# Sparse-grid sampling recovery and deep ReLU neural networks in high-dimensional approximation

Dinh Dũng[a] and Van Kien Nguyen[b]

[a]Vietnam National University, Hanoi, Information Technology Institute
144 Xuan Thuy, Cau Giay, Hanoi, Vietnam
Email: dinhzung@gmail.com
[b]Department of Mathematics, University of Transport and Communications
No.3 Cau Giay Street, Lang Thuong Ward, Dong Da District, Hanoi, Vietnam
Email: kiennv@utc.edu.vn

July 17, 2020

**Abstract**

We investigate approximations of functions from the Hölder-Zygmund space of mixed smoothness $H^\alpha_\infty(\mathbb{I}^d)$ defined on the $d$-dimensional unit cube $\mathbb{I}^d := [0,1]^d$, by linear algorithms of sparse-grid sampling recovery and by deep ReLU (Rectified Linear Unit) neural networks when the dimension $d$ may be very large. The approximation error is measured in the norm of the isotropic Sobolev space $W^{1,p}_0(\mathbb{I}^d)$. The optimality of this sampling recovery is studied in terms of sampling $n$-widths. Optimal linear sampling algorithms are constructed on sparse grids using the piece-wise linear B-spline interpolation representation. We prove some tight dimension-dependent bounds of the sampling $n$-widths explicit in $d$ and $n$. Based on the results on sampling recovery, we investigate the expressive power of deep ReLU neural networks to approximate functions in Hölder-Zygmund space. Namely, for any function $f \in H^\alpha_\infty(\mathbb{I}^d)$, we explicitly construct a deep ReLU neural network having an output that approximates $f$ in the $W^{1,p}_0(\mathbb{I}^d)$-norm with a prescribed accuracy $\varepsilon$, and prove tight dimension-dependent bounds of the computation complexity of this approximation, characterized as the number of weights and the depth of this deep ReLU neural network, explicitly in $d$ and $\varepsilon$. Moreover, we show that under a certain restriction the curse of dimensionality can be avoided in the approximations by sparse-grid sampling recovery and deep ReLU neural networks.

**Keywords and Phrases:** High-dimensional approximation; Hölder-Zygmund spaces of mixed smoothness; Sampling recovery; Sparse grids; Deep ReLU neural networks.

**Mathematics Subject Classifications (2000):** 41A25; 41A30; 68T99; 82C32; 92B20.

## 1 Introduction

The purpose of the present paper is to investigate sampling recovery of functions on the $d$-dimensional unit cube $\mathbb{I}^d := [0,1]^d$ with Hölder-Zygmund mixed smoothness by truncated on sparse grids tensor product Faber series when the dimension $d$ may be very large, and then apply this sparse-grid sampling recovery to the high-dimensional approximation by deep ReLU (Rectified Linear Unit) neural networks.

The approximation error is measured in the norm of the isotropic Sobolev space $W_0^{1,p}$. We also study the optimality for these approximations.

In recent decades, the high-dimensional approximation by sparse-grid sampling recovery of functions or signals has been of great interest since they can be applied in a striking number of fields such as Information Technology, Mathematical Finance, Chemistry, Quantum Mechanics, Meteorology, and, in particular, in Uncertainty Quantification and Deep Machine Learning. A numerical method for such problems may require a computational cost increasing exponentially in dimension when the accuracy increases. This phenomenon is called the curse of dimensionality, coined by Bellman [3]. Hence for an efficient computation in high dimensional approximation, one of the key prerequisites is that the curse of dimension can be avoided at least to some extent. In some cases this can be obtained, particularly when the functions to be approximated have appropriate mixed smoothness. With this restriction one can apply approximation methods and sampling algorithms constructed on hyperbolic crosses and sparse grids which give a surprising effect since hyperbolic crosses and sparse grids have the number of elements much less than those of standard domains and grids but give the same approximation error. This essentially reduces the computational cost, and therefore makes the problem tractable.

Sparse grids for approximate sampling recovery and integration were first considered by Smolyak [35]. In computational mathematics, the sparse grid approach was initiated by Zenger [45]. There is a large number of papers on sparse-grid sampling recovery and numerical applications. The reader can consult [4, 10] for surveys about results and bibliography. We also refer to the monographs [27, 28] for concepts and results on high dimensional problems and computation complexity.

Let us mention that high-dimensional functions having mixed smoothness play a fundamental role not only in approximation theory but also in mathematical physics and finance and other fields. For instance, in a recent work on regularity properties of solutions of the electronic Schrödinger equation, Yserentant [44] has shown that the eigenfunctions of the electronic Schrödinger operator have a certain mixed smoothness. Triebel [40, Chapter 6] has indicated a relation between Faber bases and sampling recovery in the context of spaces with mixed smoothness and solutions of Navier-Stokes equations. In particular, when initial data belonging to spaces with mixed smoothness, Navier-Stokes equations admit a unique solution. In mathematical finance, many problems are expressed as the expectation of some payoff function depending on quantities, such as stock prices, which are solutions of stochastic equations governed by Brownian motions. The payoff function normally has kinks and jumps and belongs to a very high dimensional space. To approximate the expected value one can apply preliminary integration method with respect to a single well chosen variable to obtain a function of $d-1$ variables which belongs to appropriate mixed Sobolev spaces in which Quasi-Monte Carlo can be applied efficiently, see [16] and references therein. For a survey on various aspects of high-dimensional approximation of functions having a mixed smoothness we refer the reader to the book [10].

The object of our interest in high-dimensional sampling recovery are functions on $\mathbb{I}^d$ having Hölder-Zygmund mixed smoothness $\alpha > 0$ when the dimension $d$ may be very large. Let us first introduce the space $H_\infty^\alpha(\mathbb{I}^d)$ of all such functions. For a univariate function $f$ on $\mathbb{I}$, the $r$th difference operator $\Delta_h^r$ is defined by

$$\Delta_h^r(f,x) := \sum_{j=0}^{r}(-1)^{r-j}\binom{r}{j}f(x+jh),$$

for all $x$ and $h \geq 0$ such that $x, x+rh \in \mathbb{I}$. If $u$ is any subset of $[d] := \{1,\dots,d\}$, for a multivariate function $f$ on $\mathbb{I}^d$ the mixed $(r,u)$th difference operator $\Delta_{\boldsymbol{h}}^{r,u}$ is defined by

$$\Delta_{\boldsymbol{h}}^{r,u} := \prod_{i\in u}\Delta_{h_i}^r, \quad \Delta_{\boldsymbol{h}}^{r,\varnothing} = \mathrm{Id},$$

where the univariate operator $\Delta_{h_i}^r$ is applied to the univariate function $f$ by considering $f$ as a function of variable $x_i$ with the other variables held fixed, and Id is the identity operator. If $0 < \alpha \le r$, the Hölder-Zygmund space $H_\infty^\alpha(\mathbb{I}^d)$ of mixed smoothness $\alpha$ is defined as the set of functions $f \in C(\mathbb{I}^d)$ for which the norm

$$\|f\|_{H_\infty^\alpha(\mathbb{I}^d)} := \max_{u\subset[d]} \left\{ \sup_{\boldsymbol{h}} \prod_{i\in u} h_i^{-\alpha} \|\Delta_{\boldsymbol{h}}^{r,u}(f)\|_{C(\mathbb{I}^d(r\boldsymbol{h},u))} \right\} \tag{1.1}$$

is finite, where $\mathbb{I}^d(r\boldsymbol{h},u) := \{\boldsymbol{x} \in \mathbb{I}^d : x_i + rh_i \in \mathbb{I},\ i \in u\}$. Note, that when $u = \varnothing$ the term in brackets of (1.1) is $\|f\|_{C(\mathbb{I}^d)}$. By the definition we have the inclusions $H_\infty^\alpha(\mathbb{I}^d) \subset C(\mathbb{I}^d) \subset L_p(\mathbb{I}^d)$ for $0 < p \le \infty$. Moreover, it is well-known that for $\alpha = r$, the space $H_\infty^r(\mathbb{I})$ coincides with the Sobolev space $W_\infty^r(\mathbb{I})$. Hence, by a tensor product argument one can deduce the equality $H_\infty^\alpha(\mathbb{I}^d) = W_\infty^r(\mathbb{I}^d)$ where $W_\infty^r(\mathbb{I}^d)$ is the Sobolev space of functions having bounded mixed derivatives of order $r$.

Denote by $\mathring{U}_\infty^\alpha$ the set of all functions $f$ in the unit ball of $H_\infty^\alpha(\mathbb{I}^d)$ such that $\|f\|_{H_\infty^\alpha(\mathbb{I}^d)} \le 1$ and $f$ vanishes on the boundary $\partial\mathbb{I}^d$ of $\mathbb{I}^d$, i.e., $f(\boldsymbol{x}) = 0$ if $x_j = 0$ or $x_j = 1$ for some index $j \in [d]$. Let $X_n = \{\boldsymbol{x}^j\}_{j=1}^n$ be a set of $n$ sample points in $\mathbb{I}^d$ and $\Phi_n = \{\phi_j\}_{j=1}^n$ a family of functions on $\mathbb{I}^d$. We define the linear sampling algorithm $L(\Phi_n, X_n, \cdot)$ for an approximate recovery of functions $f \in \mathring{U}_\infty^\alpha$ from the sampling values $f(\boldsymbol{x}^1), \dots, f(\boldsymbol{x}^n)$ by

$$L(\Phi_n, X_n, f) := \sum_{j=1}^n f(\boldsymbol{x}^j)\phi_j. \tag{1.2}$$

The error of approximation will be measured in the isotropic Sobolev space $W_0^{1,p} := W_0^{1,p}(\mathbb{I}^d)$, $1 \le p \le \infty$, consisting of all functions $f \in L_p(\mathbb{I}^d)$ vanishing on the boundary $\partial\mathbb{I}^d$ of $\mathbb{I}^d$ in the sense of trace such that the "energy" norm

$$\|f\|_{W_0^{1,p}} := \begin{cases} \left( \sum_{i=1}^d \int_{\mathbb{I}^d} \left| \frac{\partial}{\partial x_i} f(\boldsymbol{x}) \right|^p \mathrm{d}\boldsymbol{x} \right)^{1/p}, & 1 \le p < \infty, \\ \max\limits_{1\le i\le d} \operatorname{ess\,sup}\limits_{\boldsymbol{x}\in\mathbb{I}^d} \left| \frac{\partial}{\partial x_i} f(\boldsymbol{x}) \right|, & p = \infty, \end{cases}$$

is finite (this is a norm due to Poincaré inequality). It is known that the norm of spaces $W_0^{1,p}$, especially, the energy norm of the space $H_0^1 := W_0^{1,2}$ is of great interest in approximation and numerical methods of PDEs. To characterizes the error of best recovery algorithm from $n$ sampling values of $f \in \mathring{U}_\infty^\alpha$, we consider the quantity

$$r_n := r_n(\mathring{U}_\infty^\alpha, W_0^{1,p}) := \inf_{X_n,\Phi_n} \sup_{f\in\mathring{U}_\infty^\alpha} \|f - L(\Phi_n, X_n, f)\|_{W_0^{1,p}},$$

where the infimum is taken over all sets of $n$ sample points $X_n = \{\boldsymbol{x}^j\}_{j=1}^n$ and all the family $\Phi_n = \{\phi_j\}_{j=1}^n \subset W_0^{1,p}$. This quantity is called sampling $n$-width. For its properties and relations to other approximation quantities we refer the reader to [27, 10].

Our present investigation on sampling recovery can be considered as a continuation of the recent study in [11] where the explicit dimension-dependent estimates of the approximation error for linear algorithms of sampling recovery from sampling values on Smolyak sparse grids by using piecewise linear B-splines are obtained. In this paper, however, the optimal linear sampling algorithms is constructed on energy-based sparse grids which leads to the tight dimension-dependent approximation rate of optimal recovery for the class $\mathring{U}_\infty^\alpha$ with $1 < \alpha \le 2$

$$C_1(p)B_*^{-d}n^{-(\alpha-1)} \ \le\ r_n \ \le\ C_2(\alpha,p)B^{*-d}n^{-(\alpha-1)},$$

where $B_* = B_*(p) > 1$ and $B^* = B^*(d, \alpha) > 0$. Moreover, we show that under a very light additional restriction the constant $B^* > 1$ whenever $d > d_0(\alpha, p)$. We can also explicitly construct an optimal linear sampling operator on a sparse grid of the form (1.2) by using truncated tensor product Faber series. This is one of the main result of our paper.

It has been shown that there is a close relation between the approximation by sampling recovery based on B-spline interpolation and quasi-interpolation representation, and the approximation by deep ReLU neural networks and[42, 43], [25], [12], [36], [34]. Neural networks have been studied and used for almost 70 years, dating back to the foundational work of Hebb [19] and of Rosenblatt [33]. They are composed of layers of computational nodes interconnected by activation functions. The theory of approximating functions using shallow networks goes back to [6, 20] where they showed that that any continuous functions can be approximated by shallow networks which have one hidden layer. These approximation results, however, do not provide any information on the required neurons of a network to achieve a given accuracy. In [24, 32] it has shown that a shallow network with $\mathcal{O}(\varepsilon^{-d/s})$ neurons can approximate functions of $d$-variables in differential space $C^s$ with prescribed error $\varepsilon$. Hence, standard convergence results for shallow networks suffer from the curse of dimensionality when the dimension $d$ may be very large.

Neural networks used recently in machine learning are distinguished from those popular in the 1980's and 1990's by emphasis on the depth of networks. In applications, it is observed that deep networks with many hidden layers appear to perform better than shallow ones of comparable size. In recent years, deep neural networks have been successfully applied to a striking variety of Machine Learning problems, including computer vision [21], natural language processing [41], speech recognition and image classification [22]. The main advantage of deep neural networks is that they can output compositions of functions cheaply. Since their application range is getting wider, theoretical analysis to reveal the reason why deep neural networks could lead to significant practical improvements attracts substantial attention [2, 23, 26, 37, 38]. In the last several years, there has been a number of interesting papers that address the role of depth and architecture of deep neural networks in approximating sets of functions which have a very special regularity properties such as analytic functions [14, 24], differentiable functions [31, 42], oscillatory functions [17], and functions in isotropic Sobolev spaces [18]. Most of them use deep ReLU neural networks for approximation since the rectified linear unit is a simple and preferable activation function in many applications. The output of such a network is a continuous piece-wise linear function which is easily and cheaply computed. In the context of function spaces with mixed smoothness, using decomposition by Faber series, it has been proven in [25] that for functions in Koborov space the number of parameters in a deep ReLU neural network needed to achieve an error tolerance of $\varepsilon$ is $\mathcal{O}(\varepsilon^{-1/2}\log(\varepsilon^{-1})^d)$. In another study [36], the author has employed results in [7, 8] of the first author of the present paper on B-spline interpolation or quasi-interpolation representation to approximate functions in Besov spaces with mixed smoothness by deep ReLU neural networks. However, these results did not give explicit dimension-dependent estimates for number of weights of the neural network needed.

From our results on sparse-grid sampling recovery we deduce approximation by deep ReLU neural networks for $d$-variate functions in Hölder-Zygmund classes of mixed smoothness. Namely, we investigate the high-dimensional approximation by deep ReLU neural networks of functions from the classes $\mathring{U}^{\alpha}_{\infty}$. We focus our attention on estimation of the computation complexity of a deep ReLU neural network characterized by the number of its weights and its depth required to achieve a given accuracy for the approximation (cf. [1, 12, 42]), emphasizing the dimension dependence of the computation complexity of the network.

Let us briefly describe our main results on the approximation by deep ReLU neural networks. For

any $f \in \mathring{U}^{\alpha}_{\infty}$ we explicitly construct a deep ReLU neural networks $\Phi_f$ having the output $\mathcal{N}(\Phi_f, \cdot)$ that approximates $f$ in the $W^{1,p}_0$-norm with a prescribed accuracy $\varepsilon$ and having computation complexity expressing the dimension-dependent number of weights

$$W(\Phi_f) \le C_3(\alpha, p) B^{-d} (\varepsilon^{-1})^{\frac{1}{\alpha-1}} \log(\varepsilon^{-1}),$$

and the dimension-dependent depth

$$L(\Phi_f) \le C_4(\alpha, p) \log d \log(\varepsilon^{-1}),$$

where $B > 0$. If a further restriction is given, in particular when $1 \le p \le 2$, we show that $B > 1$ when $d > d_0(\alpha, p)$ and therefore overcoming the curse of dimensionality. We also prove that these bounds cannot improved up to logarithm terms. Moreover, the continuous piece-wise linear function $\mathcal{N}(\Phi_f, \cdot)$ can be designed also as an output $\mathcal{N}(\Phi^*_f, \cdot)$ of another "very" deep ReLU neural network $\Phi^*_f$ having computation complexity expressing the $\varepsilon$-independent width $N_w(\Phi^*_f) \le C_5 d$ and the dimension-dependent depth

$$L(\Phi^*_f) \le C_6(\alpha, p) B^{-d} (\varepsilon^{-1})^{\frac{1}{\alpha-1}} \log(\varepsilon^{-1}).$$

The outline of this paper is as follows. Section 2 is devoted to recalling decomposition of continuous functions on the unit cube $\mathbb{I}^d$ by Faber system. There an estimate for its coefficients in the Sobolev spaces is given. Our results on sampling recovery for functions in Hölder-Zygmund classes are formulated in Section 3. In Section 4 we construct a deep ReLU neural network that approximates functions in $\mathring{U}^{\alpha}_{\infty}$ and prove upper and lower estimates for number of weights and the depth required. We conclude our main results in Section 5.

**Notation.** As usual, $\mathbb{N}$ denotes the natural numbers, $\mathbb{Z}$ denotes the integers, $\mathbb{R}$ the real numbers and $\mathbb{N}_0 := \{s \in \mathbb{Z} : s \ge 0\}$; $\mathbb{N}_{-1} = \mathbb{N}_0 \cup \{-1\}$. The letter $d$ is always reserved for the underlying dimension of $\mathbb{R}^d$, $\mathbb{N}^d$, etc., and $[d]$ denotes the set of all natural numbers from 1 to $d$. Vectorial quantities are denoted by boldface letters and $x_i$ denotes the $i$th coordinate of $\boldsymbol{x} \in \mathbb{R}^d$, i.e., $\boldsymbol{x} := (x_1, \dots, x_d)$. We use the notation $\boldsymbol{xy}$ for the usual Euclidean inner product in $\mathbb{R}^d$ and $2^{\boldsymbol{x}} := (2^{x_1}, \dots, 2^{x_d})$. For $\boldsymbol{k}, \boldsymbol{s} \in \mathbb{N}^d_0$, we denote $2^{-\boldsymbol{k}}\boldsymbol{s} := (2^{-k_1}s_1, \dots, 2^{-k_d}s_d)$. For $\boldsymbol{x} \in \mathbb{R}^d$ we write $|\boldsymbol{x}|_0 = |\{x_j \ne 0,\ j = 1, \dots, d\}|$ and if $0 < p \le \infty$ we denote $|\boldsymbol{x}|_p := \left(\sum_{i=1}^d |x_i|^p\right)^{1/p}$ with the usual modification when $p = \infty$. The notations $|\cdot|_0$ and $|\cdot|_p$ are extended to matrices $\boldsymbol{W} \in \mathbb{R}^{m \times n}$. For the function $f(p)$ of variable $p$, $f(\infty)$ is understood as $f(\infty) = \lim_{p \to \infty} f(p)$ when the limit exists.

