Journal of Machine Learning ISSN: 2790-2048(e), 2790-203X(p)

Approximation of Functionals by Neural Network without Curse of Dimensionality

Yahong Yang * [1] and Yang Xiang † [1,2]

[1]Department of Mathematics, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong.
[2]HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute, Futian, Shenzhen, China.

Abstract. In this paper, we establish a neural network to approximate functionals, which are maps from infinite dimensional spaces to finite dimensional spaces. The approximation error of the neural network is $\mathcal{O}(1/\sqrt{m})$ where m is the size of networks. In other words, the error of the network is no dependence on the dimensionality respecting to the number of the nodes in neural networks. The key idea of the approximation is to define a Barron space of functionals.

Keywords:
Functionals,
Neural networks,
Infinite dimensional spaces,
Barron spectral space,
Fourier series.

Article Info.:
Volume: 1
Number: 4
Pages: 342 - 372
Date: December/2022
doi.org/10.4208/jml.221018

Article History:
Received: 18/10/2022
Accepted: 20/12/2022

Communicated by:
Weinan E

1 Introduction

Recently, neural networks have revolutionized many fields of science and engineering including computational and applied mathematics. As one of the important applications of neural networks in applied mathematics, many methods have been developed on employing neural networks to approximate functionals and operators, which are maps from spaces of infinite dimensions.

One application of learning in infinite dimensional spaces is to solve partial differential equations (PDEs) by neural networks, e.g., [8,12,15,16,18,20,27,29,32–34]. The boundary value problem of a PDE in a domain Ω in d-dimensional space takes the form

$$\begin{cases} \mathcal{L}u = g_1 & \text{in } \Omega, \\ \mathcal{A}u = g_2 & \text{on } \partial\Omega, \end{cases} \tag{1.1}$$

where u is the unknown function, $\mathcal{L}$ is a partial differential operator, $\mathcal{A}$ is the operator for specifying an appropriate boundary condition, g_1 and g_2 are given functions, and without loss of generality, $\Omega = [0,1]^d$. The key idea of using neural networks to solve PDEs is to obtain $u(\boldsymbol{x};\boldsymbol{\theta})$ from a neural network, where $\boldsymbol{\theta}$ denotes all the parameters in the neural network that are trained by optimizing some loss function associated with the PDE. That is, in these methods, a neural network is established for the solution function $u(\boldsymbol{x})$ with

*`yyangct@connect.ust.hk`
†Corresponding author. `maxiang@ust.hk`

given function pair g_1 and g_2; for a different pair of functions g_1, g_2 in Eq. (1.1), the $u(\boldsymbol{x};\boldsymbol{\theta})$ from the neural network has to be learned again although operators $\mathcal{L}$ and $\mathcal{A}$ are the same.

The PDE boundary value problem in Eq. (1.1) can be considered as an operator $u(\boldsymbol{x}) = G(g_1, g_2)$. If we can learn the operator G directly by a neural network, we will be able to obtain the solution of Eq. (1.1) for any given function pair g_1, g_2 without learning again. A few methods have been proposed for learning operators by neural networks for solving PDEs, such as DeepONet [28], DeepGreen [14], Fourier Neural Operator (FNO) [22], Neural Operator [23, 24], MOD-Net [37], and the deep learning-based nonparametric estimation [26]. The DeepGreen, Neural Operator and MOD-Net methods are based on Green's functions for solving PDEs, i.e., these methods learn the Green's function instead of learning the operator directly. Since in general only solutions of linear PDEs have the Green's function formulation, those Green's function based methods cannot be used directly to solve nonlinear PDEs, and accuracy of some proposed attempts for the extension of those Green's function based methods to nonlinear PDEs has not been rigorously proved in the literature. The DeepONet [28] is a method that learns nonlinear operators associated with PDEs from data based on the approximation theorem for operators by neural networks [6]. The method in [26] is to learn the operator by model reduction [4] of reducing the operator to a finite dimensional space. Most of these available works on learning operators focused on the development of algorithms.

The curse of dimensionality is a serious issue that generally exists for approximations in high dimensions. Note that for the space of functions as the domain of the operator associated with PDE boundary value problem in Eq. (1.1), the dimension is ∞. The curse of dimensionality [3] summarizes this property that in order to maintain the accuracy of an approximation, the number of sample points grows exponentially with the increase of dimension. This means that for a fixed number of sample points, the accuracy will be lost in an exponential way as the dimension increases. Only a few analyses have been performed on overcoming the curse of dimensionality [17, 21, 26], and they all focused on reducing the infinite dimensional space to a low dimensional space. Exponential dependence on the dimension for the sample points still exists in these methods, which requires that the dimension of the reduced space has to be low enough. Moreover, the Bayesian inversion learning method in [17] is only for linear operators.

Approximating functionals and operators directly by reducing the input space to a finite dimensional space (e.g., DeepONet [28] reviewed above) will suffer from the curse of dimensionality, unless the input space has some low-dimensional structure. In Appendix A, we present a general error analysis for the DeepONet method in terms of the size and the number of parameters of the neural network. It serves as an examination of the sources of the curse of dimensionality in these type of methods.

In this paper, we focus on the approximation of functionals, i.e., maps from a space of functions which has infinite dimensions to $\mathbb{R}$, by neural networks without curse of dimensionality. Functionals, such as linear functionals (e.g., integrations, norms and inner-products of functions) and energy functionals, have a wide range of important applications in science and engineering fields. Moreover, approximating functionals by neural networks is also a crucial step in many available methods for approximating operators by neural networks and these methods suffer from curse of dimensionality. There are only limited attempts in the literature on using neural networks to approximate functionals [5, 35]. In [5], they reduced the approximation of functionals to the approximation of functions by reducing the function space into a finite dimensional space ([5, Theorem B.2]), which was adopted in the second step of the DeepONet method, and as we discussed above, this treatment has a serious curse of dimensionality. In [35], they approximated functionals directly by neural networks but did not show the error of their method.

Several methods for overcoming the curse of dimensionality in approximations of functions have been developed in the literature. An approximation method was proposed by Barron [2] based on some function space with spectral norm; see also further developments of this approach [1, 19, 27, 31]. Another type of function spaces based on the neural network representation and probability in the parameter space were also introduced and developed [9–11, 25, 29, 36]. Following [10, 11], in this paper, the former type of spaces are referred to as Barron spectral spaces, and the latter Barron spaces. The approximation error in a Barron/Barron spectral space is able to reduce to $\mathcal{O}(1/\sqrt{m})$, where m is the size of networks. However, the curse of dimensionality in approximation functionals cannot be solved by these Barron/Barron spectral space methods directly. In fact, the domain of functionals is an infinite dimensional space which is essentially different from the space of functions. A naive idea of generalizing the Barron/Barron spectral space methods for functions to functionals is to approximate the infinite dimensional space of functions by some finite dimensional space. However, as we demonstrated above for the DeepONet method, the approximation of an infinite dimensional space by a finite dimensional space through sample points still has the problem of curse of dimensionality (the second error in Eq. (A.5)), even though the curse of dimensionality in the last step in the DeepONet method (the last error in Eq. (A.5)) in principle can be overcome by some form of the Barron/Barron spectral space method for functions without curse of dimensionality. Furthermore, in such a straightforward generalization, we only know that the finite dimensional function h_k in Eq. (A.3) exists, and it is not easy to check if h_k belongs to the Barron/Barron spectral space.

In this paper, we establish a new method for the approximation of functionals by neural networks without curse of dimensionality, which is based on Fourier series of functionals and the associated Barron spectral space of functionals. Specifically, we first establish Fourier series of functionals, and then prove that any functional satisfying some proper assumption (Assumption 2.1) can be approximated by these Fourier series (Theorem 2.1). We then define a Barron spectral space $\mathcal{B}_s$ and a Hilbert space $\mathcal{H}_s$ of the functionals (Definition 2.4) based on the Fourier series, and estimate the error of the approximating neural network based on the Barron spectral space (Theorem 3.1). The approximation error of the neural network is $\mathcal{O}(1/\sqrt{m})$ where m is the size of networks, which overcomes the curse of dimensionality. Under some stronger conditions (including smoothness of the functions in the domain of the functional), a simpler method for learning functionals by neural networks has been proved (Theorem 4.1). Applications of these obtained theorems on the approximation of functionals by neural networks including the application to solving PDEs by neural networks are discussed.

Our main contributions are:

- We establish a new method for the approximation of functionals by a neural network, by defining (i) a Fourier-type series in the infinite-dimensional space of functionals and (ii) the associated spectral Barron spectral space $\mathcal{B}_s$ and a Hilbert space $\mathcal{H}_s$ of functionals. We show that the proposed method for the approximation of functionals overcomes the curse of dimensionality.

- The established method for approximation of functionals without curse of dimensionality can be employed in learning functionals, such as linear functionals and energy functionals in science and engineering fields. It can also be used to solve PDE problems by neural networks at some given points. This method provides a basis for the further development of methods for learning operators.

2 Fourier Series and Barron Spectral Space of Functionals

In this section, we define a Barron spectral space of functionals. For this purpose, we first define a Fourier series in the infinite-dimensional space of functionals. These definitions associated with the space of functionals are based on a basis of the domain of the functionals (which is a function space).

First of all, we define the index set of a basis of the space of functionals.

Definition 2.1 (Index set of basis of functionals). $\mathbb{K} \subset \mathbb{Z}^\infty$ *is defined by*

$$\mathbb{K} := \bigcup_{m=1}^{\infty} \mathbb{K}^m, \tag{2.1}$$

where

$$\mathbb{K}^m := \big\{ \boldsymbol{k} := (k_1, k_2, \cdots, k_i, \cdots) \in \mathbb{Z}^\infty : N_k := \max\{i : k_i \neq 0\} \leq m \big\}.$$

Due to the definition of $\mathbb{K}$, we know that $\boldsymbol{k} := (k_1, k_2, \cdots, k_i, \cdots) \in \mathbb{K}$ has only finite nonzero k_i. Furthermore, $\mathbb{K}$ is a countable set due to Axiom of Choice; see Lemma 2.1 below. Note that all the proofs in this paper including that of this lemma are given in Appendix A.

Lemma 2.1. $\mathbb{K}$ *is a countable set.*

Proof. Denote the cardinality of a set $\mathbb{A}$ as $|\mathbb{A}|$ and $|\mathbb{N}| = \aleph_0$. Hence,

$$\begin{aligned} |\mathbb{K}^m| &= \big| \big\{ \boldsymbol{k} := (k_1, k_2, \cdots, k_i, \cdots) \in \mathbb{N}^\infty : N_{\boldsymbol{k}} := \max\{i : k_i \neq 0\} \leq m \big\} \big| \\ &= |\mathbb{N}^m| = \aleph_0. \end{aligned} \tag{2.2}$$

Since a countable union of countable sets is countable (by using the Axiom of Choice), we have

$$|\mathbb{K}| = \left| \bigcup_{m=1}^{\infty} \mathbb{K}^m \right| = \aleph_0. \tag{2.3}$$

The proof is complete. □

DOI https://doi.org/10.4208/jml.221018 | Generated on 2024-12-19 04:40:52

Denote the domain of functionals as Ω, which is a Banach space of functions. Let $\{\Phi_i\}_{i=1}^{\infty}$ be a Schauder basis of Ω, i.e., for every element $v \in \Omega$, there exists a unique sequence $\{b_i\}_{i=1}^{\infty}$ of scalars in $\mathbb{R}$ such that $v = \sum_{i=1}^{\infty} b_i \Phi_i$. For example,

$$\{\Phi_n\}_{n=1}^{\infty} = \left\{\sqrt{2}\sin(2\pi n x) \mid n \in \mathbb{N}\right\} \cup \left\{\sqrt{2}\cos(2\pi n x) \mid n \in \mathbb{N}\right\} \cup \{1\},$$

when $\Omega = L_2(0,1)$. Note that we will define a Barron spectral space of functionals based only on the sequence $\{b_i\}_{i=1}^{\infty}$, and the defined Barron spectral space of functionals will apply to any basis $\{\Phi_i\}_{i=1}^{\infty}$ and any convergence of $v = \sum_{i=1}^{\infty} b_i \Phi_i$ as long as $\{b_i\}_{i=1}^{\infty}$ is uniquely determined. Next, we define a basis of the functionals on Ω.

Definition 2.2 (Fourier basis of functionals)**.** *For any*

$$v = \sum_{i=1}^{\infty} b_i \Phi_i \in L_{bound}(\Omega),$$

where

$$L_{bound}(\Omega) := \left\{ v = \sum_{i=1}^{\infty} b_i \Phi_i \in \Omega : -\frac{1}{2} < b_i < \frac{1}{2}, i \in \mathbb{N}_+ \right\}, \tag{2.4}$$

the Fourier basis $\{e_{\boldsymbol{k}}\}_{\boldsymbol{k} \in \mathbb{K}}$ of the functionals with domain Ω based on $\{\Phi_i\}_{i=1}^{\infty}$ is

$$e_{\boldsymbol{k}}(v) := \prod_{i=1}^{\infty} \exp\left(2\pi \mathrm{i} k_i b_i\right) = \exp\left(\sum_{i=1}^{\infty} 2\pi \mathrm{i} k_i b_i\right). \tag{2.5}$$

Notice that for some functionals such as linear functionals Example 2.1

$$f\left(\sum_{i=1}^{\infty} b_i \Phi_i\right) = \sum_{i=1}^{\infty} f(b_i \Phi_i),$$

and energy functionals Example 2.2

$$E\left(\sum_{p \in \mathbb{N}^d} b_p \Phi_p\right) = \sum_{p \in \mathbb{N}^d} E(b_p \Phi_p),$$

they can be divided into summation of countable finite dimensional functionals. We make the following assumptions for the functionals being considered.

Assumption 2.1 (Finite dimensional summation)**.** For a basis of Ω, $\{\Phi_i\}_{i=1}^{\infty}$, there is a unique sequence of sets $\mathbb{D} := \{\mathbb{D}_j\}_{j=1}^{\infty}$, where $\mathbb{D}_j \subset \mathbb{N}_+$ and $|\mathbb{D}_j| < +\infty$, such that the functional f satisfies

$$f\left(\sum_{i=1}^{\infty} b_i \Phi_i\right) = \sum_{j=1}^{\infty} f_j\left(\sum_{i \in \mathbb{D}_j} b_i \Phi_i\right) \tag{2.6}$$

for $v = \sum_{i=1}^{\infty} b_i \Phi_i \in L_{\mathrm{bound}}(\Omega)$. For each $f_j(\sum_{i \in \mathbb{D}_j} b_i \Phi_i)$, it cannot be further divided into $f_j^{(1)}(\sum_{i \in \mathbb{D}_j^1} b_i \Phi_i)$ and $f_j^{(2)}(\sum_{i \in \mathbb{D}_j^2} b_i \Phi_i)$ where $|\mathbb{D}_j^1|, |\mathbb{D}_j^2| < |\mathbb{D}_j|$.

