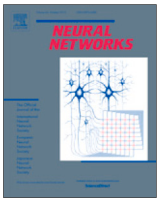
Neural Networks 134 (2021) 107–130

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



Approximation rates for neural networks with encodable weights in smoothness spaces

Ingo Gühring [*,1], Mones Raslan [1]

Institute of Mathematics, Technical University of Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany

ARTICLE INFO

Article history:
Received 17 July 2020
Received in revised form 7 October 2020
Accepted 16 November 2020
Available online 27 November 2020

Keywords:
Neural networks
Expressivity
Approximation rates
Smoothness spaces
Encodable weights

ABSTRACT

We examine the necessary and sufficient complexity of neural networks to approximate functions from different smoothness spaces under the restriction of encodable network weights. Based on an entropy argument, we start by proving lower bounds for the number of nonzero encodable weights for neural network approximation in Besov spaces, Sobolev spaces and more. These results are valid for all sufficiently smooth activation functions. Afterwards, we provide a unifying framework for the construction of approximate partitions of unity by neural networks with fairly general activation functions. This allows us to approximate localized Taylor polynomials by neural networks and make use of the Bramble–Hilbert Lemma. Based on our framework, we derive almost optimal upper bounds in higher-order Sobolev norms. This work advances the theory of approximating solutions of partial differential equations by neural networks.

1. Introduction

Deep learning algorithms have lately shown promising results for dealing with classical mathematical problems, such as the solution of *partial differential equations (PDEs)*, see for instance (Beck et al., 2018, 2019; E et al., 2017; E & Yu, 2018; Elbrächter et al., 2018; Geist et al., 2020; Grohs et al., 2018; Han et al., 2018, 2020; Jentzen et al., 2018; Kutyniok et al., 2019; Laakmann & Petersen, 2020; Lagaris et al., 1998; Lu et al., 2019; Schwab & Zech, 2019; Sirignano & Spiliopoulos, 2018). In this work, we investigate the necessary and sufficient number of non-zero, encodable[2] weights for a vanilla feedforward neural network to approximate functions that are particularly relevant for the solution of PDEs. Notable works in this direction for neural networks with the ReLU (rectified linear unit) activation function are (Gühring et al., 2020; Opschoor et al., 2020). Due to the limited regularity of the ReLU, one is only able to derive approximation rates with respect to first-order Sobolev norms. However, in order to appropriately approximate solutions of PDEs of higher-order (i.e., of order ≥ 3), approximation rates with respect to higher-order Sobolev norms are required. As an example, consider the *Dirichlet problem for the biharmonic operator* Δ^2 (see e.g. Ciarlet (2002)) on some domain $\Omega \subset \mathbb{R}^d$, a typical fourth-order problem, which is given by

$$-\Delta^2 u = f, \quad \text{on } \Omega \quad + \text{boundary conditions.} \tag{1.1}$$

In its weak formulation, this operator equation is uniquely solvable in some subspace V (incorporating the boundary conditions) of the Sobolev space $W^{2,2}(\Omega)$. Additionally (see Ciarlet (2002, Section 6)), typical solutions u of (1.1) are even in the Sobolev space $W^{n,2}(\Omega)$ for some $n \geq 3$. This motivates studying approximations of Sobolev-regular functions $f \in W^{n,p}(\Omega)$ by neural networks in higher-order Sobolev norms. In this paper, we make the following two contributions:

I. General lower bounds based on entropy arguments

Let $\mathcal{C} \subset \mathcal{D}$ be two function spaces. We will lower bound the necessary number for nonzero, encodable weights of neural network approximations of functions from $\mathcal{C}$ with respect to the norm in $\mathcal{D}$. Our notion of a lower bound for the number of nonzero, encodable weights can be summarized as follows:

For some $\gamma > 0$ (depending on $\mathcal{C}$ and $\mathcal{D}$) we have: If for every $\varepsilon > 0$ there exists some $M_\varepsilon \in \mathbb{N}$ such that every $f \in \mathcal{C}$ can be ε-approximated by a neural network $\Phi_{\varepsilon,f}$ (i.e., $\|f - \Phi_{\varepsilon,f}\|_{\mathcal{D}} \leq \varepsilon$) with M_ε nonzero, encodable weights, then (up to a logarithmic factor and for some constant C) $M_\varepsilon \geq C\varepsilon^{-\gamma}$.

In Petersen and Voigtländer (2018), the concept of the *ε-entropy* $H_\varepsilon(\mathcal{C}, \mathcal{D})$ was used to derive lower bounds for M_ε for specific choices of $\mathcal{C}$ and $\mathcal{D}$. In Theorem 3.5 we generalize that approach to a wide range of function spaces. In detail, we show that every lower bound on the *ε-entropy* $H_\varepsilon(\mathcal{C}, \mathcal{D})$ of the unit ball of $\mathcal{C}$ with respect to $\|\cdot\|_{\mathcal{D}}$ can directly be transferred to a lower bound on

[*] Corresponding author.
E-mail addresses: guehring@math.tu-berlin.de (I. Gühring), raslan@math.tu-berlin.de (M. Raslan).

[1] Both authors contributed equally.

[2] I.e., representable by a bit-string of moderate length.

https://doi.org/10.1016/j.neunet.2020.11.010
0893-6080/© 2020 Elsevier Ltd. All rights reserved.

the number of nonzero, encodable weights of an approximating neural network. Concretely, if $H_\varepsilon(\mathcal{C}, \mathcal{D}) \geq C\varepsilon^{-\gamma}$, then $M_\varepsilon \geq C\varepsilon^{-\gamma}/\log_2(1/\varepsilon)$. Since the activation function ϱ determines the smoothness of $\Phi_{\varepsilon,f}$ we only have the natural requirement that ϱ is smooth enough such that $\Phi_{\varepsilon,f} \in \mathcal{D}$.

Since lower bounds on the ε-entropy are well-studied for a variety of classical function spaces,[3] we give a nonexhaustive list of concrete lower complexity bounds in Corollary 3.8 for Sobolev and Besov spaces. Appositely to the upper bounds that we present below, we state the following special instance of these results: For $\mathcal{C} = W^{n,p}(\Omega)$ and $\mathcal{D} = W^{k,p}(\Omega)$ with $n, k \in \mathbb{N}_0,\ n > k$ and $1 \leq p \leq \infty$ we have $M_\varepsilon \geq C\varepsilon^{-d/(n-k)}/\log_2(1/\varepsilon)$.

II. Almost optimal upper bounds in Sobolev spaces for a wide class of activation functions

We build an abstract, unifying framework which allows to approximate localized Taylor polynomials by neural networks with a wide class of activation functions. This proof strategy was originally used in Yarotsky (2017) for ReLU neural networks in L^p-norms and generalized to first-order Sobolev norms in Gühring et al. (2020). Those works heavily rely on the ReLU activation function which allows for the construction of an exact partition of unity (PU). However, constructing localized bump functions that together form a PU by neural networks with general activation function is highly-nontrivial and can, in general, only be done approximately. This means that the localizing bump functions are not compactly supported anymore and their sum only approximates the one-function. We formulate conditions on the asymptotic behavior of the activation function under which such a construction becomes possible in higher-order Sobolev spaces. For this, we derive three distinct categories of PUs splitting the domain $(0, 1)^d$ into $(N+1)^d$ patches with diameter $1/N$.

- *Exact PU*: The $(N+1)^d$ localizing bump functions are compactly supported on the corresponding patch and the sum of the bumps equals one.
- *Exponential PU*: For $N \to \infty$, the bumps converge exponentially fast in N towards an exact PU.
- *Polynomial PU*: For $N \to \infty$, the bumps converge with polynomial speed in N towards an exact PU.

In other words, with increasing refinement of the partition the approximate PUs converge towards an exact PU and are categorized by their convergence speed.

Based on the above categorization, we consider ε-approximations of functions from the unit ball in $\mathcal{C} = W^{n,p}((0,1)^d)$ where the distance is measured in $\mathcal{D} = W^{k,p}((0,1)^d)$ norms ($n \in \mathbb{N}_{\geq k+1}$, $k \in \mathbb{N}_0$ and $1 \leq p \leq \infty$) and derive for each case different approximation rates. We demonstrate this for three representative examples.

- The *rectified power unit (RePU)* of order $j \in \mathbb{N}_{\geq 2}$, given by ReLU^j, allows for the construction of exact PUs. In this case, for every $k \in \{0, \ldots, j\}$, we need at most $C\varepsilon^{-d/(n-k)}$ non-zero weights.
- The *softplus function*, given by $\ln(1 + e^x)$, allows for the construction of exponential PUs. In this case, for $k \in \mathbb{N}_0$, and arbitrary $\mu > 0$, we need at most
$$\begin{cases} C\varepsilon^{-d/(n-k)}, & \text{if } k \leq 1, \\ C\varepsilon^{-d/(n-k-\mu)}, & \text{if } k \geq 2, \end{cases}$$
non-zero weights.
- The *inverse square root linear unit*, given by $\mathbb{1}_{[0,\infty)}x + \mathbb{1}_{(-\infty,0)}\frac{x}{\sqrt{1+x^2}}$, allows for the construction of polynomial PUs. In this case, for $k \in \{0, 1\}$, we need at most $C\varepsilon^{-d/(n-k)}$ non-zero weights.
Generally speaking, in the case of polynomial PUs, we are only able to show approximation rates in smoothness norms of a restricted order, depending on the asymptotic behavior of the underlying activation function. We describe the reasons for this issue in more detail in Section 4.3.

In all cases the depth of the constructed networks is constant (i.e. accuracy-independent) and greater than two. Afterwards, we additionally show that the weights of $\Phi_{\varepsilon,f}$ can be encoded by $C\log_2(1/\varepsilon)$ bits which guarantees that the approximation complexity is not hidden in weights carrying arbitrarily complex information.

As already outlined in Gühring et al. (2020, Section 1.4), we observe in both, lower and upper bounds, a *trade-off* between the complexity of the approximating neural networks and the order of the approximation norm: A higher order of k requires neural networks with asymptotically more nonzero weights. Additionally, up to a log-factor (and in some cases up to $\mu > 0$), our upper bounds are *tight* if we only allow encodable weights.

Related work

The universal approximation theorem (Cybenko, 1989; Hornik, 1991) is often regarded as the starting point of approximation theory for neural networks. It shows that every continuous function defined on a compact domain can be uniformly approximated by shallow neural networks under some assumptions on the activation function. Extensions of this theorem (see Pinkus (1999, Section 4) and the references therein) also take derivatives into account. In more detail, it has been established that shallow neural networks with sufficiently regular activation function and unrestricted width are dense in the space C^m, where $m \in \mathbb{N}$. The existence of an activation function such that restricted width and depth networks are universal is shown in Maiorov and Pinkus (1999) and an explicit activation function based on the countability of the rational numbers with that property is constructed in Guliyev and Ismailov (2018). For ReLU networks with restricted width and unbounded depth universality is established in Kidger and Lyons (2019).

The necessary and sufficient complexity of (higher-order) sigmoidal neural network approximations for (piecewise) smooth functions has been studied in Barron (1994), Bölcskei et al. (2019) and Mhaskar (1996).[4] The results in Mhaskar (1996) for function approximation in L^p are derived by approximating global (not localized) polynomials with degree increasing concurrently with the approximation accuracy. Our results include these approximation rates as a special case based on an alternative proof strategy. The ansatz in Mhaskar (1996) can be used for C^∞ activation functions with non vanishing derivatives at some point to obtain network approximations with constant depth and increasing width. Vanishing derivatives of the activation function need to be compensated by increasing depth in order to construct polynomials of increasing degree. This approach is utilized in Li et al. (2020) and Tang et al. (2019), where approximations of weighted L^2-spaces by RePU-neural networks are derived.[5] The function spaces considered therein can be efficiently described by non-localized (Jacobi or Chebyshev) polynomials. Complexity bounds for ReLU neural networks based on *localized* polynomial approximation can be found in Ohn and Kim (2019), Petersen and Voigtländer

[3] See for instance Edmunds and Triebel (1996), Triebel (1978).

[4] In Ohn and Kim (2019) rates for locally quadratic activation functions are formulated. However, in the crucial (Ohn & Kim, 2019, Lemma A.3.(d)), there is, in its present form, a gap in the author's reasoning.

[5] Which are able to represent polynomials with zero error.

(2018), Schmidt-Hieber (2017), Suzuki (2019) and Yarotsky (2017). The upper bounds in Yarotsky (2017, Thm. 1) are covered by our framework as a special case. In Petersen and Voigtländer (2018), localization is achieved by approximating characteristic functions. Our notion of PUs is general enough to include this approach but we focus on different function classes. Localization by means of wavelet approximations on manifolds is utilized in Shaham et al. (2018) and by means of general affine systems in Bölcskei et al. (2019). The approximation error in all of these papers is measured with respect to L^p-norms. Only the papers (Bölcskei et al., 2019; Petersen & Voigtländer, 2018) consider the restriction of encodable weights.

In this paper we are primarily interested in the approximation of functions with respect to Sobolev norms. In this direction, we mention two works, which examine the approximation capabilities of ReLU-neural networks with respect to $W^{1,p}$ norms. The paper (Gühring et al., 2020) derives lower complexity bounds based on a VC dimension argument for unrestricted neural network weights (similar to the one presented in Yarotsky (2017)) and upper bounds based on the emulation of localized polynomials for continuous, piecewise linear activation functions. These upper bounds are included in our results as a special case. In Opschoor et al. (2020) approximation rates were derived by re-approximating finite elements. None of these papers examine neural networks with encodable weights.

We conclude this section by giving an overview of further works that introduce different types of PUs. An approach which is similar to ours for functions of sigmoidal type has been used in Costarelli and Spigler (2013a, 2013b) and Costarelli et al. (2019). There, approximate bumps are constructed from differences of scaled and shifted sigmoidals. The key difference is that for a fixed patch the contributions of the neighboring approximate bump functions do not decrease with the number of patches N going to infinity which is an important factor in our construction. In Lin (2019), characteristic functions χ_p for each patch are L^∞-approximated in order to achieve localization. However, in this work, the Heaviside function is used as an activation function in the first layer (followed by a different activation function in the next layer), which is not transferable to our work, since it prevents higher-order Sobolev approximations.

Outline

After having introduced the necessary terminology for neural networks in Section 2, we start by proving general lower complexity bounds in Section 3. In Section 4, we derive almost optimal upper approximation rates for neural networks with fairly general activation functions. We describe the necessary ingredients for these results in Sections 4.1 and 4.2 before outlining the main results as well as the underlying proof strategy in Section 4.3. The proofs of the two main results in this section, Proposition 4.8 and Theorem 4.9, can be found in Appendices D and E, respectively. To not interrupt the flow of reading, the notation section, basic facts about Sobolev spaces and basic operations one can perform with neural networks have been deferred to Appendices A–C, respectively. An analysis of the PU-properties of many practically used activation functions can be found in Appendix F.

2. Neural networks with encodable weights: Terminology

We start by formally introducing neural networks closely sticking to the notions introduced in Petersen and Voigtländer (2018). In the following, we will distinguish between a *neural network* as a structured set of weights and the associated function implemented by the network, called its *realization*. Towards this goal, let us fix numbers $L, d = N_0, N_1, \ldots, N_L \in \mathbb{N}$.

- A family $\Phi = \big((A_\ell, b_\ell)\big)_{\ell=1}^L$ of matrix–vector tuples of the form $A_\ell \in \mathbb{R}^{N_\ell, N_{\ell-1}}$ and $b_\ell \in \mathbb{R}^{N_\ell}$ is called *neural network*.
- We refer to the entries of A_ℓ, b_ℓ as the weights of Φ and call $M(\Phi) := \sum_{\ell=1}^L (\|A_\ell\|_0 + \|b_\ell\|_0)$ its *number of nonzero weights*, $L = L(\Phi)$ its *number of layers* and we call N_ℓ the *number of neurons in layer* ℓ.
- We denote by $d := N_0$ the *input dimension* of Φ and by N_L the *output dimension*.
- Moreover, we set
$$\|\Phi\|_{\max} := \max_{\ell=1,\ldots,L} \max_{\substack{i=1,\ldots,N_\ell \\ j=1,\ldots,N_{\ell-1}}} \max\{|(A_\ell)_{i,j}|, |(b_\ell)_i|\},$$
which is the *maximum absolute value of all weights*.
- For defining the realization of a network $\Phi = \big((A_\ell, b_\ell)\big)_{\ell=1}^L$, we additionally fix an *activation function* $\varrho : \mathbb{R} \to \mathbb{R}$ and a set $\Omega \subset \mathbb{R}^d$. The *realization of the network* $\Phi = \big((A_\ell, b_\ell)\big)_{\ell=1}^L$ is the function
$$R_\varrho(\Phi) : \Omega \to \mathbb{R}^{N_L}, \quad x \mapsto x_L,$$
where x_L results from the following scheme:
$$\begin{aligned} x_0 &:= x, \\ x_\ell &:= \varrho(A_\ell x_{\ell-1} + b_\ell), \qquad \text{for } \ell = 1, \ldots, L-1, \\ x_L &:= A_L x_{L-1} + b_L, \end{aligned}$$
and where ϱ acts componentwise.
- We denote by $\mathcal{NN}_\varrho^d$ the *set of all ϱ-realizations of neural networks with input dimension d and output dimension* 1.[6]

Encodability

In the following, we study neural networks with *encodable* weights. This information-theoretic viewpoint has already been examined in Bölcskei et al. (2019) and Petersen and Voigtländer (2018) and is motivated by the observation that on a computer only weights of limited complexity (w.r.t. their bit-length) can be stored. In this paper, we consider weights that can be encoded by bit-strings with length logarithmically growing in $1/\varepsilon$, where ε is the approximation accuracy.

To make the notion of encodability more precise, we first introduce coding schemes (see Petersen and Voigtländer (2018)): A *coding scheme (for real numbers)* is a sequence $\mathcal{B} = (B_\ell)_{\ell\in\mathbb{N}}$ of maps $B_\ell : \{0,1\}^\ell \to \mathbb{R}$. Now we define sets of neural networks with weights encodable by a coding scheme. Given an arbitrary coding scheme $\mathcal{B} = (B_\ell)_{\ell\in\mathbb{N}}$, and $d \in \mathbb{N}$, $\varepsilon, M > 0$, we denote by
$$\mathcal{NN}^{\mathcal{B}}_{M, \lceil C_0 \log_2(1/\varepsilon)\rceil, d} \tag{2.1}$$
the set of all neural networks Φ with d-dimensional input, one-dimensional output and at most M nonzero weights such that *each nonzero weight of Φ is contained in* $\operatorname{Range}(B_{\lceil C_0 \log_2(1/\varepsilon)\rceil})$.

3. Lower bounds for neural networks with encodable weights and general activation functions

In this section, we derive lower bounds on the necessary number of nonzero, encodable weights of neural network approximations. The approximated function spaces include a wide variety of classical smoothness spaces and the accuracy is measured in rather general norms. Our result applies to every activation function that is sufficiently smooth to be considered in these norms. We note that the proof of our result is essentially an abstract version of the proof of Petersen and Voigtländer (2018, Theorem

[6] In the following we will denote by *(ϱ-)neural networks* both neural networks and their corresponding realizations as long it is clear from the context what is meant.

4.2). After encouragement of one of the authors[7] of Petersen and Voigtländer (2018) and after studying the paper more closely, we noticed that it is possible to consider the proof strategy of Petersen and Voigtländer (2018, Theorem 4.2) in a very general setting which we will outline below. Throughout this section (unless stated otherwise) we fix some $d \in \mathbb{N}$, some domain $\Omega \subset \mathbb{R}^d$ and two normed spaces $\mathcal{C}, \mathcal{D}$ of (equivalence classes of) functions defined on Ω with values in $\mathbb{R}$. Additionally, we assume that $\mathcal{C} \subset \mathcal{D}$.

First of all, we need the notion of the *minimax code length* $L_\varepsilon(\mathcal{C}, \mathcal{D})$ *of* $\mathcal{C}$ *with respect to* $\mathcal{D}$. The minimax code length describes the uniform description complexity of the set $\{f \in \mathcal{C} : \|f\|_{\mathcal{C}} \le 1\}$ in terms of the number of nonzero bits necessary to encode every f with distortion at most ε in $\mathcal{D}$. It can be directly related to approximation capabilities of arbitrary computing schemes and is defined as follows (see also (Petersen & Voigtländer, 2018, Definition B.2)):

Definition 3.1 (*Minimax Code Length*)**.** Let $\ell \in \mathbb{N}$. We denote by $\mathfrak{E}^\ell := \left\{E : \mathcal{C} \to \{0,1\}^\ell\right\}$ the set of binary encoders mapping elements of $\mathcal{C}$ to bit strings of length ℓ, and by $\mathfrak{D}^\ell := \{D : \{0,1\}^\ell \to \mathcal{D}\}$ the set of binary decoders mapping bit-strings of length ℓ into $\mathcal{D}$. For $\varepsilon > 0$, we define the *minimax code length* by

$$L_\varepsilon(\mathcal{C}, \mathcal{D}) := \min\left\{\ell \in \mathbb{N} : \exists (E^\ell, D^\ell) \in \mathfrak{E}^\ell \times \mathfrak{D}^\ell : \sup_{f \in \mathcal{C} : \|f\|_{\mathcal{C}} \le 1} \|D^\ell(E^\ell(f)) - f\|_{\mathcal{D}} \le \varepsilon\right\}.$$

The next observation demonstrates in the context of neural networks how the minimax code length can be employed to derive lower bounds for approximations.

Observation 3.2. *Let $\varepsilon > 0$ and $\varrho : \mathbb{R} \to \mathbb{R}$ such that $\mathcal{NN}^d_\varrho \subset \mathcal{D}$. If $\mathcal{A}$ is a neural network architecture with M unspecified nonzero weights*[8] *(but fixed number of layers, neurons and position of nonzero weights) such that for each $f \in \mathcal{C}$ there is a set of weights $w_1, \ldots, w_M$, where each weight can be encoded by at most $b \in \mathbb{N}$ bits and $\|R_\varrho(\mathcal{A}(w_1, \ldots, w_M)) - f\|_{\mathcal{D}} \le \varepsilon$, then*

$$M \ge L_\varepsilon(\mathcal{C}, \mathcal{D})/b.$$

Mapping $f \in \mathcal{C}$ to the bit representation of the M weights can be viewed as an encoder, and mapping the encoded weights to $R_\varrho(\mathcal{A}(w_1, \ldots, w_M))$ acts as a decoder with bit length $\ell = Mb$, which shows the claim. This in particular holds true, if $b \le C\log_2(1/\varepsilon)$ which is the focus of this paper.

In the following, we exploit this strategy to show that the same bound actually holds true, if we allow for the architecture to depend on the function to be approximated. That means, for each $f \in \mathcal{C}$ the number of layers, neurons and position of M nonzero encodable weights (and the weights themselves) may change but need to be encoded. The next lemma (shown in Petersen and Voigtländer (2018, Lemma B.4) under the additional restriction[9] that $\varrho(0) = 0$) shows the number of bits needed to encode this information.

Lemma 3.3. *Let $M, K \in \mathbb{N}$, and let $\mathcal{B}$ be an encoding scheme for real numbers and $\varrho : \mathbb{R} \to \mathbb{R}$ an activation function. There is a constant $C = C(d)$, such that there is an injective map $\Gamma : \{R_\varrho(\Phi) : \Phi \in \mathcal{NN}^{\mathcal{B}}_{M,K,d}\} \to \{0,1\}^{CM(K+\lceil\log_2 M\rceil)}$.*

To make the main statement of this section mathematically more precise, we introduce some further notation.

Definition 3.4. Let $C_0 > 0$ be fixed. Additionally, let $f \in \mathcal{C}$, and for some function $\varrho : \mathbb{R} \to \mathbb{R}$ assume that $\mathcal{NN}^d_\varrho \subset \mathcal{D}$. Finally, let $\varepsilon > 0$ and fix some coding scheme $\mathcal{B}$. Then, for $C_0 > 0$, we define the quantities[10]

$$\begin{aligned} &M^{\mathcal{B}}_\varepsilon(f) \\ &:= M^{\mathcal{B},\varrho,C_0,\mathcal{C},\mathcal{D}}_\varepsilon(f) \\ &:= \min\left\{M \in \mathbb{N} : \exists \Phi \in \mathcal{NN}^{\mathcal{B}}_{M,\lceil C_0\cdot\log_2 \frac{1}{\varepsilon}\rceil,d} : \|f - R_\varrho(\Phi)\|_{\mathcal{D}} \le \varepsilon\right\}, \end{aligned}$$

and

$$M^{\mathcal{B}}_\varepsilon(\mathcal{C}, \mathcal{D}) := M^{\mathcal{B},\varrho,C_0}_\varepsilon(\mathcal{C}, \mathcal{D}) := \sup_{f \in \mathcal{C},\ \|f\|_{\mathcal{C}} \le 1} M^{\mathcal{B},\varrho,C_0,\mathcal{C},\mathcal{D}}_\varepsilon(f).$$

In other words, the quantity $M^{\mathcal{B}}_\varepsilon(f)$ denotes the required number of nonzero weights of a neural network Φ to ε-approximate f with weights that can be encoded with $\lceil C_0 \log_2(1/\varepsilon)\rceil$ bits using the coding scheme $\mathcal{B}$. $M^{\mathcal{B}}_\varepsilon(\mathcal{C}, \mathcal{D})$ gives a uniform bound of this quantity over the unit ball in $\mathcal{C}$.

Theorem 3.5 now states that if we can lower bound the minimax code length, then we are also able to lower bound $M^{\mathcal{B}}_\varepsilon(\mathcal{C}, \mathcal{D})$. Lower bounds on the minimax code length (and hence for the quantity $M^{\mathcal{B}}_\varepsilon(\mathcal{C}, \mathcal{D})$) for specific, frequently used function spaces fulfilling the assumptions of the theorem will be given in Corollary 3.8.

