \leq (\sqrt{1/2})^{j+1}.$$

□

Lemma A.2. *Let* $\Omega := (0,1)$. *Let* $s, m \in \mathbb{N}$, $s, m \geq 1$. *For any* $f \in H^s(\Omega)$, *let* p_f *be the polynomial of degree* $2s-1$ *given by*

$$p_f(x) := \sum_{j=0}^{s-1} [f^{(j)}(1) - f^{(j)}(0)] \cdot [p_{s,j}(x) - q_{s,j}(x)],$$

and let $\tilde{f} \in H^s(\Omega)$ *be the periodicized version of* f*, which we define as (cf. Fig. 2)*

$$\tilde{f}(x) := \begin{cases} f(2x) + p_f(2x) & 0 \le x \le 1/2 \\ f(2x-1) & 1/2 < x \le 1. \end{cases} \tag{27}$$

Define the linear operator $T : H^s(\Omega) \to \mathbb{C}^{2m+1}$ *as*

$$T : f \mapsto \left[\int_0^1 \tilde{f}(x) e^{2\pi \mathbf{i} m x} dx, \dots, \int_0^1 \tilde{f}(x) e^{-2\pi \mathbf{i} m x} dx \right].$$

Then, $|||T||| \le 2$.

Proof. Let $f \in H^s(\Omega)$. For any $j = 0, \dots, s-1$ we have

$$|f^{(j)}(1) - f^{(j)}(0)| = \left| \int_0^1 f^{(j+1)}(x) dx \right| \le \|f^{(j+1)}\|_{L^2(\Omega)}.$$

By Lemma A.1, we have

$$\|p_f\|_{L^2(\Omega)} \le 2 \sum_{j=0}^{s-1} |f^{(j)}(1) - f^{(j)}(0)| \cdot \|p_{s,j}\|_{L^2(\Omega)} \le 2 \sum_{j=0}^{s-1} (\sqrt{1/2})^{j+1} \|f^{(j+1)}\|_{L^2(\Omega)}.$$

Then, by the Cauchy-Schwarz inequality,

$$\begin{aligned} \|p_f\|_{L^2(\Omega)} &\le 2 \sqrt{\sum_{j=0}^{s-1} \left(\frac{1}{2}\right)^{j+1}} \sqrt{\sum_{j=0}^{s-1} \|f^{(j+1)}\|_{L^2(\Omega)}^2} \le \\ &\le 2 \sqrt{\sum_{j=0}^{+\infty} \left(\frac{1}{2}\right)^{j+1}} \|f\|_{H^s(\Omega)} = 2\|f\|_{H^s(\Omega)}. \end{aligned}$$

Consequently,

$$\|f + p_f\|_{L^2(\Omega)} \le \|f\|_{L^2(\Omega)} + \|p_f\|_{L^2(\Omega)} \le 3\|f\|_{H^s(\Omega)}.$$

We now note that a simple change of variables yields

$$\|\tilde{f}\|_{L^2(\Omega)}^2 = \frac{1}{2}\|f + p_f\|_{L^2(\Omega)}^2 + \frac{1}{2}\|f\|_{L^2(\Omega)}^2,$$

implying that

$$\|\tilde{f}\|_{L^2(\Omega)} \le \frac{3}{2}\|f\|_{H^s(\Omega)} + \frac{1}{2}\|f\|_{L^2(\Omega)} \le 2\|f\|_{H^s(\Omega)}.$$

Finally, we note that T maps f onto the truncated Fourier coefficients of $\tilde{f}$. In particular, for $\|\cdot\|_2$ the Euclidean norm,

