Journal of Machine Learning Research 27 (2026) 1-41 Submitted 4/25; Revised 2/26; Published 2/26

Nonlocal Techniques for the Analysis of Deep ReLU Neural Network Approximations

Cornelia Schneider CORNELIA.SCHNEIDER@FAU.DE
Applied Mathematics III
Friedrich-Alexander Universität Erlangen
Cauerstr. 11, 91058 Erlangen, Germany

Mario Ullrich MARIO.ULLRICH@JKU.AT
Institute of Analysis and
Department of Quantum Information and Computation
Johannes Kepler University
Altenberger Str. 69, 4040, Linz, Austria

Jan Vybíral JAN.VYBIRAL@FJFI.CVUT.CZ
Department of Mathematics
Faculty of Nuclear Sciences and Physical Engineering
Czech Technical University in Prague
Trojanova 13, 12000 Praha, Czech Republic

Editor: Lorenzo Rosasco

Abstract

In recent work concerned with the approximation and expressive powers of deep neural networks, Daubechies, DeVore, Foucart, Hanin, and Petrova introduced a system of piecewise linear functions, which can be easily reproduced by artificial neural networks with the ReLU activation function, and showed that it forms a Riesz basis of $L_2([0,1])$. Their work was subsequently generalized to the multivariate setting by Schneider and Vybíral. In the work at hand, we show that this system serves as a Riesz basis also for Sobolev spaces $W^s([0,1]^d)$ and Barron classes $\mathbb{B}^s([0,1]^d)$ with smoothness $0 < s < 1$. We apply this fact to re-prove some recent results on the approximation of functions from these classes by deep neural networks. Our proof method avoids using local approximations and also allows us to track the implicit constants as well as to show that we can avoid the curse of dimension. Moreover, we also study how well one can approximate Sobolev and Barron functions by neural networks if only function values are known.

Keywords: Riesz basis, Rectified Linear Unit (ReLU), artificial neural networks, random sampling, Sobolev and Barron classes

1. Introduction

Since the appearance of *(artificial) neural networks*, abbreviated sometimes by NN in the sequel, many authors investigated which functions can be computed or well-approximated by neural networks of a given structure. This research field is known as *expressivity* of neural networks (Raghu et al., 2017) and it aims to help to explain their empirical success. The first results in this line of study were rather theoretical universal approximation results of Cybenko (1989), Hornik et al. (1989), or Leshno et al. (1993).

An important breakthrough was achieved already by Barron (1993), where a general result about convex hulls in Hilbert spaces (attributed to Maurey by Pisier, 1981) was exploited to show that approximation of functions with finite first Fourier moments by neural networks does not suffer the curse of dimensionality, in contrast to any linear method of approximation. These function classes are nowadays called Barron classes (together with their numerous variants). Since then, the approximation of functions from different function spaces (Sobolev spaces, functions of bounded variation, Lipschitz continuous functions, etc.) using neural networks attracted a lot of attention and many optimal and nearly optimal results are nowadays available. We refer to E et al. (2022); E and Wojtowytsch (2020a, 2022) for more on the representation of neural networks, as well as Barron (1994); Bölcskei et al. (2019); Daubechies et al. (2022); DeVore et al. (2021); Eckle and Schmidt-Hieber (2019); Mhaskar (1996); Shen et al. (2022); Siegel (2023); Yarotsky (2017) and references therein for further information on approximation with neural networks.

A classical approach in the field is to use some sort of local approximation (based on local decomposition of unity, Taylor approximation, or decomposition into localized building blocks like atoms or wavelets), which typically leads to a bad dependence of the implicit constants on the underlying dimension. The main aim of our work is to re-shape this technique by using certain nonlocal decompositions, which we now describe in detail.

Recently, Daubechies, DeVore, Foucart, Hanin, and Petrova (2022) introduced a system of piecewise linear functions, which resembles the behavior of a trigonometric orthonormal basis on the interval $[0,1]$ but, simultaneously, can be also easily reproduced by neural networks with the $\mathrm{ReLU}(x) := \max\{x,0\}$ activation function. To be more precise, for $x \in [0,1]$, we consider

$$\mathcal{C}(x) := 4\left|x - \frac{1}{2}\right| - 1 = \begin{cases} 1-4x, & x \in [0,1/2), \\ 4x-3, & x \in [1/2,1] \end{cases} \tag{1}$$

and

$$\mathcal{S}(x) := \left|2 - 4\left|x - \frac{1}{4}\right|\right| - 1 = \begin{cases} 4x, & x \in [0,1/4), \\ 2-4x, & x \in [1/4,3/4), \\ 4x-4, & x \in [3/4,1]. \end{cases}$$

Indeed, $\mathcal{C}(x)$ and $\mathcal{S}(x)$ are piecewise linear functions, which interpolate $\cos(2\pi x)$ and $\sin(2\pi x)$ for $x \in \{0, 1/4, 1/2, 3/4, 1\}$. We extend this definition periodically, i.e. $\mathcal{C}(x) = \mathcal{C}(x - \lfloor x \rfloor)$ and $\mathcal{S}(x) = \mathcal{S}(x - \lfloor x \rfloor)$ for all $x \in \mathbb{R}$. Moreover, for $k \geq 1$ and $x \in \mathbb{R}$, we put $\mathcal{C}_k(x) := \mathcal{C}(kx)$ and $\mathcal{S}_k(x) := \mathcal{S}(kx)$.

It was shown by Daubechies et al. (2022), that the set $\{\mathcal{C}_k,\, \mathcal{S}_k \colon k\in\mathbb{N}\}$ forms a Riesz basis of the subspace of $L_2([0,1])$ of functions with vanishing mean. This means that

$$c(\|\alpha\|_2^2+\|\beta\|_2^2)\le\left\|\sum_{k=1}^{\infty}(\alpha_k\mathcal{C}_k+\beta_k\mathcal{S}_k)\right\|_2^2\le C(\|\alpha\|_2^2+\|\beta\|_2^2) \tag{2}$$

holds for two absolute constants $C,c>0$ and for any two real sequences $\alpha=(\alpha_k)_{k=1}^\infty$ and $\beta=(\beta_k)_{k=1}^\infty$ and that every function from $L_2([0,1])$ with vanishing mean lies in its closed linear span. The values of $C,c>0$ can be chosen, for example, as $c=1/6$ and $C=1/2$. The functions $\mathcal{C}_k$ and $\mathcal{S}_k$ have L_2-norm $3^{-1/2}$, but this is not important in what follows. Note that (2) would hold for the normalized system with constants 1/2 and 3/2 instead. It is also easy to see that adding a constant function 1 to this system makes it a Riesz basis of the whole $L_2([0,1])$. Finally, Daubechies et al. (2022, Theorem 6.2) provides, for every $k\ge 1$, a construction of a feed-forward artificial neural network (called simply a neural network in the sequel) with $L=\lceil\log_2 k\rceil+1$ hidden layers and with 2 artificial neurons in each layer, which reconstructs $\mathcal{C}_k$ on $[0,1]$. The same is true for $\mathcal{S}_k$ if we increase the number of layers by one. Note that this construction depends on k, but only very mildly, as the number of hidden layers grows only logarithmically with k. Although this means that no single neural network (or a finite ensemble of neural networks) can reproduce all the elements of this system, this does not represent an essential obstacle, as we always approximate a given function f by a finite linear combination of functions from this system.

The generalization of this approach to functions of $d\ge 2$ variables and the space $L_2([0,1]^d)$ is not straightforward, and the seemingly simple way of taking tensor products suffers a number of drawbacks. Firstly, such functions could not be exactly recovered by neural networks with the ReLU activation function, they could only be approximated to some limited precision, see Elbrächter et al. (2021); Telgarsky (2015); Yarotsky (2017). And secondly, the ratio of the optimal constants C and c in the corresponding version of (2) would grow exponentially with the underlying dimension d.

A surprisingly simple and effective generalization of the univariate Riesz basis to higher dimensions was discovered by Schneider and Vybíral (2024). For this, let us define

$$\mathcal{C}_k(x):=\mathcal{C}(k\cdot x)\qquad\text{and}\qquad\mathcal{S}_k(x):=\mathcal{S}(k\cdot x),$$

where $k\cdot x=\sum_j k_jx_j$ is the usual inner product of $k\in\mathbb{Z}^d$ and $x\in\mathbb{R}^d$, and define the system

$$\mathcal{R}_d:=\{1\}\cup\Big\{\mathcal{C}_k,\,\mathcal{S}_k\colon k\in\mathbb{Z}^d,\,k\rhd 0\Big\}. \tag{3}$$

Here, $k\rhd 0$ means that $k=(k_1,\dots,k_d)\in\mathbb{Z}^d$ is not equal to zero and the first non-zero entry of k is positive. It was shown by Schneider and Vybíral (2024) that $\mathcal{R}_d$ is a Riesz basis of $L_2([0,1]^d)$ for every $d\ge 2$ and that the constants $C,c>0$ in the corresponding analogue of (2) can be chosen again as $c=1/6$ and $C=1/2$ (if we leave out the constant function from $\mathcal{R}_d$, otherwise $C=1$), independently of d. Again, the elements of (3) have L_2-norm $3^{-1/2}$, and can easily be reproduced by neural networks with the ReLU activation function, see Section 3.

The first aim of the present paper is to extend the results of Schneider and Vybíral (2024) to other function spaces than $L_2([0,1]^d)$. In fact, we show that properly normalized analogues of $\mathcal{R}_d$ are Riesz bases of the Sobolev spaces $W^s([0,1]^d)$ and the Barron classes $\mathbb{B}^s([0,1]^d)$ for every $0 < s < 1$, see Section 2 for the exact statements. In both cases, the constants in (2) can again be chosen independently of d. These results allow to bridge the gap between classical harmonic analysis (represented by the technique of multivariate Fourier series) and the analysis of expressivity of neural networks. And, in contrast to the use of wavelets, curvelets, or other local techniques of multi-scale representation, our approach scales very favorably with the underlying dimension d.

In Section 3, we apply our new results to the approximation of functions from Sobolev and Barron spaces by deep neural networks. We avoid any use of local approximation, which usually leads to a bad dependence of the implicit constants on the dimension and to some quite technical computations. Instead, we work exclusively with the building blocks of (3), which are defined on the whole unit cube of $\mathbb{R}^d$. The proof method we use is actually quite straightforward - we decompose a given f from one of these function spaces into a series involving the basis functions from (3). Truncating this series at a suitable position splits the series into two parts. The first one is recovered exactly by a suitably chosen neural network, the second one is simply the error of approximation. For Barron spaces, we need to combine this approach with the concept of *best n-term approximation*, well-known in non-linear approximation theory, to select the most important terms from the series decomposition of f. Our technique allows us to work exclusively on the sequence space level and, by the properties of (3) as shown in Section 2, we virtually pay no price for this step. In particular, we are able to track the dependence of the parameters on the underlying dimension d and we show that in some cases we can indeed avoid their exponential dependence on d. In this way, we reprove some of the known results but using an essentially different technique.

In Section 4, we discuss our results on function recovery and compare them to the existing literature. We also comment on learning the specific neural network based on (randomly chosen) function values of f in Section 5. We conclude with a discussion and comments on future research.

2. Riesz Bases of Sobolev and Barron Classes

It was shown by Daubechies, DeVore, Foucart, Hanin, and Petrova (2022) that $\mathcal{R}_1$ is a Riesz basis of $L_2([0,1])$. This result was generalized to the multivariate setting by Schneider and Vybíral (2024), where the authors discovered that $\mathcal{R}_d$ from (3) is a Riesz basis of $L_2([0,1]^d)$ for every $d \geq 2$ and that the constants in the Riesz-type estimate (2) can be chosen independently of d.

The aim of this section is to study the properties of the univariate system $\mathcal{R}_1$ and its multivariate analogue $\mathcal{R}_d$ in other function spaces than just $L_2([0,1]^d)$. To simplify the presentation, we first deal with the univariate Sobolev spaces in Subsection 2.1 before we come to their high-dimensional counterparts in Subsection 2.2. Finally, in Subsection 2.3, we investigate the properties of $\mathcal{R}_1$ and $\mathcal{R}_d$ within the context of Barron classes, which appear to be better suited for high-dimensional applications.

2.1 Univariate Sobolev Classes

Let us consider a real-valued square-integrable function $f : \mathbb{R} \to \mathbb{R}$, which is periodic with period one, i.e., f is given by

$$f(x) \,=\, a_0 \,+\, \sum_{m=1}^{\infty} a_m \cos(2\pi m x) + b_m \sin(2\pi m x), \quad x \in [0,1]. \tag{4}$$

For a real number $s \geq 0$, we define the usual Sobolev spaces of periodic functions of order s by

$$W^s([0,1]) := \left\{ f : f \text{ is given by (4) and } \|f\|_{W^s}^2 := a_0^2 + \sum_{m=1}^{\infty} m^{2s}(a_m^2 + b_m^2) < \infty \right\},$$

and note that $W^0([0,1]) = L_2([0,1])$.

It was shown by Daubechies et al. (2022) that the system $\mathcal{R}_1$ from (3) forms a Riesz basis in $W^0([0,1])$. The aim of this section is to investigate the properties of $\mathcal{R}_1$ as subsets of $W^s([0,1])$. We start by defining the analogues of $W^s([0,1])$ based on (3). Let

$$f(x) = \alpha_0 \,+\, \sum_{k=1}^{\infty} \alpha_k \mathcal{C}_k(x) + \beta_k \mathcal{S}_k(x), \quad x \in [0,1], \tag{5}$$

with $\alpha_k, \beta_k \in \mathbb{R}$, which converges in $L_2([0,1])$ if $\alpha = (\alpha_k)_{k=1}^{\infty}$ and $\beta = (\beta_k)_{k=1}^{\infty}$ are square-summable. For $s \geq 0$, we define

$$\mathcal{F}^s([0,1]) \,:=\, \left\{ f : f \text{ is given by (5) and } \|f\|_{\mathcal{F}^s}^2 := \alpha_0^2 \,+\, \sum_{m=1}^{\infty} m^{2s}(\alpha_m^2 + \beta_m^2) < \infty \right\}.$$

The following theorem shows that $\mathcal{F}^s$ is a useful tool for the analysis of W^s.

Theorem 1. *Let $0 \leq s < 1$. Then, $W^s([0,1]) = \mathcal{F}^s([0,1])$ in the sense of equivalent norms. Moreover, the system*

$$\{1\} \cup \left\{ k^{-s}\mathcal{C}_k, k^{-s}\mathcal{S}_k \colon k \in \mathbb{N} \right\} \tag{6}$$

is a Riesz basis of $W^s([0,1])$ and for constants $c, C > 0$ it holds

$$c \sum_{k\in\mathbb{N}} (\alpha_k^2 + \beta_k^2) \;\leq\; \left\| \sum_{k\in\mathbb{N}} k^{-s} \Big(\alpha_k \mathcal{C}_k + \beta_k \mathcal{S}_k \Big) \right\|_{W^s}^2 \;\leq\; C \sum_{k\in\mathbb{N}} (\alpha_k^2 + \beta_k^2). \tag{7}$$

Modifications of (7) for functions with constant term are obvious (by incorporating α_0).

Proof We assume without any loss of generality that $a_0 = \alpha_0 = 0$ throughout the proof.

Step 1. First, we show that $W^s([0,1]) \hookrightarrow \mathcal{F}^s([0,1])$. Let f be given by (4). We combine it with Daubechies et al. (2022) (cf. also Schneider and Vybíral, 2024, Lemma 2.5), which states that

$$\sqrt{2}\cos(2\pi x) = \sum_{\ell=0}^{\infty} \frac{\mu(2\ell+1)}{(2\ell+1)^2} \cdot \frac{\sqrt{3}}{\kappa} \mathcal{C}_{2\ell+1}(x), \tag{8}$$

$$\sqrt{2}\sin(2\pi x) = \sum_{\ell=0}^{\infty}(-1)^{\ell}\frac{\mu(2\ell+1)}{(2\ell+1)^2}\cdot\frac{\sqrt{3}}{\kappa}\mathcal{S}_{2\ell+1}(x), \tag{9}$$

where $\kappa^2 = 96/\pi^4$ and $\mu(n) \in \{-1, 0, 1\}$ denotes the Möbius function, i.e., the sum of the primitive n-th roots of unity. Let us note that $\mu(n) = 0$ if n is not square-free, i.e., if it is divisible by some squared prime.