Note that Assumption 2.1 does not require $\mathbb{D}_j \cap \mathbb{D}_i = \emptyset$ for $i \neq j$. We will give examples of functionals that satisfy Assumption 2.1 and demonstrate its generality at the end of this section.

Due to Assumption 2.1, the infinite-dimensional space of functions as the domain of the functional can be decomposed into finite-dimensional subspaces. As a result, in each finite-dimensional subspace, Fourier coefficients can be defined. Under this assumption, we can define the Fourier coefficients of a functional as follows.

Definition 2.3 (Fourier coefficients of a functional). *Suppose that Assumption* 2.1 *holds for the functional f, the Fourier coefficients of the functional f with basis $\{e_{\boldsymbol{k}}\}_{\boldsymbol{k}\in\mathbb{K}}$ are defined as*

$$a_{\boldsymbol{k}}(f) := \sum_{j\in\mathbb{A}_{\boldsymbol{k}}} \int_{\left(-\frac{1}{2},\frac{1}{2}\right)^{|\mathbb{D}_j|}} f_j\Big(\sum_{i\in\mathbb{D}_j} b_i\Phi_i\Big)\Big[\prod_{i\in\mathbb{D}_j} \exp\left(-2\pi \mathrm{i} k_i b_i\right)\, \mathrm{d}b_i\Big], \tag{2.7}$$

where

$$\begin{aligned} &\mathbb{A}_{\boldsymbol{k}} := \left\{ j \in \mathbb{N} : \boldsymbol{k} \in \mathbb{K}_{\mathbb{D}_j} \right\}, \\ &\mathbb{K}_{\mathbb{D}_j} := \left\{ \boldsymbol{k} := (k_1, k_2, \cdots, k_i, \cdots) \in \mathbb{K} : k_i = 0 \; for \;\; i \notin \mathbb{D}_j \right\}. \end{aligned} \tag{2.8}$$

Note that the summation over $\mathbb{A}_{\boldsymbol{k}}$ is due to the fact that each $\boldsymbol{k}$ may be associated with multiple D_j's.

By Definition 2.3, we have the Fourier series of the functionals. We prove that functionals can be expanded into such Fourier series in the following theorem.

Theorem 2.1. *For a functional f that is defined on $L_{bound}(\Omega)$ and satisfies Assumption* 2.1 *and following two assumptions:*

(i) (Smoothness): For any $j \in \mathbb{N}_+$, $f_j(\sum_{i\in\mathbb{D}_j} b_i\Phi_i)$ is a C^1-function respect to b_i for $i \in \mathbb{D}_j$ in $(-1/2, 1/2)^{|\mathbb{D}_j|}$.

(ii) (Existence of $a_{\boldsymbol{k}}(f)$): For each $\boldsymbol{k} \in \mathbb{K}$, $a_{\boldsymbol{k}}(f)$ exists and the following condition holds:

$$\sum_{\boldsymbol{k}\in\mathbb{K}} \sum_{j\in\mathbb{A}_{\boldsymbol{k}}} \left| \int_{\left(-\frac{1}{2},\frac{1}{2}\right)^{|\mathbb{D}_j|}} f_j\Big(\sum_{i\in\mathbb{D}_j} b_i\Phi_i\Big)\Big[\prod_{i\in\mathbb{D}_j} \exp(-2\pi \mathrm{i} k_i b_i)\, \mathrm{d}b_i\Big] \right| < \infty, \tag{2.9}$$

we have

$$f(v) = \sum_{\boldsymbol{k}\in\mathbb{K}} a_{\boldsymbol{k}}(f) e_{\boldsymbol{k}}(v), \quad v \in L_{bound}(\Omega). \tag{2.10}$$

The series in Eq. (2.10) *is unconditionally convergent. We call this expansion the Fourier series of the functional f.*

Proof. For a functional f satisfying Assumption 2.1, it can be written as

$$f\left(\sum_{i=1}^{\infty} b_i\Phi_i\right) = \sum_{j=1}^{\infty} f_j\Big(\sum_{i\in\mathbb{D}_j} b_i\Phi_i\Big), \tag{2.11}$$

where $|\mathbb{D}_j| < \infty$. Furthermore, due to assumption (i) in Teotem 2.1, each f_j is a C^1-function with respect to $b_i \in \mathbb{D}_j$. Therefore, the Fourier series of f_j converge to f_j, i.e.,

$$f_j\left(\sum_{i\in\mathbb{D}_j} b_i\Phi_i\right) = \sum_{\boldsymbol{k}\in\mathbb{K}_{\mathbb{D}_j}} (\widehat{f_j})_{\boldsymbol{k}} \prod_{i\in\mathbb{D}_j} \exp(2\pi \mathrm{i} k_i b_i) = \sum_{\boldsymbol{k}\in\mathbb{K}_{\mathbb{D}_j}} (\widehat{f_j})_{\boldsymbol{k}} \prod_{i=1}^{\infty} \exp(2\pi \mathrm{i} k_i b_i), \tag{2.12}$$

where $\boldsymbol{k} := (k_1, k_2, \cdots, k_i, \cdots)$, and $(\widehat{f_j})_{\boldsymbol{k}}$ is the Fourier coefficient of f_j for the basis $\prod_{i\in\mathbb{D}_j} \exp(2\pi \mathrm{i} k_i b_i)$. By the definition of $a_{\boldsymbol{k}}(f)$ and direct calculation, we have

$$a_{\boldsymbol{k}}(f) = \sum_{j\in\mathbb{A}_{\boldsymbol{k}}} (\widehat{f_j})_{\boldsymbol{k}} = \sum_{j\in\mathbb{A}_{\boldsymbol{k}}} \int_{\left(-\frac{1}{2},\frac{1}{2}\right)^{|\mathbb{D}_j|}} f_j\left(\sum_{i\in\mathbb{D}_j} b_i\Phi_i\right) \left[\prod_{i\in\mathbb{D}_j} \exp(-2\pi \mathrm{i} k_i b_i)\, \mathrm{d}b_i\right]. \tag{2.13}$$

Therefore

$$\begin{aligned} f\left(\sum_{i=1}^{\infty} b_i\Phi_i\right) &= \sum_{j=1}^{\infty} f_j\left(\sum_{i\in\mathbb{D}_j} b_i\Phi_i\right) \\ &= \sum_{j=1}^{\infty} \sum_{\boldsymbol{k}\in\mathbb{K}_{\mathbb{D}_j}} (\widehat{f_j})_{\boldsymbol{k}} \prod_{i=1}^{\infty} \exp(2\pi \mathrm{i} k_i b_i) \\ &= \sum_{\boldsymbol{k}\in\mathbb{K}} \sum_{j\in\mathbb{A}_{\boldsymbol{k}}} (\widehat{f_j})_{\boldsymbol{k}} \prod_{i=1}^{\infty} \exp(2\pi \mathrm{i} k_i b_i) \\ &= \sum_{\boldsymbol{k}\in\mathbb{K}} a_{\boldsymbol{k}}(f) \prod_{i=1}^{\infty} \exp(2\pi \mathrm{i} k_i b_i). \end{aligned} \tag{2.14}$$

The third equation in (2.14) is obtained by change of order of the summations, which is due to assumption (ii). □

We will give examples of the Fourier expansions of functionals at the end of the section. Now we define the Barron spectral space and a Hilbert space of functionals based on the Fourier series of functionals.

Definition 2.4. *(i) The Barron spectral space of the continuous functionals on $L_{bound}(\Omega)$ that satisfy Assumption 2.1 is*

$$\mathcal{B}_s[L_{bound}(\Omega)] := \left\{ f : f(v) = \sum_{\boldsymbol{k}\in\mathbb{K}} a_{\boldsymbol{k}}(f) e_{\boldsymbol{k}}(v), \quad \sum_{\boldsymbol{k}\in\mathbb{K}} (1 + (2\pi)^s |\boldsymbol{k}|_1^s) |a_{\boldsymbol{k}}(f)| < \infty \right\}, \tag{2.15}$$

for $s \geq 0$ with the norm

$$\|f\|_{\mathcal{B}_s} := \sum_{\boldsymbol{k}\in\mathbb{K}} \left(1 + (2\pi)^s |\boldsymbol{k}|_1^s\right) |a_{\boldsymbol{k}}(f)|. \tag{2.16}$$

DOI https://doi.org/10.4208/jml.221018 | Generated on 2024-12-19 04:40:52

(ii) A Hilbert space of functionals on $L_{bound}(\Omega)$ that satisfy Assumption 2.1 is

$$\mathcal{H}_s\left[L_{bound}(\Omega)\right] := \Big\{ f : f(v) = \sum_{\boldsymbol{k}\in\mathbb{K}} a_{\boldsymbol{k}}(f) e_{\boldsymbol{k}}(v), \\ \sum_{\boldsymbol{k}\in\mathbb{K}} \left(1 + (2\pi)^{2s}|\boldsymbol{k}|_1^{2s}\right)|a_{\boldsymbol{k}}(f)|^2 < \infty \Big\} \tag{2.17}$$

for $s \geq 0$ with the inner product

$$\langle f, g\rangle_{\mathcal{H}_s} = \sum_{\boldsymbol{k}\in\mathbb{K}} \left(1 + (2\pi)^{2s}|\boldsymbol{k}|_1^{2s}\right)\overline{a_{\boldsymbol{k}}(f)} \cdot a_{\boldsymbol{k}}(g), \tag{2.18}$$

where $\overline{a_{\boldsymbol{k}}(f)}$ is the complex conjugate of $a_{\boldsymbol{k}}(f)$.

In this paper, we focus on the Barron spectral space $\mathcal{B}_2[L_{\text{bound}}(\Omega)]$ and the Hilbert space $\mathcal{H}_1[L_{\text{bound}}(\Omega)]$, and for simplicity of notations, denote them as $\mathcal{B}[L_{\text{bound}}(\Omega)]$ and $\mathcal{H}[L_{\text{bound}}(\Omega)]$, respectively. Note that it is easy to check that $\mathcal{H}_s[L_{\text{bound}}(\Omega)]$ defined above is a Hilbert space.

Many widely used functionals in science and engineering satisfy Assumption 2.1 and the defined Fourier expansion applies. Here we give some examples.

Example 2.1. For a linear functional f,

$$f\left(\sum_{i=1}^{\infty} b_i\Phi_i\right) = \sum_{i=1}^{\infty} f(b_i\Phi_i).$$

In this case, Assumption 2.1 is satisfied with

$$\mathbb{D} = \left\{\mathbb{D}_j\right\}_{j=1}^{\infty} = \left\{\{1\},\{2\},\cdots,\{j\},\cdots\right\}.$$

For any $\boldsymbol{k} \in \mathbb{K}_{\mathbb{D}_j}$, there is only one component of $\boldsymbol{k}$, i.e., k_j, is nonzero, and for any $v = \sum_{i=1}^{\infty} b_i\Phi_i$, we have

$$\begin{aligned} e_{\boldsymbol{k}}(v) &= \exp\left(2\pi \mathrm{i} k_j b_j\right), \\ a_{\boldsymbol{0}}(f) &:= \sum_{i=1}^{\infty}\int_{\left(-\frac{1}{2},\frac{1}{2}\right)} f\left(b_i\Phi_i\right)\,\mathrm{d}b_i, \\ a_{\boldsymbol{k}}(f) &= \int_{\left(-\frac{1}{2},\frac{1}{2}\right)} f\left(b_j\Phi_j\right)\exp\left(-2\pi \mathrm{i} k_j b_j\right)\,\mathrm{d}b_j, \quad \boldsymbol{k} \neq \boldsymbol{0}. \end{aligned}$$

Therefore, the Fourier expansion (2.10) of the linear functional f is

$$f\left(\sum_{i=1}^{\infty} b_i\Phi_i\right) = \sum_{i=1}^{\infty}\sum_{k_i\in\mathbb{Z}}\left(\int_{\left(-\frac{1}{2},\frac{1}{2}\right)} f(b_i\Phi_i)\exp(-2\pi \mathrm{i} k_i b_i)\,\mathrm{d}b_i\right)\exp(2\pi \mathrm{i} k_i b_i). \tag{2.19}$$

The functional f is in the Barron spectral space if it further satisfies the condition in Definition 2.4(i).

Note that linear functionals play important roles in both theoretical studies and applications in science and engineering, e.g., integration, inner product with a given function, solution of initial/boundary value problem of linear PDEs at some given points, etc. This important class of functionals always satisfies Assumption 2.1.

Example 2.2. Consider the energy functional

$$E(v) = \int_{(0,1)^d} \frac{1}{2}\alpha|\nabla v|^2 \,\mathrm{d}x,$$

where $\alpha > 0$ and $d \in \mathbb{N}$. Here $v \in \Omega = H^1(0,1)^d$ with $\partial v/\partial \nu = 0$ for $x \in \partial[0,1]^d$ (where $\partial/\partial\nu$ is the outer normal derivative on the boundary), and an orthogonal basis of Ω is

$$\left\{\Phi_p(x) = \prod_{j=1}^{d} \sqrt{2}\cos(\pi p_j x_j)\right\}_{p\in\mathbb{N}^d\setminus\{0\}} \cup \{\Phi_0 = 1\},$$

where p_j and x_j are components of p and x, respectively. For any $v = \sum_{p\in\mathbb{N}^d} b_p\Phi_p$, the energy can be written as

$$E(v) = \int_{(0,1)^d} \frac{1}{2}\alpha\left|\sum_{p\in\mathbb{N}^d}\big(\pi b_p\Phi_p\big)p\right|^2 \mathrm{d}x = \sum_{p\in\mathbb{N}^d}\frac{1}{2}\pi^2\alpha|p|^2 b_p^2 = \sum_{p\in\mathbb{N}^d} E\big(b_p\Phi_p\big).$$

Here

$$E\big(b_p\Phi_p\big) = \frac{1}{2}\pi^2\alpha|p|^2 b_p^2.$$

In this case, we also have that Assumption 2.1 is satisfied with $\mathbb{D} = \{\mathbb{D}_p\}_{p\in\mathbb{N}^d}$, where $\mathbb{D}_p = \{p\}$ for $p \in \mathbb{N}^d$. We further restrict the domain to

$$\left\{v(x) = \sum_{p\in\mathbb{N}^d} b_p\Phi_p : -B_p < b_p < B_p,\ B_p = C|p|^{-\frac{d}{2}-1-\varepsilon},\ p \in \mathbb{N}^d\right\},$$

where $C, \varepsilon > 0$, so that the energy is well-defined. For any $\boldsymbol{k} \in \mathbb{K}_{\mathbb{D}_p}$, there is only one component of $\boldsymbol{k}$, denoted by k_p, is nonzero, and we have

$$\begin{aligned}
e_{\boldsymbol{k}}(v) &= \frac{1}{(2B_p)^{\frac{1}{2}}}\exp\left(\frac{\pi}{B_p}\mathrm{i}k_p b_p\right),\\
a_{\boldsymbol{0}}(E) &= \sum_{p\in\mathbb{Z}^d}\frac{1}{(2B_p)^{\frac{1}{2}}}\int_{-B_p}^{B_p} E\big(b_p\Phi_p\big)\,\mathrm{d}b_p,\\
a_{\boldsymbol{k}}(E) &= \frac{1}{(2B_p)^{\frac{1}{2}}}\int_{-B_n}^{B_n} E\big(b_p\Phi_p\big)\exp\left(-\frac{\pi}{B_p}\mathrm{i}k_p b_p\right)\mathrm{d}b_p.
\end{aligned}$$

The Fourier expansion (2.10) of this energy functional E is

$$E = \sum_{p\in\mathbb{N}^d}\sum_{k_p\in\mathbb{Z}}\left(\frac{1}{2B_p}\int_{-B_p}^{B_p} E\big(b_p\Phi_p\big)\exp\left(-\frac{\pi}{B_p}\mathrm{i}k_p b_p\right)\,\mathrm{d}b_p\right)\exp\left(\frac{\pi}{B_p}\mathrm{i}k_p b_p\right). \qquad (2.20)$$

The functional E is in the Barron spectral space if it further satisfies the condition in Definition 2.4(i).

