CONSTRUCTIVE
APPROXIMATION

Check for
updates

Nonlocality and Nonlinearity Implies Universality in Operator Learning

Samuel Lanthaler[1] · Zongyi Li[2] · Andrew M. Stuart[2]

Abstract

Neural operator architectures approximate operators between infinite-dimensional Banach spaces of functions. They are gaining increased attention in computational science and engineering, due to their potential both to accelerate traditional numerical methods and to enable data-driven discovery. However, basic questions about minimal requirements for universal approximation remain open. It is clear that any general approximation of operators must be both nonlocal and nonlinear. In this paper we describe how these two attributes may be combined in a simple way to deduce universal approximation. In so doing we unify the analysis of a wide range of neural operator architectures. A popular variant of neural operators is the Fourier neural operator (FNO). Previous universality theorems for FNO rely on intuition from spectral methods and require an unbounded number of Fourier modes. The present work challenges this point of view: (i) the work reduces FNO to its core essence, resulting in a minimal architecture termed the "averaging neural operator" (ANO); and (ii) analysis of the ANO shows that even this minimal ANO architecture benefits from universal approximation. This result is obtained based on only a spatial average as its only nonlocal ingredient. This corresponds to retaining only a *single* Fourier mode in the special case of the FNO, taking the analysis far from that of spectral methods. In addition to our discussion of universality, we present numerical results which give empirical insight into complexity issues related to the roles of channel width (embedding dimension) and number of Fourier modes.

Communicated by Wolfgang Dahmen.

✉ Samuel Lanthaler
samuel.lanthaler@univie.ac.at

[1] University of Vienna, Kolingasse 14-16 / 7.40, 1090 Wien, Austria

[2] California Institute of Technology, 1200 E California Blvd, Pasadena, CA 91125, USA

1 Introduction

1.1 Motivation and Literature Review

Neural networks [18] are receiving growing interest in computational science and engineering. In many applications of interest in the sciences, the task at hand is to approximate an underlying operator, which defines a mapping between two infinite-dimensional Banach spaces of functions. New neural network-based frameworks, termed *neural operators* [1, 3, 27, 38], generalize neural networks to this infinite-dimensional setting, with the aim of learning such operators from data. They have the potential to *accelerate* traditional numerical methods when a mathematical description, often in the form of a partial differential equation, is known. When no model is available, this data-driven methodology has the potential to *discover* the underlying input-output map.

Several neural operator architectures have been proposed in recent years. The first neural operator architecture (i) appeared in [9] and was subsequently generalized by adding depth to the underlying networks (DeepONet) [38]; further extensions include [24, 29, 42, 48]. DeepONets have been successfully deployed in a variety of application areas [6, 15, 39, 53]. Concurrently with DeepONet a number of other approaches to operator learning have also been developed and successfully deployed including: (ii) methods which combine ideas from principal component analysis with neural networks (PCA-Net) [3, 21]; (iii) operator learning based on random features, which may be viewed as Monte Carlo approximation of kernel methods, as introduced in [41]; and (iv) the wide class of neural operators introduced in [1, 27]. The class of methods (iv) defines neural operators in analogy with conventional neural networks, but appends the weight matrices in the hidden layers with additional linear integral operators acting on the input function. Special cases of this framework include graph neural operators [1, 34] and the Fourier neural operator (FNO) [33]. The FNO in particular has received considerable interest due to its state-of-the-art performance on many tasks. The FNO, and extensions thereof, remains a very active research direction: see, for example, [5, 31, 36, 45, 51, 52]. We also mention the closely related convolutional neural operator of [47] and a related Fourier-based approach in [43, 44]. The reliance on a Fourier basis limits the basic form of the FNO to periodic geometries although, in that setting, use of the fast Fourier transform (FFT) allows for efficient computations with total number of Fourier components limited only by the grid resolution. There is a set of papers which seek to address the restriction to periodic geometry by replacing the FFT with a wavelet [50] or multi-wavelet [19] transform; in this context, we also mention Laplace neural operators [7], which extends the FNO by replacing the Fourier transform/inversion step with a Laplace transform/inversion step. Another extension of FNO, so-called Neural Operator on Riemannian Manifolds (NORM) [8], generalize the FNO to use arbitrary orthogonal eigenfunctions of the Laplace-Beltrami operator on any given spatial domain. Another approach to extending the scope of the FFT is to learn transformations of arbitrary domains into unit cubes, suitable for application of the FFT [31]. Most closely related to the approach proposed in the present paper is

the low rank neural operator [27, Sect. 4.2] which, however, has a more complicated architecture than that proposed here.

There is a growing body of approximation theory for certain classes of operators arising from PDEs, using Galerkin and Taylor approximation in high dimensional spaces; see [10, 25] for recent overviews of this literature, in particular in the context of the Darcy flow problem as pioneered in the paper [11]. However the empirical success of neural operators on very wide classes of operator learning problems, beyond those covered by classical approximation theoretic approaches, suggests the value of developing theory for neural operators. A prerequisite for the success of any neural operator architecture, and its use in a diverse range of applications, is a *universal approximation* property. Universal approximation results are known for DeepONets [9, 28], PCA-Net [3], for general neural operators [27] and for the (more constrained) Fourier neural operator [26]. As most operators of interest in applications are not only nonlinear, but also nonlocal, any successful operator learning framework must share both of these properties. While the need for nonlinearity is well-understood even for ordinary neural networks, nonlocality is a requirement that is more specific to the function-space setting; it is a crucial ingredient for operator learning. The headline message of this paper is that universal approximation can be obtained in general geometries, with nonlocality introduced using only a low-rank operator of fixed finite rank.

Fourier neural operators introduce nonlocality via the addition of a nonlocal operator in each hidden layer, which acts on the Fourier modes of the input function by matrix multiplication. Available analysis of FNOs in [26] mostly rests on an analogy between FNOs and spectral methods: higher approximation accuracy is established by retaining larger numbers of Fourier modes. This interpretation is at odds with practical experience with FNOs. FNOs are typically implemented with a first layer which lifts the input, a scalar or vector-valued function, to a vector-valued function where the vector dimension (referred to as the number of channels, also named the model width) is much higher than that of the input function itself. In some circumstances it can be more beneficial to increase the number of channels rather than to retain more Fourier modes in the architecture. A first theoretical insight into this empirical observation has recently been achieved in [29] from the perspective of nonlinear reconstruction, where the analysis of specific settings revealed clear benefits of increasing the number of channels, and showed that, in those specific cases, arbitrary accuracy can be achieved when retaining only a *fixed* number of Fourier modes; the same paper also contains numerical experiments which exemplify the theoretical insight. These and similar results indicate that our theoretical understanding of neural operators, which has been largely guided by experience with traditional numerical methods, is still incomplete, and suggests a need for further analysis to improve our understanding of the precise role of nonlocality in operator learning.

These observations motivate the present work, in which we seek a deeper understanding of "how much" nonlocality is needed in neural operator architectures. The surprising result is that even very simple nonlocality, in the form of an integral *average*, is already sufficient for universality in operator learning. This is astonishing if one contrasts the simplicity of a humble average with the often complex nonlocal PDE solution operators of interest in engineering and the sciences, which, for example, give

rise to turbulence and other complex phenomena. The present work thus paves the way for a more systematic exploration of nonlocality in operator learning.

We highlight the fact that theory quantifying the computational complexity of achieving a given error appears very difficult for neural operators. It is currently limited to random features models [30], to linear problems [14] and to specific problems such as those with holomorphic structure and deploying specific network constructions [20, 40]; numerical studies of the cost-accuracy trade-off beyond these special settings may be found in [13]. Thus, although they do not quantify complexity, universal approximation theorems have an important role to play as neural operator methodology is developed: they highlight the importance of certain high level design choices in neural operators. We also note that our analysis and numerics provide an improved understanding of the trade-offs between increasing the channel width versus increasing the number of Fourier modes, in the FNO architecture. Indeed our numerical experiments point to problem-dependent subtleties in optimizing this trade-off.

In the next subsection, we first provide an overview of the general structure of neural operators in a unified framework, which encompasses recently proposed architectures [5, 31]. We then proceed to boil this general structure down to its bare essence, the averaging neural operator (ANO). The ANO builds on only two minimal ingredients, nonlinearity by composition with shallow neural networks, and nonlocality via a spatial average. We state, and sketch the proof of, two universal approximation theorems for the ANO; proof details are given in the Appendix A. The ANO is a subclass of numerous neural operators, including many of those listed in the preceding literature review; we thus deduce, in section 3, many universal approximation theorems from properties of the ANO. We follow this, in section 4, with numerical experiments illustrating some of the implications of our work. Section 5 contains concluding discussions.

1.2 Neural Operator

We now provide an overview of neural operator architectures which are of particular relevance to this work. To differentiate this generic class from specific instances, such as those further discussed in Section 3, we will refer to this general class as *nonlocal neural operators* (NNO).

Let Ω denote a bounded domain in $\mathbb{R}^d$ (or potentially a manifold) and let $\mathcal{X}(\Omega; \mathbb{R}^o)$, $\mathcal{Y}(\Omega; \mathbb{R}^o)$ and $\mathcal{V}(\Omega; \mathbb{R}^o)$ denote Banach spaces of $\mathbb{R}^o-$valued functions over Ω. The NNO is defined as a mapping $\Psi : \mathcal{X}(\Omega; \mathbb{R}^k) \to \mathcal{Y}(\Omega; \mathbb{R}^{k'})$ which can be written as a composition of the form $\Psi = \mathcal{Q} \circ \mathcal{L}_L \circ \cdots \circ \mathcal{L}_1 \circ \mathcal{R}$, consisting of a lifting layer $\mathcal{R}$, hidden layers $\mathcal{L}_\ell$, $\ell = 1, \ldots, L$, and a projection layer $\mathcal{Q}$. Given a **channel dimension** d_c, the **lifting layer** $\mathcal{R}$ is given by a mapping

$$\mathcal{R} : \mathcal{X}(\Omega; \mathbb{R}^k) \to \mathcal{V}(\Omega; \mathbb{R}^{d_c}), \quad u(x) \mapsto R(u(x), x), \qquad (1.1)$$

where $R : \mathbb{R}^k \times \Omega \to \mathbb{R}^{d_c}$ is a learnable neural network acting between finite dimensional Euclidean spaces. For $\ell = 1, \ldots, L$ (the number of **hidden layers**) and for $m = 0, \ldots, M$ (the number of **modes**) choose functions $\psi_{\ell,m}, \phi_{\ell,m} : \Omega \to \mathbb{R}^{d_c}$. For

$\ell = 1, \ldots, L$, each **hidden layer** $\mathcal{L}_\ell$ is of the form

$$(\mathcal{L}_\ell v)(x) := \sigma \left(W_\ell v(x) + b_\ell + \sum_{m=0}^{M} \langle T_{\ell,m} v, \psi_{\ell,m} \rangle_{L^2(\Omega; \mathbb{R}^{d_c})} \phi_{\ell,m}(x) \right). \qquad (1.2)$$

Each hidden layer defines a mapping $\mathcal{L}_\ell : \mathcal{V}(\Omega; \mathbb{R}^{d_c}) \rightarrow \mathcal{V}(\Omega; \mathbb{R}^{d_c})$. For $\ell = 1, \ldots, L$ and for $m = 0, \ldots, M$ the matrices $W_\ell, T_{\ell,m} \in \mathbb{R}^{d_c \times d_c}$, and bias $b_\ell \in \mathbb{R}^{d_c}$ are learnable parameters. The activation function $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ acts as a Nemitskii-operator, component-wise on inputs; i.e. for a vector-valued function $v(x) = (v_1(x), \ldots, v_{d_c}(x))$, we define $\sigma(v(x)) := (\sigma(v_1(x)), \ldots, \sigma(v_{d_c}(x)))$ for $x \in \Omega$. Throughout the paper the activation function $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ is assumed to be smooth, $\sigma \in C^\infty(\mathbb{R})$, nonpolynomial and Lipschitz continuous. Finally, the **projection layer** $\mathcal{Q}$ is given by a mapping,

$$\mathcal{Q} : \mathcal{V}(\Omega; \mathbb{R}^{d_c}) \rightarrow \mathcal{Y}(\Omega; \mathbb{R}^{k'}), \quad v(x) \mapsto Q(v(x), x), \qquad (1.3)$$

where $Q : \mathbb{R}^{d_c} \times \Omega \rightarrow \mathbb{R}^{k'}$ is also a learnable neural network acting between finite dimensional Euclidean spaces. Note that the form of the lifting and projection layers allows for *positional encoding.* The role of this positional encoding is discussed in further detail, in Section 2.4.

The general framework summarized above reduces to the FNO in a periodic geometry and if the expansion functions are chosen as Fourier basis functions, indexed by m and independent of ℓ. It also relates to other architectures, such as wavelet operators [19, 50], or the Laplace eigenbasis neural operators of [8]. The central question to be addressed in the following is this: "which minimal assumptions have to be imposed on the expansion functions $\psi_{\ell,m}$ and $\phi_{\ell,m}$, to ensure universal approximation of the resulting architecture?"

2 Averaging Suffices for Universal Approximation

In this section we prove that the use of a simple average provides enough nonlocality, within the broader NNO structure, leading to the two universal approximation Theorems 1 and 2 stated below. The architecture used to prove these theorems is a subclass of many concrete instantiations of the general NNO architecture, and hence implies universality approximation results for NNO as corollaries (cf. Section 3, below). When the domain is periodic, the resulting averaging neural operator is also a special case of the FNO, when only the zeroth Fourier mode is retained. Thus, our universality result also implies the universality of FNOs, even when only computing with a single, constant, Fourier mode. Thus, the theoretical results in this section provide new insight into the empirical observation, for example in the experiments in [29], that increasing the number of channels is often more beneficial than increasing the number of Fourier modes in the practical training of FNOs. In subsection 2.1 we discuss the nonlinearity of neural networks between finite dimensional Euclidean spaces and emphasize the role of nonlocality in operator learning. Subsection 2.2 defines the subclass of NNOs

used for our analysis of universal approximation, the topic of subsection 2.3. Intuition for the proof is given in subsection 2.4, and a proof sketch is provided in subsection 2.5, with details left for the appendix.

2.1 Nonlinearity and Nonlocality

To achieve a universal approximation property, a neural operator architecture must necessarily define a nonlinear operator. In the context of ordinary neural networks acting as approximators of functions between finite dimensional Euclidean spaces, it is well-known that (nonpolynomial) nonlinearity essentially also represents a *sufficient* condition for their universal approximation property [2, 12, 22, 46]. This should be contrasted with neural operators, where nonlinearity alone is not sufficient to ensure universality; to illustrate this latter fact, we note that even a single layer of an ordinary neural network actually gives rise to a nonlinear operator, which maps an input function to an output function by composition,

$$u(x) \mapsto \sigma(Wu(x) + b). \tag{2.1}$$

Here, W and b represent the weights and biases, respectively, and σ denotes the activation function. Despite being highly nonlinear, operators of such a compositional form cannot be universal. The main reason for this is that any such Ψ is *local*, in the sense that the value $\Psi(u)(x)$ of the output function at a given evaluation point x depends only on the value of the input function $u(x)$ at that same evaluation point. As a consequence, such mappings are not able to approximate even simple operators $\Psi^\dagger$ with a *nonlocal* dependence on the input, such as the shift operator $\Psi^\dagger(u)(x) := u(x + h)$ for fixed $h \neq 0$.

The arguably simplest example of a nonlocal operator with dependence on *all* point values $u(x)$, $x \in \Omega$, is given by averaging the input function u over its domain Ω:

$$u(x) \mapsto \fint_\Omega u(y)\, dy := \frac{1}{|\Omega|} \int u(y)\, dy. \tag{2.2}$$

Clearly, such averaging represents only a very special case of a nonlocal operator. In general, nonlocal operators can have a much more complicated dependence on the function values $\{u(x) \mid x \in \Omega\}$ over the whole domain, as well as the mutual correlations of these values.

2.2 Averaging Neural Operator: a Special Subclass of the NNO

In this section, we define a special subclass of the NNO, which combines nonlinearity by composition (2.1) with nonlocality by averaging (2.2). Recall that we have restricted this paper to consideration of only smooth, nonpolynomial and Lipschitz continuous activation functions σ. Generalization of our results to other activation functions is possible, but would require additional technical assumptions in our main results. We now define the subclass of NNOs found by simplifying to a special subclass of hidden

layers of the form,

$$\mathcal{L} : \mathcal{V}(\Omega; \mathbb{R}^{d_c}) \to \mathcal{V}(\Omega; \mathbb{R}^{d_c}), \quad \mathcal{L}(v)(x) := \sigma\left(Wv(x) + b + \fint_\Omega v(y)\,dy\right). \quad (2.3)$$

More specifically, we will focus on the case of *a single hidden layer*. This results in the following *averaging neural operator* (ANO) architecture:

$$\Psi : \mathcal{X}(\Omega; \mathbb{R}^o) \to \mathcal{Y}(\Omega; \mathbb{R}^o), \quad \Psi(u) = \mathcal{Q} \circ \mathcal{L} \circ \mathcal{R}(u),$$

where the lifting and projection mappings, $\mathcal{R}$ and $\mathcal{Q}$, are given by (1.1) and (1.3), respectively, with R and Q single-hidden layer neural networks of width d_c, and where $\mathcal{L}$ is of the form (2.3).

We note that this ANO is a specific subclass of the NNO. Due to its minimal structure, the ANO depends on only one *hyperparameter*; the lifting dimension d_c. The *tunable parameters* of ANO are represented by the weight matrix $W \in \mathbb{R}^{d_c \times d_c}$ and bias $b \in \mathbb{R}^{d_c}$ in the hidden layer $\mathcal{L}$, and the internal weights and biases of the ordinary neural networks R and Q, which define the lifting and projection layers, respectively.