Plugging (8) and (9) into (4), we obtain

$$\begin{aligned} f(x) &= \frac{\sqrt{3}}{\sqrt{2}\,\kappa}\Big\{\sum_{m=1}^{\infty} a_m \sum_{\ell=0}^{\infty}\frac{\mu(2\ell+1)}{(2\ell+1)^2}\,\mathcal{C}_{(2\ell+1)m}(x) \\ &\qquad + \sum_{m=1}^{\infty} b_m \sum_{\ell=0}^{\infty}(-1)^{\ell}\frac{\mu(2\ell+1)}{(2\ell+1)^2}\,\mathcal{S}_{(2\ell+1)m}(x)\Big\} \\ &= \frac{\sqrt{3}}{\sqrt{2}\,\kappa}\sum_{k=1}^{\infty}\Big[\alpha_k\mathcal{C}_k(x) + \beta_k\mathcal{S}_k(x)\Big], \end{aligned}$$

where

$$\alpha_k = \sum_{(\ell,m):k=(2\ell+1)m} a_m\cdot\frac{\mu(2\ell+1)}{(2\ell+1)^2} \quad\text{and}\quad \beta_k = \sum_{(\ell,m):k=(2\ell+1)m} b_m\cdot(-1)^{\ell}\frac{\mu(2\ell+1)}{(2\ell+1)^2}.$$

We now show that $(\alpha_k\cdot k^s)_{k=1}^{\infty}$ is square summable if $0 \le s < 1$ and $a^s := (a_m\cdot m^s)_{m=1}^{\infty}$ is square summable. We rewrite

$$\begin{aligned} \sum_{k=1}^{\infty}\alpha_k^2\,k^{2s} &= \sum_{k=1}^{\infty}k^{2s}\sum_{(\ell,m):k=(2\ell+1)m} a_m\frac{\mu(2\ell+1)}{(2\ell+1)^2}\cdot\sum_{(\ell',m'):k=(2\ell'+1)m'} a_{m'}\frac{\mu(2\ell'+1)}{(2\ell'+1)^2} \\ &= \sum_{\substack{k,\ell,m,\ell',m' \\ k=(2\ell+1)m \\ k=(2\ell'+1)m'}} k^{2s}a_m\frac{\mu(2\ell+1)}{(2\ell+1)^2}\cdot a_{m'}\frac{\mu(2\ell'+1)}{(2\ell'+1)^2} \\ &= \sum_{m,m'=1}^{\infty} a_m a_{m'} m^s (m')^s \sum_{\substack{k,\ell,\ell' \\ k=(2\ell+1)m \\ k=(2\ell'+1)m'}} \frac{\mu(2\ell+1)}{(2\ell+1)^{2-s}}\cdot\frac{\mu(2\ell'+1)}{(2\ell'+1)^{2-s}} \\ &= \sum_{m,m'=1}^{\infty} a_m m^s\cdot a_{m'}(m')^s X_{m,m'} = \langle a^s, Xa^s\rangle, \end{aligned} \tag{10}$$

where

$$X_{m,m'} = \sum_{\substack{\ell,\ell' \\ (2\ell+1)m=(2\ell'+1)m'}} \frac{\mu(2\ell+1)}{(2\ell+1)^{2-s}}\cdot\frac{\mu(2\ell'+1)}{(2\ell'+1)^{2-s}}. \tag{11}$$

We aim to show that $X = (X_{m,m'})_{m,m'=1}^{\infty}$ is a bounded operator on ℓ_2, which together with (10), finishes the proof. We shall use the symmetry of X and the following simple observation. If $\sum_{m'}|X_{m,m'}|$ is uniformly bounded over m, then X generates a bounded linear operator on ℓ_1 and also on ℓ_∞ and, by interpolation, also on ℓ_2.

We use that, for any $m \geq 1$ fixed,

$$\sum_{m'=1}^{\infty} |X_{m,m'}| \leq \sum_{\ell,\ell'=1}^{\infty} \frac{|\mu(2\ell+1)|}{(2\ell+1)^{2-s}} \cdot \frac{|\mu(2\ell'+1)|}{(2\ell'+1)^{2-s}} \sum_{m':(2\ell+1)m=(2\ell'+1)m'} 1.$$

The last sum is (at most) 1 and the sum over ℓ and ℓ' is convergent.

Step 2. The proof of $\mathcal{F}^s([0,1]) \hookrightarrow W^s([0,1])$ for $0 \leq s < 1$ follows the same pattern, only now we assume that f is given by (5) and invoke the decomposition of $\mathcal{C}$ and $\mathcal{S}$ into their respective Fourier series (cf. Daubechies et al., 2022, p. 166). As the coefficients of these series decay again quadratically (similarly to (8) and (9)), the rest of the proof follows in the same manner.

Step 3. We finally show that (6) is a Riesz basis of W^s. For this, note that the above shows $\|f\|_{W^s} \approx \|f\|_{\mathcal{F}^s}$, where "$\approx$" means upper and lower bounded with constants that are independent of f. In particular,

$$\Bigg\|\sum_{k\in\mathbb{N}} k^{-s}\Big[\alpha_k\mathcal{C}_k + \beta_k\mathcal{S}_k\Big]\Bigg\|_{W^s}^2 \approx \Bigg\|\sum_{k\in\mathbb{N}} k^{-s}\Big[\alpha_k\mathcal{C}_k + \beta_k\mathcal{S}_k\Big]\Bigg\|_{\mathcal{F}^s}^2 = \sum_{k\in\mathbb{N}} k^{2s}\,\frac{\alpha_k^2+\beta_k^2}{k^{2s}},$$

proving the claim. ■

Remark 1. We comment on the restriction to $0 \leq s < 1$ in Theorem 1.

If $s \geq 3/2$, then (8) implies that $\cos(2\pi x)$ does not lie in $\mathcal{F}^s([0,1])$ and, similarly, one can also show that $\mathcal{C} \notin W^s([0,1])$. Therefore, Theorem 1 cannot hold if $s \geq 3/2$.

The case $1 \leq s < 3/2$ is more delicate and we conjecture that also in this case Theorem 1 fails. Observe that the proof of Theorem 1 reduces the study of the embedding $W^s([0,1]) \hookrightarrow \mathcal{F}^s([0,1])$ to the boundedness of the matrix $X = (X_{m,m'})_{m,m'=1}^{\infty}$ with $X_{m,m'}$ given by (11) on ℓ_2. To rewrite (11) for fixed $m, m' \in \mathbb{N}$, let $g = \gcd(m, m')$ be the greatest common divisor of m and m'. For simplicity, assume for the moment that m and m' are odd. Then all the integer solutions of the equation $(2\ell+1)m = (2\ell'+1)m'$ are given by $2\ell+1 = (2t+1)\cdot\frac{m'}{g}$ and $2\ell'+1 = (2t+1)\cdot\frac{m}{g}$, where $t \in \mathbb{N}_0$ is arbitrary. Therefore, we obtain

$$\begin{aligned} X_{m,m'} &= \sum_{t=0}^{\infty} \frac{\mu((2t+1)\cdot\frac{m'}{g})\mu((2t+1)\cdot\frac{m}{g})}{((2t+1)\cdot\frac{m'}{g})^{2-s}((2t+1)\cdot\frac{m}{g})^{2-s}} \\ &= \frac{\gcd(m,m')^{4-2s}}{m^{2-s}m'^{2-s}} \sum_{t=0}^{\infty} \frac{\mu((2t+1)\cdot\frac{m'}{g})\mu((2t+1)\cdot\frac{m}{g})}{(2t+1)^{4-2s}}. \end{aligned}$$

The boundedness of the infinite matrix $Y = (\gcd(m,m')^{4-2s}/[m\cdot m']^{2-s})_{m,m'=1}^{\infty}$ was investigated by Lindqvist and Seip (1998), Hedenmalm et al. (1997), and Wintner (1944, page 578) and it is known to fail if $3/2 > s \geq 1$. Therefore, we believe that also X is not bounded on ℓ_2 for this range of s.

Remark 2. Finally, let us remark that by using similar techniques as above we can also prove the embedding

$$W^s([0,1]) \hookrightarrow \mathcal{F}^\gamma([0,1]) \qquad \text{for all} \qquad \gamma < \min\left\{s, \frac{1+s}{2}, \frac{3}{2}\right\},$$

which, in particular, implies $W^1([0,1]) \hookrightarrow \mathcal{F}^{1-\delta}([0,1])$ for all $\delta > 0$. We omit the details.

2.2 Multivariate Sobolev Classes

We now show how the results of the previous section can be extended to higher dimensions. In general, we follow the scheme presented for the univariate case. This means that we consider real-valued square-integrable functions $f : \mathbb{R}^d \to \mathbb{R}$ with period one in each variable. Such functions can be represented by the d-variate Fourier series

$$f(x) = \sum_{m \in \mathbb{Z}^d} \gamma_m e^{2\pi i m \cdot x}, \quad x \in [0,1]^d \tag{12}$$

with complex coefficients γ_m, which satisfy $\gamma_{-m} = \overline{\gamma_m}$ for every $m \in \mathbb{Z}^d$. As we prefer to work with real-valued functions (and real neural networks), we replace (12) with

$$f(x) = a_0 + \sum_{m \mathrel{+\!\!\!\!>} 0} \Big[a_m \cos(2\pi m \cdot x) + b_m \sin(2\pi m \cdot x)\Big], \quad x \in [0,1]^d \tag{13}$$

with real coefficients a_m and b_m. Recall that $m \mathrel{+\!\!\!\!>} 0$ means that $m = (m_1, \dots, m_d) \in \mathbb{Z}^d$ is not equal to zero and the first non-zero entry of m is positive. We define for every $s \geq 0$ the Sobolev space

$$W^s([0,1]^d) := \left\{ f : f \text{ given by (13) and } \|f\|_{W^s}^2 = a_0^2 + \sum_{m \mathrel{+\!\!\!\!>} 0} \|m\|_2^{2s} \cdot (a_m^2 + b_m^2) < \infty \right\}.$$

As mentioned already in the introduction, it was discovered by Schneider and Vybíral (2024) that $\mathcal{R}_d$ from (3) is a Riesz basis of $L_2([0,1]^d)$ for every $d \geq 2$ and that the constants in the Riesz-type estimate (2) can be chosen independently of d. The aim of this section is to extend this result also to $W^s([0,1]^d)$ for s small enough.

Recall that $\mathcal{C}_k(x) := \mathcal{C}(k \cdot x)$ and $\mathcal{S}_k(x) := \mathcal{S}(k \cdot x)$ and consider the decomposition

$$f(x) \;=\; \alpha_0 + \sum_{k \mathrel{+\!\!\!\!>} 0} \Big[\alpha_k \mathcal{C}_k(x) + \beta_k \mathcal{S}_k(x)\Big], \quad x \in [0,1]^d \tag{14}$$

with real coefficients α_m and β_m. We define for $s \geq 0$ the spaces

$$\mathcal{F}^s([0,1]^d) := \left\{ f : f \text{ is given by (14) and } \|f\|_{\mathcal{F}^s}^2 := \alpha_0^2 + \sum_{k \mathrel{+\!\!\!\!>} 0} \|k\|_2^{2s} \left(\alpha_k^2 + \beta_k^2\right) < \infty \right\}.$$

We now present the multivariate analogue of Theorem 1.

Theorem 2. *Let $0 \le s < 1$. Then, $W^s([0,1]^d) = \mathcal{F}^s([0,1]^d)$ in the sense of equivalent norms. The constants of equivalence of these norms depend on s but are independent of d. Moreover, the system*

$$\{1\} \cup \Big\{ \|k\|_2^{-s} \mathcal{C}_k,\ \|k\|_2^{-s} \mathcal{S}_k \colon k \in \mathbb{Z}^d,\ k \mathrel{+\!\!\!>} 0 \Big\} \tag{15}$$

is a Riesz basis of $W^s([0,1]^d)$ and for constants $c, C > 0$, which can be chosen independently of d, it holds

$$c \sum_{k \mathrel{+\!\!\!>} 0} (\alpha_k^2 + \beta_k^2) \ \le\ \bigg\| \sum_{k \mathrel{+\!\!\!>} 0} \|k\|_2^{-s} \Big(\alpha_k \mathcal{C}_k + \beta_k \mathcal{S}_k \Big) \bigg\|_{W^s}^2 \ \le\ C \sum_{k \mathrel{+\!\!\!>} 0} (\alpha_k^2 + \beta_k^2). \tag{16}$$

Again, modifications of (16) *for functions with constant term are obvious.*

Proof We proceed similarly as in the proof of Theorem 1.

Step 1. First, we show that $W^s([0,1]^d) \hookrightarrow \mathcal{F}^s([0,1]^d)$.

Let f be given by (13), i.e., let

$$f(x) = a_0 + \sum_{m \mathrel{+\!\!\!>} 0} a_m \cos(2\pi m \cdot x) + \sum_{m \mathrel{+\!\!\!>} 0} b_m \sin(2\pi m \cdot x).$$

We estimate only the first sum, the second can be treated similarly. Using (8) we obtain

$$\begin{aligned} \sum_{m \mathrel{+\!\!\!>} 0} a_m \cos(2\pi m \cdot x) &= \frac{\sqrt{3}}{\sqrt{2}\,\kappa} \sum_{m \mathrel{+\!\!\!>} 0} a_m \sum_{\ell=0}^{\infty} \frac{\mu(2\ell+1)}{(2\ell+1)^2} \cdot \mathcal{C}((2\ell+1) m \cdot x) \\ &= \frac{\sqrt{3}}{\sqrt{2}\,\kappa} \sum_{k \mathrel{+\!\!\!>} 0} \alpha_k \mathcal{C}(k \cdot x), \end{aligned}$$

where

$$\alpha_k = \sum_{(\ell,m):(2\ell+1)m=k} a_m \cdot \frac{\mu(2\ell+1)}{(2\ell+1)^2}.$$

From this we deduce

$$\begin{aligned} \sum_{k \mathrel{+\!\!\!>} 0} \alpha_k^2\, \|k\|_2^{2s} &= \sum_{k \mathrel{+\!\!\!>} 0} \|k\|_2^{2s} \sum_{(\ell,m):k=m(2\ell+1)} a_m \frac{\mu(2\ell+1)}{(2\ell+1)^2} \cdot \sum_{(\ell',m'):k=m'(2\ell'+1)} a_{m'} \frac{\mu(2\ell'+1)}{(2\ell'+1)^2} \\ &= \sum_{\substack{k,\ell,m,\ell',m' \\ k=m(2\ell+1) \\ k=m'(2\ell'+1)}} \|k\|_2^{2s} a_m \frac{\mu(2\ell+1)}{(2\ell+1)^2} \cdot a_{m'} \frac{\mu(2\ell'+1)}{(2\ell'+1)^2} \\ &= \sum_{m,m' \mathrel{+\!\!\!>} 0} a_m \|m\|_2^s \cdot a_{m'} \|m'\|_2^s X_{m,m'}, \end{aligned}$$

where

$$X_{m,m'} = \sum_{\substack{\ell,\ell' \\ (2\ell+1)m=(2\ell'+1)m'}} \frac{\|(2\ell+1)m\|_2^s}{\|m\|_2^s} \cdot \frac{\|(2\ell'+1)m'\|_2^s}{\|m'\|_2^s} \cdot \frac{\mu(2\ell+1)}{(2\ell+1)^2} \cdot \frac{\mu(2\ell'+1)}{(2\ell'+1)^2}$$
$$= \sum_{\substack{\ell,\ell' \\ (2\ell+1)m=(2\ell'+1)m'}} \frac{\mu(2\ell+1)}{(2\ell+1)^{2-s}} \cdot \frac{\mu(2\ell'+1)}{(2\ell'+1)^{2-s}}.$$

We finish the proof by showing that X has uniformly bounded row sums and, therefore, is bounded on ℓ_2. Indeed,

$$\sum_{m' +\!\!\!> 0} |X_{m,m'}| \leq \sum_{\ell,\ell'=0}^{\infty} \frac{|\mu(2\ell+1)|}{(2\ell+1)^{2-s}} \cdot \frac{|\mu(2\ell'+1)|}{(2\ell'+1)^{2-s}} \sum_{m':(2\ell+1)m=(2\ell'+1)m'} 1,$$

which is again convergent for $0 \leq s < 1$, and the norm is uniformly bounded in $d \geq 1$.

Step 2. The inverse embedding $\mathcal{F}^s([0,1]^d) \hookrightarrow W^s([0,1]^d)$ follows in a very similar way. We assume that f is given by (14), decompose $\mathcal{C}$ and $\mathcal{S}$ into their respective Fourier series and reduce the embedding to the boundedness of a certain infinite matrix on ℓ_2. As it resembles very much the matrix X used above, the arguments are very similar.

Step 3. The proof of (16) is analogous to Step 3 of the proof of Theorem 1. We leave out the details. ■

2.3 Barron Classes

Barron classes, which were first introduced by Barron (1993), turned out to be a very promising function class for the performance analysis of artificial neural networks for solving high-dimensional problems. In particular, if μ is a probability measure on a d-dimensional ball and f is a function from the 'original' Barron class (see (19) below with $s = 1$), Barron (1993) provides a randomized construction of a shallow neural network with one hidden layer of N neurons, which achieves an approximation of f with accuracy of the order $N^{-1/2}$ in the $L_2(\mu)$-norm. Remarkably, this approximation rate is independent of the dimension d even though the Barron class is so large that every linear approximation method for it suffers the curse of dimensionality (Barron, 1993, Theorem 6). These results were later extended and generalized in various directions, see e.g. Caragea et al. (2023); E and Wojtowytsch (2020b) and the references given therein.

We start again by introducing the (Fourier-analytic version of) Barron classes and an alternative based on the Riesz basis $\mathcal{R}_d$ of L_2. We will see below that, again, those spaces coincide with equivalent norms. Let $s \geq 0$ and define

$$\mathbb{B}^s([0,1]^d) := \left\{ f : f \text{ given by (13)}, \|f\|_{\mathbb{B}^s} := |a_0| + \sum_{m +\!\!\!> 0} \|m\|_2^s \cdot (|a_m| + |b_m|) < \infty \right\},$$

and

$$\mathcal{B}^s([0,1]^d) := \left\{ f : f \text{ given by (14)}, \|f\|_{\mathcal{B}^s} := |\alpha_0| + \sum_{k \mathrel{+\!\!\!>} 0} \|k\|_2^s \cdot (|\alpha_k| + |\beta_k|) < \infty \right\}.$$

Here, the convergence of (13) and (14) is always understood in $L_2([0,1]^d)$. Note that in contrast to the Sobolev classes from the previous section, we now invoke the ℓ_1-norm instead of the ℓ_2-norm when defining the Barron classes. For a discussion of other notions of Barron spaces we refer to Remark 4 below.