$$\|Tf\|_2 \le \|\tilde{f}\|_{L^2(\Omega)} \le 2\|f\|_{H^s(\Omega)},$$

as claimed. □

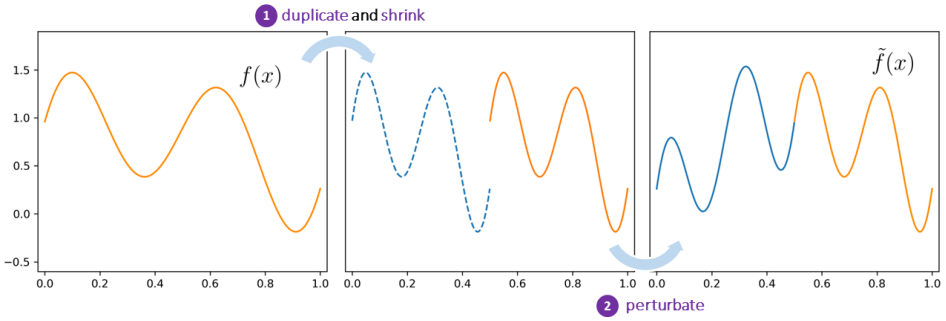

Figure 2: Visualization of the transformation $f \mapsto \tilde{f}$ used in Lemma A.2. The signal f is duplicated and a polynomial perturbation is added to ensure (smooth) periodicity.

B Auxiliary results on ReLU networks

This Appendix contains some technical details about the interplay between linear networks and ReLU networks. As noted in the proof of Theorem 1, these considerations are fundamental, as they allow us to adapt [20, Theorem 1] to our setting.

Lemma B.1. *Let $\Psi : \mathbb{R}^m \to \mathbb{R}^N$ be a linear network (no activations nor biases at any level). For every compact set $C \subset \mathbb{R}^m$, there exists a ReLU network $\tilde{\Psi}$ having the same architecture (and the same weights), such that $\tilde{\Psi}(\mathbf{c}) = \Psi(\mathbf{c})$ for all $\mathbf{c} \in C$.*

Proof. In plain words, the idea is to introduce suitable biases at the internal layers that can shift neuron entries to nonnegative values (which would be unaffected by ReLUs). Then, a terminal bias is used to shift the output back to the desired value. We shall now discuss the whole idea in a more rigorous way. Let ℓ be the number of hidden layers in Ψ. Since Ψ is linear, it must be of the form

$$\Psi(\mathbf{c}) = \mathbf{W}_{\ell+1} \cdot \cdots \cdot \mathbf{W}_1 \mathbf{c},$$

where $\mathbf{W}_i$, $i = 1, \dots, l+1$, are the matrices representing the action of the ith layer, respectively. Let us introduce the following notation

$$\mathbf{W}_{i \to j} := \prod_{k=i}^{j} \mathbf{W}_k,$$

defined for all pairs $1 \leq i \leq j \leq \ell + 1$. We construct a sequence of biases $\mathbf{b}_0, \dots, \mathbf{b}_{\ell+1}$, via the iterative scheme below,

$$\begin{cases} \mathbf{b}_0 = 0, \\ \mathbf{b}_i = -\min\limits_{\mathbf{c} \in C} \left(\mathbf{W}_{1 \to i} \mathbf{c} + \sum\limits_{k=0}^{i-1} \mathbf{W}_{k+1 \to i} \mathbf{b}_k \right), \quad i = 1, \dots, \ell, \\ \mathbf{b}_{\ell+1} = -\sum\limits_{k=0}^{\ell} \mathbf{W}_{k+1 \to \ell+1} \mathbf{b}_k \end{cases} \tag{28}$$

the minimum being defined entrywise (note that all minima are well-defined due compactness of C). For σ the ReLU activation function, consider the layers

$$L_i : x \mapsto \sigma\left(\mathbf{W}_i \mathbf{c} + \mathbf{b}_i\right),$$

defined for $i = 1, \dots, \ell$. We claim that the ReLU network

$$\tilde{\Psi} := \mathbf{W}_{\ell+1}(L_\ell \circ \cdots \circ L_1)(\mathbf{c}) + \mathbf{b}_{\ell+1}$$

coincides with Ψ over C. To see this, we start by noting that for all $\mathbf{c} \in C$, due to (28), we have

$$\mathbf{b}_1 \geq -\mathbf{W}_{1\to 1}\mathbf{c} - \cancel{\mathbf{W}_{1\to 1}\mathbf{b}_0} = -\mathbf{W}_1\mathbf{c},$$

implying that $\mathbf{W}_1\mathbf{c} + \mathbf{b}_1$ has nonnegative entries. Consequently,

$$L_1(\mathbf{c}) = \sigma\left(\mathbf{W}_1\mathbf{c} + \mathbf{b}_1\right) = \mathbf{W}_1\mathbf{c} + \mathbf{b}_1.$$

