Collocation approximation by deep neural ReLU networks for parametric and stochastic PDEs with lognormal inputs

Dinh Dũng

Information Technology Institute, Vietnam National University, Hanoi
144 Xuan Thuy, Cau Giay, Hanoi, Vietnam
Email: dinhzung@gmail.com

January 24, 2022

Abstract

We obtained convergence rates of the collocation approximation by deep ReLU neural networks of the solution u to elliptic PDEs with lognormal inputs, parametrized by $\boldsymbol{y}$ from the 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 infinite tensor product standard Gaussian probability measure on $\mathbb{R}^\infty$ and V is the energy space. Under a certain assumption on ℓ_q-summability for the lognormal inputs ($0 < q < 2$), we proved that given arbitrary number $\delta > 0$ small enough, for every integer $n > 1$, one can construct a compactly supported deep ReLU neural network $\boldsymbol{\phi}_n := \left(\phi_j\right)_{j=1}^m$ of size at most n on $\mathbb{R}^m$ with $m = \mathcal{O}\left(n^{1-\delta}\right)$, and a sequence of points $\left(\boldsymbol{y}^j\right)_{j=1}^m \subset \mathbb{R}^m$ (which are independent of u) so that the collocation approximation of u by $\Phi_n u := \sum_{j=1}^m u\left(\boldsymbol{y}^j\right)\Phi_j$, which is based on the m solvers $\left(u\left(\boldsymbol{y}^j\right)\right)_{j=1}^m$ and the deep ReLU network $\boldsymbol{\phi}_n$, gives the twofold error bounds: $\|u - \Phi_n u\|_{L_2(\mathbb{R}^\infty, V, \gamma)} = \mathcal{O}\left(m^{-(1/q-1/2)}\right) = \mathcal{O}\left(n^{-(1-\delta)(1/q-1/2)}\right)$, where Φ_j are the extensions of ϕ_j to the whole $\mathbb{R}^\infty$. We also obtained similar results for the case when the lognormal inputs are parametrized on $\mathbb{R}^M$ with very large dimension M, and the approximation error is measured in the $\sqrt{g_M}$-weighted uniform norm of the Bochner space $L_\infty^{\sqrt{g}}(\mathbb{R}^M, V)$, where g_M is the density function of the standard Gaussian probability measure on $\mathbb{R}^M$.

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

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

1 Introduction

Partial differential equations (PDEs) with parametric and stochastic inputs are a common model used in science and engineering. Stochastic nature reflects the uncertainty in various parameters presented in the physical phenomenon modelled by the equation. A central problem of computational uncertainty quantification is efficient numerical approximation for parametric and stochastic PDEs which has been of great interest and achieved significant progress in recent decades. There is a large number of non-deep-neural-network papers on this topic to mention all of them. We point out just some works [3, 4, 5, 7, 8, 9, 10, 11, 12, 14, 15, 24, 36, 60, 61] which are directly related to our paper. In particular, collocation approximations which are based on a finite number of particular solvers to parametric and stochastic PDEs, were considered in [8, 9, 10, 14, 15, 18, 24, 60].

The approximation universality of neural networks has been achieved a basis understanding since the 1980's ([6, 13, 25, 37]). Deep neural networks in recent years have been rapidly developed in theory and applications to a wide range of fields due to their advantage over shallow ones. Since their application range is getting wider, theoretical analysis discovering reasons of these significant practical improvements attracts special attention [2, 20, 44, 55, 56]. In recent years, there has been a number of interesting papers that addressed the role of depth and architecture of deep neural networks for non-adaptive and adaptive approximation of functions having a particular regularity [1, 22, 29, 32, 31, 42, 39, 50, 48, 58, 59]. High-dimensional approximations by deep neural networks have been studied in [43, 52, 16, 19], and their applications to high-dimensional PDEs in [23, 27, 28, 30, 33, 46, 51]. Most of these papers employed the rectified linear unit (ReLU) as the activation function of deep neural networks since the ReLU is a simple and preferable in many applications. The output of such a deep neural network is a continuous piece-wise linear function which is easily and cheaply computed. The reader can consult the recent survey papers [21, 47] for various problems and aspects of neural network approximation and bibliography.

Recently, a number of papers have been devoted to various problems and methods of deep neural network approximation for parametric and stochastic PDEs such as dimensionality reduction [57], deep neural network expression rates for generalized polynomial chaos expansions (gpc) of solutions to parametric elliptic PDEs [17, 49], reduced basis methods [38] the problem of learning the discretized parameter-to-solution map in practice [26], Bayesian PDE inversion [33, 34, 45], etc. Note that except [17] all of these papers treated parametric and stochastic PDEs with affine inputs on the compact set $\mathbb{I}^\infty := [-1,1]^\infty$. The authors of paper [49] proved dimension-independent deep neural network expression rate bounds of the uniform approximation of solution to parametric elliptic PDE with affine inputs on $\mathbb{I}^\infty$ 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. The paper [17] investigated 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 gpc of u. Under a certain assumption on ℓ_q-summability ($0 < q < \infty$) for the lognormal inputs, it was proven that for every integer $n > 1$, one can construct a non-adaptive compactly supported deep ReLU neural network $\boldsymbol{\phi}_n$ of size $\le n$ on $\mathbb{R}^m$ with $m = \mathcal{O}(n/\log n)$, having m outputs so that the summation constituted by replacing Hermite polynomials in the m-term truncation by these m outputs approximates u with the error bound $\mathcal{O}\left((n/\log n)^{-1/q}\right)$. The authors of [17] also obtained some results on similar problems for parametric and stochastic elliptic PDEs with affine inputs, based on the Jacobi and

Taylor gpc expansions.

In the present paper, we are interested in constructing deep ReLU neural networks for collocation approximation of the solution to parametric elliptic PDEs with lognormal inputs. We study the convergence rate of this approximation in terms of the size of deep ReLU neural networks.

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 right-hand side f and diffusion coefficient a as functions on D. Denote by $V := H^1_0(D)$ 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 $\mathbb{R}^\infty$. 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 $\boldsymbol{y} \in \mathbb{R}^\infty \mapsto u(\boldsymbol{y}) \in V$. The goal is to achieve 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 the diffusion coefficient a is of the form

$$a(\boldsymbol{y}) = \exp(b(\boldsymbol{y})) \tag{1.3}$$

with $b(\boldsymbol{y})$ in the infinite-dimensional form:

$$b(\boldsymbol{y}) = \sum_{j=1}^{\infty} y_j \psi_j, \quad \boldsymbol{y} \in \mathbb{R}^\infty, \tag{1.4}$$

where the y_j are i.i.d. standard Gaussian random variables and $\psi_j \in L_\infty(D)$. We also consider the finite-dimensional form when

$$b(\boldsymbol{y}) = \sum_{j=1}^{M} y_j \psi_j, \quad \boldsymbol{y} \in \mathbb{R}^M, \tag{1.5}$$

with finite but very large dimension M.

We briefly describe the main results of the present paper.

We investigate non-adaptive collocation methods of high-dimensional deep ReLU neural network approximation of the solution $u(\boldsymbol{y})$ to parametrized diffusion elliptic PDEs (1.2) with lognormal inputs (1.3) in the infinite-dimensional case (1.4) and finite-dimensional case (1.5). In the infinite-dimensional case (1.4), the approximation error is measured in the norm of the Bochner space $L_2(\mathbb{R}^\infty, V, \gamma)$, where γ is the infinite tensor product standard Gaussian probability on $\mathbb{R}^\infty$. Assume that there exists a sequence of positive numbers $\boldsymbol{\rho} = (\rho_j)_{j\in\mathbb{N}}$ such that for some $0 < q < 2$,

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

Then, given an arbitrary number δ with $0 < \delta < \min(1, 1/q - 1/2)$, for every integer $n > 1$, we can construct a compactly supported deep ReLU neural network $\boldsymbol{\phi}_n := (\phi_j)_{j=1}^m$ of size at most n on $\mathbb{R}^m$ with $m = \mathcal{O}\left(n^{1-\delta}\right)$, and a sequence of points $Y_n := \left(\boldsymbol{y}^j\right)_{j=1}^m \subset \mathbb{R}^m$ so that

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

(ii) The output dimension of $\boldsymbol{\phi}_n$ is $m = \mathcal{O}\left(n^{1-\delta}\right)$;

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

(iv) The support of $\boldsymbol{\phi}_n$ is contained in the hyper-cube $[-T, T]^m$ with $T = \mathcal{O}\left(n^{1-\delta}\right)$;

(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 $\boldsymbol{y} = (y_j)_{j\in\mathbb{N}} \in \mathbb{R}^\infty$, the collocation approximation of u by the function

$$\Phi_n u := \sum_{j=1}^m u\left(\boldsymbol{y}^j\right) \Phi_j,$$

which is based on the m solvers $\left(u\left(\boldsymbol{y}^j\right)\right)_{j=1}^m$ and the deep ReLU network $\boldsymbol{\phi}_n$, gives the twofold error estimates

$$\|u - \Phi_n u\|_{L_2(\mathbb{R}^\infty, V, \gamma)} = \mathcal{O}\left(m^{-\left(\frac{1}{q}-\frac{1}{2}\right)}\right) = \mathcal{O}\left(n^{-(1-\delta)\left(\frac{1}{q}-\frac{1}{2}\right)}\right). \tag{1.6}$$

Notice that the error bound in m in (1.6) is the same as the error bound of the collocation approximation of u by the sparse-grid Lagrange gpc interpolation based on m the same particular solvers $\left(u\left(\boldsymbol{y}^j\right)\right)_{j=1}^m$, which so far is the best known result [15, Corollary 3.1]. Moreover, the convergence rate $(1-\delta)(1/q - 1/2)$ with respect to the size of the deep ReLU network in the collocation approximation, is comparable with the convergence rate $1/q - 1/2$ with respect to the number of particular solvers in the collocation approximation by sparse-grid Lagrange gpc interpolation.

We also obtained similar results in manner of the items (i)–(v) in the finite-dimensional case (1.5) with the approximation error measured in the $\sqrt{g_M}$-weighted uniform norm of the Bochner space $L_\infty^{\sqrt{g}}(\mathbb{R}^M, V)$, where g_M is the density function of the standard Gaussian probability measure on $\mathbb{R}^M$.

The paper is organized as follows. In Section 2, we present a necessary knowledge about deep ReLU neural networks. Section 3 is devoted to collocation methods of deep ReLU neural network approximation of functions in Bochner spaces $L_2(\mathbb{R}^\infty, X, \gamma)$ or in $L_2(\mathbb{R}^M, X, \gamma)$ related to a Hilbert space X and the tensor product standard Gaussian probability measure γ. In Section 4, we apply the results in the previous section to the collocation approximation by deep ReLU neural networks of the solution u to the parametrized elliptic PDEs (1.2) with lognormal inputs (1.3) on in the infinite case (1.4) and finite case (1.5).

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 $\operatorname{supp}(\boldsymbol{s}) := \{j \in \mathbb{N} : s_j > 0\}$ is a finite set. For $\boldsymbol{s} \in \mathbb{F}$, put $|\boldsymbol{s}|_1 := \sum_{j \in \mathbb{N}} s_j$. For a set G, we denote by $|G|$ the cardinality of G. If $\boldsymbol{a} = (a_j)_{j \in \mathcal{J}}$ is a sequence of positive numbers with any index set $\mathcal{J}$, then we use the notation $\boldsymbol{a}^{-1} := (a_j^{-1})_{j \in \mathcal{J}}$. We use letters C and K to denote general positive constants which may take different values, and $C_{\alpha,\beta,\dots}$ and $K_{\alpha,\beta,\dots}$ when we want to emphasize the dependence of these constants on $\alpha, \beta, \dots$, or when this dependence is important in a particular situation.

2 ReLU neural networks

In this section, we present some auxiliary knowledge on deep ReLU neural networks which will be used as a tool of approximation. We will consider deep feed-forward neural networks that allows connections of neurons in non-neighboring layers. The ReLU activation function is defined by $\sigma(t) := \max\{t, 0\}, t \in \mathbb{R}$. We denote: $\sigma(\boldsymbol{x}) := (\sigma(x_1), \dots, \sigma(x_d))$ for $\boldsymbol{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$.

Let us recall a standard definition of deep ReLU neural network and relevant terminology. Let $d, L \in \mathbb{N}$, $L \geq 2$, $N_0 = d$, and $N_1, \dots, N_L \in \mathbb{N}$. Let $\boldsymbol{W}^\ell = \left(w_{i,j}^\ell\right) \in \mathbb{R}^{N_\ell \times \left(\sum_{i=1}^{\ell-1} N_i\right)}$, $\ell = 1, \dots, L$, be an $N_\ell \times \left(\sum_{i=1}^{\ell-1} N_i\right)$ matrix, and $\boldsymbol{b}^\ell = (b_j^\ell) \in \mathbb{R}^{N_\ell}$. A ReLU neural network Φ (on $\mathbb{R}^d$) with input dimension d, output dimension N_L and L layers is called a sequence of matrix-vector tuples

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

in which the following computation scheme is implemented:

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

We call $\boldsymbol{z}^0$ the input and with an ambiguity we use the notation $\Phi(\boldsymbol{x}) := \boldsymbol{z}^L$ for the output of Φ. In some places we identify a ReLU neural network with its output. We adopt 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 ReLU neural network, and otherwise, a shallow ReLU neural network.

There are two basic operations which neural networks allow for. This is the parallelelization of several neural networks and the concatenation of two neural networks. The reader can find for instance, in [32] (see also [21, 47]) for detailed decriptions as well for the following two lemmas on these operations.

Lemma 2.1 (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 of the deep ReLU neural networks Φ_j, $j = 1, \ldots, N$.

Lemma 2.2 (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 deep ReLU neural network Φ is called the concatenation of the deep ReLU neural networks Φ_1 and Φ_2.

The following lemma was proven in [49, Proposition 3.3].

Lemma 2.3 *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* $j \in \{1, \dots, d\}$ *then* $\Phi_P(\boldsymbol{x}) = 0$ *and there exists a constant* $C > 0$ *independent of* δ *and* d *such that*

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

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 $\operatorname{supp}(\varphi_1) \subset [-2, 2]$. It is easy to verify that φ_1 can be realized exactly by a deep ReLU neural network (still denoted by φ_1) with size $W(\varphi_1) \le C$ for some positive constant C. Similarly, let φ_0 be the ReLU 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 $W(\varphi_0) \le C$ for some positive constant C.

The following lemma directly derived from the realization of the functions φ_0 and φ_1 by deep ReLU neural network and Lemma 2.3 (see also [17, Lemma 2.4]).

Lemma 2.4 *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*

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

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

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

3 Deep ReLU neural network approximation in Bochner spaces

In this section, we investigate collocation methods of deep ReLU neural network approximation of functions in Bochner spaces related to a Hilbert space X and tensor product standard Gaussian probability measures γ. Functions to be approximated have the weighted ℓ_2-summable Hermite gpc expansion coefficients. The approximation is based on the sparse-grid Lagrange gpc interpolation approximation. We construct such methods and prove convergence rates of the approximation by them. The results obtained in this section will be applied to deep ReLU neural network collocation approximation of the solution of parametrized elliptic PDEs with lognormal inputs in the next section.