Theorem 3.5. *Let $\varrho : \mathbb{R} \to \mathbb{R}$ such that $\mathcal{NN}^d_\varrho \subset \mathcal{D}$. Additionally, assume that $L_\varepsilon(\mathcal{C}, \mathcal{D}) \ge C_1\varepsilon^{-\gamma}$ for some $\gamma = \gamma(\mathcal{C}, \mathcal{D}), C_1 = C_1(\mathcal{C}, \mathcal{D}) > 0$ and all $\varepsilon > 0$. Then, for each $C_0 > 0$ there exists a constant $C = C(\gamma, \mathcal{C}, \mathcal{D}, C_0) > 0$, such that for each coding scheme of real numbers $\mathcal{B}$, and for all $\varepsilon \in (0, 1/2)$ we have*

$$M^{\mathcal{B},\varrho,C_0}_\varepsilon(\mathcal{C}, \mathcal{D}) \ge C \cdot \varepsilon^{-\gamma} \Big/ \log_2\left(\frac{1}{\varepsilon}\right).$$

The idea for the proof of this theorem is the same as for Observation 3.2. Here, the encoder is $E : \mathcal{C} \to \{0,1\}^\ell, f \mapsto \Gamma(R_\varrho(\Phi_{\varepsilon,f}))$, where $\Phi_{\varepsilon,f}$ is the neural network ε-approximating f, Γ is the network encoder from Lemma 3.3 and $\ell = CM(\log_2(1/\varepsilon) + \log_2(M))$. The decoder is given by $D : \{0,1\}^\ell \to \mathcal{C}, b \mapsto \Gamma^{-1}(b)$. The bound now follows from $CM(\log_2(1/\varepsilon) + \log_2(M)) \ge C_1\varepsilon^{-\gamma}$.

Remark 3.6 (*Activation Functions*)**.** We only require sufficient smoothness of the activation function for the spaces under consideration. Hence, we are in a position to conclude suitable lower bounds for *all* practically used activation functions.

Remark 3.7 (*Bounds with Non-Encodable Weights*)**.** If one drops the restriction of encodable weights and considers the more general setting of arbitrary weights, a lesser number of weights is required in general. For this setting, we mention two examples.

- The results from Gühring et al. (2020), Yarotsky (2018) combined state:
  For $\mathcal{C} = W^{n,\infty}((0,1)^d)$ and $\mathcal{D} = W^{k,\infty}((0,1)^d)$ with $k = 0, 1$, it holds for the necessary number of nonzero weights M_ε to achieve an ε-approximation in $W^{k,\infty}$ norm that

  $$M_\varepsilon \ge C\varepsilon^{-d/(2n-k)}.$$

[7] We want to take the opportunity to thank Philipp Petersen for the fruitful suggestion.

[8] Or any computation scheme that takes as input M parameters.

[9] The lemma is proven by first noting that a network with arbitrary number of neurons and layers, but M non-zero weights, can be replaced by a network with the same number of non-zero weights, but number of neurons and layers bounded by $M + 1$. This can be done by removing neurons that do not contribute to the next layer. This strategy (see also (Bölcskei et al., 2019, Proposition 3.6)) allows us to drop the assumption that $\varrho(0) = 0$ from Petersen and Voigtländer (2018, Lemma B.4).

[10] We use the convention that $\min \varnothing = \infty$.

For $k = 0$, in Yarotsky (2018) neural networks are constructed that achieve this approximation rate. In comparison, our entropy bounds show that under the assumption of encodable weights $M_\varepsilon \geq C\varepsilon^{-d/(n-k)}$ (suppressing the $\log_2(1/\varepsilon)$ factor for simplicity of exposition).

- In Guliyev and Ismailov (2016) it is shown that there exists an activation function such that a neural network with three parameters is able to uniformly approximate each function in $\mathcal{C} = C([0,1])$ arbitrary well. Observation 3.2 now shows that there is no finite encoding bit length for the weights necessary to approximate all functions in the unit ball of $C([0,1])$, since in this case $L_\varepsilon(\mathcal{C},\mathcal{C}) = \infty$ for $0 < \varepsilon < 1$.[11]

We proceed by listing a variety of lower bounds for a selection of specific examples for frequently used function spaces. Here, we make use of the fact that, by Grohs et al. (2018, Remark 5.10), the ε-entropy $H_\varepsilon(\mathcal{C},\mathcal{D})$ is bounded by the minimax code length, i.e., $L_\varepsilon(\mathcal{C},\mathcal{D}) \geq H_\varepsilon(\mathcal{C},\mathcal{D})$. One can deduce similar lower bounds for other choices of $\mathcal{C},\mathcal{D}$. Notable examples that are not covered below include Hölder spaces, Triebel–Lizorkin, or Zygmund spaces (see for instance Edmunds and Triebel (1996), Triebel (1978) and the references therein for further examples).

Corollary 3.8. *Assume that Ω fulfills some regularity conditions.*[12] *Let $\varrho : \mathbb{R} \to \mathbb{R}$ be chosen such that $\mathcal{NN}^d_\varrho \subset \mathcal{D}$ (where $\mathcal{D}$ is a function space on Ω specified below). Moreover, let $\mathcal{B}$ be an arbitrary coding scheme. Then, the following statements hold:*

(i) ***Besov spaces:*** *Let $s,t \in \mathbb{R}$ with $s < t$ as well as $p_1,p_2,q_1,q_2 \in (0,\infty]$ such that*

$$t - s - d\max\left\{\left(\frac{1}{p_1} - \frac{1}{p_2}\right), 0\right\} > 0.$$

Moreover, let $\mathcal{C} = B^t_{p_1,q_1}(\Omega)$, and $\mathcal{D} = B^s_{p_2,q_2}(\Omega)$. Then, for some $C > 0$, we have

$$M^{\mathcal{B}}_\varepsilon(\mathcal{C},\mathcal{D}) \geq C\varepsilon^{-\frac{d}{t-s}} \Big/ \log_2\left(\frac{1}{\varepsilon}\right), \qquad \text{for all } \varepsilon \in (0, 1/2).$$

(ii) ***Sobolev spaces:*** *Let $s,t \in \mathbb{N}$ with $t > s$ and let $p \in (0,\infty]$. Then, for $\mathcal{C} = W^{t,p}(\Omega)$ and for $\mathcal{D} = W^{s,p}(\Omega)$ there exists some $C > 0$ with*

$$M^{\mathcal{B}}_\varepsilon(\mathcal{C},\mathcal{D}) \geq C\varepsilon^{-\frac{d}{t-s}} \Big/ \log_2\left(\frac{1}{\varepsilon}\right), \qquad \text{for all } \varepsilon \in (0, 1/2).$$

Proof. (i) follows immediately from Theorem 3.5 in combination with Theorem (Edmunds & Triebel, 1996, Section 3.5).

(ii) follows from Theorem 3.5 together with Edmunds and Evans (2004, Section 1.3), where we use the estimate on the approximation number $a_k(id)$ (cf. page 9) combined with the relation of $a_k(id)$ and the entropy. □

4. Upper bounds for general activation functions in Sobolev spaces

In this section, we show that for an arbitrary accuracy $\varepsilon > 0$, every function from the unit ball of the Sobolev space $W^{n,p}$

$$\mathcal{F}_{n,d,p} := \{f \in W^{n,p}((0,1)^d) : \|f\|_{W^{n,p}((0,1)^d)} \leq 1\}$$

can be ε-approximated in weaker $W^{k,p}$-Sobolev norms (with $n > k$) by neural networks with fairly general activation function. For this, we explicitly construct approximating neural networks with constant depth (i.e., independent of ε) and give upper bounds for the number of nonzero, encodable weights (depending on ε), which, in light of the results of Section 3, are almost optimal. The main idea is based on the common strategy (see e.g. Gühring et al. (2020), Schmidt-Hieber (2017) and Yarotsky (2017)) of approximating f by localized polynomials which in turn are approximated by neural networks. Our work differs from these other works in three major aspects:

(a) Depending on the smoothness j of the activation function, our approximations include $W^{k,p}$ for $k \leq j$ (instead of maximally $W^{1,p}$).

(b) Constructing a PU by neural networks with general activation function is tricky (contrary to ReLU networks) and can, in general, only be done approximately (see Section 4.1 and Fig. 1).

(c) Our polynomial approximations and approximate PUs have depth independent of ε, which results in constant-depth approximations of f.

We construct localizing bump functions that form an *(approximate) partition of unity* in Section 4.1 and efficiently *approximate polynomials* by neural networks in Section 4.2. Afterwards, the statements of the main results as well as a detailed overview of their overall proof strategies are given in Section 4.3.

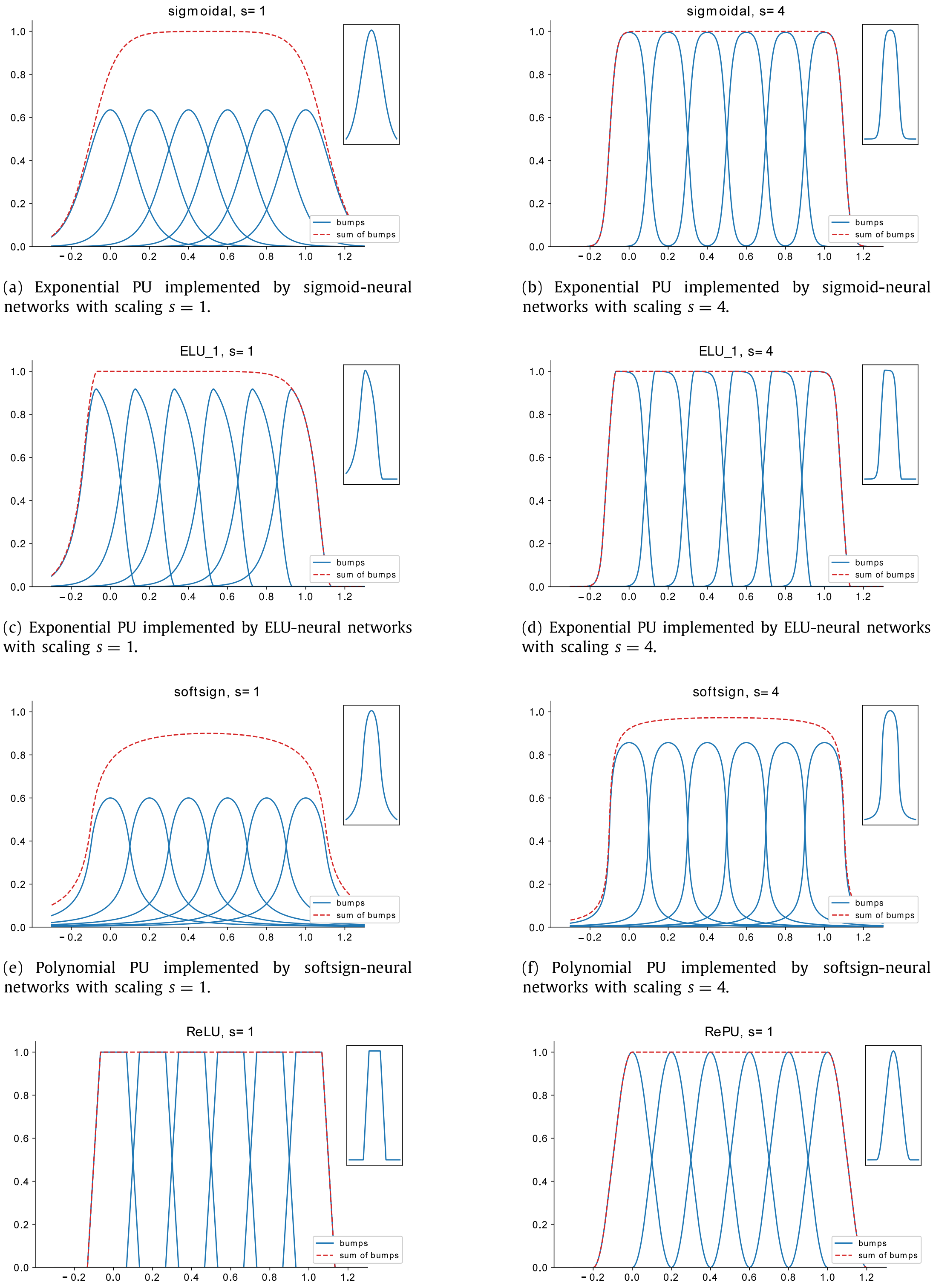
4.1. Ingredient I: (Approximate) partition of unity

In Gühring et al. (2020) and Yarotsky (2017) the ReLU activation function is used to construct continuous, piecewise linear bump functions with compact support that form a PU. However, this approach heavily relies on properties of the ReLU and is only suitable for approximations in Sobolev norms up to order one. For general activation functions, there is, to the best of our knowledge, no canonical way to build a PU by neural networks. As a remedy we introduce approximate partitions of unity which are compatible with all practically used activation functions. In detail, for a gridsize $1/N$ (with $N \in \mathbb{N}$), we divide the domain $(0,1)^d$ into $(N+1)^d$ equally large patches and construct, for each patch Ω_m where $m \in \{0,\ldots,N\}^d$, a bump function $\phi_m \in W^{j,\infty}$. Deviating from usually used bump functions, ϕ_m is in general not compactly supported on the corresponding patch and their sum only approximates $\mathbb{1}_{(0,1)^d}$, i.e., $\sum_m \phi_m \approx \mathbb{1}_{(0,1)^d}$. Additionally, we introduce a scaling factor $s \geq 1$, which regulates the closeness of the approximate PU to an exact PU. For $s \to \infty$, we have that $\|\phi^s_m\|_{\Omega^c_m} \to 0$ and $\sum_m \phi^s_m \to \mathbb{1}_{(0,1)^d}$. The overall approximation rates in our main result now also depend on properties of the approximate PU. It will later turn out that the speed of the convergence of the approximate PU is the decisive factor in showing efficient approximation rates. We distinguish between exponential and polynomial speed. Besides the smoothness j and the convergence speed there is another defining quantity τ which we call the *order of the PU*. The order τ specifies at which derivative the scaling factor starts to show. In other words, all derivatives up to order τ absorb the effect of the scaling. In Definition 4.1 we formally introduce the notion of an approximate PU. Additionally to approximate PUs with exponential and polynomial convergence properties we also include exact PUs in this definition since these include (leaky) ReLUs and powers thereof.

Definition 4.1. Let $d \in \mathbb{N}$, $j,\tau \in \mathbb{N}_0$. We say that the collection of families of functions $(\Psi^{(j,\tau,N,s)})_{N\in\mathbb{N},s\in\mathbb{R}_{\geq 1}}$, where each $\Psi^{(j,\tau,N,s)} := \{\phi^s_m : m \in \{0,\ldots,N\}^d\}$ consists of $(N+1)^d$ functions $\phi^s_m : \mathbb{R}^d \to \mathbb{R}$,

[11] $L_\varepsilon(\mathcal{C},\mathcal{C}) = \infty$ for $0 < \varepsilon < 1$ follows from the fact that the unit ball in $\mathcal{C} = C([0,1])$ is not compact. The same argument can also be used to directly deduce from the construction of the weights in Guliyev and Ismailov (2016) that their encoding bit length is not finite.

[12] Many results estimating the ε-entropy are only formulated and proven for C^∞-domains for simplicity of exposition. However, as has been described in Triebel (1978, Section 4.10.3) and Edmunds and Triebel (1996, Section 3.5), these results remain valid for function spaces on more general domains including cubes.



(a) Exponential PU implemented by sigmoid-neural networks with scaling $s = 1$.

(b) Exponential PU implemented by sigmoid-neural networks with scaling $s = 4$.

(c) Exponential PU implemented by ELU-neural networks with scaling $s = 1$.

(d) Exponential PU implemented by ELU-neural networks with scaling $s = 4$.

(e) Polynomial PU implemented by softsign-neural networks with scaling $s = 1$.

(f) Polynomial PU implemented by softsign-neural networks with scaling $s = 4$.

(g) Exact PU implemented by ReLU-neural networks used in Gühring et al. (2020), Yarotsky (2017).

(h) Exact PU implemented by quadratic RePU-neural networks ($\tau = 2$).

Fig. 1. All displayed partitions of unity have 6 bumps ($N = 5$). The red curve shows the sum of the bump functions. A single bump function can be seen in the small window in the upper right part of each plot. The first two rows depict an exponential PU for $\tau = 0$ (first row) and $\tau = 1$ (second row). A polynomial PU of order $\tau = 0$ can be seen in the third row. The impact of increasing the scaling factor s can be seen in the second column. In the last row two exact PUs are shown. Here, the sum is constant 1 on $(0, 1)$ and scaling has no impact.

is an *exponential* (respectively *polynomial*, *exact*) *partition of unity of order* τ *and smoothness* j, or short *exponential* (*polynomial*, *exact*) (j,τ)*-PU*, if the following conditions are met:

There exists some $D > 0$, $C = C(k,d) > 0$ and $S > 0$ such that for all $N \in \mathbb{N}, s \geq S, k \in \{0,\dots,j\}$ the following properties hold:

(i) $\|\phi_m^s\|_{W^{k,\infty}(\mathbb{R}^d)} \leq CN^k \cdot s^{\max\{0,k-\tau\}}$ for every $\phi_m^s \in \Psi^{(j,\tau,N,s)}$;

(ii) for $\Omega_m^c = \left\{x \in \mathbb{R}^d : \ \|x - \frac{m}{N}\|_\infty \geq \frac{1}{N}\right\}$, we have

$$\|\phi_m^s\|_{W^{k,\infty}(\Omega_m^c)} \leq \begin{cases} CN^k s^{\max\{0,k-\tau\}} e^{-Ds}, & \text{if } \textit{exponential PU}, \\ CN^k s^{\max\{0,k-\tau\}} s^{-D}, & \text{if } \textit{polynomial PU}, \\ 0, & \text{if } \textit{exact PU}, \end{cases}$$

for every $\phi_m^s \in \Psi^{(j,\tau,N,s)}$.

(iii) We have

$$\left\| \mathbb{1}_{(0,1)^d} - \sum_{m\in\{0,\dots,N\}^d} \phi_m^s \right\|_{W^{k,\infty}((0,1)^d)} \leq \begin{cases} CN^k s^{\max\{0,k-\tau\}} e^{-Ds}, & \text{if } \textit{exponential PU}, \\ CN^k s^{\max\{0,k-\tau\}} s^{-D}, & \text{if } \textit{polynomial PU}, \\ 0, & \text{if } \textit{exact PU}. \end{cases}$$

(iv) There exists a function $\varrho : \mathbb{R} \to \mathbb{R}$ such that for each $\phi_m^s \in \Psi$ there is a neural network Φ_m^s with d-dimensional input and d-dimensional output, with two layers and C nonzero weights, that satisfies

$$\prod_{l=1}^d [R_\varrho(\Phi_m^s)]_l = \phi_m^s,$$

and $\|R_\varrho(\Phi_m^s)\|_{W^{k,\infty}((0,1)^d)} \leq CN^k \cdot s^{\max\{0,k-\tau\}}$. Furthermore, for the weights of Φ_m^s it holds that $\|\Phi_m^s\|_{\max} \leq CsN$.

In the next definition, we state sufficient conditions for an activation function ϱ to admit (in the sense of Definition 4.1(iv)) an exponential (polynomial, exact) PU of order τ with smoothness j for $\tau \in \{0,1\}$ and afterwards we explicitly construct the corresponding PUs. For $\tau = 0$ the activation functions are approximately piecewise constant outside of a neighborhood of zero (e.g., sigmoidal) and for $\tau = 1$ approximately piecewise affine-linear outside of a neighborhood of zero (e.g., ELU). The speed they approach their asymptotes with (see (d)) in the next definition) defines the convergence speed of the resulting PU. Furthermore, we require ϱ to be j-smooth.

Definition 4.2. Let $j \in \mathbb{N}_0, \tau \in \{0,1\}$. We say that a function $\varrho : \mathbb{R} \to \mathbb{R}$ is *exponential (polynomial, exact)* (j,τ)*-PU-admissible*, if

(a) ϱ is $\begin{cases} \text{bounded}, & \text{if } \tau = 0, \\ \text{Lipschitz continuous}, & \text{if } \tau = 1; \end{cases}$

(b) There exists $R > 0$ such that $\varrho \in C^j(\mathbb{R} \setminus [-R,R])$;

(c) $\varrho' \in W^{j-1,\infty}(\mathbb{R})$, if $j \geq 1$;

(d) There exist $A = A(\varrho), B = B(\varrho) \in \mathbb{R}$ with $A < B$, some $C = C(\varrho, j) > 0$ and some $D = D(\varrho, j) > 0$ such that

(d.1) $\left|B - \varrho^{(\tau)}(x)\right| \leq Ce^{-Dx}$ (Cx^{-D} if *polynomial*, 0 if *exact*) for all $x > R$;

(d.2) $\left|A - \varrho^{(\tau)}(x)\right| \leq Ce^{Dx}$ ($C'|x|^{-D}$ if *polynomial*, 0 if *exact*) for all $x < -R$;

(d.3) $\left|\varrho^{(k)}(x)\right| \leq Ce^{-D|x|}$ ($C|x|^{-D}$ if *polynomial*, 0 if *exact*) for all $x \in \mathbb{R} \setminus [-R,R]$ and all $k = \tau+1,\dots,j$.

Remark 4.3. To give the reader a better intuition for the above definition we mention the similarity to τ*-degree sigmoidal functions* (see Mhaskar and Micchelli (1992)) defined as $\varrho : \mathbb{R} \to \mathbb{R}$ with

$$\lim_{x\to-\infty} \frac{\varrho(x)}{x^\tau} = 0, \quad \lim_{x\to\infty} \frac{\varrho(x)}{x^\tau} = 1.$$

Roughly speaking, we require the same asymptotic behavior (with the exception that the asymptotes do not need to be 0,1) and additionally that the asymptotes are approached with a certain speed.

In Table 1, we list a large variety of commonly used activation functions and their corresponding PU properties. The proofs of these properties can be found in Appendix F.

In the next definition, we give (depending on τ) a recipe for the construction of a one-dimensional approximate bump from which multi-dimensional bumps are derived via a tensor approach. To give the reader a better impression of the definition below and the role of the scaling factor, we present exponential, polynomial and exact bumps and resulting PUs for different activation functions and scaling s in Fig. 1.

Definition 4.4. Let $j \in \mathbb{N}_0, \tau \in \{0,1\}$. Assume that $\varrho : \mathbb{R} \to \mathbb{R}$ is exponential, polynomial or exact (j,τ)-PU-admissible. We define, for a scaling factor $s \geq 1$, the one-dimensional bump functions

$\psi^s : \mathbb{R} \to \mathbb{R},$

$$\psi^s(x) := \begin{cases} \frac{1}{B-A}\left(\varrho(s(x+3/2)) - \varrho(s(x-3/2))\right), & \text{if } \tau = 0, \\ \frac{1}{s(B-A)}\left(\varrho(s(x+2)) - \varrho(s(x+1))\right. \\ \qquad \left. -\varrho(s(x-1)) + \varrho(s(x-2))\right), & \text{if } \tau = 1. \end{cases}$$

For $N \in \mathbb{N}$, $d \in \mathbb{N}$ and $m \in \{0,\dots,N\}^d$, we define multi-dimensional bumps $\phi_m^s : \mathbb{R}^d \to \mathbb{R}$ as a tensor product of scaled and shifted versions of ψ^s. Concretely, we set

$$\phi_m^s(x) := \prod_{l=1}^d \psi^s\left(3N\left(x_l - \frac{m_l}{N}\right)\right).$$

Finally, for $N \in \mathbb{N}, s \geq 1$, the collection of bump functions is denoted by $\Psi^{(j,\tau,N,s)}(\varrho) := \{\phi_m^s : m \in \{0,\dots,N\}^d\}$.

In the next lemma we show that the conditions from Definition 4.2 together with the construction in Definition 4.4 are indeed sufficient to generate an (approximate) PU.

Lemma 4.5. *Let* $j \in \mathbb{N}_0, \tau \in \{0,1\}$ *and a function* $\varrho : \mathbb{R} \to \mathbb{R}$ *be* exponential (polynomial, exact) (j,τ)-PU-admissible. *Then, the collection of families of functions* $(\Psi^{(j,\tau,N,s)}(\varrho))_{N\in\mathbb{N},s\in\mathbb{R}_{\geq 1}}$ *defined in Definition 4.4 is an exponential (polynomial, exact) PU of order* τ *and smoothness* j.

Proof. The proof of this statement is the subject of Appendix D.1. We only give the proof for exponential PUs. The statement for the other two cases follows analogously. □

We demonstrate in Appendix F the admissibility for many practically-used activation functions. In Table 1 we have included the types of PUs these activation functions induce.

Remark 4.6. Definition 4.2 can be generalized to higher $\tau \geq 2$, resulting in an increasing amount of terms in the definition of a bump. Since most activation functions used in practice are of order $\tau \in \{0,1\}$, we did not introduce this concept for simplicity of exposition. An example of $(\tau \geq 2)$-functions are τ*-order RePUs* (short for *Rectified Power Unit*, see, e.g., Li et al. (2020)), given by ReLU^τ. Due to its obvious connections to *B-splines* of order $\tau + 1$ (see for instance De Boor (2001, Chapter IX)), and their ability to form an exact PU (De Boor, 2001, p. 96) as well as their smoothness properties, it is clear that the resulting system $\left(\Psi^{(\tau,\tau,N,s)}(\mathrm{ReLU}^\tau)\right)_{N\in\mathbb{N},s\geq 1}$ forms an exact (τ,τ)-PU.

Table 1
Commonly-used activation functions, the type of PU they admit and the approximation rates in $W^{k,p}$ in terms of the number of nonzero weights. The rates are provided by Theorem 4.9 and, for the (leaky) ReLU case, in combination with Remark 4.10. The results for the (leaky) ReLU are consistent with those rates derived in Gühring et al. (2020) and Yarotsky (2017). $\mu > 0$ is arbitrary and, unless specified otherwise, $k \in \{0, \dots, j\}$ and $n \geq k+1$.