## 2 Faber series

In this section we recall a decomposition of continuous functions on the unit cube $\mathbb{I}^d$ into tensor product Faber series and give an estimate in the $W^{1,p}_0$-norm of the components of functions from the Hölder-Zygmund space of mixed smoothness $H^{\alpha}_{\infty}(\mathbb{I}^d)$. This decomposition plays a fundamental role in construction of linear algorithms of sparse-grid sampling recovery and of deep neural networks for approximation in the $W^{1,p}_0$-norm of functions from the Hölder-Zygmund space of mixed smoothness $H^{\alpha}_{\infty}(\mathbb{I}^d)$.

We start by introducing the tensorized Faber basis. Let $\varphi(x) = (1 - |x - 1|)_+$, $x \in \mathbb{R}$, be the hat function (the piece-wise linear B-spline with knots at $0, 1, 2$), where $x_+ := \max(x, 0)$ for $x \in \mathbb{R}$. For $k \in \mathbb{N}_{-1}$ we define the functions $\varphi_{k,s}$ on $\mathbb{I}$ by

$$\varphi_{k,s}(x) := \varphi(2^{k+1}x - 2s),\ x \in \mathbb{I}, \quad k \ge 0,\ s \in Z(k) := \{0, 1, \dots, 2^k - 1\}, \tag{2.1}$$

and

$$\varphi_{-1,s}(x) := \varphi(x-s+1),\ x \in \mathbb{I}, \quad s \in Z(-1) := \{0,1\}. \tag{2.2}$$

Put $Z^d(\boldsymbol{k}) := \times_{i=1}^d Z(k_i)$. For $\boldsymbol{k} \in \mathbb{N}_{-1}^d$, $\boldsymbol{s} \in Z^d(\boldsymbol{k})$, define the $d$-variate tensor product hat functions

$$\varphi_{\boldsymbol{k},\boldsymbol{s}}(\boldsymbol{x}) \ := \ \prod_{i=1}^d \varphi_{k_i,s_i}(x_i), \quad \boldsymbol{x} \in \mathbb{I}^d. \tag{2.3}$$

For a univariate function $f$ on $\mathbb{I}$, $k \in \mathbb{N}_{-1}$, and $s \in Z(k)$ we define

$$\lambda_{k,s}(f) \ := -\frac{1}{2}\Delta^2_{2^{-k-1}} f\big(2^{-k}s\big),\ k \geq 0, \quad \lambda_{-1,s}(f) \ := f(s).$$

We also define the linear functionals $\lambda_{\boldsymbol{k},\boldsymbol{s}}$ for multivariate function $f$ on $\mathbb{I}^d$, $\boldsymbol{k} \in \mathbb{N}_{-1}^d$, and $\boldsymbol{s} \in Z^d(\boldsymbol{k})$ by

$$\lambda_{\boldsymbol{k},\boldsymbol{s}}(f) \ := \prod_{i=1}^d \lambda_{k_i,s_i}(f),$$

where the univariate functional $\lambda_{k_i,s_i}$ is applied to the univariate function $f$ by considering $f$ as a function of variable $x_i$ with the other variables held fixed.

We have the following decomposition.

**Lemma 2.1** *The Faber system* $\big\{\varphi_{\boldsymbol{k},\boldsymbol{s}} : \boldsymbol{k} \in \mathbb{N}_{-1}^d, \boldsymbol{s} \in Z^d(\boldsymbol{k})\big\}$ *is a basis in* $C(\mathbb{I}^d)$. *Moreover, any function* $f \in C(\mathbb{I}^d)$ *can be represented by the Faber series*

$$f \ = \ \sum_{\boldsymbol{k} \in \mathbb{N}_{-1}^d} q_{\boldsymbol{k}}(f) := \ \sum_{\boldsymbol{k} \in \mathbb{N}_{-1}^d} \sum_{\boldsymbol{s} \in Z^d(\boldsymbol{k})} \lambda_{\boldsymbol{k},\boldsymbol{s}}(f)\varphi_{\boldsymbol{k},\boldsymbol{s}}, \tag{2.4}$$

*converging in the norm of* $C(\mathbb{I}^d)$.

The system (2.1)-(2.2) and above decomposition for continuous functions on the interval $\mathbb{I}$ goes back to Faber [15]. The extension for tensorized Faber bases in higher dimensions was obtained in [39, Theorem 3.10] for $d = 2$ and in [7, 9] for the case $d \geq 2$. In [39, 7], the authors also established the decomposition (2.4) with equivalent discrete quasi-norm for function spaces of mixed smoothness. A generalization to B-spline interpolation and quasi-interpolation representation was obtained by the first author of this paper in [7, 9]. Note that the sequence in (2.4) converges conditionally in $C(\mathbb{I}^d)$ if $f \in C(\mathbb{I}^d)$, see comment in [39, Section 3.3.2]. Moreover, when $f \in \mathring{U}^\alpha_\infty$ we can write

$$f \ = \ \sum_{\boldsymbol{k} \in \mathbb{N}_0^d} q_{\boldsymbol{k}}(f)$$

with unconditional convergence in $C(\mathbb{I}^d)$, see [39, Theorem 3.13]. In this case we have

$$\lambda_{\boldsymbol{k},\boldsymbol{s}}(f) \ := \prod_{i=1}^d \Big( -\frac{1}{2}\Delta^2_{2^{-k_i-1}} f\big(2^{-\boldsymbol{k}}\boldsymbol{s}\big)\Big)$$

and the following estimate holds

$$|\lambda_{\boldsymbol{k},\boldsymbol{s}}(f)| \leq 2^{-(\alpha+1)d} 2^{-\alpha|\boldsymbol{k}|_1}, \quad \boldsymbol{k} \in \mathbb{N}_0^d,\ \boldsymbol{s} \in Z^d(\boldsymbol{k}). \tag{2.5}$$

We now estimate the $W_0^{1,p}$-norm of $q_{\boldsymbol{k}}(f)$.

**Lemma 2.2** *Let $d \in \mathbb{N}$, $1 \le \alpha \le 2$, and $1 \le p \le \infty$. Then for a function $f \in \mathring{U}^\alpha_\infty$ and $\boldsymbol{k} \in \mathbb{N}_0^d$ we have*

$$\|q_{\boldsymbol{k}}(f)\|_{W_0^{1,p}} \le \frac{2^{-\alpha|\boldsymbol{k}|_1+1}}{(p+1)^{\frac{d-1}{p}} 2^{(\alpha+1)d}} |2^{\boldsymbol{k}}|_p.$$

*Proof.* Let us prove the lemma for the case $1 \le p < \infty$. The case $p = \infty$ can be proven similarly with a slight modification. For $\boldsymbol{k} \in \mathbb{N}_0^d$, by disjoint supports of $\varphi_{\boldsymbol{k},\boldsymbol{s}}$, $\boldsymbol{s} \in Z^d(\boldsymbol{k})$, we have

$$\begin{aligned}
\|q_{\boldsymbol{k}}(f)\|^p_{W_0^{1,p}} &= \sum_{i=1}^d \int_{\mathbb{I}^d} \left| \sum_{\boldsymbol{s}\in Z^d(\boldsymbol{k})} \lambda_{\boldsymbol{k},\boldsymbol{s}}(f) \frac{\partial}{\partial x_i} \varphi_{\boldsymbol{k},\boldsymbol{s}}(\boldsymbol{x}) \right|^p \mathrm{d}\boldsymbol{x} \\
&\le \sup_{\boldsymbol{s}\in Z^d(\boldsymbol{k})} |\lambda_{\boldsymbol{k},\boldsymbol{s}}(f)|^p \sum_{i=1}^d |Z^d(\boldsymbol{k})| \left( \prod_{j\ne i} 2 \int_0^{2^{-k_j-1}} |2^{k_j+1} x_j|^p \,\mathrm{d}x_j \right) \left( 2 \int_0^{2^{-k_i-1}} 2^{pk_i+p} \mathrm{d}x_i \right) \\
&\le \left( \frac{2^{-\alpha|\boldsymbol{k}|_1}}{2^{(\alpha+1)d}} \right)^p \sum_{i=1}^d 2^{|\boldsymbol{k}|_1} \left( \prod_{j\ne i} \frac{2}{p+1} 2^{-k_j-1} \right) 2 \cdot 2^{(k_i+1)(p-1)} \\
&= \left( \frac{2^{-\alpha|\boldsymbol{k}|_1}}{2^{(\alpha+1)d}} \right)^p \frac{1}{(p+1)^{d-1}} \sum_{i=1}^d 2^{p(k_i+1)}.
\end{aligned}$$

This proves the claim. ∎

# 3 Sparse-grid sampling recovery

In this section we consider the approximation by explicitly constructed linear sampling operator on sparse grids with the error measured in the norm of the space $W_0^{1,p}$. The optimality is investigated in terms of sampling $n$-widths. We prove some tight dimension-dependent error estimates of the sampling $n$-widths explicit in $d$ and $n$.

We start with constructing sparse grids and sampling operators on them for approximately recovering functions in $\mathring{U}^\alpha_\infty$ from their values on these grids. For $\beta \ge 1$ and $m \in \mathbb{N}$, we define the sets of multi-indices

$$\Delta^d_\beta(m) := \big\{ \boldsymbol{k} \in \mathbb{N}_0^d : \ |\boldsymbol{k}|_1 = m - j, \ |\boldsymbol{k}|_\infty \ge m - \lfloor \beta j \rfloor \quad \text{for} \quad j = 0, \dots, m \big\},$$

and

$$D^d_\beta(m) := \big\{ (\boldsymbol{k}, \boldsymbol{s}) : \boldsymbol{k} \in \Delta^d_\beta(m), \ \boldsymbol{s} \in Z^d(\boldsymbol{k}) \big\}.$$

The definition of $\Delta^d_\beta(m)$ is similar to $\Delta(\alpha, \beta; \xi) = \big\{ \boldsymbol{k} \in \mathbb{N}_0^d : \ \alpha|\boldsymbol{k}|_1 - \beta|\boldsymbol{k}|_\infty \le \xi \big\}$ introduced in [5] but simpler. We also put

$$\Delta^d(m) := \big\{ \boldsymbol{k} \in \mathbb{N}_0^d : |\boldsymbol{k}|_1 \le m \big\}.$$

It is obvious that $\Delta^d_\beta(m)$ is a subset of $\Delta^d(m)$ for all $\beta \ge 1$.

Consider the operator

$$R_\beta(m, f) := \sum_{\boldsymbol{k}\in\Delta^d_\beta(m)} q_{\boldsymbol{k}}(f) = \sum_{\boldsymbol{k}\in\Delta^d_\beta(m)} \sum_{\boldsymbol{s}\in Z^d(\boldsymbol{k})} \lambda_{\boldsymbol{k},\boldsymbol{s}}(f) \varphi_{\boldsymbol{k},\boldsymbol{s}}, \tag{3.1}$$

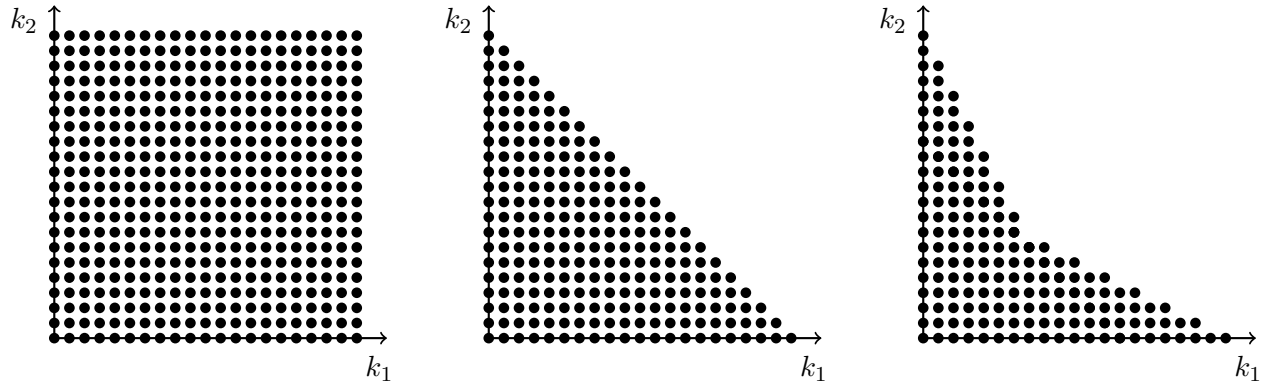


Figure 1: Illustration of different sets of multi-indices in $\mathbb{N}_0^2$. The left graph is $\{\boldsymbol{k} \in \mathbb{N}_0^2 : |\boldsymbol{k}|_\infty \le 20\}$, the middle is $\Delta^2(20)$ and the right is $\Delta_2^2(20)$.

and the energy-norm-based grid

$$G_\beta^d(m) := \big\{2^{-\boldsymbol{k}}\boldsymbol{s} : \boldsymbol{k} \in \Delta_\beta^d(m), \boldsymbol{s} \in Z^d(\boldsymbol{k}+\mathbf{1})\big\}, \qquad \mathbf{1} := (1,\dots,1) \in \mathbb{R}^d.$$

It is easy to see that

$$G_\beta^d(m) = \big\{2^{-\boldsymbol{k}}\boldsymbol{s} : \boldsymbol{k} \in \bar{\Delta}_\beta^d(m),\ \boldsymbol{s} \in Z^d(\boldsymbol{k}+\mathbf{1})\big\},$$

where

$$\bar{\Delta}_\beta^d(m) := \big\{\boldsymbol{k} \in \Delta_\beta^d(m) :\ \boldsymbol{k}+\boldsymbol{e}^j \notin \Delta_\beta^d(m) \text{ for all } j \in [d]\big\},$$

and $\{\boldsymbol{e}^j\}_{j\in[d]}$ is the standard basis in $\mathbb{R}^d$.

We notice some important properties of the operator $R_\beta(m,\cdot)$ and the grid $G_\beta^d(m)$. The function $R_\beta(m,f)$ completely determined by the values of $f$ on the grid $G_\beta^d(m)$. Moreover, $R_\beta(m,f)$ interpolates $f$ at the points of $G_\beta^d(m)$. Thus, $R_\beta(m,\cdot)$ is an interpolation sampling operator on the grid $G_\beta^d(m)$. As shown in what follows, with an appropriate choice of parameter $\beta$, the function $R_\beta(m,f)$ is suitable to approximately recovering the function $f$ in $\mathring{U}_\infty^\alpha$ from the sample values on the grid $G_\beta^d(m)$.

The following lemma gives an upper estimate of the cardinality of $D_\beta^d(m)$ and $G_\beta^d(m)$ showing their sparsity.

**Lemma 3.1** *Let $d \in \mathbb{N}$. We have for every $\beta > 1$ and $m \in \mathbb{N}$,*

$$\big|D_\beta^d(m)\big| \le \frac{\beta}{\beta-1} d\Big(1-2^{-\frac{1}{\beta-1}}\Big)^{-d} 2^m,$$

*and hence,*

$$\big|G_\beta^d(m)\big| \le \frac{\beta}{\beta-1} d 2^d\Big(1-2^{-\frac{1}{\beta-1}}\Big)^{-d} 2^m.$$

*Proof.* Since $\big|G_\beta^d(m,f)\big| \le \big|D_\beta^d(m+d)\big|$, it is sufficient to prove the first estimate in the lemma. We have

$$|D_\beta^d(m)| = \sum_{\boldsymbol{k}\in\Delta_\beta^d(m)} 2^{|\boldsymbol{k}|_1} = 2^m \sum_{j=0}^{m} 2^{-j} \sum_{\substack{|\boldsymbol{k}|_1=m-j \\ |\boldsymbol{k}|_\infty \ge m-\lfloor \beta j\rfloor}} 1.$$

Note, that for $i \in \mathbb{N}_0$ and $\ell \in [d]$ there are $\binom{d-2+\lfloor(\beta-1)j\rfloor-i}{d-2}$ multi-indices $\boldsymbol{k} \in \mathbb{N}_0^d$ satisfying

$$k_\ell = m - \lfloor \beta j \rfloor + i, \qquad \text{and} \qquad \sum_{r \neq \ell} k_r = \lfloor (\beta-1)j \rfloor - i.$$

From this we can estimate

$$\sum_{\substack{|\boldsymbol{k}|_1 = m-j \\ |\boldsymbol{k}|_\infty \geq m - \lfloor \beta j \rfloor}} 1 \leq d \sum_{i=0}^{\lfloor(\beta-1)j\rfloor} \binom{d-2+\lfloor(\beta-1)j\rfloor - i}{d-2} = d\binom{d-1+\lfloor(\beta-1)j\rfloor}{d-1}.$$

Hence,

$$|D_\beta^d(m)| \leq d 2^m \sum_{j=0}^{m} 2^{-j} \binom{d-1+\lfloor(\beta-1)j\rfloor}{d-1}.$$

Putting $\lfloor(\beta-1)j\rfloor = k$ we obtain $j \geq \frac{k}{\beta-1}$ and

$$\big|\{j \in \mathbb{N}_0 : \lfloor(\beta-1)j\rfloor = k\}\big| < \frac{1}{\beta-1} + 1$$

which leads to

$$|D_\beta^d(m)| \leq d 2^m \Big(\frac{1}{\beta-1}+1\Big) \sum_{j=0}^{\infty} 2^{-\frac{1}{\beta-1}j} \binom{d-1+j}{d-1} = d 2^m \frac{\beta}{\beta-1} \big(1 - 2^{-\frac{1}{\beta-1}}\big)^{-d}.$$

The last equality is due to

$$\sum_{j=0}^{\infty} x^j \binom{k+j}{k} = (1-x)^{-k-1} \tag{3.2}$$

for $k \in \mathbb{N}_0$ and $x \in (0,1)$ which is obtained by taking $k$th derivative both sides of $(1-x)^{-1} = \sum_{j=0}^{\infty} x^j$. The proof is completed. □

We continue by proving a supplementary result.