We now show that the Barron-type spaces above have equivalent norms.

Theorem 3. *Let $0 \leq s < 1$. Then $\mathbb{B}^s([0,1]^d) = \mathcal{B}^s([0,1]^d)$ in the sense of equivalent norms. The constants of equivalence of these norms depend on s but are independent of d. Moreover, for constants $c, C > 0$, which can be chosen independently of d, it holds that*

$$c \sum_{k \mathrel{+\!\!\!>} 0} (|\alpha_k| + |\beta_k|) \;\leq\; \left\| \sum_{k \mathrel{+\!\!\!>} 0} \|k\|_2^{-s} \Big(\alpha_k \mathcal{C}_k + \beta_k \mathcal{S}_k \Big) \right\|_{\mathbb{B}^s} \;\leq\; C \sum_{k \mathrel{+\!\!\!>} 0} (|\alpha_k| + |\beta_k|). \tag{17}$$

Modifications of (17) *for functions with constant term are obvious.*

Some authors call the system (15) a 1-Riesz basis of $\mathbb{B}^s$ if it satisfies (17), see e.g. Christensen and Stoeva (2003).

Proof *Step 1.* First, we show that $\mathbb{B}^s([0,1]^d) \hookrightarrow \mathcal{B}^s([0,1]^d)$. We proceed similarly as in the proof of Theorem 2. The main difference is now that $\mathbb{B}^s([0,1]^d)$ and $\mathcal{B}^s([0,1]^d)$ are weighted ℓ_1-spaces and not Hilbert spaces.

We shall use the following classical fact. If $A = (a_{u,v})_{u,v=1}^{\infty}$ is a double sequence of real numbers and if

$$M := \sup_{v=1,2,\dots} \sum_{u=1}^{\infty} |a_{u,v}| < \infty,$$

then $A : \ell_1 \to \ell_1$ given by $(Ax)_u = \sum_{v=1}^{\infty} a_{u,v} x_v$ is well-defined, linear and bounded. Moreover, its operator norm is equal to $M = \sup_{v=1,2,\dots} \|Ae_u\|_1$. Indeed, if $M < \infty$, then for every $x \in \ell_1$, we obtain

$$\begin{aligned} \|Ax\|_1 = \sum_{u=1}^{\infty} |(Ax)_u| &= \sum_{u=1}^{\infty} \left| \sum_{v=1}^{\infty} a_{u,v} x_v \right| \\ &\leq \sum_{v=1}^{\infty} \sum_{u=1}^{\infty} |a_{u,v}| \cdot |x_v| \leq \|x\|_1 \cdot M \end{aligned}$$

showing that $\|A\| \leq M$. The reverse inequality follows from the fact that the canonical vectors e_j have unit norm in ℓ_1.

Let $\omega = (\omega_m)_{m \mathrel{+\!\!\!>} 0}$ be a sequence given by $\omega_m = \|m\|_2^s$. The norm of the sequence $\alpha = (\alpha_m)_{m \mathrel{+\!\!\!>} 0}$ in the weighted space $\ell_1(\omega)$ is then given as usual by

$$\|\alpha\|_{\ell_1(\omega)} = \sum_{m \mathrel{+\!\!\!>} 0} \omega_m \cdot |\alpha_m|.$$

Repeating the argument of the proof of Theorem 2, we reduce the proof of the embedding $\mathbb{B}^s([0,1]^d) \hookrightarrow \mathcal{B}^s([0,1]^d)$ to the estimate of the operator norm of $T : \ell_1(\omega) \to \ell_1(\omega)$, where, for $k \mathrel{+\!\!\!\triangleright} 0$,

$$\alpha_k = (Ta)_k = \sum_{\substack{m \mathrel{+\!\!\!\triangleright} 0; \ell=0,1,\dots \\ (2\ell+1)m=k}} a_m \cdot \frac{\mu(2\ell+1)}{(2\ell+1)^2}.$$

Observe, that T can be represented by an infinite matrix $T = (T_{k,m})_{k,m \mathrel{+\!\!\!\triangleright} 0}$ with

$$\begin{aligned} T_{k,m} = (Te_m)_k &= \sum_{(\ell,n):(2\ell+1)n=k} (e_m)_n \cdot \frac{\mu(2\ell+1)}{(2\ell+1)^2} \\ &= \sum_{\substack{\ell=0,1,\dots \\ (2\ell+1)m=k}} \frac{\mu(2\ell+1)}{(2\ell+1)^2} = \begin{cases} 0, & \text{if there is no } \ell \geq 0 \text{ with } (2\ell+1)m = k, \\ \frac{\mu(2\ell+1)}{(2\ell+1)^2}, & \text{if } (2\ell+1)m = k. \end{cases} \end{aligned}$$

Similarly to the argument presented above (with A replaced by T) we can again consider only the action of T on the canonical basis, i.e.,

$$\|T\|_{\ell_1(\omega)\to\ell_1(\omega)} = \sup_{m \mathrel{+\!\!\!\triangleright} 0} \frac{\|Te_m\|_{\ell_1(\omega)}}{\|e_m\|_{\ell_1(\omega)}} = \sup_{m \mathrel{+\!\!\!\triangleright} 0} \frac{\|Te_m\|_{\ell_1(\omega)}}{\|m\|_2^s}.$$

Hence,

$$\begin{aligned} \|m\|_2^{-s}\|Te_m\|_{\ell_1(\omega)} &= \|m\|_2^{-s} \sum_{\ell=0}^{\infty} \|(2\ell+1)m\|_2^s \cdot \frac{|\mu(2\ell+1)|}{(2\ell+1)^2} \\ &= \sum_{\ell=0}^{\infty} \frac{|\mu(2\ell+1)|}{(2\ell+1)^{2-s}}, \end{aligned} \tag{18}$$

which is finite for $0 \leq s < 1$ and, of course, independent on d.

Step 2. The reverse embedding $\mathcal{B}^s([0,1]^d) \hookrightarrow \mathbb{B}^s([0,1]^d)$ can be shown in nearly the same way.

Step 3. The proof of (17) follows again Step 3 of the proof of Theorem 1. ■

Remark 3. It can be shown that the series (18) converges only for $s < 1$. Indeed, by Equation (1.2.7) in Titchmarsh (1986), it is known that the series

$$\sum_{n=1}^{\infty} \frac{|\mu(n)|}{n^\alpha}$$

converges only for $\alpha > 1$. On the other hand,

$$\begin{aligned} \sum_{n=1}^{\infty} \frac{|\mu(n)|}{n^\alpha} &= \sum_{n=0}^{\infty} \frac{|\mu(2n+1)|}{(2n+1)^\alpha} + \sum_{n=0}^{\infty} \frac{|\mu(4n+2)|}{(4n+2)^\alpha} + \sum_{n=0}^{\infty} \frac{|\mu(4n+4)|}{(4n+4)^\alpha} \\ &= \Big(1 + \frac{1}{2^\alpha}\Big) \sum_{n=0}^{\infty} \frac{|\mu(2n+1)|}{(2n+1)^\alpha}, \end{aligned}$$

where we used that $4(n+1)$ is divisible by 4, implying $\mu(4n+4) = 0$ and $\mu(4n+2) = -\mu(2n+1)$ for every $n \geq 0$. Therefore, (18) converges only when $2-s > 1$ and the restriction to $0 \leq s < 1$ in Theorem 3 is necessary.

Remark 4. Let us comment on the different notions of Barron spaces in the literature. Our Barron spaces $\mathbb{B}^s$ are in the spirit of the "classical" Fourier-analytic notion of Barron spaces of periodic functions as introduced by Barron (1993), see Example 16 in Section IX therein, at least for $s = 1$. However, it seems more common to work with the classes of functions on $\mathbb{R}^d$, or on certain subsets Ω, which are defined by

$$\mathcal{B}^s_{\mathrm{ext}}(\Omega) := \left\{ f : \Omega \to \mathbb{C} : \inf_{f_e|_\Omega = f} \int_{\mathbb{R}^d} (1 + \|\xi\|_2)^s |\hat{f}_e(\xi)| d\xi < \infty \right\}, \tag{19}$$

where the infimum is taken over all extensions $f_e \in L_1(\mathbb{R}^d)$ and $\hat{f}_e$ denotes the Fourier transform of f_e. Barron (1993) originally introduced this class for $s = 1$, and showed that functions from this class can be efficiently approximated by neural networks. However, we stress that the norm (19) is dimension-dependent in a non-transparent way: It is already a non-trivial research question to determine the norm of a constant function (for bounded Ω). See also Caragea et al. (2023); Siegel and Xu (2021b) for a discussion of issues with general domains.

Another notion of Barron-type spaces that has been proposed in the literature, see e.g. E et al. (2022); E and Wojtowytsch (2022), is the space of all "infinitely wide" neural networks with a certain control over the network parameters. More formally, given an activation function σ (which is often either the ReLU or a Heaviside function), the elements of the associated Barron-type space are all functions that can be written as

$$f(x) = \int_{\mathbb{R} \times \mathbb{R}^d \times \mathbb{R}} a \cdot \sigma(\langle w, x \rangle + b) \, d\mu(a, w, b),$$

where μ is a probability measure satisfying

$$\|f\|_{B(\sigma)} := \int_{\mathbb{R} \times \mathbb{R}^d \times \mathbb{R}} |a| \cdot \sigma(\|w\|_2 + |b|) \, d\mu(a, w, b) < \infty. \tag{20}$$

Note that this space/norm is automatically adapted to the activation function σ. It was further generalized by Li et al. (2024), where ℓ_p norms and ReLU^k activation functions were considered in (20). Let us point here that already Barron (1993) had shown that the Fourier-analytic Barron space according to (19) is embedded in the Barron space $B(H)$ associated to the Heaviside function H, i.e., for bounded domains Ω it holds that $\mathcal{B}^1_{\mathrm{ext}}(\Omega) \hookrightarrow B(H)$. On the other hand, when considering the ReLU activation function σ, we only have $\mathcal{B}^2_{\mathrm{ext}}(\Omega) \hookrightarrow B(\sigma)$, see Caragea et al. (2023, Lemma 7.1). The latter embedding is best possible as was shown by Caragea et al. (2023, Prop. 7.4). We also refer to E et al. (2022, Section 2), E and Wojtowytsch (2022, Section 3) or Klusowski and Barron (2018) in this context.

3. Approximation of Functions by Neural Networks

In this section we exploit the properties of the Riesz basis (15) derived in Section 2 to address the question how to approximate the elements of the Sobolev classes $W^s([0,1]^d)$

and Barron classes $\mathbb{B}^s([0,1]^d)$ with $0 < s < 1$ by artificial neural networks. In order to derive our results, we use the fact that our established (Riesz) basis $\mathcal{R}_d$ admits us to represent functions from the Sobolev or Barron classes in the form

$$f(x) = \alpha_0 + \sum_{k \mathrel{+\!\!\!\!>} 0} \Big[\alpha_k \mathcal{C}_k(x) + \beta_k \mathcal{S}_k(x)\Big], \quad x \in [0,1]^d.$$

Then, using Daubechies et al. (2022, Theorem 6.2) for $d = 1$ and Schneider and Vybíral (2024, Theorem 4.6) for higher dimensions, one can reproduce linear combinations of $\mathcal{C}_k$ and $\mathcal{S}_k$ via ReLU networks with a good control of the depth L. For convenience of the reader, we provide the basic constructions in Subsection 3.1 below. With these constructions at hand, we then proceed in Subsection 3.2 with the approximation of functions from the Sobolev classes $W^s([0,1]^d)$. Our main result is stated in Theorem 8 and uses standard estimates from linear approximation. In particular, we approximate our functions using linear combinations with indices corresponding to the integer lattice points in a d-dimensional ball of radius R centered in the origin. For this we need to establish upper bounds on the number of these integer lattice points, see Lemma 7 (which is new and of interest on its own).
Regarding Barron classes, which we treat in Subsection 3.3, our main result is stated in Theorem 12. In this context, the *architecture* of the neural network depends on the individual functions and is based on the best n-term approximation in the *dictionary* $\mathcal{R}_d$. As such, and based on our analysis, the error bounds can again be deduced by rather classical methods from estimates in a sequence space setting.

3.1 Reproducing the Riesz Basis with Neural Networks

We first fix some notation. A function $f : \mathbb{R}^{n_1} \to \mathbb{R}^{n_2}$ is called affine, if it can be written as $f(x) = Mx + b$, where $M \in \mathbb{R}^{n_2 \times n_1}$ is a matrix and $b \in \mathbb{R}^{n_2}$. This means that f is given by $(n_1 + 1)n_2$ real parameters. In the setting of artificial neural networks, M is usually called the weight matrix and b the bias vector. The following definition formalizes the notion of ReLU neural networks with width W and depth L, cf. Figure 1. For further information regarding the basics of neural network architectures we refer to the survey article by DeVore et al. (2021) as well as the recent book of Petersen and Zech (2024) and the references therein.

Definition 4. Let d, W, L be positive integers. Then a feed-forward artificial neural network $\mathcal{N}$ with the ReLU activation function width W, and depth L is given by a collection of $L+1$ affine mappings $A^{(0)}, \dots, A^{(L)}$, where $A^{(0)} : \mathbb{R}^d \to \mathbb{R}^W$, $A^{(j)} : \mathbb{R}^W \to \mathbb{R}^W$ for $j = 1, \dots, L-1$ and $A^{(L)} : \mathbb{R}^W \to \mathbb{R}$, and generates a function $\mathcal{N} \colon \mathbb{R}^d \to \mathbb{R}$ of the form

$$A^{(L)} \circ \mathrm{ReLU} \circ A^{(L-1)} \circ \cdots \circ \mathrm{ReLU} \circ A^{(0)}.$$

We denote by $\Upsilon_d^{W,L}$ the set of all such neural networks.

Counting all the weights and biases of a fully connected network from $\Upsilon_d^{W,L}$, we see that it has

$$W(d+1) + (L-1)W(W+1) + W + 1 = \mathcal{O}(W^2L + dW) \tag{21}$$

parameters. When $d = 1$ this reduces to $\mathcal{O}(W^2L)$, see also Daubechies et al. (2022, Remark 6.1). For a neural network, we call the graph of the connections between its artificial

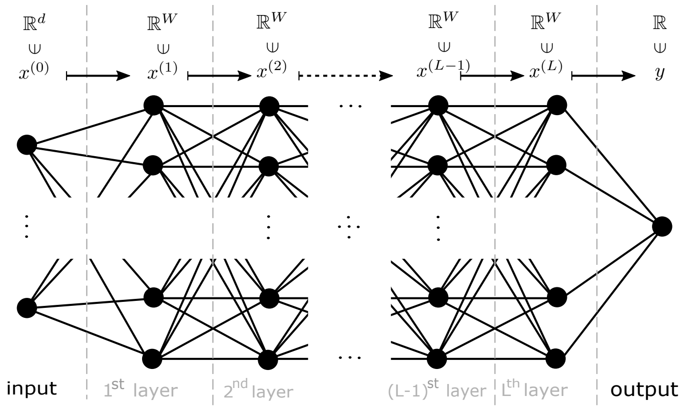

Figure 1: Feed-forward ReLU network with depth L and width W

neurons (as shown in Figure 1 for $\Upsilon_d^{W,L}$) its *architecture*. In general, a neural network could have a different number of artificial neurons in each layer, but we restrict ourselves to the architecture of $\Upsilon_d^{W,L}$ to simplify the presentation. It is obvious that every function $\mathcal{N}$ generated by such a feed-forward ReLU network is a continuous piecewise affine function on $\mathbb{R}^d$. When approximating functions from the Sobolev or Barron classes we now use the fact that these functions admit a representation of the form (14), i.e.,

$$f(x) = \alpha_0 + \sum_{k \mathrel{+\!\!\!>} 0} \Big[\alpha_k \mathcal{C}_k(x) + \beta_k \mathcal{S}_k(x)\Big], \quad x \in [0,1]^d.$$

Moreover, as was shown by Daubechies et al. (2022, Theorem 6.2) for $d = 1$ and Schneider and Vybíral (2024, Theorem 4.6) for higher dimensions, one can reproduce linear combinations of $\mathcal{C}_k$ and $\mathcal{S}_k$ via ReLU networks with a good control of the depth L. This makes the representation (14) an ideal starting point for approximating f using neural networks.

Let us briefly sketch the construction. The basic idea is to start with a representation of the hat function $H : [0,1] \to \mathbb{R}$ using the ReLU function, i.e.,

$$H(x) = \begin{rcases}\begin{cases} 2x, & 0 \le x \le \frac{1}{2} \\ 2(1-x), & \frac{1}{2} < x \le 1 \end{cases}\end{rcases} \;=\; (2,-4) \cdot \mathrm{ReLU}\left(\begin{pmatrix}1\\1\end{pmatrix} x + \begin{pmatrix}0\\-\frac{1}{2}\end{pmatrix}\right). \tag{22}$$

From the right hand side of (22) we deduce that H is the realization of a neural network with width $W = 2$ and one hidden layer $L = 1$, as illustrated in Figure 2, i.e., $H \in \Upsilon_1^{2,1}$.