Note that our method also applies to many other similar nonlinear functionals such as the elastic energy of a deformed materials

$$E = \frac{1}{2}\sum_{ijkl} C_{ijkl}\varepsilon_{ij}\varepsilon_{kl},$$

where $\{C_{ijkl}\}$ is the elastic constant tensor,

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j}\right)$$

is the strain tensor, and u_i is a component of the displacement vector.

We further consider an example in which the functions are defined in a domain of discrete points. In physics, chemistry, and materials science, atomistic models are commonly used as a tool to study the materials properties [13]. Inter-atomic potentials are used in atomistic models, and many of them are pairwise potentials with finite interaction range. The total energy of an atomistic system depends on the positions of atoms. The number of atomics is typically very large in the atomistic simulations.

Example 2.3. Consider an example of the total energy of a one dimensional atomistic system

$$E = \sum_{i\in\mathbb{N}} \left[V(a + u_{i+1} - u_i) - V(a)\right],$$

where $V(r)$ is a two-body potential such as the Lennard-Jones potential ($V(r) = 4\varepsilon[(\sigma/r)^{12} -(\sigma/r)^6]$ with ε and σ being two parameters), where r is the distance between two atoms, a is the lattice constant, and u_i the displacement of the i-th atom. Since the number of atoms is very large, we simply write the number as infinity. Note that the convergence of the summation is not a problem in a real system because the number of atoms is always finite; or we can add some decaying condition on $\{u_i\}$ as in Example 2.2.

In this case, Assumption 2.1 is satisfied with $\mathbb{D} = \{\{1,2\},\{2,3\},\cdots,\{n,n+1\},\cdots\}$. Denote $u = (u_1, u_2, \cdots, u_n, \cdots)$. The Fourier expansion (2.10) of this energy functional E is $\sum_{\boldsymbol{k}\in\mathbb{K}} a_{\boldsymbol{k}}(E)e_{\boldsymbol{k}}(u)$, which is

$$\begin{aligned} E = &\sum_{i\in\mathbb{N}} \sum_{(k_i,k_{i+1})\in\mathbb{Z}^2} \exp(2\pi \mathrm{i} k_i u_i + 2\pi \mathrm{i} k_{i+1} u_{i+1}) \\ &\times \int_{\left(-\frac{1}{2},\frac{1}{2}\right)^2} \left[V(a + u_{j+1} - u_j) - V(a)\right] \exp(2\pi \mathrm{i} k_i u_i + 2\pi \mathrm{i} k_{i+1} u_{i+1})\, \mathrm{d}u_i\, \mathrm{d}u_{i+1}. \end{aligned} \tag{2.21}$$

The functional E is in the Barron spectral space if it further satisfies the condition in Definition 2.4(i), where the Fourier coefficient $a_{\boldsymbol{k}}(E)$ is the integral expression in Eq. (2.21). Note that we have similar Fourier expansion for the total energy of an atomic system with a general atomic interaction of finite range in any dimension.

Example 2.4. Consider the following infinite dimensional nonlinear functional $f^{(1)}$ satisfying Assumption 2.1:

$$f^{(1)}\left(\sum_{i=1}^{\infty} b_i \Phi_i\right) = \sum_{i=1}^{\infty} s_i b_i^3, \tag{2.22}$$

where $s_i \in \mathbb{R}$. Here $f^{(1)}$ is well-defined in $L_{\text{bound}}(\Omega)$ when $\sum_{i=1}^{\infty} |s_i| < \infty$. In this case, $\mathbb{D} = \{\{1\}, \{2\}, \cdots, \{n\}, \cdots\}$. Hence, we obtain that

$$\sum_{\boldsymbol{k} \in \mathbb{K}} a_{\boldsymbol{k}}(f^{(1)}) e_{\boldsymbol{k}}(v) = \sum_{i=1}^{\infty} \left(\sum_{\boldsymbol{k} \in \mathbb{K}_i} a_{\boldsymbol{k}}(f^{(1)}) e_{\boldsymbol{k}}(v) \right) = \sum_{i=1}^{\infty} \left(s_i \sum_{k \in \mathbb{Z}} d_k \exp(2\pi \mathrm{i} k b_i) \right), \tag{2.23}$$

where

$$\mathbb{K}_i := \{ \boldsymbol{k} \in \mathbb{K} : k_j = 0 \text{ for } j \neq i, \text{ and } k_j \neq 0 \text{ for } j = i \},$$
$$d_k := \int_{(-\frac{1}{2}, \frac{1}{2})} b_1^3 \exp(-2\pi \mathrm{i} k b_1) \, \mathrm{d}b_1.$$

When $b_i \in (-1/2, 1/2)$, $\sum_{k \in \mathbb{Z}} d_k \exp(2\pi \mathrm{i} k b_i)$ is the Fourier series of b_i^3. Therefore, when $\sum_{i=1}^{\infty} |s_i| < \infty$,

$$\sum_{\boldsymbol{k} \in \mathbb{K}} a_{\boldsymbol{k}}(f^{(1)}) e_{\boldsymbol{k}}(v) = \sum_{i=1}^{\infty} \left(s_i \sum_{k \in \mathbb{Z}} d_k \exp(2\pi \mathrm{i} k b_i) \right) = \sum_{i=1}^{\infty} (s_i b_i^3) = f^{(1)}\left(\sum_{i=1}^{\infty} b_i \Phi_i \right). \tag{2.24}$$

Furthermore, $f^{(1)} \in \mathcal{B}[L_{\text{bound}}(\Omega)]$ ($f^{(1)} \in \mathcal{H}[L_{\text{bound}}(\Omega)]$), when $\sum_{k \in \mathbb{Z}} (1 + (2\pi)^2 k^2) |d_k| < \infty$ ($\sum_{k \in \mathbb{N}_+} (1 + (2\pi)^2 k^2) |d_k|^2 < \infty, \sum_{i \in \mathbb{N}_+} |s_i|^2 < \infty$).

Example 2.5. Consider another example

$$f^{(2)}\left(\sum_{i=1}^{\infty} b_i \Phi_i\right) = \sum_{i=1}^{\infty} s_i b_i b_{i+1}. \tag{2.25}$$

Here $f^{(2)}$ is well-defined in $L_{\text{bound}}(\Omega)$ when $\sum_{i=1}^{\infty} |s_i| < \infty$. In this case,

$$\mathbb{D} = \{\{1,2\}, \{2,3\}, \cdots, \{n, n+1\}, \cdots\},$$

and

$$\begin{aligned} a_{\boldsymbol{k}}(f^{(2)}) = & \int_{(-\frac{1}{2}, \frac{1}{2})^2} s_i b_i b_{i+1} \exp(2\pi \mathrm{i} k_i b_i) \, \mathrm{d}b_i \, \mathrm{d}b_{i+1} \\ & + \int_{(-\frac{1}{2}, \frac{1}{2})^2} s_{i-1} b_i b_{i-1} \exp(2\pi \mathrm{i} k_i b_i) \, \mathrm{d}b_i \, \mathrm{d}b_{i-1}, \end{aligned} \tag{2.26}$$

when

$$\boldsymbol{k} \in \mathbb{K}_i := \{ \boldsymbol{k} \in \mathbb{K} : k_j = 0 \text{ for } j \neq i, \text{ and } k_j \neq 0 \text{ for } j = i \},$$

and $s_0 = b_0 = 0$. It can be calculated that

$$a_{\boldsymbol{k}}(f^{(2)}) = \int_{(-\frac{1}{2},\frac{1}{2})^2} s_i b_i b_{i+1} \exp(2\pi \mathrm{i} k_i b_i + 2\pi \mathrm{i} k_{i+1} b_{i+1})\, \mathrm{d}b_i\, \mathrm{d}b_{i+1}, \tag{2.27}$$

when

$$\boldsymbol{k} \in \bar{\mathbb{K}}_i := \{\boldsymbol{k} \in \mathbb{K} : k_j = 0 \text{ for } j \neq i,\ i+1, \text{ and } k_j \neq 0 \text{ for } j = i,\ i+1\}.$$

Otherwise $a_{\boldsymbol{k}}(f^{(2)}) = 0$. Therefore,

$$\begin{aligned}\sum_{\boldsymbol{k}\in\mathbb{K}} a_{\boldsymbol{k}}(f^{(2)}) e_{\boldsymbol{k}}(v) &= \sum_{i=1}^{\infty} \left(\sum_{\boldsymbol{k}\in\mathbb{K}_i\cup\mathbb{K}_{i+1}\cup\bar{\mathbb{K}}_i} a_{\boldsymbol{k}}(f^{(2)}) \exp(2\pi \mathrm{i} k_i b_i + 2\pi \mathrm{i} k_{i+1} b_{i+1}) \right) \\ &= \sum_{i=1}^{\infty} (s_i b_i b_{i+1}). \end{aligned} \tag{2.28}$$

In the above two examples, there is a sequence $\{s_i\}_{i=1}^{\infty}$ to make the functionals well-defined in $L_{\text{bound}}(\Omega)$ (i.e., the series converges).

Note that alternatively, convergence of the series can also be achieved by restricting the domain of functionals. For example, if we want to approximate functionals such as the L_2-norm: $f(v) = \|v\|_{L_2(0,1)}$, we can restrict the domain of the functionals to

$$\left\{ v = \sum_{i=1}^{\infty} b_i \Phi_i : -Ci^{-\frac{1}{2}-\varepsilon} < b_i < Ci^{-\frac{1}{2}-\varepsilon},\ i \in \mathbb{N}_+ \right\},$$

where $C, \varepsilon > 0$, and redefine $a_{\boldsymbol{k}}(f)$ as that in Eq. (4.2).

Example 2.6. Consider the functional $f(v) = \int_0^1 v^3\, \mathrm{d}x$ and an orthogonal basis of functions on $(0,1)$

$$\{\Phi_j(x) = \sqrt{2}\cos(\pi j x)\}_{j\in\mathbb{N}\setminus\{0\}} \cup \{\Phi_0 = 1\}.$$

For any $v = \sum_{j\in\mathbb{N}} b_j \Phi_j$, the functional f can be written as

$$\begin{aligned} f\left(\sum_{j\in\mathbb{N}} b_j \Phi_j\right) &= \sum_{j=0}^{\infty} \int_0^1 b_j^3 \Phi_j^3(x)\, \mathrm{d}x + 3 \sum_{j_1\neq j_2} \int_0^1 b_{j_1}^2 b_{j_2} \Phi_{j_1}^2(x) \Phi_{j_2}(x)\, \mathrm{d}x \\ &\quad + 6 \sum_{j_1,j_2,j_3 \text{different}} \int_0^1 b_{j_1} b_{j_2} b_{j_3} \Phi_{j_1}(x) \Phi_{j_2}(x) \Phi_{j_3}(x)\, \mathrm{d}x. \end{aligned} \tag{2.29}$$

Using the definition of $\{\Phi_j(x)\}$, it can be calculated that

$$f\left(\sum_{j\in\mathbb{N}} b_j \Phi_j\right) = b_0^3 + 3 \sum_{j_1=1}^{\infty} b_{j_1}^2 b_0 + \frac{3\sqrt{2}}{2} \sum_{\substack{2j_1=j_2 \\ j_1,j_2>0}} b_{j_1}^2 b_{j_2} + 3\sqrt{2} \sum_{\substack{j_1,j_2,j_3 \text{different} \\ j_1+j_2=j_3}} b_{j_1} b_{j_2} b_{j_3}. \tag{2.30}$$

Note that the convergence of the summation is not a problem since we can add some decaying condition on $\{b_j\}$ as in Example 2.2

DOI https://doi.org/10.4208/jml.221018 | Generated on 2024-12-19 04:40:52

$$\left\{v(x) = \sum_{j\in\mathbb{N}} b_j\Phi_j : -C_p < b_p < C_p,\ C_p = C|j|^{-1-\varepsilon},\ j \in \mathbb{N}\right\},$$

where $C, \varepsilon > 0$. In this case, we also have that Assumption 2.1 is satisfied with

$$\mathbb{D} = \left\{\{0\}, \{0, j_1\}_{j_1=1}^{\infty}, \{j_1, j_2\}_{\substack{2j_1=j_2\\ j_1,j_2>0}}, \{j_1, j_2, j_3\}_{\substack{j_1,j_2,j_3\text{different,}\\ j_1+j_2=j_3}}\right\}.$$

We also can write down the Fourier expansion of $f(v)$. We put the detail in the end of this example.