**Lemma 3.2** *Let $d \in \mathbb{N}$, $\ell \in \mathbb{N}_0$, and $1 \leq p \leq \infty$. Then it holds*

$$\sum_{\boldsymbol{k} \in \mathbb{N}_0^d, |\boldsymbol{k}|_1 = \ell} |2^{\boldsymbol{k}}|_p \leq d 2^{\ell+d-1}. \tag{3.3}$$

*Proof.* By monotonicity in $p$ of $\ell_p$ norms, it is enough to prove the lemma for $p = 1$. We use induction argument with respect to $d$. It is obvious that the inequality holds for $d = 1$ and $\ell \in \mathbb{N}_0$ or $d \in \mathbb{N}$ and $\ell = 0$. Assume that

$$\sum_{\boldsymbol{k} \in \mathbb{N}_0^d, |\boldsymbol{k}|_1 = j} \sum_{i=1}^{d} 2^{k_i} \leq d 2^{j+d-1}$$

for $j = 0, \ldots, \ell$. We show that the inequality (3.3) holds for $d+1$ instead of $d$. Indeed, we have

$$\begin{aligned}
\sum_{\boldsymbol{k} \in \mathbb{N}_0^{d+1}, |\boldsymbol{k}|_1 = \ell} \sum_{i=1}^{d+1} 2^{k_i} &= \sum_{j=0}^{\ell} \sum_{k_1 + \ldots + k_d = \ell - j} \Big(2^j + \sum_{i=1}^{d} 2^{k_i}\Big) \\
&\leq \sum_{j=0}^{\ell} d 2^{\ell-j+d-1} + \sum_{j=0}^{\ell} 2^j \binom{\ell-j+d-1}{d-1} \\
&= d 2^{d-1}(2^{\ell+1}-1) + 2^\ell \sum_{j=0}^{\ell} 2^{j-\ell} \binom{\ell-j+d-1}{d-1}.
\end{aligned}$$

Using (3.2) we finally obtain

$$\sum_{\boldsymbol{k}\in\mathbb{N}_0^{d+1},|\boldsymbol{k}|_1=\ell}\sum_{i=1}^{d+1}2^{k_i}\le d2^{d-1}(2^{\ell+1}-1)+2^{\ell+d}\le(d+1)2^{\ell+d}.$$

The proof is completed. ∎

We give a preliminary dimension-dependent error estimate in terms of parameter $m$ of the approximation of a $f\in\mathring{U}_\infty^\alpha$ by the sampling operator $R_\beta(m,\cdot)$.

**Theorem 3.1** *Let $d\ge 2$, $1<\alpha\le 2$, $\beta>\alpha$, and $1\le p\le\infty$. Then for every $f\in\mathring{U}_\infty^\alpha$ we have*

$$\|f-R_\beta(m,f)\|_{W_0^{1,p}}\le K_1\frac{d^2 2^{-m(\alpha-1)}}{(p+1)^{\frac{d}{p}}2^{(\alpha+1)d}\big(1-2^{-\frac{\beta-\alpha}{\beta-1}}\big)^d},\tag{3.4}$$

*where $K_1=K_1(\alpha,\beta,p)=2(p+1)^{\frac{1}{p}}\max\big\{\frac{2\beta}{\beta-1},\frac{1}{2^{\alpha-1}-1}\big\}$.*

*Proof.* Let us prove the theorem for the case $1\le p<\infty$. The case $p=\infty$ can be proven similarly with a slight modification. Recall that $\Delta_\beta^d(m)$ with $\beta>1$ is a subset of $\Delta^d(m)$. Hence, for every $f\in\mathring{U}_\infty^\alpha$ we have

$$\begin{aligned}\|f-R_\beta(m,f)\|_{W_0^{1,p}}&\le\sum_{\boldsymbol{k}\in\mathbb{N}_0^d\setminus\Delta_\beta^d(m)}\|q_{\boldsymbol{k}}(f)\|_{W_0^{1,p}}\\&=\sum_{\boldsymbol{k}\in\mathbb{N}_0^d\setminus\Delta^d(m)}\|q_{\boldsymbol{k}}(f)\|_{W_0^{1,p}}+\sum_{\boldsymbol{k}\in\Delta^d(m)\setminus\Delta_\beta^d(m)}\|q_{\boldsymbol{k}}(f)\|_{W_0^{1,p}}.\end{aligned}\tag{3.5}$$

For the first term on the right side, from Lemma 2.2 we have

$$\begin{aligned}\sum_{\boldsymbol{k}\in\mathbb{N}_0^d\setminus\Delta^d(m)}\|q_{\boldsymbol{k}}(f)\|_{W_0^{1,p}}&\le\frac{1}{(p+1)^{\frac{d-1}{p}}2^{(\alpha+1)d}}\sum_{\boldsymbol{k}\in\mathbb{N}_0^d,|\boldsymbol{k}|_1>m}2^{-\alpha|\boldsymbol{k}|_1+1}|2^{\boldsymbol{k}}|_p\\&=\frac{2}{(p+1)^{\frac{d-1}{p}}2^{(\alpha+1)d}}\sum_{\ell=m+1}^{\infty}2^{-\alpha\ell}\sum_{\boldsymbol{k}\in\mathbb{N}_0^d,|\boldsymbol{k}|_1=\ell}|2^{\boldsymbol{k}}|_p.\end{aligned}$$

In view of Lemma 3.2 we get

$$\sum_{\boldsymbol{k}\in\mathbb{N}_0^d\setminus\Delta^d(m)}\|q_{\boldsymbol{k}}(f)\|_{W_0^{1,p}}\le\frac{d}{(p+1)^{\frac{d-1}{p}}2^{\alpha d}}\sum_{\ell=m+1}^{\infty}2^{-(\alpha-1)\ell}=\frac{d2^{-(\alpha-1)m}}{(p+1)^{\frac{d-1}{p}}2^{\alpha d}(2^{\alpha-1}-1)}.$$

We now consider the second term on the right side of (3.5). Denote $j^*$ the maximum value of $j$ such that the set

$$\big\{\boldsymbol{k}\in\mathbb{N}_0^d:\ |\boldsymbol{k}|_1=m-j,\ |\boldsymbol{k}|_\infty<m-\lfloor\beta j\rfloor\big\}$$

is not empty. Following the argument in the proof of [4, Theorem 3.10] and using Lemma 2.2 we get

$$\begin{aligned}\sum_{\boldsymbol{k}\in\Delta^d(m)\setminus\Delta^d_\beta(m)}\|q_{\boldsymbol{k}}(f)\|_{W_0^{1,p}} &\le \sum_{j=0}^{j^*}\sum_{\substack{|\boldsymbol{k}|_1=m-j\\ |\boldsymbol{k}|_\infty<m-\lfloor\beta j\rfloor}}\|q_{\boldsymbol{k}}(f)\|_{W_0^{1,p}}\\ &\le \frac{2}{(p+1)^{\frac{d-1}{p}}2^{(\alpha+1)d}}\sum_{j=0}^{j^*}\sum_{\substack{|\boldsymbol{k}|_1=m-j\\ |\boldsymbol{k}|_\infty<m-\lfloor\beta j\rfloor}}2^{-\alpha|\boldsymbol{k}|_1}|2^{\boldsymbol{k}}|_p \qquad (3.6)\\ &= \frac{2^{-\alpha m+1}}{(p+1)^{\frac{d-1}{p}}2^{(\alpha+1)d}}\sum_{j=0}^{j^*}2^{\alpha j}\sum_{\substack{|\boldsymbol{k}|_1=m-j\\ |\boldsymbol{k}|_\infty<m-\lfloor\beta j\rfloor}}|2^{\boldsymbol{k}}|_p.\end{aligned}$$

The sum on the right side can be estimated as

$$\begin{aligned}\sum_{j=0}^{j^*}2^{\alpha j}\sum_{\substack{|\boldsymbol{k}|_1=m-j\\ |\boldsymbol{k}|_\infty<m-\lfloor\beta j\rfloor}}|2^{\boldsymbol{k}}|_p &\le \sum_{j=0}^{j^*}2^{\alpha j}\sum_{i=1}^{m-1-\lfloor\beta j\rfloor}d\binom{m+d-2-i-j}{d-2}2^i\\ &= d2^{m-1}\sum_{j=0}^{j^*}2^{-\lfloor(\beta-\alpha)j\rfloor}\sum_{\ell=0}^{m-2-\lfloor\beta j\rfloor}\binom{d-1+\lfloor(\beta-1)j\rfloor+\ell}{d-2}2^{-\ell},\end{aligned}$$

where in the equality we put $i=m-1-\lfloor\beta j\rfloor-\ell$. As in the proof of Lemma 3.1 we put $\lfloor(\beta-1)j\rfloor=k$ and get $\lfloor(\beta-\alpha)j\rfloor\ge\frac{k(\beta-\alpha)}{\beta-1}-1$. Since $\frac{\beta-\alpha}{\beta-1}<1$, we have

$$\begin{aligned}\sum_{j=0}^{j^*}2^{\alpha j}\sum_{\substack{|\boldsymbol{k}|_1=m-j\\ |\boldsymbol{k}|_\infty<m-\lfloor\beta j\rfloor}}|2^{\boldsymbol{k}}|_p &\le \frac{d2^m\beta}{\beta-1}\sum_{k=0}^{\infty}\sum_{\ell=0}^{\infty}\binom{d-1+k+\ell}{d-2}2^{-\ell}2^{-\frac{\beta-\alpha}{\beta-1}k}\\ &\le \frac{d2^m\beta}{\beta-1}\sum_{k=0}^{\infty}\sum_{\ell=0}^{\infty}\binom{d-1+k+\ell}{d-2}2^{-\frac{\beta-\alpha}{\beta-1}(k+\ell)}\\ &= \frac{d2^m\beta}{\beta-1}\sum_{j=0}^{\infty}\binom{d-1+j}{d-2}2^{-\frac{\beta-\alpha}{\beta-1}j}(j+1)\\ &= (d-1)\frac{d2^m\beta}{\beta-1}\sum_{j=0}^{\infty}\binom{d-1+j}{d-1}2^{-\frac{\beta-\alpha}{\beta-1}j}.\end{aligned}$$

From the assumption $\beta>\alpha>1$ and (3.2) we arrive at

$$\sum_{j=0}^{j^*}2^{\alpha j}\sum_{\substack{|\boldsymbol{k}|_1=m-j\\ |\boldsymbol{k}|_\infty<m-\lfloor\beta j\rfloor}}|2^{\boldsymbol{k}}|_p \le \frac{d^2 2^m\beta}{\beta-1}\big(1-2^{-\frac{\beta-\alpha}{\beta-1}}\big)^{-d}.$$

Inserting this into (3.6) we get

$$\sum_{\boldsymbol{k}\in\Delta^d(m)\setminus\Delta^d_\beta(m)}\|q_{\boldsymbol{k}}(f)\|_{W_0^{1,p}} \le \frac{2\beta}{\beta-1}\frac{d^2 2^{-m(\alpha-1)}}{(p+1)^{\frac{d-1}{p}}2^{(\alpha+1)d}\big(1-2^{-\frac{\beta-\alpha}{\beta-1}}\big)^d}.$$

Since $2^{-d}\big(1-2^{-\frac{\beta-\alpha}{\beta-1}}\big)^{-d}>1$ which is due to $\alpha>1$, we finally obtain the desired estimate. ∎

**Remark 3.2** *For approximation of functions from $\mathring{U}^\alpha_\infty$, we could take the sampling operator $R_\square(m,f) := \sum_{|\boldsymbol{k}|_\infty \le m} q_{\boldsymbol{k}}(f)$ on the corresponding traditional standard grid*

$$G^d_\square(m) := \big\{2^{-\boldsymbol{k}}\boldsymbol{s} : |\boldsymbol{k}|_\infty = m+1,\ \boldsymbol{s} \in Z^d(\boldsymbol{k})\big\},$$

*and the sampling operator $R_\triangle(m,f) := \sum_{|\boldsymbol{k}|_1 \le m} q_{\boldsymbol{k}}(f)$ on the corresponding Smolyak grid*

$$G^d_\triangle(m) := \big\{2^{-\boldsymbol{k}}\boldsymbol{s} : |\boldsymbol{k}|_1 = m,\ \boldsymbol{s} \in Z^d(\boldsymbol{k}+\boldsymbol{1})\big\}.$$

*It is easy to see that the error of approximation in the $W^{1,p}_0$ norm of $f \in \mathring{U}^\alpha_\infty$ by $R_\square(m,f)$ or $R_\triangle(m,f)$ is the same as the error of approximation in the $W^{1,p}_0$ norm of $f \in \mathring{U}^\alpha_\infty$ by $R_\beta(m,f)$. On the other hand, we notice that the sparsity of the gird $G^d_\beta(m)$ in the operator $R_\beta(m,f)$, is much higher than the sparsity of the grids $G^d_\square(m)$ and $G^d_\triangle(m)$, see the estimate of $|G^d_\beta(m)|$ in Lemma 3.1 in comparing with $|G^d_\square(m)| \approx 2^{dm}$ and $|G^d_\triangle(m)| \approx 2^{m+d}\binom{m+d-1}{d-1}$.*

*Some results similar to (3.4) were obtained in [4, Theorem 3.8] for the approximation in energy norm of functions with 2nd mixed derivatives bounded in $L_2$- or $L_\infty$-norm.*

In the following theorem, from Theorem 3.1 we deduce a dimension-dependent estimate of the computation complexity characterized by the number of sample values in the operator $R_\beta(m,\cdot)$ necessary for approximating functions from $\mathring{U}^\alpha_\infty$ with an accuracy $\varepsilon$. This estimate will be used as an auxiliary result in the next section on approximation of functions from $\mathring{U}^\alpha_\infty$ by deep ReLU neural networks.

**Theorem 3.3** *Let $d \ge 2$, $1 < \alpha \le 2$, $\beta > \alpha$, and $1 \le p \le \infty$. Then there is $\varepsilon_0 \in (0,1]$ such that for every $\varepsilon \in (0,\varepsilon_0)$ and every $f \in \mathring{U}^\alpha_\infty$ we have*

$$\|f - R_\beta(m,f)\|_{W^{1,p}_0} \le \varepsilon$$

*and*

$$|D^d_\beta(m)| \le K_2 B_1^{-d} (\varepsilon^{-1})^{\frac{1}{\alpha-1}}, \quad |G^d_\beta(m)| \le K_2 (B_1/2)^{-d} (\varepsilon^{-1})^{\frac{1}{\alpha-1}},$$

*with*

$$m := \left\lceil \frac{1}{\alpha-1} \log\left( \frac{K_1 d^2 \varepsilon^{-1}}{(p+1)^{\frac{d}{p}} 2^{(\alpha+1)d} \big(1 - 2^{-\frac{\beta-\alpha}{\beta-1}}\big)^d} \right) \right\rceil \tag{3.7}$$

*and*

$$B_1 = B_1(d,\alpha,\beta,p) := \big(1 - 2^{-\frac{1}{\beta-1}}\big) \left( \frac{(p+1)^{\frac{1}{p}} 2^{(\alpha+1)} \big(1 - 2^{-\frac{\beta-\alpha}{\beta-1}}\big)}{d^{\frac{\alpha+1}{d}}} \right)^{\frac{1}{\alpha-1}}, \tag{3.8}$$

*where $K_1$ is the constant in Theorem 3.1 and $K_2 = K_2(\alpha,\beta,p) := \frac{2\beta}{\beta-1} K_1^{\frac{1}{\alpha-1}}$. Moreover, if $\alpha$ and $p$ satisfy*

$$2^{\frac{1}{\alpha}} - (p+1)^{-\frac{1}{p\alpha}}/2 > 1 \tag{3.9}$$

*and*

$$\frac{\alpha + \log\big(1 - (p+1)^{-\frac{1}{p\alpha}} 2^{-1-\frac{1}{\alpha}}\big)}{1 + \log\big(1 - (p+1)^{-\frac{1}{p\alpha}} 2^{-1-\frac{1}{\alpha}}\big)} < \beta < 1 - \frac{1}{\log\big(1 - (p+1)^{-\frac{1}{p\alpha}} 2^{-1-\frac{1}{\alpha}}\big)} \tag{3.10}$$

*then there exist constants $d(\alpha,\beta,p) \in \mathbb{N}$ and $B(\alpha,\beta,p) > 1$ such that $B_1 \ge B(\alpha,\beta,p) > 1$ for all $d \ge d(\alpha,\beta,p)$.*

*Proof.* Choose

$$\varepsilon_0 = \min\left\{1, \frac{K_1 d^2}{(p+1)^{\frac{d}{p}} 2^{(\alpha+1)d}\big(1-2^{-\frac{\beta-\alpha}{\beta-1}}\big)^d}\right\}.$$

For $\varepsilon \in (0, \varepsilon_0)$ we take $m$ as in (3.7). This implies that

$$m - 1 < \frac{1}{\alpha - 1} \log\left(\frac{K_1 d^2 \varepsilon^{-1}}{(p+1)^{\frac{d}{p}} 2^{(\alpha+1)d}\big(1-2^{-\frac{\beta-\alpha}{\beta-1}}\big)^d}\right) \le m.$$

Hence from Theorem 3.1 and Lemma 3.1 we obtain

$$\|f - R_\beta(m, f)\|_{W_0^{1,p}} \le \varepsilon$$

and

$$\begin{aligned}
|D_\beta^d(m)| &\le d 2^m \frac{\beta}{\beta-1}\big(1-2^{-\frac{1}{\beta-1}}\big)^{-d} \\
&\le 2d \frac{\beta}{\beta-1}\big(1-2^{-\frac{1}{\beta-1}}\big)^{-d}\left(\frac{K_1 d^2 \varepsilon^{-1}}{(p+1)^{\frac{d}{p}} 2^{(\alpha+1)d}\big(1-2^{-\frac{\beta-\alpha}{\beta-1}}\big)^d}\right)^{\frac{1}{\alpha-1}} \\
&\le \frac{2\beta}{\beta-1} K_1^{\frac{1}{\alpha-1}}\big(1-2^{-\frac{1}{\beta-1}}\big)^{-d}\left(\frac{d^{(\alpha+1)/d}}{(p+1)^{\frac{1}{p}} 2^{(\alpha+1)}\big(1-2^{-\frac{\beta-\alpha}{\beta-1}}\big)}\right)^{\frac{d}{\alpha-1}} (\varepsilon^{-1})^{\frac{1}{\alpha-1}} \\
&= K_2 B_1^{-d} (\varepsilon^{-1})^{\frac{1}{\alpha-1}}
\end{aligned} \tag{3.11}$$

which is the first statement. To prove the second one, we show that under the conditions (3.9) and (3.10) it holds

$$\big(1-2^{-\frac{1}{\beta-1}}\big)^{(\alpha-1)}(p+1)^{\frac{1}{p}} 2^{\alpha+1}\big(1-2^{-\frac{\beta-\alpha}{\beta-1}}\big) > 1.$$