We will realize our Riesz basis $\mathcal{R}_d$ by compositions and sums of these simple networks. For this, let us first mention that for functions that are represented by neural networks, say, with width W_i and depth L_i, $i = 1, \dots, k$, we can represent their composition by a neural network with width $\max_i\{W_i\}$ and depth $\sum_{i=1}^k L_i$, by just concatenating the neural networks.

Now, $\mathcal{C} = 1 - 2H$, with $\mathcal{C}$ from (1), can be written as

$$\mathcal{C}(x) \;=\; (-4, 8) \cdot \mathrm{ReLU}\left(\begin{pmatrix}1\\1\end{pmatrix} x + \begin{pmatrix}0\\-\frac{1}{2}\end{pmatrix}\right) \;+\; 1,$$

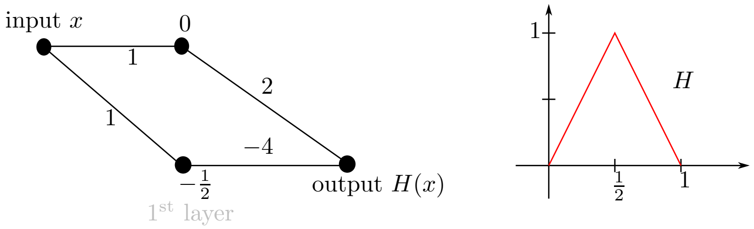

Figure 2: Neural network representing H and usual graph associated with H

which has the same architecture as H with width $W = 2$ and depth $L = 1$. Using $\mathcal{S}(x) = \mathcal{C}_2(x/2 + 3/8)$, we see that $\mathcal{S}$ can be represented using a neural network with the same architecture as $\mathcal{C}_2$, i.e., with $W = L = 2$. The univariate scaled versions $\mathcal{C}_k, \mathcal{S}_k$ are obtained using compositions of hat functions, e.g., it holds $\mathcal{C}_k(x) = \mathcal{C}(H(\dots(H(x))))$ for $k = 2^m$ where H is composed m-times. The general, multivariate case is obtained by a suitable scaling argument that ultimately leads to $\mathcal{C}_k, \mathcal{S}_k \in \Upsilon_d^{2,L}$ with $L := \lceil \log_2 (\|k\|_1) \rceil + 3 \leq \log_2 (\|k\|_1) + 4$, $k \in \mathbb{Z}^d$, see Schneider and Vybíral (2024, Lemma 4.4).

For linear combinations of these functions it remains to sum the individual terms, and in this matter we have some freedom, which is reflected by the two possibilities described in Lemma 5 below. Since the individual terms can be computed separately, we could stack them on top of each other leading to a neural network with width $\sum_{i=1}^N W_i$ and depth $\max_i\{L_i\}$. However, we could also do the summation *in line*, as in Figure 4, by using two extra channels (a source and a collation channel). This leads to a neural network with width $\max_i\{W_i\} + d + 1$ and depth $\sum_{i=1}^N L_i$. One may also mix both strategies, by arranging the summands in an array. We refer to Daubechies et al. (2022, Theorem 6.2) and Schneider and Vybíral (2024, Theorem 4.6) for further details on the constructions.

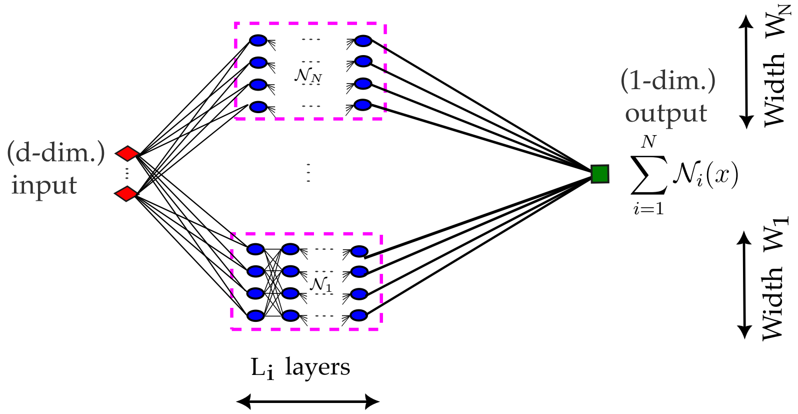

Figure 3: Neural network representation of the sum of functions $\mathcal{N}_1, \dots, \mathcal{N}_N$ according to Lemma 5 (i)

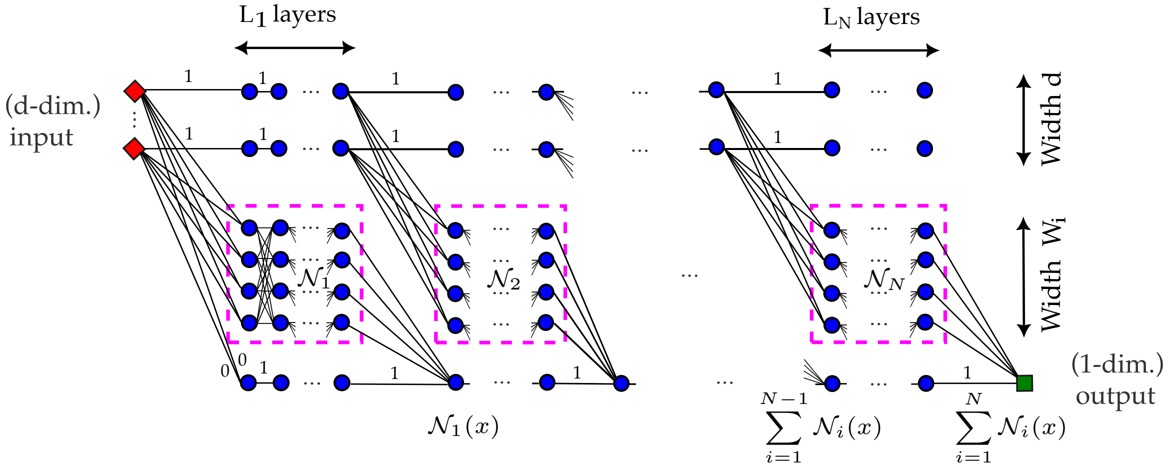

Figure 4: Neural network representing the sum of functions $\mathcal{N}_1,\dots,\mathcal{N}_N$ according to Lemma 5 (ii)

Lemma 5. *Let* $\mathcal{N}_1,\dots,\mathcal{N}_N:\mathbb{R}^d\to\mathbb{R}$ *with* $\mathcal{N}_i\in\Upsilon_d^{W_i,L_i}$. *Then*

(i)

$$\mathcal{N}_1+\dots+\mathcal{N}_N\in\Upsilon_d^{W,L}\quad \textit{with}\quad W=W_1+\dots+W_N\quad\textit{and}\quad L=\max_{i=1,\dots,N}L_i$$

and

(ii)

$$\mathcal{N}_1+\dots+\mathcal{N}_N\in\Upsilon_d^{W,L}\quad \textit{with}\quad W=\max_{i=1,\dots,N}W_i+d+1\quad\textit{and}\quad L=L_1+\dots+L_N.$$

Proof It is easy to observe, cf. also Elbrächter et al. (2021, Lemma 2.4), that the classes $\Upsilon_d^{W,L}$ are monotone in W and L. Indeed, if $W,L\ge 2$ and $\mathcal{N}\in\Upsilon_d^{W,L-1}$, then $\mathcal{N}\in\Upsilon_d^{W,L}$. To show that, let us consider a representation of $\mathcal{N}$ as described in Definition 4 with affine mappings $A^{(0)},\dots,A^{(L-1)}$. Then we define $\tilde A^{(L-1)}(z)=(A^{(L-1)}z,-A^{(L-1)}z)\in\mathbb{R}^{2W}$ and $\tilde A^{(L)}(u,v)=u-v$. Then it is easy to check that the neural network with affine mappings $A^{(0)},\dots,A^{(L-2)},\tilde A^{(L-1)},\tilde A^{(L)}$ represents $\mathcal{N}$ again. Similarly, if $W,L\ge 1$ and $\mathcal{N}\in\Upsilon_d^{W-1,L}$, we can extend the affine mappings in the representation of $\mathcal{N}$ by zero and show that $\mathcal{N}\in\Upsilon_d^{W,L}$.

Therefore, we can assume in the proof of (i) that $L_1=\dots=L_N=L_{\max}$ and in the proof of (ii) that $W_1=\dots=W_N=W_{\max}$.

The proof of (i) follows by taking the architecture of Figure 3, where we stack the representations of $\mathcal{N}_1,\dots,\mathcal{N}_N$ upon each other, cf. Elbrächter et al. (2021, Lemma II.6). To be more specific, if $\mathcal{N}_i$ is given by the affine mappings $A_i^{(0)},\dots,A_i^{(L)}$, then we define $B^{(0)}(z)=(A_1^{(0)}(z)^T,\dots,A_N^{(0)}(z)^T)^T\in\mathbb{R}^W$ for $z\in\mathbb{R}^d$. Furthermore, we put $B^{(j)}(w_1,w_2,\dots)=$

$(A_1^{(j)}(w_1,\dots,w_{W_1})^T, A_2^{(j)}(w_{W_1+1},\dots,w_{W_1+W_2})^T,\dots)^T$ for $j = 1,\dots,L-1$ and, finally, $B^{(L)}(w_1,w_2,\dots) = A_1^{(L)}(w_1,w_2,\dots,w_{W_1}) + A_2^{(L)}(w_{W_1+1},\dots,w_{W_1+W_2}) + \dots$. Then the neural network with affine mappings $B^{(0)}, B^{(1)},\dots,B^{(L)}$ represents $\mathcal{N}_1 + \dots + \mathcal{N}_N$.

The proof of (ii) for $d = 1$ is covered by Daubechies et al. (2022, Proposition 4.2) and involved one *source channel*, which shifts the inputs without the application of the ReLU function and one *collation channel*, which is also ReLU free and which sums up the partial results. The generalization to $d > 1$ is straightforward, using d source channels. We refer to Figure 4 for a sketch of the architecture of the resulting neural network. ■

Remark 5. Let us mention that if the inputs are all positive, then the use of the ReLU functions in the source channels is possible, but makes no difference.

This results in the following architecture for partial sums of (14), see Schneider and Vybíral (2024, Theorem 4.6), where also the upper bound on the size of the weights is discussed in detail. Note that adding the absolute term α_0 can be incorporated into the bias term of the last affine mapping of the neural network representing f, without increasing W or L. Note also that we replace the formula $L = 3 + \lceil \log_2(\|k\|_1)\rceil$ of Schneider and Vybíral (2024) by the (slightly larger) $L \le 4 + \log_2(\|k\|_1)$ to avoid the use of the ceiling function.

Lemma 6. *Let $d \ge 1$ and let $I \subset \mathbb{Z}^d \setminus \{0\}$ be nonempty. Then,*

$$f(x) \;=\; \alpha_0 + \sum_{k\in I}\Big[\alpha_k \mathcal{C}_k(x) + \beta_k \mathcal{S}_k(x)\Big], \quad x \in [0,1]^d, \tag{23}$$

satisfies $f \in \Upsilon_d^{W,L}$ with either

$$W = 4\cdot \#I \qquad \textit{and} \qquad L \;\le\; 4 + \log_2\left(\max_{k\in I}\|k\|_1\right), \tag{24}$$

or

$$W = d+3 \qquad \textit{and} \qquad L \;\le\; 2\,\#I \cdot \log_2\left(16\cdot \max_{k\in I}\|k\|_1\right). \tag{25}$$

Moreover, all weights in the corresponding neural network are bounded by $8\cdot\max_{k\in I}\{1, |\alpha_k|, |\beta_k|\}$.

Remark 6. The decomposition (23) features $2\,\#I + 1$ real parameters, namely $\alpha_0, (\alpha_k)_{k\in I}$ and $(\beta_k)_{k\in I}$. The neural networks constructed in Lemma 6 combine these parameters with pre-cached building blocks (i.e., neural networks, which reproduce the functions $\mathcal{C}_k$ and $\mathcal{S}_k$ for $k \in I$) and additional information passing channels. In this sense, these $2\,\#I + 1$ parameters are the only parameters, which we change in the proposed architecture, the other weights are fixed and do not depend on f. This approach was used by Elbrächter et al. (2021, Section VII) under the name *transference principle* for general decompositions of functions into a given *dictionary*. However, in the next section we take advantage of the following two facts: First, we exploit that the elements of the Riesz basis (15) can be easily and exactly recovered by small neural networks. And secondly, the elements of (15) do not rely on any localized decomposition of unity, which would lead to exponential dependence of the constants on the underlying dimension d.

If we count all nonzero parameters of the architecture, then we can employ the fact that the widths and depths of the small neural networks are at most 4 and $M := 1+\log_2[\max_{k\in I} \|k\|_1]$, respectively. Hence, there are at most $\mathcal{O}(\#I \cdot \max(d, M))$ non-zero weights in the neural network from the first part of Lemma 6, and $\mathcal{O}(d \cdot \#I \cdot M)$ non-zero weights in the second architecture.

In contrast, we know from (21) that the number of parameters in a (general, fully-connected) neural network from $\Upsilon_d^{W,L}$, with W and L chosen as above, is of the order $\mathcal{O}((\#I)^2 \cdot M + \#I \cdot d)$ for the first, and $\mathcal{O}(d^2 \cdot \#I \cdot M)$ for the second architecture. Note that in the second construction, most of the weights are actually equal to 1 due to the additional collation channel.

To ease the presentation, we will only employ the first construction (24) for the approximation results below. The modifications for the second construction (25) are straightforward.

3.2 Sobolev Classes

We start with approximation of functions from Sobolev classes. We use Theorem 2 to show that for every $d \geq 1$, $0 < s < 1$, and $\varepsilon > 0$, we can design a fixed neural network architecture, such that for every $f \in W^s([0,1]^d)$ we can choose the weights of this neural network in such a way that it approximates f in $L_2([0,1]^d)$ up to the error $\varepsilon\|f\|_{W^s}$.

The use of the Riesz basis $\mathcal{R}_d$ from (3) allows us to avoid decompositions of unity and Taylor's theorem, which were used frequently in the analysis of approximation properties of neural networks, see, e.g., Yarotsky (2017, Theorem 1) or Siegel (2023). This allows to keep track of the dependence of the approximation error on the dimension. Moreover, we achieve a (nearly) optimal number of layers, their widths and the number of weights used. This is shown by comparison with available upper and lower bounds.

We shall rely on an upper bound of the number of integer lattice points in a d-dimensional ball of radius t centered in the origin. The exact behavior of this quantity represents a classical problem in number theory, known as the *Gauss circle problem.*

For an integer $d \geq 1$ and a real number $t \geq 0$, we put

$$Z(t,d) := \{k \in \mathbb{Z}^d : \|k\|_2 \leq t\} = \mathbb{Z}^d \cap tB_2^d,$$

where B_2^d is the unit ball in $\mathbb{R}^d$ and tB_2^d is its t-multiple. Furthermore, we denote

$$N(t,d) := \#Z(t,d). \tag{26}$$

Much is known in the asymptotic regime where d is fixed and t tends to infinity. For example, if $d = 2$, then $N(t,2)$ is approximated by the area of the circle πt^2

$$N(t,2) = \pi t^2 + E(t)$$

for some error term $E(t)$. Gauss managed to prove $|E(t)| \leq 2\sqrt{2}\pi t$ in this context. It is conjectured that $|E(t)| = O(t^{1/2+\varepsilon})$ for every $\varepsilon > 0$. On the other hand, it was established independently by Landau and Hardy that the statement fails for $\varepsilon = 0$.

Here, we are interested in an upper bound, which does not have to be so sharp, but which is valid also in the non-asymptotic regime, i.e., which holds for all $d \in \mathbb{N}$ and $t \geq 0$. We provide such a bound in Lemma 7 below. Let us remark, that one could obtain a weaker bound by exploiting the well-known results for entropy numbers of embeddings of finite-dimensional sequence spaces, see Kühn et al. (2016) for a similar approach.

Let us note that

$$N(t,d) = 1$$

for $0 \leq t < 1$ and

$$N(t,d) = 2d+1$$

for $1 \leq t < \sqrt{2}$.

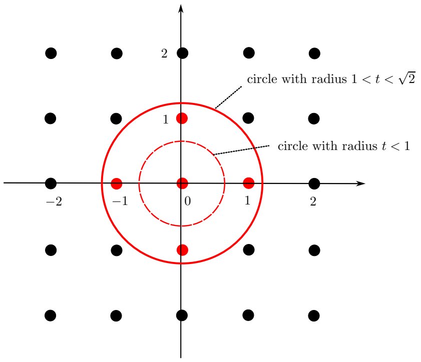
circle with radius $1 < t < \sqrt{2}$
circle with radius $t < 1$
2
1
-2
-1
0
1
2

Figure 5: Illustration of $N(t,2)$

Lemma 7. *There exist two absolute constants $c_1, c_2 > 0$ such that for every $d \geq 1$ and $t \geq 0$ the following holds.*
(i) If $t \geq \sqrt{d}/2$, then

$$N(t,d) \leq \left(\frac{c_1 t}{\sqrt{d}}\right)^d.$$

(ii) If $0 < t \leq \sqrt{d}/2$, then

$$N(t,d) \leq \left(\frac{c_2 d}{t^2}\right)^{t^2}.$$

Proof If $t \geq \sqrt{d}/2$, then

$$Z(t,d) + [0,1]^d \subset (t+\sqrt{d})B_2^d, \tag{27}$$

see Figure 6, and

$$\begin{aligned} N(t,d) &= \mathrm{vol}(Z(t,d)+[0,1]^d) \leq (t+\sqrt{d})^d \,\mathrm{vol}(B_2^d) \\ &= (t+\sqrt{d})^d \frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)} \leq \left(\frac{c_1 t}{\sqrt{d}}\right)^d. \end{aligned}$$

Using Stirling's formula, cf. Equation (3.9) in Artin (2015), it can be checked that $c_1 := 3\sqrt{2\pi e}$ works.