Similarly,

$$\mathbf{b}_2 \geq -\mathbf{W}_{1\to 2}\mathbf{c} - \cancel{\mathbf{W}_{1\to 2}\mathbf{b}_0} - \mathbf{W}_{2\to 2}\mathbf{b}_1 = -\mathbf{W}_2\mathbf{W}_1\mathbf{c} - \mathbf{W}_2\mathbf{b}_1,$$

implying that

$$\mathbf{W}_2\mathbf{W}_1\mathbf{c} + \mathbf{W}_2\mathbf{b}_1 + \mathbf{b}_2,$$

has nonnegative entries, and thus

$$L_2(L_1(\mathbf{c})) = L_2(\mathbf{W}_1\mathbf{c}+\mathbf{b}_1) = \sigma\left(\mathbf{W}_2\mathbf{W}_1\mathbf{c} + \mathbf{W}_2\mathbf{b}_1 + \mathbf{b}_2\right) = \mathbf{W}_2\mathbf{W}_1\mathbf{c}+\mathbf{W}_2\mathbf{b}_1+\mathbf{b}_2.$$

Iterating the above argument, one can easily see that

$$\begin{aligned}\left(L_\ell \circ \cdots \circ L_1\right)(\mathbf{c}) = \mathbf{W}_\ell \cdot \cdots \cdot \mathbf{W}_1\mathbf{c} + \mathbf{W}_\ell \cdot \cdots \cdot \mathbf{W}_2\mathbf{b}_1 + \mathbf{W}_\ell \cdot \cdots \cdot \mathbf{W}_3\mathbf{b}_2 + \cdots + \mathbf{b}_\ell = \\ = \mathbf{W}_{1\to\ell}\mathbf{c} + \sum_{k=0}^{\ell-1} \mathbf{W}_{k+1\to\ell}\mathbf{b}_k + \mathbf{b}_\ell,\end{aligned}$$

for all $\mathbf{c} \in C$. Then,

$$\begin{aligned}\tilde{\Psi}(\mathbf{c}) = \mathbf{W}_{\ell+1}\mathbf{W}_{1\to\ell}\mathbf{c} + \sum_{k=0}^{\ell-1} \mathbf{W}_{\ell+1}\mathbf{W}_{k+1\to\ell}\mathbf{b}_k + \mathbf{W}_{\ell+1}\mathbf{b}_\ell + \mathbf{b}_{\ell+1} = \\ = \mathbf{W}_{1\to\ell+1}\mathbf{c} + \sum_{k=0}^{\ell-1} \mathbf{W}_{k+1\to\ell+1}\mathbf{b}_k + \mathbf{W}_{\ell+1}\mathbf{b}_\ell + \mathbf{b}_{\ell+1} = \\ = \mathbf{W}_{1\to\ell+1}\mathbf{c} + \cancel{\sum_{k=0}^{\ell} \mathbf{W}_{k+1\to\ell+1}\mathbf{b}_k + \mathbf{b}_{\ell+1}} = \mathbf{W}_{1\to\ell+1}\mathbf{c} = \Psi(\mathbf{c}).\end{aligned}$$

In particular, $\tilde{\Psi}_{|C} \equiv \Psi_{|C}$, as wished. □

Corollary B.1. *Let $\Psi : \mathbb{R}^m \to \mathbb{R}^N$ be a linear CNN (no activations nor biases at any level). For every compact set $C \subset \mathbb{R}^m$, there exists a ReLU CNN $\tilde{\Psi}$ having the same architecture (and the same weights), such that $\tilde{\Psi}(\mathbf{c}) = \Psi(\mathbf{c})$ for all $\mathbf{c} \in C$.*

Proof. This is a direct consequence of Lemma B.1. In fact, convolutional layers are uniquely characterized by the fact of having a linear component that acts as a convolution operator. Since, in the lemma, the transformation $\Psi \to \tilde{\Psi}$ preserves the linear part of each layer, the conclusion follows. □

References

[1] B. Adcock, S. Brugiapaglia, N. Dexter, and S. Moraga, Deep neural networks are effective at learning high-dimensional Hilbert-valued functions from limited data, In *Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference, Proceedings of Machine Learning Research*, **145** (2022), pp. 1-36.