Indeed, the last inequality is equivalent to

$$\frac{1}{(p+1)^{\frac{1}{p}} 2^{\alpha+1}} < \big(1-2^{-\frac{1}{\beta-1}}\big)^{(\alpha-1)}\big(1-2^{-\frac{\beta-\alpha}{\beta-1}}\big). \tag{3.12}$$

If $\beta - \alpha \ge 1$ then $\big(1-2^{-\frac{1}{\beta-1}}\big) \le \big(1-2^{-\frac{\beta-\alpha}{\beta-1}}\big)$. Hence, the last inequality is fulfilled if

$$\frac{1}{(p+1)^{\frac{1}{p}} 2^{\alpha+1}} < \big(1-2^{-\frac{1}{\beta-1}}\big)^{\alpha} \qquad \Longleftrightarrow \qquad \beta < 1 - \frac{1}{\log\big(1-(p+1)^{-\frac{1}{p\alpha}} 2^{-1-\frac{1}{\alpha}}\big)}.$$

If $\beta - \alpha < 1$ then $\big(1-2^{-\frac{1}{\beta-1}}\big) > \big(1-2^{-\frac{\beta-\alpha}{\beta-1}}\big)$. Hence the inequality (3.12) is fulfilled if

$$\frac{1}{(p+1)^{\frac{1}{p}} 2^{\alpha+1}} < \big(1-2^{-\frac{\beta-\alpha}{\beta-1}}\big)^{\alpha} \qquad \Longleftrightarrow \qquad \beta > \frac{\alpha + \log\big(1-(p+1)^{-\frac{1}{p\alpha}} 2^{-1-\frac{1}{\alpha}}\big)}{1 + \log\big(1-(p+1)^{-\frac{1}{p\alpha}} 2^{-1-\frac{1}{\alpha}}\big)}.$$

Assigning

$$1 - \frac{1}{\log\big(1-(p+1)^{-\frac{1}{p\alpha}} 2^{-1-\frac{1}{\alpha}}\big)} > \frac{\alpha + \log\big(1-(p+1)^{-\frac{1}{p\alpha}} 2^{-1-\frac{1}{\alpha}}\big)}{1 + \log\big(1-(p+1)^{-\frac{1}{p\alpha}} 2^{-1-\frac{1}{\alpha}}\big)}$$

we find $2^{\frac{1}{\alpha}} - (p+1)^{-\frac{1}{p\alpha}}/2 > 1$. Since $d^{\frac{\alpha+1}{d}}$ tends to 1 when $d \to \infty$, there are $d(\alpha, \beta, p) \in \mathbb{N}$ and $B(\alpha, \beta, p) > 1$ such that $B_1 \ge B(\alpha, \beta, p) > 1$ for all $d \ge d(\alpha, \beta, p)$. The proof is completed. ∎

**Remark 3.4** *For $1 \le p \le 2$ the condition (3.9) is satisfied for all $\alpha \in (1,2]$, and, therefore, we can always find $\beta = \beta(\alpha) > \alpha$, $d(\alpha) \in \mathbb{N}$, and $B(\alpha) > 1$ such that $B_1 \ge B(\alpha) > 1$ for all $d \ge d(\alpha)$. In this case, the approximation error is going to 0 as fast as the exponent $B_1^{-d}$ when $d$ tending to $\infty$.*

We show the optimality in terms of sampling $n$-widths of the sampling operator $R_\beta(m,\cdot)$ on the grid $G_\beta^d(m)$ with the number of sample points $|G_\beta^d(m)| \le n$, by giving dimension-dependent tight upper and lower bounds of the sampling $n$-widths $r_n$.

**Theorem 3.5** *Let $d \ge 2$, $1 < \alpha \le 2$, $\beta > \alpha$, and $1 \le p \le \infty$. Then there is $n_0 \in \mathbb{N}$ such that for $n \ge n_0$ we have*

$$\frac{3}{M4^{\alpha-1}}\Big(\frac{3}{2p+2}\Big)^{1/p} B_*^{-d} n^{-(\alpha-1)} \le r_n \le K_2^{\alpha-1} B^{*-d} n^{-(\alpha-1)} \tag{3.13}$$

*where $B_* := 18 \cdot 3^{1/p} M^{-1}$, $M = \|M_3\|_{L_p(\mathbb{R})}$, $B^* := (B_1/2)^{\alpha-1}$, $K_2$ and $B_1$ are given in Theorem 3.3. The lower bound holds for $n \ge 2$. Moreover, we can explicitly define $m(n) \in \mathbb{N}$ so that $|G_\beta^d(m(n))| \le n$ and for the sampling operator $R_\beta(m(n),\cdot)$ on the grid $G_\beta^d(m(n))$, we have that*

$$r_n \le \sup_{f \in \mathring{U}_\infty^\alpha} \|f - R_\beta(m(n), f)\|_{W_0^{1,p}} \le K_2^{\alpha-1} B^{*-d} n^{-(\alpha-1)}.$$

*Proof.*
*Upper bound.* We take $n_0$ the smallest positive integer such that $\big(K_2(B_1/2)^{-d}\big)^{\alpha-1} n_0^{-(\alpha-1)} < \varepsilon_0$. For $n \ge n_0$, we define

$$\varepsilon = \big(K_2(B_1/2)^{-d}\big)^{\alpha-1} n^{-(\alpha-1)}.$$

By Theorem 3.3 there is $m(n) \in \mathbb{N}$ such that

$$\|f - R_\beta(m(n), f)\|_{W_0^{1,p}} \le \big(K_2(B_1/2)^{-d}\big)^{\alpha-1} n^{-(\alpha-1)},$$

and $|G_\beta^d(m(n))| \le n$. If we define the set of points $X_n = G_\beta^d(m(n))$ then by the definition of $R_\beta(m(n), f)$ we obtain

$$r_n \le \sup_{f \in \mathring{U}_\infty^\alpha} \|f - R_\beta(m(n), f)\|_{W_0^{1,p}} \le \big(K_2(B_1/2)^{-d}\big)^{\alpha-1} n^{-(\alpha-1)} = K_2^{\alpha-1} B^{*-d} n^{-(\alpha-1)}.$$

*Lower bound.* Let us prove the lower bound for the case $1 \le p < \infty$. The case $p = \infty$ can be proven similarly with a slight modification. Our strategy to prove the lower bound is as follows. Based on the obvious inequality

$$r_n \ge \inf_{X_n \subset \mathbb{I}^d} \sup_{f \in \mathring{U}_\infty^\alpha,\ f(\boldsymbol{x}^j)=0, \boldsymbol{x}^j \in X_n} \|f\|_{W_0^{1,p}}, \tag{3.14}$$

where the infimum is taken over all sets $X_n$ of $n$ points. For any point set $X_n = \{\boldsymbol{x}^j\}_{j=1}^n$ in $\mathbb{I}^d$, we will construct a test function $f \in \mathring{U}_\infty^\alpha$ vanishing at the points $\boldsymbol{x}^j$, $j = 1, \dots, n$, and prove that the norm $\|f\|_{W_0^{1,p}}$ is bounded from below by the quantity in the left side of (3.13).

Let $M_3$ be the B-spline with knots at the points $0, 1, 2, 3$, i.e.,

$$M_3(x) = \frac{1}{2}\begin{cases} x^2 & \text{if } 0 \le x < 1 \\ -2x^2 + 6x - 3 & \text{if } 1 \le x < 2 \\ (3-x)^2 & \text{if } 2 \le x < 3 \end{cases}$$

and $M_3(x) = 0$ otherwise. Let $\psi(x) := M_3(3x)$. We define the univariate non-negative functions $\psi_{k,s}$ by

$$\psi_{k,s}(x) := \ \psi(2^k x - s + 1),\ k \in \mathbb{N}_0,\ s = 1, \dots, 2^k.$$

One can also verify that

$$\operatorname{supp} \psi_{k,s} \ = \ I_{k,s} \ =: [2^{-k}(s-1), 2^{-k}s], \quad \operatorname{int} I_{k,s} \cap \operatorname{int} I_{k,s'} \ = \ \varnothing,\ s \neq s'.$$

For every $n \in \mathbb{N}$, $n \geq 2$, we choose $m \in \mathbb{N}$ such that $2^{m-2} < n \leq 2^{m-1}$. Let $J_m$ be the set of $s$ such that $X_n \cap \operatorname{int}\big(I_{m,s} \times [0,1]^{d-1}\big) = \emptyset$. Then we have $|J_m| \geq 2^{m-1}$. We define the function

$$f_m(\boldsymbol{x}) := 18^{-d} 2^{-\alpha m} \left( \sum_{s \in J_m} \psi_{m,s}(x_1) \right) \prod_{j=2}^{d} \psi_{0,1}(x_j), \quad \boldsymbol{x} \in \mathbb{I}^d$$

and show that

$$\Delta_{\boldsymbol{h}}^{2,u}(f_m, \boldsymbol{x}) \ \leq \ \prod_{j \in u} |h_j|^\alpha,\ \boldsymbol{x} \in \mathbb{I}^d,\ \boldsymbol{h} \in [-1,1]^d,\ u \subset [d], \tag{3.15}$$

following partly in [11]. Let us prove this inequality for $u = [d]$ and $\boldsymbol{h} \in \mathbb{I}^d$, the general case of $u$ can be proven in a similar way with a slight modification. We have

$$\Delta_{\boldsymbol{h}}^{2,[d]}(f_m, \boldsymbol{x}) \ = 18^{-d} 2^{-\alpha m} \Delta_{h_1}^2 \left( \sum_{s \in J_m} \psi_{m,s}(x_1) \right) \prod_{j=2}^{d} \Delta_{h_j}^2 \psi_{0,1}(x_j).$$

By using the formula

$$\Delta_h^2(f, x) \ = \ h^2 \int_{\mathbb{R}} f^{(2)}(x+y)[h^{-1} M_2(h^{-1} y)] \, \mathrm{d}y, \quad h \in \mathbb{R}$$

for univariate function $f$ having locally absolutely continuous $f'$, see e.g. [13, page 45], we get

$$\begin{aligned} \Delta_{h_1}^2 \left( \sum_{s \in J_m} \psi_{m,s}(x_1) \right) &= h_1^2 \int_{\mathbb{R}} \left( \sum_{s \in J_m} \psi_{m,s}(t) \right)'' h_1^{-1} M_2\big((h_1^{-1}(t - x_1)\big) \, \mathrm{d}t \\ &= 9 h_1^2 2^{2m} \int_{\mathbb{R}} \left( \sum_{s \in J_m} \chi_{m,s}^0(t) \right) h_1^{-1} M_2\big((h_1^{-1}(t - x_1)\big) \, \mathrm{d}t, \end{aligned}$$

where $\chi_{m,s}^0(t) = \chi_{I_{m,s}^1} - 2\chi_{I_{m,s}^2} + \chi_{I_{m,s}^3}$ and $\chi_{I_{m,s}^j}$ are characteristic functions of the intervals

$$I_{m,s}^j := \left[ 2^{-m}(s-1) + \frac{2^{-m}(j-1)}{3},\ 2^{-m}(s-1) + \frac{2^{-m} j}{3} \right], \qquad j = 1, 2, 3.$$

If $(2^m h_1) \leq 1$ we have from $|\chi_{m,s}^0| \leq 2$

$$\begin{aligned} 2^{-\alpha m} \left| \Delta_{h_1}^2 \left( \sum_{s \in J_m} \psi_{m,s}(x_1) \right) \right| &= 9 h_1^\alpha (2^m h_1)^{(2-\alpha)} \left| \int_{\mathbb{R}} \left( \sum_{s \in J_m} \chi_{m,s}^0(t) \right) h_1^{-1} M_2\big((h_1^{-1}(t - x_1)\big) \, \mathrm{d}t \right| \\ &\leq \ 18 h_1^\alpha \int_{\mathbb{R}} h_1^{-1} M_2\big((h_1^{-1}(t - x_1)\big) \, \mathrm{d}t = 18 h_1^\alpha, \end{aligned} \tag{3.16}$$

where in the last equality we used $\int_{\mathbb{R}} M_2(t) \mathrm{d}t = 1$. If $(2^m h_1) > 1$ we have by changing variable

$$2^{-\alpha m} \left| \Delta_{h_1}^2 \left( \sum_{s \in J_m} \psi_{m,s}(x_1) \right) \right| = 9 h_1^2 2^{m(2-\alpha)} \int_{\mathbb{R}} \left( \sum_{s \in J_m} \chi_{m,s}^0(x_1 + h_1 y) \right) M_2(y) \, \mathrm{d}y.$$

Denote $K_{m,s} = \operatorname{supp}\chi_{m,s}(x_1 + h_1\cdot) = \big[\frac{2^{-m}(s-1)-x_1}{h_1},\ \frac{2^{-m}s-x_1}{h_1}\big]$. If $K_{m,s} \subset [0,1]$ or $K_{m,s} \subset [1,2]$ we have

$$\int_{\mathbb{R}} \chi^0_{m,s}(x_1 + h_1 y)M_2(y)\mathrm{d}y = \int_{\mathbb{R}} \Big(\chi_{I^1_{m,s}} - 2\chi_{I^2_{m,s}} + \chi_{I^3_{m,s}}\Big)(x_1 + h_1 y)M_2(y)\mathrm{d}y = 0$$

and there are at most three $s^j \in J_m$, $j = 0,1,2$ such that $j \in \operatorname{int}\big(K_{m,s^j}\big)$, $j = 0,1,2$. It is not difficult to verify that

$$\left| \int_{\mathbb{R}} \chi^0_{m,s^1}(x_1 + h_1 y)M_2(y)\,\mathrm{d}y \right| \le \frac{3}{2}\Big(\frac{1}{3\cdot 2^m h_1}\Big)^2$$

and

$$\left| \int_{\mathbb{R}} \chi^0_{m,s^j}(x_1 + h_1 y)M_2(y)\,\mathrm{d}y \right| \le \frac{1}{2}\Big(\frac{1}{3\cdot 2^m h_1}\Big)^2$$

if $j = 0,2$. From this we obtain

$$\begin{aligned} 2^{-\alpha m}\left|\Delta^2_{h_1}\Big(\sum_{s\in J_m}\psi_{m,s}(x_1)\Big)\right| &= 9h_1^2 2^{m(2-\alpha)}\left|\int_{\mathbb{R}}\Big(\sum_{j=0,1,2}\chi^0_{m,s^j}(x_1+h_1y)\Big)M_2(y)\,\mathrm{d}y\right| \\ &\le 9h_1^2 2^{m(2-\alpha)}\frac{5}{2}\Big(\frac{1}{3\cdot 2^m h_1}\Big)^2 = \frac{5}{2}h^\alpha\frac{1}{(2^m h_1)^\alpha} \le \frac{5}{2}h_1^\alpha. \end{aligned}$$

Since $h_j \in [0,1]$, $j = 2,\dots,d$, similar to (3.16) we can show that

$$\left|\prod_{j=2}^d \Delta^2_{h_j}\psi_{0,1}(x_j)\right| \le 18^{d-1}\prod_{j=2}^d h_j^\alpha.$$

Consequently, the inequality (3.15) is proven. This means that $f_m \in \mathring{U}^\alpha_\infty$. Due to the disjoint supports of $\psi_{m,s}$ we have

$$\begin{aligned} \|f_m\|^p_{W^{1,p}_0} &= 18^{-pd}\sum_{j=1}^d\int_{\mathbb{I}^d}\left|2^{-\alpha m}\frac{\partial}{\partial x_j}\Big(\sum_{s\in J_m}\psi_{m,s}(x_1)\Big)\prod_{j=2}^d\psi_{0,1}(x_j)\right|^p \mathrm{d}\boldsymbol{x} \\ &\ge 18^{-pd}\int_{\mathbb{I}^d}\left|2^{-\alpha m}\sum_{s\in J_m}\frac{\partial}{\partial x_1}\psi_{m,s}(x_1)\prod_{j=2}^d\psi_{0,1}(x_j)\right|^p \mathrm{d}\boldsymbol{x} \\ &= 18^{-pd}2^{-p\alpha m}|J_m|\int_{\mathbb{I}}\big|\psi'_{m,1}(x_1)\big|^p\mathrm{d}x_1\Big(\int_{\mathbb{I}}|\psi_{0,1}(t)|^p\mathrm{d}t\Big)^{d-1} \\ &\ge 18^{-pd}2^{-p\alpha m}2^{m-1}2^{pm}3^p\int_{\mathbb{I}}\big|M'_3(3\cdot 2^m x_1)\big|^p\mathrm{d}x_1\Big(\int_{\mathbb{I}}|M_3(3t)|^p\mathrm{d}t\Big)^{d-1} \\ &= \frac{3^p}{2}18^{-pd}3^{-d}2^{-pm(\alpha-1)}\int_{\mathbb{R}}\big|M'_3(t)\big|^p\mathrm{d}t\Big(\int_{\mathbb{R}}|M_3(t)|^p\mathrm{d}t\Big)^{d-1}, \end{aligned}$$

where in the last equality we changed variables. Since $f_m(\xi) = 0$, $\xi \in X_n$, and

$$\int_{\mathbb{R}}\big|M'_3(t)\big|^p\mathrm{d}t = \frac{3}{p+1}, \qquad M := \Big(\int_{\mathbb{R}}M_3^p(t)\mathrm{d}t\Big)^{1/p},$$

by (3.14) we get

$$\begin{aligned} r_n \ge \|f_m\|_{W^{1,p}_0} &\ge 18^{-d}\Big(\frac{3^{p+1-d}}{2p+2}\Big)^{1/p}M^{d-1}2^{-m(\alpha-1)} \ge 18^{-d}\Big(\frac{3^{p+1-d}}{2p+2}\Big)^{1/p}M^{d-1}\Big(\frac{n^{-1}}{4}\Big)^{(\alpha-1)} \\ &\ge \frac{1}{M4^{\alpha-1}}\Big(\frac{3^{p+1}}{2p+2}\Big)^{1/p}\big(18\cdot 3^{1/p}M^{-1}\big)^{-d}n^{-(\alpha-1)} = \frac{1}{M4^{\alpha-1}}\Big(\frac{3^{p+1}}{2p+2}\Big)^{1/p}B_*^{-d}n^{-(\alpha-1)}. \end{aligned}$$

The proof is completed. ∎

**Corollary 3.6** *Let $d \geq 2$, $1 < \alpha \leq 2$, $\beta > \alpha$, and $1 \leq p \leq \infty$. If $2^{\frac{2}{\alpha}} - (p+1)^{-\frac{1}{p\alpha}} > 2$, then there is $n_0 \in \mathbb{N}$ such that for $n \geq n_0$ we have*

$$C_1 B_*^{-d} n^{-(\alpha-1)} \leq r_n \leq C_2 B^{*-d} n^{-(\alpha-1)}$$

*with $B_* > 1$ and $B^* > 1$, where the constants $B_* = B_*(p)$, $B^* = B^*(\alpha, p)$, $C_1 = C_1(p)$, $C_2 = C_2(\alpha, p)$ are independent of $n$ and $d$.*

*Proof.* The lower bound in the corollary for the case $1 \leq p \leq \infty$ follows from the case $p = 1$ which is obvious due to the lower bound in Theorem 3.5. By the definitions of $B_1$ in (3.8) and of $B^*$ in Theorem 3.5, to prove the upper bound it is sufficient to show that under the condition $2^{\frac{2}{\alpha}} - (p+1)^{-\frac{1}{p\alpha}} > 2$ it holds

$$2^{1-\alpha}\big(1 - 2^{-\frac{1}{\beta-1}}\big)^{(\alpha-1)}(p+1)^{\frac{1}{p}} 2^{\alpha+1}\big(1 - 2^{-\frac{\beta-\alpha}{\beta-1}}\big) > 1$$

or

$$\big(1 - 2^{-\frac{1}{\beta-1}}\big)^{(\alpha-1)}(p+1)^{\frac{1}{p}} 2^{2}\big(1 - 2^{-\frac{\beta-\alpha}{\beta-1}}\big) > 1.$$

The further argument is similar to the proof of Theorem 3.3. ∎

# 4 Approximation by deep ReLU neural networks

In this section, based on the result on sparse-grid sampling recovery in Theorem 3.3, for any function $f \in \mathring{U}^{\alpha}_{\infty}$, we explicitly construct a deep ReLU neural network having an output that approximates $f$ in the $W_0^{1,p}$-norm with a prescribed accuracy $\varepsilon$ and prove dimension-dependent error estimates of the number of its weights and its depth. We also prove some lower bounds for the number of weights of this deep ReLU neural network necessary for this approximation.