If $t \leq \sqrt{d}/2$, the proof is more delicate. Recall that, if $0 < t < 1$, then $N(t,d) = 1$ and the result follows. If $1 \leq t < \sqrt{2}$, then $N(t,d) = 2d+1$ and the upper bound again follows. This covers the cases $1 \leq d \leq 7$.

Next, we consider the case when $d \geq 8$ and $\sqrt{d-1}/2 \leq t \leq \sqrt{d}/2$. Then, by (i),

$$N(t,d) \leq N(\sqrt{d}/2, d) \leq (c_1/2)^d \leq (c_2 d/t^2)^{t^2},$$

where we used also that $4 \leq d/t^2 \leq 4 \cdot \frac{d}{d-1} \leq 5$.

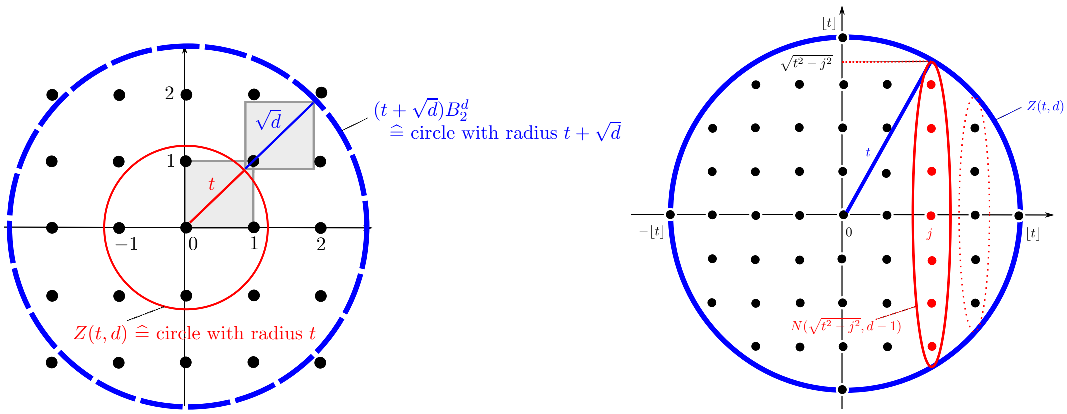

Figure 6: Visualization of (27), and splitting of $Z(t,2)$ into disks with $|k_1| \leq \lfloor t\rfloor$

If $d \geq 8$ and $\sqrt{2} \leq t \leq \sqrt{d-1}/2$, we proceed by induction.
We split

$$Z(t,d) = \{k \in \mathbb{Z}^d : \|k\|_2 \leq t\} = \bigcup_{j=-\lfloor t\rfloor}^{j=\lfloor t\rfloor} \{k \in \mathbb{Z}^d : \|k\|_2 \leq t,\ k_1 = j\},$$

see Figure 6, and obtain

$$\begin{aligned} N(t,d) &= \sum_{j=-\lfloor t\rfloor}^{j=\lfloor t\rfloor} N(\sqrt{t^2-j^2}, d-1) = N(t,d-1) + 2\sum_{j=1}^{j=\lfloor t\rfloor} N(\sqrt{t^2-j^2}, d-1) \qquad (28)\\ &\leq N(t,d-1) + 2\sum_{j=1}^{j=\lfloor t\rfloor}\left(\frac{c_2(d-1)}{t^2-j^2}\right)^{t^2-j^2} \leq \left(\frac{c_2(d-1)}{t^2}\right)^{t^2} + 4\left(\frac{c_2(d-1)}{t^2-1}\right)^{t^2-1}, \end{aligned}$$

where we used inductively (ii) for $d-1$ combined with the fact that $t \leq \sqrt{d-1}/2$. Furthermore, to estimate the sum, we used that every term in this sum is at least twice as large as the next term (for c_2 large enough). Indeed, the ratio of the $(j+1)$-st and the j-th term in (28) is equal to

$$\left[\frac{t^2-j^2}{c_2(d-1)}\right]^{2j+1}\left(1+\frac{1}{\lambda_j}\right)^{\lambda_j(2j+1)}, \quad \text{where} \quad \lambda_j = \frac{t^2-(j+1)^2}{2j+1}. \qquad (29)$$

Using the elementary inequality $(1+1/z)^z \leq e$ valid for $z > 0$, we observe that (29) is smaller than $1/2$ for $c_2 > 0$ large enough. Therefore, taking the largest term with $j = 1$ out of the sum, we can bound the sum by a geometric series with the sum $\frac{1}{1-\frac{1}{2}}$, which in the end gives the additional factor 2 in the last step in (28).

Finally, for $c_2 \geq 4e$, we apply the mean value theorem to the function $f(u) = u^{t^2}$ and obtain

$$\begin{aligned} 4\left(\frac{c_2(d-1)}{t^2-1}\right)^{t^2-1} &= \frac{4}{c_2} \cdot \left(1 + \frac{1}{t^2-1}\right)^{t^2-1} \cdot \left(\frac{c_2}{t^2}\right)^{t^2} \cdot t^2(d-1)^{t^2-1} \\ &\leq \frac{4e}{c_2} \cdot \left(\frac{c_2}{t^2}\right)^{t^2} [d^{t^2} - (d-1)^{t^2}] \leq \left(\frac{c_2 d}{t^2}\right)^{t^2} - \left(\frac{c_2(d-1)}{t^2}\right)^{t^2}. \end{aligned}$$

Together with (28) we conclude that

$$N(t,d) \leq \left(\frac{c_2 d}{t^2}\right)^{t^2}.$$

■

The last combinatorial lemma directly leads to our first approximation result using neural networks. Note that the architecture and the actual neural network mentioned in the theorem is given explicitly, by means of partial sums of (14), see Lemma 6.

Theorem 8. *Let $0 < s < 1$. Then there is a constant $C_s > 0$, depending only on s, such that for every $d \in \mathbb{N}$ and $0 < \varepsilon < 1$ the following statement is true.*
Let $R := (C_s/\varepsilon)^{1/s}$ and $N(R,d)$ be as in Lemma 7. Then, for every $f \in W^s([0,1]^d)$ there is a neural network $\mathcal{N} \in \Upsilon_d^{W,L}$ with

$$W = 4 \cdot N(R,d) \quad \text{and} \quad L \leq 4 + \log_2\left(R \cdot \sqrt{\min(R,d)}\right)$$

such that

$$\|f - \mathcal{N}\|_2 \leq \varepsilon \cdot \|f\|_{W^s}.$$

Moreover, $\mathcal{N}$ can be given explicitly depending on α_k, β_k with $\|k\|_2 \leq R$ in (14) and the architecture of $\mathcal{N}$ is independent of f.

Proof Let $f \in W^s([0,1]^d)$ with $\|f\|_{W^s([0,1]^d)} \leq 1$. By Theorem 2, $f \in \mathcal{F}^s([0,1]^d)$ with $\|f\|_{\mathcal{F}^s} \leq c$ with c independent of d. Therefore, we can decompose f as in (14) with $\alpha_0^2 + \sum_{k \mathrel{+\!\!\!\!>} 0} \|k\|_2^{2s}(|\alpha_k|^2 + |\beta_k|^2) \leq c^2$.

For $R > 0$ we decompose

$$f = f_R + f^R = \Big(\alpha_0 + \sum_{k \mathrel{+\!\!\!\!>} 0 : \|k\|_2 \leq R} [\alpha_k \mathcal{C}_k(x) + \beta_k \mathcal{S}_k(x)]\Big) + \Big(\sum_{k \mathrel{+\!\!\!\!>} 0 : \|k\|_2 > R} [\alpha_k \mathcal{C}_k(x) + \beta_k \mathcal{S}_k(x)]\Big).$$

We reconstruct f_R exactly as a neural network, the error is given by f^R:

$$\|f - f_R\|_2^2 = \|f^R\|_2^2 \leq c' \sum_{k \mathrel{+\!\!\!\!>} 0 : \|k\|_2 > R} (|\alpha_k|^2 + |\beta|_k^2) \cdot \frac{\|k\|_2^{2s}}{R^{2s}} \leq C^2 R^{-2s}.$$

To make this smaller than ε^2, we choose $R := (C/\varepsilon)^{1/s}$.

By Lemma 6 with $I := Z(R,d)$, we see that $f_R \in \Upsilon_d^{W,L}$ with

$$W = 4N(R,d) \quad \text{and} \quad L = 4 + \max_{k\in Z(R,d)} \log_2(\|k\|_1) \leq 4 + \log_2\left(R \cdot \sqrt{\min(R,d)}\right).$$

In the last estimate we used Hölder's inequality together with $\|k\|_2 \leq R$ and the simple observation that the number of non-zero indices of $k \in Z(R,d)$ is bounded from above by $\min(R,d)$. ∎

Remark 7. As the proof above is based on standard estimates from linear approximation, the result can easily be generalized to approximation in other norms, like the uniform norm. However, this might lead to additional assumptions and d-dependent factors.

We now combine Theorem 8 with Lemma 7 to bound the parameters W and L of the constructed neural network by quantities involving only ε and d.

Corollary 9. *Let $0 < s < 1$, $d \in \mathbb{N}$, and $0 < \varepsilon < 1/2$. There are constants c_3, c_4, c_5 that depend only on s such that for some*

$$L \;\leq\; c_4 \log_2(1/\varepsilon)$$

and

$$W \;\leq\; \begin{cases} \left(c_4 \varepsilon^{2/s} d\right)^{c_5\,\varepsilon^{-2/s}}, & \textit{if } \varepsilon \geq c_3\, d^{-s/2}, \\ \left(\dfrac{c_4}{\varepsilon^{1/s}\sqrt{d}}\right)^d, & \textit{otherwise,} \end{cases}$$

we have

$$\inf_{\mathcal{N}\in\Upsilon_d^{W,L}} \|f - \mathcal{N}\|_2 \;\leq\; \varepsilon \cdot \|f\|_{W^s}$$

for every $f \in W^s([0,1]^d)$.

3.3 Barron Classes

When dealing with the Barron classes $\mathbb{B}^s([0,1]^d)$ we first need to select the most important terms from (14), which get reconstructed by the neural network. This is captured by the concept of best n-term approximation, a concept, which seems to go back as far as to Schmidt (1907) and which is frequently used in nonlinear approximation theory (DeVore, 1998).

Let $X \subset Y$ be two Banach spaces and let $\Phi \subset X$ be a set of elements of X. For $f \in X$ and a positive integer $n \geq 1$, we denote

$$\sigma_n(f) := \sigma_n(f, Y, \Phi) = \inf\left\{\left\| f - \sum_{j=1}^n \alpha_j \varphi_j \right\|_Y : \alpha_1, \dots, \alpha_n \in \mathbb{C}, \varphi_1, \dots, \varphi_n \in \Phi\right\}$$

and

$$\sigma_n(X, Y, \Phi) := \sup_{\|f\|_X \leq 1} \sigma_n(f).$$

Here, we only consider the case where X and Y are Banach spaces of sequences, in which case we always choose $\Phi := \{e_n\}$ the set of canonical sequences with $(e_n)_j = 1$ if $n = j$ and $(e_n)_j = 0$ otherwise, and we write $\sigma_n(X, Y)$ for $\sigma_n(X, Y, \Phi)$.

We start with the sequence spaces corresponding to univariate Barron classes.

Lemma 10. *Let $s \geq 0$ and let b^1_s be the space of bounded sequences $\alpha = (\alpha_k)_{k=1}^\infty$, for which the norm*

$$\|\alpha\|_{b^1_s} := \sum_{k=1}^\infty k^s |\alpha_k|$$

is finite. Then, $\sigma_n(b^1_s, \ell_2) \leq c\, n^{-s-1/2}$ for every $n \in \mathbb{N}$ where $c > 0$ only depends on s.

Proof First, note that $b^1_s \hookrightarrow \ell_1 \hookrightarrow \ell_2$ and that $\sigma_n(b^1_s, \ell_2)$ is well-defined. We shall use a classical result on rearrangements of sequences, see Hardy et al. (1936, Section 10.2, Theorem 368), which states that for two non-negative sequences $a = (a_j)_{j=1}^\ell$ and $b = (b_j)_{j=1}^\ell$ with non-increasing rearrangements $(a^*_j)_{j=1}^\ell$ and $(b^*_j)_{j=1}^\ell$, we have that

$$\sum_{j=1}^\ell a^*_j b^*_{\ell-j+1} \leq \sum_{j=1}^\ell a_j b_j \leq \sum_{j=1}^\ell a^*_j b^*_j.$$

Actually, using the same proof as in Hardy et al. (1936), one can show that for every non-negative sequence $a = (a_j)_{j=1}^\infty$ converging to zero and for every non-negative non-decreasing sequence $b = (b_j)_{j=1}^\infty$ it holds that

$$\sum_{j=1}^\infty a^*_j b_j \leq \sum_{j=1}^\infty a_j b_j. \tag{30}$$

Fix now $\alpha \in b^1_s$ with $\|\alpha\|_{b^1_s} \leq 1$. Then we can apply (30) to $a_j = |\alpha_j|$ and $b_j = j^s$ and obtain

$$|\alpha_\ell|^* \sum_{k=1}^\ell k^s \leq \sum_{k=1}^\ell k^s a^*_k \leq \sum_{k=1}^\infty k^s a^*_k \leq \sum_{k=1}^\infty k^s |\alpha_k| \leq 1.$$

This gives $|\alpha_\ell|^* \leq (s+1) \cdot \ell^{-s-1}$. The best n-term approximation of α in ℓ_2 is given by the n largest coefficients of α. Therefore we get

$$\sigma_n(\alpha)^2 = \sum_{k=n+1}^\infty (|\alpha_k|^*)^2 \leq (s+1)^2 \sum_{k=n+1}^\infty k^{-2s-2} \leq \frac{(s+1)^2}{2s+1} n^{-2s-1}.$$

■

The generalization to sequence spaces corresponding to Barron classes of multivariate functions follows the same pattern, but is more technical. Let $d \geq 1$ and $s \geq 0$. Then b^d_s is the space of bounded sequences $\alpha = (\alpha_k)_{k \mathrel{+\!\!\!\!>} 0}$ for which the norm

$$\|\alpha\|_{b^d_s} := \sum_{k \mathrel{+\!\!\!\!>} 0} \|k\|_2^s \cdot |\alpha_k|$$

is finite.

Lemma 11. *Let $d \geq 2$ and $s \geq 0$. Then, there are absolute constants $c_1, c_2 > 0$, and some $C_s > 0$ that depends only on s, such that*

$$\sigma_n(b_s^d, \ell_2)^2 \;\leq\; C_s \cdot \begin{cases} n^{-1}, & \textit{if } 1 \leq n \leq c_2 d, \\ \left(\frac{\log(c_2 d)}{\log(n)}\right)^s \cdot n^{-1}, & \textit{if } c_2 d \leq n \leq (c_1/2)^d, \\ d^{-s} \cdot n^{-2s/d-1}, & \textit{if } (c_1/2)^d \leq n. \end{cases} \tag{31}$$

Proof *Step 1.*
Let $\alpha \in b_s^d$ with $\|\alpha\|_{b_s^d} \leq 1$. We define the weight sequence $\omega = (\omega_k)_{k \mathrel{+\!\!\!\!\rhd} 0}$ with $\omega_k = \|k\|_2^s$. We denote by $\alpha^* = (\alpha_\ell^*)_{\ell=1}^\infty$ the one-dimensional non-increasing rearrangement of $(|\alpha_k|)_{k \mathrel{+\!\!\!\!\rhd} 0}$ and by $\omega^\# = (\omega_\ell^\#)_{\ell=1}^\infty$ the one-dimensional non-decreasing rearrangement of ω. Again, we have

$$\alpha_\ell^* \sum_{k=1}^{\ell} \omega_k^\# \leq \sum_{k=1}^{\ell} \alpha_k^* \omega_k^\# \leq \sum_{k=1}^{\infty} \alpha_k^* \omega_k^\# \leq \sum_{k \mathrel{+\!\!\!\!\rhd} 0} \omega_k |\alpha_k| \leq 1,$$

i.e. $\alpha_\ell^* \leq W(\ell, s, d)^{-1}$, where $W(\ell, s, d) = \sum_{k=1}^{\ell} \omega_k^\#$. The best n-term approximation of α in ℓ_2 is then again given by

$$\sigma_n(\alpha)^2 = \sum_{k=n+1}^{\infty} (\alpha_k^*)^2 \leq \sum_{k=n+1}^{\infty} W(k, s, d)^{-2}. \tag{32}$$

Step 2. Before we prove a lower bound for $W(k, s, d)$, we first establish lower bounds for the terms of $\omega^\#$. This we do by exploiting Lemma 7. Indeed, if $N(t, d) < n$, then also

$$\#\{k \in \mathbb{Z}^d : k \mathrel{+\!\!\!\!\rhd} 0 \text{ and } \|k\|_2 \leq t\} \leq N(t, d) < n.$$

Hence, the lattice point with n-th smallest value of ω lies outside of tB_2^d and, therefore,

$$\omega_n^\# > t^s.$$

Combining this observation with Lemma 7, we will prove that

$$\omega_n^\# \geq \begin{cases} 1, & \text{if } 1 \leq n \leq c_2 d, \\ \left(\dfrac{\log(n)}{\log(c_2 d)}\right)^{s/2}, & \text{if } c_2 d < n \leq (c_1/2)^d, \\ \left(\dfrac{n^{1/d}\sqrt{d}}{c_1}\right)^s, & \text{if } (c_1/2)^d < n \end{cases} \tag{33}$$

with $c_1, c_2 > 0$ from Lemma 7, with possibly larger c_2 to ensure $c_2 \geq (c_1/2)^4$.