3.1 Tensor product Gaussian measures and Bochner spaces

Let $\gamma(y)$ be the standard Gaussian probability measure on $\mathbb{R}$ with the density

$$g(y) := \frac{1}{\sqrt{2\pi}} e^{-y^2/2}, \text{ i.e.,} \quad \mathrm{d}\gamma(y) := g(y)\,\mathrm{d}y. \tag{3.1}$$

For $M \in \mathbb{N}$, the standard Gaussian probability measures $\gamma(\boldsymbol{y})$ on $\mathbb{R}^M$ can be defined by

$$\mathrm{d}\gamma(\boldsymbol{y}) := g_M(\boldsymbol{y})\mathrm{d}(\boldsymbol{y}) = \bigotimes_{j=1}^{M} g(y_j)\mathrm{d}(y_j), \quad \boldsymbol{y} = (y_j)_{j=1}^M \in \mathbb{R}^M,$$

where $g_M(\boldsymbol{y}) := \bigotimes_{j=1}^M g(y_j)$.

We next recall a concept of standard Gaussian probability measure $\gamma(\boldsymbol{y})$ on $\mathbb{R}^\infty$ as the infinite tensor product of the standard Gaussian probability measures $\gamma(y_i)$:

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

The sigma algebra for $\gamma(\boldsymbol{y})$ is generated by the set of cylinders $A := \prod_{j\in\mathbb{N}} A_j$, where $A_j \subset \mathbb{R}$ are univariate μ-measurable sets and only a finite number of A_i are different from $\mathbb{R}$. For such a set A, we have $\gamma(A) = \prod_{j\in\mathbb{N}} \gamma(A_j)$. (For details on infinite tensor product of probability measures, see, e.g., [35, pp. 429–435].)

In what follows, we use letter U to denote either $\mathbb{R}^M$ or $\mathbb{R}^\infty$ and letter J to denote either M or ∞, respectively. If X is a Hilbert space, the standard Gaussian probability measure γ on U induces the Bochner space $L_2(U, X, \gamma)$ of γ-measurable mappings v from U to X, equipped with the norm

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

For a γ-measurable subset Ω in U the spaces $L_2(\Omega, X, \gamma)$ and $L_2(\Omega, \gamma)$ is defined in the usual way.

In the case $U = \mathbb{R}^M$, we introduce also the space $L_\infty^{\sqrt{g}}(\mathbb{R}^M, X)$ as the set of all γ-measurable functions $v : \mathbb{R}^M \to X$ for which the $\sqrt{g_M}$-weighted uniform norm

$$\|v\|_{L_\infty^{\sqrt{g}}(\mathbb{R}^M,X)} := \operatorname*{ess\,sup}_{\boldsymbol{y}\in\mathbb{R}^M} \left(\|v(\boldsymbol{y})\|_X \sqrt{g_M(\boldsymbol{y})}\right) < \infty.$$

One may expect an infinite-dimensional version of this space. Unfortunately, we could not give a correct definition of space $L_\infty^{\sqrt{g}}(\mathbb{R}^\infty, X)$ because there is no an infinite-dimensional counterpart of the weight g_M. We make use of the abbreviations: $L_\infty^{\sqrt{g}}(\mathbb{R}^M) = L_\infty^{\sqrt{g}}(\mathbb{R}^M, \mathbb{R})$ and $L_\infty^{\sqrt{g}}(\mathbb{R}) = L_\infty^{\sqrt{g}}(\mathbb{R}, \mathbb{R})$.

In this section, we will investigate the problem of deep ReLU neural network approximation of functions in $L_2(\mathbb{R}^\infty, X, \gamma)$ or $L_2(\mathbb{R}^M, X, \gamma)$ with the error measured in the norms of the space $L_2(\mathbb{R}^\infty, X, \gamma)$ or of the space $L_\infty^{\sqrt{g}}(\mathbb{R}^M, X)$, respectively. (Notice that these norms are the most important in evaluation of the error of collocation approximation of solutions of parametric and stochastic PDEs). It is convenient to us to incorporate these different approximation problems into unified consideration. Hence, in what follows, we use the joint notations:

$$\mathcal{L}(U, X) := \begin{cases} L_\infty^{\sqrt{g}}(\mathbb{R}^M, X) & \text{if } U = \mathbb{R}^M, \\ L_2(\mathbb{R}^\infty, X, \gamma) & \text{if } U = \mathbb{R}^\infty; \end{cases}$$

$$\mathcal{F} := \begin{cases} \mathbb{N}_0^M & \text{if } \ U = \mathbb{R}^M, \\ \mathbb{F} & \text{if } \ U = \mathbb{R}^\infty; \end{cases}$$

and

$$\mathcal{N} := \begin{cases} \{1, ..., M\} & \text{if } \ U = \mathbb{R}^M, \\ \mathbb{N} & \text{if } \ U = \mathbb{R}^\infty. \end{cases}$$

Here $\mathbb{F}$ is the set of all sequences of non-negative integers $\boldsymbol{s} = (s_j)_{j\in\mathbb{N}}$ such that their support $\operatorname{supp}(\boldsymbol{s}) := \{j \in \mathbb{N} : s_j > 0\}$ is a finite set.

Let $(H_k)_{k\in\mathbb{N}_0}$ be the Hermite polynomials normalized according to $\int_\mathbb{R} |H_k(y)|^2\, g(y)\, \mathrm{d}y = 1$. Then a function $v \in L_2(U, X, \gamma)$ can be represented by the Hermite gpc expansion

$$v(\boldsymbol{y}) = \sum_{\boldsymbol{s}\in\mathcal{F}} v_{\boldsymbol{s}}\, H_{\boldsymbol{s}}(\boldsymbol{y}), \quad v_{\boldsymbol{s}} \in X, \tag{3.2}$$

with

$$H_{\boldsymbol{s}}(\boldsymbol{y}) = \bigotimes_{j\in\mathcal{N}} H_{s_j}(y_j), \quad v_{\boldsymbol{s}} := \int_U v(\boldsymbol{y})\, H_{\boldsymbol{s}}(\boldsymbol{y})\, \mathrm{d}\gamma(\boldsymbol{y}), \quad \boldsymbol{s} \in \mathcal{F}. \tag{3.3}$$

Notice that $(H_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathcal{F}}$ is an orthonormal basis of $L_2(U, \gamma) := L_2(U, \mathbb{R}, \gamma)$. Moreover, for every $v \in L_2(U, X, \gamma)$ represented by the series (3.2), the Parseval's identity holds

$$\|v\|^2_{L_2(U,X,\gamma)} \ = \ \sum_{\boldsymbol{s}\in\mathcal{F}} \|v_{\boldsymbol{s}}\|_X^2.$$

For $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{F}$, the inequality $\boldsymbol{s}' \le \boldsymbol{s}$ means that $s_j' \le s_j$, $j \in \mathcal{N}$. A sequence $\boldsymbol{\sigma} = (\sigma_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathcal{F}}$ is called increasing if $\sigma_{\boldsymbol{s}'} \le \boldsymbol{s}$ for $\boldsymbol{s}' \le \boldsymbol{s}$.

Assumption (I) For $v \in L_2(U, X, \gamma)$ represented by the series (3.2), there exists an increasing sequence $\boldsymbol{\sigma} = (\sigma_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathcal{F}}$ of positive numbers strictly larger than 1 such that $\left\|\boldsymbol{\sigma}^{-1}\right\|_{\ell_q(\mathcal{F})} \le C < \infty$ for some q with $0 < q < 2$, and

$$\left(\sum_{\boldsymbol{s}\in\mathcal{F}} (\sigma_{\boldsymbol{s}} \|v_{\boldsymbol{s}}\|_X)^2\right)^{1/2} \ \le C \ < \infty, \tag{3.4}$$

where the constants C are independent of J.

Lemma 3.1 *For $v \in L_2(U, X, \gamma)$ sastisfying Assumption* (I)*, the series* (3.2) *converges absolutely and therefore, unconditionally in $\mathcal{L}(U, X)$ to v and*

$$\sum_{\boldsymbol{s}\in\mathcal{F}} \|v_{\boldsymbol{s}}\|_X \ \le \ C \ < \infty, \tag{3.5}$$

where the constant C is independent of J.

Proof. By applying the Hölder inequality from Assumption (I) we obtain

$$\sum_{\boldsymbol{s}\in\mathcal{F}} \|v_{\boldsymbol{s}}\|_X \le \left(\sum_{\boldsymbol{s}\in\mathcal{F}} (\sigma_{\boldsymbol{s}}\|v_{\boldsymbol{s}}\|_X)^2\right)^{1/2} \left(\sum_{\boldsymbol{s}\in\mathcal{F}} \sigma_{\boldsymbol{s}}^{-2}\right)^{1/2} \le \ C\left\|\boldsymbol{\sigma}^{-1}\right\|_{\ell_q(\mathcal{F})} \ < \ \infty.$$

This proves (3.5). Hence, by the equality $\|H_{\boldsymbol{s}}\|_{L_2(\mathbb{R}^\infty)} = 1$, $\boldsymbol{s} \in \mathbb{F}$, and the inequality $\|H_{\boldsymbol{s}}\|_{L_\infty(\mathbb{R}^\infty)} < 1$, $\boldsymbol{s} \in \mathbb{N}_0^M$ (which follows from (A.9) in Appendix), the series (3.2) converges absulutely, and therefore, unconditionally to $v \in L_2(U, X, \gamma)$ since by the Parseval's identity it already converges to v in the norm of $L_2(U, X, \gamma)$. ◻

3.2 Sparse-grid Lagrange gpc interpolation

For $m \in \mathbb{N}_0$, let $Y_m = (y_{n;k})_{k\in\pi_m}$ be the increasing sequence of the $m+1$ roots of the Hermite polynomial H_{m+1}, ordered as

$$y_{m,-j} < \cdots < y_{m,-1} < y_{m,0} = 0 < y_{m,1} < \cdots < y_{m,j} \quad \text{if } m = 2j,$$

$$y_{m,-j} < \cdots < y_{m,-1} < y_{m,1} < \cdots < y_{m,j} \quad \text{if } m = 2j-1,$$

where

$$\pi_m := \begin{cases} \{-j, -j+1, ..., -1, 0, 1, ..., j-1, j\} & \text{if } m = 2j; \\ \{-j, -j+1, ..., -1, 1, ..., j-1, j\} & \text{if } m = 2j-1. \end{cases}$$

(in particular, $Y_0 = (y_{0;0})$ with $y_{0;0} = 0$).

If v is a function on $\mathbb{R}$ taking values in a Hilbert space X and $m \in \mathbb{N}_0$, we define the function $I_m(v)$ on $\mathbb{R}$ taking values in X by

$$I_m(v) := \sum_{k\in\pi_m} v(y_{m;k})L_{m;k}, \quad L_{m;k}(y) := \prod_{j\in\pi_m \ j\neq k} \frac{y - y_{m;j}}{y_{n;k} - y_{m;j}}, \tag{3.6}$$

interpolating v at $y_{m;k}$, i.e., $I_m(v)(y_{m;k}) = v(y_{m;k})$. Notice that for a function $v : \mathbb{R} \to \mathbb{R}$, the function $I_m(v)$ is the Lagrange polynomial having degree $\le m$, and that $I_m(\varphi) = \varphi$ for every polynomial φ of degree $\le m$.

Let

$$\lambda_m(Y_m) := \sup_{\|v\|_{L_\infty^{\sqrt{g}}(\mathbb{R})}\le 1} \|I_m(v)\|_{L_\infty^{\sqrt{g}}(\mathbb{R})}$$

be the Lebesgue constant. It was proven in [40, 41, 53] that

$$\lambda_m(Y_m) \ \le \ C(m+1)^{1/6}, \quad m \in \mathbb{N},$$

for some positive constant C independent of n (with the obvious inequality $\lambda_0(Y_0) \le 1$). Hence, for every $\varepsilon > 0$, there exists a positive constant $C_\varepsilon \ge 1$ independent of n such that

$$\lambda_m(Y_m) \ \le \ (1 + C_\varepsilon m)^{1/6+\varepsilon}, \quad \forall m \in \mathbb{N}_0. \tag{3.7}$$

We define the univariate operator Δ_m for $m \in \mathbb{N}_0$ by

$$\Delta_m := I_m - I_{m-1},$$

with the convention $I_{-1} = 0$.

Lemma 3.2 *For every $\varepsilon > 0$, there exists a positive constant C_ε independent of m such that for every function v on $\mathbb{R}$,*

$$\|\Delta_m(v)\|_{L_\infty^{\sqrt{g}}(\mathbb{R})} \le (1 + C_\varepsilon m)^{1/6+\varepsilon} \|v\|_{L_\infty^{\sqrt{g}}(\mathbb{R})}, \quad \forall m \in \mathbb{N}_0, \tag{3.8}$$

whenever the norm in the right-hand side is finite.

Proof. From the assumptions we have that

$$\|\Delta_m(v)\|_{L_\infty^{\sqrt{g}}(\mathbb{R})} \le 2(1 + Cm)^{1/6} \|v\|_{L_\infty^{\sqrt{g}}(\mathbb{R})}, \quad \forall m \in \mathbb{N}_0,$$

which implies (3.8). □

We will use a sparse-grid Lagrange gpc interpolation as an intermediate approximation in the deep ReLU neural network approximation of functions $v \in L_2(U, X, \gamma)$. In order to have a correct definition of interpolation operator we have to impose some neccessary restrictions on v. Let $\mathcal{E}$ be a γ-measurable subset in U such that $\gamma(\mathcal{E}) = 1$ and $\mathcal{E}$ contains all $\boldsymbol{y} \in U$ with $|\boldsymbol{y}|_0 < \infty$ in the case $U = \mathbb{R}^\infty$, where $|\boldsymbol{y}|_0$ denotes the number of nonzero components y_j of $\boldsymbol{y}$. For a given $\mathcal{E}$ and Hilbert space X, we define $L_2^{\mathcal{E}}(U, X, \gamma)$ as the subspace in $L_2(U, X, \gamma)$ of all elements v such that the point value $v(\boldsymbol{y})$ (of a representative of v) is well-defined for all $\boldsymbol{y} \in \mathcal{E}$. In what folllows, $\mathcal{E}$ is fixed.

For $v \in L_2^{\mathcal{E}}(U, X, \gamma)$, we introduce the tensor product operator $\Delta_{\boldsymbol{s}}$, $\boldsymbol{s} \in \mathcal{F}$, by

$$\Delta_{\boldsymbol{s}}(v) := \bigotimes_{j \in \mathcal{N}} \Delta_{s_j}(v),$$

where the univariate operator Δ_{s_j} is applied to the univariate function v by considering v as a function of variable y_i with the other variables held fixed. From the definition of $L_2^{\mathcal{E}}(U, X, \gamma)$ one can see that the operators $\Delta_{\boldsymbol{s}}$ are well-defined for all $\boldsymbol{s} \in \mathcal{F}$. We define for $\boldsymbol{s} \in \mathcal{F}$,

$$I_{\boldsymbol{s}}(v) := \bigotimes_{j \in \mathcal{N}} I_{s_j}(v), \quad L_{\boldsymbol{s};\boldsymbol{k}}(v) := \bigotimes_{j \in \mathcal{N}} L_{s_j;k_j}(v), \quad \pi_{\boldsymbol{s}} := \prod_{j \in \mathcal{N}} \pi_{s_j}.$$

For $\boldsymbol{s} \in \mathcal{F}$ and $\boldsymbol{k} \in \pi_{\boldsymbol{s}}$, let $E_{\boldsymbol{s}}$ be the subset in $\mathcal{F}$ of all $\boldsymbol{e}$ such that e_j is either 1 or 0 if $s_j > 0$, and e_j is 0 if $s_j = 0$, and let $\boldsymbol{y}_{\boldsymbol{s};\boldsymbol{k}} := (y_{s_j;k_j})_{j \in \mathcal{N}} \in U$. Put $|\boldsymbol{s}|_1 := \sum_{j \in \mathcal{N}} s_j$ for $\boldsymbol{s} \in \mathcal{F}$. It is easy to check that the interpolation operator $\Delta_{\boldsymbol{s}}$ can be represented in the form

$$\Delta_{\boldsymbol{s}}(v) = \sum_{\boldsymbol{e} \in E_{\boldsymbol{s}}} (-1)^{|\boldsymbol{e}|_1} I_{\boldsymbol{s}-\boldsymbol{e}}(v) = \sum_{\boldsymbol{e} \in E_{\boldsymbol{s}}} (-1)^{|\boldsymbol{e}|_1} \sum_{\boldsymbol{k} \in \pi_{\boldsymbol{s}-\boldsymbol{e}}} v(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}) L_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}. \tag{3.9}$$

For a given finite set $\Lambda \subset \mathcal{F}$, we introduce the gpc interpolation operator I_Λ by

$$I_\Lambda := \sum_{\boldsymbol{s}\in\Lambda} \Delta_{\boldsymbol{s}}.$$

From (3.9) we obtain

$$I_\Lambda(v) \ = \ \sum_{\boldsymbol{s}\in\Lambda} \sum_{\boldsymbol{e}\in E_{\boldsymbol{s}}} (-1)^{|\boldsymbol{e}|_1} \sum_{\boldsymbol{k}\in\pi_{\boldsymbol{s}-\boldsymbol{e}}} v(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}) L_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}. \tag{3.10}$$

A set $\Lambda \subset \mathcal{F}$ is called downward closed if the inclusion $\boldsymbol{s} \in \Lambda$ yields the inclusion $\boldsymbol{s}' \in \Lambda$ for every $\boldsymbol{s}' \in \mathcal{F}$ such that $\boldsymbol{s}' \le \boldsymbol{s}$.

