ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Seminar for Applied Mathematics SAM

Exponential ReLU DNN expression of holomorphic maps in high dimension

J. A. A. Opschoor and Ch. Schwab and J. Zech

Research Report No. 2019-35

July 2019
Latest revision: August 2020

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Funding SNF: 184530

Exponential ReLU DNN expression of holomorphic maps in high dimension

Joost A.A. Opschoor* Christoph Schwab* Jakob Zech†

August 31, 2020

Abstract

For a parameter dimension $d \in \mathbb{N}$, we consider the approximation of many-parametric maps $u : [-1,1]^d \to \mathbb{R}$ by deep ReLU neural networks. The input dimension d may possibly be large, and we assume quantitative control of the domain of holomorphy of u: i.e., u admits a holomorphic extension to a Bernstein polyellipse $\mathcal{E}_{\rho_1} \times ... \times \mathcal{E}_{\rho_d} \subset \mathbb{C}^d$ of semiaxis sums $\rho_i > 1$ containing $[-1,1]^d$. We establish the exponential rate $O(\exp(-bN^{1/(d+1)}))$ of expressive power in terms of the total NN size N and of the input dimension d of the ReLU NN in $W^{1,\infty}([-1,1]^d)$. The constant $b > 0$ depends on $(\rho_j)_{j=1}^d$ which characterizes the coordinate-wise sizes of the Bernstein-ellipses for u. We also prove exponential convergence in stronger norms for the approximation by DNNs with more regular, so-called "rectified power unit" (RePU) activations. Finally, we extend DNN expression rate bounds also to two classes of non-holomorphic functions, in particular to d-variate, Gevrey-regular functions, and, by composition, to certain multivariate probability distribution functions with Lipschitz marginals.

Key words: Deep ReLU neural networks, approximation rates, exponential convergence
Subject Classification: 41A25, 41A10, 41A46

Contents

*Seminar for Applied Mathematics, ETH Zürich, CH 8092 Zürich, Switzerland
†Department of Mathematics and IWR, Heidelberg University, 69120 Heidelberg, Germany.

1 Introduction

In recent years, so-called *deep artificial neural networks* ('DNNs' for short) have seen dramatic development in applications from data science and machine learning.

Accordingly, after early results in the '90s on genericity and universality of DNNs (see [28] for a survey and references), in recent years the refined mathematical analysis of their approximation properties viz. "expressive power" has received increasing attention. A particular class of many-parametric maps whose DNN approximation needs to be considered in many applications are real-analytic and holomorphic maps. Accordingly, the question of DNN expression rate bounds for such maps has received some attention in the approximation theory literature [21, 22, 10, 9].

It is well-known that multi-variate, holomorphic maps admit *exponential expression rates by multivariate polynomials*. In particular, countably-parametric maps $u : [-1,1]^\infty \to \mathbb{R}$ can be represented under certain conditions by so-called *generalized polynomial chaos* expansions with quantified sparsity in coefficient sequences. This, in turn, implies N-term truncations with controlled approximation rate bounds in terms of N, with approximation rates which do not depend on the dimension of the active parameters in the truncated approximation [6, 5]. The polynomials which appear in such expansions can, in turn, be represented by DNNs, either exactly for certain activation functions, or approximately for example for the so-called rectified linear unit ("ReLU") activation with exponentially small representation error [18, 37].

The purpose of the present paper is to establish corresponding DNN expression rate bounds in Lipschitz-norm (i.e. $W^{1,\infty}$-norm) for high-dimensional, analytic maps $u : [-1,1]^d \to \mathbb{R}$. We focus on ReLU DNNs, but comment in passing also on versions of our results for other DNN activation functions. Next, we briefly discuss the relation of previous results to the present work and also outline the structure of this paper.

1.1 Recent mathematical results on expressive power of DNNs

The survey [28] presented succinct proofs of genericity of shallow NNs in various function classes, as shown originally e.g. in [16, 15, 20] and reviewed the state of mathematical theory of DNNs up to that point. Moreover, exponential expression rate bounds for analytic functions by neural networks had already been achieved in the '90s. We mention in particular [22] where smooth, nonpolynomial activation functions were considered.

More closely related to the present work are the references [10, 21]. In [21], approximation rates for deep NN approximations of multivariate functions which are analytic have been investigated. Exponential rate bounds in terms of the total size of the NN have been obtained, for sigmoidal activation functions. In [37], it was observed that the multiplication of two real numbers, and consequently polynomials, can efficiently be approximated by deep ReLU NNs. This was used in [10] to prove bounds on the DNN approximation of certain functions $u : [-1,1]^d \to \mathbb{R}$ which admit holomorphic extensions to some open subset of $\mathbb{C}^d$ by deep ReLU NNs. In particular, it was assumed that u admits a Taylor expansion about the origin of $\mathbb{C}^d$ which converges absolutely and uniformly on $[-1,1]^d$. It is well-known that not every u which is real-analytic in $[-1,1]^d$ admits such an expansion. In the present paper, we prove sharper expression rate bounds for both, the ReLU activation σ_1 and RePU activations σ_r, for functions which merely are assumed to be real-analytic in $[-1,1]^d$, in $L^\infty([-1,1]^d)$ and in stronger norms, thereby generalizing both [10] and [21]. For σ_1-NNs similar results were recently presented in [9], albeit with slightly larger bounds on the network size.

1.2 Contributions of the present paper

We prove exponential expression rate bounds of DNNs for d-variate, real-valued functions which depend analytically on their d inputs. Specifically, for holomorphic mappings $u : [-1,1]^d \to \mathbb{R}$, we prove expression error bounds in $L^\infty([-1,1]^d)$ and in $W^{k,\infty}([-1,1]^d)$, for $k \in \mathbb{N}$ (the precise range of k depending on properties of the NN activation σ). We consider both, ReLU activation $\sigma_1 : \mathbb{R} \to \mathbb{R}_+ : x \mapsto x_+$ and RePU activations $\sigma_r : \mathbb{R} \to \mathbb{R}_+ : x \mapsto (x_+)^r$ for some integer $r \geq 2$. Here, $x_+ = \max\{x, 0\}$. The expression error bounds in our main result, Theorem 3.6, with ReLU activation σ_1 are in $W^{1,\infty}([-1,1]^d)$ and of the general type $O(\exp(-bN^{1/(d+1)}))$ in terms of the NN size N, with a constant $b > 0$ depending on the domain of analyticity, but independent of N (however, with the constant implied in the Landau symbol $O(\cdot)$ depending exponentially on d, in general). With activation σ_r for $r \geq 2$, Theorem 3.10 has corresponding expression error bounds in $W^{k,\infty}([-1,1]^d)$ for arbitrary fixed $k \in \mathbb{N}$ and of the type $O(\exp(-bN^{1/d}))$ in terms of the NN size N. For all $r \in \mathbb{N}$, the parameters of the σ_r-neural networks approximating u (so-called "weights" and "biases") are continuous functions of u in appropriate norms. All of our proofs are constructive. I.e., they demonstrate how to build sparsely connected DNNs achieving the claimed convergence rates. We comment in Rmk. 3.7 and Rmk. 3.11 how these statements imply results for (the simpler architecture of) fully connected neural networks.

The main results, Theorems 3.6 and 3.10, are expression rate bounds for holomorphic functions. Similar bounds for Gevrey-regular functions are given in Section 4.3.4. In Section 4.3.5, we conclude the same bounds also for certain classes of nonholomorphic, merely Lipschitz-continuous functions, by leveraging the compositional nature of DNN approximation and Theorems 3.6 and 3.10.

1.3 Outline

The structure of the paper is as follows. In Section 2, we present the definition of the DNN architectures and fix notation and terminology. We also review in Section 2.2 a "ReLU DNN calculus", from recent work [27, 11], which will facilitate the ensuing DNN expression rate analysis. A first set of key results are ReLU DNN expression rates in $W^{1,\infty}([-1,1]^d)$ for multivariate Legendre polynomials, which are proved in Section 2.3. These novel expression rate bounds are explicit in the $W^{1,\infty}$-accuracy and in the polynomial degree. They are of independent interest and remarkable in that the ReLU DNNs which emulate the polynomials at exponential rates, as we prove, realize continuous, piecewise affine functions. They are based on [18, 37]. The proofs, being constructive, shed a rather precise light on the architecture, in particular depth and width of the ReLU DNNs, that is sufficient for polynomial emulation. In Section 2.4, we briefly comment on corresponding results for RePU activations; as a rule, the same exponential rates are achieved for slightly smaller NNs and in norms which are stronger than $W^{1,\infty}$.

Section 3 then contains the main results of this note: exponential ReLU DNN expression rate bounds for d-variate, holomorphic maps. They are based on a) polynomial approximation of these maps and on b) ReLU DNN reapproximation of the approximating polynomials. These are presented in Sections 3.1 and 3.2. Again we comment in Section 3.3 on modifications in the results for RePU activations. Section 4 contains a brief indication of further directions and open problems.

Acknowledgement: This work was supported in part under an SNSF Early Postdoc.Mobility Fellowship 184530 to JZ. Research performed in part during a visit to the CRM Montreal, Canada, of CS and JZ in May 2019.

1.4 Notation

We adopt standard notation consistent with our previous works [40, 42]: $\mathbb{N} = \{1, 2, \dots\}$ and $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. We write $\mathbb{R}_+ := \{x \in \mathbb{R} : x \geq 0\}$. The symbol C will stand for a generic, positive

constant independent of any asymptotic quantities in an estimate, which may change its value even within the same equation.

In statements about polynomial expansions we require multiindices $\boldsymbol{\nu} = (\nu_j)_{j=1,\dots,d} \in \mathbb{N}_0^d$ for $d \in \mathbb{N}$. The *total order* of a multiindex $\boldsymbol{\nu}$ is denoted by $|\boldsymbol{\nu}|_1 := \sum_{j=1}^d \nu_j$. The notation $\operatorname{supp} \boldsymbol{\nu}$ stands for the *support* of the multiindex, i.e. $\operatorname{supp} \boldsymbol{\nu} = \{j \in \{1,\dots,d\} : \nu_j \neq 0\}$. The size of the support of $\boldsymbol{\nu} \in \mathbb{N}_0^d$ is $|\operatorname{supp} \boldsymbol{\nu}|$; it will, subsequently, indicate the number of active coordinates in the multivariate monomial term $\boldsymbol{y}^{\boldsymbol{\nu}} := \prod_{j=1}^d y_j^{\nu_j}$.

A subset $\Lambda \subseteq \mathbb{N}_0^d$ is called *downward closed*[1], if $\boldsymbol{\nu} = (\nu_j)_{j=1}^d \in \Lambda$ implies $\boldsymbol{\mu} = (\mu)_{j=1}^d \in \Lambda$ for all $\boldsymbol{\mu} \leq \boldsymbol{\nu}$. Here, the ordering "$\leq$" on $\mathbb{N}_0^d$ is defined as $\mu_j \leq \nu_j$, for all $j = 1,\dots,d$. We write $|\Lambda|$ to denote the finite cardinality of a set Λ.

We write $B_\varepsilon^{\mathbb{C}} := \{z \in \mathbb{C} : |z| < \varepsilon\}$. Elements of $\mathbb{C}^d$ will be denoted by boldface characters such as $\boldsymbol{y} = (y_j)_{j=1}^d \in [-1,1]^d \subset \mathbb{C}^d$. For $\boldsymbol{\nu} \in \mathbb{N}_0^d$, standard notations $\boldsymbol{y}^{\boldsymbol{\nu}} := \prod_{j=1}^d y_j^{\nu_j}$ and $\boldsymbol{\nu}! = \prod_{j=1}^d \nu_j!$ will be employed (with the conventions $0! := 1$ and $0^0 := 1$). For $n \in \mathbb{N}_0$ we let $\mathbb{P}_n := \operatorname{span}\{y^j : 0 \leq j \leq n\}$ be the space of polynomials of degree at most n, and for a finite index set $\Lambda \subset \mathbb{N}_0^d$ we denote $\mathbb{P}_\Lambda := \operatorname{span}\{\boldsymbol{y}^{\boldsymbol{\nu}} : \boldsymbol{\nu} \in \Lambda\}$.