Now we present the Fourier coefficients of this functional $f(v)$ in the Fourier expansion (2.10). For any $\boldsymbol{k} \in \mathbb{K}$ with only one nonzero component of $\boldsymbol{k}$, denoted by k_{s_1}, and we have when $s_1 = 0$,

$$\begin{aligned} e_{\boldsymbol{k}}(v) &= \frac{1}{(2C_0)^{\frac{1}{2}}}\exp\left(\frac{\pi}{C_0}\mathrm{i}k_0b_0\right), \\ a_{\boldsymbol{k}}(f) &= \frac{1}{(2C_0)^{\frac{1}{2}}}\int_{-C_0}^{C_0} b_0^3 e_{\boldsymbol{k}}\,\mathrm{d}b_0 + 3\sum_{j_1=1}^{\infty}\frac{1}{(2C_0)^{\frac{1}{2}}2C_{j_1}}\int_{-C_{j_1}}^{C_{j_1}}\int_{-C_0}^{C_0} b_0 b_{j_1}^2 e_{\boldsymbol{k}}\,\mathrm{d}b_0\,\mathrm{d}b_{j_1}, \end{aligned} \tag{2.31}$$

for an odd integer s_1,

$$\begin{aligned} e_{\boldsymbol{k}}(v) &= \frac{1}{(2C_{s_1})^{\frac{1}{2}}}\exp\left(\frac{\pi}{C_{s_1}}\mathrm{i}k_{s_1}b_{s_1}\right), \\ a_{\boldsymbol{k}}(f) &= \frac{3}{(2C_{s_1})^{\frac{1}{2}}2C_0}\int_{-C_0}^{C_0}\int_{-C_{s_1}}^{C_{s_1}} b_0 b_{s_1}^2 e_{\boldsymbol{k}}\,\mathrm{d}b_{s_1}\,\mathrm{d}b_0 \\ &\quad + \frac{3\sqrt{2}}{2}\frac{1}{(2C_{s_1})^{\frac{1}{2}}2C_{2s_1}}\int_{-C_{s_1}}^{C_{s_1}}\int_{-C_{2s_1}}^{C_{2s_1}} b_{2s_1}b_{s_1}^2 e_{\boldsymbol{k}}\,\mathrm{d}b_{2s_1}\,\mathrm{d}b_{s_1} \\ &\quad + 3\sqrt{2}\sum_{\substack{s_1,j_1,j_2\text{different}\\ s_1+j_1=j_2\text{ or }j_1+j_2=s_1}}\frac{1}{(2C_{s_1})^{\frac{1}{2}}4C_{j_1}C_{j_2}}\int_{-C_{j_2}}^{C_{j_2}}\int_{-C_{j_1}}^{C_{j_1}}\int_{-C_{s_1}}^{C_{s_1}} b_{j_2}b_{j_1}b_{s_1}e_{\boldsymbol{k}}\,\mathrm{d}b_{s_1}\,\mathrm{d}b_{j_1}\,\mathrm{d}b_{j_2}, \end{aligned} \tag{2.32}$$

and for an even integer $s_1 \neq 0$,

$$\begin{aligned} e_{\boldsymbol{k}}(v) &= \frac{1}{(2C_{s_1})^{\frac{1}{2}}}\exp\left(\frac{\pi}{C_{s_1}}\mathrm{i}k_{s_1}b_{s_1}\right), \\ a_{\boldsymbol{k}}(f) &= \frac{3}{(2C_{s_1})^{\frac{1}{2}}2C_0}\int_{-C_0}^{C_0}\int_{-C_{s_1}}^{C_{s_1}} b_0 b_{s_1}^2 e_{\boldsymbol{k}}\,\mathrm{d}b_{s_1}\,\mathrm{d}b_0 \\ &\quad + \frac{3\sqrt{2}}{2}\frac{1}{(2C_{s_1})^{\frac{1}{2}}2C_{2s_1}}\int_{-C_{s_1}}^{C_{s_1}}\int_{-C_{2s_1}}^{C_{2s_1}} b_{2s_1}b_{s_1}^2 e_{\boldsymbol{k}}\,\mathrm{d}b_{2s_1}\,\mathrm{d}b_{s_1} \\ &\quad + \frac{3\sqrt{2}}{2}\frac{1}{(2C_{s_1})^{\frac{1}{2}}2C_{s_1/2}}\int_{-C_{s_1}}^{C_{s_1}}\int_{-C_{s_1/2}}^{C_{s_1/2}} b_{s_1}b_{s_1/2}^2 e_{\boldsymbol{k}}\,\mathrm{d}b_{s_1/2}\,\mathrm{d}b_{s_1} \end{aligned}$$

$$+ 3\sqrt{2} \sum_{\substack{s_1, j_1, j_2 \text{different} \\ s_1+j_1=j_2 \text{ or } j_1+j_2=s_1}} \frac{1}{(2C_{s_1})^{\frac{1}{2}} 4C_{j_1} C_{j_2}} \int_{-C_{j_2}}^{C_{j_2}} \int_{-C_{j_1}}^{C_{j_1}} \int_{-C_{s_1}}^{C_{s_1}} b_{j_2} b_{j_1} b_{s_1} e_{\boldsymbol{k}} \, \mathrm{d}b_{s_1} \, \mathrm{d}b_{j_1} \, \mathrm{d}b_{j_2}. \quad (2.33)$$

For any $\boldsymbol{k} \in \mathbb{K}$ with only two nonzero components, denoted by k_{s_1}, k_{s_2}, when $s_2 = 0$,

$$e_{\boldsymbol{k}}(v) = \frac{1}{(4C_0 C_{s_1})^{\frac{1}{2}}} \exp\left(\frac{\pi}{C_0} \mathrm{i} k_0 b_0 + \frac{\pi}{C_{s_1}} \mathrm{i} k_{s_1} b_{s_1}\right),$$
$$a_{\boldsymbol{k}}(f) = 3 \frac{1}{(4C_0 C_{s_1})^{\frac{1}{2}}} \int_{-C_{s_1}}^{C_{s_1}} \int_{-C_0}^{C_0} b_0 b_{s_1}^2 e_{\boldsymbol{k}} \, \mathrm{d}b_0 \, \mathrm{d}b_{s_1}, \quad (2.34)$$

when $s_2 = 2s_1$,

$$e_{\boldsymbol{k}}(v) = \frac{1}{(4C_{s_2} C_{s_1})^{\frac{1}{2}}} \exp\left(\frac{\pi}{C_{s_2}} \mathrm{i} k_{s_2} b_{s_2} + \frac{\pi}{C_{s_1}} \mathrm{i} k_{s_1} b_{s_1}\right),$$
$$a_{\boldsymbol{k}}(f) = 3 \frac{1}{(4C_0 C_{s_1})^{\frac{1}{2}}} \int_{-C_{s_1}}^{C_{s_1}} \int_{-C_{s_2}}^{C_{s_2}} b_{s_2} b_{s_1}^2 e_{\boldsymbol{k}} \, \mathrm{d}b_{s_2} \, \mathrm{d}b_{s_1}$$
$$+ 3\sqrt{2} \frac{1}{(4C_{s_1} C_{s_2})^{\frac{1}{2}} 2C_{s_1+s_2}} \int_{-C_{s_1+s_2}}^{C_{s_1+s_2}} \int_{-C_{s_2}}^{C_{s_2}} \int_{-C_{s_1}}^{C_{s_1}} b_{j_2} b_{s_2} b_{s_1} e_{\boldsymbol{k}} \, \mathrm{d}b_{s_1} \, \mathrm{d}b_{s_2} \, \mathrm{d}b_{s_1+s_2}, \quad (2.35)$$

and when $s_2 \neq 2s_1$, $s_1 \neq 2s_2$ and $s_2 \neq s_1$,

$$e_{\boldsymbol{k}}(v) = \frac{1}{(4C_{s_2} C_{s_1})^{\frac{1}{2}}} \exp\left(\frac{\pi}{C_{s_2}} \mathrm{i} k_{s_2} b_{s_2} + \frac{\pi}{C_{s_1}} \mathrm{i} k_{s_1} b_{s_1}\right),$$
$$a_{\boldsymbol{k}}(f) = 3\sqrt{2} \frac{1}{(4C_{s_1} C_{s_2})^{\frac{1}{2}} 2C_{s_1+s_2}} \int_{-C_{s_1+s_2}}^{C_{s_1+s_2}} \int_{-C_{s_2}}^{C_{s_2}} \int_{-C_{s_1}}^{C_{s_1}} b_{j_2} b_{s_2} b_{s_1} e_{\boldsymbol{k}} \, \mathrm{d}b_{s_1} \, \mathrm{d}b_{s_2} \, \mathrm{d}b_{s_1+s_2}. \quad (2.36)$$

For any $\boldsymbol{k} \in \mathbb{K}$ with only three nonzero components, denoted by $k_{s_1}, k_{s_2}, k_{s_3}$, when $s_1 + s_2 = s_3$,

$$e_{\boldsymbol{k}}(v) = \frac{1}{(8C_{s_3} C_{s_2} C_{s_1})^{\frac{1}{2}}} \exp\left(\frac{\pi}{C_{s_3}} \mathrm{i} k_{s_3} b_{s_3} + \frac{\pi}{C_{s_2}} \mathrm{i} k_{s_2} b_{s_2} + \frac{\pi}{C_{s_1}} \mathrm{i} k_{s_1} b_{s_1}\right),$$
$$a_{\boldsymbol{k}}(f) = 3\sqrt{2} \frac{1}{(8C_{s_1} C_{s_2} C_{s_3})^{\frac{1}{2}}} \int_{-C_{s_3}}^{C_{s_3}} \int_{-C_{s_2}}^{C_{s_2}} \int_{-C_{s_1}}^{C_{s_1}} b_{s_3} b_{s_2} b_{s_1} e_{\boldsymbol{k}} \, \mathrm{d}b_{s_1} \, \mathrm{d}b_{s_2} \, \mathrm{d}b_{s_3}. \quad (2.37)$$

Since our method works similarly for the functional $f(v) = \int_0^1 v^q \, \mathrm{d}x, q \in \mathbb{Z}$, it can be applied to more general cases, e.g., $g(v) = \int_0^1 \sin v \, \mathrm{d}x$ by using Taylor expansions

$$\int_0^1 \sin v \, \mathrm{d}x = \sum_{n=0}^{\infty} \int_0^1 \frac{(-1)^n v(x)^{2n+1}}{(2n+1)!} \, \mathrm{d}x, \quad |v| < 1. \quad (2.38)$$

For this case, Assumption 2.1 is also satisfied. This shows that our method can be applied to more general functionals by using Taylor expansions together with Assumption 2.1.

Note that Assumption 2.1 requires that the functional can be divided into countable finite-dimensional parts. There are examples of functionals that do not satisfy this assumption, e.g., the functional f defined by

$$f\left(\sum_{j=1}^{\infty} b_j\Phi_j\right) := \prod_{j=1}^{\infty} g_j(b_j),$$

where $\{g_j \in C(\Omega) : \mathbb{R} \to \mathbb{R}\}_{j=1}^{\infty}$. In this case, f does not satisfy Assumption 2.1 because it cannot be written as a countable summation of finite-dimensional parts directly. (Fortunately, for this functional f, Assumption 2.1 can be satisfied after taking logarithm.) Moreover, in some cases, it may not be straightforward to examine whether this assumption can be satisfied by the functionals. Further generalizations beyond this assumption will be explored in the future work.

3 Approximation of Functionals Based on Barron Spectral Space

Based on the Barron spectral space for functionals defined above, we prove the approximation of functionals by neural networks without curse of dimensionality. This is the main result of this paper and is summarized in the following theorem, whose proof is given in Section 5.1.

Theorem 3.1. *For any functional $f \in \mathcal{B}[L_{bound}(\Omega)]$, there is $f_m \in \mathcal{G}_{ReLU,m,f}$, where*

$$\mathcal{G}_{ReLU,m,f} := \left\{ g(v) = c + \sum_{j=1}^{m} \gamma_j ReLU\left(\sum_{i\in\mathbb{S}_{\boldsymbol{k}_j}} w_{ij}b_i - t_j\right) : v = \sum_{i=1}^{\infty} b_i\Phi_i, \boldsymbol{k}_j \in \mathbb{K}_f \right.$$
$$\left. |c| \le 2\|f\|_{\mathcal{B}}, \sum_{j=1}^{m} |\gamma_j| \le 4\|f\|_{\mathcal{B}}, |\boldsymbol{w}_j|_1 = 1, |t| \le 1 \right\} \tag{3.1}$$

with

$$\mathbb{S}_{\boldsymbol{k}_j} := \left\{ i \in \mathbb{N} : k_{ij} \neq 0,\ \boldsymbol{k}_j = (k_{1j}, k_{2j}, \cdots, k_{ij}, \cdots) \in \mathbb{K}_f \right\},$$
$$\mathbb{K}_f := \{\boldsymbol{k} \in \mathbb{K} : a_{\boldsymbol{k}}(f) \neq 0\},$$

that satisfies

$$\|f - f_m\|_{\mathcal{H}} \le \frac{4\sqrt{5}\|f\|_{\mathcal{B}}}{\sqrt{m}}. \tag{3.2}$$

Note that there is no dependence on the dimensionality in the error estimate in this theorem, which overcomes the curse of the dimensionality. In fact, Theorem 3.1 states that the functional f in $\mathcal{B}[L_{\text{bound}}(\Omega)]$ can be approximated by a two layer network with m nodes in the hidden layer with $\mathcal{O}(1/\sqrt{m})$ accuracy in $\mathcal{H}[L_{\text{bound}}(\Omega)]$. For the j-th node in the hidden layer, the information of f associated with its j-th chosen Fourier basis functional is used, whose dimension is $|\mathbb{S}_{\boldsymbol{k}_j}|$, i.e., the number of nonzero components of $\boldsymbol{k}_j$, and is bounded by the number of elements in the block where the Fourier basis functional lies in by Assumption 2.1. As a result, only information of finite dimensions of f is needed in this $\mathcal{O}(1/\sqrt{m})$ approximation. Note that $|\mathbb{S}_{k_j}|$ depends only on f. Theorem 3.1 in fact provides a way to approximate a functional in an infinite dimensional space by a neural network with finite number of parameters, i.e., $\mathcal{O}(mN_{f,m})$, where $N_{f,m} = \max_{1\le j\le m} |\mathbb{S}_{k_j}|$.

For the linear functionals in Example 2.1 and the gradient energy functional in Example 2.2, we have $\max_{k\in\mathbb{K}_f}\{|\mathbb{S}_k|\} = 1$, and the $\mathcal{O}(1/\sqrt{m})$ neural network approximation given by Theorem 3.1 contains only $3m+1$ parameters. In these cases, the approximating neural network can be made more efficient and accurate; see Section 6.1(i). For other functionals given in the previous section, $\max_{k\in\mathbb{K}_f}\{|\mathbb{S}_k|\} = 1$ in Example 2.4, $\max_{k\in\mathbb{K}_f}\{|\mathbb{S}_k|\} = 2$ in Examples 2.3 and 2.5, and $\max_{k\in\mathbb{K}_f}\{|\mathbb{S}_k|\} = 3$ for Example 2.6. In general, if there is an upper bound for $\max_{k\in\mathbb{K}_f}\{|\mathbb{S}_k|\}$, then the number of parameters in the neural network given in Theorem 3.1 is $\mathcal{O}(m)$. Neural network approximation with reduced number of parameters will be given in Theorem 4.1 and will be further discussed in Section 6.1(ii).