Name	Given by	Smoothness Boundedness	PU-Decay (j, τ)	Approximation rates $(k \leq j)$
(leaky) ReLU, $a \in [0, 1)$	$\max\{ax, x\}$	$C(\mathbb{R}) \cap W^{1,\infty}_{\mathrm{loc}}(\mathbb{R})$ Unbounded	Exact $(1, 1)$	$\varepsilon^{-d/(n-k)} \log(1/\varepsilon)$
Exponential linear unit (ELU$_a$), $a > 0$, $a \neq 1$	$\begin{cases} x, & x \geq 0 \\ a(e^x - 1), & x < 0 \end{cases}$	$C(\mathbb{R}) \cap W^{1,\infty}_{\mathrm{loc}}(\mathbb{R})$ Unbounded	Exponential $(1, 1)$	$\varepsilon^{-d/(n-k)}$
Exponential linear unit (ELU$_1$)	$\begin{cases} x, & x \geq 0 \\ e^x - 1, & x < 0 \end{cases}$	$C^1(\mathbb{R}) \cap W^{2,\infty}_{\mathrm{loc}}(\mathbb{R})$ Unbounded	Exponential $(2, 1)$	$\varepsilon^{-d/(n-k)}$ for $k \leq 1$, and $\varepsilon^{-d/(n-2-\mu)}$ for $k = 2$
Softsign	$\dfrac{x}{1 + \lvert x \rvert}$	$C^1(\mathbb{R}) \cap W^{2,\infty}(\mathbb{R})$ Bounded	Polynomial $(2, 0)$	$\varepsilon^{-d/(n-k)}$ for $k = 0$
Inverse square root linear unit, $a > 0$	$\begin{cases} x, & x \geq 0 \\ \frac{x}{\sqrt{1+ax^2}}, & x < 0 \end{cases}$	$C^2(\mathbb{R}) \cap W^{3,\infty}_{\mathrm{loc}}(\mathbb{R})$ Unbounded	Polynomial $(3, 1)$	$\varepsilon^{-d/(n-k)}$ for $k \leq 1$
Inverse square root unit, $a > 0$	$\dfrac{x}{\sqrt{1 + ax^2}}$	Analytic Bounded	Polynomial $(j, 0)$ $\forall j \in \mathbb{N}_0$	$\varepsilon^{-d/(n-k)}$ for $k = 0$
Sigmoid/logistic	$\dfrac{1}{1 + e^{-x}}$	Analytic Bounded	Exponential $(j, 0)$ $\forall j \in \mathbb{N}_0$	$\varepsilon^{-d/n}$ for $k = 0$, and $\varepsilon^{-d/(n-k-\mu)}$ for $k \geq 1$
tanh	$\dfrac{e^x - e^{-x}}{e^x + e^{-x}}$	Analytic Bounded	Exponential $(j, 0)$ $\forall j \in \mathbb{N}_0$	$\varepsilon^{-d/n}$ for $k = 0$, and $\varepsilon^{-d/(n-k-\mu)}$ for $k \geq 1$
arctan	$\arctan(x)$	Analytic Bounded	Polynomial $(j, 0)$ $\forall j \in \mathbb{N}_0$	$\varepsilon^{-d/(n-k)}$ for $k = 0$
Softplus	$\ln(1 + e^x)$	Analytic Unbounded	Exponential $(j, 1)$ $\forall j \in \mathbb{N}_0$	$\varepsilon^{-d/n}$ for $k \leq 1$, and $\varepsilon^{-d/(n-k-\mu)}$ for $k \geq 2$
Swish	$\dfrac{x}{1 + e^{-x}}$	Analytic Unbounded	Exponential $(j, 1)$ $\forall j \in \mathbb{N}_0$	$\varepsilon^{-d/n}$ for $k \leq 1$, and $\varepsilon^{-d/(n-k-\mu)}$ for $k \geq 2$
Rectified power unit (RePU), $a \in \mathbb{N}_{\geq 2}$	$\max\{0, x\}^a$	$C^{a-1}(\mathbb{R}) \cap W^{a,\infty}_{\mathrm{loc}}(\mathbb{R})$ Unbounded	Exact (a, a)	$\varepsilon^{-d/(n-k)}$

4.2. Ingredient II: Approximation of polynomials

Later on, we approximate our target function by localized polynomials $\sum \phi_m \cdot \mathrm{poly}_m$, where the ϕ_m are the localizing functions from Section 4.1.[13] Afterwards, we emulate these localized polynomials by neural networks.[14] For this, we need to approximate polynomials in an efficient way. We start with approximating monomials $x \mapsto x^r$ on $\mathbb{R}$ by two-layered neural networks with activation functions that have a non-vanishing Taylor coefficient of order $r \in \mathbb{N}$. The construction is mainly based on a generalization of a standard approach for approximating the function $x \mapsto x^2$ by using finite differences. This has been studied in Rolnick and Tegmark (2018) and variations thereof have been considered, e.g., in Ohn and Kim (2019) and Schwab and Zech (2019).

Proposition 4.7. *Let $\varrho : \mathbb{R} \to \mathbb{R}$ be a function. Assume, that for some $n \in \mathbb{N}$ there exists $x_0 \in \mathbb{R}$ such that ϱ is $n+1$ times continuously differentiable in some open neighborhood U around x_0 and $\varrho^{(r)}(x_0) \neq 0$ for some $r \in \{1, \dots, n\}$. Then, for every $\varepsilon \in (0, 1)$, and every $B > 0$ there exists a constant $C = C(B, \varrho, r, n) > 0$ as well as a neural network Φ^r_ε with $R_\varrho(\Phi^r_\varepsilon)|_{[-B,B]} \in C^{n+1}([-B, B])$ and the following properties:*

(i) $\left\| R_\varrho(\Phi^r_\varepsilon) - x^r \right\|_{C^k([-B,B])} \leq \varepsilon$ for all $k = 0, \dots, n$;
(ii) $|R_\varrho(\Phi^r_\varepsilon)|_{W^{k,\infty}([-B,B])} \leq C \frac{r!}{(r-k)!} B^{r-k}$ for $k = 0, \dots, r$ and $|R_\varrho(\Phi^r_\varepsilon)|_{W^{k,\infty}([-B,B])} \leq \varepsilon$ for $k = r+1, \dots, n$;
(iii) $L\left(\Phi^r_\varepsilon\right) = 2$, as well as $M\left(\Phi^r_\varepsilon\right) \leq 3(r+1)$;
(iv) $\|\Phi^r_\varepsilon\|_{\max} \leq C\varepsilon^{-r}$.

Proof. The proof of this result can be found in Appendix C.1. □

Proposition 4.7 comes handy for two other usages besides monomial approximation:

- We construct neural networks which implement an *approximate multiplication* (see Corollary C.3) via the polarization identity

$$xy = \frac{1}{4}\left((x+y)^2 - (x-y)^2\right) \quad \text{for } x, y \in \mathbb{R}.$$

  This can by now be considered a standard approach in neural network approximation theory (originally used in Yarotsky (2017)). For this, the assumptions from Proposition 4.7 need to be fulfilled for $n = 2$ and $r = 2$, which holds true for all activation functions listed in Table 1 *except for the (leaky) ReLU.*[15] We use the approximate multiplication to obtain approximations of the multi-dimensional bumps ϕ_m from one-dimensional bumps which are in turn by construction neural networks. Furthermore, we can now deal with the multiplication of and ϕ_m with poly_m (see Corollary C.3 in Appendix C.1 and Lemma D.4 in Appendix D.3).
- It is often useful to pass output from a layer to a non neighboring layer deeper in the network. Previous works have solved this issue for the ReLU activation function by

[13] See Appendix D.2 for the precise statement and its proof.

[14] See Lemma D.5 in Appendix D.3 for the final statement and its proof.

[15] For these activation functions see Remark 4.10.

constructing an identity network (e.g., Gühring et al. (2020) and Petersen and Voigtländer (2018)). For general activation functions this is not possible. With help of Proposition 4.7 (for $n = 1$ and $r = 1$) an *approximate identity neural network* can be constructed (see Corollary C.4). It is clear that all activation functions listed in Table 1 fulfill the requirements.

4.3. Main results based on ingredients I & II

The proof of the main statement of this section can be roughly divided into two steps: In Proposition 4.8, the approximating neural networks are constructed with weights whose absolute values are bounded polynomially in ε^{-1}. In Theorem 4.9, the encodability of the weights is enforced. Before we state the actual results, we give an overview of the proof of Proposition 4.8, in which we explain the different approximation rates that can be obtained from different PUs. We hope that this overview will make it easier for the reader to keep track of the different approximation rates presented in the results of this section.

Overview of our proof strategy. *Let $\varepsilon > 0$. The proof of Proposition 4.8 is based on approximating a sum of $N^d = N(\varepsilon)^d$ localized Taylor polynomials (which are close to f) by a neural network $\Phi_{P,\varepsilon}$, such that we get*

$$\left\|f - R_\varrho(\Phi_{P,\varepsilon})\right\|_{W^{k,\infty}((0,1)^d)} \leq \underbrace{\left\|f - \sum_m \phi_m^s \mathrm{poly}_m\right\|_{W^{k,\infty}((0,1)^d)}}_{\textit{Step 1}} + \underbrace{\left\|\sum_m \phi_m^s \mathrm{poly}_m - R_\varrho(\Phi_{P,\varepsilon})\right\|_{W^{k,\infty}((0,1)^d)}}_{\textit{Step 2}}.$$

***Step 1**: We start by depicting how our PUs are used together with localized Taylor polynomials. In the process the interplay between the convergence speed of the PUs and the approximation rates that can be obtained becomes clear. When approximating a function f by localized Taylor polynomials poly_m, where the localization is realized by a PU from Section 4.1, we estimate the error on a fixed patch $\Omega_{\widetilde{m}}$ by*

$$\left\|f - \sum_m \phi_m^s \mathrm{poly}_m\right\|_{W^{k,\infty}(\Omega_{\widetilde{m}})} \leq \left\|\Big(\mathbb{1}_{(0,1)^d} - \sum_m \phi_m^s\Big) f\right\|_{W^{k,\infty}(\Omega_{\widetilde{m}})} + \left\|\sum_m \phi_m^s (f - \mathrm{poly}_m)\right\|_{W^{k,\infty}(\Omega_{\widetilde{m}})}.$$

The first term can be handled by Definition 4.1(iii) of the PU. Here, we only focus in detail on the second term. We have

$$\begin{aligned}&\left\|\sum_m \phi_m^s (f - \mathrm{poly}_m)\right\|_{W^{k,\infty}(\Omega_{\widetilde{m}})}\\ &\leq C \underbrace{\sum_{\|m-\widetilde{m}\|_\infty > 1} \|\phi_m^s\|_{W^{k,\infty}(\Omega_{\widetilde{m}})}}_{\text{(a)}}\\ &\quad + \underbrace{\sum_{\|m-\widetilde{m}\|_\infty \leq 1} \|\phi_m^s (f - \mathrm{poly}_{\widetilde{m}})\|_{W^{k,\infty}(\Omega_{\widetilde{m}})}}_{\text{(b)}}.\end{aligned}$$

In the cases of exponential/polynomial PUs, we will make use of the decay property of Definition 4.1(ii). In general we get

$$\begin{aligned}&\sum_{\|m-\widetilde{m}\|_\infty > 1} \|\phi_m^s\|_{W^{k,\infty}(\Omega_{\widetilde{m}})}\\ &\lesssim N^d \cdot \begin{cases} CN^k s^{\max\{0,k-\tau\}} e^{-Ds}, & \textit{if exponential PU},\\ CN^k s^{\max\{0,k-\tau\}} s^{-D}, & \textit{if polynomialPU},\\ 0, & \textit{if exact PU}.\end{cases}\end{aligned}$$

The closeness of the approximate bump to an exact bump is determined by the scaling factor s which we now couple with N.

- *For the* exponential case *we set $s := N^\mu$ for arbitrarily small $\mu > 0$ and can now use that the exponential term decays faster than any polynomial in N grows. In particular, we have*
  $$N^d N^k s^{\max\{0,k-\tau\}} e^{-Ds} = N^d N^k N^{\mu(k=2)} e^{-DN^\mu} \leq N^{-(n-k)}$$
  for N large enough.
- *In the* polynomial case *an arbitrarily small exponent is not sufficient to get rid of N^d, instead we must set $s := N^{\frac{d+k+(n-k)}{D}}$ and get*
  $$\begin{aligned}N^d N^k s^{\max\{0,k-\tau\}} s^{-D} &= N^d N^k N^{-d-k-(n-k)}\\ &= N^{-(n-k)}, \quad \textit{for} \quad k \leq \tau.\end{aligned}$$
  Here, we can only compensate the term N^d for $k \leq \tau$, since only the derivatives up to order τ absorb the effect of the scaling.
- *Finally, in case of an exact PU, term (a) is zero.*

For term (b) we only consider $m = \widetilde{m}$. For $k \geq \tau + 1$, we now pay the price for the scaling in the exponential case, since there is no exponential decay for the derivative of $\phi_{\widetilde{m}}^s$ on the patch $\Omega_{\widetilde{m}}$. From Definition 4.1(i) together with the Bramble–Hilbert Lemma B.4 we get the estimate

$$\begin{aligned}&\|\phi_m^s (f - \mathrm{poly}_{\widetilde{m}})\|_{W^{k,\infty}(\Omega_{\widetilde{m}})}\\ &\lesssim \begin{cases} N^{-(n-k-\mu(k=2))}, & \textit{if exponential PU},\\ N^{-(n-k)}, & \textit{for } k \leq \tau,\ \textit{if polynomial PU},\\ N^{-(n-k)}, & \textit{if exact PU}.\end{cases}\end{aligned}$$

Combining the computations for (a) and (b) we get the total estimate in Step 1

$$\begin{aligned}&\left\|\sum_m f - \phi_m^s \mathrm{poly}_m\right\|_{W^{k,\infty}(\Omega_{\widetilde{m}})}\\ &\lesssim \begin{cases} N^{-(n-k-\mu(k=2))}, & \textit{if exponential PU},\\ N^{-(n-k)}, & \textit{for } k \leq \tau,\ \textit{if polynomialPU},\\ N^{-(n-k)}, & \textit{if exact PU}.\end{cases}\end{aligned}$$

By choosing $N := \lceil \varepsilon^{-1/(n-k-\mu(k=2))} \rceil$ in the exponential case and $N := \lceil \varepsilon^{-1/(n-k)} \rceil$ in the other two cases, we get that the term from Step 1 can be bounded by ε.

***Step 2:** To construct the neural network we use the results from Section 4.2 to*

(i) approximate Taylor polynomials by neural networks;
(ii) approximate the multi-dimensional PU. Since only the d factors of its tensor structure can be exactly represented by a neural network (see Definition 4.1(iv)), their multiplication must be approximated;
(iii) approximate the multiplication of (i) with (ii) by neural networks $\Phi_{m,\widetilde{\varepsilon}}$ with accuracy $\widetilde{\varepsilon}$ (chosen below);
(iv) build the sum of all approximations of localized Taylor polynomials by neural networks.

The network $\Phi_{P,\varepsilon}$ thus consists of the subnetworks from step (iii). We get the estimate

$$\left\|\sum_m \phi_m^s \mathrm{poly}_m - R_\varrho(\Phi_{P,\varepsilon})\right\|_{W^{k,\infty}((0,1)^d)}$$

$$\le \sum_m \|\phi_m^s \mathrm{poly}_m - \Phi_{m,\widetilde{\varepsilon}}\|_{W^{k,\infty}((0,1)^d)} \lesssim N^d \widetilde{\varepsilon}.$$

Consequently, we need to chose $\widetilde{\varepsilon} := \varepsilon N^{-d} \approx \varepsilon^{-d/(n-k-\mu(k=2))+1}$ (some terms are suppressed here for simplicity of exposition). We can only do this, since neither the number of weights of $\Phi_{m,\widetilde{\varepsilon}}$ nor its number of layers depends on $\widetilde{\varepsilon}$ (only the values of the weights do). In other words, each $\Phi_{m,\widetilde{\varepsilon}}$ has a constant number of weights and layers. Combining $\sim N^d$ of such networks to get $\Phi_{P,\varepsilon}$ yields a network with about $N^d = \varepsilon^{-d/(n-k-\mu(k=2))}$ weights and constant number of layers for the exponential case (with obvious adaptations for the other two cases).

***Conclusion:** For activation functions ϱ with an exponential PU, we obtain optimal rates for Sobolev norms $k \le \tau$ and almost optimal rates for $k \ge \tau + 1$; in the polynomial case, we get optimal approximation rates only in $W^{k,p}$-norms if $k \le \tau$; in the case of an exact PU, we get optimal approximation rates for Sobolev norms up to order j (smoothness of ϱ).*

We now give the statement of Proposition 4.8, which can be proven by using the ideas and concepts presented so far in this section. The detailed proofs are executed in Appendices D.1–D.4, mostly for the case of *exponential* (j, τ)-PUs. The statements for the other two cases can be proven in an analogous way.

Proposition 4.8. *We make the following assumptions:*

- *Let $d \in \mathbb{N}$, $j, \tau \in \mathbb{N}_0$, $k \in \{0, \dots, j\}$, $n \in \mathbb{N}_{\ge k+1}$, $1 \le p \le \infty$, and $\mu > 0$;*
- *let $\varrho : \mathbb{R} \to \mathbb{R}$ such that $\left(\Psi^{(j,\tau,N,s)}(\varrho)\right)_{N\in\mathbb{N},s\ge 1}$ is an exponential (polynomial, exact) (j, τ)-PU;*
- *there exists $x_0 \in \mathbb{R}$ such that ϱ is three times continuously differentiable in a neighborhood of x_0 and $\varrho''(x_0) \neq 0$.*

Then, there exist constants $L, C, \theta, \widetilde{\varepsilon}$ depending on d, n, p, k, μ with the following properties:

For every $\varepsilon \in (0, \widetilde{\varepsilon})$ and every $f \in \mathcal{F}_{n,d,p}$, there is a neural network $\Phi_{\varepsilon,f}$ with d-dimensional input and one-dimensional output, at most L layers and at most

$$\begin{cases} C\varepsilon^{-d/(n-k-\mu(k=2))}, & \text{if exponential PU ,} \\ C\varepsilon^{-d/(n-k)}, & \text{for } k \le \tau, \text{ if polynomial PU ,} \\ C\varepsilon^{-d/(n-k)}, & \text{if exact PU ,} \end{cases}$$

nonzero weights bounded in absolute value by $C\varepsilon^{-\theta}$ such that

$$\|R_\varrho(\Phi_{\varepsilon,f}) - f\|_{W^{k,p}((0,1)^d)} \le \varepsilon.$$

The main theorem now states that Proposition 4.8 also holds with encodable weights, i.e. for each $\varepsilon > 0$, every element of the set of weights $W_\varepsilon = \bigcup_f W_{\varepsilon,f}$ (where $W_{\varepsilon,f}$ denotes the weights of $\Phi_{\varepsilon,f}$) can be uniquely encoded by $\lceil C \log_2(1/\varepsilon) \rceil$ bits. To state this in a formal way, we use the notation introduced in Eq. (2.1).

Theorem 4.9. *We make the following assumptions:*

- *Let $d \in \mathbb{N}$, $j, \tau \in \mathbb{N}_0$, $k \in \{0, \dots, j\}$, $n \in \mathbb{N}_{\ge k+1}$, $1 \le p \le \infty$, and $\mu > 0$;*
- *let $\varrho : \mathbb{R} \to \mathbb{R}$ such that $\left(\Psi^{(j,\tau,N,s)}(\varrho)\right)_{N\in\mathbb{N},s\ge 1}$ is an exponential (polynomial, exact) (j, τ)-PU;*
- *there exists $x_0 \in \mathbb{R}$ such that ϱ is three times continuously differentiable in a neighborhood of x_0 and $\varrho''(x_0) \neq 0$.*

Then, there exist constants L, C and $\widetilde{\varepsilon}$, and a coding scheme $\mathcal{B} = (B_\ell)_{\ell\in\mathbb{N}}$ depending on d, n, p, k, μ with the following properties:

For every $\varepsilon \in (0, \widetilde{\varepsilon})$ and every $f \in \mathcal{F}_{n,d,p}$, there is a neural network $\Phi_{\varepsilon,f} \in \mathcal{NN}^{\mathcal{B}}_{M_\varepsilon, \lceil C\log_2(1/\varepsilon)\rceil, d}$ with d-dimensional input, one-dimensional output, at most L layers and at most

$$M_\varepsilon = \begin{cases} C\varepsilon^{-d/(n-k-\mu(k=2))}, & \text{if exponential PU ,} \\ C\varepsilon^{-d/(n-k)}, & \text{for } k \le \tau, \text{ if polynomial PU ,} \\ C\varepsilon^{-d/(n-k)}, & \text{if exact PU ,} \end{cases}$$

nonzero weights, such that

$$\|R_\varrho(\Phi_{\varepsilon,f}) - f\|_{W^{k,p}((0,1)^d)} \le \varepsilon.$$

Proof. We give a short outline of the proof here, the details can be found in Appendix E. Let $\Phi_{\varepsilon,f} = ((A_1, b_1), \dots, (A_{L-1}, b_{L-1}), (A_L, b_L))$ be the network from Proposition 4.8 (where the main work has already been done). From the proof of the proposition (see Eq. (D.18)) it follows that $A_L = A_f \widetilde{A}_L$ and $b_L = A_f \widetilde{b}_L$ where the entries of the block diagonal matrix A_f depend on f and the entries of $A_1, b_1, \dots, A_{L-1}, b_{L-1}, \widetilde{A}_L, \widetilde{b}_L$ are independent from f (i.e., they only depend on $\varepsilon, n, d, p, k, \mu$). We denote the collection of nonzero entries of $A_1, b_1, \dots, A_{L-1}, b_{L-1}, \widetilde{A}_L, \widetilde{b}_L$ by W_ε.

- The number of independent weights $|W_\varepsilon|$ is bounded by $C \cdot \varepsilon^{-d/(n-k-\mu(k=2))}$ since the total number of nonzero weights is bounded by this quantity.
- We round the entries of A_f, b_f with a suitable precision ν to the mesh $[-\varepsilon^{-\theta}, \varepsilon^{-\theta}] \cap \varepsilon^\nu \mathbb{Z}$, where we also use the fact that the weights of $\Phi_{\varepsilon,f}$ are bounded in absolute value by $C\varepsilon^{-\theta}$.
- The nonzero entries of A_L in the last layer of $\Phi_{\varepsilon,f}$ are in the set $G_{\mathrm{mult}} := \{x_1 x_2 : x_1 \in W_\varepsilon, x_2 \in [-\varepsilon^{-\theta}, \varepsilon^{-\theta}] \cap \varepsilon^\nu \mathbb{Z}\}$ with cardinality bounded by $\varepsilon^{-\widetilde{s}}$ (similar for b_L).

Hence, the weights of the approximating neural networks can be chosen from a set $\widetilde{W}_\varepsilon$ with less than ε^{-s} real numbers, where $s > 0$ only depends on d, n, p, k, μ and not on f. Consequently, there exists a surjective mapping $B_\varepsilon : \{0, 1\}^{\lceil s \log_2(1/\varepsilon) \rceil} \to W_\varepsilon$. The collection of these maps constitutes the coding scheme. □

Remark 4.10 (*Plug & Play*). Some well-known activation functions, e.g., the (leaky) ReLU, do not fulfill all assumptions stated in Proposition 4.8 and Theorem 4.9 (ϱ should be three times continuously differentiable in a neighborhood of some $x_0 \in \mathbb{R}$ with $\varrho''(x_0) \neq 0$). However, we note that our proof strategy only requires the approximation of monomials and an approximate multiplication. In case of the (leaky) ReLU this can be done with $\mathcal{O}(\log(1/\varepsilon))$ weights and layers (see Yarotsky (2017, Proposition 2 and 3)). Generally speaking: As long as an activation allows for

- the construction of an (approximate) PU along the lines of Definition 4.1,
- an efficient approximation of polynomials and the identity function,

our proof strategy can be employed to yield efficient convergence rates. As such, our framework is very general and unifies several previous approaches (e.g. Gühring et al. (2020) and Yarotsky (2017)) as well as extends the previously known rates to a very general class of activation functions and rather general smoothness norms.

Remark 4.11 (*Tightness of the Bounds*). From Corollary 3.8(ii) it follows that our bounds for encodable neural network weights are tight up to a log factor for $k \le \tau$. For $k \ge \tau + 1$, they are (up to a log factor) tight in the case of exact PUs and we get arbitrarily close to the optimal bound (again up to a log factor) in case of exponential PUs. If we allow for *arbitrary* weights, then this upper bound might be drastically improved (see Remark 3.7).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank Philipp Petersen for fruitful discussions on the topic. Moreover, they would like to thank the anonymous reviewers for suggestions to improve the manuscript. I. Gühring acknowledges support from the Research Training Group "Differential Equation- and Data-driven Models in Life Sciences and Fluid Dynamics: An Interdisciplinary Research Training Group (DAEDALUS)" (GRK 2433) funded by the German Research Foundation (DFG).

Appendix A. Notation and auxiliary results

In this subsection, we depict the (mostly standard) notation used throughout this paper. We set $\mathbb{N} := \{1, 2, \dots\}$ and $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. For $k \in \mathbb{N}_0$ we define $\mathbb{N}_{\geq k} := \{k, k+1, \dots\}$. For a set A we denote its cardinality by $|A| \in \mathbb{N} \cup \{\infty\}$ and by $\mathbb{1}_A$ its indicator function of A. If $x \in \mathbb{R}$, then we write $\lceil x \rceil := \min\{k \in \mathbb{Z} : k \geq x\}$ where $\mathbb{Z}$ is the set of integers and $\lfloor x \rfloor := \max\{k \in \mathbb{Z} : k \leq x\}$.

If $d \in \mathbb{N}$ and $\| \cdot \|$ is a norm on $\mathbb{R}^d$, then we denote for $x \in \mathbb{R}^d$ and $r > 0$ by $B_{r,\|\cdot\|}(x)$ the open ball around x in $\mathbb{R}^d$ with radius r, where the distance is measured in $\|\cdot\|$. By $|x|$ we denote the euclidean norm of x and by $\|x\|_\infty$ the maximum norm. We endow $\mathbb{R}^d$ with the standard topology and for $A \subset \mathbb{R}^d$ we denote by $\overline{A}$ the closure of A.

For $d_1, d_2 \in \mathbb{N}$ and a matrix $A \in \mathbb{R}^{d_1,d_2}$ the number of nonzero entries of A is counted by $\|\cdot\|_0$, i.e.

$$\|A\|_0 := |\{(i,j) : A_{i,j} \neq 0\}|.$$

If $f : X \to Y$ and $g : Y \to Z$ are two functions, then we write $g \circ f : X \to Z$ for their composition. If additionally $U \subset X$, then $f|_U : U \to Y$ denotes the restriction of f onto U. We use the usual *multiindex* notation, i.e. for $\alpha \in \mathbb{N}_0^d$ we write $|\alpha| := \alpha_1 + \cdots + \alpha_d$ and $\alpha! := \alpha_1! \cdot \ldots \cdot \alpha_d!$. Moreover, if $x \in \mathbb{R}^d$, then we have

$$x^\alpha := \prod_{i=1}^d x_i^{\alpha_i}.$$

Let from now on $\Omega \subset \mathbb{R}^d$ be open. For a function $f : \Omega \to \mathbb{R}$, we denote by

$$D^\alpha f := \frac{\partial^{|\alpha|} f}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \cdots \partial x_d^{\alpha_d}}.$$

its (weak or classical) derivative of order α. For $n \in \mathbb{N}_0 \cup \{\infty\}$, we denote by $C^n(\Omega)$ the set of n times continuously differentiable functions on Ω. Additionally, if $\overline{\Omega}$ is compact, we set, for $f \in C^n(\Omega)$

$$\|f\|_{C^n(\overline{\Omega})} := \max_{0 \leq |\alpha| \leq n} \sup_{x \in \Omega} |D^\alpha f(x)|.$$

We denote by $L^p(\Omega)$, $1 \leq p \leq \infty$ the standard Lebesgue spaces.