2 Deep neural network approximations

2.1 DNN architecture

We consider *deep neural networks (DNNs for short)* of feed forward type. Such a NN f can mathematically be described as a repeated composition of affine transformations with a nonlinear *activation function*.

More precisely: For an *activation function* $\sigma : \mathbb{R} \to \mathbb{R}$, a fixed *number of hidden layers* $L \in \mathbb{N}$, numbers $N_\ell \in \mathbb{N}$ of *computation nodes in layer* $\ell \in \{1,\dots,L+1\}$, $f : \mathbb{R}^{N_0} \to \mathbb{R}^{N_{L+1}}$ is realized by a feedforward neural network, if for certain *weights* $w_{i,j}^\ell \in \mathbb{R}$, and *biases* $b_j^\ell \in \mathbb{R}$ it holds for all $\boldsymbol{x} = (x_i)_{i=1}^{N_0}$

$$z_j^1 = \sigma\left(\sum_{i=1}^{N_0} w_{i,j}^1 x_i + b_j^1\right), \quad j \in \{1,\dots,N_1\}, \tag{2.1}$$

and

$$z_j^{\ell+1} = \sigma\left(\sum_{i=1}^{N_\ell} w_{i,j}^{\ell+1} z_i^\ell + b_j^{\ell+1}\right), \quad \ell \in \{1,\dots,L-1\}, \ \ j \in \{1,\dots,N_{\ell+1}\}, \tag{2.2}$$

and finally

$$f(\boldsymbol{x}) = (z_j^{L+1})_{j=1}^{N_{L+1}} = \left(\sum_{i=1}^{N_L} w_{i,j}^{L+1} z_i^L + b_j^{L+1}\right)_{j=1}^{N_{L+1}}. \tag{2.3}$$

In this case N_0 is the dimension of the input, and N_{L+1} is the dimension of the output. Furthermore z_j^ℓ denotes the output of unit j in layer ℓ. The weight $w_{i,j}^\ell$ has the interpretation of connecting the ith unit in layer $\ell - 1$ with the jth unit in layer ℓ.

Except when explicitly stated, we will not distinguish between the network (which is defined through σ, the $w_{i,j}^\ell$ and b_j^ℓ) and the function $f : \mathbb{R}^{N_0} \to \mathbb{R}^{N_{L+1}}$ it realizes. We note in passing that this relation is typically not one-to-one, i.e. different NNs may realize the same function as their output. Let us also emphasize that we allow the weights $w_{i,j}^\ell$ and biases b_j^ℓ for $\ell \in \{1,\dots,L+1\}$, $i \in \{1,\dots,N_{\ell-1}\}$ and $j \in \{1,\dots,N_\ell\}$ to take any value in $\mathbb{R}$, i.e. we do not consider quantization as e.g. in [1, 27].

[1]Index sets with the "downward closed" property are also referred to in the literature [24] as *lower sets*.

As is customary in the theory of NNs, the number of hidden layers L of a NN is referred to as *depth*[2] and the total number of nonzero weights and biases as the *size* of the NN. Hence, for a DNN f as in (2.1)-(2.3), we define

$$\mathrm{size}(f) := |\{(i,j,\ell) \,:\, w_{i,j}^{\ell} \neq 0\}| + |\{(j,\ell) \,:\, b_j^{\ell} \neq 0\}| \qquad \text{and} \qquad \mathrm{depth}(f) := L.$$

In addition, $\mathrm{size}_{\mathrm{in}}(f) := |\{(i,j) : w_{i,j}^1 \neq 0\}| + |\{j \,:\, b_j^1 \neq 0\}|$ and $\mathrm{size}_{\mathrm{out}}(f) := |\{(i,j) : w_{i,j}^{L+1} \neq 0\}| + |\{j \,:\, b_j^{L+1} \neq 0\}|$, which are the number of nonzero weights and biases in the input layer of f and in the output layer, respectively.

The proofs of our main results Theorem 3.6 and Theorem 3.10 are constructive, in the sense that we will explicitly construct NNs with the desired properties. We construct these NNs by assembling smaller networks, using the operations of concatenation and parallelization, as well as so-called "identity-networks" which realize the identity mapping. Below, we recall the definitions. For these operations, we also provide bounds on the number of nonzero weights in the input layer and the output layer of the corresponding network, which can be derived from the definitions in [27].

2.2 DNN calculus

Throughout, as activation function σ we consider either the ReLU activation function

$$\sigma_1(x) := \max\{0, x\} \qquad x \in \mathbb{R} \tag{2.4}$$

or, as suggested in [21, 19, 17], for $r \in \mathbb{N}$, $r \geq 2$, the RePU activation function

$$\sigma_r(x) := \max\{0, x\}^r \qquad x \in \mathbb{R}. \tag{2.5}$$

If a NN uses σ_r as activation function, we refer to it as σ_r-NN. ReLU NNs are referred to as σ_1-NNs. We assume throughout that all activations in a DNN are of equal type.

Remark 2.1 (Historical note on rectified power units)**.** *"Rectified power unit" (RePU) activation functions are particular cases of so-called* sigmoidal functions of order $k \in \mathbb{N}$ for $k \geq 2$, *i.e.* $\lim_{x\to\infty} \frac{\sigma(x)}{x^k} = 1$, $\lim_{x\to-\infty} \frac{\sigma(x)}{x^k} = 0$ *and* $|\sigma(x)| \leq K(1+|x|)^k$ *for* $x \in \mathbb{R}$. *The use of NNs with such activation functions for function approximation dates back to the early 1990's, cf. e.g. [21, 19]. Proofs in [21, Section 3] proceed in three steps. First, a given function f was approximated by a polynomial, then this polynomial was expressed as a linear combination of powers of a RePU, and finally it was shown that for $r \geq 2$ and arbitrary $A > 0$ the RePU σ_r can be approximated on $[-A, A]$ with arbitrarily small $L^\infty([-A,A])$-error ε by a NN with a sigmoidal activation function of order $k{=}r$, which has depth 1 and fixed network size independent of A and ε ([21, Lemma 3.6]). As remarked directly below [21, Lemma 3.6], this result remains true for the $L^\infty(\mathbb{R})$-norm (instead of $L^\infty([-A,A])$) if, additionally, σ is uniformly continuous on $\mathbb{R}$. As also remarked below [21, Lemma 3.6], a similar statement holds for the approximation of the ReLU σ_1 by a NN with sigmoidal activation function of the order $k = 1$.*

For any $r \in \mathbb{N}$, in the proof of [21, Lemma 3.6] it was observed that for continuous, sigmoidal σ of order $k = r$, the σ-NN that approximates σ_r is uniformly continuous on $[-A, A]$. From this, it follows that σ_r-NNs can be approximated up to arbitrarily small $L^\infty([-A,A])$-error by σ-NNs with NN size independent of A and ε. Again, uniform continuity of σ on $\mathbb{R}$ implies the same result w.r.t. the $L^\infty(\mathbb{R})$-norm.

The exact realization of polynomials by σ_r-networks for $r \geq 2$ was observed in the proof of [21, Theorem 3.3], based on ideas in the proof of [4, Theorem 3.1]. The same result was recently rediscovered in [17, Theorem 3.1], whose authors were apparently not aware of [4, 21].

[2] In other recent references (e.g. [25]), slightly different terminology for the number L of layers in the DNN differing from the convention in the present paper by a constant factor, is used. This difference will be inconsequential for all results that follow.

We now indicate several fundamental operations on NNs which will be used in the following. These operations have been frequently used in recent works [27, 25, 11].

2.2.1 Parallelization

We now recall the parallelization of two networks f and g, which in parallel emulates f and g. We first describe the parallelization of networks with the same inputs as in [27, Definition 2.7], the parallelization of networks with different inputs is similar and introduced directly afterwards.

Let f and g be two NNs with the same depth $L \in \mathbb{N}_0$ and the same input dimension $n \in \mathbb{N}$. Denote by m_f the output dimension of f and by m_g the output dimension of g. Then there exists a neural network (f,g), called *parallelization* of f and g, which in parallel emulates f and g, i.e.

$$(f,g) : \mathbb{R}^n \to \mathbb{R}^{m_f} \times \mathbb{R}^{m_g} : \boldsymbol{x} \mapsto (f(\boldsymbol{x}), g(\boldsymbol{x})).$$

It holds that $\operatorname{depth}((f,g)) = L$ and that $\operatorname{size}((f,g)) = \operatorname{size}(f)+\operatorname{size}(g)$, $\operatorname{size}_{\text{in}}((f,g)) = \operatorname{size}_{\text{in}}(f)+\operatorname{size}_{\text{in}}(g)$ and $\operatorname{size}_{\text{out}}((f,g)) = \operatorname{size}_{\text{out}}(f) + \operatorname{size}_{\text{out}}(g)$.

We next recall the parallelization of networks with inputs of possibly different dimension as in [11, Setting 5.2]. To this end, we let f and g be two NNs with the same depth $L \in \mathbb{N}_0$ whose input dimensions n_f and n_g may be different, and whose output dimensions we will denote by m_f and m_g, respectively.

Then there exists a neural network $(f,g)_{\text{d}}$, called *full parallelization of networks with distinct inputs* of f and g, which in parallel emulates f and g, i.e.

$$(f,g)_{\text{d}} : \mathbb{R}^{n_f} \times \mathbb{R}^{n_g} \to \mathbb{R}^{m_f} \times \mathbb{R}^{m_g} : (\boldsymbol{x}, \tilde{\boldsymbol{x}}) \mapsto (f(\boldsymbol{x}), g(\tilde{\boldsymbol{x}})).$$

It holds that $\operatorname{depth}((f,g)_{\text{d}}) = L$ and that $\operatorname{size}((f,g)_{\text{d}}) = \operatorname{size}(f) + \operatorname{size}(g)$, $\operatorname{size}_{\text{in}}((f,g)_{\text{d}}) = \operatorname{size}_{\text{in}}(f) + \operatorname{size}_{\text{in}}(g)$ and $\operatorname{size}_{\text{out}}((f,g)_{\text{d}}) = \operatorname{size}_{\text{out}}(f) + \operatorname{size}_{\text{out}}(g)$.

Parallelizations of networks with possibly different inputs can be used consecutively to emulate multiple networks in parallel.

2.2.2 Identity networks

We now recall identity networks ([27, Lemma 2.3]), which emulate the identity map.

For all $n \in \mathbb{N}$ and $L \in \mathbb{N}_0$ there exists a σ_1-*identity network* $\operatorname{Id}_{\mathbb{R}^n}$ of depth L which emulates the identity map $\operatorname{Id}_{\mathbb{R}^n} : \mathbb{R}^n \to \mathbb{R}^n : \boldsymbol{x} \mapsto \boldsymbol{x}$. It holds that

$$\operatorname{size}(\operatorname{Id}_{\mathbb{R}^n}) \leq 2n(\operatorname{depth}(\operatorname{Id}_{\mathbb{R}^n}) + 1), \qquad \operatorname{size}_{\text{in}}(\operatorname{Id}_{\mathbb{R}^n}) \leq 2n, \qquad \operatorname{size}_{\text{out}}(\operatorname{Id}_{\mathbb{R}^n}) \leq 2n. \tag{2.6}$$

Analogously, for $r \geq 2$ there exist σ_r-*identity networks*. To construct them, we use the *concatenation* $f \bullet g$ of two NNs f and g as introduced in [27, Definition 2.2]. As we shall make use of it subsequently in Propositions 2.3 and 2.4, we recall its definition here for convenience of the reader.