For $\theta, \lambda \ge 0$, we define the sequence $\boldsymbol{p}(\theta,\lambda) := (p_{\boldsymbol{s}}(\theta,\lambda))_{\boldsymbol{s}\in\mathcal{F}}$ by

$$p_{\boldsymbol{s}}(\theta,\lambda) := \prod_{j\in\mathcal{N}} (1+\lambda s_j)^\theta, \quad \boldsymbol{s}\in\mathcal{F}, \tag{3.11}$$

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

Let $0 < q < \infty$ and $\boldsymbol{\sigma} = (\sigma_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathcal{F}}$ be a sequence of positive numbers. For $\xi > 0$, define the set

$$\Lambda(\xi) := \{\boldsymbol{s}\in\mathcal{F} : \sigma_{\boldsymbol{s}}^q \le \xi\}. \tag{3.12}$$

By the formula (3.10) we can represent the operator $I_{\Lambda(\xi)}$ in the form

$$I_{\Lambda(\xi)}(v) \ = \ \sum_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k})\in G(\xi)} (-1)^{|\boldsymbol{e}|_1} v(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}) L_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}, \tag{3.13}$$

where

$$G(\xi) := \ \{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k}) \in \mathcal{F}\times\mathcal{F}\times\mathcal{F} : \ \boldsymbol{s}\in\Lambda(\xi), \ \boldsymbol{e}\in E_{\boldsymbol{s}}, \ \boldsymbol{k}\in\pi_{\boldsymbol{s}-\boldsymbol{e}}\}. \tag{3.14}$$

The following theorem gives an estimate for the error of the approximation of $v \in \mathcal{L}_2^{\mathcal{E}}(U,X,\gamma)$ by the sparse-grid Lagrange gpc interpolation $I_{\Lambda(\xi)}v$ on the sampling points in the set $G(\xi)$, which will be used in the deep ReLU neural approximation in the next section.

Theorem 3.1 *Let $v \in \mathcal{L}_2^{\mathcal{E}}(U,X,\gamma)$ satisfy Assumption* (I) *and let $\varepsilon > 0$ be a fixed number. Assume that $\|\boldsymbol{p}(\theta,\lambda)\boldsymbol{\sigma}^{-1}\|_{\ell_q(\mathcal{F})} \le C < \infty$, where $\theta = 7/3 + 2\varepsilon$, $\lambda := C_\varepsilon$ is the constant in Lemma 3.2, and the constant C is independent of J. Then for each $\xi > 1$, we have that*

$$\|v - I_{\Lambda(\xi)}v\|_{\mathcal{L}(U,X)} \le C\xi^{-(1/q-1/2)}, \tag{3.15}$$

where the constant C in (3.15) *is independent of J, v and ξ.*

A proof of this theorem is given in Appendix.

Corollary 3.1 *Let $v \in \mathcal{L}_2^{\mathcal{E}}(U, X, \gamma)$ satisfy Assumption* (I) *and let $\varepsilon > 0$ be a fixed number. Assume that $\left\|\boldsymbol{p}(\theta, \lambda)\boldsymbol{\sigma}^{-1}\right\|_{\ell_q(\mathcal{F})} \le C < \infty$, where $\theta = \max(7/3 + 2\varepsilon, 2/q)$, $\lambda := C_\varepsilon$ is the constant in Lemma 3.2, and the constant C is independent of J. Then for each $n > 1$, we can construct a sequence of points $Y_{\Lambda(\xi_n)} := (\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}})_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k}) \in G(\xi_n)}$ so that $|G(\xi_n)| \le n$ and*

$$\|v - I_{\Lambda(\xi_n)} v\|_{\mathcal{L}(U,X)} \le C n^{-(1/q - 1/2)}, \tag{3.16}$$

where the constant C in (3.16) *is independent of J, v and n.*

Proof. Notice that this corollary was proven in [15, Corollary 3.1] for the case $U = \mathbb{R}^\infty$. By Lemma A.2 in Appendix $|G(\xi)| \le C_q \xi$ for every $\xi > 1$. Hence, the corollary follows from Theorem 3.1 by sellection of ξ_n as the maximal number satisfying $|G(\xi_n)| \le n$. ∎

3.3 Approximation by deep ReLU neural networks

In this section, we construct deep ReLU neural networks for collocation approximation of functions $v \in L_2(U, X, \gamma)$. We primarily approximate v by the sparse-grid Lagrange gpc interpolation $I_{\Lambda(\xi)} v$. Under the assumptions of Lemma A.1(iii) in Appendix, $I_{\Lambda(\xi)} v$ can be seen as a function on $\mathbb{R}^m$, where $m := \min\{M, \lfloor K_q \xi \rfloor\}$. In the next step, we approximate $I_{\Lambda(\xi)} v$ by its truncation $I^\omega_{\Lambda(\xi)} v$ on a sufficiently large hyper-cube

$$B^m_\omega := [-2\sqrt{\omega}, 2\sqrt{\omega}]^m \subset \mathbb{R}^m, \tag{3.17}$$

where the parameter ω depending on ξ is chosen in an appropriate way. Finally, the function $I^\omega_{\Lambda(\xi)} v$ and therefore, v is approximated by a function $\Phi_{\Lambda(\xi)} v$ on $\mathbb{R}^m$ which is constructed from a deep ReLU neural network. Let us discribe this construction.

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

Suppose that deep ReLU neural networks $\phi_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}$ on the cube B^m_ω are already constructed for approximation of the polynomials $L_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}$, $(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k}) \in G(\xi)$. Then the network $\boldsymbol{\phi}_{\Lambda(\xi)} := (\phi_{\boldsymbol{s}})_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k}) \in G(\xi)}$ on B^m_ω with $|G(\xi)|$ outputs which is constructed by parallelization, is used to construct an approximation of $I^\omega_{\Lambda(\xi)} v$ and hence of v. Namely, we approximate v by

$$\Phi_{\Lambda(\xi)} v(\boldsymbol{y}) := \sum_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k}) \in G(\xi)} (-1)^{|\boldsymbol{e}|_1} v(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}) \phi_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}. \tag{3.18}$$

For the set $\Lambda(\xi)$, we introduce the following numbers:

$$m_1(\xi) := \max_{\boldsymbol{s} \in \Lambda(\xi)} |\boldsymbol{s}|_1, \tag{3.19}$$

and

$$m(\xi) := \max\left\{j \in \mathcal{N} : \exists \boldsymbol{s} \in \Lambda(\xi) \text{ such that } s_j > 0\right\}. \tag{3.20}$$

Denote by $\boldsymbol{e}^i = (e^i_j)_{j\in\mathcal{N}} \in \mathcal{F}$ the element with $e^i_i = 1$ and $e^i_j = 0$ for $j \neq i$.

In this section, we will prove our main results on deep ReLU neural network approximation of functions $v \in L_2^{\mathcal{E}}(U, X, \gamma)$ with the error measured in the norm of the space $L_2(\mathbb{R}^\infty, X, \gamma)$ or of the space $L_\infty^{\sqrt{g}}(\mathbb{R}^M, X)$, which are incorporated into the following joint theorem.

Theorem 3.2 *Let $v \in L_2^{\mathcal{E}}(U, X, \gamma)$ satisfy Assumption (I). Let θ be any number such that $\theta \geq 3/q$. Assume that the sequence $\boldsymbol{\sigma} = (\sigma_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathcal{F}}$ in Assumption (I) satisfies $\sigma_{\boldsymbol{e}^{i'}} \leq \sigma_{\boldsymbol{e}^i}$ if $i' < i$, and that $\left\|\boldsymbol{p}(\theta)\boldsymbol{\sigma}^{-1}\right\|_{\ell_q(\mathcal{F})} \leq C < \infty$, where the constant C is independent of J. Let K_q, $K_{q,\theta}$ and C_q be the constants in the assumptions of Lemma A.1 and of Lemma A.2 in Appendix. Then for every $\xi > 1$, we can construct a deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi)} := (\phi_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}})_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k})\in G(\xi)}$ on $\mathbb{R}^m$ with*

$$m := \begin{cases} \min\{M, \lfloor K_q\xi \rfloor\} & \text{if } \ U = \mathbb{R}^M, \\ \lfloor K_q\xi \rfloor & \text{if } \ U = \mathbb{R}^\infty, \end{cases}$$

and a sequence of points $Y_{\Lambda(\xi)} := (\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}})_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k})\in G(\xi)}$ having the following properties.

(i) *The deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi)}$ and sequence of points $Y_{\Lambda(\xi)}$ are independent of v;*

(ii) *The output dimension of $\boldsymbol{\phi}_{\Lambda(\xi)}$ are at most $\lfloor C_q\xi \rfloor$;*

(iii) $W\big(\boldsymbol{\phi}_{\Lambda(\xi)}\big) \leq C\xi^{1+2/\theta q}\log\xi$*;*

(iv) $L\big(\boldsymbol{\phi}_{\Lambda(\xi)}\big) \leq C\xi^{1/\theta q}(\log\xi)^2$*;*

(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*

$$\|v - \Phi_{\Lambda(\xi)}v\|_{\mathcal{L}(U,X)} \leq C\xi^{-(1/q-1/2)}. \tag{3.21}$$

Here the constants C are independent of J, v and ξ.

Let us briefly draw a plan of the proof of this theorem. We will give a detailted proof for the case $U = \mathbb{R}^\infty$ and then point out that the case $U = \mathbb{R}^M$ can be proven in the same way with slight modification.

In what follows in this section, all definitions, formulas and assertions are given for the case $U = \mathbb{R}^\infty$, and for $\xi > 1$, we use the letters m and ω only for the notations

$$m := \lfloor K_q\xi \rfloor, \quad \omega := \lfloor K_{q,\theta}\xi \rfloor, \tag{3.22}$$

where K_q and $K_{q,\theta}$ are the constants defined in Lemma A.1 in Appendix. As mentioned above, we primarily approximate $v \in L_2(\mathbb{R}^\infty, X, \gamma)$ by the gpc interpolation $I_{\Lambda(\xi)}v$. In the next step, we approximate $I_{\Lambda(\xi)}v$ by its truncation $I^\omega_{\Lambda(\xi)}v$ on the hyper-cube B^m_ω, which will be constructed below. The final step is to construct a deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi)} := (\phi_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}})_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k})\in G(\xi)}$ to approximate $I^{\omega}_{\Lambda(\xi)}v$ by $\Phi_{\Lambda(\xi)}v$ of the form (3.18).

For a function φ defined on $\mathbb{R}$, we denote by φ^{ω} the truncation of φ on B^1_{ω}, i.e.,

$$\varphi^{\omega}(y) := \begin{cases} \varphi^{\omega}(y) & \text{if } y \in B^1_{\omega} \\ 0 & \text{otherwise.} \end{cases} \tag{3.23}$$

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

$$L^{\omega}_{\boldsymbol{s},\boldsymbol{k}}(\boldsymbol{y}) := \prod_{j=1}^{m} L^{\omega}_{s_j;k_j}(y_j), \qquad \boldsymbol{y} \in \mathbb{R}^m.$$

We have $L^{\omega}_{\boldsymbol{s},\boldsymbol{k}}(\boldsymbol{y}) = \prod_{j=1}^{m} L_{s_j;k_j}(y_j)$ if $\boldsymbol{y} \in B^m_{\omega}$, and $L^{\omega}_{\boldsymbol{s},\boldsymbol{k}}(\boldsymbol{y}) = 0$ otherwise. For a function $v \in L^{\mathcal{E}}_2(\mathbb{R}^{\infty}, X, \gamma)$, we define

$$I^{\omega}_{\Lambda(\xi)}(v) := \sum_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k})\in G(\xi)} (-1)^{|\boldsymbol{e}|_1} v(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}) L^{\omega}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}. \tag{3.24}$$

Let the assumptions of Theorem 3.2 hold. By Lemma A.1(iii) in Appendix for every $\xi > 1$ we have $m(\xi) \le m$. Hence, for every $(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k}) \in G(\xi)$, $L_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}$ and $L^{\omega}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}$ and therefore, $I_{\Lambda(\xi)}v$ and $I^{\omega}_{\Lambda(\xi)}v$ 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 use some intermediate approximations for estimation of the approximation error as in (3.21). Suppose that the deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi)}$ and therefore, the function $\Phi_{\Lambda(\xi)}$ are already constructed. By the triangle inequality we have

$$\begin{aligned} \|v - \Phi_{\Lambda(\xi)}v\|_{L_2(\mathbb{R}^{\infty},X,\gamma)} \le \|v - I_{\Lambda(\xi)}v\|_{L_2(\mathbb{R}^{\infty},X,\gamma)} + \|I_{\Lambda(\xi)}v - I^{\omega}_{\Lambda(\xi)}v\|_{L_2(\mathbb{R}^m\setminus B^m_{\omega},X,\gamma)} \\ + \|I^{\omega}_{\Lambda(\xi)}v - \Phi_{\Lambda(\xi)}v\|_{L_2(B^m_{\omega},X,\gamma)} + \|\Phi_{\Lambda(\xi)}v\|_{L_2(\mathbb{R}^m\setminus B^m_{\omega},X,\gamma)}. \end{aligned} \tag{3.25}$$

Hence the estimate (3.21) will be done via the bound $C\xi^{-(1/q-1/2)}$ for every of the four terms in the right-hand side. The first term is already estimated as in Theorem 3.1. The estimates for the others will be carried out in the following lemmas (Lemmas 3.3–3.5). To complete the proof of Theorem 3.2 we have also to prove the bounds of the size and depth of $\boldsymbol{\phi}_{\Lambda(\xi)}$ according to the items (iii) and (iv) which are given in Lemma 3.6 below.