2.3 Universal Approximation

Despite the apparent simplicity of averaging as the only nonlocal ingredient in the definition of the averaging neural operator, the following theorem shows that this architecture is universal:

Theorem 1 *Let $\Omega \subset \mathbb{R}^d$ be a bounded domain with Lipschitz boundary. For given integers $s, s' \geq 0$, let $\Psi^\dagger : C^s(\overline{\Omega}; \mathbb{R}^k) \to C^{s'}(\overline{\Omega}; \mathbb{R}^{k'})$ be a continuous operator, and fix a compact set $\mathsf{K} \subset C^s(\overline{\Omega}; \mathbb{R}^k)$. Then for any $\epsilon > 0$, there exists an averaging neural operator $\Psi : \mathsf{K} \subset C^s(\overline{\Omega}; \mathbb{R}^k) \to C^{s'}(\overline{\Omega}; \mathbb{R}^{k'})$ such that*

$$\sup_{u \in \mathsf{K}} \|\Psi^\dagger(u) - \Psi(u)\|_{C^{s'}} \leq \epsilon.$$

The previous result is formulated for operators on spaces of continuously differentiable functions. However, it is possible to obtain similar results in many other natural settings. To illustrate the generality of the underlying ideas, we provide a corresponding result in the scale of Sobolev spaces $W^{s,p}$:

Theorem 2 *Let $\Omega \subset \mathbb{R}^d$ be a bounded domain with Lipschitz boundary. For given integers $s, s' \geq 0$, and reals $p, p' \in [1, \infty)$, let $\Psi^\dagger : W^{s,p}(\Omega; \mathbb{R}^k) \to W^{s',p'}(\Omega; \mathbb{R}^{k'})$ be a continuous operator. Fix a compact set $\mathsf{K} \subset W^{s,p}(\Omega; \mathbb{R}^k)$ of bounded functions, $\sup_{u \in K} \|u\|_{L^\infty} < \infty$. Then for any $\epsilon > 0$, there exists an averaging neural operator $\Psi : W^{s,p}(\Omega; \mathbb{R}^k) \to W^{s',p'}(\Omega; \mathbb{R}^{k'})$ such that*

$$\sup_{u \in K} \|\Psi^\dagger(u) - \Psi(u)\|_{W^{s',p'}} \leq \epsilon.$$

A straight-forward consequence of the above results is that any more general NNO architecture, with hidden layers of the form (1.2), is universal as long as an average can be represented; under this condition, the resulting NNO reduces to ANO with a specific setting of the tunable weights (by setting certain parameters to zero). This is ensured, provided that there exists an index m such that $\psi_{\ell,m}, \phi_{\ell,m} \equiv \text{const.}$ are constant functions, and satisfied for most architectures that are successfully employed in practice (cf. Section 3 for several examples).

Remark 1 Theorems 1 and 2 show that the ANO architecture is universal in approximating a large class of operators, uniformly over a compact set of input functions K. In practice, neural operators are often trained by minimizing an empirical loss,

$$\mathcal{L}(\Psi) = \frac{1}{N} \sum_{n=1}^{N} \|\Psi^{\dagger}(u_n) - \Psi(u_n)\|_{W^{s',p'}(\Omega)}^2, \quad u_1, \ldots, u_N \sim \mu,$$

where the data are iid random samples from an underlying input probability measure μ. A popular prototypical choice is sampling input functions from a Gaussian random field. In this case, the set of input functions is no longer bounded, and hence not compact. As previously observed in [26, Remark 17 and Thm. 18], we note that it is possible to combine universality results with respect to the uniform norm over a compact set K, with a technical cut-off argument, to derive corresponding results in expectation; i.e. for given $\epsilon > 0$, it is possible to show the existence of Ψ, such that

$$\mathbb{E}_{u \sim \mu} \left[\|\Psi(u) - \Psi^{\dagger}(u)\|_{W^{s',p'}}^2 \right] < \epsilon,$$

including the case where μ does not have compact support.

2.4 Intuition

Before providing a sketch of the main technical elements that go into the proof of Theorems 1 and 2, in the next subsection 2.5, we would like to provide some further remarks on the general intuition, as well as possible variants and extensions of these theorems.

Encoder-Decoder Structure. The main theoretical insight of this work, brought out in the ANO architecture, is the identification of a hidden encoder-decoder structure which is inherent in NNOs. This structure becomes particularly apparent when setting the matrix W and bias b in the hidden layer (2.3) zero, in which case we note that the mapping $\mathcal{L} \circ \mathcal{R} : \mathcal{X} \to \mathbb{R}^{d_c}$, $u \mapsto \mathcal{L} \circ \mathcal{R}(u)$ can be thought of as a nonlinear encoding of the input function by a (constant) vector $v \in \mathbb{R}^{d_c}$, while the mapping $\mathcal{Q} : \mathbb{R}^{d_c} \to \mathcal{Y}$, $v \mapsto \mathcal{Q}(v, \cdot)$ is a nonlinear decoding of the corresponding output function $\Psi^{\dagger}(u)$. Indeed, with this choice of the hidden layer, the composition $\mathcal{L} \circ \mathcal{R}$ has constant output,

$$v := \mathcal{L} \circ \mathcal{R}(u) = \sigma \left(\fint_{\Omega} R(u(x), x)\, dx \right) \in \mathbb{R}^{d_c}, \tag{2.4}$$

thus serving as an encoder, $\mathcal{L} \circ \mathcal{R} : \mathcal{X}(\Omega; \mathbb{R}^o) \to \mathbb{R}^{d_c}$. The encoded input v is in turn decoded by the neural network $\mathcal{Q} : v \mapsto Q(v, \cdot)$ to produce the output function $\Psi(u)(x) = Q(v, x)$.

This encoder/decoder perspective provides further insight as to why the universal approximation property can hold even when relying on simple averaging as the only nonlocal ingredient; the main difficulty in proving such a result lies in showing that composition of the input function with a non-linear transformation, and subsequent averaging, can encode the relevant features about the input.

The Role of Positional Encodings.

For the encoder-decoder point of view presented above, the explicit x-dependence ("positional encodings") of the lifting and the projection layers appears crucial at first sight, and we would like to further clarify the precise sense in which this is so. Specifically, we would like to address the fact that in practical implementations, of e.g. FNO, it is standard to add such positional encoding to the inputs, while it is not common to add such features to the output layer.

First, we point out that some form of positional encoding is necessary even from the point of view of universality. E.g. in the case of the FNO (on a periodic domain), positional encoding represents the only mechanism by which the architecture can break translation equivariance, i.e. ensure that the FNO Ψ does not commute with the shift operator, $\tau_h \circ \Psi = \Psi \circ \tau_h$, where $\tau_h(u) = u(\cdot + h)$. In many cases, such as for PDEs with non-constant coefficients, the operator of interest is not translation equivariant, making this mechanism for breaking translation equivariance crucial. In previous theoretical work on universality for FNO [26], translation equivariance was broken by assuming an architecture with x-dependent *bias functions*, $b = b(x)$. However, in practical implementations, it is more common to add a positional encoding explicitly in the input layer $\mathcal{R} : u \mapsto R(u(x), x)$, consistent with the formulation in the present work.

Second, we note that the explicit x-dependence in the output layer is not necessary, provided that the pointwise matrix multiplication $v(x) \mapsto Wv(x) + b$ is included in the hidden layers:

$$v(x) \mapsto \sigma \left(Wv(x) + b + \fint_\Omega v(y) \, dy \right).$$

Indeed, these pointwise operations provide the network a mechanism (akin to skip connections) to forward positional encoding information from the input layer to the output layer.

To be more specific, if $\mathcal{Q}(v)(x) = Q(v(x))$ has no explicit x-dependence, then we can choose a lifting layer of the form $R(u(x), x) = [R_1(x), R_2(u(x), x)]^T$, where $R_1(x) \approx x$ is an approximation of the identity on Ω, and we can choose a block matrix $W = [W_1, 0]$ and bias $b = [b_1, 0]^T$ with $b_1 = -\fint_\Omega R_1(y) \, dy + b_*$, which implies that

$$\mathcal{Q} \circ \mathcal{L} \circ \mathcal{R}(u) = \mathcal{Q} \circ \sigma \left(\begin{bmatrix} R_1(x) + \fint_\Omega R_1(y) \, dy + b_1 \\ \fint_\Omega R_2(u(y), y) \, dy \end{bmatrix} \right).$$

$$= Q \left(\begin{bmatrix} \sigma \left(W_1 R_1(x) + b_* \right) \\ \sigma \left(\fint_\Omega R_2(u(y), y) \, dy \right) \end{bmatrix} \right).$$

Comparing with (2.4), the second input component of Q, i.e. $v := \sigma \left(\fint_\Omega R_2(u(y), y) \, dy \right)$, is of the same encoder-form as before. As a slight difference, the explicit positional encoding by x is now replaced by $\xi(x) = \sigma \left(W_1 R_1(x) + b_* \right)$. However, since both $R_1(x) \approx x$ and $\sigma \left(W_1 x + b_* \right)$ can be chosen to be an arbitrarily accurate approximation of the identity, then with a suitable choice of R_1, W_1 and b_*, this first component $\xi(x) \approx x$ can be used to replace the explicit position encoding x.

Introducing an explicit x-dependence directly in the output layer Q is thus not crucial from a universality point of view. We nevertheless assume such dependence, because (i) it allows us to simplify the proofs by avoiding technical details that would be required to make the above statement precise, and (ii) the assumed explicit x-dependence also points to a generalization of the architecture to approximate operators on input-/output-functions with distinct domains, as will be explained below.

Other Extensions and Variants. There are several possible extensions and variants of our main theorems, two of which we briefly mention below.

Firstly, our derivation actually implies that neural operators of the form (2.7) are universal in approximating continuous operators $\Psi^\dagger : \mathcal{X}(\Omega; \mathbb{R}^k) \to \mathcal{Y}(\Omega'; \mathbb{R}^{k'})$ on *distinct* domains $\Omega \subset \mathbb{R}^d$ and $\Omega' \subset \mathbb{R}^{d'}$, and with potentially different dimensions $d \neq d'$. We only need to replace Q in (2.7) by a neural network mapping $Q' : \mathbb{R}^{d_c} \times \Omega' \to \mathbb{R}^{k'}$. The input and output spaces $\mathcal{X}, \mathcal{Y}$ can be any combination of $C^s(\overline{\Omega}; \mathbb{R}^k)$, $W^{s,p}(\Omega; \mathbb{R}^k)$ and $C^{s'}(\overline{\Omega}'; \mathbb{R}^{k'})$, $W^{s',p'}(\Omega'; \mathbb{R}^{k'})$, respectively, for any integers $s, s' \geq 0$ and reals $p, p' \in [1, \infty)$. Extension to this setting specifically requires that the pointwise matrix multiplication, $v(x) \mapsto W v(x)$, in the hidden layer is removed, by setting $W = 0$. It would be interesting to investigate the practical utility of this observation in numerical experiments, in the future.

Secondly, it should be straight-forward, although technically more involved, to extend the results of this work to operators mapping between functions defined on compact manifolds $M \subset \mathbb{R}^d$ in place of Ω; this is particularly relevant to applications of neural operators to numerical weather forecasting [5, 45].

2.5 Sketch Of The Proof Of Universality

In the present section, we provide an overview of the proof of Theorems 1 and 2. The details are included in Appendix A, see Sections A.5 and A.6, in particular. In the following overview, let $\mathcal{X} = \mathcal{X}(\Omega; \mathbb{R}^k)$ and $\mathcal{Y} = \mathcal{Y}(\Omega; \mathbb{R}^{k'})$ denote spaces of continuously differentiable functions (respectively Sobolev spaces) as in the statement of Theorem 1 (respectively Theorem 2).

The first step in the proof of universality is to determine a reduction to a special class of operators, explained in detail in Section 1, which have the structure of an encoder-decoder. Starting from a general continuous operator $\Psi^\dagger : \mathcal{X} \to \mathcal{Y}$, and given a compact set $\mathsf{K} \subset \mathcal{X}$, we first show that for any $\epsilon > 0$, there exist $J \in \mathbb{N}$,

elements $\eta_1, \ldots, \eta_J \in \mathcal{Y}$ and continuous functionals $\alpha_1, \ldots, \alpha_J : \mathcal{X} \to \mathbb{R}$, such that

$$\sup_{u \in \mathsf{K}} \left\| \Psi^\dagger(u) - \sum_{j=1}^J \alpha_j(u) \eta_j \right\|_{\mathcal{Y}} \le \epsilon. \tag{2.5}$$

Furthermore, our construction ensures that each of these functionals α_j actually defines a continuous functional on the larger input function space consisting of L^1-functions, i.e. we have continuous mappings $\alpha_j : L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}$. Since any input function space $\mathcal{X}$ considered in Theorems 1 and 2 embeds into $L^1(\Omega; \mathbb{R}^k)$, this first step in the proof reduces the original problem to the approximation of a continuous operator $\widetilde{\Psi}^\dagger : L^1(\Omega; \mathbb{R}^k) \to \mathcal{Y}$, of the form,

$$\widetilde{\Psi}^\dagger(u) = \sum_{j=1}^J \alpha_j(u) \eta_j, \tag{2.6}$$

over a compact subset $\mathsf{K} \subset L^1(\Omega; \mathbb{R}^k)$ of uniformly bounded functions, $\sup_{u \in K} \|u\|_{L^\infty} < \infty$. The structure of $\widetilde{\Psi}$ is that of an encoder-decoder, with the functionals α_j providing the encoding and the functions η_j the decoding. We refer to Proposition A.6 for further details in the setting of continuously differentiable spaces, and Proposition A.8 for a corresponding statement in the scale of Sobolev spaces.

The second step in the proof, described in detail in Section 1, concerns approximation of the continuous nonlinear functionals $\alpha_j : L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}$, the encoders. Specifically, we show that they can be approximated to any desired accuracy ϵ by an averaging neural operator $\widetilde{\alpha} : L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}^J$ (with constant output functions), and with components $\widetilde{\alpha}(u) = (\widetilde{\alpha}_1(u), \ldots, \widetilde{\alpha}_J(u))$. To be concrete,

$$\sup_{u \in \mathsf{K}} \left| \alpha_j(u) - \widetilde{\alpha}_j(u) \right| \le \epsilon, \quad \forall \, j = 1, \ldots, J.$$

This result is contained in Lemma A.9 of Section 1, which constitutes the main ingredient in the proof of universality. More precisely, the constructed averaging neural operator $\widetilde{\alpha}$ is of the form

$$\widetilde{\alpha} : u \mapsto R(u(x), x) \mapsto v := \sigma \left(\fint_\Omega R(u(y), y) \, dy \right) \mapsto q_1(v),$$

where $R : \mathbb{R}^k \times \Omega \to \mathbb{R}^{d_c}$ and $q_1 : \mathbb{R}^{d_c} \to \mathbb{R}^J$ are ordinary neural networks; see Remark A.10 after the proof of Lemma A.9. This step builds on well-known approximation results for ordinary neural networks, summarized in Section A.2, and combines them with an averaging operation.

The final third step in the proof of Theorem 1 is given in Section 5.1 (see p. 30) and is built on approximations of the functions η_j which define the decoder. We first show that there exists another (ordinary) neural network $q_2 : \mathbb{R}^J \times \Omega \to \mathbb{R}^{k'}$, such

that $q_2(v, x) \approx \sum_{j=1}^{J} v_j \eta_j(x)$ for all relevant (v, x), approximating the decoding. We then define a neural network Q by the composition $Q(v, x) := q_2(q_1(v), x)$.

Given the neural network R and averaging layer $\mathcal{L}(w) := \sigma\left(\fint_\Omega w(y)\, dy\right)$ from step two of the proof, and the neural network Q from step three, we show that the averaging neural operator $\Psi : L^1(\Omega; \mathbb{R}^k) \to \mathcal{Y}$, given by the composition,

$$\Psi : u(x) \overset{\mathcal{R}}{\mapsto} R(u(x), x) \overset{\mathcal{L}}{\mapsto} v := \sigma\left(\fint_\Omega R(u(y), y)\, dy\right) \overset{\mathcal{Q}}{\mapsto} Q(v, x), \qquad (2.7)$$

approximates the encoder-decoder (2.6). Concretely, given any $\epsilon > 0$, we may design the neural networks R and Q so that

$$\sup_{u \in \mathsf{K}} \left\| \Psi(u) - \sum_{j=1}^{J} \alpha_j(u)\eta_j \right\|_{\mathcal{Y}} \le \epsilon.$$

Step one of the proof shows that an encoder-decoder can be constructed to $\epsilon-$approximate the true map $\Psi^\dagger$, as stated in (2.5). It thus follows that averaging neural operators are universal approximators of general continuous operators $\Psi^\dagger : \mathcal{X} \to \mathcal{Y}$.

Two important properties of $\mathcal{X}$ and $\mathcal{Y}$ that are used in this proof are that (a) $\mathcal{X}$ has a continuous embedding in $L^1(\Omega; \mathbb{R}^k)$, and (b) that functions $\eta_1, \dots, \eta_J \in \mathcal{Y}(\Omega; \mathbb{R}^{k'})$ can be approximated by ordinary neural networks, i.e. for any $\epsilon > 0$, there exist neural networks $\widetilde{\eta}_1, \dots, \widetilde{\eta}_J : \Omega \to \mathbb{R}^{k'}$, such that $\|\eta_j - \widetilde{\eta}_j\|_{\mathcal{Y}} \le \epsilon$. In addition, a more technical ingredient for the reduction to an operator of the form (2.6) is the existence of a "mollification", defining a family of maps $\mathcal{M}_\delta : L^1(\Omega; \mathbb{R}^k) \to \mathcal{X}$, and satisfying $\|\mathcal{M}_\delta u - u\|_{\mathcal{X}} \to 0$ as $\delta \to 0$, for any $u \in \mathcal{X} \subset L^1$. Relevant mathematical background, including a result on boundary-adapted mollification, is summarized in Appendix A.1. For simplicity of exposition, rather than striving for the greatest generality, we have formulated our universal approximation theorem for two concrete families of function spaces, C^s and $W^{s,p}$, which satisfy the required properties and which cover most settings encountered in applications.

Remark 2 We note that additional properties, such as boundary conditions, are often implicitly enforced, provided that the operator is approximated with respect to a suitable norm on the output function space. For example, if the output functions belong to the Sobolev space $W_0^{1,2}(\Omega; \mathbb{R}^{k'})$, respecting a homogeneous Dirichlet boundary condition, then approximation of the operator with respect to the $W^{1,2}(\Omega; \mathbb{R}^{k'})$-norm automatically ensures that the boundary conditions are at least approximately satisfied, as a consequence of the trace theorem [17, Sect. 5.5].