Indeed, if $1 \leq n \leq c_2 d$, then (33) follows easily. If $n > (c_1/2)^d$, then we choose any t with $\sqrt{d}/2 \leq t < n^{1/d}\sqrt{d}/c_1$. Then, by Lemma 7, $N(t, d) \leq (c_1 t/\sqrt{d})^d < n$ and $\omega_n^\# > t^s$. As t was arbitrary within these limits, we get $\omega_n^\# \geq (n^{1/d}\sqrt{d}/c_1)^s$.

Finally, if $c_2 d < n \leq (c_1/2)^d$, then we put $t^2 = \log(n)/\log(c_2 d) > 1$ and obtain

$$t^2 \leq \frac{d \log(c_1/2)}{\log(c_2)} \leq \frac{d}{4},$$

where we used that $c_2 \geq (c_1/2)^4$. Therefore, Lemma 7, gives

$$\log(N(t,d)) \leq t^2 \log(c_2 d/t^2) < t^2 \log(c_2 d) = \log(n)$$

and $\omega_n^\# > t^s = (\log(n)/\log(c_2 d))^{s/2}$, which finishes the proof of (33).

Step 3. Next, we show that

$$W(\ell,s,d) \geq \begin{cases} \ell, & \text{if } 1 \leq \ell \leq c_2 d, \\ (1-s/2)\ell \left(\frac{\log(\ell)}{\log(c_2 d)}\right)^{s/2}, & \text{if } c_2 d < \ell \leq (c_1/2)^d, \\ c\, d^{s/2} \cdot \ell^{s/d+1}, & \text{if } (c_1/2)^d \leq \ell, \end{cases} \tag{34}$$

where $c > 0$ depends only on s.

If $1 \leq \ell \leq c_2 d$, this follows immediately from (33). If $c_2 d \leq \ell \leq (c_1/2)^d$, then we estimate

$$\begin{aligned} W(\ell,s,d) &= \sum_{k=1}^{c_2 d} \omega_k^\# + \sum_{k=c_2 d+1}^{\ell} \omega_k^\# \geq c_2 d + \frac{1}{\log(c_2 d)^{s/2}} \sum_{k=c_2 d+1}^{\ell} (\log k)^{s/2} \\ &\geq c_2 d + \frac{1}{\log(c_2 d)^{s/2}} \int_{c_2 d}^{\ell} (\log t)^{s/2} dt \\ &= c_2 d + \frac{\ell \log(\ell)^{s/2} - c_2 d \log(c_2 d)^{s/2}}{\log(c_2 d)^{s/2}} - \frac{s}{2} \frac{1}{\log(c_2 d)^{s/2}} \int_{c_2 d}^{\ell} (\log t)^{s/2-1} dt \\ &\geq \ell \cdot \frac{\log(\ell)^{s/2}}{\log(c_2 d)^{s/2}} - \ell \cdot \frac{s}{2} \frac{\log(\ell)^{s/2}}{\log(c_2 d)^{s/2}}, \end{aligned}$$

which gives the second estimate in (34).

It remains to prove (34) for $\ell \geq (c_1/2)^d$. To simplify the presentation, we assume that $c_1 \geq 4$ is an even number. We start with $\ell = (c_1/2)^d$. We set $t^2 = \gamma d$ for $\gamma > 0$ small enough to ensure that

$$N(t,d) \leq (c_2 d/t^2)^{t^2} = (c_2/\gamma)^{\gamma d} < (c_1/2)^d/2 = \ell/2.$$

This ensures that $\omega_{\ell/2}^\# > t^s = (\gamma d)^{s/2}$ and also

$$W(\ell,s,d) \geq \sum_{k=\ell/2}^{\ell} \omega_k^\# \geq \frac{\ell}{2} \omega_{\ell/2}^\# \geq \frac{\ell}{2} (\gamma d)^{s/2} = c'\, d^{s/2} \ell^{s/d+1},$$

where c' depends only on s (and the absolute constants c_1 and c_2).

If $(c_1/2)^d \leq \ell \leq 4(c_1/2)^d$, the proof of (34) follows by monotonicity

$$W(\ell,s,d) \geq W((c_1/2)^d,s,d) \geq c' d^{s/2} [4(c_1/2)^d]^{s/d+1} \cdot 4^{-1-s/2} \geq c\, d^{s/2} \ell^{s/d+1},$$

where $c = c'\, 4^{-1-s/2}$ depends only on s.

Finally, for $\ell \geq 4(c_1/2)^d$, we use (33) and estimate

$$W(\ell,s,d) \geq \sum_{k=(c_1/2)^d+1}^{\ell} \omega_k^\# \geq c_1^{-s} d^{s/2} \sum_{k=(c_1/2)^d+1}^{\ell} k^{s/d},$$

which finishes the proof of (34).

Step 4. As the last step, we prove (31).

If $n \geq (c_1/2)^d$, we use (32) and (34), and obtain

$$\sigma_n(\alpha)^2 \leq \sum_{k=n+1}^{\infty} W(k,s,d)^{-2} \leq \sum_{k=n+1}^{\infty} c^{-2} d^{-s} k^{-2(s/d+1)} \leq c^{-2}\, d^{-s} n^{-2s/d-1}.$$

If $c_2 d \leq n \leq (c_1/2)^d$, then we estimate similarly

$$\begin{aligned}\sigma_n(\alpha)^2 &= \sum_{k=n+1}^{(c_1/2)^d} W(k,s,d)^{-2} + \sum_{k=(c_1/2)^d+1}^{\infty} W(k,s,d)^{-2} \\ &\leq c\, \log(c_2 d)^s \sum_{k=n+1}^{(c_1/2)^d} k^{-2} \log(k)^{-s} + c\, d^{-s} (c_1/2)^{-2s-d} \qquad (35) \\ &\leq c\left(\frac{\log(c_2 d)}{\log(n)}\right)^s \cdot n^{-1}.\end{aligned}$$

And finally, if $1 \leq n \leq c_2 d$ we combine (32) and (35) to

$$\sigma_n(\alpha)^2 \leq \sum_{k=n+1}^{c_2 d} W(k,s,d)^{-2} + c\,(c_2 d)^{-1} \leq \sum_{k=n+1}^{c_2 d} k^{-2} + c\,(c_2 d)^{-1}$$

and the result follows. ■

We now show the analogue of our neural network approximation result in Theorem 8 for Barron classes.

Theorem 12. *Let $0 < s < 1$. Then there are constants $c, C > 0$, depending only on s, such that for every $d \in \mathbb{N}$ and $0 < \varepsilon < 1$ the following statement is true.*
Let $R := (C/\varepsilon)^{1/s}$ and n such that $\sigma_n(b_s^d, \ell_2) < c\varepsilon$. Then, for every $f \in \mathbb{B}^s([0,1]^d)$ there is a neural network $\mathcal{N} \in \Upsilon_d^{W,L}$ with

$$W = 4n \quad \text{and} \quad L \leq 4 + \log_2\left(R \cdot \sqrt{\min(R,d)}\right)$$

such that

$$\|f - \mathcal{N}\|_2 \;\leq\; \varepsilon \cdot \|f\|_{\mathbb{B}^s}.$$

Moreover, $\mathcal{N}$ can be given explicitly depending on the coefficients of the best n-term approximation of f.

Proof Let f be the unit ball of $\mathbb{B}^s([0,1]^d)$. By Theorem 3, there is a constant $C_s > 0$, such that $\|f\|_{\mathcal{B}^s([0,1]^d)} \leq C_s$. We can therefore decompose f as in (14) into a series

$$f = \alpha_0 + \sum_{k +\!\!\!\!> 0} \Big[\alpha_k \mathcal{C}_k + \beta_k \mathcal{S}_k\Big]$$

with

$$\|\alpha\|_{b_s^d} = \sum_{k \mathrel{+\mkern-12mu>} 0} \|k\|_2^s \cdot |\alpha_k| \le C_s \quad \text{and} \quad \|\beta\|_{b_s^d} = \sum_{k \mathrel{+\mkern-12mu>} 0} \|k\|_2^s \cdot |\beta_k| \le C_s.$$

We define $R := (C/\varepsilon)^{1/s}$ as well as $\alpha^R = (\alpha_k^R)_{k \mathrel{+\mkern-12mu>} 0}$ by $\alpha_k^R = \alpha_k$ if $\|k\|_2 \le R$ and $\alpha_k^R = 0$ otherwise. Then

$$\begin{aligned} \|\alpha - \alpha^R\|_2 = \Bigg(\sum_{k \mathrel{+\mkern-12mu>} 0 : \|k\|_2 > R} |\alpha_k|^2 \Bigg)^{1/2} &\le \sum_{k \mathrel{+\mkern-12mu>} 0 : \|k\|_2 > R} |\alpha_k| \\ &\le R^{-s} \sum_{k \mathrel{+\mkern-12mu>} 0 : \|k\|_2 > R} \|k\|_2^s \cdot |\alpha_k| \le C_s R^{-s}. \end{aligned} \tag{36}$$

Next, we define $\widehat{\alpha}^{R,n}$ to be the best n-term approximation of α^R in ℓ_2 (with respect to the canonical basis of ℓ_2). Hence,

$$\|\alpha^R - \widehat{\alpha}^{R,n}\|_2 \le \|\alpha^R\|_{b_s^d} \cdot \sigma_n(b_s^d, \ell_2) \le C_s \cdot \sigma_n(b_s^d, \ell_2).$$

Furthermore, we define β^R and $\widehat{\beta}^{R,n}$ similarly.

We approximate f by $f^{R,n}$, which is defined as follows

$$f^{R,n} = \alpha_0 + \sum_{k \mathrel{+\mkern-12mu>} 0} \Big[\widehat{\alpha}_k^{R,n} \mathcal{C}_k + \widehat{\beta}_k^{R,n} \mathcal{S}_k \Big].$$

Then, using the triangle inequality and the result of Schneider and Vybíral (2024) that (3) is a Riesz basis of $L_2([0,1]^d)$ with the upper Riesz constant $C = 1/2$ (cf. also Theorem 1 with $s = 0$), we obtain

$$\begin{aligned} \|f - f^{R,n}\|_2 &= \Bigg\| \sum_{k \mathrel{+\mkern-12mu>} 0} \Big[(\alpha_k - \widehat{\alpha}_k^{R,n}) \mathcal{C}_k + (\beta_k - \widehat{\beta}_k^{R,n}) \mathcal{S}_k \Big] \Bigg\|_2 \\ &\le \frac{\sqrt{2}}{2} \cdot \Big(\|\alpha - \widehat{\alpha}^{R,n}\|_2 + \|\beta - \widehat{\beta}^{R,n}\|_2 \Big) \\ &\le \frac{\sqrt{2}}{2} \cdot \Big(\|\alpha - \alpha^R\|_2 + \|\alpha^R - \widehat{\alpha}^{R,n}\|_2 + \|\beta - \beta^R\|_2 + \|\beta^R - \widehat{\beta}^{R,n}\|_2 \Big) \\ &\le C_s' \cdot [R^{-s} + \sigma_n(b_s^d, \ell_2)] \le \varepsilon, \end{aligned}$$

where we use the choice of n.

Finally, by Lemma 6, $f^{R,n}$ can be reproduced by an artificial neural network $f^{R,n} \in \Upsilon_d^{W,L}$ with

$$W = 4n \quad \text{and} \quad L \le 4 + \log_2 \Big(R \cdot \sqrt{\min(R,d)} \Big)$$

where we again use that $\|k\|_1 \le \sqrt{\min(R,d)} \|k\|_2$ for $k \in \mathbb{Z}^d$. ■

4. Discussion

In the last section, we studied the approximation of functions from Sobolev and Barron classes using different neural networks. Let us finally highlight and discuss a few interesting cases and compare them with the relevant literature.

4.1 Sobolev Spaces

The potential of shallow and deep neural networks in reproducing functions from classical function spaces has been studied intensively for several decades in different regimes (cf. Mhaskar et al., 2016; Mhaskar and Poggio, 2016; Pinkus, 1999; Telgarsky, 2015, 2016). The influential paper by Yarotsky (2017) studied a setting very similar to ours, although the function spaces (and the error of approximation) involved the uniform norm instead of the L_2-norm used in our work. To be more specific, for positive integers n, d, let $F_{n,d}$ denote a set of functions defined on $[0,1]^d$, which have all partial derivatives up to the order n uniformly bounded by one. Then Yarotsky (2017) showed that

- there is a fixed architecture of a neural network, which (by a suitable choice of weights and biases) can approximate every function $f \in F_{n,d}$ uniformly up to error $0 < \varepsilon < 1$ and which has depth $\mathcal{O}(\log_2(1/\varepsilon) + 1)$ and $\mathcal{O}(\varepsilon^{-d/n}(\log(1/\varepsilon) + 1))$ weights;
- choosing the network architecture adaptively depending on f can lead to a lower number of weights;
- if a network architecture can approximate every $f \in F_{n,d}$ uniformly up to error ε, then it must have at least $c\,\varepsilon^{-d/(2n)}$ weights.

We stress that essentially all implicit constants in Yarotsky (2017) depend both on n and d. From this point of view, we prove a variant of Yarotsky (2017, Theorem 1) for Sobolev classes built upon L_2 and approximation in L_2 and, in contrast to Yarotsky (2017), we achieve explicit d-dependence. Note that we have very little requirements on the functions' regularity, but are also limited to small smoothness $0 < s < 1$.

Indeed, if we fix $d \geq 1$ to be a constant and let $\varepsilon \to 0$, then Theorem 8 (see Corollary 9) provides a fixed architecture, which allows to approximate every f from the unit ball of $W^s([0,1]^d)$ up to error ε (in the $L_2([0,1]^d)$ norm) with

$$W \sim_{s,d} \varepsilon^{-d/s} \quad \text{and} \quad L \sim_{s,d} \log(1/\varepsilon).$$

The corresponding neural network has $\mathcal{O}(\varepsilon^{-d/s} \max(\log(1/\varepsilon), d))$ nonzero parameters (counting weights and biases), see the discussion after Lemma 6. This rate is optimal due to the continuous dependence of the parameters of the neural network on f, see DeVore et al. (1989) and Yarotsky (2017, Theorem 1).

Another line of research that comes close to our work is represented, for example, by DeVore et al. (2021); Gühring et al. (2020); Shen et al. (2022); Siegel (2023). To compare our results with Siegel (2023), we note that Theorem 8 shows that, for $f \in W^s([0,1]^d)$, $0 < s < 1$, it holds

$$\inf_{f_R \in \Upsilon_d^{W,L}} \|f - f_R\|_{L_2([0,1]^d)} \leq C \|f\|_{W^s([0,1]^d)} R^{-s},$$

where the appearing constant C is independent of d and width and depth are chosen as in Theorem 8, i.e., $W = 4N(R,d)$, $L = 4 + \log_2(R \cdot \sqrt{\min(R,d)})$. If we prove Theorem 8 using the second construction from Lemma 6, we can obtain the same result also with

$$W = d + 3 \quad \text{and} \quad L \leq 2N(R,d) \cdot \log_2\left(16 \cdot R\sqrt{\min(R,d)}\right). \tag{37}$$

In contrast to this Siegel (2023, Theorem 1) showed that for $0 < s < \infty$, $1 \leq p, q \leq \infty$, and $W = 25d + 31$ it holds

$$\inf_{f_L \in \Upsilon_d^{W,L}} \|f - f_L\|_{L_p([0,1]^d)} \leq C\|f\|_{W_q^s([0,1]^d)} L^{-2s/d}, \tag{38}$$

for a constant $C := C(s, r, q, p, d)$. Additionally, Siegel (2023, Theorem 3) also gives a lower bound which shows that the rate above is sharp in terms of the number of parameters. For $p = q = \infty$ and $0 < s < 1$ this corresponds to the setting of Yarotsky (2017) and for all $s > 0$ we refer to Lu et al. (2021). Note that Siegel's results (38) are more general in terms of the parameters considered in the underlying functions spaces whereas we only cover $p = q = 2$ so far.

If we now consider d fixed and let the error of approximation $\varepsilon > 0$ go to zero, then the width W considered in (38) is fixed, but the length L grows to infinity as $L = \mathcal{O}(\varepsilon^{-d/(2s)})$. Note that classical methods of approximation using piecewise polynomials or wavelets can attain an approximation rate of $L^{-s/d}$ with L wavelet coefficients or piecewise polynomials with L pieces. Therefore, the approximation rate of $CL^{-2s/d}$ is significantly faster than traditional methods of approximation. This phenomenon has been called the *super-convergence* of deep ReLU networks (Daubechies et al., 2022; DeVore et al., 2021; Shen et al., 2022; Yarotsky, 2018) and is obtained using a special bit extraction technique (Bartlett et al., 1998), which gives an optimal encoding of sparse vectors.