For $v \in L^{\mathcal{E}}_2(\mathbb{R}^{\infty}, X, \gamma)$ satisfying Assumption (I), by Lemma 3.1 the series (3.2) converges unconditionally in $L_2(\mathbb{R}^{\infty}, X, \gamma)$ to v. Therefore, the formula (3.10) for $\Lambda = \Lambda(\xi)$ can be rewritten as

$$I_{\Lambda(\xi)}(v) \;=\; \sum_{\boldsymbol{s}\in\Lambda(\xi)} \sum_{\boldsymbol{s}'\in\mathbb{F}} v_{\boldsymbol{s}'} \sum_{\boldsymbol{e}\in E_{\boldsymbol{s}}} (-1)^{|\boldsymbol{e}|_1} \sum_{\boldsymbol{k}\in\pi_{\boldsymbol{s}-\boldsymbol{e}}} H_{\boldsymbol{s}'}(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}) L_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}. \tag{3.26}$$

Hence, we also have by the definition (3.24)

$$I^{\omega}_{\Lambda(\xi)}(v) = \sum_{\boldsymbol{s}\in\Lambda(\xi)} \sum_{\boldsymbol{s}'\in\mathbb{F}} v_{\boldsymbol{s}'} \sum_{\boldsymbol{e}\in E_{\boldsymbol{s}}} (-1)^{|\boldsymbol{e}|_1} \sum_{\boldsymbol{k}\in\pi_{\boldsymbol{s}-\boldsymbol{e}}} H_{\boldsymbol{s}'}(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}) L^{\omega}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}. \tag{3.27}$$

Lemma 3.3 *Under the assumptions of Theorem 3.2, for every $\xi > 1$, we have that*

$$\left\| I_{\Lambda(\xi)} v - I^{\omega}_{\Lambda(\xi)} v \right\|_{L_2(\mathbb{R}^\infty, X, \gamma)} \le C \xi^{-(1/q - 1/2)}, \tag{3.28}$$

where the constant C is independent of v and ξ.

Proof. By the equality

$$\left\| L_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}} - L^{\omega}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}} \right\|_{L_2(\mathbb{R}^\infty, \gamma)} = \left\| L_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}} \right\|_{L_2(\mathbb{R}^m \setminus B^m_\omega, \gamma)}, \quad \forall (\boldsymbol{s}, \boldsymbol{e}, \boldsymbol{k}) \in G(\xi),$$

and the triangle inequality, noting (3.26) and (3.27), we obtain

$$\left\| I_{\Lambda(\xi)} v - I^{\omega}_{\Lambda(\xi)} v \right\|_{L_2(\mathbb{R}^\infty, X, \gamma)} \le \sum_{\boldsymbol{s} \in \Lambda(\xi)} \sum_{\boldsymbol{s}' \in \mathbb{F}} \|v_{\boldsymbol{s}'}\|_X \sum_{\boldsymbol{e} \in E_{\boldsymbol{s}}} \sum_{\boldsymbol{k} \in \pi_{\boldsymbol{s}-\boldsymbol{e}}} |H_{\boldsymbol{s}'}(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}})| \left\| L_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}} \right\|_{L_2(\mathbb{R}^m \setminus B^m_\omega, \gamma)}.$$

Let $(\boldsymbol{s}, \boldsymbol{e}, \boldsymbol{k}) \in G(\xi)$ be given. Then we have

$$L_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}} = \prod_{j=1}^{m} L_{s_j - e_j; k_j}(y_j), \quad \boldsymbol{y} \in \mathbb{R}^m,$$

where $L_{s_j - e_j; k_j}$ is a polynomial in variable y_j, of degree not greater than $m_1(\xi) \le \omega$. Hence, applying Lemma A.7 in Appendix with taking account of (3.22) gives

$$\left\| L_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}} \right\|_{L_2(\mathbb{R}^m \setminus B^m_\omega, \gamma)} \le C \xi e^{-K_1 \xi} \left\| L_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}} \right\|_{L_2(\mathbb{R}^m, \gamma)}.$$

From Lemmas A.3 and A.4 and Lemma A.1(ii) in Appendix we derive that

$$\begin{aligned} \left\| L_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}} \right\|_{L_2(\mathbb{R}^m, \gamma)} &= \prod_{j \in \mathbb{N}} \left\| L_{s_j - e_j; k_j} \right\|_{L_2(\mathbb{R}, \gamma)} \le \prod_{j \in \mathbb{N}} e^{K_2 (s_j - e_j)} \\ &\le \prod_{j \in \mathbb{N}} e^{K_2 s_j} = e^{K_2 |\boldsymbol{s}|_1} \le e^{K_2 m_1(\xi)} \le e^{K_3 \xi^{1/\theta q}}, \end{aligned}$$

and

$$\sum_{\boldsymbol{k} \in \pi_{\boldsymbol{s}}} |H_{\boldsymbol{s}'}(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}})| \le e^{K_4 |\boldsymbol{s}|_1} \le e^{K_4 m_1(\xi)} \le e^{K_5 \xi^{1/\theta q}}. \tag{3.29}$$

Summing up, we arrive at

$$\begin{aligned} \left\| I_{\Lambda(\xi)} v - I^{\omega}_{\Lambda(\xi)} v \right\|_{L_2(\mathbb{R}^\infty, X, \gamma)} &\le C_1 \xi \exp\Big(-K_1 \xi + (K_2 + K_5) \xi^{1/\theta q} \Big) \sum_{\boldsymbol{s} \in \Lambda(\xi)} \sum_{\boldsymbol{s}' \in \mathbb{F}} \|v_{\boldsymbol{s}'}\|_X \sum_{\boldsymbol{e} \in E_{\boldsymbol{s}}} \sum_{\boldsymbol{k} \in \pi_{\boldsymbol{s}-\boldsymbol{e}}} 1 \\ &\le C_1 \xi \exp\Big(-K_1 \xi + K_6 \xi^{1/\theta q} \Big) |G(\xi)| \sum_{\boldsymbol{s}' \in \mathbb{F}} \|v_{\boldsymbol{s}'}\|_X. \end{aligned}$$

Hence, by Lemma 3.1, Lemma A.2 in Appendix and the inequality $1/\theta q \le 1/3$ we get

$$\left\| I_{\Lambda(\xi)} v - I^{\omega}_{\Lambda(\xi)} v \right\|_{L_2(\mathbb{R}^\infty, X, \gamma)} \le C_2 \xi^2 \exp\Big(-K_1 \xi + K_6 \xi^{1/\theta q} \Big) \le C \xi^{-(1/q - 1/2)}.$$

□

The previous lemma gives the bound of the second term in the right-hand side of (3.25), i.e., the error bound for the approximation of sparse-grid Lagrange interpolation $I_{\Lambda(\xi)}v$ by its truncation $I^{\omega}_{\Lambda(\xi)}v$ on B^{ω}_m for $v \in L_2(\mathbb{R}^\infty, X, \gamma)$. As the next step, we will construct a deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi)} := (\phi_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}})_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k})\in G(\xi)}$ on $\mathbb{R}^m$ for approximating $I^{\omega}_{\Lambda(\xi)}v$ by the function $\Phi_{\Lambda(\xi)}v$ given as in (3.18), and prove the bound of the error as the third term in the right-hand side of (3.25).

For $s \in \mathbb{N}_0$, we represent the univariate interpolation polynomial $L_{s;k}$ in the form of linear combination of monomials:

$$L_{s;k}(y) =: \sum_{\ell=0}^{s} b^{s;k}_{\ell} y^{\ell}. \tag{3.30}$$

From (3.30) for each $(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k}) \in G(\xi)$ we have

$$L_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}} = \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}-\boldsymbol{e}} b^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}_{\boldsymbol{\ell}} \boldsymbol{y}^{\boldsymbol{\ell}}, \tag{3.31}$$

where the summation $\sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}-\boldsymbol{e}}$ means that the sum is taken over all $\boldsymbol{\ell}$ such that $\boldsymbol{0} \le \boldsymbol{\ell} \le \boldsymbol{s}-\boldsymbol{e}$, and

$$b^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}_{\boldsymbol{\ell}} = \prod_{j=1}^{m} b^{s_j-e_j;k_j}_{\ell_j}, \quad \boldsymbol{y}^{\boldsymbol{\ell}} = \prod_{j=1}^{m} y_j^{\ell_j}.$$

Indeed, we have

$$\begin{aligned} L_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}} &= \prod_{j=1}^{m} L_{s_j-e_j;k_j}(y_j) = \prod_{j=1}^{m} \sum_{\ell_j=0}^{s_j-e_j} b^{s_j-e_j;k_j}_{\ell_j} y_j^{\ell_j} \\ &= \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}-\boldsymbol{e}} \left(\prod_{j=1}^{m} b^{s_j-e_j;k_j}_{\ell_j} \right) \boldsymbol{y}^{\boldsymbol{\ell}} = \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}-\boldsymbol{e}} b^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}_{\boldsymbol{\ell}} \boldsymbol{y}^{\boldsymbol{\ell}}. \end{aligned}$$

By (3.27) and (3.31) we get for every $\boldsymbol{y} \in B^m_\omega$,

$$I^{\omega}_{\Lambda(\xi)}(v)(\boldsymbol{y}) = \sum_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k})\in G(\xi)} (-1)^{|\boldsymbol{e}|_1} v(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}) \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}-\boldsymbol{e}} b^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}_{\boldsymbol{\ell}} \left(4\sqrt{\omega}\right)^{|\boldsymbol{\ell}|_1} \prod_{j=1}^{m} \left(\frac{y_j}{4\sqrt{\omega}} \right)^{\ell_j}. \tag{3.32}$$

Let $\boldsymbol{\ell} \in \mathbb{F}$ be such that $\boldsymbol{0} \le \boldsymbol{\ell} \le \boldsymbol{s}-\boldsymbol{e}$. For $\boldsymbol{\ell} \ne \boldsymbol{0}$, with an appropriate change of variables, the term $\prod_{j=1}^{m} \big(\frac{y_j}{4\sqrt{\omega}}\big)^{\ell_j}$ can be represented in the form $\prod_{j=1}^{|\boldsymbol{\ell}|_1} \varphi_1(x_j)$ where φ_1 is defined before Lemma 2.4. We put

$$B_{\boldsymbol{s}} := \max_{\boldsymbol{e}\in E_{\boldsymbol{s}},\, \boldsymbol{k}\in\pi_{\boldsymbol{s}-\boldsymbol{e}}} \ \max_{\boldsymbol{0}\le\boldsymbol{\ell}\le\boldsymbol{s}-\boldsymbol{e}} \left| b^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}_{\boldsymbol{\ell}} \right|, \tag{3.33}$$

and

$$\delta^{-1} := \xi^{1/q-1/2} \sum_{\boldsymbol{s}\in\Lambda(\xi)} e^{K|\boldsymbol{s}|_1} p_{\boldsymbol{s}}(2) \left(4\sqrt{\omega}\right)^{|\boldsymbol{s}|_1} B_{\boldsymbol{s}}, \tag{3.34}$$

where K is the constant in Lemma A.3 in Appendix. Hence, by Lemma 2.4, for every $(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k}) \in G(\xi)$ and $\boldsymbol{\ell}$ satisfying $\boldsymbol{0} < \boldsymbol{\ell} \leq \boldsymbol{s}-\boldsymbol{e}$, there exists a deep ReLU neural network $\phi_{\boldsymbol{\ell}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}$ on $\mathbb{R}^m$ such that

$$\sup_{\boldsymbol{y}\in B^m_\omega} \left| \prod_{j=1}^m \left(\frac{y_j}{4\sqrt{\omega}}\right)^{\ell_j} - \phi_{\boldsymbol{\ell}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}\left(\frac{\boldsymbol{y}}{4\sqrt{\omega}}\right)\right| \leq \delta, \tag{3.35}$$

and

$$\operatorname{supp}\left(\phi_{\boldsymbol{\ell}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}\left(\frac{\cdot}{4\sqrt{\omega}}\right)\right) \subset B_{4\omega}^{|\nu_{\boldsymbol{\ell}}|}. \tag{3.36}$$

In the case when $\boldsymbol{\ell}=\boldsymbol{0}$, we fix an index $j \in \nu_{\boldsymbol{s}}$ and define the deep ReLU neural network

$$\phi_{\boldsymbol{0}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}(\boldsymbol{y}) := b_{\boldsymbol{0}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}} \varphi_0\left(\frac{y_j}{4\sqrt{\omega}}\right).$$

Then $|b_{\boldsymbol{0}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}} - \phi_{\boldsymbol{s},\boldsymbol{0}}(\boldsymbol{y})| = 0$ for $\boldsymbol{y} \in B^m_\omega$. Observe that size and the depth of $\phi_{\boldsymbol{s},\boldsymbol{0}}$ are bounded by a constant. For $\boldsymbol{\ell} \neq \boldsymbol{0}$, from Lemma 2.4 one can see that the size and the depth of $\phi_{\boldsymbol{\ell}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}$ are bounded as

$$W\left(\phi_{\boldsymbol{\ell}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}\right) \leq C\left(1+|\boldsymbol{\ell}|_1\left(\log|\boldsymbol{\ell}|_1 + \log\delta^{-1}\right)\right) \leq C\left(1+|\boldsymbol{\ell}|_1 \log\delta^{-1}\right) \tag{3.37}$$

and

$$L\left(\phi_{\boldsymbol{\ell}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}\right) \leq C\left(1+\log|\boldsymbol{\ell}|_1\left(\log|\boldsymbol{\ell}|_1 + \log\delta^{-1}\right)\right) \leq C\left(1+\log|\boldsymbol{\ell}|_1 \log\delta^{-1}\right) \tag{3.38}$$

due to the inequality $|\boldsymbol{\ell}|_1 \leq \delta^{-1}$. In the following we will use the convention $|\boldsymbol{0}|_1 = 1$. Then the estimates (3.37) and (3.38) holds true for $\boldsymbol{0} \leq \boldsymbol{\ell} \leq \boldsymbol{s}$.

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

$$\phi_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}(\boldsymbol{y}) := \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}-\boldsymbol{e}} b_{\boldsymbol{\ell}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}} \left(4\sqrt{\omega}\right)^{|\boldsymbol{\ell}|_1} \phi_{\boldsymbol{\ell}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}\left(\frac{\boldsymbol{y}}{4\sqrt{\omega}}\right), \quad \boldsymbol{y}\in\mathbb{R}^m, \tag{3.39}$$

which is the parallelization deep ReLU neural network of the component deep ReLU neural networks $\phi_{\boldsymbol{\ell}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}\left(\frac{\cdot}{4\sqrt{\omega}}\right)$. From (3.36) it follows

$$\operatorname{supp}\left(\phi_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}\right) \subset B_{4\omega}^{|\operatorname{supp}(\boldsymbol{s})|}. \tag{3.40}$$

We define $\boldsymbol{\phi}_{\Lambda(\xi)} := \left(\phi_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}\right)_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k})\in G(\xi)}$ as the deep ReLU neural network realized by parallelization of $\phi_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}$, $(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k}) \in G(\xi)$. Consider the approximation of $I^\omega_{\Lambda(\xi)} v$ by

$$\Phi_{\Lambda(\xi)} v(\boldsymbol{y}) := \sum_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k})\in G(\xi)} (-1)^{|\boldsymbol{e}|_1} v(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}) \phi_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}. \tag{3.41}$$

Lemma 3.4 *Under the assumptions of Theorem 3.2, for every $\xi > 1$, we have*

$$\left\| I^{\omega}_{\Lambda(\xi)} v - \Phi_{\Lambda(\xi)} u \right\|_{L_2(B^m_\omega, X, \gamma)} \le C \xi^{-(1/q - 1/2)}, \tag{3.42}$$

where the constant C is independent of v and ξ.