In the following, we will also make use of the following well-known fact stating that the exponential function decays faster than any polynomial.

Proposition A.1. *Let $\alpha, \beta, c, c' > 0$. Then*

$$\lim_{x \to \infty} \frac{c' x^\alpha}{e^{c \cdot x^\beta}} = 0.$$

This implies that for all $\gamma > 0$ there exists some constant $C = C(\alpha, \beta, \gamma) > 0$ such that for all $x > 0$ there holds

$$\frac{c' x^\alpha}{e^{c \cdot x^\beta}} \leq C x^{-\gamma}.$$

Appendix B. Sobolev spaces

In this section, we introduce Sobolev spaces (see Adams (1975)) which constitute a crucial concept within the theory of PDEs (see e.g. Evans (1999) and Roubíček (2013)).

Definition B.1. Given some domain $\Omega \subset \mathbb{R}^d$, $1 \leq p < \infty$, and $n \in \mathbb{N}$, the Sobolev space $W^{n,p}(\Omega)$ is defined as

$$W^{n,p}(\Omega) := \Big\{ f : \Omega \to \mathbb{R} : \|D^\alpha f\|^p_{L^p(\Omega)} < \infty, \quad \text{for all } \alpha \in \mathbb{N}_0^d \text{ with } |\alpha| \leq n \Big\},$$

and is equipped with the norm

$$\|f\|_{W^{n,p}(\Omega)} := \left(\sum_{0 \leq |\alpha| \leq n} \|D^\alpha f\|^p_{L^p(\Omega)} \right)^{1/p}.$$

Additionally, we set

$$W^{n,\infty}(\Omega) := \Big\{ f : \Omega \to \mathbb{R} : \|D^\alpha f\|_{L^\infty(\Omega)} < \infty, \quad \text{for all } \alpha \in \mathbb{N}_0^d \text{ with } |\alpha| \leq n \Big\},$$

and we equip this space with the norm $\|f\|_{W^{n,\infty}(\Omega)} := \max_{|\alpha| \leq n} \|D^\alpha f\|_{L^\infty(\Omega)}$. Moreover, for $0 \leq k \leq n$, on $W^{n,p}(\Omega)$ we introduce the family of *semi-norms*

$$|f|_{W^{k,p}(\Omega)} := \left(\sum_{|\alpha| = k} \|D^\alpha f\|^p_{L^p(\Omega)} \right)^{1/p},$$

$$|f|_{W^{k,\infty}(\Omega)} := \max_{|\alpha| = k} \|D^\alpha f\|_{L^\infty(\Omega)},$$

respectively. Finally, let $W^{n,p}_{\mathrm{loc}}(\Omega) := \{ f : \Omega \to \mathbb{R} : f|_{\widetilde{\Omega}} \in W^{n,p}(\widetilde{\Omega})$ for all compact $\widetilde{\Omega} \subset \Omega \}$.

Remark B.2. If Ω is bounded and fulfills a local Lipschitz condition, arguments from Adams (1975) show that $W^{2,\infty}(\Omega)$ can be continuously embedded into $C^1(\overline{\Omega})$. This can be seen as follows: Adams (1975, Theorem 4.12) shows that $W^{2,p}(\Omega)$ can be continuously embedded into $C^1(\overline{\Omega})$ for $p > d$. Since also $W^{2,\infty}(\Omega)$ can be continuously embedded into $W^{2,p}(\Omega)$, the claim follows.

Remark B.3. For purely technical reasons we sometimes make use of an extension operator. For this, let $E : W^{n,p}((0,1)^d) \to W^{n,p}(\mathbb{R}^d)$ be the extension operator from Stein (1979, Theorem VI.3.1.5) and set $\widetilde{f} := Ef$. Note that for arbitrary $\Omega \subset \mathbb{R}^d$ and $0 \leq k \leq n$ it holds

$$|\widetilde{f}|_{W^{k,p}(\Omega)} \leq \|\widetilde{f}\|_{W^{n,p}(\mathbb{R}^d)} \leq C \|f\|_{W^{n,p}((0,1)^d)}, \tag{B.1}$$

where $C = C(n,p,d)$ is the norm of the extension operator.

The following lemma which will be crucial for the proofs of our results can be stated in much more generality (see Brenner and Scott (2008, Chapter 4.1)) and relies on the use of averaged Taylor polynomials. We only state a version tailored to our specific needs and will not give a proof since the details of this specific version have been worked out in Gühring et al. (2020, Section B.3 and Lemma C.4).

Lemma B.4 (*Bramble–Hilbert*)**.** *Let $d, n \in \mathbb{N}$ and $1 \leq p \leq \infty$. Furthermore, let $N \in \mathbb{N}$ and set for $m \in \{0, \dots, N\}^d$*

$$\Omega_{m,N} := B_{\frac{1}{N}, \|\cdot\|_\infty}\left(\frac{m}{N}\right).$$

Then there exists a constant $C = C(n,d) > 0$ such that for all $f \in W^{n,p}(\mathbb{R}^d)$ and $m \in \{0, \dots, N\}^d$ there is a polynomial $p_m(x) =$

$\sum_{|\alpha|\le n-1} c_\alpha x^\alpha$ such that

$$\|f-p_m\|_{W^{k,p}(\Omega_{m,N})} \le C\left(\frac{1}{N}\right)^{n-k}\|f\|_{W^{n,p}(\Omega_{m,N})},\quad \text{for } k=0,1,\dots,n$$

and the coefficients c_α are bounded by $|c_\alpha| \le CN^{d/p}\|f\|_{W^{n,p}(\Omega_{m,N})}$ for all α with $|\alpha| \le n-1$.

Now we turn our attention to a version of a product rule tailored to our needs.

Lemma B.5. *Let $k \in \mathbb{N}$, and assume that $f \in W^{k,\infty}(\Omega)$ and $g \in W^{k,p}(\Omega)$ with $1 \le p \le \infty$. If $k \ge 3$, additionally assume that $f \in C^k(\Omega)$ or $g \in C^k(\Omega)$. Then $fg \in W^{k,p}(\Omega)$ and there exists a constant $C = C(d,p,k) > 0$ such that*

$$\|fg\|_{W^{k,p}(\Omega)} \le C\sum_{i=0}^{k}\|f\|_{W^{i,\infty}(\Omega)}\|g\|_{W^{k-i,p}(\Omega)},$$

and, consequently

$$\|fg\|_{W^{k,p}(\Omega)} \le C\|f\|_{W^{k,\infty}(\Omega)}\|g\|_{W^{k,p}(\Omega)}.$$

Proof. For $k=0$ the statement is obvious.

For $k=1$ we get from Gühring et al. (2020, Lemma B.6) that there exists a constant $C = C(d,p) > 0$ such that

$$|fg|_{W^{1,p}(\Omega)} \le C\left(\|f\|_{W^{1,\infty}(\Omega)}\|g\|_{L^p(\Omega)} + \|f\|_{L^\infty(\Omega)}\|g\|_{W^{1,p}(\Omega)}\right),$$

from which the statement can easily be deduced.

For $k=2$ it follows from Gilbarg and Trudinger (1998, Chap. 7.3) that the usual product rule also holds for the second order derivatives such that we have

$$\begin{aligned}|fg|_{W^{2,p}(\Omega)} \le C \sum_{i,j=1,\dots,d} &\left\|\frac{\partial^2}{\partial x_i\partial x_j}fg\right\|_{L^p(\Omega)} + \left\|\frac{\partial}{\partial x_i}f\frac{\partial}{\partial x_j}g\right\|_{L^p(\Omega)}\\ &+ \left\|\frac{\partial}{\partial x_j}f\frac{\partial}{\partial x_i}g\right\|_{L^p(\Omega)} + \left\|f\frac{\partial^2}{\partial x_i\partial x_j}g\right\|_{L^p(\Omega)}\\ \le C\Big(&\|f\|_{W^{2,\infty}(\Omega)}\|g\|_{L^p(\Omega)} + \|f\|_{W^{1,\infty}(\Omega)}\|g\|_{W^{1,p}(\Omega)}\\ &+\|f\|_{L^\infty(\Omega)}\|g\|_{W^{2,p}(\Omega)}\Big).\end{aligned}$$

Again the overall statement follows easily. The statement for $k \in \mathbb{N}_{\ge 3}$ can directly be concluded from the Leibniz formula (see Bressan (2012, Lemma 8.18)), which, for a multi-index α with $|\alpha| \le k$ yields

$$D^\alpha(fg) = \sum_{|\beta|\le|\alpha|}\binom{\alpha}{\beta}D^\beta f D^{\alpha-\beta}g. \quad \square$$

The following corollary establishes a chain rule estimate for $W^{k,\infty}$.

Corollary B.6. *Let $d, m \in \mathbb{N}$, $k \in \mathbb{N}_{\ge 2}$ and $\Omega_1 \subset \mathbb{R}^d$, $\Omega_2 \subset \mathbb{R}^m$ both be open, bounded, and convex. Then, there is a constant $C = C(d,m,k) > 0$ with the following properties:*

(i) If $k = 2$ and $f \in W^{2,\infty}(\Omega_1;\ \mathbb{R}^m) \cap C^1(\Omega_1;\mathbb{R}^m)$ and $g \in W^{2,\infty}(\Omega_2) \cap C^1(\Omega_2)$ such that $\mathrm{Range}(f) \subset \Omega_2$, then for the composition $g \circ f$ it holds that $g \circ f \in W^{2,\infty}(\Omega_1) \cap C^1(\Omega_1)$ and we have

$$|g\circ f|_{W^{1,\infty}(\Omega_1)} \le C|g|_{W^{1,\infty}(\Omega_2)}|f|_{W^{1,\infty}(\Omega_1;\,\mathbb{R}^m)},$$

and

$$|g\circ f|_{W^{2,\infty}(\Omega_1)} \le C\Big(|g|_{W^{2,\infty}(\Omega_2)}|f|^2_{W^{1,\infty}(\Omega_1;\,\mathbb{R}^m)} + |g|_{W^{1,\infty}(\Omega_2)}|f|_{W^{2,\infty}(\Omega_1;\,\mathbb{R}^m)}\Big).$$

(ii) If $k \ge 3$, $f \in C^k(\overline{\Omega}_1;\mathbb{R}^m)$ and $g \in C^k(\overline{\Omega}_2)$ such that $\mathrm{Range}(f) \subset \Omega_2$, then for the composition $g\circ f$ it holds that $g \circ f \in C^k(\Omega_1)$ and

(a) if $|f|_{W^{l,\infty}(\Omega_1;\,\mathbb{R}^m)} \le CN^l$ for all $l = 1,\dots,k$, then

$$|g\circ f|_{W^{k,\infty}(\Omega_1)} \le C\sum_{l=1}^{k}|g|_{W^{l,\infty}(\Omega_2)}N^k; \tag{B.2}$$

(b) if $\tau \in \mathbb{N}_0$ and $|f|_{W^{l,\infty}(\Omega_1;\,\mathbb{R}^m)} \le CN^{l+\mu\max\{0,l-\tau\}}$ for all $l = 1,\dots,k$, then

$$|g\circ f|_{W^{k,\infty}(\Omega_1)} \le C\sum_{l=1}^{k}|g|_{W^{l,\infty}(\Omega_2)}N^{k+\mu(k=2)}. \tag{B.3}$$

Proof. (i) can be shown by basic computations using the classical first derivative and Gühring et al. (2020, Corollary B.5, Lemma B.6). For (ii), we make use of the multivariate Faa Di Bruno formula (see Constantine and Savits (1996, Theorem 2.1)) and get that

$$|g\circ f|_{W^{k,\infty}(\Omega_1)} \le C\max_{|\nu|=k}\sum_{l=1}^{k}|g|_{W^{l,\infty}(\Omega_2)}\sum_{|\lambda|=l}\sum_{p(\nu,\lambda)}\prod_{j=1}^{k}|f|^{|r_j|}_{W^{|l_j|,\infty}(\Omega_1;\,\mathbb{R}^m)},$$

where

$$p(\nu,\lambda) := \left\{\begin{array}{c}(r_1,\dots,r_k;\, l_1,\dots,l_k):\\ \text{for some } 1\le s\le k,\ r_i = 0\\ \text{and } l_i = 0 \text{ for } 1\le i\le k-s;\\ |r_i| > 0 \text{ for } k-s+1\le i\le k;\\ \text{and } 0\le l_{k-s+1}\le\cdots\le l_k \text{ are such that}\\ \sum_{i=1}^k r_i = \lambda,\ \sum_{i=1}^k |r_i| l_i = \nu.\end{array}\right\}.$$

Eq. (B.2) now follows from $\prod_{j=1}^k |f|^{|r_j|}_{W^{|l_j|,\infty}(\Omega_1;\,\mathbb{R}^m)} \le C\prod_{j=1}^k N^{|l_j||r_j|} = CN^{\sum_{j=1}^k |l_j||r_j|} = CN^k$. Eq. (B.3) for $\tau = 0$ follows from (a) with $N = N^{1+\mu}$. For $\tau \ge 1$, we have

$$\prod_{j=1}^{k} N^{\mu\max\{0,|l_j|-\tau\}|r_j|} = N^{\mu\sum_{j=1}^k \max\{0,|l_j|-\tau\}|r_j|}$$

and

$$\sum_{j=1}^{k}\max\{0,|l_j|-\tau\}|r_j| = \sum_{j:|l_j|\ge\tau}^{k}(|l_j|-\tau)|r_j|.$$

If $|l_j| < \tau$ for all $j = 1,\dots,k$, then $\sum_{j:|l_j|\ge\tau}^k(|l_j|-\tau)|r_j| = 0 \le \mu\max\{0,k-\tau\}$. If there exists some j' with $|l_{j'}| \ge \tau$ and $|r_j| = 0$ for all j with $|l_j| \ge \tau$, then also $\sum_{j:|l_j|\ge\tau}^k(|l_j|-\tau)|r_j| = 0 \le \mu\max\{0,k-\tau\}$. Otherwise, there exists some j' with $|l_{j'}| \ge \tau$ and $|r_{j'}| \ge 1$. We then have

$$\begin{aligned}\sum_{j:|l_j|\ge\tau}^{k}(|l_j|-\tau)|r_j| &\le \sum_{j:|l_j|\ge 1}^{k}|l_j||r_j| - \tau\sum_{j:|l_j|\ge\tau}|r_j| = k - \tau\sum_{j:|l_j|\ge\tau}|r_j|\\ &\le k - \tau|l_{j'}|\,|r_{j'}| \le k-\tau\end{aligned}$$

from which the statement in combination with (a) follows. $\square$

Appendix C. Neural network calculus

In this section, we introduce several operations one can perform with neural networks, namely the *concatenation* and the *parallelization* of neural networks. Moreover, Appendix C.1 is devoted to approximations of polynomials. We give the proof of Proposition 4.7 (approximation of monomials by neural networks) and show how to derive approximations of the identity function as well as of approximate multiplications.

We first consider the *concatenation* of two neural networks as given in Petersen and Voigtländer (2018).

Definition C.1. Let $\Phi^1 = \left((A_1^1, b_1^1), \dots, (A_{L_1}^1, b_{L_1}^1)\right)$ and $\Phi^2 = \left((A_1^1, b_1^1), \dots, (A_{L_1}^1, b_{L_1}^1)\right)$ be two neural networks such that the input dimension of Φ^1 is equal to the output dimension of Φ^2. Then the *concatenation* of Φ^1, Φ^2 is defined as the L_1+L_2-1-layer neural network

$$\Phi^1 \bullet \Phi^2 := \left((A_1^2, b_1^2), \dots, (A_{L_2-1}^2, b_{L_2-1}^2), (A_1^1 A_{L_2}^2, A_1^1 b_{L_2}^2 + b_1^1), \right. \\ \left. (A_2^1, b_2^1), \dots, (A_{L_1}^1, b_{L_1}^1)\right).$$

It is easy to see that $R_\varrho(\Phi^1 \bullet \Phi^2) = R_\varrho(\Phi^1) \circ R_\varrho(\Phi^2)$.

Now, we introduce the *parallelization* of neural networks with the same number of layers, inspired by the construction in Petersen and Voigtländer (2018).

Lemma C.2. *Let $\varrho : \mathbb{R} \to \mathbb{R}$. Additionally, let $\Phi^1, \dots \Phi^n$ be neural networks with d-dimensional input and $L \in \mathbb{N}$ layers, respectively. Then, there exists a neural network $\mathrm{P}(\Phi^1, \dots, \Phi^n)$ with d-dimensional input and*

(i) There holds $R_\varrho\left(\mathrm{P}(\Phi^1, \dots, \Phi^n)\right)(x) = \left(R_\varrho(\Phi^1)(x), \dots, R_\varrho(\Phi^n)(x)\right)$ for all $x \in \mathbb{R}^d$;
(ii) L layers;
(iii) $M\left(\mathrm{P}(\Phi^1, \dots, \Phi^n)\right) = \sum_{i=1}^n M(\Phi^i)$;
(iv) $\left\|\mathrm{P}\left(\Phi^1, \dots, \Phi^n\right)\right\|_{\max} = \max\left\{\left\|\Phi^1\right\|_{\max}, \dots, \|\Phi^n\|_{\max}\right\}$.

Proof. The neural network

$$\mathrm{P}(\Phi^1, \dots, \Phi^n) := \left((\widetilde{A}_1, \widetilde{b}_1), \dots, (\widetilde{A}_L, \widetilde{b}_L)\right),$$

with

$$\widetilde{A}_1 := \begin{pmatrix} A_1^1 \\ \vdots \\ A_1^n \end{pmatrix}, \quad \widetilde{b}_1 := \begin{pmatrix} b_1^1 \\ \vdots \\ b_1^n \end{pmatrix} \quad \text{and}$$

$$\widetilde{A}_\ell := \begin{pmatrix} A_\ell^1 & & & \\ & A_\ell^2 & & \\ & & \ddots & \\ & & & A_L^n \end{pmatrix}, \quad \widetilde{b}_\ell := \begin{pmatrix} b_\ell^1 \\ \vdots \\ b_\ell^n \end{pmatrix}, \quad \text{for } 1 < \ell \le L,$$

fulfills all the desired properties. □

C.1. Approximate monomials and multiplication

We first give the proof of Proposition 4.7:

Proof of Proposition 4.7. Choose $C_0 > 1$ so that $[x_0 - \frac{nB}{C_0}, x_0 + \frac{nB}{C_0}] \subset U$. Moreover, let $\delta \ge C_0$ be arbitrary. Define the function

$$\varrho_\delta^r : \mathbb{R} \to \mathbb{R}, \quad x \mapsto \frac{\delta^r}{\varrho^{(m)}(x_0)} \sum_{j=0}^r (-1)^j \binom{r}{j} \cdot \varrho\left(x_0 - j\frac{x}{\delta}\right).$$

Then $\varrho_\delta^r|_{[-B,B]} \in C^{n+1}([-B, B])$. Using the Taylor expansion and the following identity from (Katsuura, 2009)

$$\sum_{j=1}^r (-1)^j \binom{r}{j} j^k = \begin{cases} 0, & \text{if } 1 \le k < r, \\ (-1)^r r!, & \text{if } k = r, \end{cases} \tag{C.1}$$

it can easily be shown that $\varrho_\delta^r(x) \approx x^r$ for $\delta > 0$ sufficiently large. In detail, we have by Taylor's Theorem (where ξ_j is between x_0 and $x_0 - j\frac{x}{\delta}$ for $j = 1, \dots, r$) that

$$\sum_{j=0}^r (-1)^j \binom{r}{j} \cdot \varrho\left(x_0 - j\frac{x}{\delta}\right)$$

$$= \varrho(x_0) + \sum_{j=1}^r (-1)^j \binom{r}{j} \cdot \left(\sum_{k=0}^r \frac{\varrho^{(k)}(x_0)}{k!} \left(\frac{-jx}{\delta}\right)^k \right.$$
$$\left. + \frac{\varrho^{(r+1)}(\xi_j)}{(r+1)!} \left(\frac{-(r+1)x}{\delta}\right)^{r+1}\right)$$
$$= \varrho(x_0) + \sum_{k=0}^r \left(\frac{-x}{\delta}\right)^k \frac{\varrho^{(k)}(x_0)}{k!} \sum_{j=1}^r (-1)^j \binom{r}{j} j^k$$
$$+ \underbrace{\sum_{j=1}^r (-1)^j \binom{r}{j} \frac{\varrho^{(r+1)}(\xi_j)}{(r+1)!} \left(\frac{-(r+1)x}{\delta}\right)^{r+1}}_{=: r_\delta^r(x)}$$
$$= \varrho(x_0) \underbrace{\sum_{j=0}^r (-1)^j \binom{r}{j}}_{=0} + \sum_{k=1}^r \left(\frac{-x}{\delta}\right)^k \frac{\varrho^{(k)}(x_0)}{k!} \underbrace{\sum_{j=1}^r (-1)^j \binom{r}{j} j^k}_{\text{use Eq. (C.1)}}$$
$$+ r_\delta^r(x)$$
$$= \left(\frac{x}{\delta}\right)^r \varrho^{(r)}(x_0) + r_\delta^r(x).$$

Hence, for every $k = 0, \dots, n$ and every $x \in [-B, B]$, we have

$$\left|(\varrho_\delta^r)^{(k)}(x) - (x^r)^{(k)}\right|$$
$$= \left|\frac{\delta^r}{\varrho^{(r)}(x_0)} (r_\delta^r)^{(k)}(x)\right|$$
$$\le \underbrace{\sum_{j=1}^r \binom{r}{j} \left|\frac{\varrho^{(r+1)}(\xi_j)}{(r+1)!}\right|}_{\le 2^n \|\varrho\|_{C^{n+1}(U)}} \cdot \underbrace{\left|\frac{\delta^r}{\varrho^{(r)}(x_0)} \left(\frac{-(r+1)}{\delta}\right)^{r+1}\right|}_{\le \frac{(n+1)^{n+1}}{\delta \min_{i=0,\dots,n} |\varrho^{(i)}(x_0)|}} \cdot \underbrace{\left|(x^{r+1})^{(k)}\right|}_{\le n! \max\{B,1\}^{n+1}}$$
$$\le 2^n \cdot (n+1)^{n+1} n! \cdot \frac{\|\varrho\|_{C^{n+1}(U)}}{\min_{i=0,\dots,n} |\varrho^{(i)}(x_0)|} \max\{B, 1\}^{n+1} \cdot \frac{1}{\delta}$$
$$=: \frac{C'(B, n, \varrho)}{\delta}.$$

This implies, that there exists some $C \ge \max\{C_0, C'(B, n, \varrho)\}$ such that for every $\varepsilon \in (0, 1)$ and the neural network $\Phi_\varepsilon^r := ((A_1, b_1), (A_2, b_2))$ with

$$A_1 := \left(0, -\frac{\varepsilon}{C}, \dots, -\frac{r\varepsilon}{C}\right)^T \in \mathbb{R}^{r+1,1},$$
$$b_1 := (x_0, \dots, x_0)^T \in \mathbb{R}^{r+1},$$
$$A_2 := \frac{C^r}{\varepsilon^r \varrho^{(r)}(x_0)} \left(\binom{r}{0}, (-1)^1 \binom{r}{1}, \dots, (-1)^r \binom{r}{r}\right) \in \mathbb{R}^{1,r+1},$$
$$b_2 := 0 \in \mathbb{R},$$

fulfills

$$\left\|R_\varrho(\Phi_\varepsilon^r) - x^r\right\|_{C^n([-B,B])} \le \varepsilon.$$

Moreover, $L\left(\Phi_\varepsilon^r\right) = 2$ and $M\left(\Phi_\varepsilon^r\right) \le 3(r+1)$.

Additionally, for every $k = 0, \dots, r$ and for every $x \in [-B, B]$ we have

$$\left|\left(R_\varrho(\Phi_\varepsilon^r)\right)^{(k)}(x)\right| \le \left\|\left(R_\varrho(\Phi_\varepsilon^r)\right)^{(k)} - (x^r)^{(k)}\right\|_{C^n([-B,B])} + \left|(x^r)^{(k)}\right|$$
$$\le \varepsilon + \frac{n!}{(n-k)!} |\max\{1, B\}|^{r-k}.$$

Finally, for all $k = r+1, \dots, n$ we have that

$$\left|\left(R_\varrho(\Phi_\varepsilon^r)\right)^{(k)}(x)\right| \le \left\|\left(R_\varrho(\Phi_\varepsilon^r)\right)^{(k)} - (x^r)^{(k)}\right\|_{C^n([-B,B])} + \left|(x^r)^{(k)}\right|$$
$$\le \varepsilon + 0 = \varepsilon.$$

This completes the proof. □

Based on Proposition 4.7, we are now in a position to introduce neural networks that approximate the map which multiplies two real inputs.

Corollary C.3. *Let $\varrho \in W^{j,\infty}_{\mathrm{loc}}(\mathbb{R})$ for some $j \in \mathbb{N}_0$ and $x_0 \in \mathbb{R}$ such that ϱ is three times continuously differentiable in a neighborhood of some $x_0 \in \mathbb{R}$ and $\varrho''(x_0) \neq 0$. Let $B > 0$, then there exists a constant $C = C(B, \varrho) > 0$ such that for every $\varepsilon \in (0, 1/2)$, there is a neural network $\widetilde{\times}$ with two-dimensional input and one-dimensional output that satisfies the following properties:*

(i) $\|R_\varrho(\widetilde{\times})(x, y) - xy\|_{W^{j,\infty}((-B,B)^2;dxdy)} \leq \varepsilon$;
(ii) $\|R_\varrho(\widetilde{\times}_\varepsilon)\|_{W^{j,\infty}((-B,B)^2)} \leq C$;
(iii) $L(\widetilde{\times}) = 2$ *and* $M(\widetilde{\times}) \leq C$;
(iv) $\|\widetilde{\times}\|_{\max} \leq C\varepsilon^{-2}$.

