Deep ReLU neural network approximation of parametric and stochastic elliptic PDEs with lognormal inputs

Dinh Dũng *[a], Van Kien Nguyen[b], and Duong Thanh Pham[c]

[a]Information Technology Institute, Vietnam National University, Hanoi
144 Xuan Thuy, Cau Giay, Hanoi, Vietnam
Email: dinhzung@gmail.com
[b]Faculty of Basic Sciences, University of Transport and Communications
No.3 Cau Giay Street, Lang Thuong Ward, Dong Da District, Hanoi, Vietnam
Email: kiennv@utc.edu.vn
[c]Vietnamese German University,
Le Lai street, Binh Duong New City, Binh Duong Province, Vietnam
Email: duong.pt@vgu.edu.vn

January 25, 2022

Abstract

We investigate non-adaptive methods of deep ReLU neural network approximation of the solution u to parametric and stochastic elliptic PDEs with lognormal inputs on non-compact set $\mathbb{R}^\infty$. The approximation error is measured in the norm of the Bochner space $L_2(\mathbb{R}^\infty, V, \gamma)$, where γ is the tensor product standard Gaussian probability on $\mathbb{R}^\infty$ and V is the energy space. The approximation is based on an m-term truncation of the Hermite generalized polynomial chaos expansion (gpc) of u. Under a certain assumption on ℓ_q-summability condition for lognormal inputs ($0 < q < \infty$), we proved that for every integer $n > 1$, one can construct a non-adaptive compactly supported deep ReLU neural network $\boldsymbol{\phi}_n$ of size not greater than n on $\mathbb{R}^m$ with $m = \mathcal{O}(n/\log n)$, having m outputs so that the summation constituted by replacing polynomials in the m-term truncation of Hermite gpc expansion by these m outputs approximates u with an error bound $\mathcal{O}\left((n/\log n)^{-1/q}\right)$. This error bound is comparable to the error bound of the best approximation of u by n-term truncations of Hermite gpc expansion which is $\mathcal{O}(n^{-1/q})$. We also obtained some results on similar problems for parametric and stochastic elliptic PDEs with affine inputs, based on the Jacobi and Taylor gpc expansions.

Keywords and Phrases: High-dimensional approximation; Deep ReLU neural networks; Parametric and stochastic elliptic PDEs; Lognormal inputs.

Mathematics Subject Classifications (2010): 65C30, 65D05, 65D32, 65N15, 65N30, 65N35.

*Corresponding author

1 Introduction

The aim of the present paper is to construct deep ReLU neural networks for approximation of parametric and stochastic elliptic PDEs with lognormal or affine inputs. We investigate the convergence rate of this approximation in terms of the size of the approximating deep ReLU neural networks.

The universal approximation capacity of neural networks has been known since the 1980's ([13, 35, 23, 6]). In recent years, deep neural networks have been rapidly developed and successfully applied to a wide range of fields. The main advantage of deep neural networks over shallow ones is that they can output compositions of functions cheaply. Since their application range is getting wider, theoretical analysis revealing reasons of these significant practical improvements attracts substantial attention [2, 18, 40, 49, 50]. In the last several years, there has been a number of interesting papers that addressed the role of depth and architecture of deep neural networks in approximating functions that possess special regularity properties such as analytic functions [20, 38], differentiable functions [45, 52], oscillatory functions [29], functions in Sobolev or Besov spaces [1, 27, 30, 53]. High-dimensional approximations by deep neural networks have been studied in [39, 48, 16, 17], and their applications to high-dimensional PDEs in [47, 21, 43, 31, 25, 26, 28]. Most of these papers used deep ReLU (Rectified Linear Unit) neural networks since the rectified linear unit is a simple and preferable activation function in many applications. The output of such a neural network is a continuous piece-wise linear function which is easily and cheaply computed. We refer the reader to the recent surveys [19, 44] for various problems and aspects of neural network approximation and bibliography.

In computational uncertainty quantification, the problem of efficient (non-neural-network) numerical approximation for parametric and stochastic partial differential equations (PDEs) has been of great interest and achieved significant progress in recent years. There is a vast number of works on this topic to mention all of them. We point out just some works [3, 5, 4, 8, 10, 11, 12, 7, 9, 15, 14, 22, 34, 54, 55] which are directly related to our paper.

Recently, a number of works have been devoted to various problems and methods of deep neural network approximation for parametric and stochastic PDEs such as dimensionality reduction [51], deep neural network expression rates for the Taylor generalized polynomial chaos expansion (gpc) of solutions to parametric elliptic PDEs [46], reduced basis methods [36] the problem of learning the discretized parameter-to-solution map in practice [24], Bayesian PDE inversion [42, 32, 31], etc. In particular, in [46] the authors proved dimension-independent deep neural network expression rate bounds of the uniform approximation of solution to parametric elliptic PDE with affine inputs on $\mathbb{I}^\infty := [-1,1]^\infty$ based on n-term truncations of the non-orthogonal Taylor gpc expansion. The construction of approximating deep neural networks relies on weighted summability of the Taylor gpc expansion coefficients of the solution which is derived from its analyticity.

Let $D \subset \mathbb{R}^d$ be a bounded Lipschitz domain. Consider the diffusion elliptic equation

$$-\operatorname{div}(a\nabla u) \;=\; f \quad \text{in} \quad D, \quad u|_{\partial D} \;=\; 0, \tag{1.1}$$

for a given fixed right-hand side f and a spatially variable scalar diffusion coefficient a. Denote by $V := H_0^1(D)$ the energy space and $H^{-1}(D)$ the dual space of V. Assume that $f \in H^{-1}(D)$ (in what follows this preliminary assumption always holds without mention). If $a \in L_\infty(D)$ satisfies the ellipticity assumption

$$0 < a_{\min} \le a \le a_{\max} < \infty,$$

by the well-known Lax-Milgram lemma, there exists a unique solution $u \in V$ to the equation (1.1) in the weak form

$$\int_D a\nabla u \cdot \nabla v \, \mathrm{d}\boldsymbol{x} \;=\; \langle f, v\rangle, \quad \forall v \in V.$$

We consider diffusion coefficients having a parametrized form $a = a(\boldsymbol{y})$, where $\boldsymbol{y} = (y_j)_{j\in\mathbb{N}}$ is a sequence of real-valued parameters ranging in the set $\mathbb{U}^\infty$ which is either $\mathbb{R}^\infty$ or $\mathbb{I}^\infty$. Denote by $u(\boldsymbol{y})$ the solution to the parametrized diffusion elliptic equation

$$-\operatorname{div}(a(\boldsymbol{y})\nabla u(\boldsymbol{y})) \;=\; f \quad \text{in} \quad D, \quad u(\boldsymbol{y})|_{\partial D} \;=\; 0. \tag{1.2}$$

The resulting solution operator maps $\boldsymbol{y} \in \mathbb{U}^\infty \mapsto u(\boldsymbol{y}) \in V$. The objective is to achieve a numerical approximation of this complex map by a small number of parameters with a guaranteed error in a given norm. Depending on the nature of the modeled object, the parameter $\boldsymbol{y}$ may be either deterministic or random. In the present paper, we consider the so-called lognormal case when $\mathbb{U}^\infty = \mathbb{R}^\infty$ and the diffusion coefficient a is of the form

$$a(\boldsymbol{y}) = \exp(b(\boldsymbol{y})), \quad \text{with} \;\; b(\boldsymbol{y}) = \sum_{j=1}^\infty y_j \psi_j, \tag{1.3}$$

where the y_j are i.i.d. standard Gaussian random variables and $\psi_j \in L_\infty(D)$. We also consider the affine case when $\mathbb{U}^\infty = \mathbb{I}^\infty$ and the diffusion coefficient a is of the form

$$a(\boldsymbol{y}) = \bar{a} + \sum_{j=1}^\infty y_j \psi_j. \tag{1.4}$$

Let us briefly describe the main contribution of the present paper. We investigate non-adaptive methods of deep ReLU neural network approximation of the solution $u(\boldsymbol{y})$ to parametric and stochastic elliptic PDEs (1.2) with lognormal inputs (1.3) on non-compact set $\mathbb{R}^\infty$. The approximation is based on truncations of the orthonormal Hermite gpc expansion of $u(\boldsymbol{y})$:

$$u(\boldsymbol{y}) = \sum_{\boldsymbol{s}\in\mathbb{F}} u_{\boldsymbol{s}}\, H_{\boldsymbol{s}}(\boldsymbol{y}), \quad u_{\boldsymbol{s}} \in V.$$

The approximation error is measured in the norm of the Bochner space $L_2(\mathbb{R}^\infty, V, \gamma)$, where γ is the tensor product standard Gaussian probability measure on $\mathbb{R}^\infty$. By using the results on some weighted ℓ_2-summability of the energy norm of V of the Hermite gpc expansion coefficients of $u(\boldsymbol{y})$ obtained in [4], we prove the following. Assume that there exists a sequence of positive numbers $(\rho_j)_{j\in\mathbb{N}}$ such that for some $0 < q < \infty$,

$$\left\| \sum_{j\in\mathbb{N}} \rho_j |\psi_j| \right\|_{L_\infty(D)} < \infty \quad \text{and} \quad \left(\rho_j^{-1}\right)_{j\in\mathbb{N}} \in \ell_q(\mathbb{N}).$$

Let δ be an arbitrary positive number. Then for every integer $n > 1$, we can construct an index sequence $\Lambda_n := \left(\boldsymbol{s}^j\right)_{j=1}^m \subset \mathbb{F}$ with $m = \mathcal{O}(n/\log n)$, and a compactly supported deep ReLU neural network $\boldsymbol{\phi}_n := \left(\phi_j\right)_{j=1}^m$ of size at most n on $\mathbb{R}^m$ so that

(i) The index sequence Λ_n and deep ReLU neural network $\boldsymbol{\phi}_n$ are independent of u;

(ii) The input and output dimensions of $\boldsymbol{\phi}_n$ are at most m;

(iii) The depth of $\boldsymbol{\phi}_n$ is $\mathcal{O}\left(n^\delta\right)$;

(iv) The support of $\boldsymbol{\phi}_n$ is contained in the cube $[-T,T]^m$ with $T = \mathcal{O}(n/\log n)$;

(v) If Φ_j is the extension of ϕ_j to the whole $\mathbb{R}^\infty$ by $\Phi_j(\boldsymbol{y}) = \phi_j\left((y_j)_{j=1}^m\right)$ for $\boldsymbol{y} = (y_j)_{j\in\mathbb{N}}$, then

$$\left\| u - \sum_{j=1}^m u_{\boldsymbol{s}^j} \Phi_j \right\|_{L_2(\mathbb{R}^\infty, V, \gamma)} = \mathcal{O}\Big((n/\log n)^{-1/q}\Big).$$

Notice that the error bound of this approximation is comparable to the error bound of the best approximation of u by n-term truncations of the Hermite gpc expansion as well as of the approximation by the particular truncation $S_n u := \sum_{j=1}^n u_{\boldsymbol{s}^j} H_{\boldsymbol{s}^j}$ which is $\mathcal{O}(n^{-1/q})$. We also obtained some results in manner of the items (i)–(v) on similar problems for parametric and stochastic elliptic PDEs (1.2) with affine inputs (1.4). The proofs of these results rely on the Jacobi and Taylor gpc expansions of $u(\boldsymbol{y})$.

The paper is organized as follows. In Section 2, we present a necessary knowledge about deep ReLU neural networks. Section 3 is devoted to the investigation of non-adaptive methods of deep ReLU neural network approximation of the solution u to the parameterized diffusion elliptic equation (1.2) with lognormal inputs (1.3) on $\mathbb{R}^\infty$. In Section 4, we extend the theory presented in Section 3 to the parameterized diffusion elliptic equation (1.2) with the affine inputs (1.4). Some concluding remarks are presented in Section 5.

Notation As usual, $\mathbb{N}$ denotes the natural numbers, $\mathbb{Z}$ the integers, $\mathbb{R}$ the real numbers and $\mathbb{N}_0 := \{s \in \mathbb{Z} : s \geq 0\}$. We denote $\mathbb{R}^\infty$ the set of all sequences $\boldsymbol{y} = (y_j)_{j\in\mathbb{N}}$ with $y_j \in \mathbb{R}$. Denote by $\mathbb{F}$ the set of all sequences of non-negative integers $\boldsymbol{s} = (s_j)_{j\in\mathbb{N}}$ such that their support $\nu_{\boldsymbol{s}} := \operatorname{supp}(\boldsymbol{s}) := \{j \in \mathbb{N} : s_j > 0\}$ is a finite set. We use $(\boldsymbol{e}^j)_{j\in\mathbb{N}}$ for the standard basis of $\ell_2(\mathbb{N})$. For a set G, we denote by $|G|$ the cardinality of G. We use letters C and K to denote general positive constants which may take different values, and $C_{\alpha,\beta,\ldots}$ and $K_{\alpha,\beta,\ldots}$ when we want to emphasize the dependence of these constants on $\alpha, \beta, \ldots$, or when this dependence is important in a particular situation.

2 Deep ReLU neural networks

In this section, we present some necessary definitions and elementary facts on deep ReLU neural networks. There is a wide variety of neural network architectures and each of them is adapted to specific tasks. We will consider a general type of deep feed-forward neural networks that also allows connections of neurons in non-neighboring layers. In deep neural network approximation, we will employ the ReLU activation function that is defined by $\sigma(t) := \max\{t, 0\}$, $t \in \mathbb{R}$. We will use the notation $\sigma(\boldsymbol{x}) := (\sigma(x_1), \ldots, \sigma(x_d))$ for $\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$.

Definition 2.1 *Let $d, L \in \mathbb{N}$, $L \geq 2$, $N_0 = d$, and $N_1, \ldots, N_L \in \mathbb{N}$. Let $\boldsymbol{W}^\ell = \left(w_{i,j}^\ell\right) \in \mathbb{R}^{N_\ell \times \left(\sum_{i=1}^{\ell-1} N_i\right)}$, $\ell = 1, \ldots, L$, be $N_\ell \times \left(\sum_{i=1}^{\ell-1} N_i\right)$ matrices, and $\boldsymbol{b}^\ell = (b_j^\ell) \in \mathbb{R}^{N_\ell}$. A ReLU neural network Φ with input dimension d, output dimension N_L and L layers is a sequence of matrix-vector tuples*

$$\Phi = \left((\boldsymbol{W}^1, \boldsymbol{b}^1), \ldots, (\boldsymbol{W}^L, \boldsymbol{b}^L)\right),$$

in which the following computation scheme is implemented

$$\begin{aligned}
\boldsymbol{z}^0 &:= \boldsymbol{x} \in \mathbb{R}^d; \\
\boldsymbol{z}^\ell &:= \sigma\left(\boldsymbol{W}^\ell\left(\boldsymbol{z}^0,\ldots,\boldsymbol{z}^{\ell-1}\right)^{\mathrm{T}} + \boldsymbol{b}^\ell\right), \quad \ell = 1,\ldots,L-1; \\
\boldsymbol{z}^L &:= \boldsymbol{W}^L\big(\boldsymbol{z}^0,\ldots,\boldsymbol{z}^{L-1}\big)^{\mathrm{T}} + \boldsymbol{b}^L.
\end{aligned}$$

We call $\boldsymbol{z}^0$ the input and with an ambiguity denote $\Phi(\boldsymbol{x}) := \boldsymbol{z}^L$ the output of Φ and in some places we identify a deep ReLU neural network with its output. We will use the following terminology.

- *The number of layers $L(\Phi) = L$ is the depth of Φ;*
- *The number of nonzero $w_{i,j}^\ell$ and b_j^ℓ is the size of Φ and denoted by $W(\Phi)$;*
- *When $L(\Phi) \geq 3$, Φ is called a deep neural network, and otherwise, a shallow neural network.*

The following two results is easy to verify from the definition above. We also refer the reader to [30, Remark 2.9 and Lemma 2.11] for further remarks and comments.

Lemma 2.2 (Parallelization) *Let $N \in \mathbb{N}$, $\lambda_j \in \mathbb{R}$, $j = 1,\ldots,N$. Let Φ_j, $j = 1,\ldots,N$ be deep ReLU neural networks with input dimension d. Then we can explicitly construct a deep ReLU neural network denoted by Φ so that*

$$\Phi(\boldsymbol{x}) = \sum_{j=1}^N \lambda_j \Phi_j(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^d.$$

Moreover, we have

$$W(\Phi) \leq \sum_{j=1}^N W_j \qquad \text{and} \qquad L(\Phi) = \max_{j=1,\ldots,N} L_j.$$

The network Φ is called the parallelization network of Φ_j, $j = 1,\ldots,N$.