Proof. According to Lemma 3.1 the series (3.2) converges uncondionally to v. Hence, for every $\boldsymbol{y} \in B^m_\omega$, we have by (3.27)

$$I^{\omega}_{\Lambda(\xi)}(v)(\boldsymbol{y}) = \sum_{\boldsymbol{s} \in \Lambda(\xi)} \sum_{\boldsymbol{s}' \in \mathbb{F}} v_{\boldsymbol{s}'} \sum_{\boldsymbol{e} \in E_{\boldsymbol{s}}} (-1)^{|\boldsymbol{e}|_1} \sum_{\boldsymbol{k} \in \pi_{\boldsymbol{s}-\boldsymbol{e}}} H_{\boldsymbol{s}'}(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}) \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}-\boldsymbol{e}} b^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}_{\boldsymbol{\ell}} \left(4\sqrt{\omega}\right)^{|\boldsymbol{\ell}|_1} \prod_{j=1}^{m} \left(\frac{y_j}{4\sqrt{\omega}} \right)^{\ell_j}, \tag{3.43}$$

and by (3.41)

$$\Phi_{\Lambda(\xi)} v(\boldsymbol{y}) = \sum_{\boldsymbol{s} \in \Lambda(\xi)} \sum_{\boldsymbol{s}' \in \mathbb{F}} v_{\boldsymbol{s}'} \sum_{\boldsymbol{e} \in E_{\boldsymbol{s}}} (-1)^{|\boldsymbol{e}|_1} \sum_{\boldsymbol{k} \in \pi_{\boldsymbol{s}-\boldsymbol{e}}} H_{\boldsymbol{s}'}(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}) \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}-\boldsymbol{e}} b^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}_{\boldsymbol{\ell}} \left(4\sqrt{\omega}\right)^{|\boldsymbol{\ell}|_1} \phi^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}_{\boldsymbol{\ell}} \left(\frac{\boldsymbol{y}}{4\sqrt{\omega}} \right). \tag{3.44}$$

From these formulas and (3.35) we derive the inequality

$$\left\| I^{\omega}_{\Lambda(\xi)} v - \Phi_{\Lambda(\xi)} v \right\|_{L_2(B^m_\omega, X, \gamma)} \le \sum_{\boldsymbol{s} \in \Lambda(\xi)} \sum_{\boldsymbol{s}' \in \mathbb{F}} \|v_{\boldsymbol{s}'}\|_X \sum_{\boldsymbol{e} \in E_{\boldsymbol{s}}} \sum_{\boldsymbol{k} \in \pi_{\boldsymbol{s}-\boldsymbol{e}}} |H_{\boldsymbol{s}'}(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}})| \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}-\boldsymbol{e}} \left| b^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}_{\boldsymbol{\ell}} \right| \left(4\sqrt{\omega}\right)^{|\boldsymbol{\ell}|_1} \delta. \tag{3.45}$$

We have by (3.33)

$$\sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}-\boldsymbol{e}} \left| b^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}_{\boldsymbol{\ell}} \right| \le B_{\boldsymbol{s}} \prod_{j \in \nu_{\boldsymbol{s}-\boldsymbol{e}}} s_j \le p_{\boldsymbol{s}}(1) B_{\boldsymbol{s}},$$

and by Lemma A.3 in Appendix

$$\sum_{\boldsymbol{e} \in E_{\boldsymbol{s}}} \sum_{\boldsymbol{k} \in \pi_{\boldsymbol{s}-\boldsymbol{e}}} |H_{\boldsymbol{s}'}(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}})| \le \sum_{\boldsymbol{e} \in E_{\boldsymbol{s}}} e^{K|\boldsymbol{s}-\boldsymbol{e}|_1} \le 2^{|\boldsymbol{s}|_0} e^{K|\boldsymbol{s}|_1} \le p_{\boldsymbol{s}}(1) e^{K|\boldsymbol{s}|_1}. \tag{3.46}$$

This together with (3.45), Lemma 3.1 and (3.34) yields that

$$\begin{aligned} \left\| I^{\omega}_{\Lambda(\xi)} v - \Phi_{\Lambda(\xi)} v \right\|_{L_2(B^m_\omega, X, \gamma)} &\le \sum_{\boldsymbol{s} \in \Lambda(\xi)} \delta B_{\boldsymbol{s}} p_{\boldsymbol{s}}(1) \sum_{\boldsymbol{s}' \in \mathbb{F}} \|v_{\boldsymbol{s}'}\|_X \left(4\sqrt{\omega}\right)^{|\boldsymbol{s}|_1} \sum_{\boldsymbol{e} \in E_{\boldsymbol{s}}} \sum_{\boldsymbol{k} \in \pi_{\boldsymbol{s}-\boldsymbol{e}}} |H_{\boldsymbol{s}'}(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}})| \\ &\le \sum_{\boldsymbol{s}' \in \mathbb{F}} \|v_{\boldsymbol{s}'}\|_X \ \delta \sum_{\boldsymbol{s} \in \Lambda(\xi)} e^{K|\boldsymbol{s}|_1} p_{\boldsymbol{s}}(2) \left(4\sqrt{\omega}\right)^{|\boldsymbol{s}|_1} B_{\boldsymbol{s}} \\ &\le C \xi^{-(1/q - 1/2)}. \end{aligned}$$

∎

In the previous lemma, we proved the bound of the third term in the right-hand side of (3.25), i.e., the error bound for the approximation of $I^{\omega}_{\Lambda(\xi)} v$ by the function $\Phi_{\Lambda(\xi)} v$ for $v \in L_2(\mathbb{R}^\infty, X, \gamma)$. As the last step in the error estimation, we will establish the bound for the fourth term in the right-hand side of (3.25).

Lemma 3.5 *Under the assumptions of Theorem 3.2, for every $\xi > 1$, we have*

$$\left\|\Phi_{\Lambda(\xi)} v\right\|_{L_2((\mathbb{R}^m \setminus B^m_\omega), X, \gamma)} \le C \xi^{-(1/q - 1/2)}, \tag{3.47}$$

where the constant C is independent of v and ξ.

Proof. We use the formula (3.44) to estimate the norm $\left\|\Phi_{\Lambda(\xi)} v\right\|_{L_2((\mathbb{R}^m \setminus B^m_\omega), X, \gamma)}$. From (3.35) one can easily see that $\left|\phi_{\boldsymbol{\ell}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}\left(\frac{\boldsymbol{y}}{4\sqrt{\omega}}\right)\right| \le 2, \ \forall \boldsymbol{y} \in \mathbb{R}^m$. Hence, by Lemma A.7 in Appendix,

$$\left\|\phi_{\boldsymbol{\ell}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}\left(\frac{\cdot}{4\sqrt{\omega}}\right)\right\|_{L_2(\mathbb{R}^m \setminus B^m_\omega, \gamma)} \le 2\left\|1\right\|_{L_2(\mathbb{R}^m \setminus B^m_\omega, \gamma)} \le C_1 m \exp\left(-K_1 \omega\right).$$

This together with (3.44) implies that

$$\begin{aligned}
&\left\|\Phi_{\Lambda(\xi)} v\right\|_{L_2(\mathbb{R}^m \setminus B^m_\omega, X, \gamma)} \\
&\le \sum_{\boldsymbol{s} \in \Lambda(\xi)} \sum_{\boldsymbol{s}' \in \mathbb{F}} \|v_{\boldsymbol{s}'}\|_X \sum_{\boldsymbol{e} \in E_{\boldsymbol{s}}} \sum_{\boldsymbol{k} \in \pi_{\boldsymbol{s}-\boldsymbol{e}}} \left|H_{\boldsymbol{s}'}(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}})\right| \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}-\boldsymbol{e}} \left|b_{\boldsymbol{\ell}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}\right| \left(4\sqrt{\omega}\right)^{|\boldsymbol{\ell}|_1} \left\|\phi_{\boldsymbol{\ell}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}\left(\frac{\cdot}{4\sqrt{\omega}}\right)\right\|_{L_2(\mathbb{R}^m \setminus B^m_\omega, \gamma)} \\
&\le C_1 m \exp\left(-K_1 \omega\right) \sum_{\boldsymbol{s} \in \Lambda(\xi)} \left(4\sqrt{\omega}\right)^{|\boldsymbol{s}|_1} \sum_{\boldsymbol{s}' \in \mathbb{F}} \|v_{\boldsymbol{s}'}\|_X \sum_{\boldsymbol{e} \in E_{\boldsymbol{s}}} \sum_{\boldsymbol{k} \in \pi_{\boldsymbol{s}-\boldsymbol{e}}} \left|H_{\boldsymbol{s}'}(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}})\right| \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}-\boldsymbol{e}} \left|b_{\boldsymbol{\ell}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}\right|.
\end{aligned}$$

By a tensor product argument from Lemma A.6 in Appendix and the inequality $\boldsymbol{s} - \boldsymbol{e} \le \boldsymbol{s}$ for $\boldsymbol{e} \in E_{\boldsymbol{s}}$, we deduce the estimates

$$\sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}-\boldsymbol{e}} \left|b_{\boldsymbol{\ell}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}\right| \le e^{K_2 |\boldsymbol{s}|_1} \boldsymbol{s}! \le e^{K_2 |\boldsymbol{s}|_1} |\boldsymbol{s}|_1^{|\boldsymbol{s}|_1}, \tag{3.48}$$

which and (3.46) give

$$\sum_{\boldsymbol{k} \in \pi_{\boldsymbol{s}-\boldsymbol{e}}} \left|H_{\boldsymbol{s}'}(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}})\right| \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}-\boldsymbol{e}} \left|b_{\boldsymbol{\ell}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}\right| \le \sum_{\boldsymbol{k} \in \pi_{\boldsymbol{s}-\boldsymbol{e}}} \left|H_{\boldsymbol{s}'}(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}})\right| e^{K_2 |\boldsymbol{s}|_1} |\boldsymbol{s}|_1^{|\boldsymbol{s}|_1} \le p_{\boldsymbol{s}}(1) e^{K_2 |\boldsymbol{s}|_1} |\boldsymbol{s}|_1^{|\boldsymbol{s}|_1}. \tag{3.49}$$

This in combining with (3.22), (3.29), Lemma 3.1 allows us to continue the estimation as

$$\begin{aligned}
\left\|\Phi_{\Lambda(\xi)} v\right\|_{L_2(\mathbb{R}^m \setminus B^m_\omega, X, \gamma)} &\le C_1 m \exp\left(-K_1 \omega\right) \sum_{\boldsymbol{s}' \in \mathbb{F}} \|v_{\boldsymbol{s}'}\|_X \sum_{\boldsymbol{s} \in \Lambda(\xi)} \left(4\sqrt{\omega}\right)^{|\boldsymbol{s}|_1} p_{\boldsymbol{s}}(1) e^{K_2 |\boldsymbol{s}|_1} |\boldsymbol{s}|_1^{|\boldsymbol{s}|_1} \\
&\le C_2 m \exp\left(-K_1 \omega\right) \sum_{\boldsymbol{s} \in \Lambda(\xi)} \left(4\sqrt{\omega}\right)^{|\boldsymbol{s}|_1} p_{\boldsymbol{s}}(1) e^{K_2 |\boldsymbol{s}|_1} |\boldsymbol{s}|_1^{|\boldsymbol{s}|_1} \\
&\le C_2 \xi \exp\left(-K_1 \xi\right) \left(C_3 \xi^{1/2}\right)^{m_1(\xi)} e^{K_2 m_1(\xi)} \left[m_1(\xi)\right]^{m_1(\xi)} \sum_{\boldsymbol{s} \in \Lambda(\xi)} p_{\boldsymbol{s}}(1).
\end{aligned}$$

Hence, by Lemma A.1(ii) in Appendix we have that

$$\begin{aligned}
\left\|\Phi_{\Lambda(\xi)} v\right\|_{L_2(\mathbb{R}^m \setminus B^m_\omega, X, \gamma)} &\le C_2 \xi \exp\left(-K_1 \xi\right) \left(C_3 \xi^{1/2}\right)^{K_{q,\theta} \xi^{1/\theta q}} e^{K_3 K_{q,\theta} \xi^{1/\theta q}} \left(K_{q,\theta} \xi^{1/\theta q}\right)^{K_{q,\theta} \xi^{1/\theta q}} C_4 \xi \\
&\le C_5 \xi^2 \exp(-K_1 \xi + K_4 \xi^{1/\theta q} \log \xi + K_5 \xi^{1/\theta q}).
\end{aligned}$$

Since $1/\theta q \le 1/3$, we obtain

$$\left\|\Phi_{\Lambda(\xi)} v\right\|_{L_2(\mathbb{R}^m \setminus B^m_\omega, X, \gamma)} \le C\xi^{-(1/q-1/2)}.$$

□

To complete the proof of Theorem 3.2, we have to establish the bounds of the size and depth of the deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi)}$ as in (iii) and (iv).

Lemma 3.6 *Under the assumptions of Theorem 3.2, the input dimension of $\boldsymbol{\phi}_{\Lambda(\xi)}$ is at most $\lfloor K_q \xi \rfloor$, for every $\xi > 1$, the output dimension of $\boldsymbol{\phi}_{\Lambda(\xi)}$ at most $\lfloor C_q \xi \rfloor$,*

$$W\big(\boldsymbol{\phi}_{\Lambda(\xi)}\big) \le C\xi^{1+2/\theta q} \log \xi, \tag{3.50}$$

and

$$L\big(\boldsymbol{\phi}_{\Lambda(\xi)}\big) \le C\xi^{1/\theta q} (\log \xi)^2, \tag{3.51}$$

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.1(iii) in Appendix. The output dimension of $\boldsymbol{\phi}_{\Lambda(\xi)}$ is the number $|G(\xi)|$ which is at most $\lfloor C_q \xi \rfloor$ by Lemma A.2 in Appendix.