In the outline above, the parameter J directly relates to the required width of the network Q defining the projection layer $\mathcal{Q}$. Thus, J can be interpreted as a measure of the complexity of the output function space. The channel dimension d_c corresponds to the number of necessary "features", encoded via the components $\sigma\left(\fint_\Omega R_1(u(y), y)\, dy\right), \dots, \sigma\left(\fint_\Omega R_{d_c}(u(y), y)\, dy\right)$ of the composition $\mathcal{L} \circ \mathcal{R}$, needed

to approximate the functionals α_j to a given accuracy. These nonlinear features provide an encoding of the input function, and d_c is a measure of the information content that needs to be retained.

3 Connection With Other Neural Operator Architectures

In this section we discuss the implications of the work in the previous section for, and links with, other neural operator architectures. Several neural operator architectures can be viewed as generalizations of the ANO, and a specific choice of the weights in the hidden layers will reproduce the simple averaging considered in the present work. Examples include the Fourier neural operator, the wavelet neural operator and neural operator with general integral kernel. As a consequence, the universal approximation property for the ANO immediately implies corresponding universal approximation results for these architectures. Furthermore, there are links to emerging and existing neural operator architectures such as the Laplace neural operator [8], DeepONet [38] and NOMAD [48]. We now detail these implications and links.

3.1 Neural Operator With General Integral Kernel

Recall that the hidden layers of an ANO are of the form

$$\mathcal{L}_\ell(v)(x) = \sigma\left(W_\ell v(x) + b_\ell + (\mathcal{K}_\ell v)(x)\right), \tag{3.1}$$

where $\mathcal{K}_\ell$ is the nonlocal operator given by averaging, $\mathcal{K}_\ell v = \fint_\Omega v(y)\,dy$. This is a special case of the general form of neural operators defined in [27] (General Neural Operator), where the hidden layers are of the form (3.1), with nonlocal operator given by integration against a matrix-valued integral kernel $K_\ell(x, y) \in \mathbb{R}^{d_c \times d_c}$:

$$(\mathcal{K}_\ell v)(x) = \int_\Omega K_\ell(x, y)v(y)\,dy. \tag{3.2}$$

The particular choice $K_\ell(x, y) \equiv |\Omega|^{-1} I_{d_c \times d_c}$ with $I_{d_c \times d_c}$ the identity matrix, recovers the ANO. Hence, as an immediate consequence of the universal approximation theorems 1 and 2, we obtain:

Corollary 1 *(General Neural Operator) The neural operator architecture with general integral kernel is universal in the settings of Theorem 1 and Theorem 2.*

This result generalizes the universal approximation theorem of [27] to even more general input and output spaces.

3.2 Low-Rank Neural Operator

A particular choice of the general integral kernel (3.2), proposed in [27, Sect. 4.2], leads to the low-rank neural operator. Here, the integral kernel $K_\ell(x, y)$ of each layer

is expanded in the low-rank form,

$$K_\ell(x, y) = \sum_{k=1}^{r} \phi_{\ell,j}(x)\psi_{\ell,j}(y)^T,$$

where $\phi_{\ell,j}, \psi_{\ell,j} : \Omega \to \mathbb{R}^{d_c}$ are vector-valued functions determined by neural networks, and optimized during the training of the low-rank neural operator. If $r = d_c$, and choosing the neural networks to be constant, $\phi_{\ell,j} \equiv |\Omega|^{-1} e_j$, $\psi_{\ell,j} \equiv e_j$ with $e_j \in \mathbb{R}^{d_c}$ the j-th unit vector, we obtain $K_\ell(x, y) = |\Omega|^{-1} I_{d_c \times d_c}$. This recovers the ANO. Thus, we conclude:

Corollary 2 *(Low-Rank Neural Operator) The neural operator architecture with low-rank kernel with $r \geq d_c$ is universal in the settings of Theorem 1 and Theorem 2.*

Note that the NNO proposed in the present work differs from what is proposed in [27, Sect. 4.2]: here we pre-specify the functions $\phi_{\ell,j}, \psi_{\ell,j}$, and learn only a multiplier; in contrast, the pre-existing work proposed learning these functions as neural networks. In this sense, the NNO can be viewed as a special case of the low-rank neural operator.

3.3 Fourier Neural Operator

The Fourier neural operator is obtained by restricting to a class of convolutional integral kernels in (3.2), i.e. kernels satisfying $K_\ell(x, y) = K_\ell(x - y)$, where $K_\ell(x)$ is a trigonometric polynomial of the form $K_\ell(x) = \sum_{|k| \leq k_{\max}} \widehat{P}_{\ell,k} e^{ikx}$ with $\widehat{P}_{\ell,k} \in \mathbb{C}^{d_c \times d_c}$. Upon setting $\ell = 1, k_{\max} = 0$, and $\widehat{P}_{1,0} = |\Omega|^{-1} I_{d_c \times d_c}$, we again recover the averaging neural operator. Thus, Theorem 1 immediately implies:

Corollary 3 *(Fourier Neural Operator) The Fourier neural operator architecture is universal in the settings of Theorem 1 and Theorem 2.*

This result generalizes the previous universal approximation theorem of [26] to arbitrary Sobolev input and output spaces.

3.4 Wavelet Neural Operator

Multi-wavelet neural operators have been introduced in [19]; see also [50]. Following the approach in [19], the integral kernel $K_\ell(x, y)$ in a given hidden layer is expanded in terms of a wavelet basis $\psi_0, \psi_1, \ldots, \psi_N$,

$$K_\ell(x, y) = \sum_{i,j=0}^{N} C^{(i,j)} \psi_i(x) \psi_j(y).$$

Here, the coefficients $C^{(i,j)} \in \mathbb{R}^{d_c \times d_c}$ are themselves matrix-valued. We consider the case of a single hidden layer, $\ell = 1$, and the setting where the span of the wavelet basis $\psi_0, \ldots$ includes constant functions, and hence we assume that $\psi_0 \equiv$ const. is constant. The ANO is then again a special case, and we have:

Corollary 4 *(Wavelet Neural Operator) If the wavelet basis includes a constant function $\psi_0(x) \equiv$ const., then the wavelet neural operator architecture is universal in the settings of Theorem 1 and Theorem 2.*

This result provides a first universal approximation theorem for wavelet neural operators, and thus gives a firm theoretical underpinning for the methodology proposed in [19].

3.5 Laplace Neural Operator

Recently another alternative to the Fourier neural operator has been proposed in [8]; this so-called *Laplace neural operator*, is applicable to general bounded domains $\Omega \subset \mathbb{R}^d$, and indeed extends to functions defined on quite general smooth manifolds. The approach of [8] employs an expansion in a (truncated) eigenbasis of the Laplacian, $\phi_0, \phi_1, \ldots, \phi_K \in L^2(\Omega)$, and defines the nonlocal kernel $\mathcal{K}_\ell$ in the ℓ-th hidden layer (3.1) by an expression of the form,

$$\mathcal{K}_\ell v := \sum_{k=0}^{K} \langle T_{\ell,k} v, \phi_k \rangle_{L^2} \phi_k,$$

where $T_{\ell,k} \in \mathbb{R}^{d_c \times d_c}$ is a learnable matrix multiplier for $k = 0, \ldots, K$.

If the eigenbasis is computed using, for example, homogeneous Neumann boundary conditions, then the constant function $\phi_0(x) \equiv 1/\sqrt{|\Omega|}$ is the eigenfunction corresponding to the lowest eigenvalue $\lambda_0 = 0$ of the Laplacian, this architecture reproduces the averaging neural operator with the particular choice $T_{\ell,0} := I_{d_c \times d_c}$, and $T_{\ell,k} := 0$, for $k \geq 1$. Thus, we conclude from the universality of averaging neural operators, a result which places the Laplace neural operator [8] on a firm theoretical basis:

Corollary 5 *(Laplace Neural Operator) The Laplace neural operator architecture is universal in the settings of Theorem 1 and Theorem 2.*

3.6 DeepONet

In the course of proving universality for the averaging neural operator, outlined in Section 2.5, we prove several approximation results which demonstrate connections with other architectures, such as the Deep Operator Network (DeepONet) architecture [38]. We recall that a DeepONet is based on two ingredients which we now detail. Firstly, a set of linear functionals $\ell = (\ell_1, \ldots, \ell_m) : \mathcal{X}(\Omega; \mathbb{R}^k) \to \mathbb{R}^{m \times k}$ have to be specified; often, these functionals are defined by simple point evaluation, e.g. $\ell_j(u) = u(x_j)$, $j = 1, \ldots, m$. Secondly, two neural networks $\beta : \mathbb{R}^{m \times k} \to \mathbb{R}^p$ (the branch net) and $\tau : \Omega \to \mathbb{R}^{p \times k'}$ (the trunk net) are introduced, giving rise to an operator of the form

$$\Psi : u \mapsto \sum_{k=1}^{p} \beta_k(\ell(u)) \tau_k. \tag{3.3}$$

In practice, the networks β, τ are trained from given data consisting of input- and output-pairs $(u_1, \Psi^{\dagger}(u_1)), \ldots, (u_N, \Psi^{\dagger}(u_N))$, where $\Psi^{\dagger} : \mathcal{X} \to \mathcal{Y}$ is the underlying (truth) operator. Given a compact set $\mathsf{K} \subset \mathcal{X}$, the following approximation error is of interest:

$$\sup_{u \in \mathsf{K}} \left\| \Psi^{\dagger}(u) - \sum_{k=1}^{p} \beta_k(\ell(u))\tau_k \right\|_{\mathcal{Y}}. \tag{3.4}$$

Comparing (3.4) with (2.5), we observe that the composition $\beta_k \circ \ell : \mathcal{X} \to \mathbb{R}$ can be thought of as providing an approximation to the nonlinear functional α_k, whereas the trunk net τ_k approximates ϕ_k.

On unstructured domains, the branch net $\beta : \mathbb{R}^{m \times k} \to \mathbb{R}^p$ is often chosen as a fully connected neural network. The analysis in the present work suggests an alternative choice where $\beta \circ \ell$ is replaced by a functional $\widetilde{\alpha} : \mathcal{X} \to \mathbb{R}^p$, defined by

$$\widetilde{\alpha}(u) := \sigma \left(\fint_{\Omega} R(u(x), x) \, dx \right), \tag{3.5}$$

where $R : \mathbb{R}^k \times \Omega \to \mathbb{R}^p$ is an ordinary neural network. In practice, the integral average can be replaced by a sum over a finite number of sensor points $x_1, \ldots, x_m$, resulting in a mapping of the form

$$\widetilde{\alpha}(u) = \sigma \left(\textstyle\sum_{j=1}^{m} R(u(x_j), x_j) \right).$$

A variant of DeepONet is then obtained by defining $\widetilde{\Psi} : \mathcal{X} \to \mathcal{Y}$ by $\widetilde{\Psi}(u) := \sum_{k=1}^{p} \widetilde{\alpha}_k(u)\tau_k$. Our analysis implies that this operator learning architecture $\widetilde{\Psi}$ is universal in the settings of Theorem 1 and 2.
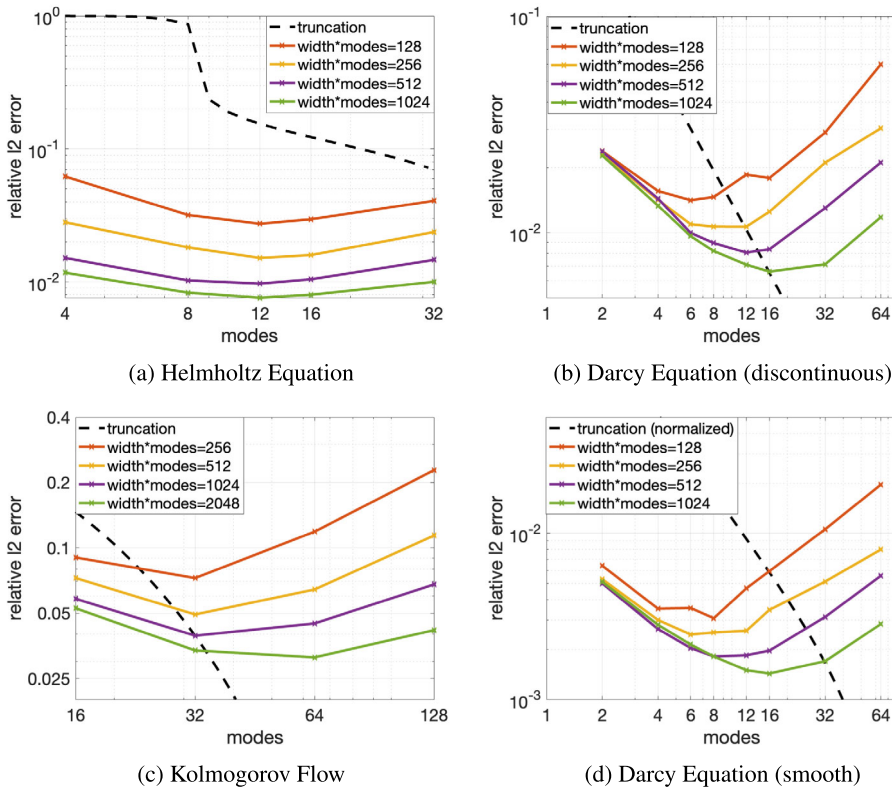
3.7 NOMAD

Going beyond the previous subsection we see that our analysis is also closely linked to a recent extension of DeepONets, called "Nonlinear Manifold Decoder" (NOMAD) [48]. In this approach, the functionals $\ell_1, \ldots, \ell_m : \mathcal{X} \to \mathbb{R}^{m \times k}$ and the branch net $\beta : \mathbb{R}^{m \times k} \to \mathbb{R}^p$ are retained, but the linear expansion in (3.3) is replaced by a nonlinear neural network $Q : \mathbb{R}^p \times \Omega \to \mathbb{R}^{k'}$. The resulting NOMAD operator $\Psi : \mathcal{X} \to \mathcal{Y}$ is defined by

$$\Psi(u)(x) := Q(\beta(\ell(u)), x). \tag{3.6}$$

Replacing the composition $\beta \circ \ell$ by the alternative functional $\widetilde{\alpha}$ defined in (3.5), the NOMAD approach results in an operator $\widetilde{\Psi} : \mathcal{X} \to \mathcal{Y}$ of the form

$$\widetilde{\Psi}(u)(x) := Q(\widetilde{\alpha}(u), x),$$

thus recovering the specific construction of the ANO in (2.7).

(a) Helmholtz Equation

(b) Darcy Equation (discontinuous)

(c) Kolmogorov Flow

(d) Darcy Equation (smooth)

Fig. 1 FNO model with different combinations of channel dimension (width) d_c and Fourier modes K. In each curve, the models share roughly the same amount of model parameters because $d_c K$ is fixed at C, with C changing between each curve. As C increases, the overall error drops. The curves at any given fixed C exhibit a "U"-shape, where the valley determines the optimal choice of modes K. The black dotted line is the error given by simply truncating the truth at the given number K of Fourier modes ("Fourier truncation"). Two things are notable: (i) for Helmholtz the optimal number of modes is fixed as computational budget increases, whilst for the two Darcy problems and Kolmogorov flow it grows; (ii) in all examples the trained FNO is able to considerably improve on the Fourier truncation, when smaller numbers of Fourier modes are used; this suggests the nonlinear approximation theoretic mechanisms at play and resulting from the architecture (Color figure online)

4 Numerical Experiments

There are two types of numerical experiments that are suggested by the analysis in this paper: (i) to deploy the NNO for problems in non-periodic geometries, and in particular to explore choices of problem-adapted functions $\phi_{\ell,m}, \psi_{\ell,m}$ in (1.2), aiming to develop intuition as to when it is an effective operator approximator; (ii) to determine the optimal distribution of parameters to obtain a given approximation error at least cost. We defer experiments of type (i) for future work and concentrate here on experiments of type (ii).

To carry out experiments of type (ii) we work in the setting outlined at the start of subsection 1.2. The channel width is d_c and we employ $L = 4$ layers. The domain

$\Omega = [0, 1]^d$ and in all examples considered $d = 2$. The layer takes the specific form

$$(\mathcal{L}_\ell v)(x)$$
$$:= \sigma \left(W_\ell v(x) + b_\ell + \sum_{k \in \mathbb{K}} \langle T_{\ell,m} v, \exp(2\pi i \langle k, x \rangle) \rangle_{L^2(\Omega; \mathbb{C}^{d_c})} \exp(2\pi i \langle k, x \rangle)(x) \right),$$

where $\mathbb{K}$ is a finite square lattice set of cardinality K^2, centred at the origin. Thus $|k|_\infty \le K$. In our experiments we fix a measure of the total number of parameters: $d_c \times K$. We then identify the optimal choice of number of basis functions K, in terms of the achieved test accuracy, for fixed total number of parameters. We conduct experiments on three sets of partial differential equations to study the balance between number of channels and number of Fourier modes: the Helmholtz Equation for wave-propagation in a heterogeneous medium, the Darcy Equation for porous medium flow, for both smooth and piecewise constant coefficients, and the forced two-dimensional Navier-Stokes equation, more specifically a Kolmogorov Flow.

For all three test cases, we use a 4-layer FNO model with the GeLU activation function, as in [33], and a cosine annealing optimizer [37]. In the Helmholtz and Darcy equations, we normalize the input and output functions by subtracting the pointwise mean and dividing by the pointwise standard deviation, with mean and pointwise variance computed from the training dataset; however in the Kolmogorov Flow, we do not use normalization. The results show that, in the range of parameter scenarios we consider, the Helmholtz Equation and Kolmogorov Flow are optimized by fixing the number of basis functions; in contrast the Darcy problem is optimized by increasing the total number of basis functions, given a fixed budget of total number of parameters. These results resonate with the different proofs of universal approximation for the FNO, the original one using an increasing number of Fourier modes to achieve smaller error [26] and the proof in this paper using only a single Fourier mode found via integration against the constant function; more generally the results suggest the need for deeper analysis of the deployment of parameters in neural operators to obtain the optimal error/cost tradeoff.

4.1 Helmholtz Equation

We consider the Helmholtz Equation as defined in [13].