Lemma 2.3 (Concatenation) *Let Φ_1 and Φ_2 be two ReLU neural networks such that output layer of Φ_1 has the same dimension as input layer of Φ_2. Then, we can explicitly construct a ReLU neural network Φ such that $\Phi(\boldsymbol{x}) = \Phi_2(\Phi_1(\boldsymbol{x}))$ for $\boldsymbol{x} \in \mathbb{R}^d$. Moreover we have*

$$W(\Phi) \leq 2W(\Phi_1) + 2W(\Phi_2) \qquad \text{and} \qquad L(\Phi) = L(\Phi_1) + L(\Phi_2).$$

The network Φ is called the concatenation network of Φ_1 and Φ_2.

We recall the following result, see [46, Proposition 3.3].

Lemma 2.4 *For every $\delta \in (0,1)$, $d \in \mathbb{N}$, $d \geq 2$, we can explicitly construct a deep ReLU neural network Φ_P so that*

$$\sup_{\boldsymbol{x} \in [-1,1]^d} \left| \prod_{j=1}^d x_j - \Phi_P(\boldsymbol{x}) \right| \leq \delta.$$

Furthermore, if $x_j = 0$ for some $j \in \{1,\ldots,d\}$ then $\Phi_P(\boldsymbol{x}) = 0$ and there exists a constant $C > 0$ independent of δ and d such that

$$W(\Phi_P) \leq Cd\log(d\delta^{-1}) \qquad \text{and} \qquad L(\Phi_P) \leq C\log d\log(d\delta^{-1}).$$

The statement $\Phi_P(\boldsymbol{x}) = 0$ ($d = 2$) when $x_1 \cdot x_2 = 0$ was proved in [46, Proposition 3.1], see also [30, Proposition C.2] and [41, Proposition 4.1]. But this implies that the statement also holds for general d since the network Φ_P is constructed as an binary tree of the network Φ_P when $d = 2$.

Let φ_1 be the continuous piece-wise function with break points $\{-2, -1, 1, 2\}$ such that $\varphi_1(x) = x$ if $x \in [-1, 1]$ and $\operatorname{supp}(\varphi_1) \subset [-2, 2]$. By this definition, we find that φ can be realized exactly by a deep neural network (still denoted by φ_1) with size $W(\varphi_1) \leq C$ for some positive constant C. Similarly, let φ_0 be the neural network that realizes the continuous piece-wise function with break points $\{-2, -1, 1, 2\}$ and $\varphi_0(x) = 1$ if $x \in [-1, 1]$, $\operatorname{supp}(\varphi_0) \subset [-2, 2]$. Clearly $W(\phi) \leq C$ for some positive constant C.

From Lemma 2.4 we obtain

Lemma 2.5 *Let φ be either φ_0 or φ_1. For every $\delta \in (0, 1)$, $d \in \mathbb{N}$, we can explicitly construct a deep ReLU neural network Φ so that*

$$\sup_{\boldsymbol{x} \in [-2,2]^d} \left| \prod_{j=1}^{d} \varphi(x_j) - \Phi(\boldsymbol{x}) \right| \leq \delta.$$

Furthermore, $\operatorname{supp}(\Phi) \subset [-2, 2]^d$ *and there exists a constant $C > 0$ independent of δ and d such that*

$$W(\Phi) \leq C\big(1 + d \log(d\delta^{-1})\big) \qquad \text{and} \qquad L(\Phi) \leq C\big(1 + \log d \log(d\delta^{-1})\big). \tag{2.1}$$

Proof. The network Φ is constructed as a concatenation of $\{\varphi(x_j)\}_{j=1}^d$ and Φ_P. The estimate (2.1) follows directly from Lemmas 2.3 and 2.4. ∎

3 Parametrized elliptic PDEs with lognormal inputs

In this section, we investigate non-adaptive methods of deep ReLU neural network approximation of the solution $u(\boldsymbol{y})$ to parametrized elliptic PDEs (1.2) with lognormal inputs (1.3) on $\mathbb{R}^\infty$. We construct such methods and prove convergence rates of the approximation by them. The results are derived from a general theory on deep ReLU neural network approximation in Bochner space $L_2(\mathbb{R}^\infty, X, \gamma)$ of functions v on $\mathbb{R}^\infty$ taking values in a Hilbert space X and satisfying some weighted ℓ_2-summability conditions of the Hermite gpc expansion coefficients of v.

3.1 Approximation by truncations of the Hermite gpc expansion

We first recall a concept of infinite tensor product of probability measures. Let $\mu(y)$ be a probability measure on $\mathbb{U}$. We introduce the probability measure $\mu(\boldsymbol{y})$ on $\mathbb{U}^\infty$ as the infinite tensor product of the probability measures $\mu(y_i)$:

$$\mu(\boldsymbol{y}) := \bigotimes_{j \in \mathbb{N}} \mu(y_j), \quad \boldsymbol{y} = (y_j)_{j \in \mathbb{N}} \in \mathbb{U}^\infty.$$

The sigma algebra for $\mu(\boldsymbol{y})$ is generated by the set of cylinders $A := \prod_{j \in \mathbb{N}} A_j$, where $A_j \subset \mathbb{U}$ are univariate μ-measurable sets and only a finite number of A_i are different from $\mathbb{U}$. For such a set A, we have $\mu(A) = \prod_{j\in\mathbb{N}} \mu(A_j)$. If $\varrho(y)$ is the density of $\mu(y)$, i.e., $\mathrm{d}\mu(y) = \varrho(y)\mathrm{d}y$, then we write

$$\mathrm{d}\mu(\boldsymbol{y}) := \bigotimes_{j\in\mathbb{N}} \varrho(y_j)\mathrm{d}(y_j), \quad \boldsymbol{y} = (y_j)_{j\in\mathbb{N}} \in \mathbb{U}^\infty.$$

(For details on infinite tensor product of probability measures, see, e.g., [33, pp. 429–435].)

Let X be a Hilbert space. The probability measure $\mu(\boldsymbol{y})$ induces the Bochner space $L_2(\mathbb{U}^\infty, X, \mu)$ of μ-measurable mappings v from $\mathbb{U}^\infty$ to X for which the norm

$$\|v\|_{L_2(\mathbb{U}^\infty,X,\mu)} := \left(\int_{\mathbb{U}^\infty} \|v(\cdot,\boldsymbol{y})\|_X^2 \,\mathrm{d}\mu(\boldsymbol{y})\right)^{1/2} < \infty.$$

In this section, we consider the lognormal case with $\mathbb{U}^\infty = \mathbb{R}^\infty$ and $\mu(\boldsymbol{y}) = \gamma(\boldsymbol{y})$, the infinite tensor product standard Gaussian probability measure. More precisely, let $\gamma(y)$ be the probability measure on $\mathbb{R}$ with the standard Gaussian density:

$$\mathrm{d}\gamma(y) := g(y)\,\mathrm{d}y, \quad g(y) := \frac{1}{\sqrt{2\pi}} e^{-y^2/2}.$$

Then the infinite tensor product standard Gaussian probability measure $\gamma(\boldsymbol{y})$ on $\mathbb{R}^\infty$ can be defined by

$$\mathrm{d}\gamma(\boldsymbol{y}) := \bigotimes_{j\in\mathbb{N}} g(y_j)\mathrm{d}(y_j).$$

In this section, we make use of the abbreviation $\mathcal{L}_2(X) := L_2(\mathbb{R}^\infty, X, \gamma)$. Denote by $\mathbb{F}$ the set of all sequences of non-negative integers $\boldsymbol{s} = (s_j)_{j\in\mathbb{N}}$ such that their support $\nu_{\boldsymbol{s}} := \{j \in \mathbb{N} : s_j > 0\}$ is a finite set. A powerful strategy for approximation of functions $v \in \mathcal{L}_2(X)$ is based on truncations of the Hermite gpc expansion

$$v(\boldsymbol{y}) = \sum_{\boldsymbol{s}\in\mathbb{F}} v_{\boldsymbol{s}}\, H_{\boldsymbol{s}}(\boldsymbol{y}), \quad v_{\boldsymbol{s}} \in X, \tag{3.1}$$

where

$$H_{\boldsymbol{s}}(\boldsymbol{y}) = \bigotimes_{j\in\mathbb{N}} H_{s_j}(y_j), \quad v_{\boldsymbol{s}} := \int_{\mathbb{R}^\infty} v(\boldsymbol{y})\, H_{\boldsymbol{s}}(\boldsymbol{y})\,\mathrm{d}\gamma(\boldsymbol{y}), \quad \boldsymbol{s} \in \mathbb{F},$$

with $(H_k)_{k\in\mathbb{N}_0}$ being the Hermite polynomials normalized according to $\int_{\mathbb{R}} |H_k(y)|^2\, g(y)\,\mathrm{d}y = 1$. Notice that $(H_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathbb{F}}$ is an orthonormal basis of $L_2(\mathbb{R}^\infty,\gamma) := L_2(\mathbb{R}^\infty,\mathbb{R},\gamma)$. Moreover, for every $v \in \mathcal{L}_2(X)$ represented by the series (3.1), the Parseval's identity holds

$$\|v\|_{\mathcal{L}_2(X)}^2 = \sum_{\boldsymbol{s}\in\mathbb{F}} \|v_{\boldsymbol{s}}\|_X^2.$$

For $\boldsymbol{s}, \boldsymbol{s}' \in \mathbb{F}$, the inequality $\boldsymbol{s}' \le \boldsymbol{s}$ means that $s_j' \le s_j$ for $j \in \mathbb{N}$. A set $\Lambda \subset \mathbb{F}$ is called downward closed if the inclusion $\boldsymbol{s} \in \mathbb{F}$ yields the inclusion $\boldsymbol{s}' \in \mathbb{F}$ for every $\boldsymbol{s}' \in \mathbb{F}$ such that $\boldsymbol{s}' \le \boldsymbol{s}$. A sequence $(\sigma_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathbb{F}}$ is called increasing if $\sigma_{\boldsymbol{s}'} \le \sigma_{\boldsymbol{s}}$ when $\boldsymbol{s}' \le \boldsymbol{s}$.

Assumption A For $v \in \mathcal{L}_2(X)$ represented by the series (3.1), there exists an increasing sequence $\boldsymbol{\sigma} = (\sigma_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathbb{F}}$ of positive numbers such that $(\sigma_{\boldsymbol{s}}^{-1})_{\boldsymbol{s}\in\mathbb{F}} \in \ell_q(\mathbb{F})$ for some q with $0 < q < \infty$ and

$$\left(\sum_{\boldsymbol{s}\in\mathbb{F}} (\sigma_{\boldsymbol{s}} \|v_{\boldsymbol{s}}\|_X)^2\right)^{1/2} \le M < \infty.$$

Assume that $0 < q < \infty$ and $\boldsymbol{\sigma} = (\sigma_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathbb{F}}$ is an increasing sequence of positive numbers. For $\xi > 0$, we introduce the set

$$\Lambda(\xi) := \left\{\boldsymbol{s} \in \mathbb{F} : \sigma_{\boldsymbol{s}}^q \le \xi\right\}, \tag{3.2}$$

and the following numbers (when $\Lambda(\xi)$ is finite):

$$m_1(\xi) := \max_{\boldsymbol{s}\in\Lambda(\xi)} |\boldsymbol{s}|_1, \tag{3.3}$$

and

$$m(\xi) := \max\left\{j \in \mathbb{N} : \exists \boldsymbol{s} \in \Lambda(\xi) \text{ such that } s_j > 0\right\}. \tag{3.4}$$

Sometimes in this paper, without ambiguity we will use m and m_1 instead of $m(\xi)$ and $m_1(\xi)$. Observe that under Assumption A, the set $\Lambda(\xi)$ is finite and downward closed.

For a function $v \in \mathcal{L}_2(X)$ represented by the series (3.1), we define the truncation

$$S_{\Lambda(\xi)}v := \sum_{\boldsymbol{s}\in\Lambda(\xi)} v_{\boldsymbol{s}} H_{\boldsymbol{s}}. \tag{3.5}$$

Notice that if Assumption A holds, then m is finite and, therefore, the truncation $S_{\Lambda(\xi)}v$ of the series (3.1) can be seen as a function on $\mathbb{R}^m$.

Lemma 3.1 *For $v \in \mathcal{L}_2(X)$ satisfying Assumption A and for every $\xi > 1$, there holds*

$$\|v - S_{\Lambda(\xi)}v\|_{\mathcal{L}_2(X)} \le C\xi^{-1/q}.$$

Proof. Applying the Parseval's identity, noting (3.5), (3.2) and Assumption A, we obtain

$$\begin{aligned}\|v - S_{\Lambda(\xi)}\|^2_{\mathcal{L}_2(X)} &= \sum_{\sigma_{\boldsymbol{s}}>\xi^{1/q}} \|v_{\boldsymbol{s}}\|_X^2 = \sum_{\sigma_{\boldsymbol{s}}>\xi^{1/q}} (\sigma_{\boldsymbol{s}}\|v_{\boldsymbol{s}}\|_X)^2\sigma_{\boldsymbol{s}}^{-2} \\ &\le \xi^{-2/q}\sum_{\boldsymbol{s}\in\mathbb{F}}(\sigma_{\boldsymbol{s}}\|v_{\boldsymbol{s}}\|_X)^2 = M^2\xi^{-2/q}.\end{aligned}$$

□

3.2 Approximation by deep ReLU neural networks

In this section, we construct a deep ReLU neural network which can be used to approximate $v \in \mathcal{L}_2(X)$. We primarily approximate v by the truncation $S_{\Lambda(\xi)}v$ (see (3.5)) of the series (3.1). Under the assumptions of Lemma A.2 in Appendix, $S_{\Lambda(\xi)}v$ can be seen as a function on $\mathbb{R}^m$, where we recall that $m := m(\xi)$. Then we approximate $S_{\Lambda(\xi)}v$ by its truncation $S^{\omega}_{\Lambda(\xi)}v$ on a sufficiently large cube

$$B^m_\omega := [-2\sqrt{\omega}, 2\sqrt{\omega}]^m \subset \mathbb{R}^m,$$

where the parameter ω depending on ξ is chosen in an appropriate way.

In what follows, for convenience we consider $\mathbb{R}^m$ as the subset of all $\boldsymbol{y} \in \mathbb{R}^\infty$ such that $y_j = 0$ for $j = m+1, \ldots$. If g is a function on $\mathbb{R}^m$ taking values in a Hilbert space X, then g has an extension to the whole $\mathbb{R}^\infty$ which is denoted again by g, by the formula $g(\boldsymbol{y}) = g\left((y_j)_{j=0}^m\right)$

for $\boldsymbol{y} = (y_j)_{j\in\mathbb{N}}$. The tensor product of standard Gaussian probability measures $\gamma(\boldsymbol{y})$ on $\mathbb{R}^m$ is defined by

$$\mathrm{d}\gamma(\boldsymbol{y}) := \bigotimes_{j=1}^{m} g(y_j)\mathrm{d}(y_j).$$

For a γ-measurable subset Ω in $\mathbb{R}^m$, the spaces $L_2(\Omega, X, \gamma)$ and $L_2(\Omega, \gamma)$ are defined in the usual way.

Our next task is to construct deep ReLU neural networks $\phi_{\boldsymbol{s}}$ on the cube B^m_ω to approximate $H_{\boldsymbol{s}}$, $s \in \Lambda(\xi)$. The network $\boldsymbol{\phi}_{\Lambda(\xi)} := (\phi_{\boldsymbol{s}})_{\boldsymbol{s}\in\Lambda(\xi)}$ on B^m_ω with $|\Lambda(\xi)|$ outputs which is constructed by parallelization is used to construct an approximation of $S^\omega_{\Lambda(\xi)}v$ and hence of v. Namely, we approximate v by

$$\Phi_{\Lambda(\xi)}v(\boldsymbol{y}) := \sum_{\boldsymbol{s}\in\Lambda(\xi)} v_{\boldsymbol{s}}\phi_{\boldsymbol{s}}(\boldsymbol{y}). \tag{3.6}$$

For $\theta, \lambda \geq 0$, we define the sequence

$$p_{\boldsymbol{s}}(\theta, \lambda) := \prod_{j\in\mathbb{N}} (1 + \lambda s_j)^\theta, \quad \boldsymbol{s} \in \mathbb{F}, \tag{3.7}$$

with abbreviation $p_{\boldsymbol{s}}(\theta) := p_{\boldsymbol{s}}(\theta, 1)$.

Our result in this section is read as follows.