By Lemmas 2.1 and 2.4 and (3.37) the size of $\boldsymbol{\phi}_{\Lambda(\xi)}$ is estimated as

$$W\big(\boldsymbol{\phi}_{\Lambda(\xi)}\big) \le \sum_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k}) \in G(\xi)} W\left(\phi_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}\right) \le \sum_{\boldsymbol{s} \in \Lambda(\xi)} \sum_{\boldsymbol{e} \in E_{\boldsymbol{s}}} \sum_{\boldsymbol{k} \in \pi_{\boldsymbol{s}-\boldsymbol{e}}} \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}-\boldsymbol{e}} W\left(\phi_{\boldsymbol{\ell}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}\right) \tag{3.52}$$

$$\le C_1 \sum_{\boldsymbol{s} \in \Lambda(\xi)} \sum_{\boldsymbol{e} \in E_{\boldsymbol{s}}} \sum_{\boldsymbol{k} \in \pi_{\boldsymbol{s}-\boldsymbol{e}}} \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}-\boldsymbol{e}} \left(1 + |\boldsymbol{\ell}|_1 \log \delta^{-1}\right), \tag{3.53}$$

where we recall,

$$\delta^{-1} := \xi^{1/q-1/2} \sum_{\boldsymbol{s} \in \Lambda(\xi)} e^{K_1 |\boldsymbol{s}|_1} p_{\boldsymbol{s}}(2) \left(4\sqrt{\omega}\right)^{|\boldsymbol{s}|_1} B_{\boldsymbol{s}},$$

$$B_{\boldsymbol{s}} := \max_{\boldsymbol{e} \in E_{\boldsymbol{s}},\, \boldsymbol{k} \in \pi_{\boldsymbol{s}-\boldsymbol{e}}} \ \max_{\boldsymbol{0} \le \boldsymbol{\ell} \le \boldsymbol{s}-\boldsymbol{e}} \left|b_{\boldsymbol{\ell}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}\right|.$$

From (3.48) it follows that

$$B_{\boldsymbol{s}} \le \max_{\boldsymbol{e} \in E_{\boldsymbol{s}},\, \boldsymbol{k} \in \pi_{\boldsymbol{s}-\boldsymbol{e}}} \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}-\boldsymbol{e}} \left|b_{\boldsymbol{\ell}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}\right| \le \exp\left(K_2 \xi^{1/\theta q} \log \xi\right),$$

which by Lemma A.1(i) in Appendix implies

$$\delta^{-1} \le \xi^{1/q-1/2} \exp\left(K_2 \xi^{1/\theta q} \log \xi\right) \sum_{\boldsymbol{s} \in \Lambda(\xi)} p_{\boldsymbol{s}}(2)$$
$$\le C_2 \xi^{1/q+1/2} \exp\left(K_3 \xi^{1/\theta q} \log \xi\right) \le C_2 \exp\left(K_3 \xi^{1/\theta q} \log \xi\right).$$

Hence,

$$\log(\delta^{-1}) \le K_4 \xi^{1/\theta q} \log \xi. \tag{3.54}$$

and consequently,

$$\left(1 + |\boldsymbol{\ell}|_1 \log \delta^{-1}\right) \le \left(1 + |\boldsymbol{s}|_1 K_4 \xi^{1/\theta q} \log \xi\right) \le C_2 \xi^{2/\theta q} \log \xi.$$

From (3.52)–(3.53) and Lemma A.2 in Appendix we obtain the desired bound of the size of $\boldsymbol{\phi}_{\Lambda(\xi)}$:

$$W\left(\boldsymbol{\phi}_{\Lambda(\xi)}\right) \le C_2 \xi^{2/\theta q} \log \xi \sum_{\boldsymbol{s} \in \Lambda(\xi)} \sum_{\boldsymbol{e} \in E_{\boldsymbol{s}}} \sum_{\boldsymbol{k} \in \pi_{\boldsymbol{s}-\boldsymbol{e}}} \sum_{\boldsymbol{\ell}=\boldsymbol{0}}^{\boldsymbol{s}-\boldsymbol{e}} 1$$
$$\le C_2 \xi^{2/\theta q} \log \xi \sum_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k}) \in G(\xi)} p_{\boldsymbol{s}}(1) \le C_3 \xi^{1+2/\theta q} \log \xi.$$

By using Lemma 2.1, (3.38), (3.54) and Lemma A.1(ii) in Appendix, we prove that the depth of $\boldsymbol{\phi}_{\Lambda(\xi)}$ is bounded as in (3.51):

$$L\left(\boldsymbol{\phi}_{\Lambda(\xi)}\right) \le \max_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k}) \in G(\xi)} L\left(\phi_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}\right) \le \max_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k}) \in G(\xi)} \max_{\boldsymbol{0} \le \boldsymbol{\ell} \le \boldsymbol{s}-\boldsymbol{e}} L\left(\phi_{\boldsymbol{\ell}}^{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}\right)$$
$$\le C_4 \max_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k}) \in G(\xi)} \max_{\boldsymbol{0} \le \boldsymbol{\ell} \le \boldsymbol{s}-\boldsymbol{e}} \left(1 + \log |\boldsymbol{\ell}|_1 \log \delta^{-1}\right)$$
$$\le C_4 \max_{\boldsymbol{s} \in \Lambda(\xi)} \left(1 + \log |\boldsymbol{s}|_1 \log \delta^{-1}\right)$$
$$\le C_4 \max_{\boldsymbol{s} \in \Lambda(\xi)} \left(1 + \log\left(K_{q,\theta} \xi^{1/\theta q}\right) \left(K_5 \xi^{1/\theta q} \log \xi\right)\right) \le C_5 \xi^{1/\theta q} (\log \xi)^2.$$

∎

We are now in a position to give a formal proof of Theorem 3.2.

Proof. [Proofs of Theorem 3.2] From (3.25), Theorem 3.1 and Lemmas 3.3–3.5, for every $\xi > 1$, we deduce that

$$\left\|v - \Phi_{\Lambda(\xi)} v\right\|_{L_2(\mathbb{R}^\infty, X, \gamma)} \le C \xi^{-(1/q-1/2)}.$$

The claim (vi) is proven. The claim (i) follows directly from the construction of the deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi)}$ and the sequence of points $Y_{\Lambda(\xi)}$, the claims (ii)–(iv) from Lemma 3.6 and the claim (v) from Lemma A.1(iii) in Appendix and (3.40). Thus, Theorem 3.2 is proven for the case when $U = \mathbb{R}^\infty$.

The case $U=\mathbb{R}^M$ can be proven in the same way with a slight modification. Counterparts of all definitions, formulas and assertions which have been used in the proof of the case $U=\mathbb{R}^\infty$, are true for the case $U=\mathbb{R}^M$. In the proof of this case, in parlicular, the used equality $\|H_{\boldsymbol{s}}\|_{L_2(\mathbb{R}^\infty)}=1$, $\boldsymbol{s}\in\mathbb{F}$, is replaced by the inequality $\|H_{\boldsymbol{s}}\|_{L_\infty^{\sqrt{g}}(\mathbb{R}^M)}<1$, $\boldsymbol{s}\in\mathbb{N}_0^M$. □

4 Application to parametrized elliptic PDEs

In this section, we apply the results in the previous section to the deep ReLU neural network approximation of the solution u to the parametrized elliptic PDEs (1.2) with lognormal inputs (1.3). This is based on the weighted ℓ_2-summability of the series $(\|u_{\boldsymbol{s}}\|_V)_{\boldsymbol{s}\in\mathcal{F}}$ in following lemma which has been proven in [4, Theorems 3.3 and 4.2].

Lemma 4.1 *Assume that there exist a number $0<q<\infty$ and an increasing sequence $\boldsymbol{\rho}=(\rho_j)_{j\in\mathcal{N}}$ of numbers strictly larger than 1 such that $\left\|\boldsymbol{\rho}^{-1}\right\|_{\ell_q(\mathcal{N})}\le C<\infty$ and*

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

where the constants C are independent of J. Then we have that for any $\eta\in\mathcal{N}$,

$$\sum_{\boldsymbol{s}\in\mathcal{F}}(\sigma_{\boldsymbol{s}}\|u_{\boldsymbol{s}}\|_V)^2\le C<\infty\quad \text{with}\quad \sigma_{\boldsymbol{s}}^2:=\sum_{\|\boldsymbol{s}'\|_{\ell_\infty(\mathcal{F})}\le\eta}\binom{\boldsymbol{s}}{\boldsymbol{s}'}\prod_{j\in\mathcal{N}}\rho_j^{2s_j'},\tag{4.1}$$

where the constant C is independent of J.

The following two lemmas are proven in [15, Lemmas 5.2 and 5.3].

Lemma 4.2 *Let the assumptions of Lemma 4.1 hold. Then the solution map $\boldsymbol{y}\mapsto u(\boldsymbol{y})$ is γ-measurable and $u\in L_2(U,V,\gamma)$. Moreover, $u\in L_2^{\mathcal{E}}(U,V,\gamma)$ where*

$$\mathcal{E}:=\left\{\boldsymbol{y}\in\mathbb{R}^\infty:\ \sup_{j\in\mathbb{N}}\rho_j^{-1}|y_j|<\infty\right\}\tag{4.2}$$

having $\gamma(\mathcal{E})=1$ and containing all $\boldsymbol{y}\in\mathbb{R}^\infty$ with $|\boldsymbol{y}|_0<\infty$ in the case when $U=\mathbb{R}^\infty$.

Lemma 4.3 *Let $0<q<\infty$, $\boldsymbol{\rho}=(\rho_j)_{j\in\mathcal{N}}$ be a sequence of positive numbers such that $\left\|\boldsymbol{\rho}^{-1}\right\|_{\ell_q(\mathcal{N})}\le C<\infty$, where the constant C is independent of J. Let θ be an arbitrary non-negative number and $\boldsymbol{p}(\theta)=(p_{\boldsymbol{s}}(\theta))_{\boldsymbol{s}\in\mathcal{F}}$ the sequence given as in (3.11). For $\eta\in\mathbb{N}$, let the sequence $\boldsymbol{\sigma}=(\sigma_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathcal{F}}$ be defined as in (4.1). Then for any $\eta>\frac{2(\theta+1)}{q}$, we have*

$$\left\|\boldsymbol{p}(\theta)\boldsymbol{\sigma}^{-1}\right\|_{\ell_q(\mathcal{F})}\le C<\infty,$$

where the constant C is independent of J.

We are now in position to formulate our main results on collocation deep ReLU neural network approximation of the solution u to parametric elliptic PDEs with lognormal inputs.

Theorem 4.1 *Under the assumptions of Lemma 4.1, let $0 < q < 2$. Then, given an arbitrary number $\delta > 0$, for every integer $n > 1$, we can construct a deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi_n)} := (\phi_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}})_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k})\in G(\xi_n)}$ of the size $W(\boldsymbol{\phi}_{\Lambda(\xi_n)}) \le n$ on $\mathbb{R}^m$ with*

$$m := \begin{cases} \min\left\{M, \left\lfloor K\left(\frac{n}{\log n}\right)^{\frac{1}{1+\delta}}\right\rfloor\right\} & \text{if } U = \mathbb{R}^M, \\ \left\lfloor K\left(\frac{n}{\log n}\right)^{\frac{1}{1+\delta}}\right\rfloor & \text{if } U = \mathbb{R}^\infty, \end{cases}$$

and a sequence of points $Y_{\Lambda(\xi_n)} := (\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}})_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k})\in G(\xi_n)}$ having the following properties.

(i) *The deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi_n)}$ and sequence of points $Y_{\Lambda(\xi_n)}$ are independent of u;*

(ii) *The output dimension of $\boldsymbol{\phi}_{\Lambda(\xi_n)}$ is at most $\left\lfloor K\left(\frac{n}{\log n}\right)^{\frac{1}{1+\delta}}\right\rfloor$;*

(iii) $L(\boldsymbol{\phi}_{\Lambda(\xi_n)}) \le C_\delta \left(\frac{n}{\log n}\right)^{\frac{\delta}{2(1+\delta)}} (\log n)^2$*;*

(iv) $\operatorname{supp}(\boldsymbol{\phi}_{\Lambda(\xi_n)}) \subset [-T,T]^m$*, where* $T := C_\delta \left(\frac{n}{\log n}\right)^{\frac{1}{2(1+\delta)}}$*;*

(v) *The approximation of u by $\Phi_{\Lambda(\xi_n)}u$ defined as in* (3.18)*, gives the error estimate*

$$\|u - \Phi_{\Lambda(\xi_n)}u\|_{\mathcal{L}(U,V)} \le C\left(\frac{n}{\log n}\right)^{-\frac{1}{1+\delta}\left(\frac{1}{q}-\frac{1}{2}\right)}.$$

Here the constants C, K and C_δ are independent of J, 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 $\delta \le 1/6$. We take first the number $\theta := 2/\delta q$ satisfying the inequality $\theta \ge 3/q$, and then choose a number $\eta \in \mathbb{N}$ satisfying the inequality $\eta > 2(\theta+1)/q$. By using Lemmas 4.1–4.3, one can check that $u \in L_2^{\mathcal{E}}(U,V,\gamma)$ satisfies the assumptions of Theorem 3.2 for $X = V$ and the sequence $(\sigma_{\boldsymbol{s}})_{\boldsymbol{s}\in\mathbb{F}}$ defined as in (4.1), where $\mathcal{E}$ is the set defined in Lemma 4.2. For a given integer $n > 1$, we choose $\xi_n > 1$ as the maximal number satisfying the inequality $C\xi_n^\delta \log \xi_n \le n$, 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

$$C_1\left(\frac{n}{\log n}\right)^{\frac{1}{1+\delta}} \le \xi_n \le C_2\left(\frac{n}{\log n}\right)^{\frac{1}{1+\delta}}.$$

From Theorem 3.2 with $\xi = \xi_n$ we deduce the desired results. ∎

From Theorem 4.1 one can directly derive the following

Theorem 4.2 *Under the assumptions of Lemma 4.1, let $0 < q < 2$ and $\delta_q := \min(1, 1/q - 1/2)$. Then, given an arbitrary number $\delta \in (0, \delta_q)$, for every integer $n > 1$, we can construct a deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi_n)} := (\phi_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}})_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k})\in G(\xi_n)}$ of the size $W\big(\boldsymbol{\phi}_{\Lambda(\xi_n)}\big) \le n$ on $\mathbb{R}^m$ with*

$$m := \begin{cases} \min\left\{M, \lfloor Kn^{1-\delta}\rfloor\right\} & \text{if} \quad U = \mathbb{R}^M, \\ \lfloor Kn^{1-\delta}\rfloor & \text{if} \quad U = \mathbb{R}^\infty, \end{cases}$$

and a sequence of points $Y_{\Lambda(\xi_n)} := (\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}})_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k})\in G(\xi_n)}$ having the following properties.

(i) *The deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi_n)}$ and sequence of points $Y_{\Lambda(\xi_n)}$ are independent of u;*

(ii) *The output dimension of $\boldsymbol{\phi}_{\Lambda(\xi_n)}$ are at most $\lfloor Kn^{1-\delta}\rfloor$;*

(iii) $L\big(\boldsymbol{\phi}_{\Lambda(\xi_n)}\big) \le C_\delta n^\delta$;

(iv) $\operatorname{supp}\big(\boldsymbol{\phi}_{\Lambda(\xi_n)}\big) \subset [-T, T]^m$, *where $T := C_\delta n^{1-\delta}$;*

(v) *The approximation of u by $\Phi_{\Lambda(\xi_n)}u$ defined as in (3.18), gives the error estimates*

$$\|u - \Phi_{\Lambda(\xi_n)}u\|_{\mathcal{L}(U,V)} \le Cm^{-\left(\frac{1}{q}-\frac{1}{2}\right)} \le C_\delta n^{-(1-\delta)\left(\frac{1}{q}-\frac{1}{2}\right)}. \tag{4.3}$$

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

Let us compare the collocation approximation of u by the function

$$\Phi_{\Lambda(\xi_n)}u := \sum_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k})\in G(\xi_n)} (-1)^{|\boldsymbol{e}|_1} u(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}})\phi_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}, \tag{4.4}$$

generated from the deep ReLU neural network $\boldsymbol{\phi}_{\Lambda(\xi_n)}$ as in Theorem 4.2, and the collocation approximation of u by the sparse-grid Lagrange gpc interpolation

$$I_{\Lambda(\xi_n)}u := \sum_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k})\in G(\xi_n)} (-1)^{|\boldsymbol{e}|_1} u(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}})L_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}}. \tag{4.5}$$

Both the methods are based on m the same particular solvers $\big(u(\boldsymbol{y}_{\boldsymbol{s}-\boldsymbol{e};\boldsymbol{k}})\big)_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k})\in G(\xi_n)}$. From Corollary 3.1 one can see that under the assumptions of Theorem 4.2, there holds the error bound in m for the last approximation:

$$\left\|u - I_{\Lambda(\xi_n)}u\right\|_{\mathcal{L}(U,V)} \le Cm^{-\left(\frac{1}{q}-\frac{1}{2}\right)},$$

which is the same as that in (4.3) for the first approximation since by the construction the parameter m in (4.3) can be treated as independent.

Acknowledgments. This work 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 author was working at the Vietnam Institute for Advanced Study in Mathematics (VIASM). He would like to thank the VIASM for providing a fruitful research environment and working condition.