[2] B. Adcock, S. Brugiapaglia, N. Dexter, and S. Moraga, Near-optimal learning of Banach-valued, high-dimensional functions via deep neural networks, *arXiv preprint* arXiv:2211.12633 (2022).

[3] B. Adcock, S. Brugiapaglia, N. Dexter, and S. Moraga, Learning smooth functions in high dimensions: From sparse polynomials to deep neural networks, In *Handbook of Numerical Analysis*, In press, (2024).

[4] B. Adcock, S. Brugiapaglia, and C.G. Webster, Sparse Polynomial Approximation of High-Dimensional Functions, *Society for Industrial and Applied Mathematics*, Philadelphia, PA, 2022

[5] B. Adcock and N. Dexter, The gap between theory and practice in function approximation with deep neural networks, *SIAM Journal on Mathematics of Data Science*, **3**(2) (2021), pp. 624-655.

[6] S.E. Ahmed, S. Pawar, O. San, and A. Rasheed, Reduced order modeling of fluid flows: Machine learning, Kolmogorov barrier, closure modeling, and partitioning, In *AIAA Aviation 2020 Forum* (2020), p. 2946.

[7] J. Barnett and C. Farhat, Quadratic approximation manifold for mitigating the Kolmogorov barrier in nonlinear projection-based model order reduction, *Journal of Computational Physics*, **464** (2022), p. 111348.

[8] W.M. Boon, N.R. Franco, A. Fumagalli, and P. Zunino, Deep learning based reduced order modeling of Darcy flow systems with local mass conservation, *arXiv preprint* arXiv:2311.14554 (2023).

[9] S. Brivio, S. Fresca, N.R. Franco, and A. Manzoni, Error estimates for POD-DL-ROMs: a deep learning framework for reduced order modeling of nonlinear parametrized PDEs enhanced by proper orthogonal decomposition, *arXiv preprint* arXiv:2305.04680 (2023).

[10] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, Deep convolutional autoencoder-based lossy image compression, In *2018 Picture Coding Symposium (PCS)*, June 2018, pp. 253-257. IEEE.

[11] A. Cohen and R. DeVore, Approximation of high-dimensional parametric PDEs, *Acta Numerica*, **24** (2015), pp. 1-159.

[12] A. Cohen, C. Schwab, and J. Zech, Shape holomorphy of the stationary Navier–Stokes equations, *SIAM Journal on Mathematical Analysis*, **50**(2) (2018), pp. 1720-1752.

[13] I. Daubechies, R. DeVore, S. Foucart, B. Hanin, and G. Petrova, Nonlinear approximation and (deep) ReLU networks, *Constructive Approximation*, **55**(1) (2022), pp. 127-172.

[14] C. De Boor and R.E. Lynch, On splines and their minimum properties, *Journal of Mathematics and Mechanics*, **15**(6) (1966), pp. 953-969.

[15] J. Dugundji, An extension of Tietze's theorem, *Pacific J. Math.*, **1** (1951), pp. 353–367.

[16] D. Elbrächter, D. Perekrestenko, P. Grohs, and H. Bölcskei, Deep neural network approximation theory, *IEEE Transactions on Information Theory*, **67**(5) (2021), pp. 2581-2623.

[17] L.C. Evans, Partial differential equations, vol. 19 of Grad. Stud. Math., *American Mathematical Society*, Providence, RI, 2nd ed., 2010.

[18] C. Fefferman, A. Israel, and G. Luli, Sobolev extension by linear operators, *Journal of the American Mathematical Society*, **27**(1) (2014), pp. 69–145.

[19] N.R. Franco, D. Fraulin, A. Manzoni, and P. Zunino, On the latent dimension of deep autoencoders for reduced order modeling of PDEs parametrized by random fields, *arXiv preprint* arXiv:2310.12095 (2023).

[20] N.R. Franco, S. Fresca, A. Manzoni and P. Zunino, Approximation bounds for convolutional neural networks in operator learning, *Neural Networks*, **161** (2023), pp. 129-141.

[21] N.R. Franco, A. Manzoni, and P. Zunino, A deep learning approach to reduced order modelling of parameter dependent partial differential equations, *Mathematics of Computation*, **92**(340) (2023), pp. 483-524.