Definition 2.2 ([27, Definition 2.2])**.** *Let f, g be such that the output dimension of g equals the input dimension of f, which we denote by k. Denote the weights and biases of f by $\{u_{i,j}^\ell\}_{i,j,\ell}$ and $\{a_j^\ell\}_{j,\ell}$ and those of g by $\{v_{i,j}^\ell\}_{i,j,\ell}$ and $\{c_j^\ell\}_{j,\ell}$. Then, we denote by $f \bullet g$ be the NN with weights and biases*

$$w_{i,j}^\ell = \begin{cases} v_{i,j}^\ell & \ell \leq \operatorname{depth}(g), \\ \sum_{q=1}^k v_{i,q}^\ell u_{q,j}^1 & \ell = \operatorname{depth}(g)+1, \\ u_{i,j}^{\ell-\operatorname{depth}(g)} & \ell > \operatorname{depth}(g)+1, \end{cases} \qquad b_j^\ell = \begin{cases} c_{i,j}^\ell & \ell \leq \operatorname{depth}(g), \\ \sum_{q=1}^k c_q^\ell u_{q,j}^1 + a_j^1 & \ell = \operatorname{depth}(g)+1, \\ a_j^{\ell-\operatorname{depth}(g)} & \ell > \operatorname{depth}(g)+1, \end{cases}$$

for $\ell = 1, \ldots, \operatorname{depth}(f) + \operatorname{depth}(g) + 1$.

It is easy to check, that the network $f \bullet g$ emulates the composition $\boldsymbol{x} \mapsto f(g(\boldsymbol{x}))$ and satisfies $\operatorname{depth}(f \bullet g) = \operatorname{depth}(f) + \operatorname{depth}(g)$.

The concatenation of Definition 2.2 will only be used in the proof of Propositions 2.3 and 2.4 below. *Throughout the remainder of this work, we use sparse concatenations $f \circ g$ introduced in Section 2.2.3*, whose network size can be estimated by $C(\operatorname{size}(f) + \operatorname{size}(g))$ for an absolute constant C. The reason for introducing $\circ$ in addition to $\bullet$, is that the size of $f \bullet g$ cannot be bounded by $C(\operatorname{size}(f) + \operatorname{size}(g))$ for an absolute constant C. This can be seen by considering the number of nonzero weights in layer $\ell = \operatorname{depth}(g) + 1$, e.g. for $k = 1$, $N_\ell = 1$ and arbitrary layer sizes $N_{\ell-1}, N_{\ell+1} \in \mathbb{N}$.

Proposition 2.3. *For all $r \geq 2$, $n \in \mathbb{N}$ and $L \in \mathbb{N}_0$ there exists a σ_r-NN $\mathrm{Id}_{\mathbb{R}^n}$ of depth L which emulates the identity function $\mathrm{Id}_{\mathbb{R}^n} : \mathbb{R}^n \to \mathbb{R}^n : \boldsymbol{x} \mapsto \boldsymbol{x}$. It holds that*

$$\operatorname{size}(\mathrm{Id}_{\mathbb{R}^n}) \leq nL(4r^2 + 2r), \qquad \operatorname{size}_{\mathrm{in}}(\mathrm{Id}_{\mathbb{R}^n}) \leq 4nr, \qquad \operatorname{size}_{\mathrm{out}}(\mathrm{Id}_{\mathbb{R}^n}) \leq n(2r + 1).$$

Proof. First we consider $n = 1$ and proceed in two steps: We discuss $L = 0$,1 in Step 1 and $L > 1$ in Step 2.

Step 1. For $L = 0$, let $\mathrm{Id}_{\mathbb{R}^n}$ be the network with weights $w^1_{i,j} = \delta_{i,j}$, $b^1_j = 0$, $i, j = 1, \ldots, n$. We next consider $L = 1$. It was shown in [17, Theorem 2.5] that there exist $(a_k)_{k=0}^r \in \mathbb{R}^{r+1}$ and $(b_k)_{k=1}^r \in \mathbb{R}^r$ such that for all $x \in \mathbb{R}$

$$x = a_0 + \sum_{k=1}^{r} a_k (x + b_k)^r = a_0 + \sum_{k=1}^{r} a_k \sigma_r(x + b_k) + \sum_{k=1}^{r} a_k (-1)^r \sigma_r(-x - b_k).$$

This shows the existence of a network $\mathrm{Id}_{\mathbb{R}^1} : \mathbb{R} \to \mathbb{R}$ of depth 1 realizing the identity on $\mathbb{R}$. The network employs $2r$ weights and $2r$ biases in the first layer, and $2r$ weights and one bias (namely a_0) in the output layer. Its size is thus $6r + 1$.

Step 2. For $L > 1$, we consider the L-fold concatenation $\mathrm{Id}_{\mathbb{R}^1} \bullet \cdots \bullet \mathrm{Id}_{\mathbb{R}^1}$ of the identity network $\mathrm{Id}_{\mathbb{R}^1}$ from Step 1. The resulting network has depth L, input dimension 1 and output dimension 1. The number of weights and the number of biases in the first layer both equal $2r$, the number of weights in the output layer equals $2r$, and the number of biases 1. In each of the $L - 1$ other hidden layers, the number of weights is $4r^2$, and the number of biases $2r$. In total, the network has size at most $4r + (L - 1)(4r^2 + 2r) + 2r + 1 \leq L(4r^2 + 2r)$, where we used that $r \geq 2$.

Identity networks with input size $n \in \mathbb{N}$ are obtained as the full parallelization with distinct inputs of n identity networks with input size 1. □

2.2.3 Sparse concatenation

The *sparse concatenation* of two σ_1-NNs f and g was introduced in [27, Definition 2.5].

Let f and g be σ_1-NNs, such that the number of nodes in the output layer of g equals the number of nodes in the input layer of f. Denote by n the number of nodes in the input layer of g, and by m the number of nodes in the output layer of f. Then, with "$\bullet$" as in Definition 2.2, the *sparse concatenation of the NNs f and g* is defined as the network

$$f \circ g := f \bullet \mathrm{Id}_{\mathbb{R}^k} \bullet g, \tag{2.7}$$

where $\mathrm{Id}_{\mathbb{R}^k}$ is the σ_1-identity network of depth 1. The network $f \circ g$ realizes the function

$$f \circ g : \mathbb{R}^n \to \mathbb{R}^m : \boldsymbol{x} \mapsto (f(g(\boldsymbol{x})), \tag{2.8}$$

i.e., by abuse of notation, the symbol "$\circ$" has two meanings here, depending on whether we interpret $f \circ g$ as a function or as a network. This will not be the cause of confusion however. It holds $\operatorname{depth}(f \circ g) = \operatorname{depth}(f) + 1 + \operatorname{depth}(g)$,

$$\operatorname{size}(f \circ g) = \operatorname{size}(f) + \operatorname{size}_{\mathrm{in}}(f) + \operatorname{size}_{\mathrm{out}}(g) + \operatorname{size}(g) \leq 2 \operatorname{size}(f) + 2 \operatorname{size}(g) \tag{2.9}$$

and

$$\mathrm{size}_{\mathrm{in}}(f \bullet g) = \begin{cases} \mathrm{size}_{\mathrm{in}}(g) & \mathrm{depth}(g) \geq 1, \\ 2\,\mathrm{size}_{\mathrm{in}}(g) & \mathrm{depth}(g) = 0, \end{cases} \qquad \mathrm{size}_{\mathrm{out}}(f \bullet g) = \begin{cases} \mathrm{size}_{\mathrm{out}}(f) & \mathrm{depth}(f) \geq 1, \\ 2\,\mathrm{size}_{\mathrm{out}}(f) & \mathrm{depth}(f) = 0. \end{cases}$$

For a proof, we refer to [27, Remark 2.6].

A similar result holds for σ_r-NNs. In this case we define the sparse concatenation $f \bullet g$ as in (2.7), but with $\mathrm{Id}_{\mathbb{R}^k}$ now denoting the σ_r-identity network of depth 1 from Proposition 2.3.

Proposition 2.4. *For $r \geq 2$ let f, g be two σ_r-NNs such that the output dimension of g, which we denote by $k \in \mathbb{N}$, equals the input dimension of f, and suppose that $\mathrm{size}_{\mathrm{in}}(f), \mathrm{size}_{\mathrm{out}}(g) \geq k$. Denote by $f \bullet g$ the σ_r-network obtained by the σ_r-sparse concatenation. Then $\mathrm{depth}(f \bullet g) = \mathrm{depth}(f) + 1 + \mathrm{depth}(g)$ and*

$$\begin{aligned} \mathrm{size}(f \bullet g) &\leq \mathrm{size}(f) + (2r-1)\,\mathrm{size}_{\mathrm{in}}(f) + (2r+1)k + (2r-1)\,\mathrm{size}_{\mathrm{out}}(g) + \mathrm{size}(g) \\ &\leq \mathrm{size}(f) + 2r\,\mathrm{size}_{\mathrm{in}}(f) + (4r-1)\,\mathrm{size}_{\mathrm{out}}(g) + \mathrm{size}(g) \\ &\leq (2r+1)\,\mathrm{size}(f) + 4r\,\mathrm{size}(g). \end{aligned} \tag{2.10}$$

Furthermore,

$$\mathrm{size}_{\mathrm{in}}(f \bullet g) \leq \begin{cases} \mathrm{size}_{\mathrm{in}}(g) & \mathrm{depth}(g) \geq 1, \\ 2r\,\mathrm{size}_{\mathrm{in}}(g) + 2rk \leq 4r\,\mathrm{size}_{\mathrm{in}}(g) & \mathrm{depth}(g) = 0, \end{cases}$$

$$\mathrm{size}_{\mathrm{out}}(f \bullet g) \leq \begin{cases} \mathrm{size}_{\mathrm{out}}(f) & \mathrm{depth}(f) \geq 1, \\ 2r\,\mathrm{size}_{\mathrm{out}}(f) + k \leq (2r+1)\,\mathrm{size}_{\mathrm{out}}(f) & \mathrm{depth}(f) = 0. \end{cases}$$

Proof. It follows directly from Definition 2.2 and Proposition 2.3 that $\mathrm{depth}(f \bullet g) = \mathrm{depth}(f) + 1 + \mathrm{depth}(g)$. To bound the size of the network, note that the weights in layers $\ell = 1, \ldots, \mathrm{depth}(g)$ equal those in the first $\mathrm{depth}(g)$ layers of g. Those in layers $\ell = \mathrm{depth}(g)+3, \ldots, \mathrm{depth}(g)+2+\mathrm{depth}(f)$ equal those in the last $\mathrm{depth}(f)$ layers of f. Layer $\ell = \mathrm{depth}(g)+1$ has $2r\,\mathrm{size}_{\mathrm{out}}(g)$ weights and $2rk$ biases, whereas layer $\ell = \mathrm{depth}(g)+2$ has $2r\,\mathrm{size}_{\mathrm{in}}(f)$ weights and k biases. This shows Equation (2.10) and the bound on $\mathrm{size}_{\mathrm{in}}(f \bullet g)$ and $\mathrm{size}_{\mathrm{out}}(f \bullet g)$. □

Identity networks are often used in combination with parallelizations. In order to parallelize two networks f and g with $\mathrm{depth}(f) < \mathrm{depth}(g)$, the network f can be concatenated with an identity network, resulting in a network whose depth equals $\mathrm{depth}(g)$ and which emulates the same function as f.