There is a wide variety of neural network architectures and each of them is adapted to specific tasks. For our purpose of approximation functions from Höder-Zygmund spaces, in this section we introduce feed-forward deep ReLU neural networks with one-dimension output. We are interested in deep neural networks where only connections between neighboring layers are allowed. Let us introduce necessary definitions and elementary facts on deep ReLU neural networks.

**Definition 4.1** *Let $d, L \in \mathbb{N}$, $L \geq 2$.*

- *A deep neural network $\Phi$ with input dimension $d$ and $L$ layers is a sequence of matrix-vector tuples*

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

*where $\boldsymbol{W}^\ell = (w^\ell_{i,j})$ is an $N_\ell \times N_{\ell-1}$ matrix, and $\boldsymbol{b}^\ell = (b^\ell_j) \in \mathbb{R}^{N_\ell}$ with $N_0 = d$, $N_L = 1$, and $N_1, \dots, N_{L-1} \in \mathbb{N}$. We call the number of layers $L(\Phi) = L$ the depth and $\boldsymbol{N}(\Phi) = (N_0, N_1, \dots, N_L)$ the dimension of the network. The real numbers $w^\ell_{i,j}$ and $b^\ell_j$ are called edge and node weights of the network $\Phi$, respectively. The number of nonzero weights $w^\ell_{i,j}$ and $b^\ell_j$ is called the number of weights of the network $\Phi$ and denoted by $W(\Phi)$, i.e., $W(\Phi) := \sum_{\ell=1}^L |\boldsymbol{W}^\ell|_0 + \sum_{\ell=1}^L |\boldsymbol{b}^\ell|_0$. We call $N_w(\Phi) = \max_{\ell=0,\dots,L}\{N_\ell\}$ the width of the network $\Phi$.*

- *A neural network architecture $\mathbb{A}$ with input dimension $d$ and $L$ layers is a neural network*

$$\mathbb{A} = \big((\boldsymbol{W}^1, \boldsymbol{b}^1), \dots, (\boldsymbol{W}^L, \boldsymbol{b}^L)\big),$$

*where elements of $\boldsymbol{W}^\ell$ and $\boldsymbol{b}^\ell$, $\ell = 1, \dots, L$, are in $\{0, 1\}$.*

Since we only are interested in neural network with scalar output, $\boldsymbol{b}^L$ is a constant. However for consistent notation, we still use bold letter.

A graph associated to a deep neural network $\Phi$ defined in Definition 4.1 is a graph consisting of $|\boldsymbol{N}(\Phi)|_1$ nodes and $\sum_{\ell=1}^{L}|\boldsymbol{W}^\ell|_0$ edges. $|\boldsymbol{N}(\Phi)|_1$ nodes are placed in $L+1$ layers which are numbered from 0 to $L$. The $\ell$th layer has $N_\ell$ nodes which are numbered from 1 to $N_\ell$. If $w_{i,j}^\ell \neq 0$, then there is an edge connecting the node $j$ in the layer $\ell-1$ to the node $i$ in the layer $\ell$. See Figure 2 for an illustration of a graph associated to a deep neural network.

**Definition 4.2** *Given $L \in \mathbb{N}$, $L \geq 2$, and a deep neural network architecture $\mathbb{A} = \big((\overline{\boldsymbol{W}}^1, \overline{\boldsymbol{b}}^1), \ldots, (\overline{\boldsymbol{W}}^L, \overline{\boldsymbol{b}}^L)\big)$. We say that a neural network $\Phi = \big((\boldsymbol{W}^1, \boldsymbol{b}^1), \ldots, (\boldsymbol{W}^L, \boldsymbol{b}^L)\big)$ has architecture $\mathbb{A}$ if*

- $\boldsymbol{N}(\Phi) = \boldsymbol{N}(\mathbb{A})$
- *$\overline{w}_{i,j}^\ell = 0$ implies $w_{i,j}^\ell = 0$, $\overline{b}_i^\ell = 0$ implies $b_i^\ell = 0$ for all $i = 1, \ldots, N_\ell$, $j = 1, \ldots, N_{\ell-1}$, and $\ell = 1, \ldots, L$. Here $\overline{w}_{i,j}^\ell$ are entries of $\overline{\boldsymbol{W}}^\ell$ and $\overline{b}_i^\ell$ are elements of $\overline{\boldsymbol{b}}^\ell$, $\ell = 1, \ldots, L$.*

*For a given deep neural network $\Phi = \big((\boldsymbol{W}^1, \boldsymbol{b}^1), \ldots, (\boldsymbol{W}^L, \boldsymbol{b}^L)\big)$, there exists a unique deep neural network architecture $\mathbb{A} = \big((\overline{\boldsymbol{W}}^1, \overline{\boldsymbol{b}}^1), \ldots, (\overline{\boldsymbol{W}}^L, \overline{\boldsymbol{b}}^L)\big)$ such that*

- $\boldsymbol{N}(\Phi) = \boldsymbol{N}(\mathbb{A})$
- *$\overline{w}_{i,j}^\ell = 0 \iff w_{i,j}^\ell = 0$, $\overline{b}_i^\ell = 0 \iff b_i^\ell = 0$ for all $i = 1, \ldots, N_\ell$, $j = 1, \ldots, N_{\ell-1}$, and $\ell = 1, \ldots, L$.*

*We call this architecture $\mathbb{A}$ the minimal architecture of $\Phi$ (this definition is proper in the sense that any architecture of $\Phi$ is also an architecture of $\mathbb{A}$.)*

A deep neural network associated with an activation function. In this paper we focus on ReLU (Rectified Linear Unit) activation function defined by $\sigma(t) := \max\{t, 0\}, t \in \mathbb{R}$. We will use the notation $\sigma(\boldsymbol{x}) := (\sigma(x_1), \ldots, \sigma(x_d))$ for $\boldsymbol{x} \in \mathbb{R}^d$.

**Definition 4.3** *A deep ReLU neural network with input dimension $d$ and $L$ layers is a neural network*

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

*in which the following computation scheme is implemented*

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

*We call $\boldsymbol{z}^0$ the input and $\mathcal{N}(\Phi, \boldsymbol{x}) := \boldsymbol{z}^L$ the output of $\Phi$.*

Several deep ReLU neural networks can be combined into a larger deep ReLU neural network whose output is a linear combination of outputs of sub-networks as in the following lemma. This combination is called parallelization. For other combinations, such as concatenation, we refer to [18, Section 2] or [12]. Note that our parallelization construction differs slightly from that in [18] since we only consider deep ReLU neural networks with scalar output.

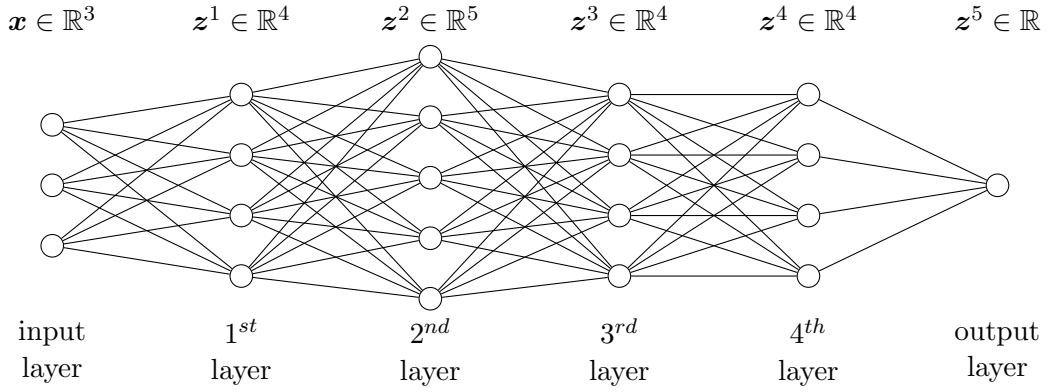


Figure 2: The graph associated to a deep neural network with input dimension 3 and 5 layers

**Lemma 4.1** *Let $N \in \mathbb{N}$, $\Omega \subset \mathbb{R}^d$ be a bounded set, $\lambda_j \in \mathbb{R}$, $j = 1, \ldots, N$. Let $\Phi_j$, $j = 1, \ldots, N$ be deep neural networks with input dimension $d$, $L_j$ layers, and $W_j$ number of weights respectively. Then there is a deep neural network denoted by $\Phi$ such that*

$$\mathcal{N}(\Phi, \boldsymbol{x}) = \sum_{j=1}^{N} \lambda_j \mathcal{N}(\Phi_j, \boldsymbol{x}), \quad \boldsymbol{x} \in \Omega,$$

*with $L(\Phi) = \max_{j=1,\ldots,N}\{L_j\}$ and $W(\Phi) = \sum_{j=1}^{N} W_j + \sum_{j:L_j<L}(L - L_j + 2)$.*

*Proof.* We prove first for $N = 2$. Without loss of generality we assume that $L_1 \leq L_2$ and

$$\Phi_1 = \big((\boldsymbol{W}_1^1, \boldsymbol{b}_1^1), \ldots, (\boldsymbol{W}_1^{L_1}, \boldsymbol{b}_1^{L_1})\big); \qquad \Phi_2 = \big((\boldsymbol{W}_2^1, \boldsymbol{b}_2^1), \ldots, (\boldsymbol{W}_2^{L_2}, \boldsymbol{b}_2^{L_2})\big).$$

If $L = L_1 = L_2$, then we can choose

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

where

$$\boldsymbol{W}^1 = \begin{bmatrix} \boldsymbol{W}_1^1 \\ \boldsymbol{W}_2^1 \end{bmatrix}, \ \boldsymbol{W}^\ell = \begin{bmatrix} \boldsymbol{W}_1^\ell & 0 \\ 0 & \boldsymbol{W}_2^\ell \end{bmatrix}, \ \ell = 2, \ldots, L-1, \ \boldsymbol{W}^L = \begin{bmatrix} \lambda_1 \boldsymbol{W}_1^L & \lambda_2 \boldsymbol{W}_2^L \end{bmatrix}$$

and

$$\boldsymbol{b}^1 = \begin{bmatrix} \boldsymbol{b}_1^1 \\ \boldsymbol{b}_2^1 \end{bmatrix}, \ \ell = 1, \ldots, L_1, \quad \boldsymbol{b}^L = \lambda_1 \boldsymbol{b}_1^L + \lambda_2 \boldsymbol{b}_2^L.$$

In this case we have $W(\Phi) \leq W_1 + W_2$. If $L_1 < L_2$ we construct a network $\tilde{\Phi}_1$ with output $\mathcal{N}(\tilde{\Phi}, \boldsymbol{x}) = \mathcal{N}(\Phi, \boldsymbol{x})$ and having $L_2$ layers. The strategy here is to modify the network $\Phi_1$ by making its output layer satisfying $\mathcal{N}(\Phi, \boldsymbol{x}) + C \geq 0$, for some constant $C$, so that this value does not change when we apply function $\sigma$ for layers from $L_1 + 1$ to $L_2$. For this we put $M_1 = \sup_{\boldsymbol{x}\in\Omega} |\mathcal{N}(\Phi_1, \boldsymbol{x})|$. Note that $\mathcal{N}(\Phi_1, \cdot)$ is a continuous function on $\Omega$ hence $M_1 < \infty$. The network $\tilde{\Phi}_1$ is

$$\big((\boldsymbol{W}_1^1, \boldsymbol{b}_1^1), \ldots, (\boldsymbol{W}_1^{L_1-1}, \boldsymbol{b}_1^{L_1-1}), (\boldsymbol{W}_1^{L_1}, \boldsymbol{b}_1^{L_1} + M_1), (1, 0), \ldots, (1, 0), (1, -M_1)\big).$$

Hence $W(\tilde{\Phi}_1) \leq W_1 + L_2 - L_1 + 2$. Now following procedure as the case $L_1 = L_2$ with $\Phi_1$ replaced by $\tilde{\Phi}_1$ we obtain the assertion when $N = 2$. The case $N > 2$ is extended in a similar manner. ∎

An illustration of parallelization of neural networks is given in Figure 3.

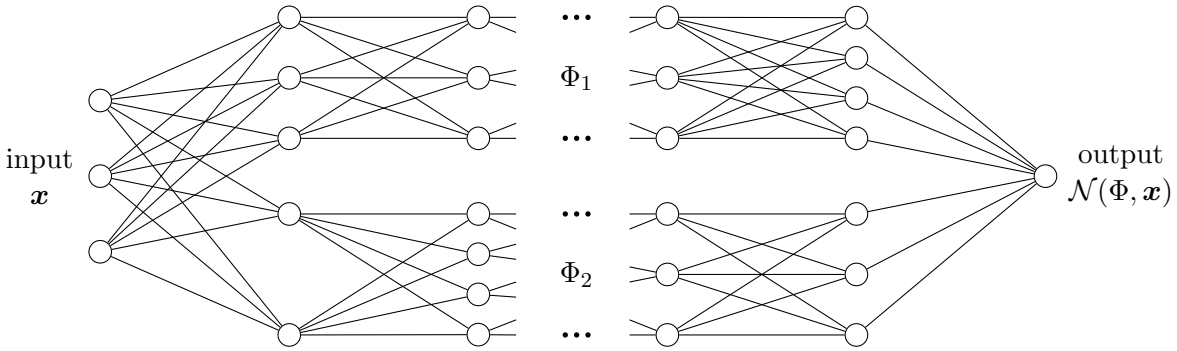


Figure 3: The graph associated to parallelization of two neural networks

Let us introduce a concept of special deep neural network borrowed from [12]. It is quite useful in construction deep ReLU neural networks with a fixed width, whose outputs are able to approximate multivariate functions.

**Definition 4.4** *A special deep neural network with input dimension $d$ and depth $L$ (and a given activation function) can be defined as follows. In each hidden layer a special role is reserved for $d$ first (top) nodes and the last (bottom) node. The top $d$ nodes and the bottom node are free of the activation function, other nodes in each hidden layer have the activation function. The top $d$ nodes are used to simply copy the input $\boldsymbol{x}$. The $d$ parallel concatenations of all these top nodes can be viewed as special channels that skip computation altogether and just carry $\boldsymbol{x}$ forward. They are called the source channels. The bottom node in each hidden lawyer is used to collect intermediate outputs by addition. The concatenation of all these nodes is called collation channel. This channel never feeds forward into subsequent calculation, it only accepts previous calculations.*

An illustration of special deep neural network is given in Figure 4.

**Lemma 4.2** *Let $\Phi$ be a special deep ReLU neural network with input dimension $d$ depth $L$. Then there is a deep ReLU neural network $\Phi'$ such that, $N_w(\Phi') = N_w(\Phi)$, $L(\Phi') = L$, and $\mathcal{N}(\Phi', \boldsymbol{x}) = \mathcal{N}(\Phi, \boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{I}^d$.*

*Proof.* The proof follows from [12, Remark 3.1]. First note, that the input $\boldsymbol{x}$ belongs to $\mathbb{I}^d = [0,1]^d$, we have $\boldsymbol{x} = \sigma(\boldsymbol{x})$. For $\ell = 1, \ldots, L$, the bottom node in the $\ell$-th layer collects a continuous piece-wise linear function $g_\ell(\boldsymbol{x})$ on $\mathbb{I}^d$, and the output is $\mathcal{N}(\Phi, \boldsymbol{x}) = \sum_{\ell=1}^{L-1} g_\ell(\boldsymbol{x})$. Thus, there is a constant $c_\ell$ such that $g_\ell(\boldsymbol{x}) + c_\ell \geq 0$ for all $\boldsymbol{x} \in \mathbb{I}^d$. Hence we take $\Phi'$ having the same graph as $\Phi$ but the computation at $\ell$-th nodes in the collation channel is replaced by $\sigma(g_\ell(\boldsymbol{x}) + c_\ell) = g_\ell(\boldsymbol{x}) + c_\ell$, and the output node is

$$\mathcal{N}(\Phi', \boldsymbol{x}) = \sum_{\ell=1}^{L-1} \sigma(g_\ell(\boldsymbol{x}) + c_\ell) - \sum_{\ell=1}^{L-1} c_\ell = \mathcal{N}(\Phi, \boldsymbol{x}).$$

□

We now consider the problem of approximation of functions from $\mathring{U}_\infty^\alpha$ by deep ReLU neural networks. For every $\varepsilon > 0$ and every $f \in \mathring{U}_\infty^\alpha$, we will construct a deep ReLU neural network $\Phi_f$ having an architecture $\mathbb{A}_\varepsilon$ independent of $f$, and the output $\mathcal{N}(\Phi_f, \cdot)$ which approximates $f$ with accuracy $\varepsilon$, and give dimension-dependent upper bounds for the number of weights and the depth of $\Phi_f$. Our strategy is as follows. Based on the result in Theorem 3.3 on approximation of $f$ by the sparse-grid

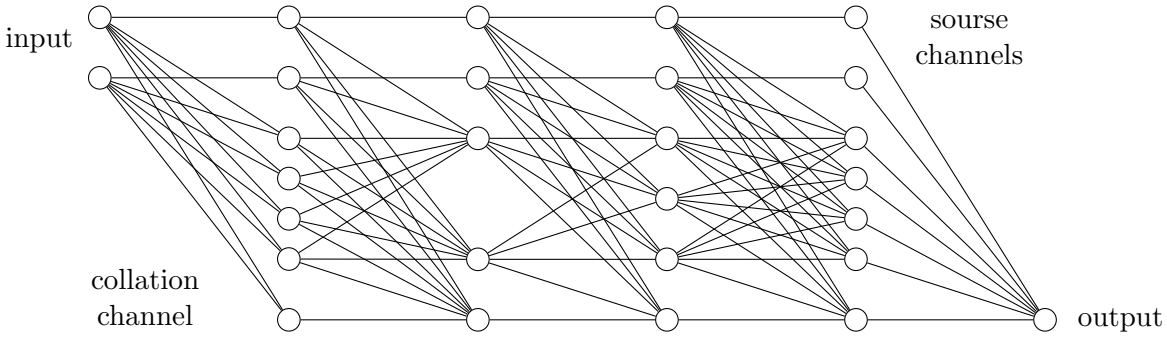


Figure 4: The graph associated to a special neural network with two source channels and 5 layers

sampling sum $R_\beta(m,f)$, we use $R_\beta(m,f)$ as a mediate approximation, and then construct a deep ReLU neural network $\Phi_f$ for approximating this sum by the output $\mathcal{N}(\Phi_f,\cdot)$. Since $R_\beta(m,f)$ is a sum of tensor products of hat functions, first of all we will process the approximation such tensor products by deep ReLU neural networks which can be done based on a result in the following lemma which has been proven in [30, Proposition 2.6] on approximating by deep ReLU neural networks the product of $d$ numbers.