Theorem 3.2 *Let $v \in \mathcal{L}_2(X)$ satisfy Assumption A. Let θ be any number such that $\theta \geq 4/q$. Assume that the sequence $(\sigma_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathbb{F}}$ in Assumption A satisfies $\sigma_{\boldsymbol{e}^{i'}} \leq \sigma_{\boldsymbol{e}^i}$ if $i' < i$ and $(p_{\boldsymbol{s}}(\theta)\sigma_{\boldsymbol{s}}^{-1})_{\boldsymbol{s}\in\mathbb{F}} \in \ell_q(\mathbb{F})$. Let K_q and $K_{q,\theta}$ be the constants defined in Lemma A.1 in Appendix. Then for every $\xi > 1$, we can construct a deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi)} := (\phi_{\boldsymbol{s}})_{\boldsymbol{s}\in\Lambda(\xi)}$ on $\mathbb{R}^m$, $m \leq \lfloor K_q\xi \rfloor$, having the following properties.*

(i) *The deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi)}$ is independent of v;*

(ii) *The input and output dimensions of $\boldsymbol{\phi}_{\Lambda(\xi)}$ are at most m;*

(iii) $W\big(\boldsymbol{\phi}_{\Lambda(\xi)}\big) \leq C\xi \log \xi$;

(iv) $L\big(\boldsymbol{\phi}_{\Lambda(\xi)}\big) \leq C\xi^{1/\theta q}$;

(v) $\operatorname{supp}\big(\boldsymbol{\phi}_{\Lambda(\xi)}\big) \subset [-T, T]^m$, *where* $T := 4\sqrt{\lfloor K_{q,\theta}\xi \rfloor}$;

(vi) *The approximation of v by $\Phi_{\Lambda(\xi)}v$ gives the error estimate*

$$\|v - \Phi_{\Lambda(\xi)}v\|_{\mathcal{L}_2(X)} \leq C\xi^{-1/q}. \tag{3.8}$$

Here the constants C are independent of v and ξ.

Let us first introduce the above mentioned function $S^\omega_{\Lambda(\xi)}v$ with a special choice of ω. In this section, for $\xi > 1$, we use the letter ω only for the notation

$$\omega := \lfloor K_{q,\theta}\xi \rfloor, \tag{3.9}$$

where $K_{q,\theta}$ is the constant defined in Lemma A.1 in Appendix. For a function φ defined on $\mathbb{R}$, we denote by φ^ω the truncation of φ on B^1_ω, i.e.,

$$\varphi^\omega(y) := \begin{cases} \varphi(y) & \text{if } y \in B^1_\omega \\ 0 & \text{otherwise.} \end{cases}$$

If $\nu_{\boldsymbol{s}} \subset \{1, \dots, m\}$, we put

$$H^\omega_{\boldsymbol{s}}(\boldsymbol{y}) := \prod_{j=1}^m H^\omega_{s_j}(y_j), \qquad \boldsymbol{y} \in \mathbb{R}^m.$$

We have $H^\omega_{\boldsymbol{s}}(\boldsymbol{y}) = \prod_{j=1}^m H_{s_j}(y_j)$ if $\boldsymbol{y} \in B^m_\omega$ and $H^\omega_{\boldsymbol{s}}(\boldsymbol{y}) = 0$ otherwise.

For a function $v \in \mathcal{L}_2(X)$ represented by the series (3.1), noting the truncation $S_{\Lambda(\xi)}v$ given by (3.5) and (3.2), we define

$$S^\omega_{\Lambda(\xi)}v := \sum_{\boldsymbol{s}\in\Lambda(\xi)} v_{\boldsymbol{s}} H^\omega_{\boldsymbol{s}}.$$

From Lemma A.2 in Appendix one can see that for every $\boldsymbol{s} \in \Lambda(\xi)$, $H_{\boldsymbol{s}}$ and $H^\omega_{\boldsymbol{s}}$ and therefore, $S_{\Lambda(\xi)}v$ and $S^\omega_{\Lambda(\xi)}v$ can be considered as functions on $\mathbb{R}^m$. For $g \in L_2(\mathbb{R}^m, X, \gamma)$, we have $\|g\|_{L_2(\mathbb{R}^m,X,\gamma)} = \|g\|_{L_2(\mathbb{R}^\infty,X,\gamma)}$ in the sense of extension of g. We will make use of these facts without mention.

To prove Theorem 3.2 we will employ a so-called technique of intermediate approximation for estimation of the approximation error in Theorem 3.2 which in our case is as follows. Suppose that the function $\Phi_{\Lambda(\xi)}$ is already constructed. Due to the inequality

$$\begin{aligned} \|v - \Phi_{\Lambda(\xi)}v\|_{\mathcal{L}_2(X)} \le \|v - S_{\Lambda(\xi)}v\|_{\mathcal{L}_2(X)} + \|S_{\Lambda(\xi)}v - S^\omega_{\Lambda(\xi)}v\|_{L_2(\mathbb{R}^m\setminus B^m_\omega,X,\gamma)} \\ + \|S^\omega_{\Lambda(\xi)}v - \Phi_{\Lambda(\xi)}v\|_{L_2(B^m_\omega,X,\gamma)} + \|\Phi_{\Lambda(\xi)}v\|_{L_2(\mathbb{R}^m\setminus B^m_\omega,X,\gamma)}, \end{aligned} \tag{3.10}$$

the estimate (3.8) will be done via estimates of the four terms in the right-hand side. The first term is already estimated as in Lemma 3.1. The estimates for the others will be carried out in the below corresponding lemmas (Lemmas 3.4–3.6). In order to do this we need an auxiliary lemma on estimation of the $L_2(\mathbb{R}^m\setminus B^m_\omega, \gamma)$-norm of a polynomial on $\mathbb{R}^m$.

Lemma 3.3 *Let $\varphi(\boldsymbol{y}) = \prod_{j=1}^m \varphi_j(y_j)$ for $\boldsymbol{y} \in \mathbb{R}^m$, where φ_j is a polynomial in the variable y_j of degree not greater than ω for $j = 1, \dots, m$. Then there holds*

$$\|\varphi\|_{L_2(\mathbb{R}^m\setminus B^m_\omega,\gamma)} \le Cm \exp(-K\omega) \|\varphi\|_{L_2(\mathbb{R}^m,\gamma)},$$

where the constants C and K are independent of ω, m and φ.

Proof. The proof of the lemma relies on the following inequality which is an immediate consequence of [37, Theorem 6.3]. Let ψ be a polynomial of degree at most ℓ. Applying [37, Theorem 6.3] for polynomial $\psi(\sqrt{2}t)$ with weight $\exp(-t^2)$ (in this case $a_\ell = \sqrt{\ell}$, see [37, Page 41]) and $\kappa = \sqrt{2} - 1$ we get

$$\|\psi\|_{L_2(\mathbb{R}\setminus[-2\sqrt{\ell},2\sqrt{\ell}],\gamma)} \ \le\ C \exp(-K\ell)\|\psi\|_{L_2([-\sqrt{2\ell},\sqrt{2\ell}],\gamma)} \tag{3.11}$$

for some positive number C and K independent of ℓ and ψ. We denote

$$I_j := \mathbb{R} \times \dots \times \overset{j\text{th}}{\overset{\downarrow}{\left(\mathbb{R}\setminus[-2\sqrt{\omega}, 2\sqrt{\omega}\,]\right)}} \times \dots \times \mathbb{R} \subset \mathbb{R}^m.$$

Since $\mathbb{R}^m \backslash B^m_\omega = \bigcup_{j=1}^m I_j$, we have

$$\|\varphi\|_{L_2(\mathbb{R}^m\backslash B^m_\omega,\gamma)} \ \le \ \sum_{j=1}^m \|\varphi\|_{L_2(I_j,\gamma)} \ = \ \sum_{j=1}^m \bigg(\|\varphi_j\|_{L_2(\mathbb{R}\backslash B^1_\omega,\gamma)} \prod_{i\neq j} \|\varphi_i\|_{L_2(\mathbb{R},\gamma)} \bigg). \tag{3.12}$$

Applying (3.11) for the polynomials φ_j, for $j = 1,\ldots,m$, whose degree is not greater than ω we obtain

$$\|\varphi_j\|_{L_2(\mathbb{R}\backslash B^1_\omega,\gamma)} \ \le \ C\exp(-K\omega)\,\|\varphi_j\|_{L_2(\mathbb{R},\gamma)}$$

with some constants C and K independent of ω, m and φ. This together with (3.12) yields

$$\|\varphi\|_{L_2(\mathbb{R}^m\backslash B^m_\omega,\gamma)} \ \le \ C\exp(-K\omega)\, m \prod_{i=1}^m \|\varphi_i\|_{L_2(\mathbb{R},\gamma)} \ = \ Cm\exp(-K\omega)\,\|\varphi\|_{L_2(\mathbb{R}^m,\gamma)}\,.$$

□

Lemma 3.4 *Let the assumptions of Theorem 3.2 be satisfied. Then for every $\xi > 1$, we have that*

$$\|S_{\Lambda(\xi)}v - S^\omega_{\Lambda(\xi)}v\|_{L_2(\mathbb{R}^m\backslash B^m_\omega,X,\gamma)} \le C\xi^{-1/q},$$

where the constant C independent of v and ξ.

Proof. By the equality

$$\|H_{\boldsymbol{s}} - H^\omega_{\boldsymbol{s}}\|_{L_2(\mathbb{R}^m,\gamma)} \ = \ \|H_{\boldsymbol{s}}\|_{L_2(\mathbb{R}^m\backslash B^m_\omega,\gamma)}\,, \quad \boldsymbol{s}\in\Lambda(\xi),$$

and the triangle inequality we obtain

$$\begin{aligned}\|S_{\Lambda(\xi)}v - S^\omega_{\Lambda(\xi)}v\|_{L_2(\mathbb{R}^m\backslash B^m_\omega,X,\gamma)} &\le \sum_{\boldsymbol{s}\in\Lambda(\xi)} \|v_{\boldsymbol{s}}\|_X\,\|H_{\boldsymbol{s}} - H^\omega_{\boldsymbol{s}}\|_{L_2(\mathbb{R}^\infty,\gamma)} \\ &= \sum_{\boldsymbol{s}\in\Lambda(\xi)} \|v_{\boldsymbol{s}}\|_X\,\|H_{\boldsymbol{s}}\|_{L_2(\mathbb{R}^m\backslash B^m_\omega,\gamma)}\,.\end{aligned} \tag{3.13}$$

Notice that for every $\boldsymbol{s}\in\Lambda(\xi)$, $H_{\boldsymbol{s}}(\boldsymbol{y}) = \prod_{j=1}^m H_{s_j}(y_j)$, $\boldsymbol{y}\in\mathbb{R}^m$, where H_{s_j} is a polynomial in variable y_j, of degree not greater than $m_1(\xi)\le\omega$. Applying Lemma 3.3 gives

$$\|H_{\boldsymbol{s}}\|_{L_2(\mathbb{R}^m\backslash B^m_\omega,\gamma)} \le Cm\exp(-K\omega)\,\|H_{\boldsymbol{s}}\|_{L_2(\mathbb{R}^m,\gamma)} \le Cm\exp(-K\omega)\,,$$

where the constants C and K are independent of ω, m and $\boldsymbol{s}$. This together with (3.13), (3.9) and the Caushy–Schwarz inequality yields that

$$\begin{aligned}\|S_{\Lambda(\xi)}v - S^\omega_{\Lambda(\xi)}v\|_{L_2(\mathbb{R}^m\backslash B^m_\omega,X,\gamma)} &\le Cm\exp(-K\omega)\sum_{\boldsymbol{s}\in\Lambda(\xi)}\|v_{\boldsymbol{s}}\|_X \\ &\le Cm\exp(-K\omega)\,|\Lambda(\xi)|^{1/2}\bigg(\sum_{\boldsymbol{s}\in\Lambda(\xi)}\|v_{\boldsymbol{s}}\|_X^2\bigg)^{1/2} \\ &\le C\xi^{3/2}\exp(-K\xi) \le C\xi^{-1/q},\end{aligned}$$

where in the last estimate we used Lemmas A.1 and A.2 in Appendix. □

We will now construct a deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi)} := (\phi_{\boldsymbol{s}})_{\boldsymbol{s}\in\Lambda(\xi)}$ on $\mathbb{R}^m$ for approximating $S^{\omega}_{\Lambda(\xi)}v$ by the function $\Phi_{\Lambda(\xi)}v$ defined as in (3.6).

It is well-known that for each $s \in \mathbb{N}_0$, the univariate Hermite polynomial H_s can be written as

$$H_s(x) = s! \sum_{\ell=0}^{\lfloor \frac{s}{2} \rfloor} \frac{(-1)^\ell}{\ell!(s-2\ell)!} \frac{x^{s-2\ell}}{2^\ell} := \sum_{\ell=0}^{s} a_{s,\ell} x^\ell. \tag{3.14}$$

From (3.14) for each $\boldsymbol{s} \in \Lambda(\xi)$ we have

$$H_{\boldsymbol{s}}(\boldsymbol{y}) = \prod_{j=1}^m H_{s_j}(y_j) = \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}} \left(\prod_{j=1}^m a_{s_j,\ell_j} \right) \boldsymbol{y}^{\boldsymbol{\ell}} = \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}} a_{\boldsymbol{\ell}} \boldsymbol{y}^{\boldsymbol{\ell}},$$

where we put $a_{\boldsymbol{\ell}} := \prod_{j=1}^m a_{s_j,\ell_j}$ and $\boldsymbol{y}^{\boldsymbol{\ell}} := \prod_{j=1}^m y_j^{\ell_j}$. Hence, we get for every $\boldsymbol{y} \in B^m_\omega$,

$$S^{\omega}_{\Lambda(\xi)}v(\boldsymbol{y}) = \sum_{\boldsymbol{s}\in\Lambda(\xi)} v_{\boldsymbol{s}} \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}} a_{\boldsymbol{\ell}} \left(\boldsymbol{y}^{\boldsymbol{\ell}}\right)^{\omega} = \sum_{\boldsymbol{s}\in\Lambda(\xi)} v_{\boldsymbol{s}} \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}} a_{\boldsymbol{\ell}} \left(2\sqrt{\omega}\right)^{|\boldsymbol{\ell}|_1} \prod_{j=1}^m \left(\frac{y_j}{2\sqrt{\omega}}\right)^{\ell_j}. \tag{3.15}$$

Let $\boldsymbol{\ell} \in \mathbb{F}$ be such that $\boldsymbol{0} \le \boldsymbol{\ell} \le \boldsymbol{s}$. For $\boldsymbol{\ell} \ne \boldsymbol{0}$, with an appropriate change of variables, the term $\prod_{j=1}^m \left(\frac{y_j}{2\sqrt{\omega}}\right)^{\ell_j}$ can be represented in the form $\prod_{j=1}^{|\boldsymbol{\ell}|_1} \varphi_1(x_j)$, where φ_1 is defined before Lemma 2.5. Hence by Lemma 2.5, for every $\boldsymbol{\ell}$ satisfying $\boldsymbol{0} < \boldsymbol{\ell} \le \boldsymbol{s}$, with

$$\delta_{\boldsymbol{s}}^{-1} := \xi^{1/q+1/2} p_{\boldsymbol{s}}(1) \left(2\sqrt{\omega}\right)^{|\boldsymbol{s}|_1} \max_{\boldsymbol{0}\le\boldsymbol{\ell}\le\boldsymbol{s}} \{|a_{\boldsymbol{\ell}}|\}, \tag{3.16}$$

there exists a deep ReLU neural network $\phi_{\boldsymbol{s},\boldsymbol{\ell}}$ on $\mathbb{R}^m$ such that

$$\sup_{\boldsymbol{y}\in B^m_\omega} \left| \prod_{j=1}^m \left(\frac{y_j}{2\sqrt{\omega}}\right)^{\ell_j} - \phi_{\boldsymbol{s},\boldsymbol{\ell}}\left(\frac{\boldsymbol{y}}{2\sqrt{\omega}}\right) \right| \le \delta_{\boldsymbol{s}}, \tag{3.17}$$

and

$$\operatorname{supp}\left(\phi_{\boldsymbol{s},\boldsymbol{\ell}}\Big(\frac{\cdot}{2\sqrt{\omega}}\Big)\right) \subset B^{|\nu_{\boldsymbol{\ell}}|}_{4\omega}. \tag{3.18}$$

In the case when $\boldsymbol{\ell} = \boldsymbol{0}$, we fix an index $j \in \nu_{\boldsymbol{s}}$ and define the deep ReLU neural network $\phi_{\boldsymbol{s},\boldsymbol{0}}(\boldsymbol{y}) := a_{\boldsymbol{0}}\varphi_0(\frac{y_j}{2\sqrt{\omega}})$ for $\boldsymbol{y} \in \mathbb{R}^m$, where φ_0 is defined before Lemma 2.5. Then $|a_{\boldsymbol{0}} - \phi_{\boldsymbol{s},\boldsymbol{0}}(\boldsymbol{y})| = 0$ for $\boldsymbol{y} \in B^m_\omega$. Observe that the size and depth of $\phi_{\boldsymbol{s},\boldsymbol{0}}$ are bounded by a constant. For $\boldsymbol{\ell} \ne \boldsymbol{0}$, the size and the depth of $\phi_{\boldsymbol{s},\boldsymbol{\ell}}$ are bounded as

$$W\left(\phi_{\boldsymbol{s},\boldsymbol{\ell}}\right) \le C\left(1 + |\boldsymbol{\ell}|_1 \left(\log|\boldsymbol{\ell}|_1 + \log\delta_{\boldsymbol{s}}^{-1}\right)\right) \le C\left(1 + |\boldsymbol{\ell}|_1 \log\delta_{\boldsymbol{s}}^{-1}\right) \tag{3.19}$$

and

$$L\left(\phi_{\boldsymbol{s},\boldsymbol{\ell}}\right) \le C\left(1 + \log|\boldsymbol{\ell}|_1 \left(\log|\boldsymbol{\ell}|_1 + \log\delta_{\boldsymbol{s}}^{-1}\right)\right) \le C\left(1 + \log|\boldsymbol{\ell}|_1 \log\delta_{\boldsymbol{s}}^{-1}\right) \tag{3.20}$$

due to the inequality $|\boldsymbol{\ell}|_1 \le \delta_{\boldsymbol{s}}^{-1}$. In the following we will use the convention $|\boldsymbol{0}|_1 = 1$. Then the estimates (3.19) and (3.20) holds true for all ℓ with $\boldsymbol{0} \le \boldsymbol{\ell} \le \boldsymbol{s}$.