2.3 ReLU DNN approximation of polynomials

2.3.1 Basic results

In [18] it was shown that deep networks employing both ReL and BiS ("binary step") units are capable of approximating the product of two numbers with a network whose size and depth increase merely logarithmically in the accuracy. In other words, certain neural networks achieve uniform exponential convergence of the operation of multiplication (of two numbers in a bounded interval) w.r.t. the network size. Independently, a similar result for ReLU networks was obtained in [37]. Here, we shall use the latter result in the following slightly more general form shown in [33]. Contrary to [37], it provides a bound of the error in the $W^{1,\infty}([-1,1])$ norm (instead of the $L^\infty([-1,1])$ norm).

Proposition 2.5. *For any $\delta \in (0,1)$ and $M \geq 1$ there exists a σ_1-NN $\tilde{\times}_{\delta,M} : [-M,M]^2 \to \mathbb{R}$ such that*

$$\sup_{|a|,|b|\leq M} |ab - \tilde{\times}_{\delta,M}(a,b)| \leq \delta, \qquad \operatorname*{ess\,sup}_{|a|,|b|\leq M} \max\left\{\left|b - \frac{\partial}{\partial a}\tilde{\times}_{\delta,M}(a,b)\right|, \left|a - \frac{\partial}{\partial b}\tilde{\times}_{\delta,M}(a,b)\right|\right\} \leq \delta, \tag{2.11}$$

where $\frac{\partial}{\partial a}\tilde{\times}_{\delta,M}(a,b)$ and $\frac{\partial}{\partial b}\tilde{\times}_{\delta,M}(a,b)$ denote weak derivatives. There exists a constant $C > 0$ independent of $\delta \in (0,1)$ and $M \geq 1$ such that $\operatorname{size}_{\text{in}}(\tilde{\times}_{\delta,M}) \leq C$, $\operatorname{size}_{\text{out}}(\tilde{\times}_{\delta,M}) \leq C$,

$$\operatorname{depth}(\tilde{\times}_{\delta,M}) \leq C(1+\log_2(M/\delta)), \qquad \operatorname{size}(\tilde{\times}_{\delta,M}) \leq C(1+\log_2(M/\delta)).$$

Moreover, for every $a \in [-M,M]$, there exists a finite set $\mathcal{N}_a \subseteq [-M,M]$ such that $b \mapsto \tilde{\times}_{\delta,M}(a,b)$ is strongly differentiable at all $b \in (-M,M)\backslash\mathcal{N}_a$.

It is immediate, that Proposition 2.5 implies the existence of networks approximating the multiplication of n different numbers. We now show such a result, generalizing [33, Proposition 3.3] in that we consider the error again in the $W^{1,\infty}$ norm (instead of the L^∞ norm).

Proposition 2.6. *For any $\delta \in (0,1)$, $n \in \mathbb{N}$ and $M \geq 1$ there exists a σ_1-NN $\tilde{\prod}_{\delta,M} : [-M,M]^n \to \mathbb{R}$ such that*

$$\sup_{(x_i)_{i=1}^n \in [-M,M]^n} \left|\prod_{j=1}^n x_j - \tilde{\prod}_{\delta,M}(x_1,\dots,x_n)\right| \leq \delta, \tag{2.12a}$$

$$\operatorname*{ess\,sup}_{(x_i)_{i=1}^n \in [-M,M]^n} \sup_{i=1,\dots,n} \left|\frac{\partial}{\partial x_i}\prod_{j=1}^n x_j - \frac{\partial}{\partial x_i}\tilde{\prod}_{\delta,M}(x_1,\dots,x_n)\right| \leq \delta, \tag{2.12b}$$

where $\frac{\partial}{\partial x_i}$ denotes a weak derivative.

There exists a constant C independent of $\delta \in (0,1)$, $n \in \mathbb{N}$ and $M \geq 1$ such that

$$\operatorname{size}(\tilde{\prod}_{\delta,M}) \leq C(1+n\log(nM^n/\delta)) \quad \text{and} \quad \operatorname{depth}(\tilde{\prod}_{\delta,M}) \leq C(1+\log(n)\log(nM^n/\delta)). \tag{2.13}$$

Proof. We proceed analogously to the proof of [33, Proposition 3.3], and construct $\tilde{\prod}_{\delta,1}$ as a binary tree of $\tilde{\times}_{\cdot,\cdot}$-networks from Proposition 2.5 with appropriately chosen parameters for the accuracy and the maximum input size.

We define $\tilde{n} := \min\{2^k : k \in \mathbb{N}, 2^k \geq n\}$, and consider the product of $\tilde{n}$ numbers $x_1,\dots,x_{\tilde{n}} \in [-M,M]$. In case $n < \tilde{n}$, we define $x_{n+1},\dots,x_{\tilde{n}} := 1$, which can be implemented by a bias in the first layer. Because $\tilde{n} < 2n$, the bounds on network size and depth in terms of $\tilde{n}$ also hold in terms of n, possibly with a larger constant.

It suffices to show the result for $M = 1$, since for $M > 1$, the network defined through $\tilde{\prod}_{\delta,M}(x_1,\dots,x_n) := M^n\tilde{\prod}_{\delta/M^n,1}(x_1/M,\dots,x_n/M)$ for all $(x_i)_{i=1}^n \in [-M,M]^n$ achieves the desired bounds as is easily verified. Therefore, w.l.o.g. $M = 1$ throughout the rest of this proof.

Equation (2.12a) follows by the argument given in the proof of [33, Proposition 3.3], we recall it here for completeness. By abuse of notation, for every even $k \in \mathbb{N}$ let a (k-dependent) mapping $R = R^1$ be defined via

$$R(y_1,\dots,y_k) := \left(\tilde{\times}_{\delta/\tilde{n}^2,2}(y_1,y_2),\dots,\tilde{\times}_{\delta/\tilde{n}^2,2}(y_{k-1},y_k)\right) \in \mathbb{R}^{k/2}\,. \tag{2.14}$$

For $\ell \geq 2$ set $R^\ell := R \circ R^{\ell-1}$. That is, for each product network $\tilde{\times}_{\delta/\tilde{n}^2,2}$ as in Proposition 2.5 we choose maximum input size "$M = 2$" and accuracy "$\delta/\tilde{n}^2$". Hence R^ℓ can be interpreted as a mapping from $\mathbb{R}^{2^\ell} \to \mathbb{R}$. We now define $\tilde{\prod}_{\delta,1} : [-1,1]^n \to \mathbb{R}$ via

$$\tilde{\prod}_{\delta,1}(x_1,\dots,x_n) := R^{\log_2(\tilde{n})}(x_1,\dots,x_{\tilde{n}})$$

and next show the error bounds in (2.12) (recall that by definition $x_{n+1} = \cdots = x_{\tilde{n}} = 1$ in case $\tilde{n} > n$).

First, by induction we show that for $\ell \in \{1, \dots, \log_2(\tilde{n})\}$ and for all $x_1, \dots, x_{2^\ell} \in [-1, 1]$

$$\left| \prod_{j=1}^{2^\ell} x_j - R^\ell(x_1, \dots, x_{2^\ell}) \right| \leq \delta \frac{2^{2\ell}}{\tilde{n}^2}. \tag{2.15}$$

For $\ell = 1$ it holds that $R(x_1, x_2) = \tilde{\times}_{\delta/\tilde{n}^2,2}(x_1, x_2)$, hence (2.15) follows directly from the choice for the accuracy of $\tilde{\times}_{\delta/\tilde{n}^2,2}$, which is $\delta/\tilde{n}^2$. For $\ell \in \{2, \dots, \log_2(\tilde{n})\}$, we assume that Equation (2.15) holds for $\ell - 1$. With $|\prod_{j=1}^{2^{(\ell-1)}} x_j| \leq 1$ and $\frac{2^{2(\ell-1)}}{\tilde{n}^2}\delta < 1$, it follows that $\left|R^{\ell-1}(x_1, \dots, x_{2^{(\ell-1)}})\right| < 2$, hence $R^{\ell-1}(x_1, \dots, x_{2^{(\ell-1)}})$ may be used as input of $\tilde{\times}_{\delta/\tilde{n}^2,2}$. We find

$$\begin{aligned}
\left| \prod_{j=1}^{2^\ell} x_j - R^\ell(x_1, \dots, x_{2^\ell}) \right| &\leq \left| \prod_{j=1}^{2^{\ell-1}} x_j - R^{\ell-1}(x_1, \dots, x_{2^{\ell-1}}) \right| \cdot \left| \prod_{j=2^{\ell-1}+1}^{2^\ell} x_j \right| \\
&\quad + \left|R^{\ell-1}(x_1, \dots, x_{2^{\ell-1}})\right| \cdot \left| \prod_{j=2^{\ell-1}+1}^{2^\ell} x_j - R^{\ell-1}(x_{2^{\ell-1}+1}, \dots, x_{2^\ell}) \right| \\
&\quad + \Big| R^{\ell-1}(x_1, \dots, x_{2^{\ell-1}}) R^{\ell-1}(x_{2^{\ell-1}+1}, \dots, x_{2^\ell}) \\
&\quad - \tilde{\times}_{\delta/\tilde{n}^2,2}\left(R^{\ell-1}(x_1, \dots, x_{2^{\ell-1}}), R^{\ell-1}(x_{2^{\ell-1}+1}, \dots, x_{2^\ell})\right) \Big| \\
&\leq \frac{2^{2(\ell-1)}}{\tilde{n}^2}\delta + \frac{2^{2(\ell-1)}}{\tilde{n}^2}\delta \left(1 + \frac{2^{2(\ell-1)}}{\tilde{n}^2}\delta\right) + \frac{1}{\tilde{n}^2}\delta \\
&\leq \frac{2^{2(\ell-1)} + 2 \cdot 2^{2(\ell-1)} + 1}{\tilde{n}^2}\delta \leq \frac{2^{2\ell}}{\tilde{n}^2}\delta,
\end{aligned}$$

where we used $(1 + \delta 2^{2(\ell-1)}/\tilde{n}^2) \leq 2$. This shows (2.15) for ℓ. Inserting $\ell = \log_2(\tilde{n})$ into (2.15) gives (2.12a).

We next show (2.12b). Without loss of generality, we only consider the derivative with respect to x_1, because each $\tilde{\times}_{\delta/\tilde{n}^2,2}$-network is symmetric under permutations of its arguments. For $\ell \in \{1, \dots, \log_2(\tilde{n})\}$ we show by induction that for almost every $(x_i)_{i=1}^{2^\ell} \in [-1, 1]^{2^\ell}$

$$\left| \frac{\partial}{\partial x_i} \prod_{j=1}^{2^\ell} x_j - \frac{\partial}{\partial x_i} R^\ell(x_1, \dots, x_{2^\ell}) \right| \leq \delta \frac{2^{2\ell}}{\tilde{n}^2}. \tag{2.16}$$

Again, $R(x_1, x_2) = \tilde{\times}_{\delta/\tilde{n}^2,2}(x_1, x_2)$ and for $\ell = 1$ Equation (2.16) follows from Proposition 2.5 and the choice for the accuracy of $\tilde{\times}_{\delta/\tilde{n}^2,2}$, which is $\delta/\tilde{n}^2$.