**Lemma 4.3** *For any $\delta\in(0,1)$, $d\in\mathbb{N}$, $d\ge 2$, there exists a deep ReLU neural network $\Phi_P$ such that*

$$\sup_{\boldsymbol{x}\in[-1,1]^d}\left|\prod_{i=1}^d x_i-\mathcal{N}(\Phi_P,\boldsymbol{x})\right|\le\delta,$$

*and*

$$\operatorname*{ess\,sup}_{\boldsymbol{x}\in[-1,1]^d}\sup_{j=1,\dots,d}\left|\frac{\partial}{\partial x_j}\prod_{i=1}^d x_i-\frac{\partial}{\partial x_j}\mathcal{N}(\Phi_P,\boldsymbol{x})\right|\le\delta,$$

*where $\frac{\partial}{\partial x_j}$ denotes a weak partial derivative. Furthermore, there exists a constant $C$ independent of $\delta\in(0,1)$ and $d\in\mathbb{N}$ such that*

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

*Moreover, if $x_j=0$ for some $j\in[d]$, then $\mathcal{N}(\Phi_P,\boldsymbol{x})=0$.*

The last statement $\mathcal{N}(\Phi_P,\boldsymbol{x})=0$ ($d=2$) when $x_1\cdot x_2=0$ was proved in [34, Proposition 3.1], see also [18, Proposition C.2] and [29, Proposition 4.1]. But this implies that the statement also holds for general $d$ since the network $\Phi_P$ is constructed as an binary tree of the network $\Phi_P$ when $d=2$. Inspecting the proof of Lemma 4.3, see [30, Proposition 2.6] and [34, Proposition 3.3] we also find that when $\boldsymbol{x}\in\mathbb{I}^d$ it holds $N_w(\Phi_P)\le 12d$. The above lemma immediately yields the following lemma on approximating by deep ReLU neural networks tensor products of $d$ hat functions.

**Lemma 4.4** *For any dimension $d\ge 2$, $0<\delta<1$ and for the $d$-variate hat functions $\varphi_{\boldsymbol{k},\boldsymbol{s}}$, $\boldsymbol{k}\in\mathbb{N}_0^d$, $\boldsymbol{s}\in Z^d(\boldsymbol{k})$, defined as in (2.3), there is a deep ReLU neural network $\Phi_{\boldsymbol{k},\boldsymbol{s}}$ such that $\mathcal{N}(\Phi_{\boldsymbol{k},\boldsymbol{s}},\cdot)$ approximates $\varphi_{\boldsymbol{k},\boldsymbol{s}}$ with accuracy $\delta$, and*

$$N_w(\Phi_{\boldsymbol{k},\boldsymbol{s}})\le Cd,\quad W(\Phi_{\boldsymbol{k},\boldsymbol{s}})\le Cd\log(d\delta^{-1}),\quad\text{and}\quad L(\Phi_{\boldsymbol{k},\boldsymbol{s}})\le C\log d\log(d\delta^{-1})\,.$$

*Moreover, $\operatorname{supp}(\mathcal{N}(\Phi_{\boldsymbol{k},\boldsymbol{s}},\cdot))\subset\operatorname{supp}\varphi_{\boldsymbol{k},\boldsymbol{s}}$ and*

$$\operatorname*{ess\,sup}_{\boldsymbol{x}\in\mathbb{I}^d}\left|\frac{\partial}{\partial x_j}\varphi_{\boldsymbol{k},\boldsymbol{s}}(\boldsymbol{x})-\frac{\partial}{\partial x_j}\mathcal{N}(\Phi_{\boldsymbol{k},\boldsymbol{s}},\boldsymbol{x})\right|\le 2^{k_j+1}\delta\,.$$

*Proof.* Indeed, we write

$$y_i := \varphi_{k_i,s_i}(x_i) = \sigma\big(1 - \sigma\big(2^{k_i+1}x_i - 2(s_i+1)\big) - \sigma\big(2(s_i+1) - 2^{k_i+1}x_i\big)\big).$$

Let $\Phi_P$ be the deep ReLU neural network in Lemma 4.3 and $\boldsymbol{y} = (y_1, \dots, y_d)$ be the inputs of $\Phi_P$. Then we obtain a deep ReLU neural network denoted by $\Phi_{\boldsymbol{k},\boldsymbol{s}}$. We have

$$\sup_{\boldsymbol{x}\in\mathbb{I}^d}\left|\prod_{i=1}^d \varphi_{k_i,s_i}(x_i) - \mathcal{N}(\Phi_{\boldsymbol{k},\boldsymbol{s}},\boldsymbol{x})\right| = \sup_{\boldsymbol{y}\in\mathbb{I}^d}\left|\prod_{i=1}^d y_i - \mathcal{N}(\Phi_P,\boldsymbol{y})\right| \le \delta$$

and

$$\operatorname*{ess\,sup}_{\boldsymbol{x}\in\mathbb{I}^d}\left|\frac{\partial}{\partial x_j}\varphi_{\boldsymbol{k},\boldsymbol{s}}(\boldsymbol{x}) - \frac{\partial}{\partial x_j}\mathcal{N}(\Phi_{\boldsymbol{k},\boldsymbol{s}},\boldsymbol{x})\right| = \operatorname*{ess\,sup}_{\boldsymbol{y}\in\mathbb{I}^d}\left|\left(\frac{\partial}{\partial y_j}\prod_{i=1}^d y_i - \frac{\partial}{\partial y_j}\mathcal{N}(\Phi_P,\boldsymbol{y})\right)\frac{\mathrm{d}y_j}{\mathrm{d}x_j}\right| \le 2^{k_j+1}\delta\,.$$

Moreover, we have

$$L(\Phi_{\boldsymbol{k},\boldsymbol{s}}) = L(\Phi_P) + 2 \quad \text{and} \quad W(\Phi_{\boldsymbol{k},\boldsymbol{s}}) \le W(\Phi_P) + 7d.$$

Now, from Lemma 4.3 we obtain the desired result. ∎

We now in position to formulate and prove the following theorem.

**Theorem 4.5** *Let $d \ge 2$, $1 < \alpha \le 2$, $\beta \ge \alpha$ and $1 \le p \le \infty$. Let*

$$\varepsilon_0 = \min\left\{1, \frac{d}{2^{\alpha d}(1-2^{1-\alpha})}, \frac{K_1 d^2}{(p+1)^{d/p}2^{(\alpha+1)d}\big(1-2^{-\frac{\beta-\alpha}{\beta-1}}\big)^d}\right\},$$

*where $K_1$ is the constant given in Theorem 3.3.*

*Then for every $\varepsilon \in (0,\varepsilon_0)$ there exists a deep neural network architecture $\mathbb{A}_\varepsilon$ with the following property. For every $f \in \mathring{U}^\alpha_\infty$ there exist a deep ReLU neural network $\Phi_f$ having the architecture $\mathbb{A}_\varepsilon$, and positive constants $K_3 = K_3(\alpha)$, $K_4 = K_4(\alpha,\beta)$, such that*

$$\|f - \mathcal{N}(\Phi_f,\cdot)\|_{W_0^{1,p}} \le \varepsilon, \tag{4.1}$$

*and there hold the estimates*

$$L(\mathbb{A}_\varepsilon) \le K_3 \log d \log(\varepsilon^{-1}) \quad \text{and} \quad W(\mathbb{A}_\varepsilon) \le K_4 B_2^{-d}(\varepsilon^{-1})^{\frac{1}{\alpha-1}}\log(\varepsilon^{-1}),$$

*where*

$$B_2 = B_2(d,\alpha,\beta) := \big(1-2^{-\frac{1}{\beta-1}}\big)\left(\frac{(p+1)^{\frac{1}{p}}2^{(\alpha+1)}\big(1-2^{-\frac{\beta-\alpha}{\beta-1}}\big)}{d^{\frac{2\alpha}{d}}}\right)^{\frac{1}{\alpha-1}}. \tag{4.2}$$

*Proof.* Let $f \in \mathring{U}^\alpha_\infty$. For $\varepsilon \in (0,\varepsilon_0)$ we take

$$m = \left\lceil \frac{1}{\alpha-1}\log\left(\frac{2K_1 d^2\varepsilon^{-1}}{(p+1)^{d/p}2^{(\alpha+1)d}\big(1-2^{-\frac{\beta-\alpha}{\beta-1}}\big)^d}\right)\right\rceil. \tag{4.3}$$

Let $(\boldsymbol{k},\boldsymbol{s}) \in D^d_\beta(m)$ and $\Phi_{\boldsymbol{k},\boldsymbol{s}}$ be the deep ReLU neural network obtained in Lemma 4.4 such that $\mathcal{N}(\Phi_{\boldsymbol{k},\boldsymbol{s}},\cdot)$ approximates $\varphi_{\boldsymbol{k},\boldsymbol{s}}$ with accuracy $\delta \in (0,1)$ which is chosen later. Let $\Phi_f$ be the deep ReLU neural network obtained by parallelization as in Lemma 4.1 with the output

$$\mathcal{N}(\Phi_f,\boldsymbol{x}) = \sum_{\boldsymbol{k}\in\Delta^d_\beta(m)}\sum_{\boldsymbol{s}\in Z^d(\boldsymbol{k})}\lambda_{\boldsymbol{k},\boldsymbol{s}}(f)\mathcal{N}(\Phi_{\boldsymbol{k},\boldsymbol{s}},\boldsymbol{x}), \quad \boldsymbol{x}\in\mathbb{I}^d.$$

Then we can write

$$\|f-\mathcal{N}(\Phi_f,\cdot)\|_{W_0^{1,p}} \le \|f-R_\beta(m,f)\|_{W_0^{1,p}} + \|R_\beta(m,f)-\mathcal{N}(\Phi_f,\cdot)\|_{W_0^{1,p}}, \tag{4.4}$$

where $R_\beta(m,f)$ is the operator given in (3.1). With the choice of $m$ as in (4.3) we get from Theorem 3.1

$$\|f-R_\beta(m,f)\|_{W_0^{1,p}} \le K_1 \frac{d^2 2^{-m(\alpha-1)}}{(p+1)^{\frac{d}{p}} 2^{(\alpha+1)d}\big(1-2^{-\frac{\beta-\alpha}{\beta-1}}\big)^d} \le \varepsilon/2. \tag{4.5}$$

Let us estimate the norm $\|R_\beta(m,f)-\mathcal{N}(\Phi_f,\cdot)\|_{W_0^{1,p}}$. Since $\operatorname{supp}\mathcal{N}(\Phi_{\boldsymbol{k},\boldsymbol{s}},\cdot)\subset \operatorname{supp}\varphi_{\boldsymbol{k},\boldsymbol{s}}$ we have

$$\begin{aligned}
\|R_\beta(m,f)-\mathcal{N}(\Phi_f,\cdot)\|_{W_0^{1,p}} &\le \sum_{\boldsymbol{k}\in\Delta_\beta^d(m)} \Big\| \sum_{\boldsymbol{s}\in Z^d(\boldsymbol{k})} \lambda_{\boldsymbol{k},\boldsymbol{s}}(f)\big(\varphi_{\boldsymbol{k},\boldsymbol{s}}-\mathcal{N}(\Phi_{\boldsymbol{k},\boldsymbol{s}},\cdot)\big)\Big\|_{W_0^{1,p}} \\
&= \sum_{\boldsymbol{k}\in\Delta_\beta^d(m)} \left( \sum_{j=1}^d \int_{\mathbb{I}^d} \Big| \sum_{\boldsymbol{s}\in Z^d(\boldsymbol{k})} \lambda_{\boldsymbol{k},\boldsymbol{s}}(f)\Big(\frac{\partial}{\partial x_j}\varphi_{\boldsymbol{k},\boldsymbol{s}}(\boldsymbol{x}) - \frac{\partial}{\partial x_j}\mathcal{N}(\Phi_{\boldsymbol{k},\boldsymbol{s}},\boldsymbol{x})\Big)\Big|^p \,\mathrm{d}\boldsymbol{x}\right)^{1/p} \\
&= \sum_{\boldsymbol{k}\in\Delta_\beta^d(m)} \left( \sum_{j=1}^d \sum_{\boldsymbol{s}\in Z^d(\boldsymbol{k})} |\lambda_{\boldsymbol{k},\boldsymbol{s}}(f)|^p \int_{\mathbb{I}^d} \Big|\frac{\partial}{\partial x_j}\varphi_{\boldsymbol{k},\boldsymbol{s}}(\boldsymbol{x}) - \frac{\partial}{\partial x_j}\mathcal{N}(\Phi_{\boldsymbol{k},\boldsymbol{s}},\boldsymbol{x})\Big|^p \,\mathrm{d}\boldsymbol{x}\right)^{1/p}.
\end{aligned}$$

Using Lemma 4.4 and estimate (2.5) we get

$$\begin{aligned}
\|R_\beta(m,f)-\mathcal{N}(\Phi_f,\cdot)\|_{W_0^{1,p}} &\le \sum_{\boldsymbol{k}\in\Delta_\beta^d(m)} \sup_{\boldsymbol{s}\in Z^d(\boldsymbol{k})} |\lambda_{\boldsymbol{k},\boldsymbol{s}}(f)| \left( \sum_{j=1}^d \sum_{\boldsymbol{s}\in Z^d(\boldsymbol{k})} 2^{-|\boldsymbol{k}|_1 2^{p(k_j+1)}} \right)^{1/p} \delta \\
&\le 2^{-(\alpha+1)d} \sum_{\boldsymbol{k}\in\Delta_\beta^d(m)} 2^{-|\boldsymbol{k}|_1\alpha} \left( \sum_{j=1}^d 2^{p(k_j+1)} \right)^{1/p} \delta \\
&\le 2^{-(\alpha+1)d} \sum_{\ell=0}^\infty 2^{-\ell\alpha} \sum_{|\boldsymbol{k}|_1=\ell} \left( \sum_{j=1}^d 2^{p(k_j+1)} \right)^{1/p} \delta.
\end{aligned}$$

Now Lemma 3.2 leads to

$$\|R_\beta(m,f)-\mathcal{N}(\Phi_f,\cdot)\|_{W_0^{1,p}} \le d2^d 2^{-(\alpha+1)d}\delta \sum_{\ell=0}^\infty 2^{-\ell(\alpha-1)} \le \frac{d2^{-\alpha d}}{1-2^{1-\alpha}}\delta.$$

Define $\delta=\delta(\varepsilon) := \frac{1-2^{1-\alpha}}{d2^{-\alpha d}}\frac{\varepsilon}{2}$. Since $\varepsilon < \frac{2d}{2^{\alpha d}(1-2^{1-\alpha})}$ we get $\delta<1$. This choice of $\delta$ gives

$$\|R_\beta(m,f)-\mathcal{N}(\Phi_f,\cdot)\|_{W_0^{1,p}} \le \varepsilon/2$$

which together with (4.4) and (4.5) proves (4.1).

We now prove the bounds for the depth and the number of weights of $\Phi_f$. From Lemmata 4.1 and 3.2 we have

$$\begin{aligned}
L(\Phi_f) &= \max_{(\boldsymbol{k},\boldsymbol{s})\in D_\beta^d(m)} L(\Phi_{\boldsymbol{k},\boldsymbol{s}}) \le C\log d\log(d\delta^{-1}) \\
&\le C\log d\log\left(\frac{2d^2\varepsilon^{-1}}{2^{\alpha d}(1-2^{1-\alpha})}\right) \le K_3\log d\log(\varepsilon^{-1})
\end{aligned} \tag{4.6}$$

for some positive constant $K_3 = K_3(\alpha)$, and

$$\begin{aligned} W(\Phi_f) &\le \sum_{(\boldsymbol{k},\boldsymbol{s})\in D_\beta^d(m)} W(\Phi_{\boldsymbol{k},\boldsymbol{s}}) + \sum_{(\boldsymbol{k},\boldsymbol{s}):L(\Phi_{\boldsymbol{k},\boldsymbol{s}})<L(\Phi_f)} \big(L(\Phi_f) - L(\Phi_{\boldsymbol{k},\boldsymbol{s}}) + 2\big) \\ &\le \sum_{(\boldsymbol{k},\boldsymbol{s})\in D_\beta^d(m)} W(\Phi_{\boldsymbol{k},\boldsymbol{s}}) + \sum_{(\boldsymbol{k},\boldsymbol{s})\in D_\beta^d(m)} \Big( \max_{(\boldsymbol{k},\boldsymbol{s})\in D_\beta^d(m)} W(\Phi_{\boldsymbol{k},\boldsymbol{s}}) + 2 \Big) \\ &\le 2|D_\beta^d(m)| \max_{(\boldsymbol{k},\boldsymbol{s})\in D_\beta^d(m)} \big(W(\Phi_{\boldsymbol{k},\boldsymbol{s}}) + 1\big) \le \frac{2\beta}{\beta-1} \frac{d2^m}{\big(1-2^{-\frac{1}{\beta-1}}\big)^d} \big(Cd\log(d\delta^{-1}) + 1\big). \end{aligned} \tag{4.7}$$

Similar to (3.11) we have

$$\begin{aligned} d^2 2^m \big(1-2^{-\frac{1}{\beta-1}}\big)^{-d} &\le 2d^2 \big(1-2^{-\frac{1}{\beta-1}}\big)^{-d} \left( \frac{2K_1 d^2 \varepsilon^{-1}}{(p+1)^{\frac{d}{p}} 2^{(\alpha+1)d} \big(1-2^{-\frac{\beta-\alpha}{\beta-1}}\big)^d} \right)^{\frac{1}{\alpha-1}} \\ &\le 2(2K_1)^{\frac{1}{\alpha-1}} \big(1-2^{-\frac{1}{\beta-1}}\big)^{-d} \left( \frac{d^{\frac{2\alpha}{d}}}{(p+1)^{\frac{1}{p}} 2^{(\alpha+1)} \big(1-2^{-\frac{\beta-\alpha}{\beta-1}}\big)} \right)^{\frac{d}{\alpha-1}} (\varepsilon^{-1})^{\frac{1}{\alpha-1}}. \end{aligned}$$

Inserting this into (4.7) we find

$$W(\Phi_f) \le K_4 B_2^{-d} (\varepsilon^{-1})^{\frac{1}{\alpha-1}} (\log \varepsilon^{-1})$$

with $B_2$ given in (4.2) and some positive constant $K_4$ depending on $\alpha$ and $\beta$.