Proof. Let C be the constant from Corollary B.6 and set $\widetilde{\varepsilon} := \varepsilon/2C$. Proposition 4.7 yields that there exists a neural network $\Phi^2_{\widetilde{\varepsilon}}$ with 2 layers and at most 9 nonzero weights such that for all $k \in \{0, \dots, j\}$ we have

$$\left|R_\varrho(\Phi^2_{\widetilde{\varepsilon}}) - x^2\right|_{W^{k,\infty}([-2B,2B];dx)} \leq \widetilde{\varepsilon}.$$

As in Yarotsky (2017), we make use of the polarization identity

$$xy = \frac{1}{4}\left((x+y)^2 - (x-y)^2\right) \quad \text{for } x, y \in \mathbb{R}.$$

In detail, we define the neural network

$$\widetilde{\times}_\varepsilon := \left(\left(\frac{1}{4}, \frac{-1}{4}\right), 0\right) \bullet \mathrm{P}\left(\Phi^2_{\widetilde{\varepsilon}}, \Phi^2_{\widetilde{\varepsilon}}\right) \bullet \left(\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, 0\right),$$

which fulfills for all $(x, y) \in \mathbb{R}^2$ that

$$R_\varrho(\widetilde{\times}_\varepsilon)(x, y) = \frac{1}{4}\left(R_\varrho\left(\Phi^2_{\widetilde{\varepsilon}}\right)(x+y) - R_\varrho\left(\Phi^2_{\widetilde{\varepsilon}}\right)(x-y)\right).$$

Now, setting $f : [-2B, 2B] \to \mathbb{R}, x \mapsto x^2$ as well as

$$u : [-B, B]^2 \to [-2B, 2B],\ (x, y) \mapsto x + y \quad \text{and}$$
$$v : [-B, B]^2 \to [-2B, 2B],\ (x, y) \mapsto x - y,$$

we see that for all $(x, y) \in [-B, B]^2$ there holds $xy = 1/4\,(f \circ u(x, y) - f \circ v(x, y))$. We estimate

$$\begin{aligned}
&\left\|R_\varrho(\widetilde{\times}_\varepsilon)(x, y) - xy\right\|_{W^{k,\infty}([-B,B]^2;dxdy)} \\
&= \frac{1}{4}\left\|R_\varrho\left(\Phi^2_{\widetilde{\varepsilon}}\right) \circ u - R_\varrho\left(\Phi^2_{\widetilde{\varepsilon}}\right) \circ v - (f \circ u - f \circ v)\right\|_{W^{k,\infty}([-B,B]^2)} \\
&\leq \frac{1}{4}\left\|R_\varrho\left(\Phi^2_{\widetilde{\varepsilon}}\right) \circ u - f \circ u\right\|_{W^{k,\infty}([-B,B]^2)} \\
&\qquad + \frac{1}{4}\left\|R_\varrho\left(\Phi^2_{\widetilde{\varepsilon}}\right) \circ v - f \circ v\right\|_{W^{k,\infty}([-B,B]^2)},
\end{aligned}$$

and directly see for $k = 0$ that

$$\begin{aligned}
&\left|R_\varrho(\widetilde{\times}_\varepsilon)(x, y) - xy\right|_{W^{0,\infty}([-B,B]^2;dxdy)} \\
&\leq \frac{2}{4}\|R_\varrho\left(\Phi^2_{\widetilde{\varepsilon}}\right) - x^2\|_{L^\infty([-B,B]^2;dx)} \leq \frac{1}{2}\widetilde{\varepsilon} \leq \varepsilon.
\end{aligned}$$

Now, we proceed with the case $k \in \{1, \dots, j\}$. We first note that

$$\begin{aligned}
|u|_{W^{0,\infty}([-B,B]^2)} &= |v|_{W^{0,\infty}([-B,B]^2)} = 2B, \\
|u|_{W^{1,\infty}([-B,B]^2)} &= |v|_{W^{1,\infty}([-B,B]^2)} = 1, \\
|u|_{W^{k,\infty}([-B,B]^2)} &= |v|_{W^{k,\infty}([-B,B]^2)} = 0, \ \text{for all } k \geq 2.
\end{aligned}$$

The composition rule from Corollary B.6 then yields that

$$\begin{aligned}
&\left|R_\varrho(\widetilde{\times}_\varepsilon)(x, y) - xy\right|_{W^{k,\infty}([-B,B]^2;dxdy)} \\
&\leq 2C\sum_{i=1}^{k}\left|R_\varrho\left(\Phi^2_{\widetilde{\varepsilon}}\right) - x^2\right|_{W^{i,\infty}([-2B,2B];dx)} |u|^i_{W^{1,\infty}([-B,B]^2)} \\
&\leq 2C\widetilde{\varepsilon} = \varepsilon.
\end{aligned}$$

and, thus, claim (i) is shown. Finally, we have for $k \in \{0, \dots, j\}$

$$\begin{aligned}
\left|R_\varrho(\widetilde{\times}_\varepsilon)\right|_{W^{k,\infty}([-B,B]^2)} &\leq \left|R_\varrho(\widetilde{\times}_\varepsilon) - xy\right|_{W^{k,\infty}([-B,B]^2;dxdy)} \\
&\qquad + |xy|_{W^{k,\infty}([-B,B]^2;dxdy)} \leq C_1,
\end{aligned}$$

for a constant $C_1 = C_1(B) > 0$, yielding (ii). Claims (iii), (iv) immediately follow from the construction of $\widetilde{\times}$ in combination with Proposition 4.7 and Lemma C.5.(i). □

Another statement that can be deduced from Proposition 4.7 is connected to the construction of neural networks which approximate the identity on $\mathbb{R}^d$.

Corollary C.4. *Let $\varrho : \mathbb{R} \to \mathbb{R}$ be such that ϱ is twice times continuously differentiable in a neighborhood of some $x_0 \in \mathbb{R}$ and $\varrho'(x_0) \neq 0$ fulfill the assumptions of Proposition 4.7 for some $n = 2$, for $r = 1$ and assume that for some $k \leq n$ we have that $\varrho \in W^{k,\infty}_{\mathrm{loc}}(\mathbb{R})$. Then, for every $B > 0$, $d \in \mathbb{N}$, for every $L \in \mathbb{N}_{\geq 2}$ and for every $\varepsilon \in (0, 1)$ there exists a constant $C = C(B, \varrho) > 0$ and a neural network $\Phi^{L,B,d}_\varepsilon$ with d-dimensional input, d-dimensional output and the following properties:*

(i) $\left\|R_\varrho(\Phi^{L,B,d}_\varepsilon) - x\right\|_{W^{k,\infty}([-B,B]^d;\mathbb{R}^d)} \leq \varepsilon$;
(ii) $\|R_\varrho(\Phi^{L,B,d}_\varepsilon)\|_{W^{k,\infty}([-B,B]^d;\mathbb{R}^d)} \leq C\max\{1, B\}$;
(iii) $L\left(\Phi^{L,B,d}_\varepsilon\right) = L$*, as well as* $M\left(\Phi^{L,B,d}_\varepsilon\right) \leq 4dL - 3d$;
(iv) $\|\Phi^{L,B,d}_\varepsilon\|_{\max} \leq CL\varepsilon^{-1}$.

Proof. W.l.o.g., we assume that $d = 1$. The other cases follow from a minor modification of the parallelization of neural networks with the same number of layers. Let $\Phi^1_{\varepsilon/L}$ be the neural network from Proposition 4.7 for $B = B + 1$. We define $\Phi^{L,B,d}_\varepsilon := \Phi^1_{\varepsilon/L} \bullet \dots \bullet \Phi^1_{\varepsilon/L}$, where we perform $L - 2$ concatenations. It is easy to see that $\Phi^{L,B,d}_\varepsilon = ((A_1, b_1), (A_2, b_2), \dots, (A_L, b_L))$, where

$$\begin{aligned}
A_1 &= \left(0, -\frac{\varepsilon}{LC}\right)^T \in \mathbb{R}^{2,1}, \\
b_1 &= (x_0, x_0)^T \in \mathbb{R}^2, \\
A_\ell &= \begin{pmatrix} 0 & 0 \\ -\frac{1}{\varrho'(x_0)} & \frac{1}{\varrho'(x_0)} \end{pmatrix} \in \mathbb{R}^{2,2}, \qquad \text{for } \ell = 2, \dots, L-1, \\
b_\ell &= (x_0, x_0)^T \in \mathbb{R}^2, \qquad \text{for } \ell = 2, \dots, L-1, \\
A_L &= \frac{LC}{\varepsilon\varrho'(x_0)}(1, -1) \in \mathbb{R}^{1,2}, \\
b_L &= 0 \in \mathbb{R},
\end{aligned}$$

and where $C > 0$ is a suitable constant provided by Proposition 4.7. By Proposition 4.7 we also have that $R_\varrho(\Phi^1_{\varepsilon/L})(x) \in [-B - \varepsilon/L, B + \varepsilon/L]$ for all $x \in [-B, B]$ as well as

$$\left\|R_\varrho(\Phi^1_{\varepsilon/L}) - x\right\|_{W^{k,\infty}([-B,B])} \leq \frac{\varepsilon}{L}.$$

Iterating this argument shows that $R_\varrho(\Phi^{L,B}_\varepsilon)(x) \in [-B - \varepsilon, B + \varepsilon]$ for all $x \in [-B, B]$ and that

$$\left\|R_\varrho(\Phi^{L,B,d}_\varepsilon) - x\right\|_{W^{k,\infty}([-B,B])} \leq \varepsilon.$$

The other properties follow immediately from (i) in combination with the definition of $\Phi^{L,B,d}_\varepsilon$. □

Before we continue, let us have a closer look at the properties of the concatenation of two neural networks in the following special cases.

Lemma C.5. *Let Φ be a neural network with m-dimensional output.*

(i) If $a \in \mathbb{R}^{1\times m}$, then,

$$M(((a, 0)) \bullet \Phi) \leq M(\Phi) \quad \textit{and} \quad \|((a, 0)) \bullet \Phi)\|_{\max}$$

$$\leq m\|\Phi\|_{\max} \max_{i=1,\dots,m} a_i.$$

(ii) Let $\Phi_\varepsilon^{L,B,m}$ be the approximate identity network from Corollary C.4. Then, for some constant $C = C(B,\varrho)$ there holds

$$M(\Phi_\varepsilon^{L,B,m} \bullet \Phi) \leq M(\Phi) + M(\Phi_\varepsilon^{L,B,m}), \quad \text{and}$$
$$\|(\Phi_\varepsilon^{L,B,m} \bullet \Phi)\|_{\max} \leq C\max\{\|\Phi\|_{\max}, \varepsilon^{-1}\}.$$

(iii) Let $\widetilde{\times}$ be the approximate multiplication network from Corollary C.3. If $m = 2$, then, for some constant $C = C(B,\varrho)$ there holds

$$M(\widetilde{\times} \bullet \Phi) \leq CM(\Phi) \quad \text{and} \quad \|\widetilde{\times} \bullet \Phi\|_{\max} \leq C\max\{\|\Phi\|_{\max}, \varepsilon^{-2}\}.$$

Proof. For the first part of the proof of (i), see Kutyniok et al. (2019). The second part is clear.

From now on, let $\Phi = ((A_1, b_1), \dots, (A_{L(\Phi)}, b_{L(\Phi)}))$.

For the proof of (ii), let $\Phi_\varepsilon^{L,B,m} = ((A_1^{\mathrm{id}}, b_1^{\mathrm{id}}), \dots, (A_L^{\mathrm{id}}, b_L^{\mathrm{id}}))$ and recall that

$\Phi_\varepsilon^{L,B,m} \bullet \Phi = ((A_1, b_1), \dots, (A_{L(\Phi)-1}, b_{L(\Phi)-1}), (A_1^{\mathrm{id}} A_{L(\Phi)}, A_1^{\mathrm{id}} b_{L(\Phi)} + b_1^{\mathrm{id}}), (A_2^{\mathrm{id}}, b_2^{\mathrm{id}}), \dots, (A_L^{\mathrm{id}}, b_L^{\mathrm{id}}))$. Hence, in order to prove (ii), we only need to examine $(A_1^{\mathrm{id}} A_{L(\Phi)}, A_1^{\mathrm{id}} b_{L(\Phi)} + b_1^{\mathrm{id}})$. From the construction of $\Phi_\varepsilon^{L,B,m}$ we have that $\|A_1^{\mathrm{id}}\|_0 = m$ and that A_1^{id} has block diagonal structure. Additionally, all entries of A_1^{id} are bounded in absolute value by $\frac{\varepsilon}{L\widetilde{C}} \leq 1$ for some $\widetilde{C} \geq 1$. From this, the claim follows.

The proof of (iii) can be done in a similar manner as the proof of (ii). □

Appendix D. Proof of Proposition 4.8

In this section we provide the proofs of those statements of Section 4 as well as additional auxiliary statements which together lead to the proof of Proposition 4.8. Appendix D.1 is concerned with the proof of Lemma 4.5 which establishes the conditions of the PU. Appendix D.2, which contains the proof of Lemma D.1, shows that we are in a position to efficiently approximate $f \in \mathcal{F}_{n,d,p}$ by sums of polynomials multiplied with the functions from the PU. Appendix D.3 in turn shows that these sums of localized polynomials can be approximated by neural networks. Appendix D.4 concludes the proof of Proposition 4.8.

D.1. Approximate partition of unity

We start with the proof of Lemma 4.5 which establishes the properties of the exponential (polynomial, exact) (j,τ)-PU.

Proof of Lemma 4.5. For the proof of the properties (i) and (ii), we will always assume w.l.o.g. that $m = 0$ unless stated otherwise. Moreover, we only give the proof for the case of an exponential PU. The other cases follow in essentially the same way with some simplifications.

ad (i): First of all, assume that $d = 1$. For $\underline{\tau = 0}$ and $j = 0$ this follows directly from the boundedness of ϱ. For $\underline{\tau = 1}$ and $j = 0$, we have that ϱ is Lipschitz continuous, and, thus,

$$\left|\phi_0^s(x)\right| \leq \frac{1}{s(B-A)}\left|\varrho(3sNx + 2s) - \varrho(3sNx + s)\right|$$
$$+ \frac{1}{s(B-A)}\left|\varrho(3sNx - s) - \varrho(3sNx - 2s)\right|$$
$$\leq 2\frac{\mathrm{Lip}(\varrho)\cdot s}{s(B-A)} = 2\frac{\mathrm{Lip}(\varrho)}{(B-A)}.$$

For $\underline{\tau \in \{0,1\}}$ and $j \geq 1$ this follows from the case $j = 0$ together with $\varrho' \in W^{j-1,\infty}(\mathbb{R})$ and the chain rule.

Now, let $d \in \mathbb{N}$ be arbitrary. Since we will need it in the proof of (ii), we prove the following more general statement (Statement (i) follows by considering $I = \{1,\dots,d\}$). Moreover, we will prove this statement only for $k \leq \min\{j, 2\}$, since the rest of the proof can be done in exactly the same way by exploiting the tensor structure of ϕ_m^s.

Let $I \subset \{1,\dots,d\}$ be arbitrary. Moreover, for $m \in \{0,\dots,N\}^{|I|}$ we define $\phi_{m,I}^s : \mathbb{R}^{|I|} \to \mathbb{R}, x \mapsto \prod_{1\leq l\leq |I|} \psi^s\left(3N\left(x_l - \frac{m_l}{N}\right)\right)$ as well as $\phi_m^s := \phi_{m,I}^s$, if $I = \{1,\dots,d\}$. Then for $k \in \{0,\dots,j\}$ it holds that

$$\left|\phi_{m,I}^s\right|_{W^{k,\infty}(\mathbb{R}^{|I|})} \leq C^{|I|} \cdot N^k \cdot s^{\max\{0,k-\tau\}}.$$

It is clear that by the definition of $\phi_{m,I}^s$ and what we have shown for $d = 1$ that for $k = 0$ there holds

$$\left|\phi_{m,I}^s\right|_{W^{0,\infty}(\mathbb{R}^{|I|})} \leq C^{|I|}. \tag{D.1}$$

Now, let $i \in I$ be arbitrary. Then, by using the tensor product structure of $\phi_{m,I}^s$ in combination with what we have shown before for $d = 1$, for the case $k = 1$ and (D.1) for $I' := I \setminus \{i\}$ we obtain for a.e. $x \in \mathbb{R}^{|I|}$

$$\left|\frac{\partial}{\partial x_i}\phi_{m,I}^s(x)\right|$$
$$= \left|\phi_{m,I'}^s(x_1,\dots,x_{i-1},x_{i+1},\dots,x_{|I|})\right| \cdot \left|\left(\psi^s\left(3N\left(\cdot - {}^{m_i}/_{N}\right)\right)\right)'(x_i)\right|$$
$$\leq C^{|I|-1}\cdot CN = C^{|I|}Ns^{\max\{0,k-\tau\}}$$

which implies that $|\phi_{m,I}^s|_{W^{1,\infty}(\mathbb{R}^{|I|})} \leq C^{|I|}N$.

Finally, let additionally be $r \in I$ be arbitrary. If $i = r$ then we have that (by using (D.1) in combination with what we have shown for $d = 1$) that

$$\left|\frac{\partial^2}{\partial x_i^2}\phi_{m,I}^s(x)\right|$$
$$= \left|\phi_{m,I'}^s(x_1,\dots,x_{i-1},x_{i+1},\dots,x_{|I|})\right| \cdot \left|\left(\psi^s\left(3N\left(\cdot - {}^{m_i}/_{N}\right)\right)\right)''(x_i)\right|$$
$$\leq C^{|I|-1}\cdot CN^2s^{\max\{0,k-\tau\}} = C^{|I|}N^2s^{\max\{0,k-\tau\}}.$$

Moreover, if $i \neq r$, then, if we set $I'' := I \setminus \{i, r\}$ we obtain with similar arguments as before that

$$\left|\frac{\partial^2}{\partial x_i \partial x_r}\phi_{m,I}^s(x)\right|$$
$$= \left|\phi_{m,I''}^s(x_1,\dots,x_{i-1},x_{i+1},\dots,x_{r-1},x_{r+1},\dots,x_{|I|})\right|$$
$$\cdot \left|\left(\psi^s\left(3N\left(\cdot - {}^{m_i}/_{N}\right)\right)\right)'(x_i)\right| \cdot \left|\left(\psi^s\left(3N\left(\cdot - {}^{m_r}/_{N}\right)\right)\right)'(x_r)\right|$$
$$\leq C^{|I|-2}\cdot CN\cdot CNs^{\max\{0,k-\tau\}} = C^{|I|}N^2s^{\max\{0,k-\tau\}},$$

where we assumed w.l.o.g. that $i < r$. This implies $|\phi_{m,I}^s|_{W^{2,\infty}(\mathbb{R}^{|I|})} \leq C^{|I|}N^2s^{\max\{0,k-\tau\}}$.

ad (ii): First of all, assume that $d = 1$. Let $\underline{\tau = 0}$ and let $x \leq -1/N$. Then, since $s > R$, we have that $3Nsx + 3/2s, 3Nsx - 3/2 \leq -R$. We then have by the triangle inequality and the assumption on ϱ that

$$\left|\phi_0^s(x)\right| = \left|\frac{\varrho(3Nsx + 3/2s) - \varrho(3Nsx - 3/2s)}{B-A}\right|$$
$$\leq \left|\frac{\varrho(3Nsx + 3/2s) - A}{B-A}\right| + \left|\frac{\varrho(3Nsx - 3/2s) - A}{B-A}\right|$$
$$\leq \frac{C'e^{D(3Nsx+3/2s)} + C'e^{D(3Nsx-3/2s)}}{B-A}$$

$$\leq \frac{C'e^{D(-3s+3/2s)} + C'e^{D(-3s-3/2s)}}{B-A} \leq 2C'\frac{e^{-Ds}}{B-A}.$$

Now, let $k \in \{1, \dots, j\}$. Then, by the assumption on ϱ, we have

$$\begin{aligned}
&\left|(\phi_0^s)^{(k)}(x)\right| \\
&= (3Ns)^k \left|\frac{\varrho^{(k)}(3Nsx + 3/2s) - \varrho^{(k)}(3Nsx - 3/2s)}{B-A}\right| \\
&\leq (3Ns)^k \left|\frac{\varrho^{(k)}(3Nsx + 3/2s)}{B-A}\right| + \left|\frac{\varrho^{(k)}(3Nsx - 3/2s)}{B-A}\right| \\
&\leq \frac{C'(3Ns)^k \left(e^{D(3Nsx+3/2s)} + e^{D(3Nsx-3/2s)}\right)}{B-A} \\
&\leq \frac{C'(3Ns)^k \left(e^{D(-3s+3/2s)} + e^{D(-3s-3/2s)}\right)}{B-A} \\
&\leq \frac{2C'}{B-A}(3Ns)^k e^{-Ds}.
\end{aligned}$$

The case $x \geq 1/N$ can be proven in the same way.

Now let $\underline{\tau = 1}$ and let again $x \leq -1/N$. Then $3Nsx + 2s, 3Nsx + s, 3Nsx - 2s, 3Nsx - s \leq -s < -R$. By the mean value theorem there exist $\xi_1 \in (3Nsx + s, 3Nsx + 2s)$ and $\xi_2 \in (3Nsx - 2s, 3Nsx - s)$ such that

$$\phi_0^s(x) = \frac{1}{s(B-A)}\left(\varrho'(\xi_1^x) - \varrho'(\xi_2^x)\right).$$

The remainder of the proof follows in exactly the same way as the proof of the analogous statement for $\tau = 0$. The statement for $x \geq 1/N$ can be done in exactly the same manner. Now, let $d \in \mathbb{N}$ and let $x \in \Omega_m^c$. Then there exists some $l \in \{1, \dots d\}$ with $|x_l - \frac{m_l}{N}| \geq 1/N$. This implies for $I' = \{1, \dots, d\} \setminus \{l\}$ by employing Eq. (D.1) that

$$\begin{aligned}
|\phi_m^s(x)| &= \left|\phi_{m,I'}^s(x_1, \dots, x_{l-1}, x_{l+1}, \dots, x_d)\right| \cdot \left|\psi^s\left(3N\left(x_l - {}^{m_l}\!/\!{}_N\right)\right)\right| \\
&\leq C^{d-1} \cdot Ce^{-Ds}.
\end{aligned}$$

This shows that $|\phi_m^s|_{W^{0,\infty}(\Omega_m^c)} \leq C^d e^{-Ds}$. By proceeding in a similar manner and with the same techniques as in the proof of (i), one can show the remaining Sobolev semi-norm estimates for the higher-order derivatives.

ad (iii): First of all, assume that $d = 1$. Let $\underline{\tau = 0}$. It is not hard to see that

$$\sum_{m=0}^{N} \phi_m^s(x) = \frac{1}{B-A}\left(\varrho(3Nsx + 3/2s) - \varrho(3Ns(x-1) - 3/2s)\right).$$

We now have for all $x \in (0, 1)$ and using the properties of ϱ that

$$\begin{aligned}
\left|1 - \sum_{m=0}^{N}\phi_m^s(x)\right| &= \left|\frac{B - A - (\varrho(3Nsx + 3/2s) - \varrho(3Ns(x-1) - 3/2s))}{B-A}\right| \\
&\leq \left|\frac{B - \varrho(3Nsx + 3/2s)}{B-A}\right| \\
&\quad + \left|\frac{A - \varrho(3Nsx - 3Ns - 3/2s)}{B-A}\right| =: \mathrm{I} + \mathrm{II}.
\end{aligned}$$

We continue by estimating I. Since $3Nsx + 3/2s \geq 3/2s > 3/2R$, we obtain that

$$\mathrm{I} \leq \frac{C'e^{-D(3Nsx+3/2s)}}{B-A} \leq \frac{C'e^{-3/2\cdot Ds}}{B-A}$$

On the other hand, since $3Nsx - 3Ns - 3/2s \leq -3/2s \leq 0$ we obtain that

$$\mathrm{II} \leq \frac{C'e^{D(3Nsx-3Ns-3/2s)}}{B-A} \leq \frac{C'e^{-3/2\cdot Ds}}{B-A}.$$

For the multidimensional case we have,

$$\begin{aligned}
&\left|1 - \sum_{m\in\{0,\dots,N\}^d}\phi_m^s(x)\right| \\
&= \left|1 - \sum_{m\in\{0,\dots,N\}^d}\prod_{l=1}^{d}\psi^s\left(3N\left(x_l - \frac{m_l}{N}\right)\right)\right| \\
&= \left|1 - \prod_{l=1}^{d}\sum_{m=0}^{N}\psi^s\left(3N\left(x_l - \frac{m}{N}\right)\right)\right| \\
&= \left|1 - \prod_{l=1}^{d}\left(\underbrace{\frac{1}{B-A}\left(\varrho(3Nsx_l + 3/2s) - \varrho(3Ns(x_l - 1) - 3/2s)\right)}_{:=\pi_l,\ \text{and}\ \pi_0:=1}\right)\right| \\
&\leq \sum_{l=1}^{d}|\pi_0 \cdot \ldots \cdot \pi_l(1 - \pi_{l+1})| \leq C \cdot e^{-3/2Ds},
\end{aligned}$$

which follows from the one-dimensional case. Now, let $k \in \{1, \dots, j\}$ and we consider only the case $d = 1$. The multi-dimensional case follows in exactly the same manner as the analogous considerations in (i) and (ii). We have that

$$\begin{aligned}
\left|\left(\sum_{m=0}^{N}\phi_m^s\right)^{(k)}(x)\right| &\leq (3Ns)^k \cdot \frac{1}{B-A}\left(\left|\varrho^{(k)}(3Nsx + 3/2s)\right|\right. \\
&\quad \left. + \left|\varrho^{(k)}(3Nsx - 3Ns - 3/2s)\right|\right)
\end{aligned}$$

Since $x > 0$, we have that $3Nsx + 3/2s \geq 3/2s > R$. Since $x < 1$, $3Nsx - 3Ns - 3/2s \leq -3/2R < -R$. Hence, by the assumptions on ϱ we obtain that

$$\begin{aligned}
\left|\left(\sum_{m=0}^{N}\phi_m^s\right)^{(k)}(x)\right| &\leq \frac{C'(3Ns)^k}{B-A}\left(e^{-D(3Nsx+3/2s)} + e^{D(3Nsx-3Ns-3/2s)}\right) \\
&\leq \frac{2C'(3Ns)^k}{B-A}e^{-3/2Ds}.
\end{aligned}$$

The multidimensional case for $k \in \{0, \dots, j\}$ follows in a similar manner as above from the tensor structure. Now, let $\underline{\tau = 1}$. It is not hard to see that for all $x \in \mathbb{R}$ there holds

$$\begin{aligned}
\sum_{m=0}^{N}\phi_m^s(x) = \frac{1}{s(B-A)}&\left(\varrho(3Nsx + 2s) - \varrho(3Nsx + s)\right. \\
&\left. -\varrho(3Nsx - 3Ns - s) + \varrho(3Nsx - 3Ns - 2s)\right).
\end{aligned}$$

Now, let $x \in (0, 1)$. We have that $3Nsx + 2s, 3Nsx + s \geq s > R$ and $3Nsx - 3Ns - s, 3Nsx - 3Ns - 2s \leq -s < -R$. Hence, by the mean value theorem, for every $x \in \mathbb{R}$ there exist $\xi_1 \in (3Nxs + s, 3Nxs + 2s)$ and $\xi_2 \in (3Nxs - 3Ns - 2s, 3Nxs - 3Ns - s)$ such that

$$\sum_{m=0}^{N}\phi_m^s(x) = \frac{1}{B-A}\left(\varrho'(\xi_1) - \varrho'(\xi_2)\right).$$

Now we have that

$$\left|1-\sum_{m=0}^{N}\phi_m^s(x)\right| \le \left|\frac{B-\varrho'(\xi_1)}{B-A}\right| + \left|\frac{A-\varrho'(\xi_2)}{B-A}\right|.$$

The remainder of the statement can be proven in exactly the same way as the analogous statement for $\tau = 0$.

ad (iv): This immediately follows from the definition of the functions ϕ_m^s. □

D.2. Approximation by localized polynomials

In this section, we demonstrate how to approximate a function $f \in \mathcal{F}_{n,d,p}$ by localized polynomials based on the exponential (polynomial, exact) (j,τ)-PU. We only give the proof for the case of an exponential PU. The other cases follow in essentially the same way with some simplifications.