In the DeepONet method as reviewed in the introduction section, the domain of the input functions is first discretized to reduce the infinite dimensional problem into a finite dimensional one and then the functional is learned by a neural network, and such process is suffered from curse of dimensionality (cf. the last two errors in Eq. (A.5)). In contrast, our Theorem 3.1 approximates the functional directly by a neural network without discretization of the domain of the input functions, and there is no curse of dimensionality. Moreover, since our method does not approximate the domain of the input functions, the number of parameters and the network structure in our method only depends on the functional, and it is not sensitive to the input functions in training.

In practice, when we set up the neural network, we do not know $\mathbb{S}_{k_j}$, $j = 1, 2, \ldots, m$, because we do not know f yet. This can be solved by choosing a large N and defining

$$f_m\left(\sum_{i=1}^{\infty} b_i\Phi_i\right) = c + \sum_{j=1}^{m}\gamma_j \mathrm{ReLU}\left(\sum_{i=1}^{N} w_{ij}b_i - t_j\right). \tag{3.3}$$

When $N \ge N_{f,m}$, this form of f_m includes all the functionals in $\mathcal{G}_{\mathrm{ReLU},m,f}$ including the desired one with error of $\mathcal{O}(1/\sqrt{m})$ given in Theorems 3.1, thus it is able to approximate f with $\mathcal{O}(1/\sqrt{m})$ accuracy. In this case, the number of parameters in the network is $\mathcal{O}(mN)$, instead of exponential dependence on the dimension N; and once $N \ge N_{f,m}$, the desired approximation of $\mathcal{O}(1/\sqrt{m})$ accuracy given in Theorem 3.1 is included by f_m, and no further increase of N is needed. Note that $N_{f,m}$ and accordingly N depend only on the functional f and do not depend on the input functions. There is still no curse of dimensionality by using this treatment in practice. The input data for the training of the network will be the finite Fourier coefficients $\{b_i\}_{i=1}^{N}$ of the input functions.

4 Reducing the Number of Parameters in $\mathcal{G}_{\mathbf{ReLU,m,f}}$

Theorem 3.1 shows that a functional $f \in \mathcal{B}[L_{\mathrm{bound}}(\Omega)]$ can be approximated well from the set $\mathcal{G}_{\text{ReLU,m,f}}$ without curse of dimensionality. A challenge when using this theorem in deep learning is that the set $S_{\boldsymbol{k}_j}$ for each j associated with f that appear in $\mathcal{G}_{\text{ReLU,m,f}}$ are unknown because f is a functional to be learnt.

Recall that when the Fourier basis is used, a smoother function has faster decaying Fourier coefficients. Here we consider a generalization of this property for any basis $\{\Phi_i\}$, so that the number of parameters for $i \in \mathbb{S}_{\boldsymbol{k}_j}$ can be replaced by a fixed number N based on the domain of the functional.

We restrict the domain of the functional to

$$L_{\text{cut}}(\Omega) := \left\{ v \in L_{\text{bound}}(\Omega) : v = \sum_{i=1}^{\infty} b_i \Phi_i,\ |b_i| < \delta \text{ for } i > N \right\}, \tag{4.1}$$

where δ is a small constant and $N \in \mathbb{N}_+$. Based on $L_{\text{cut}}(\Omega)$, we modify the Fourier coefficients in Definition 2.3 as

$$a_{\boldsymbol{k}}(f) := \sum_{j \in \mathbb{A}_{\boldsymbol{k}}} \frac{1}{(2\delta)^{|\mathbb{D}_j \cap [N+1,+\infty)|/2}} \int_{(-\delta,\delta)^{|\mathbb{D}_j \cap [N+1,+\infty)|} \times \left(-\frac{1}{2},\frac{1}{2}\right)^{|\mathbb{D}_j \cap [0,N]|}} f_j\left(\sum_{i \in \mathbb{D}_j} b_i \Phi_i \right)$$
$$\times \left[\prod_{i \in \mathbb{D}_j \cap [0,N]} \exp(-2\pi \mathrm{i} k_i b_i)\, \mathrm{d}b_i \prod_{i \in \mathbb{D}_j \cap [N+1,+\infty)} \exp\left(-\frac{1}{\delta} \pi \mathrm{i} k_i b_i \right) \mathrm{d}b_i \right]. \tag{4.2}$$

With the space $L_{\text{cut}}(\Omega)$, we can use the parameter N in $L_{\text{cut}}(\Omega)$ instead of the unknown number of parameters for $i \in \mathbb{S}_{\boldsymbol{k}_j}$ of f in the approximation by neural network. The result is summarized in the following theorem.

Theorem 4.1. *For any $f_m \in \mathcal{G}_{ReLU,m,f}$ that is given by*

$$f_m\left(\sum_{i=1}^{\infty} b_i \Phi_i \right) = c + \sum_{j=1}^{m} \gamma_j ReLU\left(\sum_{i \in \mathbb{S}_{\boldsymbol{k}_j}} w_{ij} b_i - t_j \right), \tag{4.3}$$

there are f_m^*, f_m^{**} from

$$\mathcal{G}^*_{ReLU,m,f} := \left\{ g\left(\sum_{i=1}^{\infty} b_i \Phi_i \right) = c + \sum_{j=1}^{m} \gamma_j ReLU\left(\sum_{i=1}^{N} w_{ij} b_i - t_j \right) : 2|c|, \right.$$
$$\left. \sum_{j=1}^{m} |\gamma_j| \leq 4\|f\|_{\mathcal{B}},\ |\boldsymbol{w}_j|_1 = 1,\ |t| \leq 1 \right\}, \tag{4.4}$$

such that for any $v \in L_{cut}(\Omega)$,

$$\left| f_m^*(v) - f_m(v) \right| \leq 4\|f\|_{\mathcal{B}} \delta, \tag{4.5}$$

$$\|f_m^{**} - f_m\|_{\mathcal{H}} \leq 2\sqrt{13}\|f\|_{\mathcal{B}} (2\delta)^{\frac{1}{2}}. \tag{4.6}$$

Here the constant N in $\mathcal{G}^*_{\text{ReLU,m,f}}$ in (4.4) is that in the definition of $L_{\text{cut}}(\Omega)$.

Theorem 4.1 combined with Theorem 3.1 show that there exists a neural network in $G^*_{ReLU,m,f}$ that approximates f well on the domain $L_{\text{cut}}(\Omega)$, and the error is

$$\|f_m^{**} - f\|_{\mathcal{H}} \leq \left(\frac{4\sqrt{5}}{\sqrt{m}} + 2\sqrt{13}(2\delta)^{\frac{1}{2}}\right)\|f\|_{\mathcal{B}}. \tag{4.7}$$

For example, if we want to learn the gradient energy in Example 2.2, where the functional is

$$E(u) = \int_0^1 \frac{1}{2}\alpha\left(\frac{\mathrm{d}u}{\mathrm{d}x}\right)^2 \mathrm{d}x,$$

we may restrict the domain of the functional to

$$\left\{v = \sum_{n=0}^{\infty} b_n\Phi_n : -Cn^{-\frac{3}{2}-\varepsilon} < b_n < Cn^{-3/2-\varepsilon},\ n \in \mathbb{N}\right\},$$

where $C, \varepsilon > 0$ and $\{\Phi_n = \sqrt{2}\cos(\pi n x)\}_{n=1}^{\infty} \cup \{\Phi_0 = 1\}$. In this case, we can select $\delta = CN^{-3/2-\varepsilon}$ for a large N in $L_{\text{cut}}(\Omega)$. Combining Theorems 3.1 and 4.1, the approximation error of $\mathcal{G}^*_{\text{ReLU,m,f}}$ in this case is $\mathcal{O}(1/\sqrt{m}) + \mathcal{O}(N^{-3/4-\varepsilon/2})$. For the second error, we can obtain a smaller error for a fixed N, when we consider a functional on a space with smoother functions, such as $H^k(0,1)$. For $H^k(0,1)$, we should consider the space

$$\left\{v = \sum_{n=0}^{\infty} b_n\Phi_n : -Cn^{-\frac{(1+k)}{2}-\varepsilon} < b_n < Cn^{-\frac{(1+k)}{2}-\varepsilon},\ n \in \mathbb{N}\right\},$$

where $C, \varepsilon > 0$, and the second error is $\mathcal{O}(N^{-(1+k)/4-\varepsilon/2})$. With the same level of the error, a space with smoother functions will lead to fewer parameters in the neural network.

5 Proofs of Theorems 3.1 and 4.1

5.1 Proofs of Theorem 3.1 and related propositions and lemmas

We first sketch the main steps in the proof of Theorem 3.1 and then give the full proof.

Main steps of the proof of Theorem 3.1:

(i) Show that for any $f \in \mathcal{B}[L_{\text{bound}}(\Omega)]$, $f - a_{\mathbf{0}}(f)$ is in the closure of the convex hull of a set $\mathcal{G}_{\mathbb{K}_f\backslash\{\mathbf{0}\},f}$ in Hilbert space $\mathcal{H}[L_{\text{bound}}(\Omega)]$, where

$$\mathcal{G}_{\mathbb{K}_f\backslash\{\mathbf{0}\},f} := \left\{g\left(\sum_{i=1}^{\infty} b_i\Phi_i\right) = \frac{\gamma}{1+(2\pi)^2|\boldsymbol{k}|_1^2}\cos\left[2\pi\left(\sum_{i\in S_{\boldsymbol{k}_j}} k_i b_i + b\right)\right]\right.$$
$$\left. : |\gamma| \leq \|f\|_{\mathcal{B}},\ b \in [0,1],\ \boldsymbol{k} \in \mathbb{K}_f\backslash\{\mathbf{0}\}\right\}. \tag{5.1}$$

DOI https://doi.org/10.4208/jml.221018 | Generated on 2024-12-19 04:40:52

(ii) Show that each element in $\bar{c} + \mathcal{G}_{\mathbb{K}_f \backslash \{\mathbf{0}\}, f}$, where $|\bar{c}| \leq \|f\|_{\mathcal{B}}$, is in the closure of the convex hull of a set $\mathcal{G}_{\mathrm{ReLU},f}$ in Hilbert space $\mathcal{H}[L_{\mathrm{bound}}(\Omega)]$, where

$$\mathcal{G}_{\mathrm{ReLU},f} := \Bigg\{ g\left(\sum_{i=1}^{\infty} b_i \Phi_i\right) = c + \gamma \mathrm{ReLU}\Bigg(\sum_{i \in \mathbb{S}_{\boldsymbol{k}_j}} w_i b_i - t\Bigg) : 2|c|, |\gamma| \leq 4\|f\|_{\mathcal{B}}, \\ |\boldsymbol{w}|_1 = 1,\ |t| \leq 1,\ \boldsymbol{k} \in \mathbb{K}_f \backslash \{\mathbf{0}\} \Bigg\}. \tag{5.2}$$

(iii) Then using Lemma 5.1 below, we can show that the convex combination of elements in $\mathcal{G}_{\mathrm{ReLU},f}$ can approximate $f(v)$ with the convergence rate $\mathcal{O}(1/\sqrt{m})$, where m is the number of elements in the convex combination.

Lemma 5.1 ([2, 31])**.** *Let h belongs to the closure of the convex hull of a set $\mathcal{G}$ in Hilbert space. Denote the Hilbert norm as $\|\cdot\|$. If each element of $\mathcal{G}$ be upper bounded by $B > 0$ then for every $m \in \mathbb{N}$, there are $\{h_i\}_{i=1}^m \subset \mathcal{G}$ and $\{c_i\}_{i=1}^m \subset [0, 1]$ with $\sum_{i=1}^m c_i = 1$, such that*

$$\left\| h - \sum_{i=1}^{m} c_i h_i \right\|^2 \leq \frac{B^2}{m}. \tag{5.3}$$

Proposition 5.1. $\mathcal{B}[L_{bound}(\Omega)] \hookrightarrow \mathcal{H}[L_{bound}(\Omega)]$.

Proof. For any functional f in $\mathcal{B}[L_{\mathrm{bound}}(\Omega)]$, we have that for any $\boldsymbol{k} \in \mathbb{K}$

$$|a_{\boldsymbol{k}}(f)| \leq \sum_{\boldsymbol{k} \in \mathbb{K}} |a_{\boldsymbol{k}}(f)| \leq \|f\|_{\mathcal{B}}.$$

Hence, we have

$$\begin{aligned} \|f\|_{\mathcal{H}}^2 &= \sum_{\boldsymbol{k} \in \mathbb{K}} \left(1 + (2\pi)^2 |\boldsymbol{k}|^2\right) |a_{\boldsymbol{k}}(f)|^2 \\ &\leq \sum_{\boldsymbol{k} \in \mathbb{K}} \left(1 + (2\pi)^2 |\boldsymbol{k}|^2\right) |a_{\boldsymbol{k}}(f)| \cdot \|f\|_{\mathcal{B}} \leq \|f\|_{\mathcal{B}}^2. \end{aligned} \tag{5.4}$$

The proof is complete. □

Based on Proposition 5.1, we can prove the following lemma.