For $\ell > 1$, under the assumption that (2.16) holds for $\ell - 1$, we find

$$
\begin{aligned}
&\left| \frac{\partial}{\partial x_1} \prod_{j=1}^{2^\ell} x_j - \frac{\partial}{\partial x_1} R^\ell(x_1, \dots, x_{2^\ell}) \right| \\
&\leq \left| \prod_{j=2^{\ell-1}+1}^{2^\ell} x_j \right| \cdot \left| \frac{\partial}{\partial x_1} \prod_{j=1}^{2^{\ell-1}} x_j - \frac{\partial}{\partial x_1} R^{\ell-1}(x_1, \dots, x_{2^{\ell-1}}) \right| \\
&\quad + \left| \prod_{j=2^{\ell-1}+1}^{2^\ell} x_j - R^{\ell-1}(x_{2^{\ell-1}+1}, \dots, x_{2^\ell}) \right| \cdot \left| \frac{\partial}{\partial x_1} R^{\ell-1}(x_1, \dots, x_{2^{\ell-1}}) \right| \\
&\quad + \left| R^{l-1}(x_{2^{\ell-1}+1}, \dots, x_{2^\ell}) - \left(\frac{\partial}{\partial a} \tilde{\times}_{\delta/\tilde{n}^2,2} \right) \left(R^{l-1}(x_1, \dots, x_{2^{\ell-1}}), R^{l-1}(x_{2^{\ell-1}+1}, \dots, x_{2^\ell}) \right) \right| \\
&\quad \cdot \left| \frac{\partial}{\partial x_1} R^{\ell-1}(x_1, \dots, x_{2^{\ell-1}}) \right| \\
&\leq \frac{2^{2(\ell-1)}}{\tilde{n}^2}\delta + \frac{2^{2(\ell-1)}}{\tilde{n}^2}\delta \left(1 + \frac{2^{2(\ell-1)}}{\tilde{n}^2}\delta\right) + \frac{1}{\tilde{n}^2}\delta \left(1 + \frac{2^{2(\ell-1)}}{\tilde{n}^2}\delta\right) \\
&\leq \frac{2^{2(\ell-1)} + 2 \cdot 2^{2(\ell-1)} + 2}{\tilde{n}^2}\delta \leq \frac{2^{2\ell}}{\tilde{n}^2}\delta,
\end{aligned}
$$

where $\frac{\partial}{\partial a} \tilde{\times}_{\delta/\tilde{n}^2,2}$ denotes the (weak) derivative of $\tilde{\times}_{\delta/\tilde{n}^2,2} : [-2,2] \times [-2,2] \to \mathbb{R}$ w.r.t. its first argument as in Proposition 2.5. This shows (2.16) for $\ell > 1$, as desired. Filling in $\ell = \log_2(\tilde{n})$ gives (2.12b).

The number of binary tree layers (each denoted by R) is bounded by $O(\log_2(\tilde{n}))$. With the bound on the network depth from Proposition 2.5, for $M = 1$ the second part of (2.13) follows.

To estimate the network size, we cannot use the estimate $\operatorname{size}(f \circ g) \leq 2\operatorname{size}(f) + 2\operatorname{size}(g)$ from Equation (2.9), because the number of concatenations $\log_2(\tilde{n}) - 1$ depends on n, hence the factors 2 would give an extra n-dependent factor in the estimate on the network size. Instead, from Equation (2.9) we use $\operatorname{size}(f \circ g) \leq \operatorname{size}(f) + \operatorname{size}_{\text{in}}(f) + \operatorname{size}_{\text{out}}(g) + \operatorname{size}(g)$ and the bounds from Proposition 2.5. We find ($2^{\log_2(\tilde{n})-\ell}$ being the number of product networks in binary tree layer ℓ)

$$
\begin{aligned}
\operatorname{size}(\tilde{\prod}_{\delta,1}) &\leq \sum_{\ell=1}^{\log_2(\tilde{n})} 2^{\log_2(\tilde{n})-\ell} \left(\operatorname{size}_{\text{in}} \left(\tilde{\times}_{\delta/\tilde{n}^2,2} \right) + \operatorname{size} \left(\tilde{\times}_{\delta/\tilde{n}^2,2} \right) + \operatorname{size}_{\text{out}} \left(\tilde{\times}_{\delta/\tilde{n}^2,2} \right) \right) \\
&\leq \sum_{\ell=1}^{\log_2(\tilde{n})} 2^{\log_2(\tilde{n})-\ell} \left(C + C\big(1 + \log\left(2\tilde{n}^2/\delta\right)\big) + C \right) \\
&\leq (\tilde{n} - 1) C \big(1 + \log\left(\tilde{n}/\delta\right)\big) \leq C(1 + n\log(n/\delta)),
\end{aligned}
$$

which finishes the proof of (2.13) for $M = 1$. □

The previous two propositions can be used to deduce bounds on the approximation of univariate polynomials on compact intervals w.r.t. the norm $W^{1,\infty}$. One such result was already proven in [25, Proposition 4.2], which we present in Proposition 2.9 in a slightly modified form, allowing for the simultaneous approximation of multiple polynomials reusing the same approximate monomial basis. This yields a smaller network, and thus gives a slight improvement over using the parallelization of networks obtained by applying [25, Proposition 4.2] to each polynomial separately. To prove the result we first recall the following lemma:

Lemma 2.7 ([25, Lemma 4.5]). *For all $\ell \in \mathbb{N}$ and $\delta \in (0,1)$ there exists a σ_1-NN $\tilde{\mathbf{\Psi}}_\delta^\ell$ with input dimension one and output dimension $2^{\ell-1}+1$ such that*

$$\max_{\ell=2^{\ell-1},\dots,2^\ell} \left\| x^\ell - (\tilde{\mathbf{\Psi}}_\delta^\ell)_{1+\ell-2^{\ell-1}} \right\|_{W^{1,\infty}([-1,1])} \leq \delta, \tag{2.17}$$

$$\operatorname{depth}(\tilde{\mathbf{\Psi}}_\delta^\ell) \leq C(\ell^3 + \ell \log_2(1/\delta)), \qquad \operatorname{size}(\tilde{\mathbf{\Psi}}_\delta^\ell) \leq C(\ell 2^\ell + 2^\ell \log_2(1/\delta)), \tag{2.18}$$

where C is independent of ℓ and δ.

Corollary 2.8. *Let $n \in \mathbb{N}$ and $\delta \in (0,1)$. There exists a NN $\mathbf{\Psi}_\delta^n$ with input dimension one and output dimension $n+1$ such that $(\mathbf{\Psi}_\delta^n(x))_1 = 1$ and $(\mathbf{\Psi}_\delta^n(x))_2 = x$ for all $x \in \mathbb{R}$, and*

$$\max_{\ell \in \{3,\dots,n+1\}} \left\| x^{\ell-1} - (\mathbf{\Psi}_\delta^n)_\ell \right\|_{W^{1,\infty}([-1,1])} \leq \delta, \tag{2.19a}$$

and

$$\operatorname{size}(\mathbf{\Psi}_\delta^n) \leq C(1 + n\log(n) + n\log(1/\delta)), \qquad \operatorname{depth}(\mathbf{\Psi}_\delta^n) \leq C(1 + \log(n)^3 + \log(n)\log(1/\delta)), \tag{2.19b}$$

where C is independent of n and δ.

Proof. Define $k := \lceil \log_2(n) \rceil$ and for $\ell \in \{1,\dots,k\}$ let $\phi_\ell : \mathbb{R} \to \mathbb{R}$ be an identity network with $\operatorname{depth}(\phi_\ell) = \max_{i\in\{1,\dots,k\}} \operatorname{depth}(\tilde{\mathbf{\Psi}}_\delta^i) - \operatorname{depth}(\tilde{\mathbf{\Psi}}_\delta^\ell)$ as in (2.6). Set

$$\hat{\mathbf{\Psi}}_\delta^n := \left(\tilde{\mathbf{\Psi}}_\delta^1 \circ \phi_1, \dots, \tilde{\mathbf{\Psi}}_\delta^k \circ \phi_k \right).$$

Then by Lemma 2.7, $\hat{\mathbf{\Psi}}_\delta^n(x)$ is an approximation to

$$(\underbrace{x^1, x^2}_{\tilde{\mathbf{\Psi}}_\delta^1 \circ \phi_1}, \underbrace{x^2, \dots, x^4}_{\tilde{\mathbf{\Psi}}_\delta^2 \circ \phi_2}, x^4, \dots, x^{2^{k-1}}, \underbrace{x^{2^{k-1}}, \dots, x^{2^k}}_{\tilde{\mathbf{\Psi}}_\delta^k \circ \phi_k}),$$

where the braces indicate which part of the network approximates these outputs. Adding one layer to eliminate the double entries and (in case $2^k > n$) the approximations x^k with $k > n$, and adding the first entry which always equals $1 = x^0$, we obtain a network $\mathbf{\Psi}^n : \mathbb{R} \to \mathbb{R}^{n+1}$ satisfying (2.19a). The depth bound is an immediate consequence of $\operatorname{depth}(\mathbf{\Psi}_\delta^n) \leq C + \max_{i\in\{1,\dots,k\}} \operatorname{depth}(\tilde{\mathbf{\Psi}}_\delta^i)$, (2.18) and $k \leq C\log(n)$. To bound the size, first note that by (2.6) and (2.18) holds $\operatorname{size}(\phi_\ell) \leq C(k^3 + k\log(1/\delta))$ for a constant $C > 0$ independent of n and δ. Thus

$$\begin{aligned}
\operatorname{size}(\mathbf{\Psi}_\delta^n) &\leq C(n+1) + \operatorname{size}(\hat{\mathbf{\Psi}}_\delta^n) \leq Cn + C\sum_{\ell=1}^k (\operatorname{size}(\tilde{\mathbf{\Psi}}_\delta^\ell) + \operatorname{size}(\phi_\ell)) \\
&\leq Cn + C\sum_{\ell=1}^k \left(\ell 2^\ell + 2^\ell \log(1/\delta) + (k^3 + k\log(1/\delta)) \right) \\
&\leq C(n + n\log(n) + n\log(1/\delta)),
\end{aligned}$$

where we used $k \leq C\log(n)$ and $n \geq 1$. This shows (2.19b). □

Proposition 2.9. *There exists a constant $C > 0$ such that the following holds: For every $\delta > 0$, $n \in \mathbb{N}_0$, $N \in \mathbb{N}$ and N polynomials $p_i = \sum_{j=0}^n c_j^i y^j \in \mathbb{P}_n$, $i = 1,\dots,N$ there exists a σ_1-NN $\tilde{\boldsymbol{p}}_\delta : [-1,1] \to \mathbb{R}^N$ such that*

$$\max_{i=1,\dots,N} \|p_i - (\tilde{\boldsymbol{p}}_\delta)_i\|_{W^{1,\infty}([-1,1])} \leq \delta$$

and, with $C_0 := \max\{\max_{i=1,\dots,N} \sum_{j=2}^n |c_j^i|, \delta\}$,

$$\operatorname{size}(\tilde{\boldsymbol{p}}_\delta) \leq C(1 + nN + n\log(n) + n\log(C_0/\delta)), \quad \operatorname{depth}(\tilde{\boldsymbol{p}}_\delta) \leq C(1 + \log(n)^3 + \log(n)\log(C_0/\delta)).$$

Proof. We merely have to apply a linear transformation to the network in Cor. 2.8. Specifically, let $\Phi : \mathbb{R}^n \to \mathbb{R}^N$ be the network expressing the linear function with ith component $(\Phi(\boldsymbol{x}))_i = \sum_{j=0}^{i} c_j^i x_{j+1}$, where $\boldsymbol{x} = (x_i)_{i=1}^{n+1}$. In other words, with $W \in \mathbb{R}^{N\times(n+1)}$ given by $W_{i\ell} = c_{\ell-1}^i$, Φ is the depth 0 ReLU NN $\Phi(\boldsymbol{x}) = W\boldsymbol{x}$ of size at most $N(n+1)$. Then by (2.19a),

$$\tilde{\boldsymbol{p}}_\delta := \Phi \circ \boldsymbol{\Psi}^n_{\delta/C_0}$$

satisfies for each $i \in \{1, \dots, N\}$

$$\|p_i - (\tilde{\boldsymbol{p}}_\delta)_i\|_{W^{1,\infty}([-1,1])} \le \sum_{\ell=0}^{n} c_\ell^i \|x^\ell - (\boldsymbol{\Psi}^n_{\delta/C_0}(x))_{\ell+1}\|_{W^{1,\infty}([-1,1])} \le \sum_{\ell=2}^{n} \frac{\delta}{C_0} c_\ell^i \le \delta.$$