[22] N.R. Franco, A. Manzoni, and P. Zunino, Mesh-informed neural networks for operator learning in finite element spaces, *Journal of Scientific Computing*, **97**(2) (2023), 35.

[23] J. Frankle and M. Carbin, The lottery ticket hypothesis: finding sparse, trainable neural networks, In *Proceedings of the 7th International Conference on Learning Representations (ICLR)* (2019).

[24] S. Fresca and A. Manzoni, POD-DL-ROM: Enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition, *Computer Methods in Applied Mechanics and Engineering*, **388** (2022), p. 114181.

[25] S. Fresca, L. Dede', and A. Manzoni, A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized PDEs, *Journal of Scientific Computing*, **87** (2021), pp. 1-36.

[26] Q. Hernandez, A. Badias, D. Gonzalez, F. Chinesta, and E. Cueto, Deep learning of thermodynamics-aware reduced-order models from data, *Computer Methods in Applied Mechanics and Engineering*, **379** (2021), pp. 113763.

[27] L. Herrmann, C. Schwab, and J. Zech, Neural and GPC operator surrogates: construction and expression rate bounds, *arXiv preprint arXiv:2207.04950*, (2022).

[28] J.S. Hesthaven and S. Ubbiali, Non-intrusive reduced order modeling of nonlinear problems using neural networks, *Journal of Computational Physics*, **363** (2018), pp. 55-78.

[29] K. Hornik, Approximation capabilities of multilayer feedforward networks, *Neural networks*, **4**(2) (1991), pp. 251-257.

[30] N. Kovachki, S. Lanthaler, and S. Mishra, On universal approximation and error bounds for Fourier neural operators, *The Journal of Machine Learning Research*, **22**(1) (2021), pp. 13237-13312.

[31] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar, Neural operator: Learning maps between function spaces, *arXiv preprint* arXiv:2108.08481 (2021).

[32] G. Kutyniok, P. Petersen, M. Raslan, and R. Schneider, A theoretical analysis of deep neural networks and parametric PDEs, *Constructive Approximation*, **55**(1) (2022), pp. 73-125.

[33] S. Lanthaler, S. Mishra, and G.E. Karniadakis, Error estimates for DeepONets: A deep learning framework in infinite dimensions, *Transactions of Mathematics and Its Applications*, **6**(1), tnac001 (2022).

[34] S. Lanthaler, Operator learning with PCA-Net: upper and lower complexity bounds, *Journal of Machine Learning Research*, **24**(318) (2023), pp. 1–67.

[35] Y. LeCun and Y. Bengio, Convolutional networks for images, speech, and time series, In *The handbook of brain theory and neural networks*, MIT Press, **3361**(10) (1995).

[36] K. Lee and K.T. Carlberg, Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders, *Journal of Computational Physics*, **404** (2020), p. 108973.

[37] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, Fourier neural operator for parametric partial differential equations, *arXiv preprint* arXiv:2010.08895 (2020).

[38] H. Liu, B. Dahal, R. Lai, W. Liao, Generalization Error Guaranteed Auto-Encoder-Based Nonlinear Model Reduction for Operator Learning, *arXiv preprint arXiv:2401.10490*, 2024.

[39] L. Lu, P. Jin, G. Pang, Z. Zhang, and G.E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, *Nature Machine Intelligence*, **3**(3) (2021), pp. 218-229.

[40] C. Marcati and C. Schwab, Exponential convergence of deep operator networks for elliptic partial differential equations, *SIAM Journal on Numerical Analysis*, **61**(3) (2023), pp. 1513–1545.

[41] S. Masoumi-Verki, F. Haghighat, N. Bouguila, and U. Eicker, The use of GANs and transfer learning in model-order reduction of turbulent wake of an isolated high-rise building, *Building and Environment*, **246** (2023), p. 110948.

[42] S. Mishra and T. K. Rusch, Enhancing Accuracy of Deep Learning Algorithms by Training with Low-Discrepancy Sequences, *SIAM Journal on Numerical Analysis*, **59**(3) (2021), pp. 1811-1834.

[43] N.T. Mücke, S.M. Bohté, and C.W. Oosterlee, Reduced order modeling for parameterized time-dependent PDEs using spatially and memory aware deep learning, *Journal of Computational Science*, **53** (2021), p. 101408.