Specifically, we consider the following PDE,

$$\left(-\Delta - \frac{\omega^2}{c^2} \right) u = 0, \quad \text{in } \Omega,$$

with frequency $\omega > 0$ and variable coefficient field $c = c(x)$, on a square domain $\Omega = [0, 1]^2$. We impose Neumann boundary conditions,

$$\frac{\partial u}{\partial n} = 0, \text{ on } \partial\Omega_1, \partial\Omega_2, \partial\Omega_4, \quad \frac{\partial u}{\partial n} = u_N, \text{ on } \partial\Omega_3.$$

Here $\partial\Omega_1$, $\partial\Omega_2$, $\partial\Omega_3$ and $\partial\Omega_4$ denote the lower, right, upper and left edges of the square. We fix $u_N(x) = 1_{[0.35 \leq x \leq 0.65]}$.

The frequency ω is fixed at 10^3. The target operator maps the wave speed field to the disturbance field $\Phi^\dagger : c \mapsto u$. The probability measure on the inputs is a pointwise nonlinear transformation of a Gaussian random field: $c(x) = 20 + \tanh(\widetilde{c}(x))$, $\widetilde{c} \sim \mathcal{N}(0, (-\Delta + 9)^{-2})$ where the Laplace operator is equipped with homogeneous Neumann boundary conditions. We use 10^4 instances of data pairs to train the FNO model, each with resolution $10^2 \times 10^2$. We use Adam optimizer and cosine annealing learning rate scheduler, with initial learning rate 1×10^{-3} and weight decay 1×10^{-4}.

4.2 Darcy Equation

The Darcy equation is widely studied in the operator learning literature and we use the set-up employed in [1, 3]. We consider the following PDE,

$$-\nabla \cdot (a\nabla u) = 1,$$

posed in $\Omega = [0, 1]^2$ and subject to homogeneous Dirichlet boundary conditions.

The target operator maps the coefficient field (permeability) to the solution field (pressure) $\Phi^\dagger : a \mapsto u$. We use 10^3 data pairs to train the FNO model, each with resolution 141×141. We use Adam optimizer and cosine annealing learning rate scheduler, with initial learning rate 1×10^{-3} and weight decay 1×10^{-4}.

We study two cases of the Darcy equation, differing in our choice of random input fields: piecewise-constant coefficients and log-normal coefficients. For the piecewise-constant case, the coefficient functions a are samples from Gaussian random field $\widetilde{a} \sim \mathcal{N}(0, (-\Delta + 9)^{-2})$ again with homogeneous Neumann boundary conditions. We apply a pointwise nonlinearity which assigns value 4 or 12 depending on the sign of the input: $a(x) = 4$ if $\widetilde{a}(x) \leq 0$, $a(x) = 12$ if $\widetilde{a}(x) > 0$. For the log-normal case, the coefficient function a is sampled from a lognormal distribution, which is equivalent to the exponential of the same Gaussian random field: $a(x) = \exp(\widetilde{a}(x))$.

4.3 Kolmogorov Flow

Our final example is the Kolmogorov Flow with set-up as defined in [32]. For this problem, we solve the periodic Navier-Stokes equation on a torus [1]$\mathbb{T}^2 \simeq [0, 2\pi]^2$

$$\partial_t u + u \cdot \nabla u + \nabla p = \frac{1}{Re}\Delta u + \sin(ny)\widehat{x}, \quad \nabla \cdot u = 0,$$

with Reynolds number $Re = 5 \times 10^2$; this Reynolds number leads to a challenging example in which more high-frequency structures are present in comparison to the previous cases considered in the literature. We define the vorticity $\omega = \text{curl } u$ and seek to learn the map from the initial vorticity field to the vorticity field one time unit

---

[1] A straightforward rescaling of the domain used to define the neural operator above, from $[0, 1]^d$ to $[0, 2\pi]^2$, is employed.

later: $\Phi^\dagger : \omega(t) \mapsto \omega(t + h)$. We choose the time unit $h = 0.1$, noting that learning the solution operator on long time-horizons can be difficult because of the chaotic behavior present at this Reynolds number. [2] The equations may be formulated using the vorticity-stream function formulation and the numerical method we employ to generate the data is a pseudo-spectral method applied to this formulation. The dataset of input-output pairs is computed from 100 trajectories, with 80 used for training and 20 for testing. Each trajectory consists of 500 time units; the burn-in phase, during which the dynamics evolves towards to attractor, is removed from the trajectory. We sample the initial conditions from a Gaussian random field of similar form to the two preceding examples, but with $-\Delta$ subjected to periodic boundary conditions. In total, we use 4×10^4 data pairs to train the FNO model, each with resolution 128×128. Note that the data is not i.i.d. We use the Adam optimizer and cosine annealing learning rate scheduler, with initial learning rate 5×10^{-4} and weight decay 1×10^{-4}.

4.4 Results

We study the accuracy of the trained FNO for different combinations of channel dimension (or width) d_c and Fourier modes K, with a given budget of the total number of parameters. Since the number of parameters scales as $O(d_c^2 K^2)$ for these 2-dimensional problems, we fix $d_c K = C$ as a convenient proxy to control the sizes of our models. The main theoretical result in this paper demonstrates that, in principle, universality can be achieved by simply retaining a single Fourier mode. In practice, it may be beneficial to include a larger number of modes, and we are interested in the *optimal* choice of K (and hence d_c), given a fixed model size C. To study this refined question empirically, we consider different model sizes, with $C = 128, 256, 512, ...,$ and scan over increments of the Fourier mode cut-off $K = 2, 4, 8, ...$ The results of this empirical study are collected in Figure 1, for the four settings summarized above; in each case, the relative L^2-errors are plotted as a function of the number of the Fourier truncation parameter K, and each solid curve represents the result of computations with a fixed number C of model parameters. As a baseline, the dashed black line shows the average relative L^2-error that is achieved by a direct Fourier truncation of the reference output functions, as a function of K, i.e. the average error over the test distribution that is obtained by projecting the reference solution of the PDE to an expansion of its Fourier modes with wavenumbers $|k|_\infty \leq K$.

Overall, we observe that models with more parameters perform better. Moreover, for a fixed model size C, the solid curves describe a "U" shape as a function of K, implying that an optimal choice of the Fourier cut-off K and channel dimension d_c exists, at each given model size C. For the Helmholtz equation the optimal choice of Fourier cut-off K is fixed across different model sizes, within the resolution of our experiments. The trained FNO for the Helmholtz Equation performs best with $K = 12$. For the Kolmogorov Flow, the optimal choice is $K = 32$ for $C = 256, 512, 1024$ except for $C = 2048$ where the optimal mode is $K = 64$. The optimal model is the smallest K larger than the corresponding Fourier truncation mode. A priori, our theoretical

---

[2] Short-time solution operators can be composed using the semigroup property to make longer-term statistical predictions [35].

results are not informative about the *optimal* distribution of parameters. However, they show that even a fixed choice of Fourier modes K, independent of the model size, is not in contradiction with the observed convergence to the underlying operator. This empirical observation would have been difficult to reconcile with previous universal approximation results, which require K to grow without bound.

For the Darcy Equation, the optimal choice of modes clearly shifts to the right with increasing model size. When $C = d_c K = 128$, the optimal choice of modes is $K = 6$. When $d_c K = 1024$, the optimal choice of modes is $K = 16$. Thus, in this case, it appears to be optimal to invest additional degrees of freedom in a combination of increasing both K and d_c. An interesting empirical observation in Figure 1 is that the trained FNO often beats the corresponding Fourier truncation of the true solution with the same number of modes. This is particularly apparent for the Darcy Equation, where even the FNO with a cut-off of only $K = 2$ is able to achieve very accurate results, going considerably beyond the accuracy of the corresponding Fourier truncation baseline. This strongly indicates that the trained FNO discovers a non-linear mechanism to generate a well-calibrated set of higher-order Fourier modes, beyond $K = 2$. Thus, non-linear composition combined with non-locality of the lowest Fourier modes enables accurate results even with very small K; this is the basic mechanism which underlies our main universality result. The empirical findings thus indicate the practical relevance of the basic mechanism identified in the present work, with implications beyond universal approximation.

We include one final set of experiments designed to check that the effect of normalization (applied on the Helmholtz and both Darcy examples) is not responsible for effects apparent in Figure 1. Indeed we show that the desirable approximation effects we observe there cannot simply be attributed to the pre- and post-processing of the inputs and outputs of the neural network to ensure (empirical) mean zero and pointwise standard deviation one. To verify this assertion we show, in Figure 2, the error found simply by Fourier truncation, with and without the normalization applied. In Figure 2 (a), the truncation is done without the normalization. In Figure 2 (b), the truncation is done with the normalization: the solution function is first normalized by the pointwise mean and standard deviation, then truncated to the requisite number of Fourier modes, and finally denormalized, mimicking the procedure used in the model training. The truncation with normalization has a smaller error rate on the lower frequency modes, but the error at higher wavenumbers is similar to that arising without the truncation. The pointwise mean and standard deviation are shown in 2 (c, d), which demonstrates that the normalization captures some lower frequency structures. This experiment shows that normalization help to express the lower frequency components, but it does not account for the desirable approximation properties observed for the neural operator in Figure 1. The Figure 2 is for the Helmholtz equation, but similar effects are observed for the Darcy flow.

5 Discussion

This paper considers a general framework of nonlocal neural operators (NNO), which includes known frameworks such as the Fourier neural operator (FNO). In a nutshell,

(a) Helmholtz Equation

(b) Helmholtz Equation (normalized)

(c) Mean of Helmholtz

(d) Standard deviation of Helmholtz
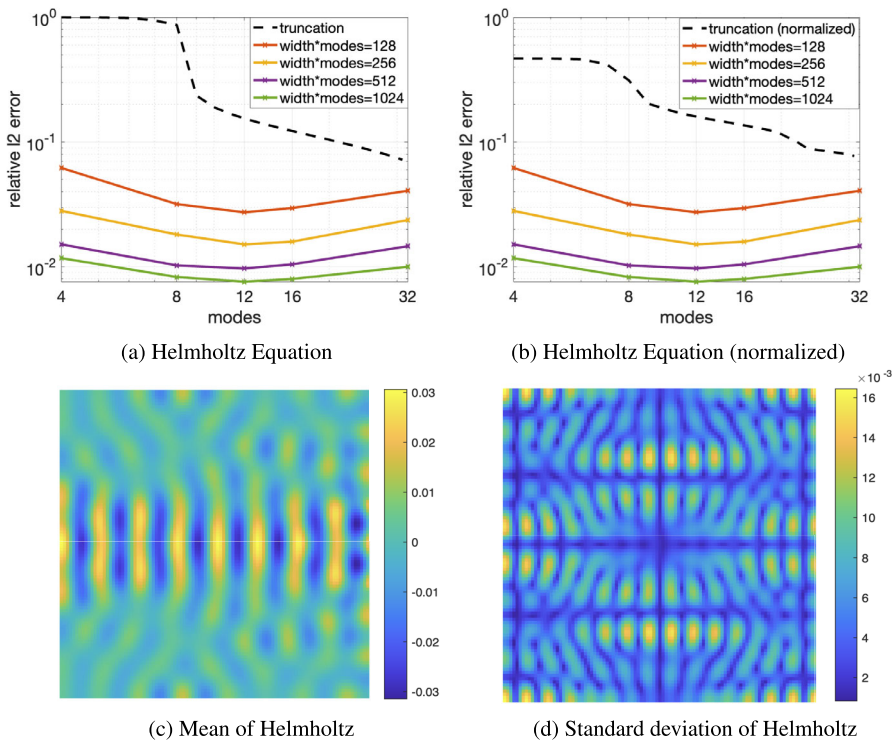
Fig. 2 The effect of normalization demonstrated for the Helmholtz Equation example. (a) Error rates as shown in Figure 1a). (b) Same as Figure 1a) but with error rates of the Fourier truncation including the effect of normalization. (c) and (d) show the mean and standard deviation, respectively, as used in the normalization process. Normalization assists in capturing lower modes but it does not affect the higher modes

we derive sufficient (minimal) conditions under which specific instances of this general framework are guaranteed to possess a universal approximation property. To answer this question, the generic NNO architecture is reduced to its simplest possible form, the averaging neural operator (ANO). This ANO is based on only two minimal ingredients, both of which are indispensable for operator learning: *nonlinearity*, by composition with ordinary neural networks, and *nonlocality*, by averaging. Despite the simplicity of the resulting architecture, we show that the averaging neural operator (ANO) possesses a universal approximation property for a large class of operators, cf. Theorems 1 and 2.

The methodology provides a unifying perspective on universal approximation, across a wide-range of neural network architectures, and also suggests new architectures, to which universal approximation will also apply. It allows us to unify much of the theoretical analyses of, for example, the general NO, FNO, low-rank NO, wavelet NO and Laplace-NO; the ANO is employed as an analysis tool to derive new universality results for all of these architectures, leading to Corollaries 1–5. These corollaries improve on known universal approximation results for general NOs and FNOs, and

provide a first rigorous theoretical underpinning for the methodology in the case of low-rank, wavelet and Laplace NOs. Furthermore we believe that our general universality result for the ANO will serve as a useful reference to prove universality for many other emerging neural operator architectures. In this context, we also point out close links of the ANO introduced in the present work with other recent proposed neural operator architectures, including the DeepONet and NOMAD architectures. In fact, our proof of universality of the ANO is based on identifying an inherent encoder-decoder structure in this architecture, and proving universality of such encoder-decoders.

At a more fundamental level, the present work aims to shed new light on the role of nonlocality in operator learning for a general class of neural operators. It is shown that even very simple nonlocality in the form of an averaging operation is sufficient for the universality of neural operator architectures. As a consequence, we show that neural operator architectures possess a universal approximation property even if the nonlocality in the hidden layers is not capable of exploring the space of all integral operators, thus providing considerable scope for the introduction of nonlocality in new ways which may not rely on expansions in complete (e.g. Fourier or wavelet) bases. As a byproduct, we also further deepen our understanding of the Fourier neural operator; in particular, the present work strongly indicates that the importance of the Fourier transform for FNOs may be mainly in providing nonlocality to the architecture, with the particular choice of a Fourier basis playing only a subordinate role.

The present work provides substantial motivation to further study the optimal combination of nonlinearity and nonlocality in operator learning. First empirical results in this direction are presented in Section 4, focusing on the Fourier neural operator. Here, the relevant parameters are the channel dimension and the number of Fourier modes, corresponding to different distributions of nonlinearity and nonlocality in the architecture. Our empirical results show that the optimal distribution of parameters is likely problem dependent, but strongly indicate that the basic mechanism underlying our proof of universality is also relevant in practice; further numerical results, in tandem with extended analysis, will be necessary to gain deeper insight into the optimal trade-off, in terms of cost versus accuracy, between non-linearity and non-locality.

It is also interesting to compare and contrast the ANO architecture with convolutional neural networks (CNN) and convolutional neural operators (CNO). The ANO employs a global average, applied to a non-linear transformation of the input function, to encode relevant features; in contrast, convolutional architectures like CNN and CNO are based on a composition of non-linear transformations, convolutions with localized kernels, and potentially interspersed with max-pooling (or average pooling) operations. From the point of view of universality, we note that the ANO encoder defined via a global average in the present work, could equally well have been replaced by local averaging over a set of patches covering the domain, providing a link to average pooling in CNNs. One major difference between ANO and CNN/CNO is that the layerwise nonlocality is not global for the latter, thus requiring a minimal depth to communicate nonlocal information across the entire domain. We believe it would be interesting to extend the analysis of the present work to CNOs, in the future.

The fact that universality can be obtained even with a simple global average as the only nonlocal ingredient in the hidden layers opens the way for the exploration of alternative neural operator architectures with *guaranteed universality*, relying on

different choices of how nonlocality is introduced. The results of this work are also particularly relevant for future extensions of neural operator architectures to problems with non-periodic geometries and mappings between functions on different domains. We plan to expand on this topic, and the potential benefits of specific choices, in the future.

In another direction, it would also be interesting to derive quantitative error estimates for ANO, to complement the qualitative analysis of the present work. Our initial numerical experiments clearly indicate practical benefits, in terms of the accuracy of the trained models, of introducing nonlocality beyond simple averaging; it would be interesting to see if this is also born out by a quantitative analysis. However, in view of the great (theoretical) approximation power of neural networks as function approximators, here used to define the encoder and decoder in the ANO, it is possible that an explanation of the observed empirical differences would require going beyond a purely approximation theoretic viewpoint, and taking into account their training by gradient-based optimization on finite data.

A Detailed Proof Of Universal Approximation For ANO

A.1 Analysis Ingredients: Extension And Mollification

Lipschitz Domains

We recall that a subset $\Omega \subset \mathbb{R}^d$ is a **bounded Lipschitz domain**, if Ω is open, the closure $\overline{\Omega}$ is compact, and the boundary $\partial \Omega = \overline{\Omega} \setminus \Omega$ is at least "Lipschitz regular", in the sense that it can locally be thought of as the graph of a Lipschitz map. We refer to [16, Section 2.1] for a rigorous mathematical definition. Lipschitz domains cover most domains encountered in physical applications. In particular, any bounded domain with piecewise smooth boundary is a Lipschitz domain.

Function Extension for Lipschitz Domains

In our proofs, it will sometimes be convenient to extend a given function $u : \Omega \to \mathbb{R}$, defined on a domain $\Omega \subset \mathbb{R}^d$, to a function $u : \mathbb{R}^d \to \mathbb{R}$, defined on all of $\mathbb{R}^d$. The following lemma shows that this is possible, while preserving smoothness of u, and relies on a classical result of Stein [49, Chapter 6, Theorem 5]:

Lemma A.1 *(Periodic extension operator, see e.g. [26, Lemma 41]) Let $\Omega \subset \mathbb{R}^d$ be a bounded Lipschitz domain. There exists a continuous, linear operator $\mathcal{E} : W^{s,p}(\Omega) \to W^{s,p}_{\mathrm{per}}(B)$ for any $s \geq 0$ and $p \in [1, \infty]$, where $B \subset \mathbb{R}^d$ is a bounded hypercube containing $\Omega \subset B$, such that for any $u \in W^{s,p}(\Omega)$:*

1. *$\mathcal{E}(u)|_\Omega = u$;*
2. *$\mathcal{E}(u) \in W^{s,p}_{\mathrm{per}}(B)$ is periodic on B (including its derivatives).*

Furthermore, $\mathcal{E}$ maps continuously differentiable functions to continuously differentiable functions, i.e. $\mathcal{E}(C^s(\overline{\Omega})) \subset C^s_{\mathrm{per}}(B)$ and hence defines a continuous mapping $\mathcal{E} : C^s(\overline{\Omega}) \to C^s_{\mathrm{per}}(B)$.