By (2.19b)

$$\operatorname{size}(\tilde{\boldsymbol{p}}_\delta) \le C(\operatorname{size}(\Phi) + \operatorname{size}(\boldsymbol{\Psi}^n_{\delta/C_0}) \le C(1 + nN + n\log(n) + n\log(C_0/\delta)),$$

and finally

$$\operatorname{depth}(\tilde{\boldsymbol{p}}_\delta) \le \operatorname{depth}(\Phi) + 1 + \operatorname{depth}(\tilde{\boldsymbol{\Psi}}^n_{\delta/C_0}) \le C(1 + \log(n)^3 + \log(n)\log(C_0/\delta)). \qquad \square$$

Remark 2.10. *If $y_0 \in \mathbb{R}$ and $p_i(y) = \sum_{j=0}^n c_j^i (y - y_0)^j$, $i = 1, \dots, N$, then Proposition 2.9 can still be applied for the approximation of $p_i(y)$ for $y \in [y_0 - 1, y_0 + 1]$, since the substitution $z = y - y_0$ corresponds to a shift, which can be realized exactly in the first layer of a NN, cp. (2.1). Thus, if $q_i(z) := \sum_{j=0}^n c_j^i z^j$ and if $\|q_i - (\tilde{\boldsymbol{q}}_\delta)_i\|_{W^{1,\infty}([-1,1])} \le \delta$ as in Proposition 2.9, then $y \mapsto \tilde{\boldsymbol{p}}_\delta(y) := \tilde{\boldsymbol{q}}_\delta(y - y_0)$ is a NN satisfying the accuracy and size bounds of Proposition 2.9 w.r.t. the $[W^{1,\infty}([y_0 - 1, y_0 + 1])]^N$ norm.*

2.3.2 ReLU DNN approximation of univariate Legendre polynomials

For $j \in \mathbb{N}_0$ we denote by L_j the jth Legendre polynomial, normalized in $L^2([-1,1], \lambda/2)$, where $\lambda/2$ denotes the uniform probability measure on $[-1,1]$. For $j \in \mathbb{N}_0$ it holds that $L_j(x) = \sum_{\ell=0}^j c_\ell^j x^\ell$, where, with $m(\ell) := (j-\ell)/2$,

$$c_\ell^j = \begin{cases} 0 & j - \ell \in \{0, \dots, j\} \cap 2\mathbb{Z} + 1, \\ (-1)^m 2^{-j} \binom{j}{m} \binom{j+\ell}{j} \sqrt{2j+1} & j - \ell \in \{0, \dots, j\} \cap 2\mathbb{Z}, \end{cases} \tag{2.20}$$

see e.g. [12, Section 10.10 Equation (16)], (the factor $\sqrt{2j+1}$ is needed to obtain the desired normalization). We define $c_\ell^j := 0$ for $\ell > j$.

Analogous to [25, Equation (4.13)] it holds that $\sum_{\ell=0}^j |c_\ell^j| \le 4^j$ for all $j \in \mathbb{N}$ (we use that $\sqrt{2j+1} \le \sqrt{\pi j}$). Inserting this into Proposition 2.9 with $N = n$ and $p_i = L_i$ for $i = 1, \dots, N$, we find the following result on the approximation of univariate Legendre polynomials by σ_1-NNs (similar to [23, Proposition 2.5] for the approximation of Chebyšev polynomials).

Proposition 2.11 ([25, Proposition 4.2 and Equation (4.13)])**.** *For every $n \in \mathbb{N}$ and for every $\delta \in (0,1)$ there exists a σ_1-NN $\tilde{\boldsymbol{L}}_{n,\delta}$ with input dimension one and with output dimension n such that for a positive constant C independent of n and δ there holds*

$$\begin{aligned} \|L_j - (\tilde{\boldsymbol{L}}_{n,\delta})_j\|_{W^{1,\infty}([-1,1])} &\le \delta, \qquad j = 1, \dots, n, \\ \operatorname{depth}(\tilde{\boldsymbol{L}}_{n,\delta}) &\le C(1 + \log_2 n)\big(n + \log_2(1/\delta)\big), \\ \operatorname{size}(\tilde{\boldsymbol{L}}_{n,\delta}) &\le Cn\big(n + \log_2(1/\delta)\big). \end{aligned} \tag{2.21}$$

Remark 2.12. *Alternatively, the σ_1-NN approximation of Legendre polynomials of degree n could be based on the three term recursion formula for Legendre polynomials or the Horner scheme for polynomials in general, by concatenating n product networks from Proposition 2.5 (and affine transformations). Because, depending on the scaling of the Legendre polynomials, either the accuracy δ of the product networks or the maximum input size M needs to grow exponentially with n, both the network depth and the network size of the resulting NN approximation of univariate Legendre polynomials would be bounded by $Cn(n+\log(1/\delta))$. That network size is of the same order as in Proposition 2.11, but the network depth has a worse dependence on the polynomial degree n. For more details, see [23, Proposition 2.5], where this construction is used to approximate truncated Chebyšev expansions based on the three term recursion for Chebyšev polynomials, which is very similar to that for Legendre polynomials.*

For future reference, we note that by (2.21) and Equation (2.23) below, for all $n \in \mathbb{N}$, $j = 1, \dots, n$, $\delta \in (0,1)$ and $k \in \{0,1\}$

$$\|(\tilde{\boldsymbol{L}}_{n,\delta})_j\|_{W^{k,\infty}([-1,1])} \leq (2j+1)^{1/2+2k} + \delta \leq (2j+1)^{1/2+2k} + 1 \leq (2j+2)^{2k+1}. \tag{2.22}$$

2.3.3 ReLU DNN approximation of tensor product Legendre polynomials

Let $d \in \mathbb{N}$. Denote the uniform probability measure on $[-1,1]^d$ by μ_d, i.e. $\mu_d := 2^{-d}\lambda$ where λ is the Lebesgue measure on $[-1,1]^d$. Then, for all $\boldsymbol{\nu} \in \mathbb{N}_0^d$ the tensorized Legendre polynomials $L_{\boldsymbol{\nu}}(\boldsymbol{y}) := \prod_{j=1}^d L_{\nu_j}(y_j)$ form a μ_d-orthonormal basis of $L^2([-1,1]^d, \mu_d)$. We shall require the following bound on the norm of the tensorized Legendre polynomials which itself is a consequence of the Markoff inequality, and our normalization of the Legendre polynomials: for any $k \in \mathbb{N}_0$

$$\forall \boldsymbol{\nu} \in \mathbb{N}_0^d: \quad \|L_{\boldsymbol{\nu}}\|_{W^{k,\infty}([-1,1]^d)} \leq \prod_{j=1}^d (1+2\nu_j)^{1/2+2k}. \tag{2.23}$$

To provide bounds on the size of the networks approximating the tensor product Legendre polynomials, for finite subsets $\Lambda \subset \mathbb{N}_0^d$ we will make use of the quantity

$$m(\Lambda) := \max_{\boldsymbol{\nu} \in \Lambda} |\boldsymbol{\nu}|_1. \tag{2.24}$$

Proposition 2.13. *For every finite subset $\Lambda \subset \mathbb{N}_0^d$ and every $\delta \in (0,1)$ there exists a σ_1-NN $\boldsymbol{f}_{\Lambda,\delta}$ with input dimension d and output dimension $|\Lambda|$, such that the outputs $\{\tilde{L}_{\boldsymbol{\nu},\delta}\}_{\boldsymbol{\nu}\in\Lambda}$ of $\boldsymbol{f}_{\Lambda,\delta}$ satisfy*

$$\begin{aligned} \forall \boldsymbol{\nu} \in \Lambda: \quad & \|L_{\boldsymbol{\nu}} - \tilde{L}_{\boldsymbol{\nu},\delta}\|_{W^{1,\infty}([-1,1]^d)} \leq \delta, \\ & \sup_{\boldsymbol{y} \in [-1,1]^d} |\tilde{L}_{\boldsymbol{\nu},\delta}((y_j)_{j \in \operatorname{supp} \boldsymbol{\nu}})| \leq (2m(\Lambda)+2)^d, \end{aligned}$$

and for a constant $C > 0$ that is independent of d, Λ and δ it holds

$$\begin{aligned} \operatorname{depth}(\boldsymbol{f}_{\Lambda,\delta}) &\leq C(1+d\log d)(1+\log_2 m(\Lambda))\big(m(\Lambda)+\log_2(1/\delta)\big), \\ \operatorname{size}(\boldsymbol{f}_{\Lambda,\delta}) &\leq Cd^2 m(\Lambda)^2 + Cdm(\Lambda)\log_2(1/\delta) + Cd^2|\Lambda|\big(1+\log_2 m(\Lambda)+\log_2(1/\delta)\big). \end{aligned}$$

Proof. Let $\delta \in (0,1)$ and a finite subset $\Lambda \subset \mathbb{N}_0^d$ be given.

The proof is divided into three steps. In the first step, we define ReLU NN approximations of tensor product Legendre polynomials $\{\tilde{L}_{\boldsymbol{\nu},\delta}\}_{\boldsymbol{\nu}\in\Lambda}$ and fix the parameters used in the NN approximation. In the second step, we estimate the error of the approximation, and the $L^\infty([-1,1]^d)$-norm of the $\tilde{L}_{\boldsymbol{\nu},\delta}$, $\boldsymbol{\nu} \in \Lambda$. In the third step, we describe the network $\boldsymbol{f}_{\Lambda,\delta}$ and estimate its depth and size.

Step 1. For all $\boldsymbol{\nu} \in \mathbb{N}_0^d$, we define $n_{\boldsymbol{\nu}} := |\operatorname{supp} \boldsymbol{\nu}|$ and $M_{\boldsymbol{\nu}} := 2|\boldsymbol{\nu}|_1 + 2$. We can now define

$$\tilde{L}_{\boldsymbol{\nu},\delta}((y_j)_{j\in\operatorname{supp}\boldsymbol{\nu}}) := \tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2,M_{\boldsymbol{\nu}}} \left(\left\{ (\tilde{\boldsymbol{L}}_{m(\Lambda),\delta'}(y_j))_{\nu_j} \right\}_{j\in\operatorname{supp}\boldsymbol{\nu}} \right), \tag{2.25}$$

where $\tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2,M_{\boldsymbol{\nu}}} : [-M_{\boldsymbol{\nu}}, M_{\boldsymbol{\nu}}]^{|\operatorname{supp}\boldsymbol{\nu}|} \to \mathbb{R}$ is as in Proposition 2.6. For the network approximating univariate Legendre polynomials $\tilde{\boldsymbol{L}}_{m(\Lambda),\delta'}$ from Proposition 2.11, we set the accuracy parameter as $\delta' := \frac{1}{2}d^{-1}(2m(\Lambda)+2)^{-d-1}\delta < 1$. Let us point out that by (2.22) for all $\boldsymbol{\nu} \in \mathbb{N}_0^d$ and all $j \in \operatorname{supp}\boldsymbol{\nu}$

$$\|(\tilde{\boldsymbol{L}}_{m(\Lambda),\delta'})_{\nu_j}\|_{L^\infty([-1,1])} \leq 2\nu_j + 2 \leq 2|\boldsymbol{\nu}|_1 + 2 = M_{\boldsymbol{\nu}} \leq 2m(\Lambda) + 2,$$

so that, as required by Proposition 2.6, the absolute values of the arguments of $\tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2,M_{\boldsymbol{\nu}}}$ in (2.25) are all bounded by $M_{\boldsymbol{\nu}}$.