[44] E. Oostwal, M. Straat, and M. Biehl, Hidden Unit Specialization in Layered Neural Networks: ReLU vs. Sigmoidal Activation, *Physica A: Statistical Mechanics and its Applications*, **564** (2021), p. 125517.

[45] B. Peherstorfer, Breaking the Kolmogorov barrier with nonlinear model reduction, *Notices of the American Mathematical Society*, **69**(5) (2022), pp. 725-733.

[46] P. Petersen and F. Voigtlaender, Equivalence of approximation by convolutional neural networks and fully-connected networks, *Proceedings of the American Mathematical Society*, **148**(4) (2020), pp. 1567-1581.

[47] F. Pichi, B. Moya, and J.S. Hesthaven, A graph convolutional autoencoder approach to model order reduction for parametrized PDEs, *arXiv preprint* arXiv:2305.08573 (2023).

[48] F. Pichi, F. Ballarin, G. Rozza, and J.S. Hesthaven, An artificial neural network approach to bifurcating phenomena in computational fluid dynamics, *Computers & Fluids*, **254** (2023), p. 105813.

[49] F. Romor, G. Stabile, and G. Rozza, Non-linear manifold ROM with convolutional autoencoders and reduced over-collocation method, *arXiv preprint* arXiv:2203.00360 (2022).

[50] L. Rosafalco, M. Torzoni, A. Manzoni, S. Mariani, and A. Corigliano, Online structural health monitoring by model order reduction and deep learning algorithms, *Computers & Structures*, **255** (2021), p. 106604.

[51] C. Schwab and J. Zech, Deep learning in high dimension: neural network expression rates for analytic functions in $L^2(\mathbb{R}^d, \gamma_d)$, *SIAM/ASA Journal on Uncertainty Quantification*, **11**(1) (2023), pp. 199–234.

[52] C. Sun, M. Ma, Z. Zhao, S. Tian, R. Yan, and X. Chen, Deep transfer learning based on sparse autoencoder for remaining useful life prediction of tool in manufacturing, *IEEE Transactions on Industrial Informatics*, **15**(4) (2018), pp. 2416-2425.

[53] P. Vitullo, A. Colombo, N.R. Franco, A. Manzoni, and P. Zunino, Nonlinear model order reduction for problems with microstructure using mesh informed neural networks, *Finite Elements in Analysis and Design*, **229** (2024), p. 104068.

[54] D. Yarotsky, Error bounds for approximations with deep ReLU networks, *Neural Networks*, **94** (2017), pp. 103-114.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

44/2024 Fumagalli, I.
Discontinuous Galerkin method for a three-dimensional coupled fluid-poroelastic model with applications to brain fluid mechanics

45/2024 Fumagalli, A.; Patacchini, F. S.
Numerical validation of an adaptive model for the determination of nonlinear-flow regions in highly heterogeneos porous media

46/2024 Riccobelli, D.; Ciarletta, P.; Vitale, G.; Maurini, C.; Truskinovsky, L.
Elastic Instability behind Brittle Fracture

43/2024 Antonietti, P.F.; Corti, M., Martinelli, G.
Polytopal mesh agglomeration via geometrical deep learning for three-dimensional heterogeneous domains

42/2024 Fois, M.; Katili M. A.; de Falco C.; Larese A.; Formaggia L.
Landslide run-out simulations with depth-averaged models and integration with 3D impact analysis using the Material Point Method

41/2024 Bergonzoli, G.; Rossi, L.; Masci, C.
Ordinal Mixed-Effects Random Forest

40/2024 Carrara, D.; Regazzoni, F.; Pagani, S.
Implicit neural field reconstruction on complex shapes from scattered and noisy data

39/2024 Bartsch, J.; Buchwald, S.; Ciaramella, G.; Volkwein, S.
Reconstruction of unknown nonlienar operators in semilinear elliptic models using optimal inputs

38/2024 Tonini, A., Regazzoni, F., Salvador, M., Dede', L., Scrofani, R., Fusini, L., Cogliati, C., Pontone, G., Vergara, C., Quarteroni, A.
Two new calibration techniques of lumped-parameter mathematical models for the cardiovascular system

Fumagalli, A.; Patacchini, F.S.
Numerical validation of an adaptive model for the determination of nonlinear-flow regions in highly heterogeneous porous media