Lemma 5.2. *For any functional $f \in \mathcal{B}[L_{bound}(\Omega)]$, $f - a_{\mathbf{0}}(f)$ is in the closure of the convex hull of the set $\mathcal{G}_{\mathbb{K}_f \backslash \{\mathbf{0}\}, f}$ in the Hilbert space $\mathcal{H}[L_{bound}(\Omega)]$.*

Proof. Since $f \in \mathcal{B}[L_{\mathrm{bound}}(\Omega)]$, we have

$$f(v) = \sum_{\boldsymbol{k} \in \mathbb{K}} a_{\boldsymbol{k}}(f) e_{\boldsymbol{k}}(v) = \sum_{\boldsymbol{k} \in \mathbb{K}_f} a_{\boldsymbol{k}}(f) e_{\boldsymbol{k}}(v)$$

for any $v \in L_{\mathrm{bound}}(\Omega)$. Denote $v = \sum_{i=1}^{\infty} b_i \Phi_i$, and

$$\mathbb{S}_{\boldsymbol{k}_j} := \left\{ i : \boldsymbol{k}_j := (k_{1j}, k_{2j}, \cdots), k_{ij} \neq 0, \boldsymbol{k} \in \mathbb{K}_f \right\}.$$

We obtain

$$
\begin{aligned}
& f(v)-a_{\mathbf{0}}(f) \\
= & \sum_{\boldsymbol{k}\in\mathbb{K}_f\setminus\{\mathbf{0}\}} a_{\boldsymbol{k}}(f)e_{\boldsymbol{k}}(v) = \sum_{\boldsymbol{k}\in\mathbb{K}_f\setminus\{\mathbf{0}\}} a_{\boldsymbol{k}}(f)\prod_{i=1}^{\infty}\exp(2\pi\mathrm{i}k_i b_i) \\
= & \sum_{\boldsymbol{k}\in\mathbb{K}_f\setminus\{\mathbf{0}\}} a_{\boldsymbol{k}}(f)\prod_{i\in\mathbb{S}_{\boldsymbol{k}_j}}\exp(2\pi\mathrm{i}k_i b_i) \\
= & \sum_{\boldsymbol{k}\in\mathbb{K}_f\setminus\{\mathbf{0}\}} a_{\boldsymbol{k}}(f)\exp\left(2\pi\mathrm{i}\sum_{i\in\mathbb{S}_{\boldsymbol{k}_j}} k_i b_i\right) \\
= & \sum_{\boldsymbol{k}\in\mathbb{K}_f\setminus\{\mathbf{0}\}} |a_{\boldsymbol{k}}(f)|\exp\left[2\pi\mathrm{i}\left(\sum_{i\in\mathbb{S}_{\boldsymbol{k}_j}} k_i b_i+\theta_k(f)\right)\right] \\
= & \sum_{\boldsymbol{k}\in\mathbb{K}_f\setminus\{\mathbf{0}\}} \frac{|a_{\boldsymbol{k}}(f)|(1+(2\pi)^2|\boldsymbol{k}|_1^2)}{Z_f}\frac{Z_f}{1+(2\pi)^2|\boldsymbol{k}|_1^2}\cos\left[2\pi\left(\sum_{i\in\mathbb{S}_{\boldsymbol{k}_j}} k_i b_i+\theta_k(f)\right)\right],
\end{aligned}
\tag{5.5}
$$

where

$$
\theta_{\boldsymbol{k}}(f)=\frac{1}{2\pi}\arg a_{\boldsymbol{k}}(f)=\frac{1}{2\pi}\tan^{-1}\frac{\Im a_{\boldsymbol{k}}(f)}{\Re a_{\boldsymbol{k}}(f)},
$$

and

$$
Z_f := \sum_{\boldsymbol{k}\in\mathbb{K}_f\setminus\{\mathbf{0}\}} \left|a_{\boldsymbol{k}}(f)\right|\left(1+(2\pi)^2|\boldsymbol{k}|_1^2\right) \le \|f\|_{\mathcal{B}}. \tag{5.6}
$$

The last equality is due to the fact that $f(v)-a_{\mathbf{0}}(f)$ is a real functional.

Denote

$$
\begin{aligned}
g(v,\boldsymbol{k}) &:= \frac{Z_f}{1+(2\pi)^2|\boldsymbol{k}|_1^2}\cos\left[2\pi\left(\sum_{i\in\mathbb{S}_{\boldsymbol{k}_j}} k_i b_i+\theta(f)\right)\right] \\
&= \frac{Z_f}{1+(2\pi)^2|\boldsymbol{k}|_1^2}\frac{1}{2}\left(e^{2\pi\mathrm{i}\theta_k(f)}e_{\boldsymbol{k}}(v)+e^{-2\pi\mathrm{i}\theta_k(f)}e_{-\boldsymbol{k}}(v)\right).
\end{aligned}
\tag{5.7}
$$

Therefore,

$$
f(v)-a_{\mathbf{0}}(f) = \sum_{\boldsymbol{k}\in\mathbb{K}_f\setminus\{\mathbf{0}\}} \frac{|a_{\boldsymbol{k}}(f)|\left(1+(2\pi)^2|\boldsymbol{k}|_1^2\right)}{Z_f} g(v,\boldsymbol{k}), \tag{5.8}
$$

$$
\|g(v,\boldsymbol{k})\|_{\mathcal{H}} = Z_f\sqrt{\frac{1}{2+8\pi^2|\boldsymbol{k}|_1^2}} \le Z_f \le \|f\|_{\mathcal{B}}. \tag{5.9}
$$

Due to Eq. (5.7), we know $g(v,\boldsymbol{k}) \in \mathcal{G}_{\mathbb{K}_f\setminus\{\mathbf{0}\},f}$.

Now based on Eq. (5.8), we will prove that $f(v)-a_{\mathbf{0}}(f)$ is in the closure of the convex hull of the set $\mathcal{G}_{\mathbb{K}_f\setminus\{\mathbf{0}\},f}$ in Hilbert space $\mathcal{H}[L_{\mathrm{bound}}(\Omega)]$. First of all, $f-a_{\mathbf{0}}(f) \in \mathcal{B}[L_{\mathrm{bound}}(\Omega)] \subset \mathcal{H}[L_{\mathrm{bound}}(\Omega)]$ due to Proposition 5.1.

Define a random variable $\boldsymbol{k}^*$:

$$P(\boldsymbol{k}^* = \boldsymbol{k}) = \frac{|a_{\boldsymbol{k}}(f)|(1+(2\pi)^2|\boldsymbol{k}|_1^2)}{Z_f}, \quad \boldsymbol{k} \in \mathbb{K}_f \backslash \{\mathbf{0}\}. \tag{5.10}$$

Hence, $\mathbf{E}[g(v,\boldsymbol{k}^*)] = f(v) - a_{\mathbf{0}}(f)$ due to Eq. (5.8). For any integer m, let $\{\boldsymbol{k}_j^*\}_{j=1}^m$ be an independent identically distributed random variable sequence with the same distribution as $\boldsymbol{k}^*$. From Eq. (5.9), we know that

$$\begin{aligned} &\mathbf{E}\left\| f(v) - a_{\mathbf{0}}(f) - \frac{1}{m}\sum_{j=1}^m g(v,\boldsymbol{k}_j^*)\right\|_{\mathcal{H}}^2 \\ &= \frac{\mathrm{Var}\left[g(v,\boldsymbol{k}^*)\right]}{m} \le \frac{\mathbf{E}\|g(v,\boldsymbol{k}^*)\|_{\mathcal{H}}^2}{m} \le \frac{\|f\|_{\mathcal{B}}^2}{m}, \end{aligned} \tag{5.11}$$

where the variance is defined in the Hilbert space $\mathcal{H}[L_{\text{bound}}(\Omega)]$.

Therefore, by the pigeonhole principle, there exist $\{\boldsymbol{k}_j^{**}\}_{j=1}^m \subset \mathbb{K}_f \backslash \{\mathbf{0}\}$, such that

$$\left\| f(v) - a_{\mathbf{0}}(f) - \frac{1}{m}\sum_{j=1}^m g(v,\boldsymbol{k}_j^{**})\right\|_{\mathcal{H}}^2 \le \frac{\|f\|_{\mathcal{B}}^2}{m}. \tag{5.12}$$

Here $(\sum_{j=1}^m g(v,\boldsymbol{k}_j^{**}))/m$ is a convex combination of elements in $\mathcal{G}_{\mathbb{K}_f \backslash \{\mathbf{0}\},f}$. As $m \to +\infty$, such obtained convex combinations converge to $f(v) - a_{\mathbf{0}}(f)$ in $\mathcal{H}[L_{\text{bound}}(\Omega)]$. Hence, $f(v) - a_{\mathbf{0}}(f)$ is in the closure of the convex hull of the set $\mathcal{G}_{\mathbb{K}_f \backslash \{\mathbf{0}\},f}$ in the Hilbert space $\mathcal{H}[L_{\text{bound}}(\Omega)]$. □

Lemma 5.3. *Each element in $\bar{c} + \mathcal{G}_{\mathbb{K}_f \backslash \{\mathbf{0}\},f}$, where $|\bar{c}| \le \|f\|_{\mathcal{B}}$, is in the closure of the convex hull of the set $\mathcal{G}_{ReLU,f}$ defined in Eq.* (5.2).

Proof. The conclusion comes from the fact that a convex linear combination of ReLU functions can approximate the cosine functions well, whose proof can be found in [27, Proposition 19]. In fact, they showed that each cosine function in $\mathcal{G}_{\mathbb{K}_f \backslash \{\mathbf{0}\},f}$ is in the convex hull of

$$\begin{aligned} \mathcal{G}_{\text{ReLU},f} := \Bigg\{ g\left(\sum_{i=1}^\infty b_i\Phi_i\right) = c + \gamma\text{ReLU}\left(\sum_{i\in \mathbb{S}_{\boldsymbol{k}_j}} w_i b_i - t\right) : 4|c|, |\gamma| \le 4\|f\|_{\mathcal{B}}, \\ |\boldsymbol{w}|_1 = 1,\ |t| \le 1,\ \boldsymbol{k} \in \mathbb{K}_f \Bigg\}. \end{aligned} \tag{5.13}$$

The proof is complete. □

Proposition 5.2. *Suppose that $f \in \mathcal{H}[L_{bound}(\Omega)]$ is a finite dimensional functional based on some $\mathbb{D}_j$, where recall that $|\mathbb{D}_j| < \infty$, i.e.,*

$$f\left(\sum_{i=1}^\infty b_i\Phi_i\right) = f\left(\sum_{i\in\mathbb{D}_j} b_i\Phi_i\right) \tag{5.14}$$

for all $-1/2 < b_i < 1/2$. *We have*

$$\|f\|_{\mathcal{H}} = \|g_f\|_{H^1\left((-\frac{1}{2},\frac{1}{2})^{c_j}\right)}, \tag{5.15}$$

where $c_j := |\mathbb{D}_j| < \infty$, *and* g_f *is a* c_j *dimensional function defined by*

$$g_f(b_{n_1}, b_{n_2}, \cdots, b_{n_{c_j}}) := f\left(\sum_{i=1}^{c_j} b_{n_i}\Phi_{n_i}\right), \quad n_i \in \mathbb{D}_j. \tag{5.16}$$

Here $H^1((-1/2,1/2)^{c_j})$ *is the* H^1*-Sobolev space on* $(-1/2,1/2)^{c_j}$.

Proof. Due to Eq. (2.17), we have

$$\|f\|_{\mathcal{H}}^2 = \sum_{\boldsymbol{k}\in\mathbb{K}} \left(1 + (2\pi)^2|\boldsymbol{k}|^2\right)\left|a_{\boldsymbol{k}}(f)\right|^2.$$

For $\boldsymbol{k} \notin \mathbb{K}_{\mathbb{D}_j}$ we have

$$a_{\boldsymbol{k}}(f) = 0. \tag{5.17}$$

Hence, we have

$$\|f\|_{\mathcal{H}}^2 = \sum_{\boldsymbol{k}\in\mathbb{K}_{\mathbb{D}_j}} \left(1 + (2\pi)^2|\boldsymbol{k}|^2\right)\left|a_{\boldsymbol{k}}(f)\right|^2. \tag{5.18}$$

Furthermore,

$$\|g_f\|^2_{H^1\left((-\frac{1}{2},\frac{1}{2})^{c_j}\right)} = \sum_{\boldsymbol{k}\in\mathbb{K}_{\mathbb{D}_j}} (1 + (2\pi)^2|\boldsymbol{k}|^2)|a_{\boldsymbol{k}}(f)|^2$$

by direct calculations. □

Combining Proposition 5.2 and Lemmas 5.1-5.3, we can prove Theorem 3.1.

Proof of Theorem 3.1. Due to Proposition 5.2, for each element $g \in \mathcal{G}_{\mathrm{ReLU},f}$, i.e.,

$$g\left(\sum_{i=1}^{\infty} b_i\Phi_i\right) = c + \gamma\mathrm{ReLU}\left(\sum_{i\in\mathbb{S}_{\boldsymbol{k}_j}} w_i b_i - t\right),$$

we have

$$\|g\|_{\mathcal{H}}^2 = \left\|c + \gamma\mathrm{ReLU}\left(\sum_{i\in\mathbb{S}_{\boldsymbol{k}_j}} w_i b_i - t\right)\right\|^2_{H^1((-\frac{1}{2},\frac{1}{2})^{|\mathbb{S}_{\boldsymbol{k}_j}|})}, \tag{5.19}$$

where the norm $\|\cdot\|_{H^1((-1/2,1/2)^{|\mathbb{S}_{\boldsymbol{k}_j}|})}$ is the H^1-Sobolev space norm for functions of $b_1, b_2, \cdots, b_{N_{\boldsymbol{k}}}$. By direct calculations, we obtain that

$$\left\|c + \gamma\mathrm{ReLU}\left(\sum_{i\in\mathbb{S}_{\boldsymbol{k}_j}} w_i a_i - t\right)\right\|^2_{H^1((-\frac{1}{2},\frac{1}{2})^{|\mathbb{S}_{\boldsymbol{k}_j}|})} \le \left(c + \frac{3}{2}\gamma\right)^2 + \gamma^2 \le 80\|f\|_{\mathcal{B}}^2. \tag{5.20}$$

Therefore, $\mathcal{G}_{\mathrm{ReLU},f}$ is bounded by $4\sqrt{5}\|f\|_{\mathcal{B}}$. Due to Lemmas 5.2-5.3, f is in the closure of the convex hull of the set $\mathcal{G}_{\mathrm{ReLU},f}$. Due to Lemma 5.1, we finish the proof. □

5.2 Proof of Theorem 4.1

Proof. Consider any $v = \sum_{i=1}^{\infty} b_i \Phi_i \in L_{\text{cut}}(\Omega)$. Without loss of generality, suppose $N_f \geq N$ for all $\boldsymbol{k}_j$ in f_m. We have

$$\begin{aligned} f_m\left(\sum_{i=1}^{\infty} b_i\Phi_i\right) &= c + \sum_{j=1}^{m} \gamma_j \text{ReLU}\left(\sum_{i=1}^{N_f} w_{ij} b_i - t_j\right) \\ &= c + \sum_{j=1}^{m} \gamma_j \text{ReLU}\left(\sum_{i=1}^{N} w_{ij} b_i - t_j + \sum_{i=N+1}^{N_f} w_{ij} b_i\right) \\ &:= c + \sum_{j=1}^{m} \gamma_j \text{ReLU}\left(\sum_{i=1}^{N} w_{ij} b_i - t_j + \varepsilon_j\right), \end{aligned} \tag{5.21}$$

where

$$|\varepsilon_j| = \left|\sum_{i=N+1}^{N_f} w_{ij} b_i\right| \leq |\boldsymbol{\omega}_j|_1 \delta = \delta. \tag{5.22}$$

Define

$$f_m^*\left(\sum_{i=1}^{\infty} b_i\Phi_i\right) = c + \sum_{j=1}^{m} \gamma_j \text{ReLU}\left(\sum_{i=1}^{N} w_{ij} b_i - t_j\right) \in \mathcal{G}^*_{\text{ReLU,m,f}}. \tag{5.23}$$

Since ReLU is a Lipschitz continuous function in $\mathbb{R}$ with Lipschitz constant 1 and

$$\sum_{j=1}^{m} |\gamma_j| \leq 4\|f\|_{\mathcal{B}},$$

we have

$$\left| f_m^*\left(\sum_{i=1}^{\infty} b_i\Phi_i\right) - f_m\left(\sum_{i=1}^{\infty} b_i\Phi_i\right)\right| \leq 4\|f\|_{\mathcal{B}}\delta. \tag{5.24}$$

Thus the first inequality in Theorem 4.1 is proved.