Step 2. For the $L^\infty([-1,1])$-error of $\tilde{L}_{\boldsymbol{\nu},\delta}$ we find

$$\begin{aligned}
&\sup_{\boldsymbol{y}\in[-1,1]^d} \left| L_{\boldsymbol{\nu}}(\boldsymbol{y}) - \tilde{L}_{\boldsymbol{\nu},\delta}((y_j)_{j\in\operatorname{supp}\boldsymbol{\nu}}) \right| \\
&\leq \sup_{\boldsymbol{y}\in[-1,1]^d} \left| L_{\boldsymbol{\nu}}(\boldsymbol{y}) - \prod_{j\in\operatorname{supp}\boldsymbol{\nu}} (\tilde{\boldsymbol{L}}_{m(\Lambda),\delta'}(y_j))_{\nu_j} \right| \\
&\quad + \sup_{\boldsymbol{y}\in[-1,1]^d} \left| \prod_{j\in\operatorname{supp}\boldsymbol{\nu}} (\tilde{\boldsymbol{L}}_{m(\Lambda),\delta'}(y_j))_{\nu_j} - \tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2,M_{\boldsymbol{\nu}}} \left(\left\{ (\tilde{\boldsymbol{L}}_{m(\Lambda),\delta'}(y_j))_{\nu_j} \right\}_{j\in\operatorname{supp}\boldsymbol{\nu}} \right) \right| \\
&\leq \sup_{\boldsymbol{y}\in[-1,1]^d} \sum_{k\in\operatorname{supp}\boldsymbol{\nu}} \left| \prod_{\substack{j\in\operatorname{supp}\boldsymbol{\nu}:\\ j<k}} (\tilde{\boldsymbol{L}}_{m(\Lambda),\delta'}(y_j))_{\nu_j} \right| \cdot \left| L_{\nu_k}(y_k) - (\tilde{\boldsymbol{L}}_{m(\Lambda),\delta'}(y_k))_{\nu_k} \right| \cdot \left| \prod_{\substack{j\in\operatorname{supp}\boldsymbol{\nu}:\\ j>k}} L_{\nu_j}(y_j) \right| \\
&\quad + \frac{\delta}{2M_{\boldsymbol{\nu}}^3}.
\end{aligned}$$

Using Proposition 2.13, (2.22), (2.23) and $M_{\boldsymbol{\nu}} = 2|\boldsymbol{\nu}|_1 + 2 \leq 2m(\Lambda) + 2$, the last term can be bounded by

$$|\operatorname{supp}\boldsymbol{\nu}| M_{\boldsymbol{\nu}}^{n_\nu - 1}\delta' + \frac{\delta}{2} \leq \frac{|\operatorname{supp}\boldsymbol{\nu}|}{d} \frac{M_{\boldsymbol{\nu}}^{n_\nu-1}}{(2m(\Lambda)+2)^{d+1}} \frac{\delta}{2} + \frac{\delta}{2} \leq \delta.$$

It follows that for all $\boldsymbol{\nu} \in \Lambda$

$$\begin{aligned}
\sup_{\boldsymbol{y}\in[-1,1]^d} \left| \tilde{L}_{\boldsymbol{\nu},\delta}((y_j)_{j\in\operatorname{supp}\boldsymbol{\nu}}) \right| &\leq \sup_{\boldsymbol{y}\in[-1,1]^d} |L_{\boldsymbol{\nu}}(\boldsymbol{y})| + \sup_{\boldsymbol{y}\in[-1,1]^d} \left| L_{\boldsymbol{\nu}}(\boldsymbol{y}) - \tilde{L}_{\boldsymbol{\nu},\delta}((y_j)_{j\in\operatorname{supp}\boldsymbol{\nu}}) \right| \\
&\leq \prod_{j=1}^d (1+2\nu_j)^{1/2} + \delta \\
&\leq \prod_{j=1}^d (1+2\nu_j)^{1/2} + 1 \leq M_{\boldsymbol{\nu}}^d.
\end{aligned}$$

To determine the error of the gradient, without loss of generality we only consider the derivative with respect to y_1. In the case $1 \notin \operatorname{supp}\boldsymbol{\nu}$, we trivially have $\frac{\partial}{\partial y_1}(L_{\boldsymbol{\nu}}(\boldsymbol{y}) - \tilde{L}_{\boldsymbol{\nu},\delta}(\boldsymbol{y})) = 0$ for all

$\boldsymbol{y} \in [-1,1]^d$. Thus let $\nu_1 \neq 0$ in the following. Then, with $\delta' = \frac{1}{2}d^{-1}(2m(\Lambda)+2)^{-d-1}\delta$

$$
\begin{aligned}
&\sup_{\boldsymbol{y}\in[-1,1]^d} \left| \frac{\partial}{\partial y_1} L_{\boldsymbol{\nu}}(\boldsymbol{y}) - \frac{\partial}{\partial y_1} \tilde{L}_{\boldsymbol{\nu},\delta}((y_j)_{j\in\operatorname{supp}\boldsymbol{\nu}}) \right| \\
\leq &\sup_{\boldsymbol{y}\in[-1,1]^d} \left| \frac{\partial}{\partial y_1} L_{\boldsymbol{\nu}}(\boldsymbol{y}) - \frac{\partial}{\partial y_1} \prod_{j\in\operatorname{supp}\boldsymbol{\nu}} (\tilde{\boldsymbol{L}}_{m(\Lambda),\delta'}(y_j))_{\nu_j} \right| \\
&+ \sup_{\boldsymbol{y}\in[-1,1]^d} \left| \frac{\partial}{\partial y_1} \prod_{j\in\operatorname{supp}\boldsymbol{\nu}} (\tilde{\boldsymbol{L}}_{m(\Lambda),\delta'}(y_j))_{\nu_j} - \frac{\partial}{\partial y_1} \tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2, M_{\boldsymbol{\nu}}} \left(\left\{ (\tilde{\boldsymbol{L}}_{m(\Lambda),\delta'}(y_j))_{\nu_j} \right\}_{j\in\operatorname{supp}\boldsymbol{\nu}} \right) \right| \\
\leq &\sup_{\boldsymbol{y}\in[-1,1]^d} \left| \frac{\partial}{\partial y_1} L_{\nu_1}(y_1) - \frac{\partial}{\partial y_1} (\tilde{\boldsymbol{L}}_{m(\Lambda),\delta'}(y_1))_{\nu_1} \right| \cdot \left| \prod_{\substack{j\in\operatorname{supp}\boldsymbol{\nu}:\\ j>1}} L_{\nu_j}(y_j) \right| \\
&+ \sup_{\boldsymbol{y}\in[-1,1]^d} \sum_{1\neq k\in\operatorname{supp}\boldsymbol{\nu}} \left| \frac{\partial}{\partial y_1} (\tilde{\boldsymbol{L}}_{m(\Lambda),\delta'}(y_1))_{\nu_1} \right| \cdot \left| \prod_{\substack{1\neq j\in\operatorname{supp}\boldsymbol{\nu}:\\ j<k}} (\tilde{\boldsymbol{L}}_{m(\Lambda),\delta'}(y_j))_{\nu_j} \right| \\
&\cdot \left| L_{\nu_k}(y_k) - (\tilde{\boldsymbol{L}}_{m(\Lambda),\delta'}(y_k))_{\nu_k} \right| \cdot \left| \prod_{\substack{j\in\operatorname{supp}\boldsymbol{\nu}:\\ j>k}} L_{\nu_j}(y_j) \right| \\
&+ \sup_{\boldsymbol{y}\in[-1,1]^d} \left| \prod_{1\neq j\in\operatorname{supp}\boldsymbol{\nu}} (\tilde{\boldsymbol{L}}_{m(\Lambda),\delta'}(y_j))_{\nu_j} - \left(\frac{\partial}{\partial x_1} \tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2, M_{\boldsymbol{\nu}}} \right) \left(\left\{ (\tilde{\boldsymbol{L}}_{m(\Lambda),\delta'}(y_j))_{\nu_j} \right\}_{j\in\operatorname{supp}\boldsymbol{\nu}} \right) \right| \\
&\cdot \left| \frac{\partial}{\partial y_1} (\tilde{\boldsymbol{L}}_{m(\Lambda),\delta'}(y_1))_{\nu_1} \right|,
\end{aligned}
$$

where $\frac{\partial}{\partial x_1}\tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2,M_{\boldsymbol{\nu}}}$ denotes the (weak) derivative of $\tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2,M_{\boldsymbol{\nu}}} : [-M_{\boldsymbol{\nu}}, M_{\boldsymbol{\nu}}]^{|\operatorname{supp}\boldsymbol{\nu}|} \to \mathbb{R}$ with respect to its first argument, cf. Proposition 2.6.

Using (2.23) and Proposition 2.11 for the first term, Proposition 2.11, (2.22) and (2.23) for the second term and Proposition 2.6 and (2.23) for the third term, we further bound the NN approximation error by

$$\delta' M_{\boldsymbol{\nu}}^{n_{\boldsymbol{\nu}}-1} + (|\operatorname{supp}\boldsymbol{\nu}|-1) M_{\boldsymbol{\nu}}^3 M_{\boldsymbol{\nu}}^{n_{\boldsymbol{\nu}}-2}\delta' + \frac{\delta}{2M_{\boldsymbol{\nu}}^3} M_{\boldsymbol{\nu}}^3 \leq |\operatorname{supp}\boldsymbol{\nu}| M_{\boldsymbol{\nu}}^{n_{\boldsymbol{\nu}}+1} \tfrac{1}{2} d^{-1}(2m(\Lambda)+2)^{-d-1}\delta + \frac{\delta}{2} \leq \delta.$$

Step 3. We now describe the network $\boldsymbol{f}_{\Lambda,\delta}$, which in parallel emulates $\{\tilde{L}_{\boldsymbol{\nu},\delta}\}_{\boldsymbol{\nu}\in\Lambda}$. The network is constructed as the concatenation of two subnetworks, i.e.

$$\boldsymbol{f}_{\Lambda,\delta} = \boldsymbol{f}_{\Lambda,\delta}^{(1)} \circ \boldsymbol{f}_{\Lambda,\delta}^{(2)}.$$

The subnetwork $\boldsymbol{f}_{\Lambda,\delta}^{(2)}$ evaluates, in parallel, approximate univariate Legendre polynomials in the input variables $(y_j)_{j\leq d}$. It is defined as

$$\boldsymbol{f}_{\Lambda,\delta}^{(2)} := \left(\left\{ \tilde{\boldsymbol{L}}_{m(\Lambda),\delta'} \right\}_{j=1}^{d} \right),$$

where the pair of round brackets denotes a parallelization.

The subnetwork $\boldsymbol{f}^{(1)}_{\Lambda,\delta}$ takes the output of $\boldsymbol{f}^{(2)}_{\Lambda,\delta}$ as input and computes

$$\begin{aligned}\boldsymbol{f}_{\Lambda,\delta}\left((y_j)_{j\le d}\right) &= \boldsymbol{f}^{(1)}_{\Lambda,\delta}\left(\boldsymbol{f}^{(2)}_{\Lambda,\delta}\left((y_j)_{j\le d}\right)\right)\\ &= \left(\left\{\tilde{L}_{\boldsymbol{\nu},\delta}\left((y_j)_{j\le d}\right)\right\}_{\boldsymbol{\nu}\in\Lambda}\right)\\ &= \left(\left\{\mathrm{Id}_{\mathbb{R}}\circ\tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2,M_{\boldsymbol{\nu}}}\left(\left\{(\tilde{\boldsymbol{L}}_{m(\Lambda),\delta'}(y_j))_{\nu_j}\right\}_{j\in\operatorname{supp}\boldsymbol{\nu}}\right)\right\}_{\boldsymbol{\nu}\in\Lambda}\right),\end{aligned}$$

where in the last two lines the outer pair of round brackets denotes a parallelization. The depth of the identity networks is such that all components of the parallelization have equal depth.