To complete the proof it is sufficient to notice that $\Phi_f$ has the architecture $\mathbb{A}_\varepsilon$ (independent of $f$) which is defined as the minimal architecture of the deep ReLU neural network $\Phi$ obtained by parallelization as in Lemma 4.1 with the output

$$\mathcal{N}(\Phi, \boldsymbol{x}) = \sum_{\boldsymbol{k}\in\Delta_\beta^d(m)} \sum_{\boldsymbol{s}\in Z^d(\boldsymbol{k})} \mathcal{N}(\Phi_{\boldsymbol{k},\boldsymbol{s}}, \boldsymbol{x}), \quad \boldsymbol{x}\in\mathbb{I}^d.$$

□

**Remark 4.6** *Since $d^{2\alpha/d}$ tends to 1 when $d\to\infty$, if $\alpha$ and $p$ satisfy (3.9), $\beta$ satisfies (3.10), then there are $d(\alpha,\beta,p)\in\mathbb{N}$ and $B(\alpha,\beta,p)>1$ such that $B_2 \ge B(\alpha,\beta,p) > 1$ for all $d \ge d(\alpha,\beta,p)$. In this case, the approximation error is going to 0 as fast as the exponent $B_2^{-d}$ when $d$ tending to $\infty$. See also Remark 3.4.*

For $f\in \mathring{U}_\infty^\alpha$ and $\varepsilon\in(0,\varepsilon_0)$, we observe that the width of the deep ReLU neural network $\Phi_f$ constructed in Theorem 4.5 may depend on $\varepsilon$. To construct a deep neural network $\Phi'_f$ with the same output that has width independent of $\varepsilon$ we can concatenate the deep ReLU networks $\Phi_{\boldsymbol{k},\boldsymbol{s}}$, $(\boldsymbol{k},\boldsymbol{s})\in D_\beta^d(m)$, with the help of special deep ReLU networks.

**Theorem 4.7** *Under the assumptions and notations of Theorem 4.5, for every $\varepsilon\in(0,\varepsilon_0)$ there exists a deep neural network architecture $\mathbb{A}_\varepsilon^*$ with the following property. For every $f\in\mathring{U}_\infty^\alpha$ there exist a deep ReLU neural network $\Phi_f^*$ having the architecture $\mathbb{A}_\varepsilon^*$, and positive constants $K_5$ and $K_6 = K_6(\alpha,\beta)$ such that*

$$\|f - \mathcal{N}(\Phi_f^*,\cdot)\|_{W_0^{1,p}} \le \varepsilon, \tag{4.8}$$

*and there hold the estimates*

$$N_w(\mathbb{A}_\varepsilon^*) \le K_5 d \quad \text{and} \quad L(\mathbb{A}_\varepsilon^*) \le K_6 B_2^{-d} (\varepsilon^{-1})^{\frac{1}{\alpha-1}} \log(\varepsilon^{-1}). \tag{4.9}$$

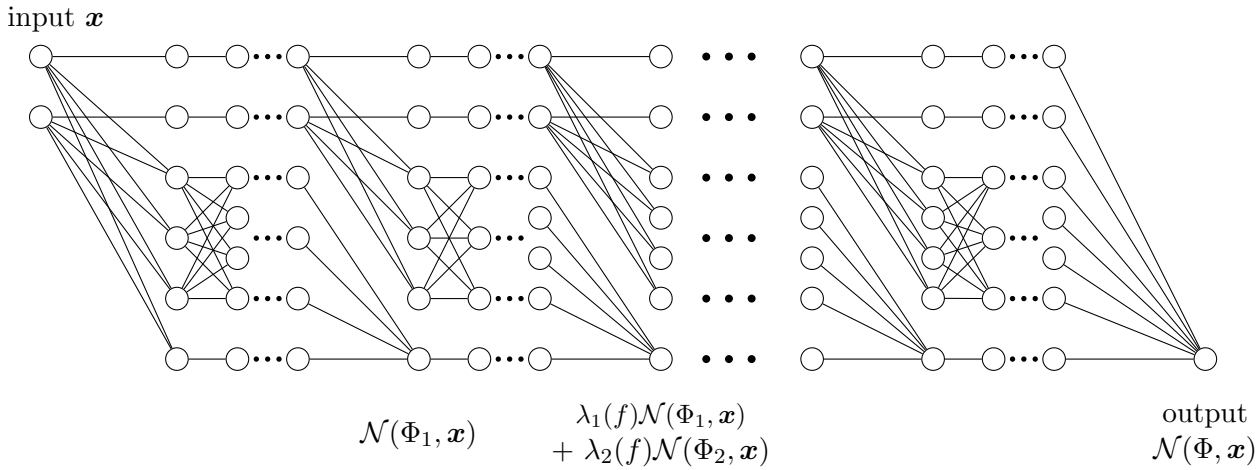


Figure 5: The graph associated to concatenation of neural networks $\Phi_{\boldsymbol{k},\boldsymbol{s}}$, $(\boldsymbol{k}, \boldsymbol{s}) \in D_\beta^d(m)$, $(d = 2)$

*Proof.* Consider the special deep ReLU neural network $\Phi'_f$ as in Figure 5 $(d = 2)$. We number the set $\{(\boldsymbol{k}, \boldsymbol{s}) \in D_\beta^d(m)\}$ from 1 to $J$, where $J = |D_\beta^d(m)|$. The $d$ source channels carry $\boldsymbol{x} \in \mathbb{I}^d$ forward so that it is the input of all networks $\Phi_\ell$, $\ell = 1, \ldots, J$. The $\big(\sum_{j=1}^{\ell} L(\Phi_j)\big)$th node in the collation channel stores the partial sum $\sum_{j=1}^{\ell} \lambda_j(f)\mathcal{N}(\Phi_j, \boldsymbol{x})$ of the outputs of $\Phi_\ell$, $\ell = 1, \ldots, J$. Hence,

$$\mathcal{N}(\Phi'_f, \boldsymbol{x}) = \mathcal{N}(\Phi_f, \boldsymbol{x}) = \sum_{j=1}^{J} \lambda_j(f)\mathcal{N}(\Phi_j, \boldsymbol{x}), \tag{4.10}$$

where $\Phi_f$ is the deep ReLU neural network in Theorem 4.5. From Lemma 4.4 we can find an absolute positive constant $K_5$ so that $N_w(\Phi'_f) \le K_5 d$ and a positive constant $K_6 = K_6(\alpha, \beta)$ such that

$$L(\Phi'_f) \le \sum_{j=1}^{J} L(\Phi_j) \le C|D_\beta^d(m)| \log d \log(d\delta^{-1}) \le K_6 B_2^{-d} (\varepsilon^{-1})^{\frac{1}{\alpha-1}} \log(\varepsilon^{-1}),$$

where the last inequality follows from (4.6) and (4.7). Hence, by Lemma 4.2 and (4.10), $\Phi'_f$ generates a deep ReLU neural network $\Phi_f^*$ such that $\mathcal{N}(\Phi_f^*, \boldsymbol{x}) = \mathcal{N}(\Phi_f, \boldsymbol{x})$ and, consequently, there hold (4.8) and

$$N_w(\Phi_f^*) \le K_5 d \quad \text{and} \quad L(\Phi_f^*) \le K_6 B_2^{-d} (\varepsilon^{-1})^{\frac{1}{\alpha-1}} \log(\varepsilon^{-1}).$$

The proof of existence of an architecture $\mathbb{A}_\varepsilon^*$ of $\Phi_f^*$ satisfying (4.9) is similar to the proof of existence of $\mathbb{A}_\varepsilon$ at the end of the proof of Theorem 4.5. ∎

In the case when the error is measured in the norm of the space $W_0^{1,\infty}$, we are able to give dimension-dependent lower bounds for the numbers of weights of deep ReLU networks whose outputs approximate functions from $\mathring{U}_\infty^\alpha$ with a given accuracy. More precisely, we have the following results.

**Theorem 4.8** *Let $d \in \mathbb{N}$, $d \ge 2$, and $1 < \alpha \le 2$. Let $\varepsilon \in (0, 24^{-d})$ and $\mathbb{A}$ be a neural network architecture such that for any $f \in \mathring{U}_\infty^\alpha$, there is a deep ReLU neural network $\Phi_f$ having the architecture $\mathbb{A}$ and*

$$\|f - \mathcal{N}(\Phi_f, \cdot)\|_{W_0^{1,\infty}} \le \varepsilon.$$

*Then there is a positive constant $K_7 = K_7(\alpha)$ such that*

$$W(\mathbb{A}) \geq K_7 24^{-\frac{d}{2(\alpha-1)}} \varepsilon^{-\frac{1}{2(\alpha-1)}}.$$

*If assume in addition that*

$$L(\mathbb{A}) \leq C(\log \varepsilon^{-1})^{\lambda}$$

*for some constant $C > 0$ and $\lambda \geq 0$, then there exists a constant $K_8 = K_8(\alpha) > 0$ such that*

$$W(\mathbb{A}) \geq K_8 24^{-\frac{d}{\alpha-1}} \varepsilon^{-\frac{1}{\alpha-1}} (\log \varepsilon^{-1})^{-2\lambda-1}.$$

Some non-dimension-dependent lower bounds have been obtained in [42, 18] for isotropic Sobolev spaces. In order to prove the above theorem we develop some techniques in [42, 18] which are relied on upper bounds for VC-dimension of ReLU networks with Boolean outputs [1]. We start with recalling a definition of VC-dimension.

**Definition 4.9** *Let $H$ be a set of functions $h : X \to \{0, 1\}$ for some set $X$. Then the VC-dimension of $H$, denoted by $\mathrm{VCdim}(H)$, is defined as the supremum of all number $m$ that there exist $x^1, \ldots, x^m \in X$ such that for any sequence $(y_j)_{j=1}^m \in \{0, 1\}^m$ there is a function $h \in H$ with $h(x^j) = y_j$ for $j = 1, \ldots, m$.*

The relation between VC-dimension and number of weights and depth of a neural network architecture is given in the following lemma, see [1, Theorems 8.7 and 8.8].

**Lemma 4.5** *Let $\mathbb{A}$ be a deep neural network architecture. Let $F$ be the class of functions-outputs $f = \mathcal{N}(\Phi, \cdot)$ of all deep ReLU neural networks $\Phi$ having the architecture $\mathbb{A}$. Let $a \in \mathbb{R}$ and $H$ be the class of all functions $h_f : \mathbb{I}^d \to \{0, 1\}$, $f \in F$, defined by threshold:*

$$h_f(\boldsymbol{x}) := \begin{cases} 1 & \text{if } f(\boldsymbol{x}) > a \\ 0 & \text{if } f(\boldsymbol{x}) \leq a. \end{cases}$$

*Then*

$$\mathrm{VCdim}(H) \leq C W(\mathbb{A})^2$$

*for some positive constant $C$. Moreover, there exists a $C' > 0$ such that*

$$\mathrm{VCdim}(H) \leq C' L(\mathbb{A})^2 W(\mathbb{A}) \log(W(\mathbb{A})).$$

The following elementary property of deep ReLU neural networks has been proven in [18, Lemma D.1] which is based on the piece-wise linearity of ReLU activation function.

**Lemma 4.6** *Let $\Phi$ be a deep ReLU neural network, $\boldsymbol{\xi} \in (0, 1)^d$, and $\boldsymbol{\nu} \in \mathbb{R}^d$. Then there exists an open set $G = G(\boldsymbol{\xi}, \boldsymbol{\nu}) \subset (0, 1)^d$ and $\delta = \delta(\boldsymbol{\xi}, \boldsymbol{\nu}) > 0$ such that $\boldsymbol{\xi} + \lambda\delta\boldsymbol{\nu} \in \overline{G}$ for $\lambda \in [0, 1]$ and $\mathcal{N}(\Phi, \cdot)$ is affine on $G$.*

We are now in position to prove Theorem 4.8.

*Proof.* Given $\varepsilon \in (0, 24^{-d})$ and $m = m(\varepsilon) \in \mathbb{N}$ which will be chosen later. Denote $L = L(\mathbb{A})$, $W = W(\mathbb{A})$ the depth and the number of weights of $\mathbb{A}$. We assume that

$$\mathbb{A} = \big((\boldsymbol{W}^1, \boldsymbol{b}^1), \ldots, (\boldsymbol{W}^L, \boldsymbol{b}^L)\big),$$

where $\boldsymbol{W}^\ell$ is an $N_\ell \times N_{\ell-1}$ matrix, and $\boldsymbol{b}^\ell \in \mathbb{R}^{N_\ell}$ with $N_0 = d$, $N_L = 1$, and $N_1, \dots, N_{L-1} \in \mathbb{N}$. Let $\Phi$ be a deep ReLU neural network of the architecture $\mathbb{A}$. For $\boldsymbol{x} = (x_1, \dots, x_d) \in \mathbb{I}^d$ and $\delta > 0$ we put $\bar{\boldsymbol{x}} = \big(x_1 + \frac{2^{-m-2}}{3}, x_2, \dots, x_d\big)$ and define the deep ReLU neural network $\Phi^\delta$ by parallelization construction in Lemma 4.1 with output

$$\mathcal{N}(\Phi^\delta, \boldsymbol{x}) := \frac{\mathcal{N}(\Phi, \bar{\boldsymbol{x}}) - \mathcal{N}(\Phi, \bar{\boldsymbol{x}} - \delta \boldsymbol{e}^1)}{\delta}, \quad \boldsymbol{e}^1 = (1, 0, \dots, 0) \in \mathbb{R}^d.$$

Then $\Phi^\delta$ is a deep ReLU neural network having the architecture

$$\tilde{\mathbb{A}} = \left( \left( \begin{bmatrix} \mathbf{I}_d \\ \mathbf{I}_d \end{bmatrix}, \begin{bmatrix} \boldsymbol{e}^1 \\ \boldsymbol{e}^1 \end{bmatrix} \right), \left( \begin{bmatrix} \boldsymbol{W}^1 & 0 \\ 0 & \boldsymbol{W}^1 \end{bmatrix}, \begin{bmatrix} \boldsymbol{b}^1 \\ \boldsymbol{b}^1 \end{bmatrix} \right), \dots, \left( \begin{bmatrix} \boldsymbol{W}^{L-1} & 0 \\ 0 & \boldsymbol{W}^{L-1} \end{bmatrix}, \begin{bmatrix} \boldsymbol{b}^{L-1} \\ \boldsymbol{b}^{L-1} \end{bmatrix} \right) \left( [\boldsymbol{W}^L, \boldsymbol{W}^L], \boldsymbol{b}^L \right) \right),$$

where $\mathbf{I}_d$ is the identity matrix of size $d$. It is clear that $\tilde{\mathbb{A}}$ has depth and number of weights

$$\tilde{L} := L(\tilde{\mathbb{A}}) = L + 1, \qquad \text{and} \qquad \tilde{W} := W(\tilde{\mathbb{A}}) = 2W + 2d + 1. \tag{4.11}$$

Let $a = a(m, d) \in \mathbb{R}$ be a constant which will be clarified later. For a deep ReLU neural network $\tilde{\Phi}$ having architecture $\tilde{\mathbb{A}}$, we define the function

$$h(\tilde{\Phi}, \boldsymbol{x}) = \begin{cases} 1 & \text{if } \mathcal{N}(\tilde{\Phi}, \boldsymbol{x}) > a/2, \\ 0 & \text{if } \mathcal{N}(\tilde{\Phi}, \boldsymbol{x}) \le a/2, \end{cases}$$

and the set

$$H := H(\tilde{\mathbb{A}}) = \big\{ h(\tilde{\Phi}, \boldsymbol{x}) : \tilde{\Phi} \text{ is a deep ReLU neural network having architecture } \tilde{\mathbb{A}} \big\}.$$

In the following we will show that $\mathrm{VCdim}(H) \ge 2^m$. For this we put

$$\boldsymbol{x}^j = \left( 2^{-m} \Big( j - \frac{3}{4} \Big), \frac{1}{2}, \dots, \frac{1}{2} \right) \in (0, 1)^d, \quad j = 1, \dots, 2^m.$$

It is clear that $\bar{\boldsymbol{x}}^j \in (0, 1)^d$ for $j = 1, \dots, 2^m$. For every $\boldsymbol{y} = (y_1, \dots, y_{2^m}) \in \{0, 1\}^{2^m}$, we define

$$f_{\boldsymbol{y}}(\boldsymbol{x}) = 18^{-d} 2^{-\alpha m} \left( \sum_{j=1}^{2^m} y_j \psi_{m,j}(x_1) \right) \prod_{\ell=2}^{d} \psi_{0,1}(x_\ell),$$

which is similar to $f_m(\boldsymbol{x})$ in the proof of Theorem 3.5. Then it holds $f_{\boldsymbol{y}} \in \mathring{U}^\alpha_\infty$. Moreover, we have

$$\frac{\partial f_{\boldsymbol{y}}}{\partial x_1}(\bar{\boldsymbol{x}}^j) = 18^{-d} 2^{-\alpha m} y_j \psi'_{m,j} \Big( x_1^j + \frac{2^{-m-2}}{3} \Big) \prod_{\ell=2}^{d} \psi(x_\ell^j) = y_j 18^{-d} 2^{-\alpha m} \psi\Big(\frac{1}{2}\Big)^{(d-1)} 2^m \psi'\Big(\frac{1}{3}\Big).$$