For the second inequality in Theorem 4.1, without loss of generality, we suppose $w_{ij} \neq 0$ for all w_{ij} in f_m. Denote

$$f_m^{**}\left(\sum_{i=1}^{\infty} b_i\Phi_i\right) = c + \sum_{j\in\mathbb{S}} \gamma_j \text{ReLU}\left(\sum_{i=1}^{N_f} w_{ij} b_i - t_j\right) \in \mathcal{G}^*_{\text{ReLU,m,f}}, \tag{5.25}$$

$$\bar{f}_m\left(\sum_{i=1}^{\infty} b_i\Phi_i\right) = \sum_{j\notin\mathbb{S}} \gamma_j \text{ReLU}\left(\sum_{i=1}^{N_f} w_{ij} b_i - t_j\right),$$

where $\mathbb{S} := \{j : N_f \leq N\}$. Therefore,

$$\|f_m^{**} - f_m\|_{\mathcal{H}} = \|\bar{f}_m\|_{\mathcal{H}} \leq \sum_{j\notin\mathbb{S}} \left\| \gamma_j \text{ReLU}\left(\sum_{i=1}^{N_f} w_{ij} b_i - t_j\right)\right\|_{H^1((-\frac{1}{2},\frac{1}{2})^N \times (-\delta,\delta)^{N_f - N})}$$

DOI https://doi.org/10.4208/jml.221018 | Generated on 2024-12-19 04:40:52

$$\leq \sum_{j \notin \mathrm{S}} \sqrt{\left(\frac{9}{4}\gamma_j^2 + \gamma_j^2\right)(2\delta)^{N_f - N}} \leq \sum_{j \notin \mathrm{S}} \frac{\sqrt{13}}{2}|\gamma_j|(2\delta)^{\frac{N_f - N}{2}}$$
$$\leq 2\sqrt{13}\|f\|_{\mathcal{B}}(2\delta)^{\frac{1}{2}}. \tag{5.26}$$

Here the second inequality is due to Proposition 5.2 in Section 5.1 (with the redefined Fourier coefficients in Eq. (4.2)). The last inequality is due to the condition $\sum_{j=1}^{m}|\gamma_j| \leq 4\|f\|_{\mathcal{B}}$ in definition of $\mathcal{G}^*_{\mathrm{ReLU,m,f}}$. □

6 Applications of the Theorems on Approximation Functionals and Solving PDEs by Neural Networks

In this section, we further discuss how to apply the obtained theorems (Theorems 3.1 and 4.1) on the approximation of functionals by neural networks, including the application to solving PDEs by neural networks.

6.1 Applications of the theorems on approximation of functionals by neural networks

Here we discuss how to develop neural networks for the learning of functionals without curse of dimensionality based on the error estimates obtained in Theorems 3.1 and 4.1, in addition to that given in Section 3.

(i) **Theorems 3.1 for functionals with special structure:** Consider the case when the functional f in $\mathcal{B}[L_{\mathrm{bound}}(\Omega)]$ has the special structure with

$$\mathbb{D} = \big\{\{1\}, \{2\}, \cdots, \{n\}, \cdots\big\},$$

such as linear functionals and the gradient energy functionals as given in Examples 2.1 and 2.2. In this case, in $\mathcal{G}_{\mathrm{ReLU,m,f}}$ in Theorem 3.1, for any $\boldsymbol{k} \in \mathbb{K}_f = \{\boldsymbol{k} \in \mathbb{K} : a_{\boldsymbol{k}}(f) \neq 0\}$, it has only one nonzero component by the definition of $a_{\boldsymbol{k}}(f)$ in Eq. (2.7). Therefore, the $\mathcal{O}(1/\sqrt{m})$ approximation given by Theorem 3.1 in this case takes the following simple form that contains only $3m+1$ parameters:

$$f_m\left(\sum_{i=1}^{\infty} b_i \Phi_i\right) = c + \sum_{j=1}^{m} \gamma_j \mathrm{ReLU}\left(w_j b_{n_j} - t_j\right), \tag{6.1}$$

where n_j is the index of the only nonzero component of $a_{\boldsymbol{k}_j}(f)$.

However, such simplification cannot be employed directly in the training of the neural network. In fact, as discussed in (i), when we set up the neural network, we do not know the index n_j, $1 \leq j \leq m$, and in this case, we still need to choose an $N \geq n_j$, $1 \leq j \leq m$. Instead of using Eq. (3.3), we establish the neural network using the following formula:

$$f_m\left(\sum_{i=1}^{\infty} b_i \Phi_i\right) = c + \sum_{j=1}^{m}\sum_{i=1}^{N} \gamma_{ij} \mathrm{ReLU}\left(w_{ij} b_i - t_{ij}\right), \tag{6.2}$$

which includes the desired approximation in Eq. (6.1). Eq. (6.2) has the same order of number parameters, $\mathcal{O}(mN)$, as that of Eq. (3.3), and the approximation error of Eq. (6.2) can reach $\mathcal{O}(1/\sqrt{mN})$ by Theorem 3.1. This $\mathcal{O}(1/\sqrt{mN})$ error is improved significantly from the original approximation error in Eq. (6.1) and that in Eq. (3.3) for a more general class of functionals discussed in (i), both of which are $\mathcal{O}(1/\sqrt{m})$.

(ii) **For Theorem 4.1:** When we can reduce the functional to a finite dimensional domain such as $L_{\text{cut}}(\Omega)$, the parameter N in $L_{\text{cut}}(\Omega)$ is given and can be regarded as the dimension of $L_{\text{cut}}(\Omega)$. When we set up the neural network, the parameter N is directly given by $L_{\text{cut}}(\Omega)$, instead of the unknown N that has to be chosen large enough in the neural network setup based on Theorem 3.1 discussed above. In this case, we can also use f_m in Eq. (3.3) with this given parameter N. By Theorem 4.1, this f_m can approximate $f \in \mathcal{B}[L_{\text{cut}}(\Omega)]$ well in $\mathcal{H}[L_{\text{cut}}(\Omega)]$ with error $\mathcal{O}(1/\sqrt{m}) + \mathcal{O}(\sqrt{\delta})$, where δ is defined in $L_{\text{cut}}(\Omega)$. The number of parameters in Eq. (3.3) is $\mathcal{O}(mN)$. Therefore, our method does not suffer from curse of dimensionality in this case either.

6.2 Method for solving PDEs by neural networks based on the obtained theorems on approximation of functionals

We can apply Theorem 3.1 or 4.1 to build a neural network to solve a PDE problem at a given point. The follow brief discussion is based on the application of Theorem 3.1.

Consider following PDE boundary value problem:

$$\begin{cases} \mathcal{L}u = g & \text{in } K_1, \\ u = 0 & \text{on } \partial K_1, \end{cases} \tag{6.3}$$

where $K_1 = (0,1)^d$ and $g \in L_{\text{bound}}(L_2(K_1))$. Here a basis of K_1 is

$$\{\Phi_i\}_{i\in\mathbb{N}} = \left\{ \exp(2\pi \mathrm{i} \boldsymbol{p}\cdot\boldsymbol{x}) \right\}_{\boldsymbol{p}\in\mathbb{Z}^d}.$$

We want to find the solution at a point $\boldsymbol{x}_0 \in K_1$, i.e., $u(\boldsymbol{x}_0, g)$. This defines a functional whose input is $g \in L^2(K_1)$ and output is $u(\boldsymbol{x}_0, g) \in \mathbb{R}$, and we denote this functional as $\mathcal{L}^{-1}_{x_0}$. Assume that $\mathcal{L}^{-1}_{x_0} \in \mathcal{B}[L_{\text{bound}}(L_2(K_1))]$. Now we approximate $\mathcal{L}^{-1}_{x_0}$ by the elements in $\mathcal{G}_{\text{ReLU,m,f}}$.

Consider the given date set $\{g_s(\boldsymbol{x}), u_s(\boldsymbol{x}_0)\}_{s=1}^{M}$, where $u_s(\boldsymbol{x}_0) := u(x_0, g_s)$. We can calculate that

$$b_{si} = \int_{K_1} g_s(\boldsymbol{x})\Phi_i \,\mathrm{d}x.$$

The input data set for the neural network is $\{\{b_{si}\}_{i=1}^{N}, u_s(\boldsymbol{x}_0)\}_{s=1}^{M}$. We build a two layer network with m nodes in the hidden layer based on the following form, with some number N (aiming at $N \geq N_f$):

$$u_s^* = c + \sum_{j=1}^{m} \gamma_j \text{ReLU}\left(\sum_{i=1}^{N} w_{ij} b_{si} - t_j\right). \tag{6.4}$$

Here u_s^* is the output for the value $u_s(\boldsymbol{x}_0)$. We then learn the coefficients $\boldsymbol{\theta} := \{c, \gamma_j, w_{ij}, t_j\}$ by the Loss function

$$\mathcal{R}(\boldsymbol{\theta}) = \frac{1}{M}\sum_{s=1}^{M} |u_s^* - u_s(\boldsymbol{x}_0)|^2.$$

Here $\mathcal{R}(\boldsymbol{\theta})$ can be regarded as an approximation of the norm $\|f - f_m\|_{\mathcal{H}_0}^2$ ($\|f - f_m\|_{\mathcal{H}_0}^2 \leq \|f - f_m\|_{\mathcal{H}}^2$) by Monte Carlo sampling. Rigorous analysis of the error of such Monte Carlo sampling in $\mathcal{H}_s$ will be explored in the future work.

Furthermore, based on this method, we can directly obtain an approximate of the solution of the PDE boundary value problem (6.3) as follows. For the date set $\{g_s(\boldsymbol{x}), u_s(\boldsymbol{y})\}_{s=1}^M$

(i) Denote the girds of K_1 as $\{\boldsymbol{y}_q\}_{q=1}^Q$. We obtain the Q function-point sets: $\{\mathbb{T}_q\}_{q=1}^Q$, where $\mathbb{T}_q := \{g_i(\boldsymbol{x}), u_i(\boldsymbol{y}_q)\}_{s=1}^M$.

(ii) For each $\mathbb{T}_q$, we learn a functional $f_q \in \mathcal{G}_{\text{ReLU,m}}$ from the neural network to approximate the solution of the problem (6.3) at $\boldsymbol{y}_q$. We obtain functional sets $\{f_q\}_{q=1}^Q$.

Here we obtained the approximation of operator $\mathcal{L}^{-1}$ in the PDE boundary value problem (6.3) at several points.

7 Conclusions and Discussion

In this paper, we establish a neural network to approximate functionals without curse of dimensionality based on the method of Barron space. The method is developed by defining a Fourier-type series on the infinite-dimensional space of functionals and the associated spectral Barron space of functionals. The approximation error of the neural network is $\mathcal{O}(1/\sqrt{m})$ where m is the size of networks, which overcomes the curse of dimensionality. The number of parameters and the network structure in our method only depends on the functional, thus it is not sensitive to the input functions in training.

The proposed method for approximation of functionals without curse of dimensionality can be employed in learning functionals, such as linear functionals and energy functionals in science and engineering fields. It can also be used to solve PDE problems by neural networks at one or a few given points. This method provides a basis for the further development of methods for learning operators and analysis of properties (e.g., stability [7]) of neural networks for functionals and operators.

Appendix A. Error Analysis for DeepONet

DeepONet [28] is a method to approximate operators (e.g., those associated with PDES) by neural networks. The structure of the neural network in DeepONet is shown in Fig. A.1, which is based on the illustration in Ref. [6]. In Fig. A.1, the network $\mathcal{G}$ is DeepONet whose input space $W_1^\infty[0,1]^{d_1}$ and output space $W_1^\infty[0,1]^{d_2}$ are infinite dimensional spaces. It can be divided into three steps. The first step $\mathcal{E}$ (encoding) is to reduce the infinite dimensional input space into a finite dimensional space (N-dimensional space), i.e., each $v \in W_1^\infty[0,1]^{d_1}$ is approximated by a piece-wise linear function v_N determined by $\{v(\boldsymbol{x}_i)\}_{i=1}^N$. The last step $\mathcal{R}$ (reconstruction) is to approximate the operator $\mathcal{G}$ by p functionals, i.e., for each $v \in W_1^\infty[0,1]^{d_1}$, a function $\sum_{k=1}^p s_k(v)\sigma(\boldsymbol{w}_k \cdot \boldsymbol{y} + \zeta_k)$ is constructed to approximate the function $\mathcal{G}(v) \in W_1^\infty[0,1]^{d_2}$, where each $s_k(v)$ is a functional. The intermediate Step $\mathcal{A}$ is to approximate the function $\boldsymbol{h} = (h_1, \ldots, h_p) : \mathbb{R}^N \to \mathbb{R}^p$ by neural networks, where

$$h_k(v(\boldsymbol{x}_1), v(\boldsymbol{x}_2), \ldots, v(\boldsymbol{x}_N)) = s_k(v).$$

Convergence of the DeepONet method [28] for operator learning is guaranteed by the approximation theorem for operators by neural networks [6, Theorem B.3] (in Appendix), which, however, does not provide the accuracy dependence on the number of sample points. The paper [21] studied the error of the DeepONet method and overcame the curse of dimensionality in this method by considering smooth functions with exponentially decaying coefficients in the Fourier series.

In DeepONet, for a continuous operator G defined in $W_1^\infty([0,1]^{d_1}) \to W_1^\infty([0,1]^{d_2})$ and a function $v \in W_1^\infty([0,1]^{d_1})$, $G(v)$ is a function belonging to $W_1^\infty([0,1]^{d_2})$. This function $G(v)$ can be approximated by a two-layer network architected by the activation function $\sigma(x) = \max\{x, 0\}$ in the following form by [30, Theorem B.1] (given in Appendix):

$$\left\| G(v)(\boldsymbol{y}) - \sum_{k=1}^{p} c_k[G(v)]\sigma(\boldsymbol{w}_k \cdot \boldsymbol{y} + \zeta_k) \right\|_{L_\infty([0,1]^{d_2})} \le Cp^{-\frac{1}{d_2}} \|G(v)\|_{W_1^\infty([0,1]^{d_2})}, \tag{A.1}$$

where $\boldsymbol{w}_k \in \mathbb{R}^{d_2}, \zeta_k \in \mathbb{R}$ for $k = 1, \ldots, p$, c_k is continuous functionals, and C (or C_1, C_2 in other inequalities in this paper) is some constant independent of the parameters. Denote $g_k(v) := c_k[G(v)]$, which is a functional from $W_1^\infty([0,1]^{d_1}) \to \mathbb{R}$. The remaining task in DeepONet is to approximate these functionals by neural networks.