Lemma D.1. *We make the following assumption:*

- *Let $d \in \mathbb{N}$, $j, \tau \in \mathbb{N}_0$, $k \in \{0,\dots,j\}$, $n \in \mathbb{N}_{\ge k+1}$ and $1 \le p \le \infty$.*
- *Assume that $(\Psi^{(j,\tau,N,s)})_{N\in\mathbb{N},s\ge 1}$ is an exponential (polynomial, exact) (j,τ)-PU from Definition 4.1. Let $\mu \in (0,1)$. For $N \in \mathbb{N}$, set*

$$s := \begin{cases} N^{\mu}, & \text{if exponential PU,} \\ N^{\frac{2d/p+d+n}{D}}, & \text{if polynomial PU,} \\ 1, & \text{if exact PU,} \end{cases}$$

Then there is a constant $C = C(d,n,p,k) > 0$ and $\widetilde{N} = \widetilde{N}(d, p, \mu, k, \tau) \in \mathbb{N}$ such that for every $f \in W^{n,p}((0,1)^d)$ and every $m \in \{0,\dots,N\}^d$, there exist polynomials $p_{f,m}(x) = \sum_{|\alpha|\le n-1} c_{f,m,\alpha} x^\alpha$ for $m \in \{0,\dots,N\}^d$ with the following properties:

Set $f_N := \sum_{m\in\{0,\dots,N\}^d} \phi_m^s p_{f,m}$. Then, the operator $T_k : W^{n,p}((0,1)^d) \to W^{k,p}((0,1)^d)$ with $T_k f = f - f_N$ is linear and bounded with

$$\|T_k f\|_{W^{k,p}((0,1)^d)} \le C\|f\|_{W^{n,p}((0,1)^d)} \begin{cases} \left(\frac{1}{N}\right)^{n-k-\mu(k=2)}, & \text{if exponential PU,} \\ \left(\frac{1}{N}\right)^{n-k}, & \text{for } k\le\tau, \text{ if polynomial PU,} \\ \left(\frac{1}{N}\right)^{n-k}, & \text{if exact PU,} \end{cases}$$

for all $N \in \mathbb{N}$ with $N \ge \widetilde{N}$.

Before the proof of this statement, we need some preparation. We start with the following observation.

Remark D.2. Since the polynomials utilized in Lemma D.1 are the averaged Taylor polynomials from the Bramble–Hilbert Lemma B.4, we get that there is a constant $C = C(d,n,k) > 0$ such that for any $f \in W^{n,p}((0,1)^d)$ the coefficients of the polynomials $p_{f,m}$ satisfy

$$|c_{f,m,\alpha}| \le C\|\widetilde{f}\|_{W^{n,p}(\Omega_{m,N})} N^{d/p},$$

for all $\alpha \in \mathbb{N}_0^d$ with $|\alpha| \le n-1$, and for all $m \in \{0,\dots,N\}^d$, where $\Omega_{m,N} := B_{\frac{1}{N},\|\cdot\|_\infty}\left(\frac{m}{N}\right)$ and $\widetilde{f} \in W^{n,p}(\mathbb{R}^d)$ is an extension of f.

We now state and prove an auxiliary result. The estimation will be very rough and can for sure be improved. This is, however, not necessary for our purpose.

Lemma D.3. *Under the conditions of Lemma D.1 and with the notation from Remark D.2 we have for all $m, \widetilde{m} \in \{0,\dots,N\}^d$ the estimate*

$$\|\widetilde{f} - p_{f,m}\|_{W^{k,p}(\Omega_{\widetilde{m},N})} \le CN^{d/p}\|f\|_{W^{n,p}((0,1)^d)},$$

for a constant $C = C(n,d,p,k)$.

Proof. We start with bounding the norm of the polynomial by using the triangle inequality. There holds

$$\|p_{f,m}\|_{W^{k,p}(\Omega_{\widetilde{m},N})} = \left\|\sum_{|\alpha|\le n-1} c_{f,m,\alpha}x^\alpha\right\|_{W^{k,p}(\Omega_{\widetilde{m},N};dx)} \le \sum_{|\alpha|\le n-1} |c_{f,m,\alpha}|\cdot\|x^\alpha\|_{W^{k,p}(\Omega_{\widetilde{m},N};dx)}.$$

Using that $\Omega_{\widetilde{m},N} \subset B_{2,\|\cdot\|_\infty}$ we get

$$\|x^\alpha\|_{W^{k,p}(\Omega_{\widetilde{m},N};dx)} \le (n-1)^k 2^{|\alpha|} \le (n-1)^k 2^{n-1}. \qquad \text{(D.2)}$$

If we now combine Remark D.2 with Eq. (D.2), we get

$$\sum_{|\alpha|\le n-1} |c_{f,m,\alpha}|\|x^\alpha\|_{W^{k,p}(\Omega_{\widetilde{m},N};dx)} \le C(n-1)^k 2^{n-1} \sum_{|\alpha|\le n-1} N^{d/p}\|\widetilde{f}\|_{W^{n,p}(\Omega_{m,N})} \le CN^{d/p}\|f\|_{W^{n,p}((0,1)^d)},$$

where we have additionally used Remark B.3 in the last step. Finally, we can estimate, by the triangle inequality

$$\|\widetilde{f} - p_{f,m}\|_{W^{k,p}(\Omega_{\widetilde{m},N})} \le C\|f\|_{W^{k,p}((0,1)^d)} + CN^{d/p}\|f\|_{W^{n,p}((0,1)^d)} \le CN^{d/p}\|f\|_{W^{n,p}((0,1)^d)},$$

where we again used the extension property from Eq. (B.1) for the first step. □

Now we are in a position to prove Lemma D.1.

Proof of Lemma D.1. We use approximation properties of the polynomials from the Bramble–Hilbert Lemma B.4 to derive local estimates and then combine them using an exponential PU to obtain a global estimate. In order to use this strategy also near the boundary, we make use of an extension operator (see Remark B.3).

Step 1 (Local estimates based on Bramble–Hilbert): For each $m \in \{0,\dots,N\}^d$ we set

$$\Omega_{m,N} := B_{\frac{1}{N},\|\cdot\|_\infty}\left(\frac{m}{N}\right)$$

and denote by $p_m = p_{f,m}$ the polynomial from Lemma B.4 so that we can directly state the estimate

$$\|\widetilde{f} - p_m\|_{W^{k,p}(\Omega_{m,N})} \le C\left(\frac{1}{N}\right)^{n-k}\|\widetilde{f}\|_{W^{n,p}(\Omega_{m,N})}. \qquad \text{(D.3)}$$

Furthermore, similarly to Gühring et al. (2020, Lemma C.4), we obtain the estimate

$$\begin{aligned}\|\phi_m^s(\widetilde{f} - p_m)\|_{W^{k,p}(\Omega_{m,N})} &\le C\sum_{\kappa=0}^{k}\|\phi_m^s\|_{W^{\kappa,\infty}(\Omega_{m,N})}\|\widetilde{f} - p_m\|_{W^{k-\kappa,p}(\Omega_{m,N})} \\ &\le C\sum_{\kappa=0}^{k} N^{\kappa+\mu(\kappa=2)}\left(\frac{1}{N}\right)^{n-k+\kappa}\|\widetilde{f}\|_{W^{n,p}(\Omega_{m,N})} \\ &\le C\left(\frac{1}{N}\right)^{n-k-\mu(k=2)}\|\widetilde{f}\|_{W^{n,p}(\Omega_{m,N})},\end{aligned}$$

where we used the product rule from Lemma B.5 for the first step and the estimate of the derivative of ϕ_m^s from Lemma 4.5(i) together with the Bramble–Hilbert estimate in Eq. (D.3) for the second step.

Step 2 (Local Estimates Based on Exponential Decay): Since our localizing bump functions ϕ_m^s do not necessarily have compact support on $\Omega_{m,N}$ we also need to bound the influence of $\phi_m^s(\widetilde{f} - p_m)$

on patches $\Omega_{\widetilde{m},N}$ with $\widetilde{m} \neq m$ where we cannot use the Bramble–Hilbert lemma. Here, we will make use of the exponential decay of the bump functions ϕ_m^s outside a certain ball centered at m/N (see Lemma 4.5(ii)).

This is possible for the case where $\Omega_{\widetilde{m},N}$ is not a neighboring patch of $\Omega_{m,N}$, i.e. $\|\widetilde{m} - m\|_\infty > 1$. Then $\Omega_{\widetilde{m},N} \subset \Omega_m^c$ and we have (by using Lemma B.5 in the first step), that

$$\begin{aligned}
\|\phi_m^s(\widetilde{f} - p_m)\|_{W^{k,p}(\Omega_{\widetilde{m},N})} &\leq C\|\phi_m^s\|_{W^{k,\infty}(\Omega_{\widetilde{m},N})}\|\widetilde{f} - p_m\|_{W^{k,p}(\Omega_{\widetilde{m},N})} \\
\text{(Lemma 4.5 (ii)) w/ } \Omega_{\widetilde{m},N} \subset \Omega_m^c) &\leq CN^{k+\mu_{(k=2)}} e^{-DN^\mu} \|\widetilde{f} - p_m\|_{W^{k,p}(\Omega_{\widetilde{m},N})} \\
\text{(Lemma D.3)} &\leq C \underbrace{N^{k+\mu_{(k=2)}} N^{d/p}}_{:=\gamma(N)} e^{-DN^\mu} \|f\|_{W^{n,p}((0,1)^d)}.
\end{aligned}$$

Then, by Proposition A.1, there exists $N_1 = N_1(\mu, d, p) \in \mathbb{N}$ such that

$$e^{-DN^\mu} \leq C\gamma(N)^{-1} \cdot (N+1)^{-d-d/p} \cdot N^{-(n-k-\mu_{(k=2)})},$$

for all $N \geq N_1$. Consequently, we have

$$\begin{aligned}
&\|\phi_m^s(\widetilde{f} - p_m)\|_{W^{k,p}(\Omega_{\widetilde{m},N})} \\
&\qquad \leq C(N+1)^{-d-d/p} N^{-(n-k-\mu_{(k=2)})} \|f\|_{W^{n,p}((0,1)^d)},
\end{aligned}$$

for all $N \geq N_1$.

Step 3 (Mixed Local Estimates): If $\Omega_{\widetilde{m},N}$ is a neighboring patch of $\Omega_{m,N}$, i.e. $\|\widetilde{m} - m\|_\infty = 1$, then we have to split the patch in a region $\Omega_{\widetilde{m},N} \cap \Omega_m^c$ where we have exponential decay of the bump function and a region $\Omega_{\widetilde{m},N} \setminus \Omega_m^c \subset \Omega_{m,N}$ where we can make use of the Bramble–Hilbert Lemma. In detail, we have

$$\begin{aligned}
&\|\phi_m^s(\widetilde{f} - p_m)\|_{W^{k,p}(\Omega_{\widetilde{m},N})} \\
&\leq \|\phi_m^s(\widetilde{f} - p_m)\|_{W^{k,p}(\Omega_{\widetilde{m},N} \setminus \Omega_m^c)} + \|\phi_m^s(\widetilde{f} - p_m)\|_{W^{k,p}(\Omega_{\widetilde{m},N} \cap \Omega_m^c)} \\
&\leq CN^{-(n-k-\mu_{(k=2)})} \left(\|\widetilde{f}\|_{W^{n,p}(\Omega_{m,N})} + (N+1)^{-d-d/p} \|f\|_{W^{n,p}((0,1)^d)}\right),
\end{aligned}$$

for all $N \geq N_1$. Here we used Step 1 to bound the first term of the sum and Step 2 for the second.

Step 4 (Global Estimate): Using that $\widetilde{f}$ is an extension of f on $(0,1)^d$ we can write

$$\begin{aligned}
&\left\| f - \sum_{m \in \{0,\dots,N\}^d} \phi_m^s p_m \right\|_{W^{k,p}((0,1)^d)} \\
&\leq \left\| \widetilde{f} - \sum_{m \in \{0,\dots,N\}^d} \phi_m^s \widetilde{f} \right\|_{W^{k,p}((0,1)^d)} \\
&\qquad + \left\| \sum_{m \in \{0,\dots,N\}^d} \phi_m^s (\widetilde{f} - p_m) \right\|_{W^{k,p}((0,1)^d)} \\
&\leq \underbrace{\left\| \widetilde{f} \left(\mathbb{1}_{(0,1)^d} - \sum_{m \in \{0,\dots,N\}^d} \phi_m^s \right) \right\|_{W^{k,p}((0,1)^d)}}_{\textbf{Step 4a}} \\
&\qquad + \left(\underbrace{\sum_{\widetilde{m} \in \{0,\dots,N\}^d} \left\| \sum_{m \in \{0,\dots,N\}^d} \phi_m^s (\widetilde{f} - p_m) \right\|^p_{W^{k,p}(\Omega_{\widetilde{m},N})}}_{\textbf{Step 4b}} \right)^{1/p},
\end{aligned} \tag{D.4}$$

where the last step follows from $(0,1)^d \subset \bigcup_{\widetilde{m} \in \{0,\dots,N\}^d} \Omega_{\widetilde{m},N}$.

Step 4a (Partition of Unity): For the first term in Eq. (D.4), we get by the product rule from Lemma B.5

$$\begin{aligned}
&\left\| \widetilde{f} \left(\mathbb{1}_{(0,1)^d} - \sum_{m \in \{0,\dots,N\}^d} \phi_m^s \right) \right\|_{W^{k,p}((0,1)^d)} \\
&\leq C\|f\|_{W^{k,p}((0,1)^d)} \left\| \mathbb{1}_{(0,1)^d} - \sum_{m \in \{0,\dots,N\}^d} \phi_m^s \right\|_{W^{k,\infty}((0,1)^d)} \\
\text{(Property (iii) from Lemma 4.5)} &\leq C\|f\|_{W^{k,p}((0,1)^d)} \cdot N^{-(n-k-\mu_{(k=2)})},
\end{aligned} \tag{D.5}$$

for all $N \geq N_2 = N_2(\mu, k, \tau)$. For the second inequality we used the same trick as in Step 2 which is based on Proposition A.1.

Step 4b (Patches): Considering the second term from Eq. (D.4), we obtain for each $\widetilde{m} \in \{0, \dots, N\}^d$

$$\begin{aligned}
&\left\| \sum_{m \in \{0,\dots,N\}^d} \phi_m^s (\widetilde{f} - p_m) \right\|_{W^{k,p}(\Omega_{\widetilde{m},N})} \\
&\leq \underbrace{\|\phi_{\widetilde{m}}^s(\widetilde{f} - p_{\widetilde{m}})\|_{W^{k,p}(\Omega_{\widetilde{m},N})}}_{(\star)} \\
&\qquad + \underbrace{\sum_{\substack{m \in \{0,\dots,N\}^d, \\ \|m - \widetilde{m}\|_\infty > 1}} \|\phi_m^s(\widetilde{f} - p_m)\|_{W^{k,p}(\Omega_{\widetilde{m},N})}}_{(\star\star)} \\
&\qquad\quad + \underbrace{\sum_{\substack{m \in \{0,\dots,N\}^d, \\ \|m - \widetilde{m}\|_\infty > 1}} \|\phi_m^s(\widetilde{f} - p_m)\|_{W^{k,p}(\Omega_{\widetilde{m},N})}}_{(\star\star\star)}.
\end{aligned} \tag{D.6}$$

The term $(\star)$ can be handled with Step 1, the term $(\star\star)$ with Step 3 and the third one $(\star\star\star)$ with Step 2. Since $(\star\star)$ and $(\star\star\star)$ require a similar strategy we only demonstrate it for the third term. We get from Step 2

$$\begin{aligned}
&\sum_{\substack{m \in \{0,\dots,N\}^d, \\ \|m - \widetilde{m}\|_\infty > 1}} \|\phi_m^s(\widetilde{f} - p_m)\|_{W^{k,p}(\Omega_{\widetilde{m},N})} \\
&\leq CN^{-(n-k-\mu_{(k=2)})}(N+1)^{-d-d/p} \sum_{\substack{m \in \{0,\dots,N\}^d, \\ \|m - \widetilde{m}\|_\infty > 1}} \|f\|_{W^{n,p}((0,1)^d)} \\
&\leq CN^{-(n-k-\mu_{(k=2)})}(N+1)^{-d/p} \|f\|_{W^{n,p}((0,1)^d)}.
\end{aligned}$$

We can now bound the sum from D.6 for each $\widetilde{m} \in \{0, \dots, N\}^d$ by

$$\begin{aligned}
&\left\| \sum_{m \in \{0,\dots,N\}^d} \phi_m^s (\widetilde{f} - p_m) \right\|_{W^{k,p}(\Omega_{\widetilde{m},N})} \\
&\leq CN^{-(n-k-\mu_{(k=2)})} \\
&\cdot \left(2(N+1)^{-d/p} \|f\|_{W^{n,p}((0,1)^d)} + \sum_{\substack{m \in \{0,\dots,N\}^d, \\ \|m - \widetilde{m}\|_\infty \leq 1}} \|\widetilde{f}\|_{W^{n,p}(\Omega_{m,N})} \right).
\end{aligned} \tag{D.7}$$

Consequently, we get

$$\begin{aligned}
&\sum_{\widetilde{m} \in \{0,\dots,N\}^d} \left\| \sum_{m \in \{0,\dots,N\}^d} \phi_m^s (\widetilde{f} - p_m) \right\|^p_{W^{k,p}(\Omega_{\widetilde{m},N})} \\
&\leq CN^{-(n-k-\mu_{(k=2)})p} \sum_{\widetilde{m} \in \{0,\dots,N\}^d} \Bigg(2(N+1)^{-d/p} \|f\|_{W^{n,p}((0,1)^d)} \\
&\qquad + \sum_{\substack{m \in \{0,\dots,N\}^d, \\ \|m - \widetilde{m}\|_\infty \leq 1}} \|\widetilde{f}\|_{W^{n,p}(\Omega_{m,N})} \Bigg)^p \\
&\leq CN^{-(n-k-\mu_{(k=2)})p}(3^d + 1)^{p/q}
\end{aligned}$$

$$
\begin{aligned}
&\cdot \left(\sum_{\widetilde{m}\in\{0,\ldots,N\}^d} 2^p (N+1)^{-d} \|f\|^p_{W^{n,p}((0,1)^d)} \right. \\
&\qquad \left. + \sum_{\widetilde{m}\in\{0,\ldots,N\}^d} \sum_{\substack{m\in\{0,\ldots,N\}^d,\\ \|m-\widetilde{m}\|_\infty\le 1}} \|\widetilde{f}\|^p_{W^{n,p}(\Omega_{m,N})} \right) \\
&\le CN^{-(n-k-\mu_{(k=2)})p} \left(\|f\|^p_{W^{n,p}((0,1)^d)} + 3^d \sum_{\widetilde{m}\in\{0,\ldots,N\}^d} \|\widetilde{f}\|^p_{W^{n,p}(\Omega_{\widetilde{m},N})} \right)
\end{aligned}
\tag{D.8}
$$

where the first step follows from plugging in Eq. (D.7), the second step follows from Hölder's inequality (with $q := 1 - 1/p$) and the last step follows from the definition of $\Omega_{\widetilde{m},N}$. Moreover, we use in the second and the last step the fact that the number of neighbors of a particular patch is bounded by $3^d - 1$. To conclude Step 4b we note that from the definition of $\Omega_{\widetilde{m},N}$ it follows that there exist 2^d disjoint subsets $\mathcal{M}_i \subset \{0,\ldots,N\}^d$ such that $\bigcup_{i=1,\ldots,2^d} \mathcal{M}_i = \{0,\ldots,N\}^d$ and $\Omega_{m_1,N} \cap \Omega_{m_2,N} = \varnothing$ for all $m_1, m_2 \in \mathcal{M}_i$ with $m_1 \neq m_2$ and all $i = 1,\ldots,2^d$. From this we get

$$
\begin{aligned}
\sum_{\widetilde{m}\in\{0,\ldots,N\}^d} \|\widetilde{f}\|^p_{W^{n,p}(\Omega_{\widetilde{m},N})} &= \sum_{i=1,\ldots,2^d} \sum_{\widetilde{m}\in\mathcal{M}_i} \|\widetilde{f}\|^p_{W^{n,p}(\Omega_{\widetilde{m},N})} \\
&\le 2^d \|\widetilde{f}\|^p_{W^{n,p}(\bigcup_{\widetilde{m}\in\{0,\ldots,N\}^d} \Omega_{\widetilde{m},N})}
\end{aligned}
\tag{D.9}
$$

and, finally, together with Remark B.3

$$
\begin{aligned}
\sum_{\widetilde{m}\in\{0,\ldots,N\}^d} \|\widetilde{f}\|^p_{W^{n,p}(\Omega_{\widetilde{m},N})} &\le 2^d \|\widetilde{f}\|^p_{W^{n,p}(\bigcup_{\widetilde{m}\in\{0,\ldots,N\}^d} \Omega_{\widetilde{m},N})} \\
&\le C\|f\|^p_{W^{n,p}((0,1)^d)}.
\end{aligned}
\tag{D.10}
$$

Step 4c (Wrap it All Up): Combining Eq. (D.8) with Eq. (D.10) from Step 4b and inserting it into Eq. (D.4) together with the estimate in Eq. (D.5) from Step 4a finally yields

$$
\|f - f_N\|_{W^{k,p}((0,1)^d)} \le CN^{-(n-k-\mu_{(k=2)})} \|f\|_{W^{n,p}((0,1)^d)},
$$

for all $N \ge \widetilde{N} := \max\{N_1, N_2\}$ and a constant $C = C(n,d,p,k) > 0$. The linearity of T_k, $k \in \{0,\ldots,j\}$ is a consequence of the linearity of the averaged Taylor polynomial (cf. Gühring et al. (2020, Remark B.8)). □

D.3. Approximation of localized polynomials by neural networks

The goal of this subsection is to demonstrate how to approximate sums of localized polynomials $\sum_p \phi_p \mathrm{poly}_p$ by neural networks. Corollary C.3 is the foundation for the following result which implements a neural network that approximates the multiplication of multiple inputs:

Lemma D.4. *Let $d, m, K \in \mathbb{N}$, $j \in \mathbb{N}_0$ and $N \ge 1$, $\mu \ge 0$, $c > 0$ be arbitrary, and let $\varrho \in W^{j,\infty}_{\mathrm{loc}}(\mathbb{R})$ fulfill the assumptions of Proposition 4.7 for $n = 3$, $r = 2$. Then there are constants $C(d,m,c,k) > 0$ such that the following holds:*

For any $\varepsilon \in (0, 1/2)$, and any neural network Φ with d-dimensional input and m-dimensional output and with number of layers and nonzero weights all bounded by K, such that

$$
\|[R_\varrho(\Phi)]_l\|_{W^{k,\infty}((0,1)^d)} \le cN^{k+\mu_{(k=2)}},
$$

for $k \in \{0,\ldots,j\}$, $l = 1,\ldots,m$ there exists a neural network $\Psi_{\varepsilon,\Phi}$ with d-dimensional input and one-dimensional output, and with

(i) number of layers and nonzero weights all bounded by CK;

(ii) $\|R_\varrho(\Psi_{\varepsilon,\Phi}) - \prod_{l=1}^n [R_\varrho(\Phi)]_l\|_{W^{k,\infty}((0,1)^d)} \le CN^{k+\mu_{(k=2)}}\varepsilon$;

(iii) $|R_\varrho(\Psi_{\varepsilon,\Phi})|_{W^{k,\infty}((0,1)^d)} \le CN^{k+\mu_{(k=2)}}$;

(iv) $\|\Psi_{\varepsilon,\Phi}\|_{\max} \le C\max\{\|\Phi\|_{\max}, \varepsilon^{-2}\}$.

Proof. We show by induction over $m \in \mathbb{N}$ that the statement holds. To make the induction argument easier we will additionally show that the network $\Psi_{\varepsilon,\Phi}$ can be chosen such that the first $L(\Phi) - 1$ layers of $\Psi_{\varepsilon,\Phi}$ and Φ coincide.

If $m = 1$, then we can choose $\Psi_{\varepsilon,\Phi} = \Phi$ for any $\varepsilon \in (0, 1/2)$ and the claim holds.

Now, assume that the claim holds for an arbitrary, but fixed $m \in \mathbb{N}$. We show that it also holds for $m+1$. For this, let $\varepsilon \in (0, 1/2)$ and let $\Phi = ((A_1, b_1), (A_2, b_2), \ldots, (A_L, b_L))$ be a neural network with d-dimensional input and $(m+1)$-dimensional output and with number of layers, and nonzero weights all bounded by K, where each A_l is an $N_l \times N_{l-1}$ matrix, and $b_l \in \mathbb{R}^{N_l}$ for $l = 1,\ldots,L$.