In the setting of d fixed and ε tending to zero, the width in (37) is also fixed but length grows faster as $L = \mathcal{O}(\varepsilon^{-d/s} \cdot \log(1/\varepsilon))$. This is due the fact that we only consider fixed architectures and (up to now) do not make use of any variant of the bit-extraction technique. We leave it as an open problem, if such an improvement would be possible also in our approach.

Finally, let us mention in this context that approximation rates when both the width and depth vary have also been obtained by Shen et al. (2022). There the authors considered Hölder continuous functions (which again corresponds to $p = q = \infty$ and $s > 0$) and proved that for any $N, L \in \mathbb{N}$ it holds that ReLU networks with width $\mathcal{O}(\max\{d\lfloor N^{1/d}\rfloor, N + 2\})$ and depth $\mathcal{O}(L)$ can approximate a Hölder function on $[0,1]^d$ with an approximation rate $\varepsilon = \mathcal{O}(\lambda\sqrt{d}(N^2L^2 \ln N)^{-\alpha/d})$, where $\alpha \in (0,1]$ and $\lambda > 0$ are Hölder order and Hölder constant, respectively. To compare this result with our work, one has to recalculate the dependence of N and/or L on ε, which again reveals the exponential dependence of the involved constants on the dimension d. We refer also to Dung and Nguyen (2021) for approximation in other norms.

Finally, our approach allows us to consider also a different regime, namely when $\varepsilon > 0$ is fixed and d grows to infinity. Then, for d large enough (to be more specific, for $d \geq (c_3/\varepsilon)^{2/s}$) we get $\varepsilon \geq c_3 d^{-s/2}$ and the first regime of Corollary 9 applies. Note that the threshold for d to hit this setting occurs earlier if s is growing. For such d, Corollary 9 shows that the number of layers $L = \mathcal{O}(\log_2(1/\varepsilon))$ stays bounded and the width $W = \mathcal{O}(\varepsilon^{2/s}d)^{\gamma\varepsilon^{-2/s}} = \mathcal{O}(d^{\gamma\varepsilon^{-2/s}})$ grows polynomially in d and so does the number of all weights in the network which is of order $\mathcal{O}(W^2L + dW)$, see (21). In this sense, we avoid the *curse of dimensionality.*

4.2 Barron Spaces

The original paper of Barron (1993) used the Maurey technique (Pisier, 1981), also known as the probabilistic Caratheodory's theorem (Vershynin, 2018, Theorem 0.0.2), to show that the functions from the Barron class $\mathcal{B}^1_{\mathrm{ext}}$, cf. (19), can be approximated by shallow neural networks with n neurons up to the precision $\mathcal{O}(n^{-1/2})$ in the L_2 sense. It also allowed to give explicit bounds on the constants involved and to show that neural network approximation of functions from this class avoids the curse of dimensionality. The result of Barron (1993) holds in a surprising generality, allowing for a general class of sigmoidal activation functions (in contrast to ReLU as used here) and for a general Borel probability measure space (compared to the unit cube with Lebesgue measure considered in our work). Note also that Barron (1993) considers spaces of smoothness $s = 1$, whereas our technique applies to $0 < s < 1$. The approach of Barron (1993) leads to a randomized construction of a shallow neural network and its architecture depends on the approximated function f. We remark that Jones (1992) gives a non-probabilistic proof of Maurey's result.

The results of Barron (1993) were generalized in several directions. Uniform approximation was considered by Barron (1992) and spaces of Barron-type based on the integral representation (20) were investigated by Caragea et al. (2023); E et al. (2022); E and Wojtowytsch (2022). Upper and lower bounds on approximation rates, metric entropy, and n-widths of Barron classes were recently obtained by Siegel and Xu (2021a).

To compare the bound of Theorem 12 with these results, we have to take into account also the best n-term approximation bounds of Lemma 11. Naturally, we distinguish several cases.

First, we observe that if $C \cdot d^{-1/2} \le \varepsilon \le 1$, then $n = \mathcal{O}(\varepsilon^{-2})$ and Theorem 12 provides a neural network with $W = \mathcal{O}(n) = \mathcal{O}(\varepsilon^{-2})$ and $L = \mathcal{O}(\log_2(1/\varepsilon))$, which approximates given function f from the unit ball of $\mathbb{B}^s([0,1]^d)$ up to ε precision. The architecture of this neural network depends adaptively on the function f, which we want to approximate. Furthermore, this neural network has $\mathcal{O}(\varepsilon^{-2} \cdot \max(d, \log_2(1/\varepsilon)))$ non-zero weights. All constants in the $\mathcal{O}$-notation are independent of d and we indeed recover the results of Barron (1993), up to the log-terms. But note that we use the advantage of growing depth L, whereas Barron (1993) works exclusively with shallow neural networks. It follows that if we fix $1 > \varepsilon > 0$ constant and let d grow to infinity, then $C \cdot d^{-1/2} \le \varepsilon$ for d large enough and we indeed avoid the curse of dimensionality.

Similarly, if $\varepsilon = \mathcal{O}(d^{-s/2} \cdot (c_1/2)^{-s-d/2})$, then the condition $\sigma_n(b^d_s, \ell_2) < c\,\varepsilon$ leads to $n = \mathcal{O}(\varepsilon^{-\frac{2d}{2s+d}} \cdot d^{-\frac{sd}{2s+d}})$ (or, equivalently, $\varepsilon = \mathcal{O}(d^{-s/2} \cdot n^{-\frac{s}{d}-\frac{1}{2}})$). An improvement in the asymptotic error decay rate from $1/2$ to $1/2 + 1/d$ was already observed by Klusowski and Barron (2018), even for the uniform approximation, but they required $s \in \{2, 3\}$.

We believe that our approach leads to a more transparent proof, at least for $s < 1$. It is clearly of interest to extend this technique to higher smoothness and to classes of functions with additional structure.

5. Approximation by Neural Networks Using Function Values

We want to study how well one can approximate a function from a class F by neural networks if only function values $f(x_i)$ (aka *samples*) for some x_i are known, i.e., we consider the *NN-sampling numbers* of F in G, which are defined by

$$g_n^{\mathrm{NN}}(F,G,W,L) := \inf_{\substack{x_1,\ldots,x_n \in D \\ \phi\colon \mathbb{R}^n \to \Upsilon_d^{W,L}}} \sup_{f\in F} \Big\| f - \phi(f(x_1),\ldots,f(x_n)) \Big\|_G,$$

where $\Upsilon_d^{W,L}$ is the set of feed-forward neural networks defined on a set D (here: $D=[0,1]^d$) with ReLU activation, width W and depth L, and G is a normed space (here: $G=L_p$) specifying the error measure. This gives the minimal error achievable with neural networks that can be found by any *algorithm* that has only access to n function evaluations of f.

Note that we consider the *worst-case error* over a class F, i.e., we want an algorithm to be "good" for all elements of F (which is often the unit ball in a normed space) simultaneously. This accounts for the fact that a specific f is only known through the data, and some assumptions, like a certain regularity. A typical *benchmark* in this setting are the *Gelfand numbers*

$$c_n(F,G) := \inf_{\substack{\psi\colon \mathbb{C}^n \to G \\ N\colon F\to\mathbb{C}^n \text{ linear}}} \sup_{f\in F} \big\| f - \psi\circ N(f) \big\|_G,$$

which represent the minimal error of an arbitrary algorithm (without a specified approximation space) that uses n linear measurements.

There has been a lot of interest, also recently, in finding neural network approximations based on samples, see e.g. Barron (1994); Daubechies et al. (2022); DeVore et al. (2021). The used methods are usually tailored to the specific setting and, again, often employ heavy computations and unknown dimension-dependent factors. The aim of this section is to show that the Riesz basis established above, together with general results on recovery based on (random) samples, gives an easy way to obtain rather explicit bounds. For this, let us denote by

$$V_R := \mathrm{span}\Big\{\mathcal{C}_k,\, \mathcal{S}_k \colon k\in\mathbb{Z}^d,\, k \mathrel{+\!\!\!\rhd} 0,\, \|k\|_2 \le R\Big\} \tag{39}$$

the finite-dimensional space where we search for the approximation. Recall from Lemma 6 (see also the proof of Theorem 8) that every $g\in V_R$ can be written explicitly as a neural network from $\Upsilon_d^{W,L}$ with suitable W and L based on the corresponding coefficients. To bound g_n^{NN}, it is therefore enough to learn a corresponding approximation in V_R.

Let us first discuss the case of Sobolev spaces $W^s([0,1]^d)$. In this case, as for more general reproducing kernel Hilbert spaces, it has been observed in recent years that a simple least squares approximation

$$\widehat{f}_{V,X}^{ls} := \operatorname*{argmin}_{g\in V} \sum_{i=1}^N |g(x_i)-f(x_i)|^2 \tag{40}$$

onto a suitable subspace $V\subset L_2$, with $X=\{x_1,\ldots,x_N\}$ being the sampling points, may lead with high probability to a near-optimal algorithm for approximation (in the worst-case setting) if $x_1,\ldots,x_N$ are randomly and independently chosen from the uniform distribution in $[0,1]^d$, and $N\sim\dim(V)\cdot\log(\dim(V))$, see Krieg and Ullrich (2021); Ullrich (2020).

This follows from the following more general result. We refer to the survey Sonnleitner and Ullrich (2023) where this, see Sections 3.4 and 4.1, as well as the randomized setting and generalizations to other classes F and G are explained.

Proposition 13. *Let μ be a Borel probability measure on a compact topological space D, $F \subset C(D)$ be compact, $\{b_1, b_2, \ldots\}$ be an orthonormal basis of $L_2(\mu)$ with $\sup_k \|b_k\|_\infty < \infty$, and $U_n := \operatorname{span}\{b_1, \ldots, b_n\}$. Moreover, assume that*

$$\sup_{f\in F} \inf_{g\in U_n} \|f - g\|_2 \;\le\; K \cdot n^{-t} \tag{41}$$

for some $t > 1/2$, $K < \infty$ and all $n \in \mathbb{N}$.

Then, there is a constant $C \in \mathbb{N}$ such that for all $n \in \mathbb{N}$, the least-squares method $\widehat{f}_N^{ls} := \widehat{f}_{U_n,X}^{ls}$ from (40) *with the $N \ge C\, n \log(n)$ points $X = \{x_1, \ldots, x_N\}$ being chosen iid w.r.t. μ satisfies with probability $1 - \frac{C}{N^2}$ and for every $2 \le p \le \infty$ the bound*

$$\sup_{f\in F} \|f - \widehat{f}_N^{ls}\|_p \;\le\; C\, K \left(\frac{N}{\log N}\right)^{-t+1/2-1/p}.$$

Proof Similarly to Krieg et al. (2025, Corollary 5), see also Sonnleitner and Ullrich (2023, Theorem 3.7), we apply the algorithm of Krieg and Ullrich (2021) together with Krieg et al. (2025, Lemma 9), and observe that the condition on $\{b_k\}$ allows to remove the weights in the algorithm, see Krieg and Ullrich (2021, Remark 1), and that $\|g\|_p \le c n^{1/2-1/p}\|g\|_2$ for all $g \in U_n$ and $2 \le p \le \infty$, see Krieg et al. (2025, Remark 8). ■

Remark 8. For (40) to be uniquely solvable, it is required to have $N \ge \dim(V)$. The additional logarithmic factor in the results above is needed due to the assumption that the sampling points are chosen independently and identically distributed, see Krieg et al. (2022). This *oversampling* can be removed with the help of (usually non-constructive) subsampling, see e.g. Dolbeault et al. (2023); Krieg et al. (2025) for theoretical results and Bartel et al. (2023) for a detailed treatment of the implementation. Note that these approaches may require additional weights in the algorithm (40).

Our approach to bound the NN-sampling numbers is to apply Proposition 13 to the Sobolev spaces $W^s([0,1]^d)$ and V_R from (39). Unfortunately, it is well known that in this case we can only have $t \le s/d$ in (41), even if we would choose the optimal subspaces there. Since the analysis above requires $s < 1$, and we need $t > 1/2$, we do not get a result (yet) from this general approach for $d > 1$. For $d = 1$, however, we obtain the following.

Corollary 14. *For $1/2 < s < 1$, there is a constant $C_s \in \mathbb{N}$ such that for all $R \in \mathbb{N}$, the least-squares method $\widehat{f}_N^{ls} := \widehat{f}_{V_R,X}^{ls}$ from* (40) *with V_R from* (39) *and the $N \ge C\, R \log(R)$ points $X = \{x_1, \ldots, x_N\}$ being chosen iid w.r.t. the uniform distribution on $[0,1]$ satisfies with probability $1 - \frac{C}{N^2}$ and for every $2 \le p \le \infty$ the bound*

$$\|f - \widehat{f}_N^{ls}\|_p \;\le\; C \left(\frac{N}{\log N}\right)^{-s+1/2-1/p} \|f\|_{W^s}$$

for all $f \in W^s([0,1])$ simultaneously.

We leave out the details of the proof, which relies on Proposition 13 but requires a few technicalities when using a Riesz basis instead of an orthonormal basis. Note that we could also take $\|f^R\|_{W^s}$ instead of $\|f\|_{W^s}$ above, see e.g. Sonnleitner and Ullrich (2023, Theorem 3.2).

With this, Lemma 6, and taking Remark 8 into account, we obtain

$$g_n^{\mathrm{NN}}(W^s, L_p, bn, \log_2(bn)) \asymp n^{-s+1/2-1/p} \asymp c_n(W^s, L_p)$$

for $W^s := W^s([0,1])$ with $1/2 < s < 1$, $2 \le p \le \infty$ and some absolute constant $b \in \mathbb{N}$. See e.g. Pinkus (1985, Theorem VII.1.1) for the classical second equivalence, and Krieg et al. (2025, Section 1.1) for a similar discussion. That is, NN-sampling numbers behave as the Gelfand numbers, whenever we allow $W \asymp n$ and $L \asymp \log n$. (Alternatively, $W = 4$ and $L \asymp n \log n$ would also work.) It would be of great interest to see whether and how this can be extended to larger s and d. However, note that unit balls of Sobolev spaces are, depending on the specific norm of the space, too large to get useful worst-case error bounds in high dimensions: Any algorithm needs at least $(d/\varepsilon)^{d/k}$ function evaluations to achieve an error $\varepsilon > 0$, if the unit ball contains all functions with directional derivatives of order k bounded by one, see Hinrichs et al. (2014, 2017).

The Barron spaces $\mathbb{B}^s([0,1]^d)$ are more interesting in the present context, because they are much better suited for high dimensions. In particular, $\mathbb{B}^s([0,1]^d)$ is continuously embedded into $C([0,1]^d)$ for all $d \in \mathbb{N}$ and $s \ge 0$. However, for these classes, it is known that linear methods are not optimal. The method of choice is *basis pursuit denoising*, i.e.,

$$\widehat{f}^{bp}_{R,X} := \operatorname*{argmin}_{g \in V_R} \|g\|_{\mathcal{B}^0} \qquad \text{subject to} \quad \sqrt{\frac{1}{N}\sum_{i=1}^{N} |g(x_i) - f(x_i)|^2} \le C\, R^{-s}, \tag{42}$$

with C chosen as in (17), and $X = \{x_1, \ldots, x_N\}$ are the sampling points. Recall that the $\mathcal{B}^0$-norm is just the absolute sum of the coefficients.

This is the most important method of *sparse approximation* or *compressed sensing*. We refer to Candès and Tao (2006); Donoho (2006) for its origins and to Foucart and Rauhut (2013) for a detailed treatment of the subject. Also note that the idea already appeared implicitly in Garnaev and Gluskin (1984) in the context of ℓ_2-approximation of ℓ_1-vectors. The recovery from function values has been treated first by Rauhut (2007), see also Rauhut and Ward (2016), and here we employ a variant of the method of Brugiapaglia et al. (2021) suitable for Riesz bases. This approach has been used for several problems in optimal recovery. Let us only highlight the recent contributions of Jahn et al. (2023); Krieg (2024); Voigtlaender (2022), where this has been discussed in the context of sampling numbers, high dimensions and (different) Barron-type spaces, respectively. In our setting, we obtain the following from Brugiapaglia et al. (2021, Theorem 2.6).

Proposition 15. *For $0 < s < 1$, there is a constant $C_s \in \mathbb{N}$ such that for all $k \in \mathbb{N}$, the basis pursuit denoising $\widehat{f}^{bp}_{R,X}$ from (42) with the points $x_1, \ldots, x_N$ being chosen iid w.r.t. the uniform distribution on $[0,1]^d$ and*

$$N \ge C_s\, k\, \log^2(k)\, \log\bigl(N(R,d)\bigr),$$

with $N(R,d)$ from (26), satisfies with probability $1-\frac{C_s}{k^2}$ and for every $2 \le p \le \infty$ the bound

$$\|f - \widehat{f}^{bp}_{X,R}\|_p \;\le\; C_s \left(k^{-1/p}\, \sigma_k(f, \mathcal{B}^0, V_R) \;+\; k^{1/2-1/p}\, R^{-s}\, \|f\|_{\mathbb{B}^s}\right) \tag{43}$$

for all $f \in \mathbb{B}^s([0,1]^d)$ simultaneously.