Mollification (Smoothing) of Functions on Lipschitz Domains Recall that there exists a smooth mapping (a mollifier) $\rho : \mathbb{R}^d \to \mathbb{R}$, with properties $\rho(x) \in [0, 1]$

for all $x \in \mathbb{R}^d$, $\rho(0) = 1$, and $\rho(x) = 0$ for $|x| \geq 1$. Furthermore we can normalize ρ to enforce $\int_{\mathbb{R}^d} \rho(y) \, dy = 1$. Any such ρ defines a family of functions $\rho_\delta(x) := \delta^{-d} \rho(x/\delta)$, supported in a δ-ball around the origin. Fixing such a family, we recall that the ϵ-mollification of a function $u : \mathbb{R}^d \to \mathbb{R}$, $u \in L^1(\mathbb{R}^d)$ is defined by a convolution $u_\delta(x) := (u * \rho_\delta)(x)$, i.e.

$$u_\delta(x) = \int_{\mathbb{R}^d} u(x - y) \rho_\delta(y) \, dy.$$

It is well-known that u_δ is a smooth function for $\delta > 0$, and that $u_\delta \to u$ in spaces such as arbitrary Sobolev spaces $W^{s,p}(\mathbb{R}^d)$, or spaces of continuously differentiable functions $C^s(\mathbb{R}^d)$, with respect to the respective norms. Mollification as defined above is a useful tool in analysis, but is not well-adapted to bounded domains. The papers [4, 16] develop a useful variant of mollification for bounded Lipschitz domains $\Omega \subset \mathbb{R}^d$, where additional care is needed to deal with the behavior near the boundary.

The following results follow from the proof of [4, Theorem 2.4], or can alternatively be obtained from the construction and arguments in [16]:

Lemma A.2 *(Adapted mollification in a Lipschitz domain) Let $\Omega \subset \mathbb{R}^d$ be a bounded Lipschitz domain. There exists a one-parameter family $\mathcal{M}_\delta$ of "mollification" operators, indexed by $\delta \geq 0$ and defining a linear mapping $\mathcal{M}_\delta : L^1(\Omega; \mathbb{R}^k) \to L^1(\Omega; \mathbb{R}^k)$ for $\delta > 0$, such that:*

1. *For $\delta = 0$, we have $\mathcal{M}_0 u = u$ for all $u \in L^1(\Omega; \mathbb{R}^k)$; for $\delta > 0$, $\mathcal{M}_\delta$ is smoothing, in the sense that it defines a mapping $\mathcal{M}_\delta : L^1(\Omega; \mathbb{R}^k) \to C^\infty(\overline{\Omega}; \mathbb{R}^k)$.*
2. *For any fixed $\delta > 0$, integer $s, r \geq 0$, the mapping,*

$$\mathcal{M}_\delta : C^s(\overline{\Omega}; \mathbb{R}^k) \to C^r(\overline{\Omega}; \mathbb{R}^k),$$

   is continuous. Furthermore, if $r \leq s$, then the operator norm is uniformly bounded, that is

$$\sup_{\delta > 0} \|\mathcal{M}_\delta\|_{C^s \to C^r} < \infty.$$

3. *If $s \geq 0$ and $\mathsf{K} \subset C^s(\overline{\Omega}; \mathbb{R}^k)$ is a compact subset, then for any $\delta_0 \geq 0$,*

$$\lim_{\delta \to \delta_0} \sup_{u \in \mathsf{K}} \|\mathcal{M}_{\delta_0} u - \mathcal{M}_\delta u\|_{C^s} = 0.$$

4. *For fixed $\delta > 0$, integer $s, r \geq 0$, and $p \in [1, \infty)$, the mapping*

$$\mathcal{M}_\delta : W^{s,p}(\Omega; \mathbb{R}^k) \to W^{r,p}(\Omega; \mathbb{R}^k)$$

   is continuous.
5. *If $s \geq 0$, $p \in [1, \infty)$ and $\mathsf{K} \subset W^{s,p}(\Omega; \mathbb{R}^k)$ is a compact subset, and $\delta_0 \geq 0$, then*

$$\lim_{\delta \to \delta_0} \sup_{u \in \mathsf{K}} \|\mathcal{M}_{\delta_0} u - \mathcal{M}_\delta u\|_{W^{s,p}} = 0.$$

Remark A.3 At first sight, one might think that the result of Lemma A.2 could easily be obtained by extending the input function $u : \Omega \to \mathbb{R}^k$, to a function $u : \mathbb{R}^d \to \mathbb{R}^k$ by setting $u(x) = 0$ for $\mathbb{R}^d \setminus \Omega$, and then use standard mollification on this extension to obtain a smooth approximant $u_\delta : \Omega \to \mathbb{R}^k$. While this approach ensures smoothness of the mollified function u_δ, it does not ensure convergence in spaces defined with respect to the supremum norm, i.e. for $u \in C^s$ we generally have $\|u_\delta - u\|_{C^s} \not\to 0$ as $\delta \to 0$. The main technical issue is that extension by zero produces a jump discontinuity at the boundary; hence special care needs to be taken near the boundary [4, 16].

The proof is based on the boundary-adapted construction of mollification in [16, eq. (3.4a)], and only requires arguments in real analysis. We will not provide a detailed proof here.

In addition to Lemma A.2, we will furthermore need the following technical lemma:

Lemma A.4 *Fix $s \geq 0$. Let $\mathsf{K} \subset C^s(\overline{\Omega}; \mathbb{R}^k)$ be compact. Then for any $\delta > 0$, the set*

$$\mathsf{K}_\delta := \bigcup_{0 \leq \delta' \leq \delta} \{\mathcal{M}_{\delta'} u \mid u \in \mathsf{K}\}, \tag{A.1}$$

is also compact in $C^s(\overline{\Omega}; \mathbb{R}^k)$.

Proof Let $v_1, v_2, \ldots$ be an arbitrary sequence in K_δ. It suffices to prove that v_j possesses a convergent subsequence $v_{j_\ell} \to v \in \mathsf{K}_\delta$. By definition of K_δ, there exists a sequence $u_1, u_2, \cdots \in \mathsf{K}$, and $\delta_1, \delta_2, \cdots \in (0, \delta]$, such that $v_j = \mathcal{M}_{\delta_j} u_j$ for all $j \in \mathbb{N}$. Since K is compact, there exists a convergent subsequence $u_{j_\ell} \to u \in \mathsf{K}$. Furthermore, extracting another subsequence if necessary (not reindexed), we may assume that $\delta_{j_\ell} \to \delta_\infty \in [0, \delta]$ converges to a limit. If $\delta_\infty > 0$, let $v := \mathcal{M}_{\delta_\infty} u$. If $\delta_\infty = 0$, we set $v := \mathcal{M}_0 u = u$. We note that in either case, we have $v \in \mathsf{K}_\delta$. We claim that $v_{j_\ell} \to v$. To see this, note that

$$\begin{aligned}
\limsup_{\ell \to \infty} \|v_{j_\ell} - v\|_{C^s} &= \limsup_{\ell \to \infty} \|\mathcal{M}_{\delta_{j_\ell}} u_{j_\ell} - \mathcal{M}_{\delta_\infty} u\|_{C^s} \\
&\leq \limsup_{\ell \to \infty} \|\mathcal{M}_{\delta_{j_\ell}} u_{j_\ell} - \mathcal{M}_{\delta_{j_\ell}} u\|_{C^s} \\
&\quad + \limsup_{\ell \to \infty} \|\mathcal{M}_{\delta_{j_\ell}} u - \mathcal{M}_{\delta_\infty} u\|_{C^s} \\
&\leq \limsup_{\ell \to \infty} \|\mathcal{M}_{\delta_{j_\ell}}\|_{C^s \to C^s} \|u_{j_\ell} - u\|_{C^s} \\
&\quad + \limsup_{\ell \to \infty} \|\mathcal{M}_{\delta_{j_\ell}} u - \mathcal{M}_{\delta_\infty} u\|_{C^s}.
\end{aligned}$$

For fixed $u \in C^s$, we have $\mathcal{M}_{\delta_{j_\ell}} u \to \mathcal{M}_{\delta_\infty} u$ in C^s, and hence the second term converges to zero. Furthermore, by Lemma A.2 (1), the one-parameter family $\mathcal{M}_{\delta'} : C^s \to C^s$ consists of uniformly bounded operators, such that

$$\sup_{\ell \in \mathbb{N}} \|\mathcal{M}_{\delta_{j_\ell}}\|_{C^s \to C^s} \leq \sup_{0 < \delta' \leq \delta} \|\mathcal{M}_{\delta'}\|_{C^s \to C^s} < \infty.$$

From the convergence $u_{j_\ell} \to u$ in C^s, it thus follows that the first term above converges to zero. Thus, $v_{j_\ell} \to v$ as claimed. This shows that K_δ is sequentially compact. □

A.2 Universal Approximation Of Neural Networks

It is well-known [46, Thm. 4.1] that a neural network architecture is universal in the class of C^s-functions between Euclidean vector spaces, provided that the activation function σ is nonpolynomial and sufficiently smooth, $\sigma \in C^s$. This in turn implies universality of neural networks in Sobolev spaces $W^{s,p}$ of functions between Euclidean vector spaces. For the convenience of the reader, we include the relevant implication for the present work. Recall that, throughout the paper, we assume that the activation function σ is C^∞, nonpolynomial and Lipschitz continuous.

Lemma A.5 *Let $\Omega \subset \mathbb{R}^d$ be a bounded Lipschitz domain. Then for any function $u : \Omega \to \mathbb{R}^k$, where u belongs to either the space of continuously differentiable functions $C^s(\overline{\Omega}; \mathbb{R}^k)$, or the Sobolev space $W^{s,p}(\Omega; \mathbb{R}^k)$, for integer $s \geq 0$ and $p \in [1, \infty)$, and for any $\epsilon > 0$, there exists a neural network $\widetilde{u} : \Omega \to \mathbb{R}^k$ with activation function σ, such that*

$$\sup_{x \in \Omega} |u(x) - \widetilde{u}(x)| \leq \epsilon.$$

Proof Step 1: We first assume that $u \in C^s(\overline{\Omega}; \mathbb{R}^k)$. By Lemma A.1, there exists a (periodic) extension $U : \mathbb{R}^d \to \mathbb{R}$, such that $u(x) = U(x)$ for $x \in \Omega$, and $U \in C^s(\mathbb{R}^d; \mathbb{R}^k)$. It follows from [46, Thm. 4.1] that for any $\epsilon > 0$, there exists a neural network $\widetilde{u} : \mathbb{R}^d \to \mathbb{R}$, such that

$$\|u - \widetilde{u}\|_{C^s(\overline{\Omega}; \mathbb{R}^k)} = \sup_{|\alpha| \leq s} \sup_{x \in \overline{\Omega}} |D^\alpha U(x) - D^\alpha \widetilde{u}(x)| \leq \epsilon.$$

Step 2: If $u \in W^{s,p}(\Omega; \mathbb{R}^k)$, then we note that by Lemma A.2, the boundary-adapted mollification $u_\delta := \mathcal{M}_\delta u$ converges to u as $\delta \to 0$. Let $\epsilon > 0$ be given. Choose $\delta > 0$ sufficiently small, such that

$$\|u - u_\delta\|_{W^{s,p}(\Omega; \mathbb{R}^k)} \leq \epsilon/2.$$

We note that $C^{s+1}(\overline{\Omega}; \mathbb{R}^k) \hookrightarrow W^{s,p}(\Omega; \mathbb{R}^k)$ has a continuous embedding, and hence there exists a constant $C_0 > 0$, such that

$$\| \cdot \|_{W^{s,p}(\Omega; \mathbb{R}^k)} \leq C_0 \| \cdot \|_{C^{s+1}(\overline{\Omega}; \mathbb{R}^k)}.$$

Now note that there exists a neural network $\widetilde{u}$, such that $\|u_\delta - \widetilde{u}\|_{C^{s+1}(\overline{\Omega}; \mathbb{R}^k)} \leq \epsilon/2C_0$; this follows since $u_\delta \in C^{s+1}(\overline{\Omega}; \mathbb{R}^k)$, by Step 1. Combining these estimates, we obtain

$$\|u - \widetilde{u}\|_{W^{s,p}(\Omega; \mathbb{R}^k)} \leq \|u - u_\delta\|_{W^{s,p}(\Omega; \mathbb{R}^k)} + \|u_\delta - \widetilde{u}\|_{W^{s,p}(\Omega; \mathbb{R}^k)}$$
$$\leq \epsilon/2 + C_0 \|u_\delta - \widetilde{u}\|_{C^{s+1}(\overline{\Omega}; \mathbb{R}^k)}$$

$$\leq \epsilon.$$

This concludes our proof.   □

A.3 A Dense Subset Of Operators

One crucial ingredient in the proof of universal approximation for averaging neural operators is the fact that it is possible to reduce the problem for a general operator $\Psi^\dagger :$ $C^s(\overline{\Omega}; \mathbb{R}^k) \to C^{s'}(\overline{\Omega}; \mathbb{R}^{k'})$ (or $\Psi^\dagger : W^{s,p}(\Omega; \mathbb{R}^k) \to W^{s',p'}(\Omega; \mathbb{R}^{k'})$, respectively), to one for a simpler class of operators which can be written in the form,

$$\widetilde{\Psi}^\dagger(u) = \sum_{j=1}^{J} \alpha_j(u)\eta_j, \tag{A.2}$$

where $\eta_1, \ldots, \eta_J$ are functions in $C^{s'}(\overline{\Omega}; \mathbb{R}^{k'})$ (resp. in $W^{s',p'}(\Omega; \mathbb{R}^{k'})$), and $\alpha_1, \ldots, \alpha_J : L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}$ are continuous nonlinear functionals, defined on the space of integrable functions. This is the content of the following two propositions. The first version is formulated for operators between spaces of continuously differentiable functions and the second between Sobolev spaces.

Proposition A.6 *Let* $\Psi^\dagger : C^s(\overline{\Omega}; \mathbb{R}^k) \to C^{s'}(\overline{\Omega}; \mathbb{R}^{k'})$ *be a continuous operator. Let* $\mathsf{K} \subset C^s(\overline{\Omega}; \mathbb{R}^k)$ *be a compact subset. Then for any $\epsilon > 0$, there exist $J \in \mathbb{N}$, functions* $\eta_1, \ldots, \eta_J \in C^{s'}(\overline{\Omega}; \mathbb{R}^{k'})$, *and continuous functionals $\alpha_1, \ldots, \alpha_J : L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}$, such that the operator* $\widetilde{\Psi}^\dagger : L^1(\Omega; \mathbb{R}^k) \to C^{s'}(\overline{\Omega}; \mathbb{R}^{k'})$, $\widetilde{\Psi}^\dagger(u) := \sum_{j=1}^{J} \alpha_j(u)\eta_j$, *satisfies*

$$\sup_{u \in \mathsf{K}} \left\| \Psi^\dagger(u) - \widetilde{\Psi}^\dagger(u) \right\|_{C^{s'}} \leq \epsilon.$$

Remark A.7 We emphasize that even though the underlying operator $\Psi^\dagger(u)$ is only defined for $u \in C^s(\overline{\Omega}; \mathbb{R}^k)$, and will generally not possess any continuous extension to an operator defined for $u \in L^1(\Omega; \mathbb{R}^k)$, the above Proposition A.6 constructs a continuous operator $\widetilde{\Psi}^\dagger : L^1(\Omega; \mathbb{R}^k) \to C^{s'}(\overline{\Omega}; \mathbb{R}^{k'})$, whose restriction to compact $\mathsf{K} \subset C^s(\overline{\Omega}; \mathbb{R}^k)$ provides a good approximation, $\Psi^\dagger|_\mathsf{K} \approx \widetilde{\Psi}^\dagger|_\mathsf{K}$. In this context note that $L^1(\Omega; \mathbb{R}^k) \supset C^s(\overline{\Omega}; \mathbb{R}^k)$.

Proof Let us first point out that we may without loss of generality assume $k' = 1$, in the following; indeed we can identify $C^{s'}(\overline{\Omega}; \mathbb{R}^k) = [C^{s'}(\overline{\Omega}; \mathbb{R})]^k$, and thus it will suffice to approximate each component of the mapping $\Psi^\dagger : C^s(\overline{\Omega}; \mathbb{R}^k) \to [C^{s'}(\overline{\Omega}; \mathbb{R})]^k$, individually. The j-th component defines a mapping $\Psi_j^\dagger : C^s(\overline{\Omega}; \mathbb{R}^k) \to C^{s'}(\overline{\Omega}; \mathbb{R})$. Henceforth we use the notation $C^{s'}(\overline{\Omega}) := C^{s'}(\overline{\Omega}; \mathbb{R})$.