From $\psi\big(\frac{1}{2}\big) = M_3\big(\frac{3}{2}\big) = \frac{3}{4}$ and $\psi'\big(\frac{1}{3}\big) = 3M_3'(1) = 3$ we get

$$\frac{\partial f_{\boldsymbol{y}}}{\partial x_1}(\bar{\boldsymbol{x}}^j) = 4 y_j 18^{-d} 2^{-(\alpha-1)m} \Big(\frac{3}{4}\Big)^d = 4 y_j 2^{-(\alpha-1)m} 24^{-d}.$$

Since $f_{\boldsymbol{y}} \in \mathring{U}^\alpha_\infty$, by the assumption, there exists a neural network $\Phi_{\boldsymbol{y}}$ having architecture $\mathbb{A}$ such that

$$\| f_{\boldsymbol{y}} - \mathcal{N}(\Phi_{\boldsymbol{y}}, \cdot) \|_{W_0^{1,\infty}} \le \varepsilon.$$

By Lemma 4.6, for $\bar{\boldsymbol{x}}^j \in (0,1)^d$, there exists an open set $G_j \subset (0,1)^d$ and $\delta_j > 0$ such that $\bar{\boldsymbol{x}}^j + \lambda\delta_j \boldsymbol{e}^1 \in \bar{G}_j$ for $\lambda \in [0,1]$ and $\mathcal{N}(\Phi_{\boldsymbol{y}}, \cdot)$ is affine on $\bar{G}_j$. Let

$$\delta^* = \min_{1\le j\le 2^m} \delta_j > 0.$$

and $B(\bar{\boldsymbol{x}}^j, \theta_j)$ be the open ball centered at $\bar{\boldsymbol{x}}^j$ with radius $\theta_j$. Then we have

$$\mathcal{N}(\Phi_{\boldsymbol{y}}^{\delta^*}, \boldsymbol{x}^j) = \frac{\mathcal{N}(\Phi_{\boldsymbol{y}}, \bar{\boldsymbol{x}}^j) - \mathcal{N}(\Phi_{\boldsymbol{y}}, \bar{\boldsymbol{x}}^j - \delta^* \boldsymbol{e}^1)}{\delta^*} = \frac{\partial \mathcal{N}(\Phi_{\boldsymbol{y}}, \boldsymbol{\xi}^j)}{\partial x_1},$$

for some $\boldsymbol{\xi}^j \in B(\bar{\boldsymbol{x}}^j, \theta_j) \cap G_j$. In case $\bar{\boldsymbol{x}}^j \in G_j$ we can choose $\boldsymbol{\xi}^j = \bar{\boldsymbol{x}}^j$. Define $m$ the largest positive integer such that

$$\varepsilon \le 2^{-(\alpha-1)m} 24^{-d}.$$

We then obtain for $y_j = 1$,

$$\begin{aligned}\mathcal{N}(\Phi_{\boldsymbol{y}}^{\delta^*}, \boldsymbol{x}^j) &= \frac{\partial f_{\boldsymbol{y}}(\bar{\boldsymbol{x}}^j)}{\partial x_1} + \frac{\partial \mathcal{N}(\Phi_{\boldsymbol{y}}, \boldsymbol{\xi}^j)}{\partial x_1} - \frac{\partial f_{\boldsymbol{y}}(\bar{\boldsymbol{x}}^j)}{\partial x_1} \\ &\ge \frac{\partial f_{\boldsymbol{y}}(\bar{\boldsymbol{x}}^j)}{\partial x_1} - \left|\frac{\partial \mathcal{N}(\Phi_{\boldsymbol{y}}, \boldsymbol{\xi}^j)}{\partial x_1} - \frac{\partial f_{\boldsymbol{y}}(\boldsymbol{\xi}^j)}{\partial x_1}\right| - \left|\frac{\partial f_{\boldsymbol{y}}(\boldsymbol{\xi}^j)}{\partial x_1} - \frac{\partial f_{\boldsymbol{y}}(\bar{\boldsymbol{x}}^j)}{\partial x_1}\right|.\end{aligned}$$

Since $\frac{\partial f_{\boldsymbol{y}}(\boldsymbol{x})}{\partial x_1}$ is a continuous function due to $f_{\boldsymbol{y}} \in \mathring{U}_\infty^\alpha$, $\alpha > 1$. Then we can choose $\theta_j$ small enough such that

$$\left|\frac{\partial f_{\boldsymbol{y}}(\boldsymbol{\xi}^j)}{\partial x_1} - \frac{\partial f_{\boldsymbol{y}}(\bar{\boldsymbol{x}}^j)}{\partial x_1}\right| \le \frac{\varepsilon}{2}$$

which implies

$$\mathcal{N}(\Phi_{\boldsymbol{y}}^{\delta^*}, \boldsymbol{x}^j) \ge 4 \cdot 2^{-(\alpha-1)m} 24^{-d} - \frac{3}{2}\varepsilon \ge \frac{5}{2} \cdot 2^{-(\alpha-1)m} 24^{-d}.$$

For $y_j = 0$,

$$\begin{aligned}\left|\mathcal{N}(\Phi_{\boldsymbol{y}}^{\delta^*}, \boldsymbol{x}^j)\right| &= \left|\frac{\partial \mathcal{N}(\Phi_{\boldsymbol{y}}, \boldsymbol{\xi}^j)}{\partial x_1} - \frac{\partial f_{\boldsymbol{y}}(\boldsymbol{\xi}^j)}{\partial x_1} + \frac{\partial f_{\boldsymbol{y}}(\boldsymbol{\xi}^j)}{\partial x_1} - \frac{\partial f_{\boldsymbol{y}}(\bar{\boldsymbol{x}}^j)}{\partial x_1}\right| \\ &\le \left|\frac{\partial \mathcal{N}(\Phi_{\boldsymbol{y}}, \boldsymbol{\xi}^j)}{\partial x_1} - \frac{\partial f_{\boldsymbol{y}}(\boldsymbol{\xi}^j)}{\partial x_1}\right| + \left|\frac{\partial f_{\boldsymbol{y}}(\boldsymbol{\xi}^j)}{\partial x_1} - \frac{\partial f_{\boldsymbol{y}}(\bar{\boldsymbol{x}}^j)}{\partial x_1}\right| \le \frac{3}{2}\varepsilon \le \frac{3}{2} \cdot 2^{-(\alpha-1)m} 24^{-d}.\end{aligned}$$

Putting $a = 4 \cdot 2^{-(\alpha-1)m} 24^{-d}$, we get

$$\begin{cases} \mathcal{N}(\Phi_{\boldsymbol{y}}^{\delta^*}, \boldsymbol{x}^j) > a/2 & \text{if } \ y_j = 1, \\ \mathcal{N}(\Phi_{\boldsymbol{y}}^{\delta^*}, \boldsymbol{x}^j) \le a/2 & \text{if } \ y_j = 0. \end{cases}$$

Then for $\boldsymbol{y} \in \{0,1\}^{2^m}$ the function

$$h(\Phi_{\boldsymbol{y}}^{\delta^*}, \boldsymbol{x}) = \begin{cases} 1 & \text{if } \ \mathcal{N}(\Phi_{\boldsymbol{y}}^{\delta^*}, \boldsymbol{x}) > a/2, \\ 0 & \text{if } \ \mathcal{N}(\Phi_{\boldsymbol{y}}^{\delta^*}, \boldsymbol{x}) \le a/2, \end{cases}$$

belongs to $H$ and satisfies $h(\Phi_{\boldsymbol{y}}^{\delta^*}, \boldsymbol{x}^j) = y_j$. By definition of VC-dimension, see Definition 4.9, we obtain $2^m \le \mathrm{VCdim}(H)$. Moreover, from (4.11) and Lemma 4.5 we have

$$2^m \le \mathrm{VCdim}(H) \le C(2W + 2d + 1)^2.$$

Since $\mathbb{A}$ has input dimension $d$ and depth $L \geq 2$, we find that $2d+1 \leq 2W$. From this and $2^{-(\alpha-1)(m+1)}24^{-d} \leq \varepsilon$ we get

$$C4^2W^2 \geq 2^m \geq \frac{1}{2}24^{-\frac{d}{\alpha-1}}\varepsilon^{-\frac{1}{\alpha-1}}$$

or

$$W \geq 2^m \geq \frac{1}{4\sqrt{2C}}24^{-\frac{d}{2(\alpha-1)}}\varepsilon^{-\frac{1}{2(\alpha-1)}}$$

which is the first statement.

Concerning second one, we have

$$\begin{aligned}\frac{1}{2}24^{-\frac{d}{\alpha-1}}\varepsilon^{-\frac{1}{\alpha-1}} &\leq C'\tilde{L}^2\tilde{W}\log\tilde{W} = C'(L+1)^2(2W+2d+1)\log(2W+2d+1)\\ &\leq C''(\log\varepsilon^{-1})^{2\lambda}W\log W\end{aligned}$$

which implies

$$W\log W \geq \frac{1}{2}24^{-\frac{d}{\alpha-1}}\varepsilon^{-\frac{1}{\alpha-1}}(\log\varepsilon^{-1})^{-2\lambda}.$$

Consider for $\kappa < 1$,

$$W_\kappa = \kappa 24^{-\frac{d}{\alpha-1}}\varepsilon^{-\frac{1}{\alpha-1}}\log(\varepsilon^{-1})^{-2\lambda-1}.$$

Then we have that

$$\log W_\kappa = \log\kappa - \frac{d\log 24}{\alpha-1} - (2\lambda+1)\log(\log\varepsilon^{-1}) + \frac{1}{\alpha-1}(\log\varepsilon^{-1}) \leq \frac{1}{\alpha-1}(\log\varepsilon^{-1}).$$

From this we obtain

$$W_\kappa\log W_\kappa \leq \frac{\kappa}{\alpha-1}24^{-\frac{d}{\alpha-1}}\varepsilon^{-\frac{1}{\alpha-1}}(\log\varepsilon^{-1})^{-2\lambda} \leq \frac{1}{2C''}24^{-\frac{d}{\alpha-1}}\varepsilon^{-\frac{1}{\alpha-1}}(\log\varepsilon^{-1})^{-2\lambda} \leq W\log(W)$$

if we choose $\frac{\kappa}{\alpha-1} \leq \min\{1, \frac{1}{2C''}\}$. Consequently, we get

$$\kappa 24^{-\frac{d}{\alpha-1}}\varepsilon^{-\frac{1}{\alpha-1}}(\log\varepsilon^{-1})^{-2\lambda-1} = W_\kappa \leq W$$

which is the second statement. The proof is completed. □

# 5 Concluding remarks

We have constructed interpolation linear algorithms of sampling recovery on sparse grids of points which are tailored for approximation in the norm of the isotropic Sobolev space $W_0^{1,p}$ of functions from the Hölder-Zygmund space of mixed smoothness $H_\infty^\alpha(\mathbb{I}^d)$, $\alpha \in (1,2]$. These grids have a definite advantage over the standard grids and classical Smolyak sparse grids. They are in some sense optimal in terms of sampling $n$-widths $r_n$ which characterizes the best approximation error by linear algorithms of sampling recovery of functions $f$ from the unit ball $\mathring{U}_\infty^\alpha$ based on $n$ sampled values of $f$. Moreover, with some restrictions we proved the tight dimension-dependent estimates $C_1B_*^{-d}n^{-(\alpha-1)} \leq r_n \leq C_2B^{*-d}n^{-(\alpha-1)}$ with $B_* > 1$ and $B^* > 1$, which show that the sampling $n$-widths are decreasing as fast as the exponent $B^{*-d}$ when the dimension $d$ going to $\infty$.

Based on the constructed linear algorithms of sparse-grid sampling recovery, we have explicitly constructed a deep ReLU neural network $\Phi_f$ having an output that approximates in $W_0^{1,p}$-norm any function $f \in \mathring{U}_\infty^\alpha$ with a prescribed accuracy $\varepsilon$ and established a dimension-dependent estimate for

the computation complexity characterized by number of weights $W(\Phi_f)$ and the depth $L(\Phi_f)$ of this deep ReLU neural network: $L(\Phi_f) \le C_3 \log d \log(\varepsilon^{-1})$ and $W(\Phi_f) \le C_4 B_2^{-d} (\varepsilon^{-1})^{\frac{1}{\alpha-1}} \log(\varepsilon^{-1})$ with $B_2 > 1$. Moreover, this output can be designed also as an output of another "very" deep ReLU neural network $\Phi_f^*$ having computation complexity expressing the $\varepsilon$-independent width $N_w(\Phi_f^*) \le C_5 d$ and the dimension-dependent depth $L(\Phi_f^*) \le C_6 B_2^{-d} (\varepsilon^{-1})^{\frac{1}{\alpha-1}} \log(\varepsilon^{-1})$. This shows in particular, that the computation complexity is decreasing as fast as the exponent ${B_2}^{-d}$ when the dimension $d$ going to $\infty$. In the case when $p = \infty$, we gave dimension-dependent lower bounds for the numbers of weights of deep ReLU networks whose outputs approximate functions from $\mathring{U}_\infty^\alpha$ with a given accuracy.

Thus, we have shown that under some reasonable restrictions the curse of dimensionality can be avoided in the both approximations by sparse-grid sampling recovery and by deep ReLU neural networks. The result also indicated that the decomposition of functions from the Hölder-Zygmund space of mixed smoothness $H_\infty^\alpha(\mathbb{I}^d)$ into tensor product Faber series plays a fundamental role in construction of linear algorithms of sparse-grid sampling recovery and of deep ReLU neural networks for approximation of functions from $H_\infty^\alpha(\mathbb{I}^d)$. In the present paper, our concerns are non-adaptive approximations by sparse-grid sampling recovery and by deep ReLU neural networks for which algorithms of sampling recovery are linear and the architecture of deep ReLU neural networks is the same for all functions. In a forthcoming paper, we will discuss a problem of adaptive nonlinear approximation by sparse-grid sampling recovery and by deep ReLU neural networks of multivariate functions having a mixed smoothness.

**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 authors were working at the Vietnam Institute for Advanced Study in Mathematics (VIASM). They would like to thank the VIASM for providing a fruitful research environment and working condition.

## References

[1] M. Anthony and P. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, Cambridge, 2009.

[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] R. Bellmann. *Dynamic Programming*. Princeton University Press, Princeton, 1957.

[4] H.-J. Bungartz and M. Griebel. Sparse grids. *Acta Numer.*, 13:147–269, 2004.

[5] G. Byrenheid, D. Dũng, W. Sickel, and T. Ullrich. Sampling on energy-norm based sparse grids for the optimal recovery of Sobolev type functions in $H_\gamma$. *J. Approx. Theory*, 207:207–231, 2016.

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

[7] D. Dũng. B-spline quasi-interpolant representations and sampling recovery of functions with mixed smoothness. *J. Complexity*, 27:541–567, 2011.

[8] D. Dũng. Optimal adaptive sampling recovery. *Adv. Comput. Math*, 34:1–41, 2011.

[9] D. Dũng. Sampling and cubature on sparse grids based on a B-spline quasi-interpolation. *Found. Comp. Math.*, 16:1193–1240, 2016.

[10] D. Dũng, V. N. Temlyakov, and T. Ullrich. *Hyperbolic Cross Approximation.* Advanced Courses in Mathematics - CRM Barcelona, Birkhäuser/Springer, 2018.

[11] D. Dũng and M. X. Thao. Dimension-dependent error estimates for sampling recovery on Smolyak grids based on B-spline quasi-interpolation. *J. Approx. Theory*, 250:185–205, 2020.

[12] I. Daubechies, R. DeVore, S. Foucart, B. Hanin, and G. Petrova. Nonlinear approximation and (Deep) ReLU networks. *arXiv:1905.02199*, 2019.

[13] R. DeVore and G. Lorentz. *Constructive Approximation.* Springer-Verlag, New York, 1993.

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

[15] G. Faber. Über stetige Funktionen. *Math. Ann.*, 66:81–94, 1909.

[16] A. Griewank, F. Y. Kuo, H. Leövey, and I. H. Sloan. High dimensional integration of kinks and jumps – smoothing by preintegration. *J. Comput. Appl. Math.*, 344:259–274, 2018.

[17] P. Grohs, D. Perekrestenko, D. Elbrachter, and H. Bolcskei. Deep neural network approximation theory. *arXiv: 1901.02220*, 2019.

[18] I. Gühring, G. Kutyniok, and P. Petersen. Error bounds for approximations with deep ReLU neural networks in $W^{s,p}$ norms. *https://doi.org/10.1142/S0219530519410021*, 00:00, 2020.

[19] D. Hebb. *The Organization of Behavior: A Neuropsychological Theory.* Wiley, 1949.

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

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. *NeurIPS*, pages 1106–1114, 2012.

[22] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[23] H. Mhaskar and T. Poggio. Deep vs. shallow networks: An approximation theory perspective. *Anal. Appl.*, 14:829–848, 2016.

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

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

[26] 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.

[27] E. Novak and H. Woźniakowski. *Tractability of Multivariate Problems, Volume I: Linear Information.* EMS Tracts in Mathematics, Vol. 6, Eur. Math. Soc. Publ. House, Zürich, 2008.

[28] E. Novak and H. Woźniakowski. *Tractability of Multivariate Problems, Volume II: Standard Information for Functionals.* EMS Tracts in Mathematics, Vol. 12, Eur. Math. Soc. Publ. House, Zürich, 2010.

[29] J. A. A. Opschoor, P. C. Petersen, and C. Schwab. Deep ReLU networks and high-order finite element methods. *Anal. Appl. (Singap.)*, https://doi.org/10.1142/S0219530519410136, 2020.

[30] J. A. A. Opschoor, C. Schwab, and J. Zech. Exponential ReLU DNN expression of holomorphic maps in high dimension. *SAM, Research Report No. 2019-35*, 2019.

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

[32] A. Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numer.*, 8:143–195, 1999.

[33] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.*, 65:386–408, 1958.

[34] 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.

[35] S. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl. Akad. Nauk*, 148:1042–1045, 1963.

[36] 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.

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

[38] 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.

[39] H. Triebel. *Bases in Function Spaces, Sampling, Discrepancy, Numerical Integration.* European Math. Soc. Publishing House, Zürich, 2010.

[40] H. Triebel. *Hybrid Function Spaces, Heat and Navier-Stokes Equations.* European Mathematical Society, 2015.

[41] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, and M. Norouzi. Googles neural machine translation system: Bridging the gap between human and machine translation. *arXiv: 1609.08144, 2016.*

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

[43] D. Yarotsky. Quantied advantage of discontinuous weight selection in approximations with deep neural networks. *arXiv: 1705.01365*, 2017.

[44] H. Yserentant. *Regularity and Approximability of Electronic Wave Functions.* Lecture Notes in Mathematics, Springer, 2010.

[45] C. Zenger. Sparse grids. *In Parallel Algorithms for Partial Differential Equations. Volume 31 of Notes on Numerical Fluid Mechanics (Vieweg, Wiesbaden 1991)*, pages 241–251.