We define the deep ReLU neural network $\phi_{\boldsymbol{s}}$ on $\mathbb{R}^m$ by

$$\phi_{\boldsymbol{s}}(\boldsymbol{y}) := \sum_{\boldsymbol{0} \le \boldsymbol{\ell} \le \boldsymbol{s}} a_{\boldsymbol{\ell}} \left(2\sqrt{\omega}\right)^{|\boldsymbol{\ell}|_1} \phi_{\boldsymbol{s},\boldsymbol{\ell}} \left(\frac{\boldsymbol{y}}{2\sqrt{\omega}} \right), \quad \boldsymbol{y} \in \mathbb{R}^m, \tag{3.21}$$

which is a parallelization of the component deep ReLU neural networks $\phi_{\boldsymbol{s},\boldsymbol{\ell}}\left(\frac{\cdot}{2\sqrt{\omega}}\right)$. From (3.18) it follows

$$\operatorname{supp}(\phi_{\boldsymbol{s}}) \subset B_{4\omega}^{|\nu_{\boldsymbol{s}}|}. \tag{3.22}$$

We define $\boldsymbol{\phi}_{\Lambda(\xi)} := (\phi_{\boldsymbol{s}})_{\boldsymbol{s} \in \Lambda(\xi)}$ as the deep ReLU neural network realized by parallelization $\phi_{\boldsymbol{s}}$, $\boldsymbol{s} \in \Lambda(\xi)$. Consider the approximation of $S^{\omega}_{\Lambda(\xi)} v$ by $\Phi_{\Lambda(\xi)} v$.

Lemma 3.5 *Under the assumptions of Theorem 3.2, we have*

$$\|S^{\omega}_{\Lambda(\xi)} v - \Phi_{\Lambda(\xi)} v\|_{L_2(B^m_\omega, X, \gamma)} \le C \xi^{-1/q},$$

where the constants C is independent of v and ξ.

Proof. From (3.15), (3.17) and (3.21) we have that

$$\begin{aligned}
\|S^{\omega}_{\Lambda(\xi)} v - \Phi_{\Lambda(\xi)} v\|_{L_2(B^m_\omega, X, \gamma)} &= \left\| \sum_{\boldsymbol{s} \in \Lambda(\xi)} v_{\boldsymbol{s}} H^{\omega}_{\boldsymbol{s}} - \sum_{\boldsymbol{s} \in \Lambda(\xi)} v_{\boldsymbol{s}} \phi_{\boldsymbol{s}}(\boldsymbol{y}) \right\|_{L_2(B^m_\omega, X, \gamma)} \\
&\le \sum_{\boldsymbol{s} \in \Lambda(\xi)} \|v_{\boldsymbol{s}}\|_X \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}} |a_{\boldsymbol{\ell}}| \left(2\sqrt{\omega}\right)^{|\boldsymbol{s}|_1} \delta_{\boldsymbol{s}} \le \xi^{-1/q-1/2} \sum_{\boldsymbol{s} \in \Lambda(\xi)} \|v_{\boldsymbol{s}}\|_X \\
&\le \xi^{-1/q-1/2} |\Lambda(\xi)|^{1/2} \left(\sum_{\boldsymbol{s} \in \Lambda(\xi)} \|v_{\boldsymbol{s}}\|_X^2 \right)^{1/2} \le C \xi^{-1/q},
\end{aligned}$$

where in the last estimate we used Lemma A.1 in Appendix. □

Lemma 3.6 *Under the assumptions of Theorem 3.2, we have*

$$\|\Phi_{\Lambda(\xi)} v\|_{L_2(\mathbb{R}^m \setminus B^m_\omega, X, \gamma)} \le C \xi^{-1/q}, \tag{3.23}$$

where the constant C is independent of v and ξ.

Proof. By (3.17) we have $\left|\phi_{\boldsymbol{s},\boldsymbol{\ell}}\left(\frac{\boldsymbol{y}}{2\sqrt{\omega}}\right)\right| \le 2, \ \forall \boldsymbol{y} \in \mathbb{R}^m$. Hence, by (3.21) we have that

$$\begin{aligned}
\left\|\Phi_{\Lambda(\xi)} v\right\|_{L_2(\mathbb{R}^m \setminus B^m_\omega, X, \gamma)} &\le \sum_{\boldsymbol{s} \in \Lambda(\xi)} \|v_{\boldsymbol{s}}\|_X \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}} \left| a_{\boldsymbol{\ell}} \left(2\sqrt{\omega}\right)^{|\boldsymbol{s}|_1} \right| \left\| \phi_{\boldsymbol{s},\boldsymbol{\ell}} \left(\frac{\cdot}{2\sqrt{\omega}} \right) \right\|_{L_2(\mathbb{R}^m \setminus B^m_\omega, \gamma)} \\
&\le 2 \sum_{\boldsymbol{s} \in \Lambda(\xi)} \|v_{\boldsymbol{s}}\|_X \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}} \left| a_{\boldsymbol{\ell}} \left(2\sqrt{\omega}\right)^{|\boldsymbol{s}|_1} \right| \|1\|_{L_2(\mathbb{R}^m \setminus B^m_\omega, \gamma)}.
\end{aligned}$$

Applying Lemma 3.3 to the polynomial $\varphi(\boldsymbol{y}) = 1$, we get

$$\begin{aligned}\big\|\Phi_{\Lambda(\xi)} v\big\|_{L_2(\mathbb{R}^m\setminus B^m_\omega, X,\gamma)} &\le Cm \sum_{\boldsymbol{s}\in\Lambda(\xi)} \|v_{\boldsymbol{s}}\|_X \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}} (4\omega)^{|\boldsymbol{s}|_1/2} \exp(-K\omega)\big|a_{\boldsymbol{\ell}}\big| \\ &\le Cm \sum_{\boldsymbol{s}\in\Lambda(\xi)} \|v_{\boldsymbol{s}}\|_X (4\omega)^{|\boldsymbol{s}|_1/2} \exp(-K\omega) \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}} \big|a_{\boldsymbol{\ell}}\big|.\end{aligned}$$

In order to estimate the sum $\sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}} \big|a_{\boldsymbol{\ell}}\big|$, we need an inequality for the coefficients of Hermite polynomials. By the representation (3.14) of H_s, $s \in \mathbb{N}_0$, there holds

$$\sum_{\ell=0}^{s} |a_{s,\ell}| \le s!. \tag{3.24}$$

Indeed, this inequality is obvious with $s = 0, 1, 2, 3$. When $s \ge 4$ we have $\frac{1}{\ell!(s-2\ell)!} \le \frac{1}{2}$ for all $\ell = 0, \dots, \lfloor s/2 \rfloor$. Therefore,

$$\sum_{\ell=0}^{s} |a_{s,\ell}| \le s! \sum_{\ell=0}^{\lfloor \frac{s}{2}\rfloor} \frac{2^{-\ell}}{\ell!(s-2\ell)!} \le \frac{s!}{2} \sum_{\ell=0}^{\lfloor \frac{s}{2}\rfloor} 2^{-\ell} \le s!.$$

It follows from (3.24) that

$$\sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}} \big|a_{\boldsymbol{\ell}}\big| = \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}} \prod_{j=1}^{m} |a_{s_j,\ell_j}| \le \prod_{j=1}^{m} \sum_{\ell_j=0}^{s_j} |a_{s_j,\ell_j}| \le \prod_{j=1}^{m} s_j!, \tag{3.25}$$

and hence,

$$\sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}} \big|a_{\boldsymbol{\ell}}\big| \le \prod_{j=1}^{m} s_j! \le \prod_{j=1}^{m} |\boldsymbol{s}|_1^{s_j} \le |\boldsymbol{s}|_1^{|\boldsymbol{s}|_1}. \tag{3.26}$$

By using this estimate and Lemma A.1 in Appendix, we can continue the estimation of $\big\|\Phi_{\Lambda(\xi)} v\big\|_{L_2(\mathbb{R}^m\setminus B^m_\omega, X,\gamma)}$ as

$$\begin{aligned}\big\|\Phi_{\Lambda(\xi)} v\big\|_{L_2(\mathbb{R}^m\setminus B^m_\omega, X,\gamma)} &\le Cm \sum_{\boldsymbol{s}\in\Lambda(\xi)} \|v_{\boldsymbol{s}}\|_X (4\omega)^{\frac{m_1}{2}} \exp\left(-4\omega\right) m_1^{m_1} \\ &\le Cm|\Lambda(\xi)|^{1/2}\Bigg(\sum_{\boldsymbol{s}\in\Lambda(\xi)} \|v_{\boldsymbol{s}}\|_X^2\Bigg)^{1/2} (4\omega)^{\frac{m_1}{2}} \exp\left(-K\omega\right) m_1^{m_1} \\ &\le Cm\xi^{1/2} (4\omega)^{\frac{m_1}{2}} \exp\left(-K\omega\right) m_1^{m_1}.\end{aligned}$$

We have from the inequality $\frac{1}{\theta q} \le \frac{1}{4}$ and Lemma A.1 in Appendix that $m_1 \le K_{q,\theta}\xi^{1/4}$, and from Lemma A.2 in Appendix that $m \le K_q \xi$. Taking account of the choice of ω, we derive the estimate

$$\|\Phi_{\Lambda(\xi)} v\|_{L_2(\mathbb{R}^m\setminus B^m_\omega, X,\gamma)} \le C\xi^{3/2}(4K_{q,\theta}\xi)^{K_{q,\theta}\xi^{1/4}/2}(K_{q,\theta}\xi^{1/4})^{K_{q,\theta}\xi^{1/4}} \exp(-KK_{q,\theta}\xi),$$

which implies (3.23). ∎

Denote

$$\Lambda^*(\xi) := \{(\boldsymbol{s},\boldsymbol{\ell}) \in \mathbb{F}\times\mathbb{F} : \boldsymbol{s} \in \Lambda(\xi) \text{ and } \boldsymbol{0} \le \boldsymbol{\ell} \le \boldsymbol{s}\}. \tag{3.27}$$

Now we estimate the size and depth of the deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi)}$.

Lemma 3.7 *Under the assumptions of Theorem 3.2, the input and output dimensions of* $\boldsymbol{\phi}_{\Lambda(\xi)}$ *are at most* $\lfloor K_q \xi \rfloor$,

$$W\big(\boldsymbol{\phi}_{\Lambda(\xi)}\big) \le C\xi \log \xi, \tag{3.28}$$

and

$$L\big(\boldsymbol{\phi}_{\Lambda(\xi)}\big) \le C\xi^{1/\theta q} \log \xi, \tag{3.29}$$

where the constants C *are independent of* v *and* ξ.

Proof. The input dimension of $\boldsymbol{\phi}_{\Lambda(\xi)}$ is not greater than $m(\xi)$ which is at most $\lfloor K_q \xi \rfloor$ by Lemma A.2 in Appendix. The output dimension of $\boldsymbol{\phi}_{\Lambda(\xi)}$ is the number $|\Lambda(\xi)|$ which is at most $\lfloor K_q \xi \rfloor$ by Lemma A.1(i) in Appendix.

By Lemma 2.2 and (3.19) the size of $\boldsymbol{\phi}_{\Lambda(\xi)}$ is estimated as

$$W\big(\boldsymbol{\phi}_{\Lambda(\xi)}\big) = \sum_{(\boldsymbol{s},\boldsymbol{\ell}) \in \Lambda^*(\xi)} L(\phi_{\boldsymbol{s},\boldsymbol{\ell}}) \le C \sum_{(\boldsymbol{s},\boldsymbol{\ell}) \in \Lambda^*(\xi)} \big(1 + |\boldsymbol{\ell}|_1 \log \delta_{\boldsymbol{s}}^{-1}\big). \tag{3.30}$$

From (3.16) we have

$$\log(\delta_{\boldsymbol{s}}^{-1}) \le C\bigg(\log \xi + \log p_{\boldsymbol{s}}(1) + |\boldsymbol{s}|_1 \log(4\omega) + \log\Big(\max_{\boldsymbol{0} \le \boldsymbol{\ell} \le \boldsymbol{s}} |a_{\boldsymbol{\ell}}| \Big) \bigg). \tag{3.31}$$

Noting that $|\boldsymbol{\ell}|_1 \le |\boldsymbol{s}|_1$ for all $(\boldsymbol{s},\boldsymbol{\ell}) \in \Lambda^*(\xi)$, we obtain

$$\begin{aligned} \sum_{(\boldsymbol{s},\boldsymbol{\ell}) \in \Lambda^*(\xi)} \big(1 + |\boldsymbol{\ell}|_1 \log \delta_{\boldsymbol{s}}^{-1}\big) &\le C\Bigg(\log \xi \sum_{(\boldsymbol{s},\boldsymbol{\ell}) \in \Lambda^*(\xi)} |\boldsymbol{s}|_1 + \sum_{(\boldsymbol{s},\boldsymbol{\ell}) \in \Lambda^*(\xi)} |\boldsymbol{s}|_1 \log p_{\boldsymbol{s}}(1) \\ &\quad + \log(2\omega) \sum_{(\boldsymbol{s},\boldsymbol{\ell}) \in \Lambda^*(\xi)} |\boldsymbol{s}|_1^2 + \sum_{(\boldsymbol{s},\boldsymbol{\ell}) \in \Lambda^*(\xi)} |\boldsymbol{s}|_1 \log\Big(\max_{\boldsymbol{0} \le \boldsymbol{\ell} \le \boldsymbol{s}} |a_{\boldsymbol{\ell}}| \Big) \Bigg) \\ &\le C\Bigg(\log \xi \sum_{(\boldsymbol{s},\boldsymbol{\ell}) \in \Lambda^*(\xi)} |\boldsymbol{s}|_1 \log p_{\boldsymbol{s}}(1) \\ &\quad + \log(2\omega) \sum_{(\boldsymbol{s},\boldsymbol{\ell}) \in \Lambda^*(\xi)} |\boldsymbol{s}|_1^2 + \sum_{(\boldsymbol{s},\boldsymbol{\ell}) \in \Lambda^*(\xi)} |\boldsymbol{s}|_1 \log\Big(\max_{\boldsymbol{0} \le \boldsymbol{\ell} \le \boldsymbol{s}} |a_{\boldsymbol{\ell}}| \Big) \Bigg). \end{aligned} \tag{3.32}$$