    Step 1: (construction of $\eta_1, \ldots, \eta_J$) Our first goal is to construct suitable $\eta_1, \ldots, \eta_J$. To this end, we consider $\mathsf{K}' := \Psi^\dagger(\mathsf{K}) \subset C^{s'}(\overline{\Omega})$. Note that since K is compact, and $\Psi^\dagger$ is continuous, its image K' is also compact. Let $B \supset \Omega$ be a

bounding box, containing $\overline{\Omega}$ in its interior. By Lemma A.1, there exists a continuous extension mapping

$$\mathcal{E} : C^{s'}(\overline{\Omega}) \to C^{s'}_{\text{per}}(B),$$

where $C^{s'}_{\text{per}}(B)$ denotes the space of continuously differentiable functions $w : B \to \mathbb{R}$ on the Cartesian domain B, possessing periodic derivatives up to order s'. Since $\mathsf{K}' \subset C^{s'}(\overline{\Omega})$ is compact, it follows that also $\mathsf{K}'_{\text{per}} := \mathcal{E}(\mathsf{K}')$ is a compact subset of $C^{s'}_{\text{per}}(B)$. Given this periodicity, we can identify $B \simeq \mathbb{T}^d$ with the periodic torus in a canonical way. Let $\eta_1, \eta_2, \dots$ denote an enumeration of the L^2-orthogonal (real) Fourier sine/cosine basis in $L^2_{\text{per}}(B)$. Fix $\delta > 0$, and let $w_\delta : B \to \mathbb{R}$ be the standard mollification of the periodic function $w : B \to \mathbb{R}$. Since $\mathsf{K}'_{\text{per}} \subset C^{s'}_{\text{per}}(B)$ is compact, the set $\left\{ w_\delta \mid w \in \mathsf{K}'_{\text{per}} \right\}$ is uniformly bounded in the $C^{r'}$-norm for any given $r' > s'$. In particular, it follows from classical results in harmonic analysis [23, Chap. 1, Sect. 3], that approximation by Fourier series converges uniformly over $\left\{ w_\delta \mid w \in \mathsf{K}'_{\text{per}} \right\}$, i.e.

$$\lim_{J \to \infty} \sup_{w \in \mathsf{K}'_{\text{per}}} \left\| w_\delta - \sum_{j=1}^{J} \langle w_\delta, \eta_j \rangle_{L^2} \eta_j \right\|_{C^{s'}_{\text{per}}(B)} = 0.$$

Furthermore, $\lim_{\delta \to 0} \| w_\delta - w \|_{C^{s'}} = 0$, uniformly over K'_{per}. In particular, given $\epsilon > 0$, we can first find $\delta = \delta(\epsilon) > 0$, such that $\sup_{w \in \mathsf{K}'_{\text{per}}} \| w - w_\delta \|_{C^{s'}} \le \epsilon/2$, and then $J \in \mathbb{N}$, such that $\sup_{w \in \mathsf{K}'_{\text{per}}} \left\| w_\delta - \sum_{j=1}^{J} \langle w_\delta, \eta_j \rangle_{L^2} \eta_j \right\|_{C^{s'}_{\text{per}}(B)} \le \epsilon/2$. It follows from the triangle inequality that

$$\sup_{w \in \mathsf{K}'_{\text{per}}} \left\| w - \sum_{j=1}^{J} \langle w_\delta, \eta_j \rangle_{L^2} \eta_j \right\|_{C^{s'}_{\text{per}}(B)} \le \epsilon.$$

This defines our choice of $\eta_1, \dots, \eta_J$. Note that we also have the identity

$$\sum_{j=1}^{J} \langle w_\delta, \eta_j \rangle_{L^2} = \sum_{j=1}^{J} \langle w, \eta_{j,\delta} \rangle_{L^2}.$$

We now recall that $\Omega \subset B$, and $\mathcal{E}$ is an extension operator, so that $\mathcal{E}(v)|_\Omega = v$ for all $v \in \mathsf{K}'$. Furthermore, we have $\mathsf{K}'_{\text{per}} = \mathcal{E}(\mathsf{K}')$ and $\mathsf{K}' = \Psi^\dagger(\mathsf{K})$, by definition. As a consequence, it follows that,

$$\sup_{u \in \mathsf{K}} \left\| \Psi^\dagger(u) - \sum_{j=1}^{J} \langle \mathcal{E}(\Psi^\dagger(u)), \eta_{j,\delta} \rangle_{L^2} \eta_j \right\|_{C^{s'}(\overline{\Omega})}$$

$$= \sup_{v \in \mathsf{K}'} \left\| v - \sum_{j=1}^{J} \langle \mathcal{E}(v), \eta_{j,\delta} \rangle_{L^2} \eta_j \right\|_{C^{s'}(\overline{\Omega})}$$

$$= \sup_{v \in \mathsf{K}'} \left\| \mathcal{E}(v) - \sum_{j=1}^{J} \langle \mathcal{E}(v), \eta_{j,\delta} \rangle_{L^2} \eta_j \right\|_{C^{s'}(\overline{\Omega})}$$

$$\leq \sup_{v \in \mathsf{K}'} \left\| \mathcal{E}(v) - \sum_{j=1}^{J} \langle \mathcal{E}(v), \eta_{j,\delta} \rangle_{L^2} \eta_j \right\|_{C^{s'}_{\mathrm{per}}(B)}$$

$$= \sup_{w \in \mathsf{K}'_{\mathrm{per}}} \left\| w - \sum_{j=1}^{J} \langle w, \eta_{j,\delta} \rangle_{L^2} \eta_j \right\|_{C^{s'}_{\mathrm{per}}(B)} \leq \epsilon.$$

Step 2: (construction of $\alpha_1, \ldots, \alpha_J$)

Given the results of Step 1, let us define a nonlinear functional $\beta_j : C^s(\overline{\Omega}; \mathbb{R}^k) \to \mathbb{R}$ by $\beta_j(u) := \langle \mathcal{E}(\Psi^\dagger(u)), \eta_{j,\delta} \rangle_{L^2}$. Then, by Step 1, we have

$$\sup_{u \in \mathsf{K}} \left\| \Psi^\dagger(u) - \sum_{j=1}^{J} \beta_j(u) \eta_j \right\|_{C^{s'}} \leq \epsilon. \tag{A3}$$

This is almost the claimed result, except that β_j does not define a continuous functional $L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}$. To remedy this, we rely on mollification adapted to the bounded Lipschitz domain Ω. Let $\delta > 0$ denote a mollification parameter. By Lemma A.2, there exists a continuous operator $\mathcal{M}_\delta : L^1(\Omega; \mathbb{R}^k) \to C^s(\overline{\Omega}; \mathbb{R}^k)$, such that over the compact set $\mathsf{K} \subset C^s(\overline{\Omega}; \mathbb{R}^k)$, we have

$$\sup_{u \in \mathsf{K}} \| u - \mathcal{M}_\delta u \|_{C^s} \to 0,$$

as $\delta \to 0$.

We intend to define $\alpha_j : L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}$ by $\alpha_j(u) := \beta_j(\mathcal{M}_\delta u)$ for suitably chosen $\delta > 0$. Recall that for fixed $\delta_0 > 0$, the set K_δ defined by

$$\mathsf{K}_\delta := \bigcup_{0 \leq \delta \leq \delta_0} \mathcal{M}_\delta(\mathsf{K}),$$

is a compact subset of $C^s(\overline{\Omega}; \mathbb{R}^k)$, by Lemma A.4. Note that for any $j = 1, \ldots, J$, we have a continuous mapping $\beta_j : \mathsf{K}_\delta \to \mathbb{R}$. Since K_δ is compact, it follows that there exists a continuous modulus of continuity $\omega : [0, \infty) \to [0, \infty)$, satisfying $\omega(0) = 0$, such that

$$|\beta_j(u) - \beta_j(u')| \leq \omega(\| u - u' \|_{C^s}), \quad \forall u, u' \in \mathsf{K}_\delta,$$

holds for all $j = 1, \ldots, J$. It follows that

$$|\beta_j(u) - \beta_j(\mathcal{M}_\delta u)| \leq \omega(\| u - \mathcal{M}_\delta u \|_{C^s}),$$

for any $0 < \delta \leq \delta_0$. By Lemma A.2, $\mathcal{M}_\delta u \to u$ converges uniformly over $u \in \mathsf{K}$, so that we may conclude

$$\lim_{\delta \to 0} \sup_{u \in \mathsf{K}} |\beta_j(u) - \beta_j(\mathcal{M}_\delta u)| \leq \lim_{\delta \to 0} \omega \left(\sup_{u \in \mathsf{K}} \| u - \mathcal{M}_\delta u \|_{C^s} \right) = 0.$$

In particular, we can choose $\delta > 0$ sufficiently small, to ensure that $\alpha_j(u) := \beta_j(\mathcal{M}_\delta u)$ satisfies

$$\sup_{u \in \mathsf{K}} |\beta_j(u) - \alpha_j(u)| \leq \frac{\epsilon}{J \max_{j=1,\ldots,J} \|\eta_j\|_{C^{s'}}}, \tag{A.4}$$

for all $j = 1, \ldots, J$. Since $\delta > 0$, we also note that $\mathcal{M}_\delta : L^1(\Omega; \mathbb{R}^k) \to C^s(\overline{\Omega}; \mathbb{R}^k)$ is a continuous mapping, and hence $\alpha_j = \beta_j \circ \mathcal{M}_\delta$ is continuous as a mapping $\alpha_j : L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}$.

Step 3: (Conclusion) Combining (A3) with (A.4), we obtain

$$\sup_{u \in \mathsf{K}} \left\| \Psi^\dagger(u) - \sum_{j=1}^J \alpha_j(u)\eta_j \right\|_{C^{s'}}$$

$$\leq \sup_{u \in \mathsf{K}} \left\| \Psi^\dagger(u) - \sum_{j=1}^J \beta_j(u)\eta_j \right\|_{C^{s'}}$$

$$+ \sup_{u \in \mathsf{K}} \left\| \sum_{j=1}^J [\beta_j(u) - \alpha_j(u)]\eta_j \right\|_{C^{s'}}$$

$$\leq \sup_{u \in \mathsf{K}} \left\| \Psi^\dagger(u) - \sum_{j=1}^J \beta_j(u)\eta_j \right\|_{C^{s'}}$$

$$+ J \max_{j=1,\ldots,J} \|\eta_j\|_{C^{s'}} \max_{j=1,\ldots,J} \sup_{u \in \mathsf{K}} |\beta_j(u) - \alpha_j(u)|$$

$$\leq 2\epsilon.$$

As $\epsilon > 0$ was arbitrary, the claim of Proposition A.6 now follows. □

We can also formulate a similar result for operators between Sobolev spaces. As the proof is almost identical to the proof of Proposition A.6, and essentially follows by replacing C^s by $W^{s,p}$ and analogously in the output space, we forego the details of this argument, and only state the final result.

Proposition A.8 *Let* $\Psi^\dagger : W^{s,p}(\Omega; \mathbb{R}^k) \to W^{s',p'}(\Omega; \mathbb{R}^{k'})$ *be a continuous operator. Let* $\mathsf{K} \subset W^{s,p}(\Omega; \mathbb{R}^k)$ *be a compact subset. Then for any* $\epsilon > 0$, *there exist* $\eta_1, \ldots, \eta_J \in W^{s',p'}(\Omega; \mathbb{R}^{k'})$, *and continuous functionals* $\alpha_1, \ldots, \alpha_N : L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}$, *such that*

$$\sup_{u \in \mathsf{K}} \left\| \Psi^\dagger(u) - \sum_{j=1}^J \alpha_j(u)\eta_j \right\|_{W^{s',p'}} \leq \epsilon.$$

A.4 Approximation Of Nonlinear Functionals By The Averaging Neural Operator

Given the result of Proposition A.6 and A.8 in the last section, a core ingredient in our proof of universal approximation for averaging neural operators will be the approximation of nonlinear functionals $\alpha : L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}$ by averaging neural operators. In the present section, we will show that any nonlinear a functional $\alpha : L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}$, $u \mapsto \alpha(u)$, can be approximated by an averaging neural operator in a suitable sense. This is the subject of the following lemma:

Lemma A.9 *Let $\alpha : L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}$ be a continuous nonlinear functional. Let $\mathsf{K} \subset L^1(\Omega; \mathbb{R}^k)$ be a compact set, consisting of bounded functions, $\sup_{u \in \mathsf{K}} \|u\|_{L^\infty} < \infty$. Then for any $\epsilon > 0$, there exists an averaging neural operator $\widetilde{\alpha} : L^1(\Omega; \mathbb{R}^k) \to L^1(\Omega)$, all of whose output functions are constant so that we may also view $\widetilde{\alpha}$ as a function $\widetilde{\alpha} : L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}$, such that*

$$\sup_{u \in \mathsf{K}} |\alpha(u) - \widetilde{\alpha}(u)| \le \epsilon.$$

Remark A.10 We note that the construction in our proof of Lemma A.9 (cp. (A.12) below) actually defines an averaging neural operator with (1) a lifting layer represented by $R(u(x), x)$, (2) a *single* hidden layer $v(x) \mapsto \sigma\left(\fint_\Omega v(y)\, dy\right)$, and (3) the projection layer $Q(v, x) := q_1(v)$, where q_1 is a neural network depending only on v. Hence, the result of Lemma A.9, and as a consequence the universal approximation property as described in Theorems 1 and 2, can be achieved with a *single* evaluation of the averaging operation: $L = 1$.

Proof (Lemma A.9) Let $\alpha : L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}$ be a continuous functional. Let $\mathsf{K} \subset L^1(\Omega; \mathbb{R}^k)$ be a compact set. Fix $\epsilon > 0$. Our aim is to show that there exists an averaging neural operator $\widetilde{\alpha} : L^1(\Omega; \mathbb{R}^k) \to L^1(\Omega)$ with constant output, such that

$$\sup_{u \in \mathsf{K}} |\alpha(u) - \widetilde{\alpha}(u)| \le \epsilon.$$

To prove this, note that we can identify any $u \in L^1(\Omega; \mathbb{R}^k)$ with a function in $L^1(B; \mathbb{R}^k)$, via an extension of $u(x) := 0$ for $x \in B \setminus \Omega$. Using this identification, the compact subset $\mathsf{K} \subset L^1(\Omega; \mathbb{R}^k)$ can be identified with a compact subset of $L^1(\mathbb{R}^d; \mathbb{R}^k)$. Fix a smooth mollifier $\rho \in C^\infty$, and denote $\rho_\delta(x) := \delta^{-d} \rho(x/\delta)$ for $\delta > 0$. We denote by $u_\delta(x) = (u * \rho_\delta)(x)$ the mollification of u (extended to all of $\mathbb{R}^d$ by 0 outside of Ω). Since $\mathsf{K} \subset L^1(\Omega; \mathbb{R}^k)$ is compact, it follows that

$$\lim_{\delta \to 0} \sup_{u \in \mathsf{K}} \|u - u_\delta\|_{L^1(\Omega; \mathbb{R}^k)} = 0.$$

Since $\alpha : L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}$ is continuous, it follows that the mapping $\alpha_\delta : L^1(B, \mathbb{R}^k) \to \mathbb{R}$ defined by $\alpha_\delta(u) := \alpha(u_\delta)$, for $\delta > 0$, converges uniformly over K, as $\delta \to 0$:

$$\lim_{\delta \to 0} \sup_{u \in \mathsf{K}} |\alpha(u) - \alpha_\delta(u)| \le \lim_{\delta \to 0} \sup_{u \in \mathsf{K}} |\alpha(u) - \alpha(u_\delta)| = 0. \tag{A.5}$$

The idea is now that, for any choice of orthonormal basis $\xi_1, \xi_2, \ldots$ of $L^2(\Omega; \mathbb{R}^k)$, which we may additionally choose to be smooth, $\xi_j \in C^\infty(\Omega; \mathbb{R}^k)$, we have uniform convergence,

$$\sup_{u \in \mathsf{K}} \left\| u_\delta - \sum_{j=1}^J \langle u_\delta, \xi_j \rangle_{L^2} \xi_j \right\|_{L^1(B; \mathbb{R}^k)} \to 0, \tag{A.6}$$

as $J \to \infty$. We note that the expression in (A.6) is well-defined, since $\mathsf{K} \subset L^1 \cap L^\infty \subset L^2$, by assumption. Furthermore, $\mathsf{K} \subset L^2$ is compact in the L^2-norm; this entails the uniform convergence (A.6). With the convention that u and ξ_j are expanded by 0 outside of Ω, we have

$$
\begin{aligned}
\langle u_\delta, \xi_j \rangle_{L^2} &= \int_\Omega u_\delta(y) \cdot \xi_j(y)\,dy = \int_{\mathbb{R}^d} (u * \rho_\delta)(y) \cdot \xi_j(y)\,dy \\
&= \int_{\mathbb{R}^d} u(y) \cdot (\xi_j * \rho_\delta)(y)\,dy = \fint_\Omega u(y) \cdot \xi_{j,\delta}(y)\,dy,
\end{aligned}
\tag{A.7}
$$

where we have defined $\xi_{j,\delta}(y) := |\Omega|(\xi_j * \rho_\delta)(y)$. Hence, if we define $\alpha_{\delta,J} : L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}$ by

$$
\alpha_{\delta,J}(u) := \alpha \left(\sum_{j=1}^J \left(\fint_\Omega u(y) \cdot \xi_{j,\delta}(y)\,dy \right) \xi_j \right),
$$

then it follows from (A.5), (A.6) and (A.7), that for $\delta = \delta(\mathsf{K}) > 0$ sufficiently small, and $J = J(\delta, \mathsf{K}) \in \mathbb{N}$ sufficiently large, we have $\sup_{u \in \mathsf{K}} |\alpha_\delta(u) - \alpha_{\delta,J}(u)| \leq \epsilon/2$. To indicate the connection to averaging neural operators, we note that we can write $\alpha_{\delta,J}$ as the following composition:

$$
u \mapsto \left(\fint_\Omega u(y) \cdot \xi_{1,\delta}(y)\,dy, \ldots, \fint_\Omega u(y) \cdot \xi_{J,\delta}(y)\,dy \right)
$$

$$
\mapsto \alpha \left(\sum_{j=1}^J \left(\fint_\Omega u(y) \cdot \xi_{j,\delta}(y)\,dy \right) \xi_j \right).
$$

The first of these mapping requires computation of certain averages (albeit against a function $\xi_{j,\delta}$). Approximation of this mapping clearly requires nonlocality. The second mapping defines a continuous function $\beta : \mathbb{R}^J \to \mathbb{R}$, $c = (c_1, \ldots, c_J) \mapsto \alpha \left(\sum_{j=1}^J c_j \xi_j \right)$. Approximation of this mapping is possible by ordinary neural networks; this requires nonlinearity, but does not require nonlocality.