Step 1 (Invoking Induction Hypothesis): We denote by Φ_m the neural network with d-dimensional input and m-dimensional output which results from Φ by removing the last output neuron and corresponding weights. In detail, we write

$$
A_L = \begin{bmatrix} A_L^{(1,m)} \\ a_L^{(m+1)} \end{bmatrix} \quad \text{and} \quad b_L = \begin{bmatrix} b_L^{(1,m)} \\ b_L^{(m+1)} \end{bmatrix},
$$

where $A_L^{(1,m)}$ is a $m \times N_{L-1}$ matrix and $a_L^{(m+1)}$ is a $1 \times N_{L-1}$ vector, and $b_L^{(1,m)} \in \mathbb{R}^m$ and $b_L^{(m+1)} \in \mathbb{R}^1$. Now we set

$$
\Phi_m := \left((A_1, b_1), (A_2, b_2), \ldots, (A_{L-1}, b_{L-1}), \left(A_L^{(1,m)}, b_L^{(1,m)} \right) \right).
$$

Using the induction hypothesis we get that there is a neural network

$$
\Psi_{\varepsilon,\Phi_m} = ((A'_1, b'_1), (A'_2, b'_2), \ldots, (A'_{L'}, b'_{L'}))
$$

with d-dimensional input and one-dimensional output, and at most KC layers and nonzero weights such that

$$
\|R_\varrho(\Psi_{\varepsilon,\Phi_m}) - \prod_{l=1}^m [R_\varrho(\Phi_m)]_l\|_{W^{k,\infty}((0,1)^d)} \le CN^{k+\mu_{(k=2)}}\varepsilon,
$$

and $|R_\varrho(\Psi_{\varepsilon,\Phi_m})|_{W^{k,\infty}((0,1)^d)} \le CN^{k+\mu_{(k=2)}}$. Moreover, we have that $\|\Phi_m\|_{\max} \le \|\Phi\|_{\max}$, so that there we can estimate $\|\Psi_{\varepsilon,\Phi_m}\|_{\max} \le C\max\{\|\Phi\|_{\max}, \varepsilon^{-2}\}$. Furthermore, we can assume that the first $L-1$ layers of $\Psi_{\varepsilon,\Phi_m}$ and Φ_m coincide and, thus, also the first $L-1$ layers of $\Psi_{\varepsilon,\Phi_m}$ and Φ, i.e. $A_l = A'_l$ for $l = 1,\ldots,L-1$.

Step 2 (Combining $\Psi_{\varepsilon,\Phi_m}$ and $[R_\varrho(\Phi)]_{m+1}$): Now, we construct a network $\widetilde{\Psi}_{\varepsilon,\Phi}$ where the first $L-1$ layers of $\widetilde{\Psi}_{\varepsilon,\Phi}$ and $\Psi_{\varepsilon,\Phi_m}$ (and, thus, also of Φ) coincide (by definition), and $\widetilde{\Psi}_{\varepsilon,\Phi}$ has two-dimensional output with $\left[R_\varrho\left(\widetilde{\Psi}_{\varepsilon,\Phi}\right)\right]_1 = R_\varrho\left(\Psi_{\varepsilon,\Phi_m}\right)$ and $\left[R_\varrho\left(\widetilde{\Psi}_{\varepsilon,\Phi}\right)\right]_2 \approx \left[R_\varrho(\Phi)\right]_{m+1}$. For this, we add the formerly removed neuron with corresponding weights back to the Lth layer of $\Psi_{\varepsilon,\Phi_m}$ and approximately pass the output through to the last layer. Let $\Phi_\varepsilon^{L'-L+1,c,1} = ((A_1^{\mathrm{id}}, b_1^{\mathrm{id}}), \ldots, (A^{\mathrm{id}}_{L'-L+1}, b^{\mathrm{id}}_{L'-L+1}))$ be the network from Corollary C.4. We define

$$
\begin{aligned}
&\widetilde{\Psi}_{\varepsilon,\Phi} := \\
&\left((A'_i, b'_i)_{i=1}^{L-1}, \left(\begin{bmatrix} A'_L \\ A_1^{\mathrm{id}} a_L^{(m+1)} \end{bmatrix}, \begin{bmatrix} b'_L \\ A_1^{\mathrm{id}} b_L^{(m+1)} + b_1^{(m+1)} \end{bmatrix} \right), \right. \\
&\quad \left. \left(\begin{bmatrix} A'_{L+1} \\ A_2^{\mathrm{id}} \end{bmatrix}, \begin{bmatrix} b'_{L+1} \\ b_2^{\mathrm{id}} \end{bmatrix} \right), \left(\begin{bmatrix} A'_{L'} \\ A^{\mathrm{id}}_{L'-L+1} \end{bmatrix}, \begin{bmatrix} b'_{L'} \\ b^{\mathrm{id}}_{L'-L+1} \end{bmatrix} \right) \right).
\end{aligned}
$$

Counting the number of nonzero weights of $\widetilde{\Psi}_{\varepsilon,\Phi}$ we get with Lemma C.5(ii) that

$$
M(\widetilde{\Psi}_{\varepsilon,\Phi}) \le M(\Psi_{\varepsilon,\Phi_m}) + \underbrace{M(\Phi)}_{\text{from } a_L^{(m+1)}, b_L^{(m+1)}} + \underbrace{4(L'-L+1)}_{\text{from approximative identity}}
$$

$$\leq CK + K + CK \leq CK, \tag{D.11}$$

where we used in the second step the induction hypothesis twice together with the assumption on Φ. Similarly, we get the statement for $L(\widetilde{\Psi}_{\varepsilon,\Phi})$. Furthermore, $\|\widetilde{\Psi}_{\varepsilon,\Phi}\|_{\max} \leq C\max\{\|\Phi\|_{\max}, \varepsilon^{-2}\}$.

Next, we want to apply the approximate multiplication network from Corollary C.3 to the output of $\widetilde{\Psi}_{\varepsilon,\Phi}$. For this, we need to find a bounding box for the range of $R_\varrho\big(\widetilde{\Psi}_{\varepsilon,\Phi}\big)$. We have

$$\|R_\varrho(\Psi_{\varepsilon,\Phi_m})\|_{L^\infty((0,1)^d)} \leq C \quad \text{and}$$
$$\|[R_\varrho(\widetilde{\Psi}_{\varepsilon,\Phi})]_2\|_{L^\infty((0,1)^d)} \leq c + \varepsilon \leq c + 1,$$

and get for $B := \max\{C, c+1\}$ that $\operatorname{Range} R_\varrho(\widetilde{\Psi}_{\varepsilon,\Phi}) \subset [-B,B]^2$. Now, we denote by $\widetilde{\times}$ the network from Corollary C.3 with $B = B$ and accuracy ε and define

$$\Psi_{\varepsilon,\Phi} := \widetilde{\times} \bullet \widetilde{\Psi}_{\varepsilon,\Phi}.$$

Step 3 ($\Psi_{\varepsilon,\Phi}$ **Fulfills Induction Hypothesis for** $m+1$): **ad** (i): Clearly, $\Psi_{\varepsilon,\Phi}$ has d-dimensional input, one-dimensional output and, combining Eq. (D.11) with (iii) of Corollary C.3 as well as Lemma C.5.(iii), at most CK nonzero weights.

ad (ii): The first $L-1$ layers of $\Psi_{\varepsilon,\Phi}$ and Φ coincide and for the approximation properties it holds that

$$\begin{aligned}
&\left\| R_\varrho(\Psi_{\varepsilon,\Phi}) - \prod_{l=1}^{m+1}[R_\varrho(\Phi)]_l \right\|_{W^{k,\infty}((0,1)^d)} \\
&= \left\| R_\varrho(\widetilde{\times}) \circ R_\varrho\big(\widetilde{\Psi}_{\varepsilon,\Phi}\big) - [R_\varrho(\Phi)]_{m+1} \cdot \prod_{l=1}^{m}[R_\varrho(\Phi)]_l \right\|_{W^{k,\infty}((0,1)^d)} \\
&\leq \Bigg\| R_\varrho(\widetilde{\times}) \circ \big(R_\varrho(\Psi_{\varepsilon,\Phi_m}), [R_\varrho\big(\widetilde{\Psi}_{\varepsilon,\Phi}\big)]_2\big) \\
&\qquad - R_\varrho(\Psi_{\varepsilon,\Phi_m}) \cdot [R_\varrho\big(\widetilde{\Psi}_{\varepsilon,\Phi}\big)]_2 \Bigg\|_{W^{k,\infty}((0,1)^d)} \\
&\qquad + \left\| R_\varrho(\Psi_{\varepsilon,\Phi_m})\big([R_\varrho\big(\widetilde{\Psi}_{\varepsilon,\Phi}\big)]_2 - [R_\varrho(\Phi)]_{m+1}\big) \right\|_{W^{k,\infty}((0,1)^d)} \\
&\qquad + \left\| [R_\varrho(\Phi)]_{m+1} \cdot \Big(R_\varrho(\Psi_{\varepsilon,\Phi_m}) - \prod_{l=1}^{m}[R_\varrho(\Phi)]_l\Big) \right\|_{W^{k,\infty}((0,1)^d)}.
\end{aligned} \tag{D.12}$$

We continue by considering the first term of the Inequality (D.12) and bound the k-semi-norm of this term. We apply the chain rule from Corollary B.6 for $g : \mathbb{R}^2 \to \mathbb{R}$ with $g(x,y) = R_\varrho(\widetilde{\times})(x,y) - x\cdot y$ and $f : \mathbb{R}^d \to \mathbb{R}^2$ with $f = R_\varrho(\widetilde{\Psi}_{\varepsilon,\Phi})$. We get

$$\begin{aligned}
&\Big| R_\varrho(\widetilde{\times}) \circ \big(R_\varrho(\Psi_{\varepsilon,\Phi_m}), [R_\varrho\big(\widetilde{\Psi}_{\varepsilon,\Phi}\big)]_2\big) \\
&\qquad - R_\varrho(\Psi_{\varepsilon,\Phi_m}) \cdot [R_\varrho\big(\widetilde{\Psi}_{\varepsilon,\Phi}\big)]_2 \Big|_{W^{k,\infty}((0,1)^d)} \\
&\qquad \leq C \sum_{i=1}^{k} |R_\varrho(\widetilde{\times})(x,y) - x\cdot y|_{W^{i,\infty}((-B,B)^2;dxdy)} N^{k+\mu(k=2)} \\
&\qquad \leq Ck \cdot \|R_\varrho(\widetilde{\times})(x,y) - x\cdot y\|_{W^{j,\infty}((-B,B)^2;dxdy)} N^{k+\mu(k=2)} \\
&\qquad \leq C\varepsilon N^{k+\mu(k=2)},
\end{aligned} \tag{D.13}$$

where we used the induction hypothesis together with $|[R_\varrho\big(\widetilde{\Psi}_{\varepsilon,\Phi}\big)]_2|_{W^{k,\infty}((0,1)^d)} \leq cN^{k+\mu(k=2)}$ (which follows from the properties of the approximate identity network from Corollary C.4 together with the chain rule) in the third step and assumed that $c \leq C$. Combining the statements of the semi-norms then yields the required bound for the norm. For the second term we have by the product rule and the chain rule

$$\begin{aligned}
&\left\| R_\varrho(\Psi_{\varepsilon,\Phi_m})\big([R_\varrho\big(\widetilde{\Psi}_{\varepsilon,\Phi}\big)]_2 - [R_\varrho(\Phi)]_{m+1}\big) \right\|_{W^{k,\infty}((0,1)^d)} \\
&\leq \sum_{i=0}^{k} \|R_\varrho(\Psi_{\varepsilon,\Phi_m})\|_{W^{i,\infty}((0,1)^d)} \\
&\qquad \cdot \left\| [R_\varrho\big(\widetilde{\Psi}_{\varepsilon,\Phi}\big)]_2 - [R_\varrho(\Phi)]_{m+1} \right\|_{W^{k-i,\infty}((0,1)^d)} \\
&\leq \sum_{i=0}^{k} cN^{i+\mu(i=2)} \cdot C\varepsilon N^{k-i+\mu(k-i=2)} \leq kcCN^{k+\mu(k=2)}\varepsilon.
\end{aligned} \tag{D.14}$$

To estimate the last term of (D.12) we apply the product rule from Lemma B.5 and get

$$\begin{aligned}
&\left\| [R_\varrho(\Phi)]_{m+1} \cdot \Big(R_\varrho(\Psi_{\varepsilon,\Phi_m}) - \prod_{l=1}^{m}[R_\varrho(\Phi)]_l\Big) \right\|_{W^{k,\infty}((0,1)^d)} \\
&\leq \sum_{i=0}^{k} \|[R_\varrho(\Phi)]_{m+1}\|_{W^{i,\infty}((0,1)^d)} \\
&\qquad \cdot \left\| R_\varrho(\Psi_{\varepsilon,\Phi_m}) - \prod_{l=1}^{m}[R_\varrho(\Phi)]_l \right\|_{W^{k-i,\infty}((0,1)^d)} \\
&\leq \sum_{i=0}^{k} cN^{i+\mu(i=2)} \cdot CN^{k-i+\mu(k-i=2)}\varepsilon \leq kcCN^{k+\mu(k=2)}\varepsilon.
\end{aligned} \tag{D.15}$$

For the second step, we used again the induction hypothesis together with

$$|[R_\varrho(\Phi)]_{m+1}|_{W^{k,\infty}((0,1)^d)} \leq cN^{k+\mu(k=2)}.$$

Combining (D.12) with (D.13), (D.14) and (D.15) yields

$$\left\| R_\varrho(\Psi_{\varepsilon,\Phi}) - \prod_{l=1}^{m+1}[R_\varrho(\Phi)]_l \right\|_{W^{k,\infty}((0,1)^d)} \leq CN^{k+\mu(k=2)}\varepsilon.$$

ad (iii): The estimate

$$|R_\varrho(\Psi_{\varepsilon,\Phi})|_{W^{k,\infty}((0,1)^d)} \leq CN^{k+\mu(k=2)}.$$

can be shown similarly as above.

ad (iv): Finally, we need to derive a bound for the absolute values of the weights. From the definition of $\Psi_{\varepsilon,\Phi}$ together with Lemma C.5.(iii) we get

$$\|\Psi_{\varepsilon,\Phi}\|_{\max} = \|\widetilde{\times} \bullet \widetilde{\Psi}_{\varepsilon,\Phi}\|_{\max} \leq C \cdot \max\{\varepsilon^{-2}, \|\widetilde{\Psi}_{\varepsilon,\Phi}\|_{\max}\}.$$

From $\|\widetilde{\Psi}_{\varepsilon,\Phi}\|_{\max} \leq C\max\{\|\Phi\|_{\max}, \varepsilon^{-2}\}$ (see Step 2) it follows that $\|\Psi_{\varepsilon,\Phi}\|_{\max} \leq C\max\{\|\Phi\|_{\max}, \varepsilon^{-2}\}$. This concludes the proof. □

In the last part of this subsection, we are finally in a position to construct neural networks which approximate sums of localized polynomials.

Lemma D.5. *Let $j, \tau \in \mathbb{N}_0$, $d, N \in \mathbb{N}$, $k \in \{0,\ldots,j\}$, Additionally, let ϱ be such that it fulfills the assumptions of Proposition 4.7 (for $n = 3$, $r = 2$). Let $n \in \mathbb{N}_{\geq k+1}$, $1 \leq p \leq \infty$, and $\mu > 0$. Assume that $\big(\Psi^{(j,\tau,N,s)}\big)_{N\in\mathbb{N}, s\geq 1}$ be the exponential (polynomial, exact) (j,τ)-PU from Definition 4.1. For $N \in \mathbb{N}$, set*

$$s := \begin{cases} N^{\mu}, & \text{if exponential PU,} \\ N^{\frac{2d/p+d+n}{D}}, & \text{if polynomial PU,} \\ 1, & \text{if exact PU,} \end{cases}$$

Then, there is a constant $C = C(n,d,p,k) > 0$ with the following properties:

Let $\varepsilon \in (0, 1/2)$, $f \in W^{n,p}((0,1)^d)$ and $p_m(x) := p_{f,m}(x) = \sum_{|\alpha|\leq n-1} c_{f,m,\alpha}x^\alpha$ for $m \in \{0,\ldots,N\}^d$ be the polynomials from Lemma D.1. Then there is a neural network $\Phi_{P,\varepsilon} = \Phi_{P,\varepsilon}(f, d, n, N, \mu,$

ε) with d-dimensional input and one-dimensional output, with at most C layers and $C(N+1)^d$ nonzero weights, such that

$$\Big\| \sum_{m\in\{0,\dots,N\}^d} \phi_m^s p_m - R_\varrho(\Phi_{P,\varepsilon}) \Big\|_{W^{k,p}((0,1)^d)} \le C\|f\|_{W^{n,p}((0,1)^d)}\varepsilon,$$

and $\|\Phi_{P,\varepsilon}\|_{\max} \le C\|f\|_{W^{n,p}((0,1)^d)}\varepsilon^{-2}s^2N^{2(d/p+d+k)+d/p+d}$.

Proof. As before, we only provide the proof only for the case of an exponential (j,τ)-PU.

Step 1 (Approximating Localized Monomials $\phi_m^s(x)x^\alpha$): Let $|\alpha| \le n-1$, $m \in \{0,\dots,N\}^d$ and set $\widetilde{\varepsilon} := \varepsilon N^{-(d/p+d+k+\mu_{(k=2)})}$. By Corollary C.4 and inductively using the trick that $|xy - uz| \le |x(y-z)| + |z(x-u)|$, there is a neural network Φ_α with d-dimensional input and $|\alpha|$-dimensional output, with two layers, at most $4(n-1)$ nonzero weights bounded in absolute value by $C\widetilde{\varepsilon}^{-1}$ such that

$$\Big\| x^\alpha - \prod_{l=1}^{|\alpha|}[R_\varrho(\Phi_\alpha)]_l(x) \Big\|_{W^{k,\infty}((0,1)^d;dx)} \le C\widetilde{\varepsilon} \tag{D.16}$$

and

$$\|[R_\varrho(\Phi_\alpha)]_l\|_{W^{k,\infty}((0,1)^d)} \le \widetilde{\varepsilon} + 1 \le 2, \quad \text{for all } l = 1,\dots,|\alpha|. \tag{D.17}$$

Let now Φ_m be the neural network from Lemma 4.5(iv) (for $s = N^\mu$) and define the network

$$\Phi_{m,\alpha} := \mathrm{P}(\Phi_m, \Phi_\alpha, \Phi_{n-1-|\alpha|,2}),$$

where the parallelization is provided by Lemma C.2 and $\Phi_{n-1-|\alpha|,2} = \big((0_{d,d}, 0_d), (0_{n-1-|\alpha|,d}, 1_{n-1-|\alpha|})\big)$. Consequently, $\Phi_{m,\alpha}$ has $2 \le K_0$ layers and $C + 4(n-1) \le K_0$ nonzero weights for a suitable constant $K_0 = K_0(n,d) \in \mathbb{N}$, $\|\Phi_{m,\alpha}\|_{\max} \le C\max\{\widetilde{\varepsilon}^{-1}, N^{1+\mu}\}$ and $\|\prod_{l=1}^{n-1+d}[R_\varrho(\Phi_{m,\alpha})(x)]_l - \phi_m^s(x)x^\alpha\|_{W^{k,\infty}((0,1)^d);dx} \le C\widetilde{\varepsilon}$. Moreover, as a consequence of Lemma 4.5(iv) together with Eq. (D.17) we have

$$\|[R_\varrho(\Phi_{m,\alpha})]_l\|_{W^{k,\infty}((0,1)^d)} \le CN^{k+\mu_{(k=2)}},$$
$$\text{for all } l = 1,\dots,n-1+d.$$

To construct an approximation of the localized monomials $\phi_m^s(x)x^\alpha$, let $\Psi_{\widetilde{\varepsilon},(m,\alpha)}$ be the neural network provided by Lemma D.4 (with $\Phi_{m,\alpha}$ instead of Φ, $m = |\alpha| + d \in \mathbb{N}$, $K = K_0 \in \mathbb{N}$) for $m \in \{0,\dots,N\}^d$ and $\alpha \in \mathbb{N}_0^d$, $|\alpha| \le n-1$. Then $\Psi_{\widetilde{\varepsilon},(m,\alpha)}$ has at most C layers (independently of m,α), number of nonzero weights and $\|\Psi_{\widetilde{\varepsilon},(m,\alpha)}\|_{\max} \le C\max\{N^{1+\mu}, \varepsilon^{-2}N^{2(d/p+d+k+\mu_{(k=2)})}\}$. Moreover,

$$\begin{aligned}
&\|\phi_m^s(x)x^\alpha - R_\varrho\big(\Psi_{\widetilde{\varepsilon},(m,\alpha)}\big)(x)\|_{W^{k,\infty}((0,1)^d;dx)} \\
&\le \|\phi_m^s(x)x^\alpha - \prod_{l=1}^{n-1+d}[R_\varrho(\Phi_{m,\alpha})(x)]_l\|_{W^{k,\infty}((0,1)^d;dx)} \\
&\quad + \|\prod_{l=1}^{n-1+d}[R_\varrho(\Phi_{m,\alpha})]_l - R_\varrho\big(\Psi_{\widetilde{\varepsilon},(m,\alpha)}\big)\|_{W^{k,\infty}((0,1)^d)} \\
&\le CN^{k+\mu_{(k=2)}}\widetilde{\varepsilon} \le C\varepsilon N^{-d/p-d},
\end{aligned}$$

where we used Eq. (D.16) together with the product rule for the last step.

Step 2 (Constructing $\Phi_{P,\varepsilon}$): We set

$$T := |\{(m,\alpha) : m \in \{0,\dots,N\}^d, \alpha \in \mathbb{N}_0^d, |\alpha| \le n-1\}|.$$

We note that every network $\Psi_{\widetilde{\varepsilon},(m,\alpha)}$ has the same number of layers and, by using Lemma C.2, we parallelize the localized polynomial approximations

$$\mathrm{P}\big(\Psi_{\widetilde{\varepsilon},(m,\alpha)} : m \in \{0,\dots,N\}^d, \alpha \in \mathbb{N}_0^d,\ |\alpha| \le n-1\big)$$

and note that the resulting network has at most C layers and CT nonzero weights bounded in absolute value by $C\max\{N^{1+\mu}, \varepsilon^{-2} N^{2(d/p+d+k+\mu_{(k=2)})}\} \le C\varepsilon^{-2}N^{2(d/p+d+k+\mu_{(k=2)})}$. Next, we define the matrix $A_{\mathrm{sum}} \in \mathbb{R}^{1,T}$ by $A_{\mathrm{sum}} := [c_{f,m,\alpha} : m \in \{0,\dots,N\}^d, \alpha \in \mathbb{N}_0^d, |\alpha| \le n-1]$ and the neural network $\Phi_{\mathrm{sum}} := ((A_{\mathrm{sum}}, 0))$. Finally, we set

$$\Phi_{P,\varepsilon} := \Phi_{\mathrm{sum}} \bullet \mathrm{P}\big(\Psi_{\widetilde{\varepsilon},(m,\alpha)} : m \in \{0,\dots,N\}^d, \alpha \in \mathbb{N}_0^d, |\alpha| \le n-1\big). \tag{D.18}$$

From Lemma C.5(i) we get $\Phi_{P,\varepsilon}$ is a neural network with d-dimensional input and one-dimensional output, with at most C layers and, by Lemma C.5, $CT \le C(N+1)^d$ nonzero weights. For the absolute values of the weights it holds that

$$\begin{aligned}
\|\Phi_{P,\varepsilon}\|_{\max} &\le (N+1)^d C\|f\|_{W^{n,p}((0,1)^d)}N^{d/p}\varepsilon^{-2}N^{2(d/p+d+k+\mu_{(k=2)})} \\
&\le C\|f\|_{W^{n,p}((0,1)^d)}\varepsilon^{-2}N^{2(d/p+d+k+\mu_{(k=2)})+d/p+d}
\end{aligned}$$

where we used the bound for the coefficients $c_{f,m,\alpha}$ from Remark D.2. Moreover, we have

$$R_\varrho(\Phi_{P,\varepsilon}) = \sum_{m\in\{0,\dots,N\}^d}\ \sum_{|\alpha|\le n-1} c_{f,m,\alpha}R_\varrho(\Psi_{\widetilde{\varepsilon},(m,\alpha)}).$$

Note that the network $\Phi_{P,\varepsilon}$ only depends on $p_{f,m}$ (and thus on f) via the coefficients $c_{f,m,\alpha}$.

Step 3 (Estimating the approximation error in $\|\cdot\|_{W^{k,p}}$**):** We get

$$\begin{aligned}
&\Big\| \sum_{m\in\{0,\dots,N\}^d} \phi_m^s(x)p_m(x) - R_\varrho(\Phi_{P,\varepsilon})(x) \Big\|_{W^{k,p}((0,1)^d;dx)} \\
&= \Big\| \sum_{\substack{m\in\{0,\dots,N\}^d \\ |\alpha|\le n-1}} c_{f,m,\alpha}\Big(\phi_m^s(x)x^\alpha - R_\varrho\big(\Psi_{\widetilde{\varepsilon},(m,\alpha)}\big)(x)\Big) \Big\|_{W^{k,p}((0,1)^d;dx)} \\
&\le \sum_{\substack{m\in\{0,\dots,N\}^d \\ |\alpha|\le n-1}} |c_{f,m,\alpha}|\|\phi_m^s(x)x^\alpha - R_\varrho\big(\Psi_{\widetilde{\varepsilon},(m,\alpha)}\big)(x)\|_{W^{k,p}((0,1)^d;dx)} \\
&\le \sum_{\substack{m\in\{0,\dots,N\}^d \\ |\alpha|\le n-1}} \|\widetilde{f}\|_{W^{n-1,p}(\Omega_{m,N})}N^{d/p}C\varepsilon N^{-d/p-d},
\end{aligned}$$

where we used again the bound for the coefficients $c_{f,m,\alpha}$ together with $\|\cdot\|_{W^{k,p}((0,1)^d)} \le C\|\cdot\|_{W^{k,\infty}((0,1)^d)}$ in the last step. Similar as in Eq. (D.9) we finally have

$$\begin{aligned}
&\Big\| \sum_{m\in\{0,\dots,N\}^d} \phi_m^s(x)p_m(x) - R_\varrho(\Phi_{P,\varepsilon})(x) \Big\|_{W^{k,p}((0,1)^d;dx)} \\
&\le C\varepsilon N^{-d} \sum_{m\in\{0,\dots,N\}^d} \|\widetilde{f}\|_{W^{n,p}((0,1)^d)} \\
&\le C\varepsilon\|f\|_{W^{n,p}((0,1)^d)}.
\end{aligned}$$

This concludes the proof. □

D.4. Putting everything together

Now we conclude the proof of Proposition 4.8. Again, we only provide the proof for exponential (j,τ)-PUs. The rest follows in a similar manner by adapting the calculations to come accordingly.