For the first and second terms on the right-hand side, since $\Big(p_{\boldsymbol{s}}\big(\frac{4}{q}, 1\big)\sigma_{\boldsymbol{s}}^{-1}\Big)_{\boldsymbol{s} \in \mathbb{F}} \in \ell_q(\mathbb{F})$ from Lemma A.3 in Appendix we have

$$\log \xi \sum_{(\boldsymbol{s},\boldsymbol{\ell}) \in \Lambda^*(\xi)} |\boldsymbol{s}|_1 \log p_{\boldsymbol{s}}(1) \le \log \xi \sum_{(\boldsymbol{s},\boldsymbol{\ell}) \in \Lambda^*(\xi)} p_{\boldsymbol{s}}(2) \le C\xi \log \xi \tag{3.33}$$

and

$$\log(2\omega) \sum_{(\boldsymbol{s},\boldsymbol{\ell}) \in \Lambda^*(\xi)} |\boldsymbol{s}|_1^2 \le \log(2\omega) \sum_{(\boldsymbol{s},\boldsymbol{\ell}) \in \Lambda^*(\xi)} p_{\boldsymbol{s}}(2) \le C\xi \log(2\omega) \le C\xi \log \xi, \tag{3.34}$$

where in last inequality we note that $\omega = \lfloor K_{q,\theta} \xi \rfloor$, see (3.9). Now we turn to the third term in (3.32). The inequalities (3.25) imply

$$\log\Big(\max_{\boldsymbol{0} \le \boldsymbol{\ell} \le \boldsymbol{s}} |a_{\boldsymbol{\ell}}| \Big) \le \log\bigg(\prod_{j=1}^m s_j! \bigg) \le \sum_{j=1}^m \log(s_j!) \le \sum_{j=1}^m s_j^2 \le p_{\boldsymbol{s}}(2).$$

Using Lemma A.3 in Appendix again we also obtain

$$\sum_{(\boldsymbol{s},\boldsymbol{\ell})\in\Lambda^*(\xi)} |\boldsymbol{s}|_1\, p_{\boldsymbol{s}}(2) \le \sum_{(\boldsymbol{s},\boldsymbol{\ell})\in\Lambda^*(\xi)} p_{\boldsymbol{s}}(3) \le C\xi,$$

since $\left(p_{\boldsymbol{s}}\left(\frac{4}{q},1\right)\sigma_{\boldsymbol{s}}^{-1}\right)_{\boldsymbol{s}\in\mathbb{F}} \in \ell_q(\mathbb{F})$. This together with (3.33) and (3.34) yields

$$\sum_{(\boldsymbol{s},\boldsymbol{\ell})\in\Lambda^*(\xi)} \left(1+|\boldsymbol{\ell}|_1 \log \delta_{\boldsymbol{s}}^{-1}\right) \le C\xi \log \xi,$$

which combined with (3.30) gives (3.28).

By Lemma 2.2 and (3.20) the depth of $\boldsymbol{\phi}_{\Lambda(\xi)}$ is bounded as

$$L\big(\boldsymbol{\phi}_{\Lambda(\xi)}\big) = \max_{(\boldsymbol{s},\boldsymbol{\ell})\in\Lambda^*(\xi)} L(\phi_{\boldsymbol{s},\boldsymbol{\ell}}) \le C \max_{(\boldsymbol{s},\boldsymbol{\ell})\in\Lambda^*(\xi)} \left(1+\log|\boldsymbol{\ell}|_1 \log \delta_{\boldsymbol{s}}^{-1}\right).$$

Due to (3.31), this inequality can be modified as

$$L\big(\boldsymbol{\phi}_{\Lambda(\xi)}\big) \le C \max_{\boldsymbol{s}\in\Lambda(\xi)} (\log|\boldsymbol{s}|_1) \max_{(\boldsymbol{s},\boldsymbol{\ell})\in\Lambda^*(\xi)} \left(\log \delta_{\boldsymbol{s}}^{-1}\right). \tag{3.35}$$

From Lemma A.1 in Appendix we obtain

$$\max_{\boldsymbol{s}\in\Lambda(\xi)} (\log|\boldsymbol{s}|_1) \le C\log\xi.$$

We have by (3.31) that

$$\max_{(\boldsymbol{s},\boldsymbol{\ell})\in\Lambda^*(\xi)} \left(\delta_{\boldsymbol{s}}^{-1}\right) \le C\left(\log\xi + \max_{\boldsymbol{s}\in\Lambda(\xi)} \log p_{\boldsymbol{s}}(1) + \log(2\omega)\max_{\boldsymbol{s}\in\Lambda(\xi)} |\boldsymbol{s}|_1 + \max_{\boldsymbol{s}\in\Lambda(\xi)} \log\left(\max_{\boldsymbol{0}\le\boldsymbol{\ell}\le\boldsymbol{s}} |a_{\boldsymbol{\ell}}|\right)\right). \tag{3.36}$$

For the second and third terms on the right-hand side, we have by the well-known inequality $\log p_{\boldsymbol{s}}(1) \le |\boldsymbol{s}|_1$ and Lemma A.1 in Appendix,

$$\max_{\boldsymbol{s}\in\Lambda(\xi)} \log p_{\boldsymbol{s}}(1) \le \max_{\boldsymbol{s}\in\Lambda(\xi)} |\boldsymbol{s}|_1 \le C\xi^{1/\theta q}$$

and

$$\log(2\omega)\max_{\boldsymbol{s}\in\Lambda(\xi)} |\boldsymbol{s}|_1 \le C\xi^{1/\theta q}\log\xi.$$

Now we turn to the fourth term in (3.36). From (3.26) it follows that

$$\log\left(\max_{\boldsymbol{0}\le\boldsymbol{\ell}\le\boldsymbol{s}} |a_{\boldsymbol{\ell}}|\right) \le \log\left(|\boldsymbol{s}|_1^{|\boldsymbol{s}|_1}\right) = |\boldsymbol{s}|_1 \log|\boldsymbol{s}|_1.$$

Hence,

$$\max_{(\boldsymbol{s},\boldsymbol{\ell})\in\Lambda^*(\xi)} \log\left(\max_{\boldsymbol{0}\le\boldsymbol{\ell}\le\boldsymbol{s}} |a_{\boldsymbol{\ell}}|\right) \le \max_{\boldsymbol{s}\in\Lambda(\xi)} \left(|\boldsymbol{s}|_1\log|\boldsymbol{s}|_1\right) \le C\xi^{1/\theta q}\log\xi.$$

This together with (3.35)–(3.2) yields (3.29). ∎

We are now in a position to prove Theorem 3.2 .

Proof. [Proofs of Theorem 3.2]. By (3.10) and Lemmas 3.1 and 3.4–3.6 we deduce that

$$\left\|v - \Phi_{\Lambda(\xi)} v\right\|_{\mathcal{L}_2(X)} \le C\xi^{-1/q}.$$

The claim (v) is proven. The claims (i)–(iii) follow from Lemma 3.7 and the claim (iv) from Lemma A.2 in Appendix and (3.22). ∎

3.3 Application to parameterized elliptic PDEs with lognormal inputs

In this section, we apply the results in the previous section to deep ReLU neural network approximation of the solution $u(\boldsymbol{y})$ to the parametrized elliptic PDEs (1.1) with lognormal inputs (1.3). This is based on a weighted ℓ_2-summability of the series $(\|u_{\boldsymbol{s}}\|_V)_{\boldsymbol{s}\in\mathbb{F}}$ in following lemma which has been proven in [4, Theorems 3.3 and 4.2].

Lemma 3.8 *Assume that there exist a number $0 < q < \infty$ and an increasing sequence $\boldsymbol{\rho} = (\rho_j)_{j\in\mathbb{N}}$ of numbers such that $(\rho_j^{-1})_{j\in\mathbb{N}} \in \ell_q(\mathbb{N})$ and*

$$\left\| \sum_{j\in\mathbb{N}} \rho_j |\psi_j| \right\|_{L_\infty(D)} < \infty \,.$$

Then we have that for any $\eta \in \mathbb{N}$,

$$\sum_{\boldsymbol{s}\in\mathbb{F}} (\sigma_{\boldsymbol{s}} \|u_{\boldsymbol{s}}\|_V)^2 < \infty \quad \text{with} \quad \sigma_{\boldsymbol{s}}^2 := \sum_{\|\boldsymbol{s}'\|_{\ell_\infty(\mathbb{N})}\le \eta} \binom{\boldsymbol{s}}{\boldsymbol{s}'} \prod_{j\in\mathbb{N}} \rho_j^{2s_j'} . \tag{3.37}$$

The following lemma is proven in [14, Lemma 5.3].

Lemma 3.9 *Let $0 < q < \infty$, $(\rho_j)_{j\in\mathbb{N}}$ be a sequence of positive numbers such that the sequence $(\rho_j^{-1})_{j\in\mathbb{N}}$ belongs to $\ell_q(\mathbb{N})$. Let θ be an arbitrary nonnegative number and $(p_{\boldsymbol{s}}(\theta))_{\boldsymbol{s}\in\mathbb{F}}$ the sequence given in* (3.7). *Let for $\eta \in \mathbb{N}$ the sequence $(\sigma_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathbb{F}}$ be defined as in* (3.37). *Then for any $\eta > \frac{2(\theta+1)}{q}$, we have*

$$\sum_{\boldsymbol{s}\in\mathbb{F}} p_{\boldsymbol{s}}(\theta) \sigma_{\boldsymbol{s}}^{-q} < \infty.$$

Our result for the solution u to the parametrized elliptic PDEs (1.1) with lognormal inputs (1.3) is read as follows.

Theorem 3.10 *Under the assumptions of Lemma 3.8, let $0 < q < \infty$ and δ be arbitrary positive number. Then for every integer $n > 1$, we can construct a deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi_n)} := (\phi_{\boldsymbol{s}})_{\boldsymbol{s}\in\Lambda(\xi_n)}$ on $\mathbb{R}^m$ with $m := \left\lfloor K \frac{n}{\log n} \right\rfloor$, having the following properties.*

- (i) *The deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi_n)}$ is independent of u;*
- (ii) *The input and output dimensions of $\boldsymbol{\phi}_{\Lambda(\xi_n)}$ are at most m;*
- (iii) $W\big(\boldsymbol{\phi}_{\Lambda(\xi_n)}\big) \le n$;
- (iv) $L\big(\boldsymbol{\phi}_{\Lambda(\xi_n)}\big) \le C_\delta n^\delta$;
- (v) $\operatorname{supp}\big(\boldsymbol{\phi}_{\Lambda(\xi_n)}\big) \subset [-T, T]^m$, *where* $T := C_\delta' \sqrt{\frac{n}{\log n}}$;
- (vi) *The approximation of u by $\Phi_{\Lambda(\xi_n)} u$ defined as in* (3.6), *gives the error estimate*

$$\|u - \Phi_{\Lambda(\xi_n)} u\|_{\mathcal{L}_2(V)} \le C \left(\frac{n}{\log n} \right)^{-1/q} .$$

Here the constants C, K, C_δ and C'_δ are independent of u and n.

Proof. To prove the theorem we apply Theorem 3.2 to the solution u. Without loss of generality we can assume that $\delta \le 1/4$. We take first the number $\theta := 1/\delta q$ satisfying the inequality $\theta \ge 4/q$, and then choose a number $\eta \in \mathbb{N}$ satisfying the inequality $\eta > \frac{2(\theta+1)}{q}$. By using Lemmas 3.8 and 3.9 one can check that for $X = V$ and the sequence $(\sigma_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathbb{F}}$ defined as in (3.37), $u \in \mathcal{L}_2(V)$ satisfies the assumptions of Theorem 3.2. For a given integer $n > 1$, we choose $\xi_n > 1$ as the maximal number satisfying the inequality $C\xi_n \log \xi_n \le n$, where C is the constant in the claim (ii) of Theorem 3.2. It is easy to verify that there exist positive constants C_1 and C_2 independent of n such that $C_1 \frac{n}{\log n} \le \xi_n \le C_2 \frac{n}{\log n}$. From Theorem 3.2 with $\xi = \xi_n$ we deduce the desired results. ∎

4 Parametrized elliptic PDEs with affine inputs

The theory of non-adaptive deep ReLU neural network approximation of functions in Bochner spaces with the infinite tensor product Gaussian measure, which has been discussed in Section 3 can be generalized and extended to other situations. In this section, we present some results on similar problems for the parametrized elliptic equation (1.2) with the affine inputs (1.4). The Jacobi and Taylor gpc expansions of the solution play a basic role in the proofs of these results.

4.1 Approximation by deep ReLU neural networks

For given $a, b > -1$, we consider the infinite tensor product of the Jacobi probability measures on $\mathbb{I}^\infty$

$$\mathrm{d}\nu_{a,b}(\boldsymbol{y}) := \bigotimes_{j\in\mathbb{N}} \delta_{a,b}(y_j)\,\mathrm{d}y_j,$$

where

$$\delta_{a,b}(y) := c_{a,b}(1-y)^a(1+y)^b, \quad c_{a,b} := \frac{\Gamma(a+b+2)}{2^{a+b+1}\Gamma(a+1)\Gamma(b+1)}.$$

If $v \in \mathcal{L}_2(X) := L_2(\mathbb{I}^\infty, X, \nu_{a,b})$ for a Hilbert space X, we consider the orthonormal Jacobi gpc expansion of v of the form

$$v = \sum_{\boldsymbol{s}\in\mathbb{F}} v_{\boldsymbol{s}} J_{\boldsymbol{s}}(\boldsymbol{y}), \tag{4.1}$$

where

$$J_{\boldsymbol{s}}(\boldsymbol{y}) = \bigotimes_{j\in\mathbb{N}} J_{s_j}(y_j), \quad v_{\boldsymbol{s}} := \int_{\mathbb{I}^\infty} v(\boldsymbol{y}) J_{\boldsymbol{s}}(\boldsymbol{y})\mathrm{d}\nu_{a,b}(\boldsymbol{y}),$$

and $(J_k)_{k\ge0}$ is the sequence of Jacobi polynomials on $\mathbb{I} := [-1,1]$ normalized with respect to the Jacobi probability measure, i.e., $\int_{\mathbb{I}} |J_k(y)|^2 \delta_{a,b}(y)\mathrm{d}y = 1$. One has the Rodrigues' formula

$$J_k(y) \ = \ \frac{c_k^{a,b}}{k!2^k}(1-y)^{-a}(1+y)^{-b}\frac{\mathrm{d}^k}{\mathrm{d}y^k}\left((y^2-1)^k(1-y)^a(1+y)^b\right),$$

where $c_0^{a,b} := 1$ and

$$c_k^{a,b} := \sqrt{\frac{(2k+a+b+1)k!\Gamma(k+a+b+1)\Gamma(a+1)\Gamma(b+1)}{\Gamma(k+a+1)\Gamma(k+b+1)\Gamma(a+b+2)}}, \ k \in \mathbb{N}. \tag{4.2}$$

Examples corresponding to the values $a = b = 0$ are the family of the Legendre polynomials, and to the values $a = b = -1/2$ the family of the Chebyshev polynomials.

Assumption B Let $0 < q < \infty$, $c_k^{a,b}$ be defined as in (4.2) and let $(\delta_j)_{j\in\mathbb{N}}$ be a sequence of numbers strictly larger than 1 such that $(\delta_j^{-1})_{j\in\mathbb{N}} \in \ell_q(\mathbb{N})$. For $v \in \mathcal{L}_2(X)$ represented by the series (4.1), there exists a sequence of positive numbers $(\rho_j)_{j\in\mathbb{N}}$ such that $c_k^{a,b}\rho_j^{-k} \le \delta_j^{-k}$ for $k, j \in \mathbb{N}$ and

$$\left(\sum_{\boldsymbol{s}\in\mathbb{F}} (\sigma_{\boldsymbol{s}} \|v_{\boldsymbol{s}}\|_X)^2\right)^{1/2} \le M \; < \infty,$$

where

$$\sigma_{\boldsymbol{s}} := c_{\boldsymbol{s}}^{-1} \prod_{j\in\mathbb{N}} \rho_j^{s_j}, \quad c_{\boldsymbol{s}} := \prod_{j\in\mathbb{N}} c_{s_j}^{a,b}. \tag{4.3}$$

Theorem 4.1 *Let $v \in \mathcal{L}_2(X)$ satisfy Assumption B. Then for every integer $n > 1$, we can construct a deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi_n)} := (\phi_{\boldsymbol{s}})_{\boldsymbol{s}\in\Lambda(\xi_n)}$ on $\mathbb{R}^m$ with $m := \lfloor K \frac{n}{\log n} \rfloor$, having the following properties.*

(i) *The deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi_n)}$ is independent of u;*

(ii) *The input and output dimensions of $\boldsymbol{\phi}_{\Lambda(\xi_n)}$ are at most m;*

(iii) $W\big(\boldsymbol{\phi}_{\Lambda(\xi_n)}\big) \le n$;

(iv) $L\big(\boldsymbol{\phi}_{\Lambda(\xi_n)}\big) \le C(\log n)^2$;

(v) *Let $\Phi_{\Lambda(\xi_n)} v$ be defined by the formula (3.6) with replacing $\mathbb{R}^\infty$ by $\mathbb{I}^\infty$. Then the approximation of v by $\Phi_{\Lambda(\xi_n)} v$ gives the error estimate*

$$\|v - \Phi_{\Lambda(\xi_n)} v\|_{\mathcal{L}_2(X)} \le C \left(\frac{n}{\log n}\right)^{-1/q}.$$

Here the constants C and K are independent of v and n.