A Appendix

A.1 Auxiliary lemmas

Lemma A.1 *Let $\theta \geq 0$ and $0 < q < \infty$. Let $\boldsymbol{\sigma} = (\sigma_{\boldsymbol{s}})_{\boldsymbol{s} \in \mathcal{F}}$ be a sequence of numbers strictly larger than 1. Then for every we have the following.*

(i) *If $\left\| \boldsymbol{p}\left(\frac{\theta}{q}\right) \boldsymbol{\sigma}^{-1} \right\|_{\ell_q(\mathcal{F})} \leq K < \infty$, where the constant K is independent of J, then*

$$\sum_{\boldsymbol{s} \in \Lambda(\xi)} p_{\boldsymbol{s}}(\theta) \leq K\xi \quad \forall \xi > 1. \tag{A.1}$$

In particular, if $\left\|\boldsymbol{\sigma}^{-1}\right\|_{\ell_q(\mathcal{F})}^q \leq K_q < \infty$, where the constant K_q is independent of J, then the set $\Lambda(\xi)$ is finite and

$$|\Lambda(\xi)| \leq K_q \xi \quad \forall \xi > 1. \tag{A.2}$$

(ii) *If $\left\|\boldsymbol{p}(\theta)\boldsymbol{\sigma}^{-1}\right\|_{\ell_q(\mathcal{F})}^{1/\theta} \leq K_{q,\theta} < \infty$, where the constant $K_{q,\theta}$ is independent of J, then*

$$m_1(\xi) \leq K_{q,\theta} \xi^{\frac{1}{\theta q}} \quad \forall \xi > 1. \tag{A.3}$$

(iii) *If $\sigma_{\boldsymbol{e}^{i'}} \leq \sigma_{\boldsymbol{e}^i}$ for $i' < i$, and if $\left\|\boldsymbol{\sigma}^{-1}\right\|_{\ell_q(\mathcal{F})}^q \leq K_q < \infty$, where the constant K_q is independent of J, then*

$$m(\xi) \leq K_q \xi \quad \forall \xi > 1. \tag{A.4}$$

Proof. The claim (ii) and (iii) were proven in [17, Lemmas A.1–A.2] for the case $\mathcal{F} = \mathbb{F}$. The case $\mathcal{F} = \mathbb{N}_0^M$ can be proven in a similar way. Let us prove the claim (i). Indeed, we have for every $\xi > 1$,

$$\sum_{\boldsymbol{s} \in \Lambda(\xi)} p_{\boldsymbol{s}}(\theta) \leq \sum_{\boldsymbol{s} \in \mathcal{F}:\ \sigma_{\boldsymbol{s}}^{-q}\xi \geq 1} p_{\boldsymbol{s}}(\theta)\xi\sigma_{\boldsymbol{s}}^{-q} \leq \xi \sum_{\boldsymbol{s} \in \mathcal{F}} p_{\boldsymbol{s}}(\theta)\sigma_{\boldsymbol{s}}^{-q} \leq C\xi.$$

□

Lemma A.2 *Let $\theta \geq 0$, $0 < q < \infty$ and $\xi > 1$. Let $\boldsymbol{\sigma} = (\sigma_{\boldsymbol{s}})_{\boldsymbol{s} \in \mathcal{F}}$ be a sequence of numbers strictly larger than 1. If and $\left\| \boldsymbol{p}\left(\frac{\theta+2}{q}\right) \boldsymbol{\sigma}^{-1} \right\|_{\ell_q(\mathcal{F})} \leq C < \infty$, where the constant C is independent of J, then there holds*

$$\sum_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k}) \in G(\xi)} p_{\boldsymbol{s}}(\theta) \leq C\xi \quad \forall \xi > 1. \tag{A.5}$$

In particular, if and $\left\| \boldsymbol{p}\left(\frac{2}{q}\right) \boldsymbol{\sigma}^{-1} \right\|_{\ell_q(\mathcal{F})}^q \leq C_q < \infty$, where the constant C is independent of J, then

$$|G(\xi)| \leq C_q \xi \quad \forall \xi > 1.$$

Proof. We have for every $\xi > 1$,

$$\sum_{(\boldsymbol{s},\boldsymbol{e},\boldsymbol{k})\in G(\xi)} p_{\boldsymbol{s}}(\theta) = \sum_{\boldsymbol{s}\in\Lambda(\xi)} \sum_{\boldsymbol{e}\in E_{\boldsymbol{s}}} \sum_{\boldsymbol{k}\in\pi_{\boldsymbol{s}-\boldsymbol{e}}} p_{\boldsymbol{s}}(\theta) \leq \sum_{\boldsymbol{s}\in\Lambda(\xi)} p_{\boldsymbol{s}}(\theta) \sum_{\boldsymbol{e}\in E_{\boldsymbol{s}}} |\pi_{\boldsymbol{s}-\boldsymbol{e}}| \tag{A.6}$$

$$\leq \sum_{\boldsymbol{s}\in\Lambda(\xi)} p_{\boldsymbol{s}}(\theta)|E_{\boldsymbol{s}}|p_{\boldsymbol{s}}(1) = \sum_{\boldsymbol{s}\in\Lambda(\xi)} p_{\boldsymbol{s}}(\theta+1)2^{|\boldsymbol{s}|_0} \leq \sum_{\boldsymbol{s}\in\Lambda(\xi)} p_{\boldsymbol{s}}(\theta+2) \tag{A.7}$$

$$\leq \sum_{\boldsymbol{s}\in\mathcal{F}:\ \sigma_{\boldsymbol{s}}^{-q}\xi\geq 1} p_{\boldsymbol{s}}(\theta+2)\xi\sigma_{\boldsymbol{s}}^{-q} \leq \xi \sum_{\boldsymbol{s}\in\mathcal{F}} p_{\boldsymbol{s}}(\theta+2)\sigma_{\boldsymbol{s}}^{-q} \leq C\xi.$$

□

Lemma A.3 *We have for any $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{F}$,*

$$\sum_{\boldsymbol{k}\in\pi_{\boldsymbol{s}}} |H_{\boldsymbol{s}'}(\boldsymbol{y}_{\boldsymbol{s};\boldsymbol{k}})| \leq e^{K|\boldsymbol{s}|_1}, \tag{A.8}$$

where the constant K is independent of J and $\boldsymbol{s}, \boldsymbol{s}'$.

Proof. From Cramér's bound we deduce that (see, e.g., [15, Lemma 3.2])

$$|H_s(y)\sqrt{g(y)}| \ < \ 1, \quad \forall y \in \mathbb{R},\ \forall s \in \mathbb{N}_0, \tag{A.9}$$

or, equivalently,

$$|H_s(y)| \ < \ (2\pi)^{1/4}e^{y^2/4}, \quad \forall y \in \mathbb{R},\ \forall s \in \mathbb{N}_0. \tag{A.10}$$

Let $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{F}$ and $\boldsymbol{k} \in \pi_{\boldsymbol{s}}$ be given. Notice that for the univariate Hermite polynomials, $H_s(0) = 0$ if $s > 0$, and $H_0 = 1$. Hence, $H_{\boldsymbol{s}'}(\boldsymbol{y}_{\boldsymbol{s};\boldsymbol{k}}) = \prod_{j\in\nu_{\boldsymbol{s}'}} H_{s'_j}(y_{s_j-e_j,k_j}) = 0$ if $\nu_{\boldsymbol{s}'} \not\subset \nu_{\boldsymbol{s}}$. For the case when $\nu_{\boldsymbol{s}'} \subset \nu_{\boldsymbol{s}}$, we have by (A.10),

$$|H_{\boldsymbol{s}'}(\boldsymbol{y}_{\boldsymbol{s};\boldsymbol{k}})| = \prod_{j\in\nu_{\boldsymbol{s}'}} |H_{s'_j}(y_{s_j,k_j})| \leq \prod_{j\in\nu_{\boldsymbol{s}'}} (2\pi)^{1/4}e^{y^2_{s_j,k_j}/4} \leq \prod_{j\in\nu_{\boldsymbol{s}}} (2\pi)^{1/4}e^{y^2_{s_j,k_j}/4}. \tag{A.11}$$

Therefore,

$$\sum_{\boldsymbol{k}\in\pi_{\boldsymbol{s}}} |H_{\boldsymbol{s}'}(\boldsymbol{y}_{\boldsymbol{s};\boldsymbol{k}})| \leq \sum_{\boldsymbol{k}\in\pi_{\boldsymbol{s}}} \prod_{j\in\nu_{\boldsymbol{s}}} (2\pi)^{1/4}e^{y^2_{s_j,k_j}/4} = \prod_{j\in\nu_{\boldsymbol{s}}} (2\pi)^{1/4} \sum_{k_j\in\pi_{s_j}} e^{y^2_{s_j,k_j}/4}. \tag{A.12}$$

The inequalities [54, (6.31.19)] yield that

$$|y_{s;k}| \ \leq \ K_1\frac{|k|}{\sqrt{s}}, \quad \forall k \in \pi_s,\ \forall s \in \mathbb{N}. \tag{A.13}$$

Consequently,

$$(2\pi)^{1/4} \sum_{k_j\in\pi_{s_j}} e^{y^2_{s_j,k_j}/4} \ \leq \ 2(2\pi)^{1/4} \sum_{k_j=0}^{\lfloor s_j/2\rfloor} \exp\left(\frac{K_1}{4}\frac{k_j^2}{s_j}\right) \ \leq \ e^{Ks_j}, \quad \forall s_j \in \mathbb{N}. \tag{A.14}$$

This allows us to finish the proof of the lemma as

$$\sum_{\boldsymbol{k}\in\pi_{\boldsymbol{s}}} |H_{\boldsymbol{s}'}(\boldsymbol{y}_{\boldsymbol{s};\boldsymbol{k}})| \le \prod_{j\in\nu_{\boldsymbol{s}}} e^{Ks_j} = e^{K|\boldsymbol{s}|_1}.$$

∎

Lemma A.4 *We have for any $s \in \mathbb{N}$ and $k \in \pi_s$,*

$$\|L_{s;k}\|_{L_2(\mathbb{R},\gamma)} \le e^{Ks}, \tag{A.15}$$

and

$$\|L_{s;k}\|_{L_\infty^{\sqrt{g}}(\mathbb{R})} \le e^{Ks}, \tag{A.16}$$

where the constants K are independent of s and $k \in \pi_s$.

Proof. Notice that $L_{s;k}$ is a polynomial having s single zeros $\{y_{s;j}\}_{j\in\pi_s,\, j\neq k}$, and that $L_{s;k}(y_{s;k}) = 1$. Moreover, there is no any zero in the open interval $(y_{s;k-1}, y_{s;k})$ and

$$L_{s;k}(y_{s;k}) = \max_{y\in[y_{s;k-1},y_{s;k}]} L_{s;k}(y) = 1.$$

Hence,

$$|L_{s;k}(y)| \le 1, \quad \forall y \in [y_{s;k-1}, y_{s;k+1}]. \tag{A.17}$$

Let us estimate $|L_{s;k}(y)|$ for $y \in \mathbb{R} \setminus (y_{s;k-1}, y_{s;k+1})$. From the definition one can see that

$$L_{s;k}(y) := \prod_{k'\in\pi_s\ k'\neq k} \frac{y - y_{s;k'}}{y_{s;k} - y_{s;k'}} = A_{s;k}(y - y_{s;k})^{-1} H_{s+1}(y), \tag{A.18}$$

where

$$A_{s;k} := \big((s+1)!\big)^{1/2} \prod_{k'\in\pi_s\ k'\neq k} (y_{s;k} - y_{s;k'})^{-1}. \tag{A.19}$$

From the inequalities [54, (6.31.22)]

$$\frac{\pi\sqrt{2}}{\sqrt{2s+3}} \le d_s \le \frac{\sqrt{10.5}}{\sqrt{2s+3}} \tag{A.20}$$

for the minimal distance d_s between consecusive $y_{s;k}$, $k \in \pi_s$, we have that

$$|y - y_{s;k}|^{-1} \le d_s^{-1} \le \frac{\sqrt{2s+3}}{\sqrt{10.5}} < \sqrt{s}, \quad \forall y \in \mathbb{R} \setminus (y_{s;k-1}, y_{s;k+1}),$$

and for any $s \in \mathbb{N}$ and $k, k' \in \pi_s$ with $k' \neq k$,

$$|y_{s;k} - y_{s;k'}|^{-1} \le C \frac{\sqrt{s}}{|k - k'|}, \tag{A.21}$$

which yield for any $y \in \mathbb{R} \setminus (y_{s;k-1}, y_{s;k+1})$,

$$
\begin{aligned}
|y - y_{s;k}|^{-1}|A_{s;k}| &\le \sqrt{s}\big((s+1)!\big)^{1/2} \prod_{k' \in \pi_s \; k' \ne k} |y_{s;k} - y_{s;k'}|^{-1} \le \sqrt{s} C^s \frac{\big((s+1)!\big)^{1/2} s^{s/2}}{k!(s-k)!} \\
&\le \sqrt{s} C^s \binom{s}{k} \frac{\big((s+1)!\big)^{1/2} s^{s/2}}{s!} \le \sqrt{s}(2C)^s \frac{\big((s+1)!\big)^{1/2} s^{s/2}}{s!} \le e^{K_1 s}. \qquad \text{(A.22)}
\end{aligned}
$$

In the last step we used the Stirling's approximation for factorial. Thus, we have proven that

$$
|L_{s;k}(y)| \le e^{K_1 s} |H_{s+1}(y)|, \quad \forall y \in \mathbb{R} \setminus (y_{s;k-1}, y_{s;k+1}). \qquad \text{(A.23)}
$$

With $I_{s;k} := [y_{s;k-1}, y_{s;k+1}]$, from the last estimate and (A.17) we prove (A.15):

$$
\begin{aligned}
\|L_{s;k}\|^2_{L_2(\mathbb{R},\gamma)} &= \|L_{s;k}\|^2_{L_2(I_{s;k},\gamma)} + \|L_{s;k}\|^2_{L_2(\mathbb{R}\setminus I_{s;k},\gamma)} \\
&\le 1 + e^{2K_1 s} \, \|H_s\|^2_{L_2(\mathbb{R},\gamma)} = 1 + e^{2K_1 s} \le e^{2Ks}.
\end{aligned}
$$

The inequality (A.16) can be proven similarly by using (A.9). □

Lemma A.5 *Assume that p and q are polynomials on $\mathbb{R}$ in the form*

$$
p(y) := \sum_{k=0}^{m} a_k y^k, \quad q(y) := \sum_{k=0}^{m-1} b_k y^k, \qquad \text{(A.24)}
$$

and that $p(y) = (y - y_0)q(y)$ for a point $y_0 \in \mathbb{R}$. Then we have

$$
|b_k| \le \sum_{k=0}^{m} |a_k|, \quad k = 0, ..., m-1. \qquad \text{(A.25)}
$$

Proof. From the definition we have

$$
\sum_{k=0}^{m} a_k y^k = -b_0 y_0 + \sum_{k=0}^{m-1} (b_{k-1} - b_k y_0) y^k + b_{m-1} y^m. \qquad \text{(A.26)}
$$

Hence we obtain

$$
0 = a_0 + b_0 y_0; \quad b_k = a_{k+1} + b_{k+1} y_0, \ k = 1, ..., m-2; \quad b_{m-1} = a_m. \qquad \text{(A.27)}
$$

From the last equalities one can see that the lemma is trivial if $y_0 = 0$. Consider the case $y_0 \ne 0$. If $|y_0| \le 1$, from (A.27) we deduce that

$$
b_k = \sum_{j=k+1}^{m} a_j y_0^{j-k-1}. \qquad \text{(A.28)}
$$

and, consequently,

$$|b_k| \le \sum_{j=k+1}^{m} |a_j||y_0|^{j-k-1} \le \sum_{j=0}^{m} |a_j|. \tag{A.29}$$

If $|y_0| > 1$, from (A.27) we deduce that

$$b_k = -\sum_{j=0}^{k} a_j y_0^{-(k+1-j)}, \tag{A.30}$$

and, consequently,

$$|b_k| \le \sum_{j=0}^{k} |a_j||y_0|^{-(k+1-j)} \le \sum_{j=0}^{m} |a_j|. \tag{A.31}$$

□

Lemma A.6 *Let $b_\ell^{s;k}$ be the polynomial coefficients of $L_{s;k}$ as in the representation* (3.30)*. Then we have for any $s \in \mathbb{N}_0$ and $k \in \pi_s$,*

$$\sum_{\ell=0}^{s} |b_\ell^{s;k}| \le e^{Ks} s!\,, \tag{A.32}$$

where the constant K are independent of s and $k \in \pi_s$.