Proof of Proposition 4.8. We divide the proof into two steps: First, we approximate the function f by a sum of localized polynomials. Afterwards, we proceed by approximating this sum by a neural network.

For the first step, we set

$$N := \left\lceil \left(\frac{\varepsilon}{2\widetilde{C}}\right)^{-1/(n-k-\mu_{(k=2)})} \right\rceil \quad \text{and} \quad s := N^{\mu}, \tag{D.19}$$

where $\widetilde{C} = \widetilde{C}(n,d,p,k) > 0$ is the constant from Lemma D.1. Without loss of generality we may assume that $\widetilde{C} \geq 1$. The same lemma yields that if $\Psi^{(j,\tau)} = \Psi^{(j,\tau)}(d,N,\mu) = \left\{\phi_m^s : m \in \{0,\dots,N\}^d\right\}$ is the PU from Lemma 4.5 and $\widetilde{N} = \widetilde{N}(d,p,\mu,k)$ is the constant from Lemma D.1, then there exist polynomials $p_m(x) = \sum_{|\alpha|\leq n-1} c_{f,m,\alpha} x^{\alpha}$ for $m \in \{0,\dots,N\}^d$ such that

$$\begin{aligned}\left\| f - \sum_{m\in\{0,\dots,N\}^d} \phi_m^s p_m \right\|_{W^{k,p}((0,1)^d)} &\leq \widetilde{C}\left(\frac{1}{N}\right)^{n-k-\mu_{(k=2)}} \\ &\leq \widetilde{C}\frac{\varepsilon}{2\widetilde{C}} = \frac{\varepsilon}{2},\end{aligned} \tag{D.20}$$

for all $\varepsilon \in (0,\widetilde{\varepsilon})$, where $\widetilde{\varepsilon} = \widetilde{\varepsilon}(d,p,\mu,k) > 0$ is chosen such that $N \geq \widetilde{N}$.

For the second step, let $\widetilde{C}' = \widetilde{C}'(n,d,p,k)$ be the constant from Lemma D.5 and $\Phi_{P,\varepsilon}$ be the neural network provided by Lemma D.5 with $\varepsilon/(2\widetilde{C}')$ instead of ε. Then $\Phi_{P,\varepsilon}$ has at most $\widetilde{C}'$ layers and at most

$$\begin{aligned}\widetilde{C}'\left(\left(\frac{\varepsilon}{2\widetilde{C}'}\right)^{-1/(n-k-\mu_{(k=2)})} + 2\right)^d &\leq \widetilde{C}' 3^d \left(\frac{\varepsilon}{2\widetilde{C}'}\right)^{-d/(n-k-\mu_{(k=2)})} \\ &\leq C\varepsilon^{-d/(n-k-\mu_{(k=2)})}\end{aligned}$$

nonzero weights. In the first step we have used $(2\widetilde{C}')/\varepsilon \geq 1$. The weights are bounded in absolute value by

$$\begin{aligned}\|\Phi_{P,\varepsilon}\|_{\max} &\leq \widetilde{C}'\varepsilon^{-2} N^{2(d/p+dk+\mu_{(k=2)})+d/p+d} \\ &\leq C\varepsilon^{-2-(2(d/p+d+k+\mu_{(k=2)})+d/p+d)/(n-k-\mu_{(k=2)})} = C\varepsilon^{-\theta},\end{aligned}$$

for a suitable $\theta = \theta(d,p,k,n,\mu) > 0$. Additionally, there holds

$$\left\| \sum_{m\in\{0,\dots,N\}^d} \phi_m^s p_m - R_{\varrho}(\Phi_{P,\varepsilon}) \right\|_{W^{k,p}((0,1)^d)} \leq \widetilde{C}'\frac{\varepsilon}{2\widetilde{C}'} \leq \frac{\varepsilon}{2}. \tag{D.21}$$

By applying the triangle inequality as well as Eqs. (D.20) and (D.21) we arrive at

$$\begin{aligned}&\|f - R_{\varrho}(\Phi_{P,\varepsilon})\|_{W^{k,p}((0,1)^d)} \\ &\quad\leq \left\| f - \sum_{m\in\{0,\dots,N\}^d} \phi_m^s p_m \right\|_{W^{k,p}((0,1)^d)} \\ &\qquad + \left\| \sum_{m\in\{0,\dots,N\}^d} \phi_m^s p_m - R_{\varrho}(\Phi_{P,\varepsilon}) \right\|_{W^{k,p}((0,1)^d)} \\ &\quad\leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon,\end{aligned}$$

thereby concluding the proof. □

Appendix E. Proof of Theorem 4.9 (encodability of the weights)

We now proceed with the proof of Theorem 4.9.

Proof of Theorem 4.9. Let $C = C(d,n,p,\mu,k) > 0$, $\theta = \theta(d,n,p,k,\mu) > 0$ and $\widetilde{\varepsilon} = \widetilde{\varepsilon}(d,p,\mu,k) > 0$ be the constants from Proposition 4.8 and let $\varepsilon \in (0, \min\{1/3, \widetilde{\varepsilon}\})$. Moreover, for $f \in \mathcal{F}_{n,d,p}$, let

$$\Phi_{\varepsilon,f} := ((A_{\text{sum}}, 0)) \bullet \mathrm{P}\big(\Psi_i : i = 1,\dots,T\big)$$

be the neural network from Proposition 4.8 (defined in Eq. (D.18)) with at most L layers and $M(\Phi_{\varepsilon,f}) \leq C \cdot \varepsilon^{-d/(n-k-\mu_{(k=2)})}$ nonzero weights bounded in absolute value by $C\varepsilon^{-\theta}$, such that

$$\|R_{\varrho}(\Phi_{\varepsilon,f}) - f\|_{W^{k,p}((0,1)^d)} \leq \frac{\varepsilon}{3}.$$

We will make use of the following additional properties of $\Phi_{\varepsilon,f}$:

(i) Only the entries of A_{sum} depend on the function f. In other words, the entries of $\Psi_1,\dots,\Psi_T$ are independent from f. They only depend on $\varepsilon, n, d, p, k, \mu$.

(ii) There exists $s = s(k,n,d,p) > 0$ (we assume w.l.o.g. that the same s can be used) such that

   (a) $\|R_{\varrho}(\Psi_i)\|_{W^{k,\infty}((0,1)^d)} \leq \varepsilon^{-s}$ for $i = 1,\dots,T$. This follows from Lemma D.4(iii) in combination with Step 1 and 2 of the proof of Lemma D.5 and choice of N in Eq. (D.19).

   (b) $T \leq \varepsilon^{-s}$. This follows from the definition of T (see Step 2 of the proof of Lemma D.5);

   (c) $M(\Phi_{\varepsilon,f}) \leq \varepsilon^{-s}$.

(iii) $A_{\text{sum}} = (a_m)_{m=1}^T \in \mathbb{R}^{1,T}$.

(iv) The last layer $(A_{\text{last}}, b_{\text{last}})$ of $\mathrm{P}\big(\Psi_i : i = 1,\dots,T\big)$ has a block diagonal structure, where each block is a vector (see also Lemma C.2). Thus, in every column of A_{last} there is at most one nonzero entry.

We replace the weights in the last layer of $\Phi_{\varepsilon,f}$ by elements from an appropriate set of weights with cardinality bounded polynomially in ε^{-1} and show that the resulting network is still close enough to f. Afterwards, we construct a coding scheme for the entire set of weights.

Step 1 (Rounding the Weights in A_{sum}**)**: We now show that with rounding precision $\nu := 2s{+}2$ we have for the neural network

$$\widetilde{\Phi}^{(1)}_{\varepsilon,f} := ((\widetilde{A}_{\text{sum}}, 0)) \bullet \mathrm{P}\big(\Psi_i : i = 1,\dots,T\big)$$

where $\widetilde{A}_{\text{sum}} \in ([-\varepsilon^{-\theta}, \varepsilon^{-\theta}] \cap \varepsilon^{\nu}\mathbb{Z})^{1,T}$ is the rounded weight matrix $A_{\text{sum}} \in \mathbb{R}^{1,T}$ that

$$\|R_{\varrho}(\Phi_{\varepsilon,f}) - R_{\varrho}(\widetilde{\Phi}^{(1)}_{\varepsilon,f})\|_{W^{k,p}((0,1)^d)} \leq \varepsilon/3.$$

Clearly,

$$\begin{aligned}&\Big\| R_{\varrho}\Big((A_{\text{sum}},0) \bullet \mathrm{P}\big(\Psi_i : i = 1,\dots,T\big)\Big) \\ &\quad - R_{\varrho}\Big((\widetilde{A}_{\text{sum}},0) \bullet \mathrm{P}\big(\Psi_i : i = 1,\dots,T\big)\Big)\Big\|_{W^{k,p}((0,1)^d)} \\ &\leq \left\| \sum_{i=1}^T a_i R_{\varrho}(\Psi_i) - \sum_{i=1}^T \widetilde{a}_i R_{\varrho}(\Psi_i) \right\|_{W^{k,\infty}((0,1)^d)} \\ &\leq \sum_{i=1}^T |a_i - \widetilde{a}_i| \|R_{\varrho}(\Psi_i)\|_{W^{k,\infty}((0,1)^d)} \\ \text{(rounding precision is } \varepsilon^{\nu}\text{)} \quad &\leq \sum_{i=1}^T \varepsilon^{\nu} \|R_{\varrho}(\Psi_i)\|_{W^{k,\infty}((0,1)^d)} \\ \text{((ii) and (iii) above)} \quad &\leq \varepsilon^{\nu}\varepsilon^{-s}\varepsilon^{-s} \leq \varepsilon^2 \leq \varepsilon/3.\end{aligned}$$

To get our final network, we replace the bias term $\widetilde{A}_{\text{sum}} b_{\text{last}}$ (which is also bounded in absolute value by $\varepsilon^{-\theta}$) in the last layer of $\widetilde{\Phi}^{(1)}_{\varepsilon,f}$ by the nearest element in $[-\varepsilon^{-\theta}, \varepsilon^{-\theta}] \cap \varepsilon^{\nu}\mathbb{Z}$ and denote the resulting network by $\widetilde{\Phi}_{\varepsilon,f}$. It now easily follows that $\|R_{\varrho}(\widetilde{\Phi}^{(1)}_{\varepsilon,f}) - R_{\varrho}(\widetilde{\Phi}_{\varepsilon,f})\|_{W^{k,\infty}((0,1)^d)} \leq \varepsilon/3$ which implies by the triangle inequality that $\|f - R_{\varrho}(\widetilde{\Phi}_{\varepsilon,f})\|_{W^{k,p}((0,1)^d)} \leq \varepsilon$.

Step 2 (Construction of coding scheme): We will now show that there is a constant $C_2 = C_2(d,n,p,k,\mu) > 0$ and a coding scheme $\mathcal{B} = (B_{\ell})_{\ell\in\mathbb{N}}$ such that for each $\varepsilon > 0$ and each $f \in \mathcal{F}_{n,d,p}$ the nonzero weights of $\widetilde{\Phi}_{\varepsilon,f}$ are in $\operatorname{Range} B_{\lceil C_2 \log(1/\varepsilon)\rceil}$.

If we denote by W_ε the collection of nonzero weights of $(\Psi_m)_{m=1}^{T}$ (which are independent of f), then we have $|W_\varepsilon| \le M(\Phi_{\varepsilon,f}) \le \varepsilon^{-s}$. Furthermore, we have $|[-\varepsilon^{-\theta}, \varepsilon^{-\theta}] \cap \varepsilon^{\nu}\mathbb{Z}| = 2\lfloor \varepsilon^{-\theta-\nu} \rfloor + 1 \le \varepsilon^{-s_2}$ with $s_2 := \theta + \nu + 2$.

- The matrix weights in the last layer of $\widetilde{\Phi}_{\varepsilon,f}$ are in the set $G_{\mathrm{mult}} := \{x_1 x_2 : x_1 \in W_\varepsilon, x_2 \in [-\varepsilon^{-\theta}, \varepsilon^{-\theta}] \cap \varepsilon^{\nu}\mathbb{Z}\}$ with cardinality bounded by $\varepsilon^{-(s+s_2)}$.
- The bias in the last layer is an element of $[-\varepsilon^{-\theta}, \varepsilon^{-\theta}] \cap \varepsilon^{\nu}\mathbb{Z}$.
- The weights of $\widetilde{\Phi}_{\varepsilon,f}$ in the layers $1, \ldots, L-1$ are in the set W_ε.

Setting $C_2 := 2(s + s_2)$ it follows that there exists a surjective mapping

$$B_{\lceil C_2 \log_2(1/\varepsilon)\rceil} : \\ \{0,1\}^{\lceil C_2 \log_2(1/\varepsilon)\rceil} \to G_{\mathrm{mult}} \cup W_\varepsilon \cup \left([-\varepsilon^{-\theta}, \varepsilon^{-\theta}] \cap \varepsilon^{\nu}\mathbb{Z}\right),$$

which shows the claim. □

Appendix F. PU-properties of the activation functions from Table 1

In this section, we examine the PU-properties of the activation functions listed in Table 1. The smoothness properties of all functions in Table 1 are clear. In particular, all functions are in $C^\infty(\mathbb{R} \setminus \{0\})$. In order to show that the activation functions to follow allow for exponential (polynomial) PUs, we consider the exponential (polynomial) (j, τ) admissibility conditions of Definition 4.2.

Exact PUs.

(leaky) ReLU and RePUs: These functions admit exact PUs. For the ReLU case, see for instance Yarotsky (2017) and Gühring et al. (2020). For RePUs, this follows from the properties of B-splines (see De Boor (2001, Chapter IX)).

Exponential PUs.

ELU$_a$ for $a > 0, a \neq 1$: Here, $j = 1$, $\tau = 1$, $A = 0$ and $B = 1$. Moreover, $R > 0$ can be chosen arbitrarily. Then, for $D = 1$, we have, for all $x > R$, that $|1 - \varrho'(x)| = |1 - 1| = 0$ and, for all $x < -R$ that $|\varrho'(x)| = |ae^x| = ae^{Dx}$.

ELU$_1$: Here, $j = 2$, $\tau = 1$, $A = 0$ and $B = 1$. Moreover, $R > 0$ can be chosen arbitrarily. Then, for $D = 1$, we have, for all $x > R$, that $|1 - \varrho'(x)| = |1 - 1| = 0$ and, for all $x < -R$ that $|\varrho'(x)| = |e^x| = e^{Dx}$. Moreover, we have for all $|x| > R$ that $|\varrho''(x)| \le e^{-|x|} = e^{-D|x|}$.

sigmoid: Here, $j \in \mathbb{N}_0$ is arbitrary, $\tau = 0$, $A = 0$ and $B = 1$. Moreover, $R > 0$ can be chosen arbitrarily. Then we have, for all $x > R$, that $|1 - \varrho(x)| \le e^{-x}$ and, for all $x < -R$ that $|\varrho(x)| \le e^x$. The other statements follow from the fact that, for the sigmoid activation function, the kth derivative is a finite linear combination of the powers $\varrho, \ldots, \varrho^k$ of ϱ (see, e.g., Minai and Williams (1993)). Choosing D suitably then shows the claim.

tanh: Since $\tanh(x) = 2 \cdot \mathrm{sigmoid}(2x) - 1$, the proof of this statement follows from the proof of the statement for the sigmoid activation function for $A = -1$, $B = 1$.

softplus: Here, $j \in \mathbb{N}_0$ is arbitrary, $\tau = 1$, $A = 0$ and $B = 1$. Moreover, $R > 0$ can be chosen arbitrarily. Then, for all $x > R$, there holds $|1 - \varrho'(x)| = |1 - \mathrm{sigmoid}(x)| \le e^{-x}$ and, for all $x < -R$ that $|\varrho'(x)| = |\mathrm{sigmoid}(x)| \le e^x$. The proof of (d.3) for the higher-order derivatives follows from the properties of the higher derivatives of the sigmoid function.

swish: Here, $j \in \mathbb{N}_0$ is arbitrary, $\tau = 1$, $A = 0$, and $B = 1$. It is not hard to see that for all $k \in \mathbb{N}$ there holds

$$\mathrm{swish}^{(k)}(x) = x \cdot \mathrm{sigmoid}^{(k)}(x) + k \cdot \mathrm{sigmoid}^{(k-1)}(x).$$

Now, the statement follows from the analogous observations for the sigmoid function combined with the fact that for $r, u > 0$ with $r > u$ there holds

$$\lim_{x\to\infty} \frac{xe^{-rx}}{e^{-ux}} = 0, \qquad \lim_{x\to-\infty} \frac{xe^{rx}}{e^{ux}} = 0.$$

Polynomial PUs.

softsign: Here, $j \in \mathbb{N}_0$ is arbitrary, $\tau = 0$, $A = -1$, $B = 1$. The polynomial convergence properties (d.1)–(d.3) follow immediately from the definition of the softsign function.

inverse square root linear unit: Here, $j = 3$, $\tau = 1$, $A = 0$ and $B = 1$. The polynomial convergence properties (d.1)–(d.3) follow immediately from the definition of the inverse square root linear unit.

inverse square root unit: Here, $j \in \mathbb{N}_0$ is arbitrary, $\tau = 0$, $A = -1$ and $B = 1$. The polynomial convergence properties (d.1)–(d.3) follow immediately from the definition of the inverse square root unit.

arctan: Here, $j \in \mathbb{N}_0$ is arbitrary, $\tau = 0$, $A = -\pi/2$ and $B = \pi/2$. The polynomial convergence properties (d.1)–(d.3) follow immediately from the fact that $\varrho'(x) = 1/(1 + x^2)$ which in particular implies polynomial convergence behavior for arctan itself.

References

Adams, R. (1975). *Sobolev spaces*. New York: Academic Press.

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

Beck, C., Becker, S., Grohs, P., Jaafari, N., & Jentzen, A. (2018). Solving stochastic differential equations and Kolmogorov equations by means of deep learning. arXiv preprint arXiv:1806.00421.

Beck, C., E., W., & Jentzen, A. (2019). Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations. *Journal of Nonlinear Science*, *29*, 1563–1619.

Bölcskei, H., Grohs, P., Kutyniok, G., & Petersen, P. C. (2019). Optimal approximation with sparsely connected deep neural networks. *SIAM Journal on Mathematical Data Science*, *1*(1), 8–45.

Brenner, S., & Scott, R. (2008). *Texts in applied mathematics*: *Vol. 15*, *The mathematical theory of finite element methods* (3rd ed.). New York: Springer Science+Business Media.

Bressan, A. (2012). Lecture notes on functional analysis: with applications to linear partial differential equations, vol. 143. In *Graduate studies in mathematics*: *Vol. 143*, American Mathematical Society.

Ciarlet, P. G. (2002). *The finite element method for elliptic problems*. Society for Industrial and Applied Mathematics.

Constantine, G., & Savits, T. H. (1996). A multivariate faa di bruno formula with applications. *Transactions of the American Mathematical Society*, *348*(2), 503–520.

Costarelli, D., Sambucini, A., & Vinti, G. (2019). Convergence in orlicz spaces by means of the multivariate max-product neural network operators of the kantorovich type and applications. *Neural Computation & Applications*, *31*, 5069–5078.

Costarelli, D., & Spigler, R. (2013a). Approximation results for neural network operators activated by sigmoidal functions. *Neural Networks*, *44*, 101–106.

Costarelli, D., & Spigler, R. (2013b). Multivariate neural network operators with sigmoidal activation functions. *Neural Networks*, *48*, 72–77.

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

De Boor, C. (2001). *Applied mathematical sciences*, *A practical guide to splines*. Berlin: Springer.

E, W., Han, J., & Jentzen, A. (2017). Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematical Statistics*, *5*(4), 349–380.

E, W., & Yu, B. (2018). The deep ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematical Statistics*, *6*(1), 1–12.

Edmunds, D. E., & Evans, W. D. (2004). *Springer monographs in mathematics*, *Hardy operators, function spaces and embeddings* (p. xii+326). Berlin: Springer.

Edmunds, D. E., & Triebel, H. (1996). *Cambridge tracts in mathematics*, *Function spaces, entropy numbers, differential operators*. Cambridge University Press.

Elbrächter, D., Grohs, P., Jentzen, A., & Schwab, C. (2018). DNN expression rate analysis of high-dimensional PDEs: Application to option pricing. arXiv preprint arXiv:1809.07669.

Evans, L. (1999). *Graduate studies in mathematics*: *Vol. 19*, *Partial differential equations*. American Mathematical Society.

Geist, M., Petersen, P. C., Raslan, M., Schneider, R., & Kutyniok, G. (2020). Numerical solution of the parametric diffusion equation by deep neural networks. arXiv preprint arXiv:2004.12131.

Gilbarg, D., & Trudinger, N. (1998). *A series of comprehensive studies in mathematics*: *Vol. 224*, *Elliptic partial differential equations of second order* (2nd ed.). Berlin: Springer.

Grohs, P., Hornung, F., Jentzen, A., & von Wurstemberger, P. (2018). A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of black-scholes partial differential equations. arXiv preprint arXiv:1809.02362.

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

Guliyev, N. J., & Ismailov, V. E. (2016). A single hidden layer feedforward network with only one neuron in the hidden layer can approximate any univariate function. *Neural Computations*, *28*(7), 1289–1304.

Guliyev, N. J., & Ismailov, V. E. (2018). Approximation capability of two hidden layer feedforward neural networks with fixed weights. *Neurocomputing*, *316*, 262–269.

Han, J., Jentzen, A., & E, W. (2018). Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences of the United States of America*, *115*(34), 8505–8510.

Han, J., Nica, M., & Stinchcombe, A. (2020). A derivative-free method for solving elliptic partial differential equations with deep neural networks. arXiv preprint arXiv:2001.06145.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, *4*(2), 251–257.

Jentzen, A., Salimova, D., & Welti, T. (2018). A proof that deep artificial neural networks overcome the curse of dimensionality in the numerical approximation of Kolmogorov partial differential equations with constant diffusion and nonlinear drift coefficients. arXiv preprint arXiv:1809.07321.

Katsuura, H. (2009). Summations involving binomial coefficients. *The College Mathematics Journal*, *40*(4), 275–278.

Kidger, P., & Lyons, T. (2019). Universal approximation with deep narrow networks. arXiv preprint arXiv:1905.08539.

Kutyniok, G., Petersen, P. C., Raslan, M., & Schneider, R. (2019). A theoretical analysis of deep neural networks and parametric PDEs. Accepted for Publication in *Constructive Approximation*, arXiv preprint arXiv:1904.00377.

Laakmann, F., & Petersen, P. C. (2020). Efficient approximation of solutions of parametric linear transport equations by relu DNNs. arXiv preprint arXiv:2001.11441.

Lagaris, I., Likas, A., & Fotiadis, D. (1998). Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks Learning Systems*, *9*(5), 987–1000.

Li, B., Tang, S., & Yu, H. (2020). Better approximations of high dimensional smooth functions by deep neural networks with rectified power units. *Communications in Computer Physics*, *27*, 379–411.

Lin, S.-B. (2019). Generalization and expressivity for deep nets . *IEEE Transactions on Neural Networks Learning*, *30*(5), 1392–1406.

Lu, L., Meng, X., Mao, Z., & Karniadakis, G. (2019). Deepxde: A deep learning library for solving differential equations. arXiv preprint arXiv:1907.04502.

Maiorov, V., & Pinkus, A. (1999). Lower bounds for approximation by MLP neural networks. *Neurocomputing*, *25*(1–3), 81–91.

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

Mhaskar, H., & Micchelli, C. A. (1992). Approximation by superposition of sigmoidal and radial basis functions. *Advances in Applied Mathematics*, *13*(3), 350–373.

Minai, A. A., & Williams, R. D. (1993). On the derivatives of the sigmoid. *Neural Networks*, *6*(6), 845–853.

Ohn, I., & Kim, Y. (2019). Smooth function approximation by deep neural networks with general activation functions. *Entropy*, *21*(7), 627.

Opschoor, J. A. A., Petersen, P. C., & Schwab, C. (2020). Deep relu networks and high-order finite element methods. *Analysis and Applications*, *18*(05), 715–770.

Petersen, P., & Voigtländer, F. (2018). Optimal approximation of piecewise smooth functions using deep relu neural networks. *Neural Networks*, *108*, 296–330.

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

Rolnick, D., & Tegmark, M. (2018). The power of deeper networks for expressing natural functions. In *International conference on learning representations*.

Roubíček, T. (2013). *International series of numerical mathematics*: *Vol. 153*, *Nonlinear partial differential equations with applications* (2nd ed.). Basel: Springer Science+Business Media.

Schmidt-Hieber, J. (2017). Nonparametric regression using deep neural networks with relu activation function. arXiv preprint arXiv:1708.06633.

Schwab, C., & Zech, J. (2019). Deep learning in high dimension: Neural network expression rates for generalized polynomial chaos expansions in UQ. *Analysis and Applications*, *17*(1), 19–55.

Shaham, U., Cloninger, A., & Coifman, R. (2018). Provable approximation properties for deep neural networks. *Applied and Computational Harmonic Analysis*, *44*(3), 537–557.

Sirignano, J., & Spiliopoulos, K. (2018). DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, *375*, 1339–1364.

Stein, E. (1979). *Singular integrals and differentiability properties of functions* (3rd ed.). Princeton: Princeton University Press.

Suzuki, T. (2019). Adaptivity of deep relu network for learning in besov and mixed smooth besov spaces: optimal rate and curse of dimensionality. In *7th international conference on learning representations, ICLR 2019, New Orleans, la, USA, May 6-9, 2019*.

Tang, S., Li, B., & Yu, H. (2019). Chebnet: Efficient and stable constructions of deep neural networks with rectified power units using Chebyshev approximations. arXiv preprint arXiv:1911.05467.

Triebel, H. (1978). *Interpolation theory, function spaces, differential operators*. Amsterdam: North-Holland Publishing Company.

Yarotsky, D. (2017). Error bounds for approximations with deep relu networks. *Neural Networks*, *94*, 103–114.

Yarotsky, D. (2018). Optimal approximation of continuous functions by very deep relu networks. In S. Bubeck, V. Perchet, & P. Rigollet (Eds.), *Proceedings of Machine Learning Research*: *Vol. 75*, *Proceedings of the 31st conference on learning theory* (pp. 639–649). PMLR.