The proof of Theorem 4.1 is similar to the proof of Theorem 3.2, but simpler due to Assumption B and the compact property of $\mathbb{I}^\infty$.

We now are in position to prove Theorem 4.1.

Proof. [A sketch of proof of Theorem 4.1] Similar to the proof of Theorem 3.2, this theorem is deduced from a counterpart of Theorem 3.2 for the case $\mathbb{I}^\infty$. It states that for every $\xi > 1$, we can construct a deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi)} := (\phi_{\boldsymbol{s}})_{\boldsymbol{s}\in\Lambda(\xi)}$ on $\mathbb{I}^m$ with $m \le \lfloor K_q \xi \rfloor$, having the following properties.

(i) The input and output dimensions of $\boldsymbol{\phi}_{\Lambda(\xi)}$ are at most m;

(ii) $W\big(\boldsymbol{\phi}_{\Lambda(\xi)}\big) \le C\xi \log \xi$;

(iii) $L\big(\boldsymbol{\phi}_{\Lambda(\xi)}\big) \le C(\log \xi)^2$;

(iv) The approximation of v by $\Phi_{\Lambda(\xi)} v = \sum_{\boldsymbol{s} \in \Lambda(\xi)} v_{\boldsymbol{s}} \phi_{\boldsymbol{s}}$ gives the error estimate

$$\|v - \Phi_{\Lambda(\xi)} v\|_{\mathcal{L}_2(X)} \le C\xi^{-1/q}.$$

Here the constants C are independent of v and ξ.

Let us give a brief proof of these claims. For the function $v \in \mathcal{L}_2(X)$ represented by the series (4.1) and the sequence $(\sigma_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathbb{F}}$ given as in (4.3), we define

$$S_{\Lambda(\xi)} v := \sum_{\boldsymbol{s} \in \Lambda(\xi)} v_{\boldsymbol{s}} J_{\boldsymbol{s}},$$

where $\Lambda(\xi)$ is defined by the formula (3.2) for the sequence $(\sigma_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathbb{F}}$ given as in (4.3). Then in the same way as the proof of Lemma 3.1, we prove the estimate

$$\|v - S_{\Lambda(\xi)} v\|_{\mathcal{L}_2(X)} \le C\xi^{-1/q}. \tag{4.4}$$

By Lemma A.2 in Appendix for every $\boldsymbol{s} \in \Lambda(\xi)$, $J_{\boldsymbol{s}}$ and $S_{\Lambda(\xi)} v$ can be considered as functions on $\mathbb{I}^m$. As the next step, we will construct a deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi)} := (\phi_{\boldsymbol{s}})_{\boldsymbol{s}\in\Lambda(\xi)}$ on $\mathbb{I}^m$ for approximating $S_{\Lambda(\xi)} v$ by $\Phi_{\Lambda(\xi)}$. From (A.4) for each $\boldsymbol{s} \in \mathbb{F}$ we have

$$J_{\boldsymbol{s}}(\boldsymbol{y}) = \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}} a_{\boldsymbol{\ell}} \boldsymbol{y}^{\boldsymbol{\ell}},$$

where $a_{\boldsymbol{\ell}} := \prod_{j=1}^m a_{s_j,\ell_j}$ and $\boldsymbol{y}^{\boldsymbol{\ell}} := \prod_{j=1}^m y_j^{\ell_j}$. Hence, we get for every $\boldsymbol{y} \in \mathbb{I}^m$,

$$S_{\Lambda(\xi)} v(\boldsymbol{y}) := \sum_{\boldsymbol{s} \in \Lambda(\xi)} v_{\boldsymbol{s}} J_{\boldsymbol{s}}(\boldsymbol{y}) = \sum_{\boldsymbol{s} \in \Lambda(\xi)} v_{\boldsymbol{s}} \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}} a_{\boldsymbol{\ell}} \boldsymbol{y}^{\boldsymbol{\ell}}.$$

By Lemma 2.5, for every $\boldsymbol{\ell}$ with $\boldsymbol{0} \le \boldsymbol{\ell} \le \boldsymbol{s}$, with

$$\delta_{\boldsymbol{s}}^{-1} := \xi^{1/q} p_{\boldsymbol{s}}(1) \max_{\boldsymbol{0}\le\boldsymbol{\ell}\le\boldsymbol{s}} \{|a_{\boldsymbol{\ell}}|\},$$

there exists a deep ReLU neural network $\phi_{\boldsymbol{s},\boldsymbol{\ell}}$ on $\mathbb{I}^m$ such that

$$\sup_{\boldsymbol{y}\in\mathbb{I}^m} \left| \boldsymbol{y}^{\boldsymbol{\ell}} - \phi_{\boldsymbol{s},\boldsymbol{\ell}}(\boldsymbol{y}) \right| \ \le\ \delta_{\boldsymbol{s}},$$

and the size and depth of $\phi_{\boldsymbol{s},\boldsymbol{\ell}}$ are bounded as

$$W\left(\phi_{\boldsymbol{s},\boldsymbol{\ell}}\right) \le C\left(1 + |\boldsymbol{\ell}|_1 \log \delta_{\boldsymbol{s}}^{-1}\right)$$

and

$$L\left(\phi_{\boldsymbol{s},\boldsymbol{\ell}}\right) \le C\left(1 + \log |\boldsymbol{\ell}|_1 \log \delta_{\boldsymbol{s}}^{-1}\right).$$

We define the deep ReLU neural network $\phi_{\boldsymbol{s}}$ on $\mathbb{I}^m$ by

$$\phi_{\boldsymbol{s}} := \sum_{\boldsymbol{0}\le\boldsymbol{\ell}\le\boldsymbol{s}} a_{\boldsymbol{\ell}} \phi_{\boldsymbol{s},\boldsymbol{\ell}},$$

which is a parallelization of component networks $\phi_{\boldsymbol{s},\boldsymbol{\ell}}$. We define $\boldsymbol{\phi}_{\Lambda(\xi)} := (\phi_{\boldsymbol{s}})_{\boldsymbol{s}\in\Lambda(\xi)}$ as the deep ReLU neural network realized by parallelization $\phi_{\boldsymbol{s}}$, $\boldsymbol{s} \in \Lambda(\xi)$. Consider the approximation of $S^{\omega}_{\Lambda(\xi)} v$ by $\Phi_{\Lambda(\xi)} v$. By the same way as the proof of Lemma 3.5, we can prove

$$\left\| S_{\Lambda(\xi)} v - \Phi_{\Lambda(\xi)} v \right\|_{\mathcal{L}_2(X)} \leq C \xi^{-1/q}, \tag{4.5}$$

where the constant C is independent of v and ξ.

Let us check the claims (i)–(iv) formulated at the begining of the proof. From (4.4) and (4.5) we deduce the claim (iv). The proof of the claim (i)–(iii) repeates the proof of Lemma 3.7 in Appendix with a slight modification. We indicate some particular differences in the proofs. There are no longer the fourth term in the right-hand side of (3.31) and the third term in the right-hand side of (3.36). Lemma A.3 in Appendix which is used in the proof follows from Lemma A.4 in Appendix. Lemma A.1(ii) in Appendix and the inequality (3.24) are replaced by the stronger Lemma A.5 and inequality (A.5) in Appendix. This helps us to receive the improved bound $L(\phi_v) \leq C(\log \xi)^2$. ∎

4.2 Application to parameterized elliptic PDEs with affine inputs

We now apply Theorem 4.1 to the solution $u(\boldsymbol{y})$ to the parameterized elliptic PDEs (1.1) with affine inputs (1.4).

Theorem 4.2 *Let $0 < q < \infty$, $c_k^{a,b}$ be defined as in (4.2) and let $(\delta_j)_{j\in\mathbb{N}}$ be a sequence of numbers strictly larger than 1 such that $(\delta_j^{-1})_{j\in\mathbb{N}} \in \ell_q(\mathbb{N})$. Let $\bar{a} \in L_\infty(D)$ and $\operatorname{ess\,inf} \bar{a} > 0$. Assume that there exists a sequence of positive numbers $(\rho_j)_{j\in\mathbb{N}}$ such that $c_k^{a,b} \rho_j^{-k} < \delta_j^{-k}$, $k, j \in \mathbb{N}$, and*

$$\left\| \frac{\sum_{j\in\mathbb{N}} \rho_j |\psi_j|}{\bar{a}} \right\|_{L_\infty(D)} < 1\ . \tag{4.6}$$

Then for every integer $n > 1$, we can construct a deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi_n)} := (\phi_{\boldsymbol{s}})_{\boldsymbol{s}\in\Lambda(\xi_n)}$ on $\mathbb{R}^m$ with $m := \lfloor K \frac{n}{\log n} \rfloor$, having the following properties.

- (i) *The deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi_n)}$ is independent of u;*
- (ii) *The input and output dimensions of $\boldsymbol{\phi}_{\Lambda(\xi_n)}$ are at most m;*
- (iii) $W\big(\boldsymbol{\phi}_{\Lambda(\xi_n)}\big) \leq n$;
- (iv) $L\big(\boldsymbol{\phi}_{\Lambda(\xi_n)}\big) \leq C(\log n)^2$;
- (v) *The approximation of u by $\Phi_{\Lambda(\xi_n)} u = \sum_{\boldsymbol{s}\in\Lambda(\xi_n)} u_{\boldsymbol{s}} \phi_{\boldsymbol{s}}$, where $u_{\boldsymbol{s}}$, $\boldsymbol{s} \in \mathbb{F}$, are the Jacobi gpc expansion coefficients of $u \in \mathcal{L}_2(V)$, gives the error estimate*

$$\|u - \Phi_{\Lambda(\xi_n)} u\|_{\mathcal{L}_2(V)} \leq C \left(\frac{n}{\log n} \right)^{-1/q}.$$

Here the constants C and K are independent of u and n.

Proof. It has been proven in [5] that under the assumptions of the theorem, for the sequence $(\sigma_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathbb{F}}$ given as in (4.3),

$$\sum_{\boldsymbol{s}\in\mathbb{F}}(\sigma_{\boldsymbol{s}}\|u_{\boldsymbol{s}}\|_V)^2 < \infty.$$

This means that Assumption B holds for $v = u$ with $X = V$. Hence, applying Theorem 4.1 to u, we prove the theorem. ∎

We next discuss the approximation by deep ReLU neural networks for parameterized elliptic PDEs with affine inputs and error measured in the uniform norm of $L_\infty(\mathbb{I}^\infty, V)$ by using m-term truncations of the Taylor gpc expansion of u.

If for the sequence $(\rho_j)_{j\in\mathbb{N}}$ of numbers strictly larger than 1 we have the condition 4.6 and if $(\rho_j^{-1})_{j\in\mathbb{N}} \in \ell_q(\mathbb{N})$ for some $0 < q < 2$, then the solution u to the parameterized elliptic PDEs (1.1) with affine inputs (1.4) can be decomposed in the Taylor gpc expansion

$$u = \sum_{\boldsymbol{s}\in\mathbb{F}} t_{\boldsymbol{s}}\boldsymbol{y}^{\boldsymbol{s}}, \qquad t_{\boldsymbol{s}} = \frac{1}{\boldsymbol{s}!}\partial^{\boldsymbol{s}}u(\boldsymbol{0})$$

with

$$\left(\sum_{\boldsymbol{s}\in\mathbb{F}}(\sigma_{\boldsymbol{s}}\|t_{\boldsymbol{s}}\|_V)^2\right)^{1/2} \le C \ < \infty,$$

where

$$\sigma_{\boldsymbol{s}} := \prod_{j\in\mathbb{N}} \rho_j^{s_j},$$

see [5, Theorem 2.1]. Moreover, the sequence $(\|t_{\boldsymbol{s}}\|_V)_{\boldsymbol{s}\in\mathbb{F}}$ is ℓ_p-summable with $p = \frac{2q}{2+q} < 1$. We define

$$S_{\Lambda(\xi)}v := \sum_{\boldsymbol{s}\in\Lambda(\xi)} t_{\boldsymbol{s}}\boldsymbol{y}_{\boldsymbol{s}},$$

where $\Lambda(\xi)$ is given by the formula (3.2). The following theorem is an improvement of [46, Theorem 3.9].

Theorem 4.3 *Let $\bar{a} \in L_\infty(D)$ and $\operatorname{ess\,inf} \bar{a} > 0$. Assume that there exists an increasing sequence $(\rho_j)_{j\in\mathbb{N}}$ of numbers strictly larger than 1 such that the sequence $(\rho_j^{-1})_{j\in\mathbb{N}} \in \ell_q(\mathbb{N})$ for some q with $0 < q < 2$, and there holds the condition (4.6). Then for every integer $n > 1$, we can construct a deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi_n)} := (\phi_{\boldsymbol{s}})_{\boldsymbol{s}\in\Lambda(\xi_n)}$ on $\mathbb{R}^m$ with $m := \left\lfloor K\frac{n}{\log n}\right\rfloor$, having the following properties.*

(i) *The deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi_n)}$ is independent of u;*

(ii) *The input and output dimensions of $\boldsymbol{\phi}_{\Lambda(\xi_n)}$ are at most m;*

(iii) $W\big(\boldsymbol{\phi}_{\Lambda(\xi_n)}\big) \le n$;

(iv) $L\big(\boldsymbol{\phi}_{\Lambda(\xi_n)}\big) \le C\log n \log\log n$;

(v) *The approximation of u by $\Phi_{\Lambda(\xi_n)}u := \sum_{\boldsymbol{s}\in\Lambda(\xi_n)} t_{\boldsymbol{s}}\phi_{\boldsymbol{s}}$ gives the error estimate*

$$\|u - \Phi_{\Lambda(\xi_n)}u\|_{L_\infty(\mathbb{I}^\infty,V)} \le C\left(\frac{n}{\log n}\right)^{-(1/q-1/2)}.$$

Here the constants C and K are independent of u and n.