Choose $M > 0$, such that the image of the compact set $\mathsf{K} \subset L^1(\Omega; \mathbb{R}^k)$ under the mapping

$$
L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}^J, \quad u \mapsto \left(\fint_\Omega u(y) \cdot \xi_{1,\delta}(y)\,dy, \ldots, \fint_\Omega u(y) \cdot \xi_{J,\delta}(y)\,dy \right),
$$

is contained in $[-M, M]^J$. Note that $c \mapsto \beta(c) = \alpha \left(\sum_{j=1}^J c_j \xi_j \right)$ is a continuous function, by the continuity of α. By universality of conventional neural networks, there exists a neural network $\widetilde{\beta} : \mathbb{R}^J \to \mathbb{R}$, such that

$$
\sup_{c \in [-M,M]^N} |\widetilde{\beta}(c) - \beta(c)| \leq \epsilon.
\tag{A.8}
$$

Write $\widetilde{\beta} : \mathbb{R}^J \to \mathbb{R}$ as a composition of its hidden layers:

$$\begin{cases} v_0 := c, \\ v_\ell := \sigma\left(\widetilde{A}_\ell v_{\ell-1} + \widetilde{b}_\ell\right), \quad \ell = 1, \dots, L, \\ \widetilde{\beta}(c) := \widetilde{A}_{L+1} v_L + \widetilde{b}_{L+1}, \end{cases} \tag{A.9}$$

where $\widetilde{A}_\ell, \widetilde{b}_\ell$ are the weights and biases of the hidden layers. Parallelizing the neural networks constructed in Lemma A.11, below, it follows that for any $\epsilon' > 0$, there exists a neural network $\widetilde{R} : \mathbb{R}^k \times \Omega \to \mathbb{R}^J$, $(v, x) \mapsto \widetilde{R}(v, x) = (\widetilde{R}_1(v, x), \dots, \widetilde{R}_J(v, x))$, such that

$$\sup_{u \in \mathsf{K}} \left| \fint_\Omega \widetilde{R}_j(u(y), y)\, dy - \fint_\Omega u(y) \cdot \xi_{j,\delta}(y)\, dy \right| \le \epsilon',$$

for $j = 1, \dots, J$. Composing the output layer with an affine mapping, we can construct another neural network R, such that

$$R(u(x), x) = \widetilde{A}_1 \widetilde{R}(u(x), x) + \widetilde{b}_1,$$

where $\widetilde{A}_1, \widetilde{b}_1$ are the weights and biases of the input layer of $\widetilde{\beta}$ defined by (A.9). In particular, it follows that for any input $u \in \mathsf{K}$, and defining coefficients $c(u) := (c_1, \dots, c_J)$ by $c_j := \fint_\Omega u(y) \cdot \xi_{j,\delta}(y)\, dy$, we have

$$\sup_{u \in \mathsf{K}} \left| \sigma\left(\fint_\Omega R(u(y), y)\, dy \right) - \sigma\left(\widetilde{A}_1 c(u) + \widetilde{b}_1 \right) \right| \le \|\sigma\|_{\mathrm{Lip}} \|\widetilde{A}_1\| \epsilon', \tag{A.10}$$

where $\|\widetilde{A}_1\|$ denotes the operator norm of $\widetilde{A}_1$. Note that the second term in (A.10) is exactly the output of the first hidden layer of $\widetilde{\beta}$, cp. (A.9). Let us decompose $\widetilde{\beta}(c) = \widetilde{\beta}_1 \circ \sigma(\widetilde{A}_1 c + \widetilde{b}_1)$, where $\widetilde{\beta}_1$ denotes the composition of the other hidden layers, $\ell = 2, \dots, L$, and the output layer. Composing each of the terms appearing on the left-hand side of (A.10) with $\widetilde{\beta}_1$ and choosing ϵ' sufficiently small (depending on $\widetilde{\beta}_1$, $\widetilde{A}_1$), we can ensure that

$$\sup_{u \in \mathsf{K}} \left| \widetilde{\beta}_1\left(\sigma\left(\fint_\Omega R(u(y), y)\, dy \right) \right) - \widetilde{\beta}_1\left(\sigma\left(\widetilde{A}_1 c(u) + \widetilde{b}_1 \right) \right) \right| \le \epsilon,$$

or equivalently,

$$\sup_{u \in \mathsf{K}} \left| \widetilde{\beta}_1\left(\sigma\left(\fint_\Omega R(u(y), y)\, dy \right) \right) - \widetilde{\beta}(c(u)) \right| \le \epsilon. \tag{A.11}$$

From the above, (A.8) an (A.11), it follows that the averaging neural operator $\widetilde{\alpha}$, defined by the following composition,

$$\widetilde{\alpha} : u \overset{(1)}{\mapsto} R(u(\cdot), \cdot) \overset{(2)}{\mapsto} \sigma\left(\fint_\Omega R(u(y), y)\,dy\right) \overset{(3)}{\mapsto} \widetilde{\beta}_1 \circ \sigma\left(\fint_\Omega R(u(y), y)\,dy\right),$$

(A.12)

satisfies

$$
\begin{aligned}
\sup_{u \in \mathsf{K}} |\alpha(u) - \widetilde{\alpha}(u)| &= \sup_{u \in \mathsf{K}} |\beta(c(u)) - \widetilde{\alpha}(u)| \\
&\leq \sup_{u \in \mathsf{K}} |\beta(c(u)) - \widetilde{\beta}(c(u))| \\
&\quad + \sup_{u \in \mathsf{K}} \left| \widetilde{\beta}(c(u)) - \widetilde{\beta}_1 \circ \sigma\left(\fint_\Omega R(u(y), y)\,dy\right) \right| \\
&\leq \sup_{c \in [-M,M]^N} |\beta(c) - \widetilde{\beta}(c)| \\
&\quad + \sup_{u \in \mathsf{K}} \left| \widetilde{\beta}(c(u)) - \widetilde{\beta}_1 \circ \sigma\left(\fint_\Omega R(u(y), y)\,dy\right) \right| \\
&\leq \epsilon.
\end{aligned}
$$

$\square$

We finally state the following lemma, which was used in the preceding proof of Lemma A.9:

Lemma A.11 *Let $\mathsf{K} \subset L^1(\Omega; \mathbb{R}^k)$ be compact, consisting of uniformly bounded functions $\sup_{u \in \mathsf{K}} \|u\|_{L^\infty} < \infty$. Let $\xi \in C^\infty(\overline{\Omega}; \mathbb{R}^k)$ be given and fixed. Then for any $\epsilon > 0$, there exists a neural network $\widetilde{R} : \mathbb{R}^k \times \Omega \to \mathbb{R}^k$, such that*

$$\sup_{u \in \mathsf{K}} \left| \fint_\Omega \widetilde{R}(u(y), y)\,dy - \fint_\Omega u(y) \cdot \xi(y)\,dy \right| \leq \epsilon.$$

Proof Fix $\epsilon > 0$. In the following, we denote by

$$M_\mathsf{K} := \sup_{u \in \mathsf{K}} \|u\|_{L^\infty},$$

the upper L^∞-bound on elements $u \in \mathsf{K}$. By assumption, M_K is finite. As $\xi \in C^\infty(\overline{\Omega}; \mathbb{R}^k)$ is fixed, and $\Omega \subset \mathbb{R}^d$ is bounded, there exists a neural network $\widetilde{\xi} : \Omega \to \mathbb{R}^k$, such that

$$\sup_{x \in \Omega} |\xi(x) - \widetilde{\xi}(x)|_{\ell^\infty} \leq \epsilon/(2kM_\mathsf{K}),$$

where k is the number of components of ξ. Furthermore define

$$M_\xi := \max\left\{ \|\xi\|_{L^\infty}, \|\widetilde{\xi}\|_{L^\infty} \right\} < \infty.$$

By the universality of ordinary neural networks, there exists a neural network $\widetilde{\times}$: $[-M_{\mathsf{K}}, M_{\mathsf{K}}]^k \times [-M_\xi, M_\xi]^k \to \mathbb{R}$, such that

$$\sup_{|v|_{\ell^\infty} \leq M_{\mathsf{K}}, |w|_{\ell^\infty} \leq M_\xi} |v \cdot w - \widetilde{\times}(v, w)| \leq \epsilon/2.$$

Defining a new neural network as the composition $\widetilde{R}(v, x) := \widetilde{\times}(v, \widetilde{\xi}(x))$, it now follows that for any $u \in \mathsf{K}$:

$$
\begin{aligned}
\sup_{x \in \Omega} |\widetilde{R}(u(x), x) - u(x) \cdot \xi(x)| &= \sup_{x \in \Omega} |\widetilde{\times}(u(x), \widetilde{\xi}(x)) - u(x) \cdot \xi(x)| \\
&\leq \sup_{x \in \Omega} |\widetilde{\times}(u(x), \widetilde{\xi}(x)) - u(x) \cdot \widetilde{\xi}(x)| \\
&\quad + \sup_{x \in \Omega} |u(x) \cdot \widetilde{\xi}(x) - u(x) \cdot \xi(x)| \\
&\leq \sup_{|v|_{\ell^\infty} \leq M_{\mathsf{K}}, |w|_{\ell^\infty} \leq M_\xi} |\widetilde{\times}(v, w) - v \cdot w| \\
&\quad + kM \sup_{x \in \Omega} |\widetilde{\xi}(x) - \xi(x)|_{\ell^\infty} \\
&\leq \epsilon.
\end{aligned}
$$

In particular, this upper bound implies that

$$\sup_{u \in \mathsf{K}} \left| \fint_\Omega \widetilde{R}(u(y), y)\, dy - \fint_\Omega u(y) \cdot \xi(y)\, dy \right| \leq \epsilon.$$

$\square$

5.1 Proof Of Universal Approximation $C^s \to C^{s'}$, Theorem 1

Given the results of Sections 1 and 1, we can now provide a detailed proof of the universal approximation Theorem 1. The main idea behind the proof is that using Proposition A.6 in Section 1, it suffices to approximate operators of the form

$$\widetilde{\Psi}^\dagger(u) = \sum_{j=1}^J \alpha_j(u)\eta_j,$$

where $\alpha_j : L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}$ are functionals, and $\eta_j \in C^{s'}(\overline{\Omega}; \mathbb{R}^{k'})$ are fixed functions. Given averaging neural operators $\Psi_1, \ldots, \Psi_J$, we can easily combine them to obtain a new averaging neural operator Ψ, such that $\Psi(u) = \sum_{j=1}^J \Psi_j(u)$ for all input functions u. Thus, it suffices to consider only one term in the above sum, i.e. it will suffice to prove that any mapping of the form

$$\widetilde{\Psi}_j^\dagger : L^1(\Omega; \mathbb{R}^k) \to C^{s'}(\overline{\Omega}; \mathbb{R}^{k'}), \quad u \mapsto \alpha_j(u)\eta_j,$$

can be approximated by an averaging neural operator. The main difficulty in proving this is to show that any functional $\alpha_j : L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}$ can be approximated by averaging neural operators. This is the content of Lemma A.9 in the previous Section 1. Given this core ingredient, we can now provide a complete the proof of the universal approximation Theorem 1, below:

Proof (Theorem 1) Let $\Psi^\dagger : C^s(\overline{\Omega}; \mathbb{R}^k) \to C^{s'}(\overline{\Omega}; \mathbb{R}^{k'})$ be a continuous operator. Let $\mathsf{K} \subset C^s(\overline{\Omega}; \mathbb{R}^k)$ be compact. We aim to show that for any $\epsilon > 0$, there exists an averaging neural operator Ψ of the form $\Psi = \mathcal{Q} \circ \mathcal{L}_L \circ \cdots \circ \mathcal{L}_1 \circ \mathcal{R}$, such that

$$\sup_{u \in \mathsf{K}} \|\Psi^\dagger(u) - \Psi(u)\|_{C^{s'}} \leq \epsilon.$$

Fix $\epsilon > 0$. By Proposition A.6, there exist functions $\eta_1, \ldots, \eta_J \in C^{s'}(\overline{\Omega}; \mathbb{R}^{k'})$, and continuous functionals $\alpha_1, \ldots, \alpha_J : L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}$, such that

$$\sup_{u \in \mathsf{K}} \left\| \Psi^\dagger(u) - \sum_{j=1}^J \alpha_j(u)\eta_j \right\|_{C^{s'}} \leq \epsilon/2.$$

We now make the following claim:

Claim A.12 *Let $\mathsf{K} \subset L^1(\Omega; \mathbb{R}^k)$ be a compact set, consisting of bounded functions $\sup_{u \in \mathsf{K}} \|u\|_{L^\infty} < \infty$. Let $\eta_1, \ldots, \eta_J \in C^{s'}(\overline{\Omega}; \mathbb{R}^k)$ be functions and let $\alpha_1, \ldots, \alpha_J : L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}$ be continuous nonlinear functionals. Then for any $j = 1, \ldots, J$, there exists an averaging neural operator $\Psi_j : L^1(\Omega; \mathbb{R}^k) \to C^{s'}(\overline{\Omega}; \mathbb{R}^k)$, such that*

$$\sup_{u \in \mathsf{K}} \|\alpha_j(u)\eta_j - \Psi_j(u)\|_{C^{s'}} \leq \epsilon/2J.$$

Relying on the above claim, it is easy to see that there exists an averaging neural operator, such that $\Psi(u) = \sum_{j=1}^J \Psi_j(u)$ for all input functions u. This operator satisfies, for any $u \in \mathsf{K}$:

$$\|\Psi^\dagger(u) - \Psi(u)\|_{C^{s'}} \leq \left\| \Psi^\dagger(u) - \sum_{j=1}^J \alpha_j(u)\eta_j \right\|_{C^{s'}} + \left\| \sum_{j=1}^J \alpha_j(u)\eta_j - \Psi(u) \right\|_{C^{s'}}$$

$$\leq \left\| \Psi^\dagger(u) - \sum_{j=1}^J \alpha_j(u)\eta_j \right\|_{C^{s'}} + \sum_{j=1}^J \left\| [\alpha_j(u)\eta_j - \Psi_j(u)] \right\|_{C^{s'}}$$

$$\leq \epsilon/2 + J\epsilon/2J = \epsilon,$$

where the last estimate holds for any $u \in \mathsf{K}$. Thus, taking the supremum over K, the above claim implies that

$$\sup_{u \in \mathsf{K}} \|\Psi^\dagger(u) - \Psi(u)\|_{C^{s'}} \leq \epsilon,$$

which gives the universal approximation property of Theorem 1. To finish our argument, it thus remains to prove the above Claim A.12.

To prove the claim, fix $j \in \{1, \ldots, J\}$ for all the following. We first define

$$M_\eta := \|\eta_j\|_{C^{s'}(\Omega; \mathbb{R}^{k'})}. \tag{A.13}$$

Next, we observe that by Lemma A.9, there exists an averaging neural operator $\widetilde{\alpha}_j : L^1(\Omega; \mathbb{R}^k) \to L^1(\Omega)$, with constant output functions, such that

$$\sup_{u \in \mathsf{K}} |\alpha_j(u) - \widetilde{\alpha}_j(u)| \leq \epsilon/(6JM_\eta). \tag{A.14}$$

Choose $M_\alpha > 0$, such that

$$\sup_{u \in \mathsf{K}} |\alpha_j(u)|, \ \sup_{u \in \mathsf{K}} |\widetilde{\alpha}_j(u)| \leq M_\alpha. \tag{A.15}$$

Since $\eta_j \in C^{s'}(\overline{\Omega}; \mathbb{R}^{k'})$, there exists an ordinary neural network $\widetilde{\eta}_j : \Omega \to \mathbb{R}^{k'}$, such that (cp. e.g. the universal approximation result of Lemma A.5):

$$\|\eta_j - \widetilde{\eta}_j\|_{C^{s'}} \leq \epsilon/(6JM_\alpha). \tag{A.16}$$

Let us also define

$$\widetilde{M}_\eta := \max\left\{M_\eta, \|\widetilde{\eta}_j\|_{C^{s'}}\right\}. \tag{A.17}$$

Fix a small parameter $\delta > 0$, to be determined below. Since scalar multiplication $\times : [-M_\alpha, M_\alpha] \times [-\widetilde{M}_\eta, \widetilde{M}_\eta]^k \to \mathbb{R}^k$, $(a, v) \mapsto \times(a, v) := av$ defines a smooth mapping, there similarly exists a neural network $\widetilde{\times} : [-M_\alpha, M_\alpha] \times [-\widetilde{M}_\eta, \widetilde{M}_\eta]^k \to \mathbb{R}^k$, $(a, v) \mapsto \times(a, v)$, such that

$$\| \times(\cdot, \cdot) - \widetilde{\times}(\cdot, \cdot)\|_{C^{s'}([-M_\alpha, M_\alpha] \times [-\widetilde{M}_\eta, \widetilde{M}_\eta]^{k'}; \mathbb{R}^{k'})} \leq \delta. \tag{A.18}$$

We also recall that a composition of $C^{s'}$-functions is itself $C^{s'}$, and that there exists a constant $C_0 = C_0(s', k') > 0$, depending only on s' and k', such that

$$\sup_{|a| \leq M_\alpha} \| \times(a, \widetilde{\eta}_j(\cdot)) - \widetilde{\times}(a, \widetilde{\eta}_j(\cdot))\|_{C^{s'}(\Omega; \mathbb{R}^{k'})}$$
$$\leq \| \times(\cdot, \widetilde{\eta}_j(\cdot)) - \widetilde{\times}(\cdot, \widetilde{\eta}_j(\cdot))\|_{C^{s'}([-M_\alpha, M_\alpha] \times \Omega; \mathbb{R}^{k'})}$$
$$\leq C_0 \| \times(\cdot, \widetilde{\eta}_j(\cdot)) - \widetilde{\times}(\cdot, \cdot)\|_{C^{s'}([-M_\alpha, M_\alpha] \times [-\widetilde{M}_\eta, \widetilde{M}_\eta]^{k'}; \mathbb{R}^{k'})} \|\widetilde{\eta}_j\|_{C^{s'}(\Omega; \mathbb{R}^{k'})}$$
$$\leq C_0 \delta \widetilde{M}_\eta.$$