Proof This is a rather direct application of Brugiapaglia et al. (2021, Theorem 2.6), see also Krieg (2024, Lemma 9). For this, note that Brugiapaglia et al. (2021) consider approximation of the coefficient vector (which they call f) in the ℓ_q-norm for $q \in \{1,2\}$. So, we additionally need the basic facts that this can be extended to $1 \le q \le 2$ by interpolation, and that the L_p-norm is bounded in terms of the ℓ_q-norm of the coefficients for bounded Riesz systems, like the one we use, for $2 \le p \le \infty$ and $q = \frac{p}{p-1}$. See also Brugiapaglia et al. (2021, Theorem 2.3 & Remark 2.4) for this setting. We then apply the method of Brugiapaglia et al. (2021) to recover the coefficients of f_R, where, for f as in (14), we write $f_R := \sum_{\ell \,{+\!\!\!\!>}\, 0 : \|\ell\|_2 \le R} \Big[\alpha_\ell \mathcal{C}_\ell + \beta_\ell \mathcal{S}_\ell\Big]$. The values $e_i := (f - f_R)(x_i)$ are considered noise. So, for the precise form of (42), it remains to observe that

$$\|f - f_R\|_\infty \;\le\; C\, R^{-s}\, \|f\|_{\mathbb{B}^s},$$

which follows the lines of (36) using $\|f\|_\infty \le \|f\|_{\mathcal{B}^0}$ for all $g \in \mathcal{B}^0 = \mathbb{B}^0$, since the Riesz system is bounded by one. ■

A careful analysis of $\sigma_k(f, \mathcal{B}^0, V_R)$ shall lead to bounds in (43) that also reflect the smoothness s, possibly optimal and/or with a mild dependence on d. However, since we only consider small s anyway, we only use the obvious $\sigma_k(f, \mathcal{B}^0, V_R) \le \|f\|_{\mathcal{B}^0} \le C_s' \|f\|_{\mathbb{B}^s}$, where the latter inequality follows from Theorem 3, to obtain the following corollary. This shows that a linear-in-d number of samples is enough in dimension d.

Corollary 16. *For $0 < s < 1$ and $2 \le p \le \infty$, there is $C \in \mathbb{N}$, independent of d, such that the following holds. For $\varepsilon > 0$ and*

$$N \ge C\, d\, \varepsilon^{-p} \log^3(1/\varepsilon)$$

the basis pursuit denoising $\widehat{f}^{bp}_{R,X}$ from (42) with $R = (C/\varepsilon)^{p/(2s)}$ and the points $x_1, \dots, x_N$ being chosen iid w.r.t. the uniform distribution on $[0,1]^d$ satisfies with probability $1 - C\varepsilon^2$ the bound

$$\|f - \widehat{f}^{bp}_{X,R}\|_p \;\le\; \varepsilon \cdot \|f\|_{\mathbb{B}^s}$$

for all $f \in \mathbb{B}^s([0,1]^d)$ simultaneously.

Proof We apply Proposition 15 with $R = (C/\varepsilon)^{-p/(2s)}$ for C large enough and $k = \lceil (C_s/\varepsilon)^p \rceil$, and use the (rough) bounds $N(R,d) \le (3R)^d$ for $R \ge 1$ and $\sigma_k(f, \mathcal{B}^0, V_R) \le \|f\|_{\mathcal{B}^0}$. ■

As already used in Theorem 12, the approximation $\widehat{f}^{bp}_{X,R}$ can be represented as a neural network from $\Upsilon^{W,L}_d$ whenever $W \ge 4k \asymp (1/\varepsilon)^p$ and $L \ge 4 + \log_2(R \cdot \sqrt{\min(R,d)}) \asymp \log(1/\varepsilon)$.

Rephrasing this with the number of sample points, we obtain for $\mathbb{B}^s = \mathbb{B}^s([0,1]^d)$ with $0 < s < 1$ that

$$g_n^{\mathrm{NN}}(\mathbb{B}^s, L_p, W, L) \ \leq \ C \ \left(\frac{d}{n} \cdot \log^3(n/d)\right)^{1/p}$$

for $W \geq C\, n/(d \log^3(n))$ and $L \geq C \log(n/d)$, where $C > 0$ only depends on p and s. (Again, we may use neural networks with $W = d + 3$ and $L \geq C\, n/(d \log(n))$.)

This should be compared to Barron (1994, Theorem 3) where a slightly smaller linear-in-d bound has been observed for neural networks with $L = 1$ and $W \asymp \sqrt{n/(d \log(n))}$, but only for $p = 2$ and a different Barron-type space (with $s = 1$). Again, the advantage of our approach is that we can rely on techniques from linear approximation in order to treat deep networks without using (very) tailored methods.

Conclusion

In this article, we further analyzed the system of functions introduced by Schneider and Vybíral (2024), originating from Daubechies et al. (2022), which is reminiscent of the classical Fourier basis on $[0,1]^d$ and can be realized by "small" neural networks. We show that this system of piecewise linear functions is even a Riesz basis for suitable classes of "smooth" functions. This, in particular, can be used to reprove in a slightly weaker form the well-known result of Barron (1993) on the (best-)approximation using neural networks in Barron classes, which shows a favorable dependence on the input dimension d. Our method enables us to extend these results to arbitrary small smoothness, which we believe to be essential since smoothness assumptions are problematic in applications.

However, we think that the biggest advance provided is the proof strategy. Compared to previous approaches, it is quite elementary and essentially follows the lines of "classical" approximation theory, like for Fourier series. Roughly speaking, we show that the given system of "small" neural networks can be used almost in the same way as an orthonormal basis, leading to a much more transparent and accessible analysis. It also reveals that, for (worst-case optimal) approximation with neural networks, it might be enough to consider this very small class of special neural networks (with prescribed breakpoints), and linear combinations of them. We note that we work with deep neural networks, which, in contrast to shallow ones, cannot simply be written as sums of ridge functions.

Clearly, it is of interest to extend this to other model classes like, e.g., Sobolev or Barron classes of higher smoothness. For a direct translation of our arguments to this setting, we would need to replace the ReLU activation function by a smoother variant, which makes the analysis more complicated. We leave this for future research.

Moreover, we believe that it is even more compelling to identify further Riesz bases of neural networks also for other classes of functions that may benefit from the general expressivity of deep neural networks, and to consider cases where a "classical" Riesz basis is not easily accessible. Studies of this kind can help to identify the "important structures" in the classes of neural networks and may therefore ultimately affect the choices of architectures and optimization methods used in practice.

Acknowledgments

We would like to thank the anonymous reviewers for their careful reading of the manuscript and for their valuable comments. Mario Ullrich is supported by the Austrian Federal Ministry of Education, Science and Research via the Austrian Research Promotion Agency (FFG) through the project FO999921407 (HDcode) funded by the European Union via NextGenerationEU. The work of Jan Vybíral has been supported by the grant P202/23/04720S of the Grant Agency of the Czech Republic. Jan Vybíral is a member of the Nečas center for mathematical modeling.

References

E. Artin. *The Gamma Function.* Courier Dover Publications, 2015.

A. R. Barron. Neural net approximation. In *Proc. 7th Yale Workshop on Adaptive and Learning Systems*, volume 1, pages 69–72, 1992.

A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993.

A. R. Barron. Approximation and estimation bounds for artificial neural networks. *Machine Learning*, 14(1):115–133, 1994.

F. Bartel, M. Schäfer, and T. Ullrich. Constructive subsampling of finite frames with applications in optimal function recovery. *Applied and Computational Harmonic Analysis*, 65:209–248, 2023.

P. Bartlett, V. Maiorov, and R. Meir. Almost linear VC dimension bounds for piecewise polynomial networks. In *Advances in Neural Information Processing Systems (NIPS)*, volume 11, 1998.

H. Bölcskei, P. Grohs, G. Kutyniok, and P. Petersen. Optimal approximation with sparsely connected deep neural networks. *SIAM Journal on Mathematics of Data Science*, 1:8–45, 2019.

S. Brugiapaglia, S. Dirksen, H. C. Jung, and H. Rauhut. Sparse recovery in bounded Riesz systems with applications to numerical methods for PDEs. *Applied and Computational Harmonic Analysis*, 53:231–269, 2021.

E. J. Candès and T. Tao. Near-optimal signal recovery from random projections: universal encoding strategies? *IEEE Transactions on Information Theory*, 52:5406–5425, 2006.

A. Caragea, P. Petersen, and F. Voigtlaender. Neural network approximation and estimation of classifiers with classification boundary in a Barron class. *Annals of Applied Probability*, 33(4):3039–3079, 2023.

O. Christensen and D. T. Stoeva. p-frames in separable Banach spaces. *Advances in Computational Mathematics*, 18:117–126, 2003.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

I. Daubechies, R. DeVore, S. Foucart, B. Hanin, and G. Petrova. Nonlinear approximation and (deep) ReLU networks. *Constructive Approximation*, 55:127–172, 2022.

R. DeVore. Nonlinear approximation. *Acta Numerica*, 7:51–150, 1998.

R. DeVore, R. Howard, and C. Micchelli. Optimal nonlinear approximation. *Manuscripta Mathematica*, 63(4):469–478, 1989.

R. DeVore, B. Hanin, and G. Petrova. Neural network approximation. *Acta Numerica*, 30: 327–444, 2021.

M. Dolbeault, D. Krieg, and M. Ullrich. A sharp upper bound for sampling numbers in L_2. *Applied and Computational Harmonic Analysis*, 63:113–134, 2023.

D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52:1289–1306, 2006.

D. Dung and V. K. Nguyen. Deep ReLU neural networks in high-dimensional approximation. *Neural Networks*, 142:619–635, 2021.

W. E and S. Wojtowytsch. On the Banach spaces associated with multi-layer ReLU networks: Function representation, approximation theory and gradient descent dynamics. *CSIAM Transactions on Applied Mathematics*, 1(3):387–440, 2020a.

W. E and S. Wojtowytsch. A priori estimates for classification problems using neural networks. `https://arxiv.org/abs/2009.13500`, 2020b. arXiv:2009.13500.

W. E and S. Wojtowytsch. Representation formulas and pointwise properties for Barron functions. *Calculus of Variations and Partial Differential Equations*, 61(2):1–37, 2022.

W. E, C. Ma, and L. Wu. The Barron space and the flow-induced function spaces for neural network models. *Acta Numerica*, 55:369–406, 2022.

Konstantin Eckle and Johannes Schmidt-Hieber. A comparison of deep networks with relu activation function and linear spline-type methods. *Neural Networks*, 110:232–242, 2019. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2018.11.005. URL `https://www.sciencedirect.com/science/article/pii/S0893608018303277`.

D. Elbrächter, D. Perekrestenko, P. Grohs, and H. Bölcskei. Deep neural network approximation theory. *IEEE Transactions on Information Theory*, 67(5):2581–2623, 2021.

S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing.* Birkhäuser, New York, 2013.

A. Yu. Garnaev and E. D. Gluskin. The widths of a Euclidean ball. *Soviet Mathematics - Doklady*, 30:200–204, 1984.

I. Gühring, G. Kutyniok, and P. Petersen. Error bounds for approximations with deep ReLU neural networks in $W^{s,p}$ norms. *Analysis and Applications*, 18(05):803–859, 2020.

G. H. Hardy, J. E. Littlewood, and G. Pólya. *Inequalities.* Cambridge University Press, 1936.

H. Hedenmalm, P. Lindqvist, and K. Seip. A Hilbert space of Dirichlet series and systems of dilated functions in $L_2(0,1)$. *Duke Mathematical Journal*, 86:1–37, 1997.

A. Hinrichs, E. Novak, M. Ullrich, and H. Woźniakowski. The curse of dimensionality for numerical integration of smooth functions. *Mathematics of Computation*, 83:2853–2863, 2014.

A. Hinrichs, E. Novak, M. Ullrich, and H. Woźniakowski. Product rules are optimal for numerical integration in classical smoothness spaces. *Journal of Complexity*, 38:39–49, 2017.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

T. Jahn, T. Ullrich, and F. Voigtlaender. Sampling numbers of smoothness classes via ℓ_1-minimization. *Journal of Complexity*, 79:101786, 2023.

L. K. Jones. A simple lemma on greedy approximation in hilbert space and convergence rates for projection pursuit regression and neural network training. *Annals of Statistics*, 20(1):608–613, 1992.

J. M. Klusowski and A. R. Barron. Approximation by combinations of ReLU and squared ReLU ridge functions with ℓ^1 and ℓ^0 controls. *IEEE Transactions on Information Theory*, 64(12):7649–7656, 2018.

D. Krieg. Tractability of sampling recovery on unweighted function classes. *Proceedings of the American Mathematical Society, Series B*, 11:115–125, 2024.

D. Krieg and M. Ullrich. Function values are enough for L_2-approximation. *Found. Comput. Math.*, 21:1141–1151, 2021.

D. Krieg, E. Novak, and M. Sonnleitner. Recovery of Sobolev functions restricted to iid sampling. *Mathematics of Computation*, 91:2715–2738, 2022.

D. Krieg, K. Pozharska, M. Ullrich, and T. Ullrich. Sampling recovery in L_2 and other norms. *Math. Comp.*, Published Online: October 6, 2025. doi: https://doi.org/10.1090/mcom/4148.

T. Kühn, S. Mayer, and T. Ullrich. Counting via entropy: new preasymptotics for the approximation numbers of Sobolev embeddings. *SIAM Journal on Numerical Analysis*, 54(6):3625–3647, 2016.

M. Leshno, V. Lin, A. Pinkus, and S. Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6): 861–867, 1993.

Y. Li, S. Lu, P. Mathé, and S. Pereverzyev. Two-layer networks with the ReLU^k activation function: Barron spaces and derivative approximation. *Numerische Mathematik*, 156: 319–344, 2024.

P. Lindqvist and K. Seip. Note on some greatest common divisor matrices. *Acta Arithmetica*, 84(2):149–154, 1998.

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

H. N. Mhaskar. Neural networks for optimal approximation of smooth and analytic functions. *Neural Computation*, 8(1):164–177, 1996.

H. N. Mhaskar and T. Poggio. Deep vs. shallow networks: An approximation theory perspective. *Analysis and Applications*, 14(6):829–848, 2016.

H. N. Mhaskar, Q. Liao, and T. Poggio. Learning functions: When is deep better than shallow. `https://arxiv.org/abs/1603.00988`, 2016. arXiv:1603.00988.

P. Petersen and J. Zech. Mathematical theory of deep learning. `https://arxiv.org/abs/2407.18384`, 2024. arXiv:2407.18384.

A. Pinkus. *n-widths in Approximation Theory*, volume 7 of *Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge.* Springer, Berlin, Heidelberg, 1985.

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

G. Pisier. Remarques sur un résultat non publié de B. Maurey (Remarks on an unpublished result of B. Maurey). Technical report, École Polytechnique, Centre de Mathématiques, Palaiseau, 1981.

M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein. On the expressive power of deep neural networks. In *International Conference on Machine Learning*, pages 2847–2854, 2017.

H. Rauhut. Random sampling of sparse trigonometric polynomials. *Applied and Computational Harmonic Analysis*, 22:16–42, 2007.

H. Rauhut and R. Ward. Interpolation via weighted ℓ_1-minimization. *Applied and Computational Harmonic Analysis*, 40:321–351, 2016.

E. Schmidt. Zur Theorie der linearen und nichtlinearen Integralgleichungen I. *Mathematische Annalen*, 63:433–476, 1907.

C. Schneider and J. Vybíral. A multivariate Riesz basis of ReLU neural networks. *Applied and Computational Harmonic Analysis*, 68:101605, 2024.

Z. Shen, H. Yang, and S. Zhang. Optimal approximation rate of relu networks in terms of width and depth. *Journal de Mathématiques Pures et Appliquées*, 157:101–135, 2022.

J. W. Siegel. Optimal approximation rates for deep ReLU neural networks on Sobolev and Besov spaces. *Journal of Machine Learning Research*, 24:1–52, 2023.

J. W. Siegel and J. Xu. Sharp bounds on the approximation rates, metric entropy, and n-widths of shallow neural networks. `https://arxiv.org/abs/2101.12365`, 2021a. arXiv:2101.12365.

J. W. Siegel and J. Xu. High-order approximation rates for shallow neural networks with cosine and $ReLU^k$ activation functions. *Applied and Computational Harmonic Analysis*, 58:1–26, 2021b.

M. Sonnleitner and M. Ullrich. On the power of iid information for linear approximation. *Journal of Applied and Numerical Analysis*, 1:88–126, 2023.

M. Telgarsky. Representation benefits of deep feedforward networks. `https://arxiv.org/abs/1509.08101`, 2015. arXiv:1509.08101.

M. Telgarsky. Benefits of depth in neural networks. In *Conference on Learning Theory*, volume 49, pages 1517–1539, 2016.

E. C. Titchmarsh. *The Theory of the Riemann Zeta-Function.* Clarendon Press, Oxford University Press, New York, 1986.

M. Ullrich. On the worst-case error of least squares algorithms for L_2-approximation with high probability. *Journal of Complexity*, 60:101484, 2020.

R. Vershynin. *High-dimensional Probability: An Introduction with Applications in Data Science*, volume 47 of *Cambridge Series in Statistical and Probabilistic Mathematics.* Cambridge University Press, 2018.

F. Voigtlaender. L_p-sampling numbers for the Fourier-analytic Barron space. `https://arxiv.org/abs/2208.07605`, 2022. arXiv:2208.07605.

A. Wintner. Diophantine approximations and Hilbert's space. *American Journal of Mathematics*, 66:564–578, 1944.

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

D. Yarotsky. Optimal approximation of continuous functions by very deep ReLU networks. In *Conference on Learning Theory*, pages 639–649, 2018.