Proof. This theorem can be proven in a way similar to the proof of Theorem 4.2. Let us give a brief proof. Given $\xi \ge 3$, we have the Caushy-Schwarz inequality and Lemma A.4 in Appendix that

$$\begin{aligned} \|u - S_{\Lambda(\xi)}u\|_{L_\infty(\mathbb{I}^\infty,V)} &\le \sum_{\boldsymbol{s}\notin\Lambda(\xi)} \|t_{\boldsymbol{s}}\|_V \le \left(\sum_{\sigma_{\boldsymbol{s}}>\xi^{1/q}} (\sigma_{\boldsymbol{s}}\|t_{\boldsymbol{s}}\|_V)^2\right)^{1/2} \left(\sum_{\sigma_{\boldsymbol{s}}>\xi^{1/q}} \sigma_{\boldsymbol{s}}^{-2}\right)^{1/2} \\ &\le C\left(\sum_{\sigma_{\boldsymbol{s}}>\xi^{1/q}} \sigma_{\boldsymbol{s}}^{-q}\sigma_{\boldsymbol{s}}^{-(2-q)}\right)^{1/2} \\ &\le C\xi^{-(1/q-1/2)}\left(\sum_{\boldsymbol{s}\in\mathbb{N}_0^m} \sigma_{\boldsymbol{s}}^{-q}\right)^{1/2} \le C\xi^{-(1/q-1/2)}. \end{aligned} \tag{4.7}$$

Put $\delta := \xi^{-(1/q-1/2)}$. For every $\boldsymbol{s} \in \Lambda(\xi)$, by Lemma 2.5 there exists a deep ReLU neural network $\phi_{\boldsymbol{s}}$ on $\mathbb{I}^m$ such that

$$\sup_{\boldsymbol{y}\in\mathbb{I}^m} |\boldsymbol{y}^{\boldsymbol{s}} - \phi_{\boldsymbol{s}}(\boldsymbol{y})| \ \le \ \delta,$$

and the size and depth of $\phi_{\boldsymbol{s}}$ are bounded as

$$W(\phi_{\boldsymbol{s}}) \le C\left(1 + |\boldsymbol{s}|_1 \log\delta^{-1}\right) \le C\big(1 + |\boldsymbol{s}|_1 \log\xi\big)$$

and

$$L(\phi_{\boldsymbol{s}}) \le C\left(1 + \log|\boldsymbol{s}|_1 \log\delta^{-1}\right) \le C\big(1 + \log|\boldsymbol{s}|_1 \log\xi\big).$$

We define $\boldsymbol{\phi}_{\Lambda(\xi)} := (\phi_{\boldsymbol{s}})_{\boldsymbol{s}\in\Lambda(\xi)}$ as the deep ReLU neural network realized by parallelization of $\phi_{\boldsymbol{s}}$, $\boldsymbol{s} \in \Lambda(\xi)$. Consider the approximation of u by

$$\Phi_{\Lambda(\xi)}u := \sum_{\boldsymbol{s}\in\Lambda(\xi)} t_{\boldsymbol{s}}\phi_{\boldsymbol{s}}(\boldsymbol{y}).$$

Then by the inclusion $(\|t_{\boldsymbol{s}}\|_V)_{\boldsymbol{s}\in\mathbb{F}} \in \ell_p(\mathbb{F})$, $p \in (0,1)$ and (4.7), we have

$$\begin{aligned} \|u - \Phi_{\Lambda(\xi)}u\|_{L_\infty(\mathbb{I}^\infty,V)} &\le \|u - S_{\Lambda(\xi)}u\|_{L_\infty(\mathbb{I}^\infty,V)} + \left\|S_{\Lambda(\xi)}u - \Phi_{\Lambda(\xi)}u\right\|_{L_\infty(\mathbb{I}^\infty,V)} \\ &\le \ C\xi^{-(1/q-1/2)} + \sum_{\boldsymbol{s}\in\Lambda(\xi)} \|t_{\boldsymbol{s}}\|_V \|\boldsymbol{y}_{\boldsymbol{s}} - \phi_{\boldsymbol{s}}\|_{L_\infty(\mathbb{I}^\infty,V)} \\ &\le \ C\xi^{-(1/q-1/2)} + C\xi^{-(1/q-1/2)} \sum_{\boldsymbol{s}\in\Lambda(\xi)} \|t_{\boldsymbol{s}}\|_V \le C\xi^{-(1/q-1/2)}, \end{aligned}$$

where the constants C may be different and are independent of u and ξ. By the construction of $\boldsymbol{\phi}_{\Lambda(\xi)}$ we have

$$\begin{aligned} W(\boldsymbol{\phi}_{\Lambda(\xi)}) &\le \sum_{\boldsymbol{s}\in\Lambda(\xi)} \le W(\phi_{\boldsymbol{s}}) \le \sum_{\boldsymbol{s}\in\Lambda(\xi)} C\big(1 + |\boldsymbol{s}|_1 \log\xi\big) \le C\Bigg(|\Lambda(\xi)| + \log\xi \sum_{\sigma_{\boldsymbol{s}}^q \le \xi} p_{\boldsymbol{s}}(1)\Bigg) \\ &\le C\Bigg(|\Lambda(\xi)| + \log\xi \sum_{\sigma_{\boldsymbol{s}}^q\le\xi} p_{\boldsymbol{s}}(1)\sigma_{\boldsymbol{s}}^{-q}\sigma_{\boldsymbol{s}}^q\Bigg) \le C\xi\log\xi \end{aligned}$$

where in the last estimate we used Lemmas A.1(i) and A.4 in Appendix. Similarly, we have

$$L(\boldsymbol{\phi}_{\Lambda(\xi)}) \le \max_{\boldsymbol{s}\in\Lambda(\xi)} L(\phi_{\boldsymbol{s}}) \le C \max_{\boldsymbol{s}\in\Lambda(\xi)} \big(1+\log|\boldsymbol{s}|_1 \log\xi\big) \le C \log\xi \log\log\xi,$$

see Lemma A.5 in Appendix. Now following argument at the end of the proof of Theorem 3.10, we obtain the existence of ξ_n for a given $n > 1$. ∎

5 Concluding remarks

We have established bounds in terms of the size n of deep ReLU neural networks for error of approximation of the solution u to parametric and stochastic elliptic PDEs with lognormal inputs by them. The method of this approximation is as follows. For given $n \in \mathbb{N}$, $n > 1$ we construct a compactly supported deep ReLU neural network $\boldsymbol{\phi}_n := (\phi_j)_{j=1}^m$ of the size $\le n$ on $\mathbb{R}^m$, $m = \mathcal{O}(n/\log n)$, with m outputs to approximate the m-term truncation of the Hermite gpc expansion $\sum_{j=1}^m u_{\boldsymbol{s}^j} H_{\boldsymbol{s}^j}$ of u by $u_n := \sum_{j=1}^m u_{\boldsymbol{s}^j}\phi_j$. We proved that the extension of u_n to $\mathbb{R}^\infty$ approximates u with the error bound $\mathcal{O}\big((n/\log n)^{-1/q}\big)$, and that the depth of $\boldsymbol{\phi}_n$ is $\mathcal{O}\left(n^\delta\right)$ for any $\delta > 0$. We also obtained similar results for approximation by deep ReLU neural networks of solution to parametric and stochastic elliptic PDEs with affine inputs. These results based on an m-term truncation of the Jacobi and Taylor gpc expansions of the solution.

In the present paper, we have been concerned about the parametric approximability for parametric and stochastic elliptic PDEs. Therefore, the results themselves do not yield a practically realizable approximation since they do not cover the approximation of the gpc expansion coefficients which are functions of the spatial variable. Naturally, it would be desirable to study the problem of fully discrete approximation of the solution u to parametric and stochastic elliptic PDEs as in [3, 14] by deep ReLU neural networks. We will discuss this problem in a forthcoming paper.

A Appendix: Auxiliary results

Lemma A.1 *Let $\theta > 0$, $\xi > 1$ and $(\sigma_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathbb{F}}$ be a sequence of numbers strictly larger than 1. Then we have the following.*

(i) *Assume that $\left(\sigma_{\boldsymbol{s}}^{-1}\right)_{\boldsymbol{s}\in\mathbb{F}} \in \ell_q(\mathbb{F})$. The set $\Lambda(\xi)$ is finite and it holds*

$$|\Lambda(\xi)| \le K_q \xi,$$

where $K_q := \sum_{\boldsymbol{s}\in\mathbb{F}} \sigma_{\boldsymbol{s}}^{-q} < \infty$.

(ii) *Assume that $\left(p_{\boldsymbol{s}}(\theta)\sigma_{\boldsymbol{s}}^{-1}\right)_{\boldsymbol{s}\in\mathbb{F}} \in \ell_q(\mathbb{F})$ for some $\theta > 0$. There holds*

$$m_1(\xi) \le K_{q,\theta}\xi^{\frac{1}{\theta q}},$$

where $K_{q,\theta} := \left(\sum_{\boldsymbol{s}\in\mathbb{F}} p_{\boldsymbol{s}}(\theta)^q \sigma_{\boldsymbol{s}}^{-q}\right)^{\frac{1}{\theta q}} < \infty$.

Proof. Notice that $1 \leq \sigma_{\boldsymbol{s}}^{-q}\xi$ for every $\boldsymbol{s} \in \Lambda(\xi)$. This implies (i):

$$|\Lambda(\xi)| \;=\; \sum_{\boldsymbol{s}\in\Lambda(\xi)} 1 \leq \sum_{\boldsymbol{s}\in\Lambda(\xi)} \xi\sigma_{\boldsymbol{s}}^{-q} \leq K_q\xi$$

Moreover, we have that $1 \leq s_j$ for every $j \in \nu_{\boldsymbol{s}}$. Hence, we derive the inequality

$$\max_{\boldsymbol{s}\in\Lambda(\xi)} |\boldsymbol{s}|_1^{\theta q} \leq \sum_{\boldsymbol{s}\in\Lambda(\xi)} \left(\prod_{j\in\nu_{\boldsymbol{s}}} (1+s_j)\right)^{\theta q} \leq \sum_{\boldsymbol{s}\in\Lambda(\xi)} p_{\boldsymbol{s}}(\theta)^q \xi \sigma_{\boldsymbol{s}}^{-q} \leq K_{q,\theta}^{\theta q}\xi$$

which prove (ii). ∎

By this definition we have

$$\bigcup_{\boldsymbol{s}\in\Lambda(\xi)} \nu_{\boldsymbol{s}} \subset \{1,2,\ldots,m(\xi)\} \tag{A.1}$$

Lemma A.2 *Let $\theta > 0$, $0 < q < \infty$ and $(\sigma_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathbb{F}}$ be an increasing sequence of numbers strictly larger than 1. Assume that $\left(\sigma_{\boldsymbol{s}}^{-1}\right)_{\boldsymbol{s}\in\mathbb{F}} \in \ell_q(\mathbb{F})$ and $\sigma_{\boldsymbol{e}^{i'}} \leq \sigma_{\boldsymbol{e}^i}$ if $i' < i$. Then there holds*

$$m(\xi) \leq K_q\xi, \tag{A.2}$$

where K_q is the constant given in Lemma A.1(i).

Proof. Noting (3.4), there is a $\boldsymbol{s} \in \Lambda(\xi)$ such that $s_{m(\xi)} > 0$. Then we have $\boldsymbol{e}^{m(\xi)} \leq \boldsymbol{s}$. Since $\Lambda(\xi)$ is downward closed, we have $\boldsymbol{e}^{m(\xi)} \in \Lambda(\xi)$. From the definition (3.2) of $\Lambda(\xi)$ and the assumption in the lemma, we obtain

$$\sigma_{\boldsymbol{e}^1}^q \leq \sigma_{\boldsymbol{e}^2}^q \leq \ldots \leq \sigma_{\boldsymbol{e}^{m(\xi)}}^q \leq \xi.$$

Thus, $\boldsymbol{e}^1,\ldots,\boldsymbol{e}^{m(\xi)}$ belong to $\Lambda(\xi)$. This yields the inequality $|\Lambda(\xi)| \geq m(\xi)$ which together with the inequality $|\Lambda(\xi)| \leq K_q\xi$ in Lemma A.1(i) proves (A.2). The inclusion (A.1) can then be obtained directly from (3.4). ∎

Lemma A.3 *Let $\theta \geq 0$, $0 < q < \infty$, and $\Lambda^*(\xi)$ be defined in (3.27). Assume that $\left(p_{\boldsymbol{s}}\left(\frac{\theta+1}{q},1\right)\sigma_{\boldsymbol{s}}^{-1}\right)_{\boldsymbol{s}\in\mathbb{F}} \in \ell_q(\mathbb{F})$. There holds*

$$\sum_{(\boldsymbol{s},\boldsymbol{\ell})\in\Lambda^*(\xi)} p_{\boldsymbol{s}}(\theta) \leq C\xi.$$

Proof. We have

$$\begin{aligned}\sum_{(\boldsymbol{s},\boldsymbol{\ell})\in\Lambda^*(\xi)} p_{\boldsymbol{s}}(\theta) &= \sum_{\boldsymbol{s}\in\Lambda(\xi)} \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}} p_{\boldsymbol{s}}(\theta) \leq \xi \sum_{\sigma_{\boldsymbol{s}}^{-q}\xi\geq 1} \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}} p_{\boldsymbol{s}}(\theta)\sigma_{\boldsymbol{s}}^{-q} \\ &= \xi \sum_{\sigma_{\boldsymbol{s}}^{-q}\xi\geq 1} \left(\prod_{j=1}^{m}(1+s_j)\right) p_{\boldsymbol{s}}(\theta)\sigma_{\boldsymbol{s}}^{-q} \leq \xi \sum_{\boldsymbol{s}\in\mathbb{F}} p_{\boldsymbol{s}}(\theta+1)\sigma_{\boldsymbol{s}}^{-q} \leq C\xi.\end{aligned}$$

∎

The following lemma is a direct consequence of [14, Lemma 6.2].

Lemma A.4 *Let $0 < q < \infty$ and θ and λ be arbitrary nonnegative real numbers. Assume that $\boldsymbol{\rho} = (\rho_j)_{j\in\mathbb{N}}$ be a sequence of numbers strictly larger than 1 such that $(\rho_j^{-1})_{j\in\mathbb{N}} \in \ell_q(\mathbb{N})$. Then for the sequences $(\sigma_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathbb{F}}$ and $(p_{\boldsymbol{s}}(\theta,\lambda))_{\boldsymbol{s}\in\mathbb{F}}$ given as in (4.3) and (3.7), respectively, we have*

$$\sum_{\boldsymbol{s}\in\mathbb{F}} p_{\boldsymbol{s}}(\theta,\lambda)\sigma_{\boldsymbol{s}}^{-q} < \infty.$$

Proof. Notice that $c_s^{a,b} \le (1+\lambda' s)^{\theta'}$ for $s \in \mathbb{N}_0$ with some $\lambda' > 0$ and $\theta' > 0$ depending on a, b. Hence, for any $\theta, \lambda \ge 0$, we get

$$p_{\boldsymbol{s}}(\theta,\lambda)\sigma_{\boldsymbol{s}}^{-q} = p_{\boldsymbol{s}}(\theta,\lambda)c_{\boldsymbol{s}}^{q}(\boldsymbol{\rho}^{-\boldsymbol{s}})^q \le p_{\boldsymbol{s}}(\theta,\lambda)p_{\boldsymbol{s}}(q\theta',\lambda')(\boldsymbol{\rho}^{-\boldsymbol{s}})^q \le p_{\boldsymbol{s}}(\theta^*,\lambda^*)(\boldsymbol{\rho}^{-\boldsymbol{s}})^q,$$

where $\theta^* := \theta + q\theta'$ and $\lambda^* := \max(\lambda,\lambda')$. We derive that

$$\sum_{\boldsymbol{s}\in\mathbb{F}} p_{\boldsymbol{s}}(\theta,\lambda)\sigma_{\boldsymbol{s}}^{-q} \le \sum_{\boldsymbol{s}\in\mathbb{F}} p_{\boldsymbol{s}}(\theta^*,\lambda^*)(\boldsymbol{\rho}^{-\boldsymbol{s}})^q.$$

Now applying [14, Lemma 6.2] to the right-hand side we obtain the desired result. $\square$

Lemma A.5 *Let $0 < q < \infty$, $c_k^{a,b}$ be defined as in (4.2) and let $(\delta_j)_{j\in\mathbb{N}}$ be a sequence of numbers strictly larger than 1 such that $(\delta_j^{-1})_{j\in\mathbb{N}} \in \ell_q(\mathbb{N})$. Assume that there exists a sequence of positive number $(\rho_j)_{j\in\mathbb{N}}$ such that $c_k^{a,b}\rho_j^{-k} < \delta_j^{-k}$, $k, j \in \mathbb{N}$. For the sequence $(\sigma_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathbb{F}}$ given as in (4.3), and $\xi > 1$, let $m_1(\xi)$ be the number defined by (3.3). Then we have for every $\xi > 1$,*

$$m_1(\xi) \le C\log\xi, \tag{A.3}$$

with the constant C independent of ξ.