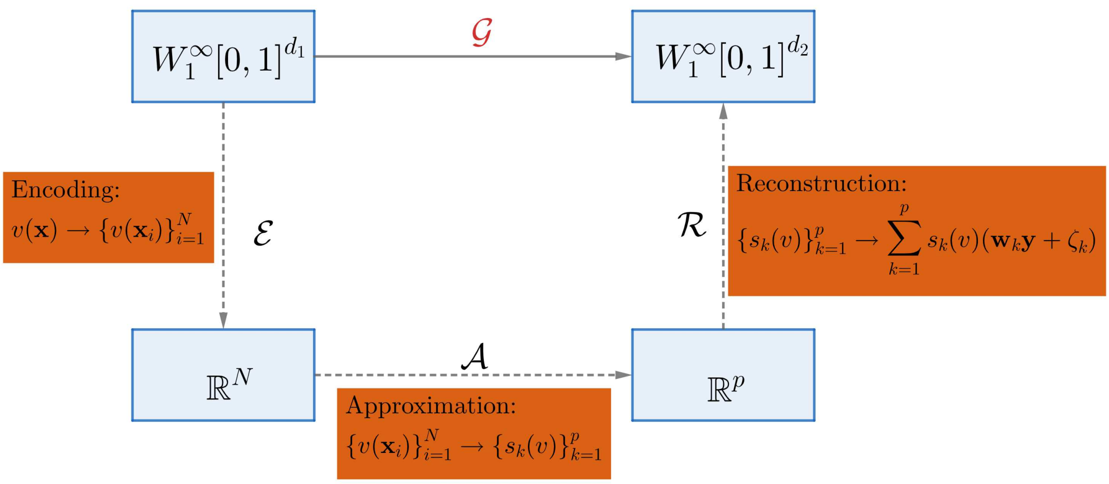

Figure A.1: The structure of DeepONet.

DOI https://doi.org/10.4208/jml.221018 | Generated on 2024-12-19 04:40:52

In DeepONet, the domain of functions $[0,1]^{d_1}$ is divided into s^{d_1} parts by the $N := (s+1)^{d_1}$ nodes denote by $\{\boldsymbol{x}_j\}_{j=1}^N$, where each $x_j \in \{0, 1/s, 2/s, \cdots, (s-1)/s, 1\}^{d_1}$. Based on these N nodes, a piece-wise linear function of $v_N(\boldsymbol{x})$ is defined in DeepONet such that $\|v_N(\boldsymbol{x}) - v(\boldsymbol{x})\|_\infty \leq C_0 s^{-1}$. Further assume that g_k is a Lipschitz continuous functional in $L_\infty([0,1]^{d_1})$ with a Lipschitz constant L_k, then denote $L = \max_{1\leq k\leq p} L_k$ and we have

$$\big|g_k(v) - g_k(v_N)\big| \leq L_k \big\|v_N(\boldsymbol{x}) - v(\boldsymbol{x})\big\|_\infty \leq C_0 L s^{-1} \leq C_1 N^{-\frac{1}{d_1}}. \tag{A.2}$$

For each $g_k(v_N)$, a function $h_k : \mathbb{R}^N \to \mathbb{R}$ is defined such that

$$h_k\big(v(\boldsymbol{x}_1), v(\boldsymbol{x}_2), \cdots, v(\boldsymbol{x}_N)\big) := g_k(v_N). \tag{A.3}$$

Suppose that v is bounded by 1 and $h_k \in W_1^\infty([-1,1]^N)$. By [30, Theorem B.1], we have

$$\begin{aligned}&\left|h_k\big(v(\boldsymbol{x}_1), v(\boldsymbol{x}_2), \cdots, v(\boldsymbol{x}_N)\big) - \sum_{i=1}^m c_i^k \sigma\left(\sum_{j=1}^N \xi_{ij}^k v(\boldsymbol{x}_j) + \theta_i^k\right)\right| \\ &\leq C_2 m^{-\frac{1}{N}} \|h_k\|_{W_1^\infty([-1,1]^N)}.\end{aligned} \tag{A.4}$$

Combining Eqs. (A.1), (A.2) and (A.4), with several further assumptions, the error of DeepONet is bounded by

$$\begin{aligned}&\left\|G(v)(\boldsymbol{y}) - \sum_{k=1}^p \sum_{i=1}^m c_i^k \sigma\left(\sum_{j=1}^N \xi_{ij}^k v(\boldsymbol{x}_j) + \theta_i^k\right)\sigma\big(\boldsymbol{w}_k \cdot \boldsymbol{y} + \zeta_k\big)\right\|_{L_\infty([0,1]^{d_2})} \\ &\leq C p^{-\frac{1}{d_2}} \|G(v)\|_{W_1^\infty([0,1]^{d_2})} + C_1(p,L) N^{-\frac{1}{d_1}} + C_2(p,L,\lambda) m^{-\frac{1}{N}},\end{aligned} \tag{A.5}$$

where $L = \max_{1\leq k\leq p} L_k$ and $\lambda = \max_{1\leq k\leq p} \|h_k\|_{W_1^\infty([-1,1]^N)}$.

It can be seen that every term in Eq. (A.5) has the problem of the curse of dimensionality. The first term in Eq. (A.5) is the error for approximating the operators by some functionals, after which the learning of operators is reduced to learning of functionals. The last two terms in Eq. (A.5) are errors from the approximation of functionals by neural networks (the second one from the approximation of the function by a piecewise linear function and the last one from the approximation of this piecewise linear function by a neural network), in which the curse of dimensionality exists even if we consider such an approximation in low dimensional spaces (i.e., small d_1 and d_2). In fact, for the second term $C_1(p,L)N^{-1/d_1} \leq \varepsilon$, where ε is small, we should have at least $N \sim 1/\varepsilon^{d_1}$, and accordingly, the last term $C_2(p,L,\lambda)m^{-1/N} \sim m^{-\varepsilon^{d_1}}$, whose curse of dimensionality is more serious than that of m^{-1/d_1}.

Appendix B. Theorems on Approximations of Functions, Functionals and Operators

In this subsection of Appendix, we summarize some available theorems on the approximations of functions, functionals and operators, which are used in our proofs.

Theorem B.1 ([30]). *Suppose σ is a continuous non-polynomial function and K is a compact in $\mathbb{R}^d$, then there are positive integers p, constants w_k, ζ_k for $k = 1, \ldots, p$ and continuous linear functionals $c_k : W_r^q(K) \to \mathbb{R}$ such that for any $v \in W_r^q(K)$,*

$$\left\| v - \sum_{k=1}^{p} c_k(v)\sigma\big(\boldsymbol{w}_k \cdot \boldsymbol{x} + \zeta_k\big) \right\|_{L_q(K)} \leq c p^{-\frac{r}{d}} \|v\|_{W_r^q(K)}, \tag{B.1}$$

where $W_r^q(K)$ is the set of function in $L_q(K)$ with finite Sobolev norms

$$\|g\|_{W_r^q(K)} := \sum_{0 \leq |\boldsymbol{j}| \leq r} \big\|D^{\boldsymbol{j}} g\big\|_{L_q(K)}. \tag{B.2}$$

Theorem B.2 ([5]). *Suppose σ is a continuous non-polynomial function, U is a compact set in $C([a,b]^{d_1})$. f is a continuous functionals defined on U. Then for any $\varepsilon > 0$, there are positive integers n, m, constants $c_i, \xi_{ij}, \theta_i \in \mathbb{R}$ for $i = 1, \ldots, n, j = 1, \ldots, m$ such that*

$$\left| f(v) - \sum_{i=1}^{n} c_i \sigma \left(\sum_{j=1}^{m} \xi_{ij} v(\boldsymbol{x}_j) + \theta_i \right) \right| \leq \varepsilon \tag{B.3}$$

holds for all $v \in U$.

Theorem B.3 ([6]). *Suppose σ is a continuous non-polynomial function, $K_1 = [0,1]^{d_1}$, $K_2 = [0,1]^{d_2}$, V is a compact set in $C(K_1)$ and G is a nonlinear continuous operator, which maps V into $C(K_2)$. Then for any $\varepsilon > 0$, there are positive integers n, p, m, constants c_i^k, ξ_{ij}^k, θ_i^k, $\zeta_k \in \mathbb{R}$, $\boldsymbol{w}_k \in \mathbb{R}^{d_2}$, $\boldsymbol{x}_j \in K_1$ for $i = 1, \ldots, n$, $k = 1, \ldots, p$, $j = 1, \ldots, m$ such that*

$$\left| G(v)(\boldsymbol{y}) - \sum_{k=1}^{p} \sum_{i=1}^{n} c_i^k \sigma \left(\sum_{j=1}^{m} \xi_{ij}^k v(\boldsymbol{x}_j) + \theta_i^k \right) \sigma\big(\boldsymbol{w}_k \cdot \boldsymbol{y} + \zeta_k\big) \right| \leq \varepsilon \tag{B.4}$$

holds for all $v \in V$ and $\boldsymbol{y} \in K_2$.

Acknowledgments

This work was supported by HKUST IEG19SC04 and the Project of Hetao Shenzhen-HKUST Innovation Cooperation Zone HZQB-KCZYB-2020083.

References

[1] F. Bach, Breaking the curse of dimensionality with convex neural networks, *J. Mach. Learn. Res.*, **18**(1): 629–681, 2017.

[2] A. R. Barron, Universal approximation bounds for superpositions of a sigmoidal function, *IEEE Trans. Inf. Theory*, **39**(3):930–945, 1993.

[3] R. E. Bellman, *Dynamic Programming*, Princeton University Press, 1957.

[4] K. Bhattacharya, B. Hosseini, N. Kovachki, and A. Stuart, Model reduction and neural networks for parametric PDEs, *arXiv:2005.03180*, 2020.
[5] T. Chen and H. Chen, Approximations of continuous functionals by neural networks with application to dynamic systems, *IEEE Trans. Neural Netw.*, **4**(6):910–918, 1993.
[6] T. Chen and H. Chen, Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems, *IEEE Trans. Neural Netw.*, **6**(4):911–917, 1995.
[7] A. Cohen, R. DeVore, G. Petrova, and P. Wojtaszczyk, Optimal stable nonlinear approximation, *Found. Comput. Math.*, **22**:607–648, 2022.
[8] W. E, J. Han, and A. Jentzen, Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations, *Commun. Math. Stat.*, **5**(4): 349–380, 2017.
[9] W. E, C. Ma, and L. Wu, A priori estimates of the population risk for two-layer neural networks, *Commun. Math. Sci.*, **17**(5):1407–1425, 2019.
[10] W. E, C. Ma, and L. Wu, Barron spaces and the compositional function spaces for neural network models, *arXiv:1906.08039*, 2019.
[11] W. E, C. Ma, and L. Wu, The Barron space and the flow-induced function spaces for neural network models, *Constr. Approx.*, **55**:369–406, 2022.
[12] W. E and B. Yu, The Deep Ritz Method: A deep learning-based numerical algorithm for solving variational problems, *Commun. Math. Stat.*, **6**(1):1–12, 2018.
[13] D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*, Academic Press, 2002.
[14] C. Gin, D. Shea, S. Brunton, and J. Kutz, DeepGreen: Deep learning of green's functions for nonlinear boundary value problems. *arXiv:2101.07206*, 2020.
[15] P. Grohs, F. Hornung, A. Jentzen, and P. Von Wurstemberger, A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of black-scholes partial differential equations, *arXiv:1809.02362*, 2018.
[16] J. Han, A. Jentzen, and W. E, Solving high-dimensional partial differential equations using deep learning, *Proc. Natl. Acad. Sci. U. S. A.*, **115**(34):8505–8510, 2018.
[17] M. de Hoop, N. Kovachki, N. Nelsen, and A. Stuart, Convergence rates for learning linear operators from noisy data, *arXiv:2108.12515*, 2021.
[18] Y. Khoo, J. F. Lu, and L. X. Ying, Solving for high-dimensional committor functions using artificial neural networks, *Res. Math. Sci.*, **6**(1), 2019.
[19] J. Klusowski and A. Barron, Approximation by combinations of relu and squared relu ridge functions with ℓ_1 and ℓ_0 controls, *IEEE Trans. Inf. Theory*, **64**(12):7649–7656, 2018.
[20] I. E. Lagaris, A. Likas, and D. I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, *IEEE Trans. Neural Netw.*, **9**(5):987–1000, 1998.
[21] S. Lanthaler, S. Mishra, and G. Karniadakis, Error estimates for deeponets: A deep learning framework in infinite dimensions, *Trans. Math. Appl.*, **6**(1):tnac001, 2022.
[22] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, Fourier neural operator for parametric partial differential equations, In: *International Conference on Learning Representations*, 2020.
[23] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, Multipole graph neural operator for parametric partial differential equations, *arXiv:2006.09535*, 2020.
[24] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, Neural operator: Graph kernel network for partial differential equations, *arXiv:2003.03485*, 2020.
[25] Z. Li, C. Ma, and L. Wu, Complexity measures for neural networks with general activation functions using path-based norms, *arXiv:2009.06132*, 2020.
[26] H. Liu, H. Yang, M. Chen, T. Zhao, and W. Liao, Deep nonparametric estimation of operators between infinite dimensional spaces, *arXiv:2201.00217*, 2022.
[27] J. Lu, Y. Lu, M. Wang, A priori generalization analysis of the deep Ritz method for solving high dimensional elliptic partial differential equations, In: *Conference on Learning Theory*, 3196–3241, PMLR, 2021.

[28] L. Lu, P. Jin, and G. Karniadakis, DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators, *arXiv:1910.03193*, 2019.
[29] T. Luo and H. Yang, Two-layer neural networks for partial differential equations: Optimization and generalization theory, *arXiv:2006.15733*, 2020.
[30] H. Mhaskar, Neural networks for optimal approximation of smooth and analytic functions, *Neural Comput.*, **8**(1):164–177, 1996.
[31] G. Pisier, Remarques sur un résultat non publié de B. Maurey, *Séminaire Analyse fonctionnelle (dit)*,1–12, 1981.
[32] M. Raissi, P. Perdikaris, and G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.*, **378**:686–707, 2019.
[33] Y. Shin, J. Darbon, and G. Karniadakis, On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs, *Commun. Comput. Phys.*, **28**:2042–2074, 2020.
[34] J. Sirignano and K. Spiliopoulos, Dgm: A deep learning algorithm for solving partial differential equations, *J. Comput. Phys.*, **375**:1339–1364, 2018.
[35] B. Thind, K. Multani, and J. Cao, Deep learning with functional inputs, *arXiv:2006.09590*, 2020.
[36] S. Wojtowytsch and W. E, Some observations on partial differential equations in Barron and multi-layer spaces, *arXiv:2012.01484*, 2020.
[37] L. Zhang, T. Luo, Y. Zhang, Z. Xu, and Z. Ma, MOD-Net: A machine learning approach via model-operator-data network for solving PDEs, *a rXiv:2107.03673*, 2021.