Thus, for any $|a|, |\widetilde{a}| \leq M_\alpha$, we obtain

$$\| \times(a, \eta_j(\cdot)) - \widetilde{\times}(\widetilde{a}, \widetilde{\eta}_j(\cdot))\|_{C^{s'}(\Omega; \mathbb{R}^{k'})}$$
$$\leq \| \times(a, \eta_j(\cdot)) - \times(\widetilde{a}, \eta_j(\cdot))\|_{C^{s'}(\Omega; \mathbb{R}^{k'})}$$

$$+ \| \times (\widetilde{a}, \eta_j(\cdot)) - \times(\widetilde{a}, \widetilde{\eta}_j(\cdot)) \|_{C^{s'}(\Omega;\mathbb{R}^{k'})}$$

$$+ \| \times (\widetilde{a}, \widetilde{\eta}_j(\cdot)) - \widetilde{\times}(\widetilde{a}, \widetilde{\eta}_j(\cdot)) \|_{C^{s'}(\Omega;\mathbb{R}^{k'})}$$

$$\leq |a - \widetilde{a}| M_\eta + M_\alpha \| \eta_j - \widetilde{\eta}_j \|_{C^{s'}(\Omega;\mathbb{R}^{k'})} + C_0 \delta \widetilde{M}_\eta.$$

Recalling (A.14) and (A.16) to bound the first two terms, and choosing $\delta :=$ $\epsilon/(6NC_0\widetilde{M}_\eta)$ in (A.18), it follows that

$$\| \times (a, \eta_j(\cdot)) - \widetilde{\times}(\widetilde{a}, \widetilde{\eta}_j(\cdot)) \|_{C^{s'}(\Omega;\mathbb{R}^{k'})} \leq \epsilon/6J + \epsilon/6J + \epsilon/6J,$$

for any $|a|, |\widetilde{a}| \leq M_\alpha$. By definition of M_α, (A.15), given arbitrary $u \in \mathsf{K}$, we have $|\alpha_j(u)| \leq M_\alpha$ and $|\widetilde{\alpha}_j(u)| \leq M_\alpha$. Hence, the above estimate finally implies that

$$\sup_{u \in \mathsf{K}} \| \alpha_j(u)\eta_j - \widetilde{\times}(\widetilde{\alpha}_j(u), \widetilde{\eta}_j(\cdot)) \|_{C^{s'}(\Omega;\mathbb{R}^{k'})} \leq \epsilon/2J. \qquad (A.19)$$

Note that the mapping $\mathbb{R} \times \Omega \to \mathbb{R}^{k'}$, $(a, x) \mapsto \widetilde{\times}(a, \widetilde{\eta}_j(x))$ is an ordinary neural network in (a, x). Since $\widetilde{\alpha}_j$ is an averaging neural operator by construction, we can write it in the form $\widetilde{\mathcal{Q}} \circ \mathcal{L}_L \circ \cdots \circ \mathcal{L}_1 \circ \mathcal{R}$, in terms of a raising operator $\mathcal{R}$, hidden layers $\mathcal{L}_\ell$, and a projection layer $\widetilde{\mathcal{Q}}$, where the values $\widetilde{\mathcal{Q}}(v)(x) := \widetilde{Q}(v(x), x) \in \mathbb{R}$ are given in terms of an ordinary neural network $\widetilde{Q}$. Let $\mathcal{Q}$ denote the composition $\mathcal{Q}(v)(x) := \widetilde{\times}(\widetilde{Q}(v(x), x), \widetilde{\eta}_j(x))$. Then

$$\Psi(u) := \mathcal{Q} \circ \mathcal{L}_L \circ \cdots \circ \mathcal{L}_1 \circ \mathcal{R}(u),$$

defines an averaging neural operator, for which

$$\Psi(u)(x) = \widetilde{\times}(\widetilde{\alpha}_j(u), \widetilde{\eta}_j(x)).$$

By (A.19), it follows that

$$\| \alpha_j(u)\eta_j - \Psi(u) \|_{C^{s'}} \leq \epsilon/2J.$$

This concludes our proof of the claim. $\qquad \square$

A.6 Proof Of Universal Approximation $W^{s,p} \to W^{s',p'}$, Theorem 2

The previous section provides the detailed proof of the universal approximation theorem 1, in the setting where the underlying function spaces $\mathcal{X}$ and $\mathcal{Y}$ consist of continuously differentiable functions. Theorem 2 states a corresponding universality result in the scale of Sobolev spaces. The proof of Theorem 2 is almost identical to the proof of Theorem 1. In the present section, we provide the necessary alterations to the proof.

Proof (Theorem 2) Let $\Psi^\dagger : W^{s,p}(\Omega;\mathbb{R}^k) \to W^{s',p'}(\Omega;\mathbb{R}^{k'})$ be a continuous operator with integer $s, s' \geq 0$ and $p, p' \in [1, \infty)$. Let $\mathsf{K} \subset W^{s,p}(\Omega;\mathbb{R}^k)$ be compact,

consisting of bounded functions, $\sup_{u \in \mathsf{K}} \|u\|_{L^\infty} < \infty$. We aim to show that for any $\epsilon > 0$, there exists an averaging neural operator Ψ of the form $\Psi = \mathcal{Q} \circ \mathcal{L}_L \circ \cdots \circ \mathcal{L}_1 \circ \mathcal{R}$, such that

$$\sup_{u \in \mathsf{K}} \|\Psi^\dagger(u) - \Psi(u)\|_{W^{s',p'}} \le \epsilon.$$

Fix $\epsilon > 0$. By Proposition A.8, there exist functions $\eta_1, \ldots, \eta_J \in W^{s',p'}(\Omega; \mathbb{R}^{k'})$, and continuous functionals $\alpha_1, \ldots, \alpha_J : L^1(\Omega; \mathbb{R}^k) \to \mathbb{R}$, such that

$$\sup_{u \in \mathsf{K}} \left\| \Psi^\dagger(u) - \sum_{j=1}^J \alpha_j(u)\eta_j \right\|_{W^{s',p'}} \le \epsilon/3.$$

Approximating each η_j by its boundary-adapted mollification, $\eta_{j,\delta} := \mathcal{M}_\delta \eta_j \in C^\infty(\overline{\Omega}; \mathbb{R}^{k'})$ with $\delta > 0$ chosen sufficiently small (cp. Lemma A.2), we can ensure that

$$\sup_{u \in \mathsf{K}} \left\| \Psi^\dagger(u) - \sum_{j=1}^J \alpha_j(u)\eta_{j,\delta} \right\|_{W^{s',p'}} \le 2\epsilon/3.$$

We next note that $C^{s'+1}(\overline{\Omega}; \mathbb{R}^k) \hookrightarrow W^{s',p'}(\Omega; \mathbb{R}^{k'})$ has a continuous embedding, hence there exists a constant $C_0 > 0$, such that

$$\|v\|_{W^{s',p'}} \le C_0 \|v\|_{C^{s'+1}}, \quad \forall \, v \in C^{s'+1}(\overline{\Omega}; \mathbb{R}^{k'}). \tag{A.20}$$

We note that $u \mapsto \sum_{j=1}^J \alpha_j(u)\eta_{j,\delta}$ defines a continuous operator $L^1(\Omega; \mathbb{R}^k) \to C^{s'+1}(\overline{\Omega}; \mathbb{R}^{k'})$ and recall that, by assumption, $\mathsf{K} \subset W^{s,p}(\Omega; \mathbb{R}^k) \subset L^1(\Omega; \mathbb{R}^k)$ is a compact set consisting of bounded functions, $\sup_{u \in \mathsf{K}} \|u\|_{L^\infty} < \infty$. It thus follows from Claim A.12 that there exists an averaging neural operator $\Psi : L^1(\Omega; \mathbb{R}^k) \to C^{s'+1}(\overline{\Omega}; \mathbb{R}^{k'})$, such that

$$\sup_{u \in \mathsf{K}} \left\| \sum_{j=1}^J \alpha_j(u)\eta_{j,\delta} - \Psi(u) \right\|_{C^{s'+1}} \le \epsilon/3C_0,$$

where $C_0 > 0$ denotes the embedding constant of (A.20). For this averaging neural operator Ψ, it follows that

$$
\begin{aligned}
\sup_{u \in \mathsf{K}} \left\| \Psi^\dagger(u) - \Psi(u) \right\|_{W^{s',p'}} &\le \sup_{u \in \mathsf{K}} \left\| \Psi^\dagger(u) - \sum_{j=1}^J \alpha_j(u)\eta_{j,\delta} \right\|_{W^{s',p'}} \\
&\quad + \sup_{u \in \mathsf{K}} \left\| \sum_{j=1}^J \alpha_j(u)\eta_{j,\delta} - \Psi(u) \right\|_{W^{s',p'}} \\
&\le \sup_{u \in \mathsf{K}} \left\| \Psi^\dagger(u) - \sum_{j=1}^J \alpha_j(u)\eta_{j,\delta} \right\|_{W^{s',p'}} \\
&\quad + C_0 \sup_{u \in \mathsf{K}} \left\| \sum_{j=1}^J \alpha_j(u)\eta_{j,\delta} - \Psi(u) \right\|_{C^{s'+1}} \\
&\le \epsilon.
\end{aligned}
$$

This concludes our proof.   □

References

1. Anandkumar, A., Azizzadenesheli, K., Bhattacharya, K., Kovachki, N., Li, Z., Liu, B., Stuart, A.: Graph Kernel Network for Partial Differential Equations. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations* (2020)
2. Barron, A.R.: Universal approximation bounds for superpositions of a sigmoidal function. IEEE Trans. Inform. Theory. **39**(3), 930–945 (1993)
3. Bhattacharya, K., Hosseini, B., Kovachki, N. B., Stuart, A. M.: Model reduction and neural networks for parametric pdes. *The SMAI Journal of Computational Mathematics*, 7:121–157, 2021. Publisher: Société de Mathématiques Appliquées et Industrielles
4. Blouza, A., Le Dret, H.: An up-to-the boundary version of Friedrichs's lemma and applications to the linear Koiter shell model. SIAM J. Math. Anal. **33**(4), 877–895 (2001)
5. Bonev, B., Kurth, T., Hundt, C., Pathak, J., Baust, M., Kashinath, K., Anandkumar, A.: Spherical fourier neural operators: Learning stable dynamics on the sphere. *arXiv preprint* arXiv:2306.03838 (2023)
6. Cai, S., Wang, Z., Lu, L., Zaki, T. A., Karniadakis G. E.: DeepM&Mnet: Inferring the electroconvection multiphysics fields based on operator approximation by neural networks. *Journal of Computational Physics*, 436:110296, 2021. Publisher: Elsevier
7. Cao, Q., Goswami, S., Karniadakis, A. M.: Lno: Laplace neural operator for solving differential equations. *arXiv preprint* arXiv:2303.10528 (2023)
8. Chen, G., Liu, X., Meng, Q., Chen, L., Liu, C., Li Y.: Learning neural operators on riemannian manifolds (2023)
9. Chen, T., Chen, H.: Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 6(4):911–917, 1995. Publisher: IEEE
10. Cohen, A., DeVore, R.: Approximation of high-dimensional parametric pdes. Acta Numer **24**, 1–159 (2015)
11. Cohen, A., DeVore, R., Schwab, C.: Convergence rates of best n-term galerkin approximations for a class of elliptic spdes. Found. Comput. Math. **10**(6), 615–646 (2010)
12. Cybenko, G.: Approximations by superpositions of sigmoidal functions. Approx. Theory Appl. **9**(3), 17–28 (1989)
13. De Hoop, M., Huang, D. Z., Qian, E., Stuart, A. M.: The cost-accuracy trade-off in operator learning with neural networks. *Journal of Machine Learning*, 1(1:3):299–341 (2022)
14. de Hoop, M.V., Kovachki, N.B., Nelsen, N.H., Stuart, A.M.: Convergence rates for learning linear operators from noisy data. SIAM/ASA Journal on Uncertainty Quantification **11**(2), 480–513 (2023)
15. Di Leoni, P. C., Lu, L., Meneveau, C., Karniadakis, G., Zaki, T. A.: DeepONet prediction of linear instability waves in high-speed boundary layers. *arXiv preprint* arXiv:2105.08697 (2021)

16. Ern, A., Guermond, J.-L.: Mollification in strongly lipschitz domains with application to continuous and discrete de rham complexes. Computational Methods in Applied Mathematics **16**(1), 51–75 (2016)
17. Evans, L.C.: Partial differential equations. American Mathematical Society, vol. 19, 2nd edn. Providence, Rhode Island (2010)
18. Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT press (2016)
19. Gupta, G., Xiao, X., Bogdan, P.: Multiwavelet-based operator learning for differential equations. Adv. Neural. Inf. Process. Syst. **34**, 24048–24062 (2021)
20. Herrmann, L., Schwab, C., Zech. J.: Neural and spectral operator surrogates: unified construction and expression rate bounds. *Advances in Computational Mathematics (accepted)* (2024)
21. Hesthaven, J. S., Ubbiali, S.: Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55–78, 2018. Publisher: Elsevier
22. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Netw. **2**(5), 359–366 (1989)
23. Jackson, D.: *The Theory of Approximation*, volume 11. American Mathematical Soc. (1930)
24. Jin, P., Meng, S., Lu, L.: Mionet: learning multiple-input operators via tensor product. SIAM J. Sci. Comput. **44**(6), A3490–A3514 (2022)
25. Kaarnioja, V., Kuo, F. Y., Sloan, I. H.: Lattice-based kernel approximation and serendipitous weights for parametric pdes in very high dimensions. *arXiv preprint* arXiv:2303.17755 (2023)
26. Kovachki, N., Lanthaler, S., Mishra, S.: On universal approximation and error bounds for Fourier neural operators. J. Mach. Learn. Res. **22**(290), 1–76 (2021)
27. Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., Anandkumar, A.: Neural operator: learning maps between function spaces with applications to PDEs. J. Mach. Learn. Res. **24**(89), 1–97 (2023)
28. Lanthaler, S., Mishra, S., Karniadakis, G. E.: Error estimates for DeepOnets: A deep learning framework in infinite dimensions. *arXiv preprint* arXiv:2102.09618 (2021)
29. Lanthaler, S., Molinaro, R., Hadorn, P., Mishra, S.: Nonlinear reconstruction for operator learning of PDEs with discontinuities. *arXiv preprint* arXiv:2210.01074 (2022)
30. Lanthaler, S., Nelsen N. H.: Error bounds for learning with vector-valued random features. *Advances in Neural Information Processing Systems*, 36, (2024)
31. Li, Z., Huang, D. Z., Liu, B., Anandkumar, A.: Fourier neural operator with learned deformations for PDEs on general geometries. *arXiv preprint* arXiv:2207.05209, (2022)
32. Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., Anandkumar, A.: Markov neural operators for learning chaotic systems. *Neurips*, arXiv:2106.06898, (2022)
33. Li, Z., Kovachki, N.B., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., Anandkumar, A.: Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, (2021)
34. Li, Z., Kovachki, N.B., Azizzadenesheli, K., Liu, B., Stuart, A.M., Bhattacharya, K., Anandkumar, A.: Multipole graph neural operator for parametric partial differential equations. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 6755–6766. Curran Associates, Inc., (2020)
35. Li, Z., Liu-Schiaffini, M., Kovachki, N., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., Anandkumar, A.: Learning dissipative dynamics in chaotic systems. *Advances in Neural Information Processing Systems (NeurIPS)* arXiv:2106.06898, (2021)
36. Li, Z., Zheng, H., Kovachki, N., Jin, D., Chen, H., Liu, B., Azizzadenesheli, K., Anandkumar, A.: Physics-informed neural operator for learning partial differential equations. *arXiv preprint* arXiv:2111.03794, (2021)
37. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint* arXiv:1608.03983, (2016)
38. Lu, L., Jin, P., Pang, G., Karniadakis, G.E.: Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. Nat Mach Intell **3**, 218–229 (2021)
39. Mao, Z., Lu, L., Marxen, O., Zaki, T.A., Karniadakis, G.E.: DeepMandMnet for hypersonics: Predicting the coupled flow and finite-rate chemistry behind a normal shock using neural-network approximation of operators. J. Comput. Phys. **447**, 110698 (2021)
40. Marcati, C., Schwab, C.: Exponential convergence of deep operator networks for elliptic partial differential equations. SIAM J. Numer. Anal. **61**(3), 1513–1545 (2023)
41. Nelsen, N.H., Stuart, A.M.: The random feature model for input-output maps between Banach spaces. SIAM J. Sci. Comput. **43**(5), A3212–A3243 (2021)

42. Patel, D., Ray, D., Abdelmalik, M.R.A., Hughes, T.J.R., Oberai, A.A.: Variationally mimetic operator networks, (2022)
43. Patel, R.G., Desjardins, O.: Nonlinear integro-differential operator regression with neural networks, (2018)
44. Patel, R.G., Trask, N.A., Wood, M.A., Cyr, E.C.: A physics-informed operator regression framework for extracting data-driven continuum models. Comput. Methods Appl. Mech. Eng. **373**, 113500 (2021)
45. Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., Kurth, T., Hall, D., Li, Z., Azizzadenesheli, K., et al.: Fourcastnet: A global data-driven high-resolution weather model using adaptive Fourier neural operators. *arXiv preprint* arXiv:2202.11214, (2022)
46. Pinkus, A.: Approximation theory of the MLP model in neural networks. Acta Numer **8**, 143–195 (1999)
47. Raonić, B., Molinaro, R., Rohner, T., Mishra, S.: and E. de Bezenac, Convolutional neural operators (2023)
48. Seidman, J.H., Kissas, G., Perdikaris, P., Pappas, G.J.: NOMAD: Nonlinear manifold decoders for operator learning. In Oh, A.H., Agarwal, A., Belgrave, D., Cho, K., editors, *Advances in Neural Information Processing Systems*, (2022)
49. Stein, E.M.: *Singular Integrals and Differentiability Properties of Functions*. Princeton University Press, (1970)
50. Tripura, T., Chakraborty, S.: Wavelet neural operator for solving parametric partial differential equations in computational mechanics problems. Comput. Methods Appl. Mech. Eng. **404**, 115783 (2023)
51. Wen, G., Li, Z., Azizzadenesheli, K., Anandkumar, A., Benson, S.M.: U-FNO–an enhanced Fourier neural operator-based deep-learning model for multiphase flow. Adv. Water Resour. **163**, 104180 (2022)
52. You, H., Zhang, Q., Ross, C.J., Lee, C.-H., Yu, Y.: Learning deep implicit Fourier neural operators (ifnos) with applications to heterogeneous material modeling. Comput. Methods Appl. Mech. Eng. **398**, 115296 (2022)
53. Zhang, Z., Leung, W.T., Schaeffer, H.: Belnet: Basis enhanced learning, a mesh-free neural operator. *arXiv preprint* arXiv:2212.07336, (2022)