Proof. The proof relies on Lemma A.1 and a technique from the proof of [46, Lemma 2.8(ii)]. Fix a number p satisfying $0 < p < q$ and let the sequence $(\beta_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathbb{F}}$ be given by

$$\beta_{\boldsymbol{s}}^{-1} := \begin{cases} \max(\sigma_{\boldsymbol{s}}^{-1}, j^{-1/p}) & \text{if } \boldsymbol{s} = \boldsymbol{e}^j, \\ \sigma_{\boldsymbol{s}}^{-1} & \text{otherwise.} \end{cases}$$

Notice that the sequence $(\alpha_{\boldsymbol{s}}^{-1})_{\boldsymbol{s}\in\mathbb{F}}$ defined by

$$\alpha_{\boldsymbol{s}}^{-1} := \begin{cases} j^{-1/p} & \text{if } \boldsymbol{s} = \boldsymbol{e}^j, \\ 0 & \text{otherwise,} \end{cases}$$

belongs to $\ell_q(\mathbb{F})$. On the other hand, from Lemma A.4 one can see that the sequence $(\sigma_{\boldsymbol{s}}^{-1})_{\boldsymbol{s}\in\mathbb{F}}$ belongs to $\ell_q(\mathbb{F})$. This implies that the sequence $(\beta_{\boldsymbol{s}}^{-1})_{\boldsymbol{s}\in\mathbb{F}}$ belongs to $\ell_q(\mathbb{F})$. Hence, by Lemma A.1 the set $\Lambda_\beta(\xi) := \{\boldsymbol{s} \in \mathbb{F} : \beta_{\boldsymbol{s}}^q \le \xi\}$ is finite. Notice also that $(\beta_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathbb{F}}$ is increasing and $\Lambda_\beta(\xi)$ is downward closed. Put $n := |\Lambda_\beta(\xi)|$. Then the set $\Lambda_\beta(\xi)$contains n largest elements of $(\beta_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathbb{F}}$. Therefore by the construction of $(\beta_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathbb{F}}$ we have

$$\min_{\boldsymbol{s}\in\Lambda_\beta(\xi)} \beta_{\boldsymbol{s}}^{-1} = \beta_{\boldsymbol{s}_n}^{-1} \ge n^{-1/p}.$$

Since $c_k^{a,b}\rho_j^{-k} \le \delta_j^{-k}$, $k, j \in \mathbb{N}$ and $(\delta_j)_{j\in\mathbb{N}}$ be a sequence of numbers strictly larger than 1 and $(\delta_j^{-1})_{j\in\mathbb{N}} \in \ell_q(\mathbb{N})$, there exists $\delta < 1$ such that $c_k^{a,b}\rho_j^{-k} \le \delta$ for $k, j \in \mathbb{N}$. Therefore have for $r > 1$,

$$\sup_{|\boldsymbol{s}|_1 = r} \beta_{\boldsymbol{s}}^{-1} = \sup_{|\boldsymbol{s}|_1 = r} \sigma_{\boldsymbol{s}}^{-1} \le \delta^r.$$

Let $\bar{r} > 1$ be an integer such that $n^{-1/p} > \delta^{\bar{r}}$. Then one can see that

$$\max_{\boldsymbol{s}\in\Lambda_\beta(\xi)} |\boldsymbol{s}|_1 < \bar{r}.$$

For the function $g(t) := \delta^t$, its inverse is defined as $g^{-1}(x) = \frac{\log x}{\log \delta}$. Hence we get $\bar{r} < g^{-1}(n^{-1/p})$, and consequently,

$$\max_{\boldsymbol{s}\in\Lambda_\beta(\xi)} |\boldsymbol{s}|_1 < g^{-1}(n^{-1/p}) \le C \log n = C \log |\Lambda_\beta(\xi)|.$$

By Lemma A.1 we obtain the inequality $|\Lambda_\beta(\xi)| \le C\xi$ which together with the inclusion $\Lambda(\xi) \subset \Lambda_\beta(\xi)$ proves (A.3). □

Lemma A.6 *Let the Jacobi polynomial J_s be written in the form*

$$J_s(y) = \sum_{\ell=0}^{s} a_{s,\ell} y^\ell, \tag{A.4}$$

then

$$\sum_{\ell=0}^{s} |a_{s,\ell}| \le K_{a+b} 9^s. \tag{A.5}$$

Proof. It is well-known that for each $s \in \mathbb{N}$, the univariate Jacobi polynomial J_s can be written as

$$J_s(y) = \frac{\Gamma(a+s+1)}{s!\Gamma(a+b+s+1)} \sum_{m=0}^{s} \binom{s}{m} \frac{\Gamma(a+b+s+m+1)}{\Gamma(a+m+1)} \left(\frac{y-1}{2}\right)^m,$$

where Γ is the gamma function. Putting

$$A_m := 2^{-m}\binom{s}{m}\frac{\Gamma(a+b+s+m+1)}{\Gamma(a+m+1)}, \quad B_s := \frac{\Gamma(a+s+1)}{s!\Gamma(a+b+s+1)},$$

we have

$$\begin{aligned} J_s(y) &= B_s \sum_{m=0}^{s} A_m (y-1)^m = B_s \sum_{m=0}^{s} A_m \sum_{\ell=0}^{m} \binom{m}{\ell} (-1)^{m-\ell} y^\ell \\ &= B_s \sum_{\ell=0}^{s}\sum_{m=\ell}^{s} A_m \binom{m}{\ell} (-1)^{m-\ell} y^\ell. \end{aligned}$$

Hence

$$\sum_{\ell=0}^{s} |a_{s,\ell}| \le B_s \sum_{\ell=0}^{s}\sum_{m=\ell}^{s} A_m \binom{m}{\ell} = B_s \sum_{m=0}^{s} A_m \sum_{\ell=0}^{m} \binom{m}{\ell} = B_s \sum_{m=0}^{s} 2^m A_m. \tag{A.6}$$

Let

$$B(x,y) := \int_0^1 t^{x-1}(1-t)^{y-1} \mathrm{d}t = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$$

be the beta function. It is decreasing in x and in y. Hence for $m \le s$,

$$\frac{\Gamma(a+b+s+m+1)}{\Gamma(a+m+1)} = \frac{\Gamma(b+s)}{B(a+m+1,b+s)} \le \frac{\Gamma(b+s)}{B(a+s+1,b+s)} = \frac{\Gamma(a+b+2s+1)}{\Gamma(a+s+1)}.$$

This together with (A.6) gives

$$\sum_{\ell=0}^{s}|a_{s,\ell}|\le B_s\frac{\Gamma(a+b+2s+1)}{\Gamma(a+s+1)}\sum_{m=0}^{s}\binom{s}{m}=2^s\frac{\Gamma(a+b+2s+1)}{s!\Gamma(a+b+s+1)}=\frac{2^s}{B(s,a+b+s+1)}. \tag{A.7}$$

Since the beta function is decreasing in each variable, it is enough to estimate the right-hand side for $a+b\ge 0$. By using Stirling's formula for the beta function $B(x,y)\sim\sqrt{2\pi}\frac{x^{x-1/2}y^{y-1/2}}{(x+y)^{x+y-1/2}}$, from (A.7) we get for every $s\in\mathbb{N}$,

$$\begin{aligned}\sum_{\ell=0}^{s}|a_{s,\ell}|&\le C\frac{2^s(a+b+2s+1)^{a+b+2s+1/2}}{s^{s-1/2}(a+b+s+1)^{a+b+s+1/2}}\le C\frac{2^s(2(a+b+s+1))^{a+b+2s+1/2}}{s^{s-1/2}(a+b+s+1)^{a+b+s+1/2}}\\&\le C2^{a+b+3s+1/2}\frac{(a+b+s+1)^s}{s^{s-1/2}}\\&\le C8^s\sqrt{s}\left(\frac{a+b+1}{s}+1\right)^s\le K_{a+b}9^s.\end{aligned}$$

□

Acknowledgments. The work of Dinh Dũng and Van Kien Nguyen is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under Grant No. 102.01-2020.03. A part of this work was done when the authors were working at the Vietnam Institute for Advanced Study in Mathematics (VIASM). They would like to thank the VIASM for providing a fruitful research environment and working condition.

References

[1] M. Ali and A. Nouy. Approximation of smoothness classes by deep ReLU networks. *arXiv:2007.15645*, 2020.

[2] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee. Understanding deep neural networks with rectified linear units. *Electronic Colloquium on Computational Complexity*, Report No. 98, 2017.

[3] M. Bachmayr, A. Cohen, D. Dũng, and C. Schwab. Fully discrete approximation of parametric and stochatic elliptic PDEs. *SIAM J. Numer. Anal.*, 55:2151–2186, 2017.

[4] M. Bachmayr, A. Cohen, R. DeVore, and G. Migliorati. Sparse polynomial approximation of parametric elliptic PDEs. Part II: lognormal coefficients. *ESAIM Math. Model. Numer. Anal.*, 51:341 – 363, 2017.

[5] M. Bachmayr, A. Cohen, and G. Migliorati. Sparse polynomial approximation of parametric elliptic PDEs. Part I: affine coefficients. *ESAIM Math. Model. Numer. Anal.*, 51:321–339, 2017.

[6] A. R. Barron. Complexity regularization with application to artificial neural networks. *In Nonparametric Functional Estimation and Related Topics. NATO ASI Series (Series C: Mathematical and Physical Sciences)*, 335:561–576, 1991.

[7] A. Chkifa, A. Cohen, R. DeVore, and C. Schwab. Sparse adaptive Taylor approximation algorithms forparametric and stochastic elliptic PDEs. *ESAIM Math. Model. Numer. Anal.*, 47:253–280, 2013.

[8] A. Chkifa, A. Cohen, and C. Schwab. High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs. *Found. Comput. Math.*, 14(4):601–633, 2013.

[9] A. Chkifa, A. Cohen, and C. Schwab. Breaking the curse of dimensionality in sparse polynomial approximation of parametric PDEs. *J. Math. Pures Appl.*, 103:400–428., 2015.

[10] A. Cohen and R. DeVore. Approximation of high-dimensional parametric PDEs. *Acta Numer.*, 24:1–159, 2015.

[11] A. Cohen, R. DeVore, and C. Schwab. Convergence rates of best N-term Galerkin approximations for a class of elliptic sPDEs. *Found. Comput. Math.*, 9:615–646, 2010.

[12] A. Cohen, R. DeVore, and C. Schwab. Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDE's. *Anal. Appl.*, 9:11–47, 2011.

[13] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Math. Control. Signals, Syst.*, 2:303–314, 1989.

[14] D. Dũng. Collocation approximation by deep neural ReLU networks for parametric elliptic PDEs with lognormal inputs. *arXiv:2111.05504*.

[15] D. Dũng. Linear collocation approximation for parametric and stochastic elliptic PDEs. *Mat. Sb.*, 210:103–227, 2019.

[16] D. Dũng and V. K. Nguyen. Deep ReLU neural networks in high-dimensional approximation. *Neural Netw.*, 142:619–635, 2021.

[17] D. Dũng, V. K. Nguyen, and M. X. Thao. Computation complexity of deep ReLU neural networks in high-dimensional approximation. *Journal of Computer Science and Cybernetics*, 37:291–320, 2021.

[18] I. Daubechies, R. DeVore, S. Foucart, B. Hanin, and G. Petrova. Nonlinear approximation and (Deep) ReLU networks. *Constr. Approx.*, 2021, https://doi.org/10.1007/s00365-021-09548-z.

[19] R. DeVore, B. Hanin, and G. Petrova. Neural network approximation. *Acta Numer.*, pages 327–444, 2021.

[20] W. E and Q. Wang. Exponential convergence of the deep neural network approximation for analytic functions. *Sci. China Math.*, 61:1733–1740, 2018.

[21] D. Elbrächter, P. Grohs, A. Jentzen, and C. Schwab. DNN expression rate analysis of high-dimensional PDEs: application to option pricing. *Constr. Approx.*, 2021, https://doi.org/10.1007/s00365-021-09541-6.

[22] O. G. Ernst, B. Sprungk, and L. Tamellini. Convergence of sparse collocation for functions of countably many Gaussian random variables (with application to elliptic PDEs). *SIAM J. Numer. Anal.*, 56(2):877–905, 2018.

[23] K.-I. Funahashi. Approximate realization of identity mappings by three-layer neural networks. *Electron. Commun. Jpn 3*, 73:61–68, 1990.

[24] M. Geist, P. C. Petersen, M. Raslan, R. Schneider, and G. Kutyniok. Numerical solution of the parametric diffusion equation by deep neural networks. *J. Sci. Comput.*, 88, 2021, https://doi.org/10.1007/s10915-021-01532-w.

[25] L. Gonon and C. Schwab. Deep ReLU network expression rates for option prices in high-dimensional, exponential Lévy models. *Finance Stoch.*, 25:615–657, 2021.

[26] L. Gonon and C. Schwab. Deep ReLU neural network approximation for stochastic differential equations with jumps. Technical Report 2021-08, Seminar for Applied Mathematics, ETH Zürich, 2021.

[27] R. Gribonval, Kutyniok, M. Nielsen, and F. Voigtländer. Approximation spaces of deep neural networks. *Constr. Approx.*, 2021, https://doi.org/10.1007/s00365-021-09543-4.

[28] P. Grohs and L. Herrmann. Deep neural network approximation for high-dimensional elliptic PDEs with boundary conditions. *IMA J. Numer. Anal.*, 2021, https://doi.org/10.1093/imanum/drab031.

[29] P. Grohs, D. Perekrestenko, D. Elbrachter, and H. Bolcskei. Deep neural network approximation theory. *IEEE Trans. Inf. Theory*, 67:2581–2623, 2021.

[30] I. Gühring, G. Kutyniok, and P. Petersen. Error bounds for approximations with deep ReLU neural networks in $W^{s,p}$ norms. *Anal. Appl. (Singap.)*, 18:803–859, 2020.

[31] L. Herrmann, J. A. A. Opschoor, and C. Schwab. Constructive deep ReLU neural network approximation. Technical Report 2021-04, Seminar for Applied Mathematics, ETH Zürich, 2021.

[32] L. Herrmann, C. Schwab, and J. Zech. Deep neural network expression of posterior expectations in Bayesian PDE inversion. *Inverse Problems*, 36:125011, 2020.

[33] E. Hewitt and K. Stromberg. *Real and Abstract Analysis.* Springer, 1965.

[34] V. Hoang and C. Schwab. N-term Galerkin Wiener chaos approximation rates for elliptic PDEs with lognormal Gaussian random inputs. *Math. Models Methods Appl. Sci.*, 24:797–826, 2014.

[35] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Netw.*, 2:359–366, 1989.

[36] G. Kutyniok, P. C. Petersen, M. Raslan, and R. Schneider. A theoretical analysis of deep neural networks and parametric PDEs. *Constr. Approx.*, 2021, https://doi.org/10.1007/s00365-021-09551-4.

[37] D. S. Lubinsky. A survey of weighted polynomial approximation with exponential weights. *Surveys in Approximation Theory*, 3:1–105, 2007.

[38] H. N. Mhaskar. Neural networks for optimal approximation of smooth and analytic functions. *Neural Comput.*, 8:164–177, 1996.

[39] H. Montanelli and Q. Du. New error bounds for deep ReLU networks using sparse grids. *SIAM J. Math. Data Sci.*, 1:78–92, 2019.

[40] G. Montúfar, R. Pascanu, K. Cho, and Y. Bengio. On the number of linear regions of deep neural networks. *In Advances in neural information processing systems*, pages 2924–2932, 2014.

[41] J. A. A. Opschoor, P. C. Petersen, and C. Schwab. Deep ReLU networks and high-order finite element methods. *Anal. Appl. (Singap.)*, 18:715–770, 2020.

[42] J. A. A. Opschoor, C. Schwab, and J. Zech. Deep learning in high dimension: ReLU network expression rates for Bayesian PDE inversion. Technical Report 2020-47, Seminar for Applied Mathematics, ETH Zürich, 2020.

[43] J. A. A. Opschoor, C. Schwab, and J. Zech. Exponential ReLU DNN expression of holomorphic maps in high dimension. *Constr. Approx.*, 2021, https://doi.org/10.1007/s00365-021-09542-5.

[44] P. C. Petersen. Neural network theory. *Available at http://pc-petersen.eu/Neural_Network_Theory.pdf.*

[45] P. C. Petersen and F. Voigtländer. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *Neural Netw.*, 108:296–330, 2018.

[46] C. Schwab and J. Zech. Deep learning in high dimension: Neural network expression rates for generalized polynomial chaos expansions in UQ. *Anal. Appl. (Singap.)*, 17:19–55, 2019.

[47] J. Sirignano and K. Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations . *J. Comput. Phys.*, 375:1339–1364, 2018.

[48] T. Suzuki. Adaptivity of deep ReLU network for learning in Besov and mixed smooth Besov spaces: optimal rate and curse of dimensionality. *International Conference on Learning Representations*, 2019.

[49] M. Telgarsky. Representation benefits of deep feedforward networks. *arXiv:1509.08101*, 2015.

[50] M. Telgrasky. Benefits of depth in neural nets. *In Proceedings of the JMLR: Workshop and Conference Proceedings, New York, NY, USA*, 49:1–23, 2016.

[51] R. Tripathy and I. Bilionis. Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification. *J. Comput. Phys.*, 375:565–588, 2018.

[52] D. Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Netw.*, 94:103–114, 2017.

[53] D. Yarotsky. Optimal approximation of continuous functions by very deep ReLU networks. *Proc. Mach. Learn. Res.*, 75:1–11, 2018.

[54] J. Zech, D. Dũng, and C. Schwab. Multilevel approximation of parametric and stochastic PDES. *Math. Models Methods Appl. Sci.*, 29:1753–1817, 2019.

[55] J. Zech and C. Schwab. Convergence rates of high dimensional smolyak quadrature. *ESAIM Math. Model. Numer. Anal.*, 54:1259–307, 2020.