Proof. For $s \in \mathbb{N}_0$, we represent the univariate Hermite polynomial H_s in the form

$$H_s(y) := \sum_{\ell=0}^{s} a_{s,\ell} y^\ell. \tag{A.33}$$

By using the well-known equality

$$H_s(y) = s! \sum_{\ell=0}^{\lfloor \frac{s}{2} \rfloor} \frac{(-1)^\ell}{\ell!(s-2\ell)!} \frac{y^{s-2\ell}}{2^\ell}, \tag{A.34}$$

one can derive that

$$\sum_{\ell=0}^{s} |a_{s,\ell}| \le s!. \tag{A.35}$$

From (A.18) we have

$$A_{s;k} H_{s+1}(y) = (y - y_{s;k}) L_{s;k}(y), \tag{A.36}$$

where $A_{s;k}$ is given as in (A.19). By Lemma A.5, (A.35) and (A.22), we obtain

$$\sum_{\ell=0}^{s} |b_\ell^{s;k}| \le \sum_{\ell=0}^{s} A_{s;k} \sum_{\ell'=0}^{s+1} |a_{s+1,\ell'}| \le e^{Ks} s!\,. \tag{A.37}$$

□

Lemma A.7 *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 $j = 1, \ldots, m$. Then there holds*

$$\|\varphi\|_{L_2(\mathbb{R}^m \setminus B_\omega^m, \gamma)} \leq Cm \exp(-K\omega) \|\varphi\|_{L_2(\mathbb{R}^m, \gamma)}, \tag{A.38}$$

and

$$\|\varphi\|_{L_\infty^{\sqrt{g}}(\mathbb{R}^m \setminus B_\omega^m)} \leq Cm \exp(-K\omega) \|\varphi\|_{L_\infty^{\sqrt{g}}(\mathbb{R}^m)}, \tag{A.39}$$

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

Proof. The inequality (A.38) was proven in [17, Lemma 3.3]. The inequality (A.39) can be proven in a similar way with a slight modification. ☐

A.2 Proof of Theorem 3.1

Proof. This theorem was proven in [15, Corollary 3.1] for the case $U = \mathbb{R}^\infty$. Let us prove it for the case $U = \mathbb{R}^M$. By Lemma 3.1 the series (3.2) converges unconditionally in the space $L_2(\mathbb{R}^M, X, \gamma)$ to v. Observe that $I_{\Lambda(\xi)} H_{\boldsymbol{s}} = H_{\boldsymbol{s}}$ for every $\boldsymbol{s} \in \Lambda(\xi)$ and $\Delta_{\boldsymbol{s}} H_{\boldsymbol{s}'} = 0$ for every $\boldsymbol{s} \not\leq \boldsymbol{s}'$. Hence for the downward closed set $\Lambda(\xi) \subset \mathbb{N}_0^M$, we can write

$$I_{\Lambda(\xi)} v \;=\; I_{\Lambda(\xi)}\Big(\sum_{\boldsymbol{s} \in \mathbb{N}_0^M} v_{\boldsymbol{s}}\, H_{\boldsymbol{s}}\Big) \;=\; \sum_{\boldsymbol{s} \in \mathbb{N}_0^M} v_{\boldsymbol{s}}\, I_{\Lambda(\xi)} H_{\boldsymbol{s}} \;=\; S_{\Lambda(\xi)} v \;+\; \sum_{\boldsymbol{s} \notin \Lambda(\xi)} v_{\boldsymbol{s}}\, I_{\Lambda(\xi) \cap R_{\boldsymbol{s}}}\, H_{\boldsymbol{s}},$$

where $R_{\boldsymbol{s}} := \{\boldsymbol{s}' \in \mathbb{N}_0^M : \boldsymbol{s}' \leq \boldsymbol{s}\}$. This implies

$$\left\|v - I_{\Lambda(\xi)} v\right\|_{L_\infty^{\sqrt{g}}(\mathbb{R}^M, X)} \;\leq\; \left\|v - S_{\Lambda(\xi)} v\right\|_{L_\infty^{\sqrt{g}}(\mathbb{R}^M, X)} + \sum_{\boldsymbol{s} \notin \Lambda(\xi)} \left\|I_{\Lambda(\xi) \cap R_{\boldsymbol{s}}}\, H_{\boldsymbol{s}}\right\|_{L_\infty^{\sqrt{g}}(\mathbb{R}^M)}. \tag{A.40}$$

Therefore, to prove the lemma it is sufficient to show that each term in the right-hand side is bounded by $C\xi^{-(1/q-1/2)}$. The bound of the first term can be obtained from the Cauchy–Schwasz inequality and (A.9):

$$\begin{aligned}
\|v - S_{\Lambda(\xi)}\|_{L_\infty^{\sqrt{g}}(\mathbb{R}^M, X)} &\leq \sum_{\sigma_{\boldsymbol{s}} > \xi^{1/q}} \|v_{\boldsymbol{s}}\|_X \, \|H_{\boldsymbol{s}}\|_{L_\infty^{\sqrt{g}}(\mathbb{R}^M)} \leq \sum_{\sigma_{\boldsymbol{s}} > \xi^{1/q}} \|v_{\boldsymbol{s}}\|_X \\
&\leq \left(\sum_{\sigma_{\boldsymbol{s}} > \xi^{1/q}} (\sigma_{\boldsymbol{s}} \|v_{\boldsymbol{s}}\|_X)^2\right)^{1/2} \left(\sum_{\sigma_{\boldsymbol{s}} > \xi^{1/q}} \sigma_{\boldsymbol{s}}^{-2}\right)^{1/2} \leq C \left(\sum_{\sigma_{\boldsymbol{s}} > \xi^{1/q}} \sigma_{\boldsymbol{s}}^{-q} \sigma_{\boldsymbol{s}}^{-(2-q)}\right)^{1/2} \\
&\leq C\xi^{-(1/q-1/2)} \left(\sum_{\boldsymbol{s} \in \mathbb{N}_0^M} \sigma_{\boldsymbol{s}}^{-q}\right)^{1/2} \leq C\xi^{-(1/q-1/2)}.
\end{aligned} \tag{A.41}$$

Let us prove the bound of the second term in the right-hand side of (A.40). We have that

$$\left\| I_{\Lambda(\xi)\cap R_{\boldsymbol{s}}} H_{\boldsymbol{s}} \right\|_{L_\infty^{\sqrt{g}}(\mathbb{R}^M)} \ \le \ \sum_{\boldsymbol{s}'\in\Lambda(\xi)\cap R_{\boldsymbol{s}}} \left\| \Delta_{\boldsymbol{s}'}(H_{\boldsymbol{s}}) \right\|_{L_\infty^{\sqrt{g}}(\mathbb{R}^M)}. \tag{A.42}$$

We estimate the norms inside the right-hand side. For $\boldsymbol{s} \in \mathbb{N}_0^M$ and $\boldsymbol{s}' \in \Lambda(\xi) \cap R_{\boldsymbol{s}}$, we have $\Delta_{\boldsymbol{s}'}(H_{\boldsymbol{s}}) = \prod_{j=1}^M \Delta_{s_j'}(H_{s_j})$. From Lemma 3.2 and (A.9) we deduce that

$$\|\Delta_{s_j'}(H_{s_j})\|_{L_\infty^{\sqrt{g}}(\mathbb{R})} \ \le \ (1 + C_\varepsilon s_j')^{1/6+\varepsilon} \, \|H_{s_j}\|_{L_\infty^{\sqrt{g}}(\mathbb{R})} \ \le \ (1 + C_\varepsilon s_j')^{1/6+\varepsilon},$$

and consequently,

$$\|\Delta_{\boldsymbol{s}'}(H_{\boldsymbol{s}})\|_{L_\infty^{\sqrt{g}}(\mathbb{R}^M)} \ = \ \prod_{j=1}^M \|\Delta_{s_j'}(H_{s_j})\|_{L_\infty^{\sqrt{g}}(\mathbb{R})} \ \le \ p_{\boldsymbol{s}'}(\theta_1, \lambda) \ \le \ p_{\boldsymbol{s}}(\theta_1, \lambda), \tag{A.43}$$

where $\theta_1 = 1/6 + \varepsilon$. Substituting $\|\Delta_{\boldsymbol{s}'}(H_{\boldsymbol{s}})\|_{L_\infty^{\sqrt{g}}(\mathbb{R}^M)}$ in (A.42) by the right-hand side of (A.43) gives that

$$\begin{aligned}
\left\| I_{\Lambda(\xi)\cap R_{\boldsymbol{s}}} H_{\boldsymbol{s}} \right\|_{L_\infty^{\sqrt{g}}(\mathbb{R}^M)} \ &\le \ \sum_{\boldsymbol{s}'\in\Lambda(\xi)\cap R_{\boldsymbol{s}}} p_{\boldsymbol{s}}(\theta_1, \lambda) \ \le \ |R_{\boldsymbol{s}}| \, p_{\boldsymbol{s}}(\theta_1, \lambda) \\
&\le \ p_{\boldsymbol{s}}(1,1) \, p_{\boldsymbol{s}}(\theta_1, \lambda) \ \le \ p_{\boldsymbol{s}}(\theta/2, \lambda).
\end{aligned}$$

From the last estimates and the assumptions we derive the bound of the second term in the right-hand side of (A.40):

$$\begin{aligned}
\sum_{\boldsymbol{s}\notin\Lambda(\xi)} \left\| I_{\Lambda(\xi)\cap R_{\boldsymbol{s}}} H_{\boldsymbol{s}} \right\|_{L_\infty^{\sqrt{g}}(\mathbb{R}^M)} \ &\le \ C \sum_{\boldsymbol{s}\notin\Lambda(\xi)} \|v_{\boldsymbol{s}}\|_X \, p_{\boldsymbol{s}}(\theta/2, \lambda) \\
&\le \ C \left(\sum_{\sigma_{\boldsymbol{s}} > \xi^{1/q}} (\sigma_{\boldsymbol{s}} \|v_{\boldsymbol{s}}\|_X)^2 \right)^{1/2} \left(\sum_{\sigma_{\boldsymbol{s}} > \xi^{1/q}} p_{\boldsymbol{s}}(\theta/2, \lambda)^2 \sigma_{\boldsymbol{s}}^{-2} \right)^{1/2} \\
&\le \ C \left(\sum_{\sigma_{\boldsymbol{s}} > \xi^{1/q}} p_{\boldsymbol{s}}(\theta/2, \lambda)^2 \sigma_{\boldsymbol{s}}^{-q} \sigma_{\boldsymbol{s}}^{-(2-q)} \right)^{1/2} \\
&\le \ C \xi^{-(1/q-1/2)} \left(\sum_{\boldsymbol{s}\in\mathbb{N}_0^M} p_{\boldsymbol{s}}(\theta, \lambda) \sigma_{\boldsymbol{s}}^{-q} \right)^{1/2} \ \le C \xi^{-(1/q-1/2)},
\end{aligned}$$

which together with (A.40) and (A.41) proves the theorem. ∎

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, 1989.

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

[15] D. Dũng. Sparse-grid polynomial interpolation approximation and integration for parametric and stochastic elliptic PDEs with lognormal inputs. *ESAIM Math. Model. Numer. Anal.*, 55:1163–1198, 2021.

[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 D. T. Pham. Deep ReLU neural network approximation of parametric and stochastic elliptic PDEs with lognormal inputs. *arXiv: 2111.05854*, 2021.

[18] D. Dũng, V. K. Nguyen, C. Schwab, and J. Zech. *Analyticity and sparsity in uncertainty quantification for PDEs with Gaussian random field inputs.* 2021.

[19] D. Dũng, V. K. Nguyen, and M. X. Thao. Computation complexity of deep ReLU neural networks in high-dimensional approximation. *J. Comp. Sci. and Cyber.*, 37:292–320, 2021.

[20] I. Daubechies, R. DeVore, S. Foucart, B. Hanin, and G. Petrova. Nonlinear approximation and (Deep) ReLU networks. *Constr. Approx.*, 2021.

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

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

[23] D. Elbrächter, P. Grohs, A. Jentzen, and C. Schwab. DNN expression rate analysis of high-dimensional PDEs: application to option pricing. Technical Report 2018-33, Seminar for Applied Mathematics, ETH Zürich, 2018.

[24] 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.

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

[26] 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.

[27] L. Gonon and C. Schwab. Deep ReLU network expression rates for option prices in high-dimensional, exponential Lévy models. Technical Report 2020-52 (revised), Seminar for Applied Mathematics, ETH Zürich, 2020.

[28] 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.

[29] 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.

[30] P. Grohs and L. Herrmann. Deep neural network approximation for high-dimensional elliptic PDEs with boundary conditions. *IMA J. Numer. Anal.*, 2021.

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

[32] 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.

[33] 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.

[34] L. Herrmann, C. Schwab, and J. Zech. Deep neural network expression of posterior expectations in bayesian pde inversion. *Inverse Problems*, 36, 2020.

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

[36] V. Hoang and C. Schwab. N-term Galerkin Wiener chaos approximation rates for elliptic PDEs with lognormal Gaussian random inputs. *M3AS*, 24:797 – 826, 2014.

[37] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Netw.*, 2, 1989.

[38] G. Kutyniok, P. C. Petersen, M. Raslan, and R. Schneider. A theoretical analysis of deep neural networks and parametric PDEs. *Constr. Approx.*, 2021.

[39] J. Lu, Z. Shen, H. Yang, and S. Zhang. Deep network approximation for smooth functions. *SIAM Journal on Mathematical Analysis*, 53(5):5465–5506, 2021.

[40] D. M. Matjila. Bounds for the weighted Lebesgue functions for Freud weights. *J. Approx. Theory*, 79:385–406, 1994.

[41] D. M. Matjila. Convergence of Lagrange interpolation for Freud weights in weighted L_p, $0 < p \leq 1$. *in: Nonlinear numerical methods and rational approximation*, pages 25–35, Kluwer, Dordrecht 1994.

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

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

[44] 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.

[45] 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.

[46] J. A. A. Opschoor, C. Schwab, and J. Zech. Exponential ReLU DNN expression of holomorphic maps in high dimension. *Constr. Approx.*, 2021.

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

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

[49] 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.

[50] Z. Shen, H. Yang, and S. Zhang. Deep network approximation characterized by a number of neurons. *Communications in Computational Physics*, 28:1768–1811, 2020.

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

[52] 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.

[53] J. Szabados. Weighted Lagrange and Hermite-Fejér interpolation on the real line. *J. Inequal. and Applns.*, 1:99–123, 1997.

[54] G. Szegö. *Orthogonal Polynomials.* Amer. Math. Soc. Colloq. Publ., 1939.

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

[56] 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.

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

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

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

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

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