We have the following expression for the network depth:

$$\operatorname{depth}(\boldsymbol{f}_{\Lambda,\delta}) = \operatorname{depth}\left(\boldsymbol{f}^{(1)}_{\Lambda,\delta}\right) + 1 + \operatorname{depth}\left(\boldsymbol{f}^{(2)}_{\Lambda,\delta}\right).$$

We can choose the depths of the identity networks in the definition of $\boldsymbol{f}^{(2)}_{\Lambda,\delta}$ such that (denoting here and in the remainder of this proof by $C>0$ constants independent of d, Λ and $\delta\in(0,1)$)

$$\begin{aligned}\operatorname{depth}\left(\boldsymbol{f}^{(2)}_{\Lambda,\delta}\right) &= \operatorname{depth}(\tilde{\boldsymbol{L}}_{m(\Lambda),\delta'})\\ &\le C(1+\log_2 m(\Lambda))\big(m(\Lambda)+\log_2(1/\delta')\big)\\ &\le C(1+\log_2 m(\Lambda))\big(m(\Lambda)+\log_2(d)+1+(d+1)\log_2(4m(\Lambda))+\log_2(1/\delta)\big)\\ &\le Cd(1+\log_2 m(\Lambda))\big(m(\Lambda)+\log_2(1/\delta)\big),\end{aligned}$$

where we used that $2m(\Lambda)+2\le 4m(\Lambda)$ when $\Lambda\neq\{\mathbf{0}\}$.

Similarly, due to $M_{\boldsymbol{\nu}} = 2|\boldsymbol{\nu}|_1+2\le 4m(\Lambda)$ (if $\Lambda\neq\{\mathbf{0}\}$), we can choose the identity networks in the definition of $\boldsymbol{f}^{(1)}_{\Lambda,\delta}$ such that

$$\begin{aligned}\operatorname{depth}\left(\boldsymbol{f}^{(1)}_{\Lambda,\delta}\right) &= 1+\max_{\boldsymbol{\nu}\in\Lambda}\operatorname{depth}\left(\tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2,M_{\boldsymbol{\nu}}}\right)\\ &\le \max_{\boldsymbol{\nu}\in\Lambda} C\left(1+\log_2(n_{\boldsymbol{\nu}})\log_2(n_{\boldsymbol{\nu}}M_{\boldsymbol{\nu}}^{n_{\boldsymbol{\nu}}+3}2/\delta)\right)\\ &\le C\max_{\boldsymbol{\nu}\in\Lambda}\left(1+\log_2(n_{\boldsymbol{\nu}})\big(\log_2 n_{\boldsymbol{\nu}}+1+(n_{\boldsymbol{\nu}}+3)\log_2(4m(\Lambda))+\log_2(1/\delta)\big)\right)\\ &\le C(1+d\log d)\big(1+\log_2 m(\Lambda)+\log_2(1/\delta)\big),\end{aligned}$$

where we used that $n_{\boldsymbol{\nu}}\le d$. Finally, we find the following bound on the network depth:

$$\operatorname{depth}(\boldsymbol{f}_{\Lambda,\delta})\le C(1+d\log d)(1+\log_2 m(\Lambda))\big(m(\Lambda)+\log_2(1/\delta)\big).$$

For the network size, we find that

$$\operatorname{size}(\boldsymbol{f}_{\Lambda,\delta})\le 2\operatorname{size}\left(\boldsymbol{f}^{(1)}_{\Lambda,\delta}\right)+2\operatorname{size}\left(\boldsymbol{f}^{(2)}_{\Lambda,\delta}\right).$$

With Proposition 2.11 we estimate the size of $\boldsymbol{f}^{(2)}_{\Lambda,\delta}$ as

$$\begin{aligned}\operatorname{size}\left(\boldsymbol{f}^{(2)}_{\Lambda,\delta}\right) &= d\operatorname{size}\left(\tilde{\boldsymbol{L}}_{m(\Lambda),\delta'}\right)\\ &\le Cdm(\Lambda)(m(\Lambda)+\log_2(1/\delta'))\\ &\le Cdm(\Lambda)\big(m(\Lambda)+\log_2(d)+1+(d+1)\log_2(4m(\Lambda))+\log_2(1/\delta)\big)\\ &\le Cd^2m(\Lambda)^2+Cdm(\Lambda)\log_2(1/\delta).\end{aligned}$$

The depth of each of the identity networks in the definition of $\boldsymbol{f}^{(1)}_{\Lambda,\delta}$ is bounded by $\mathrm{depth}(\boldsymbol{f}^{(1)}_{\Lambda,\delta}) \leq C(1+d\log d)\big(1+\log_2 m(\Lambda)+\log_2(1/\delta)\big)$. It follows that

$$\begin{aligned}
\mathrm{size}\left(\boldsymbol{f}^{(1)}_{\Lambda,\delta}\right) &= \sum_{\boldsymbol{\nu}\in\Lambda} \mathrm{size}\left(\mathrm{Id}_{\mathbb{R}} \circ \tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2, M_{\boldsymbol{\nu}}}\right) \\
&\leq \sum_{\boldsymbol{\nu}\in\Lambda} 2\,\mathrm{size}\left(\mathrm{Id}_{\mathbb{R}}\right) + 2\,\mathrm{size}\left(\tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2, M_{\boldsymbol{\nu}}}\right) \\
&\leq 4|\Lambda|\left(\mathrm{depth}\left(\boldsymbol{f}^{(1)}_{\Lambda,\delta}\right)+1\right) + C\sum_{\boldsymbol{\nu}\in\Lambda}\big(1+n_{\boldsymbol{\nu}}\log_2(n_{\boldsymbol{\nu}}M_{\boldsymbol{\nu}}^{n_{\boldsymbol{\nu}}+3}2/\delta)\big) \\
&\leq C(1+d\log d)|\Lambda|\big(1+\log_2 m(\Lambda)+\log_2(1/\delta)\big) + C(1+d\log d)|\Lambda| \\
&\quad + Cd\sum_{\boldsymbol{\nu}\in\Lambda}\big(1+(n_{\boldsymbol{\nu}}+3)\log_2(4m(\Lambda))+\log_2(1/\delta)\big) \\
&\leq Cd^2|\Lambda|\big(1+\log_2 m(\Lambda)+\log_2(1/\delta)\big).
\end{aligned}$$

Hence, we arrive at

$$\begin{aligned}
\mathrm{size}(\boldsymbol{f}_{\Lambda,\delta}) &\leq 2\,\mathrm{size}\left(\boldsymbol{f}^{(1)}_{\Lambda,\delta}\right) + 2\,\mathrm{size}\left(\boldsymbol{f}^{(2)}_{\Lambda,\delta}\right) \\
&\leq Cd^2 m(\Lambda)^2 + Cdm(\Lambda)\log_2(1/\delta) + Cd^2|\Lambda|\big(1+\log_2 m(\Lambda)+\log_2(1/\delta)\big). \qquad \square
\end{aligned}$$

2.4 RePU DNN emulation of polynomials

The approximation of polynomials by neural networks can be significantly simplified if instead of the ReLU activation σ_1 we consider as activation function the so-called *rectified power unit* ("RePU" for short): recall that for $r \in \mathbb{N}$, $r \geq 2$, the RePU activation is defined by $\sigma_r(x) = \max\{0,x\}^r$, $x \in \mathbb{R}$. In contrast to σ_1-NNs, as shown in [17], for every $r \in \mathbb{N}$, $r \geq 2$ there exist RePU networks of depth 1 realizing the multiplication of two real numbers *without error*. This yields the following result proven in [17, Theorem 4.1] for $r = 2$. With [17, Theorem 2.5] this extends to all $r \geq 2$. To render the presentation self-contained, an alternative proof is provided in Appendix A, based on ideas in [25]. Unlike in [17], it is shown that the constant C is independent of d. This is relevant in particular when considering RePU emulations of truncated polynomial chaos expansions of countably parametric maps $u : [-1,1]^{\mathbb{N}} \to \mathbb{R}$, shortly discussed in Section 4.3.3. Polynomial approximations of such maps depend on a finite number $d(\varepsilon) \in \mathbb{N}$ of parameters only, but with $d(\varepsilon) \to \infty$ as $\varepsilon \downarrow 0$.

Proposition 2.14. *Fix $d \in \mathbb{N}$ and $r \in \mathbb{N}$, $r \geq 2$. Then there exists a constant $C > 0$ independent of d but depending on r such that for any finite downward closed $\Lambda \subseteq \mathbb{N}_0^d$ and any $p \in \mathbb{P}_\Lambda$ there is a σ_r-network $\tilde{p} : \mathbb{R}^d \to \mathbb{R}$ which realizes p exactly and such that* $\mathrm{size}(\tilde{p}) \leq C|\Lambda|$ *and* $\mathrm{depth}(\tilde{p}) \leq C\log_2(|\Lambda|)$.

Remark 2.15. *Let $\psi : \mathbb{R} \to \mathbb{R}$ be an arbitrary C^2 function that is not linear, i.e. it does not hold $\psi''(x) = 0$ for all $x \in \mathbb{R}$. In [30] it is shown that ψ-networks can approximate the multiplication of two numbers a, b in a fixed bounded interval up to* arbitrary accuracy *with a* fixed number of units. *We also refer to [42, Section 3.3] where we explain this observation from [30] in more detail. From this, analogous to [17, Theorem 4.1], one can obtain a version of Proposition 2.14 for arbitrary C^2 activation functions. To state it, we fix $d \in \mathbb{N}$. Then there exists $C > 0$ (depending on d) such that for every $\delta > 0$, for every downward closed $\Lambda \subseteq \mathbb{N}_0^d$ and every $p \in \mathbb{P}_\Lambda$, there exists a ψ-neural network $q : [-M,M]^d \to \mathbb{R}$ such that $\sup_{x\in[-M,M]^d}|p(x)-q(x)| \leq \delta$,* $\mathrm{size}(q) \leq C|\Lambda|$ *and* $\mathrm{depth}(q) \leq C\log_2(|\Lambda|)$. *As discussed in Remark 2.1, the same also holds, e.g., for NNs with continuous, sigmoidal activation σ of order $k \geq 2$.*

Recently, interest has been shown in the approximation of ReLU NNs by rational functions and NNs with rational activation functions and vice versa, e.g. in [35, 3]. In the latter, $\sigma = p/q$ is used as activation for polynomials p, q of prescribed degree, but within each computational node trainable coefficients of p and q. For all prescribed $\deg(p) \geq 2$ and $\deg(q) \in \mathbb{N}_0$, each node in such a network can emulate the multiplication of two numbers exactly ([3, Proposition 10] and its proof), hence Proposition 2.14 also holds for such NNs (the proof in Appendix A applies, using that also the identity map can be emulated by networks with such activations).

As a result, Theorem 3.10 also holds for all activation functions discussed in this remark.

3 Exponential expression rate bounds

We now proceed to the statement and proof of the main result of the present note, namely the exponential rate bounds for the DNN expression of d-variate holomorphic maps. First, in Section 3.1 we recall (classical) polynomial approximation results for analytic functions, similar to those in [36]. Subsequently, these are used to deduce DNN approximation results for ReLU and RePU networks.

3.1 Polynomial approximation

Fix $d \in \mathbb{N}$. For $\rho > 1$ define the open Bernstein ellipse

$$\mathcal{E}_\rho := \left\{ \frac{z + z^{-1}}{2} \,:\, z \in \mathbb{C},\ 1 \leq |z| < \rho \right\} \subset \mathbb{C},$$

and for the poly-radius $\boldsymbol{\rho} = (\rho_j)_{j=1}^d \subseteq (1, \infty)^d$ define the poly-ellipse

$$\mathcal{E}_{\boldsymbol{\rho}} := \bigtimes_{j=1}^d \mathcal{E}_{\rho_j} \subseteq \mathbb{C}^d. \tag{3.1}$$

Let $u : [-1, 1]^d \to \mathbb{R}$ admit a complex holomorphic extension to the polyellipse $\mathcal{E}_{\boldsymbol{\rho}}$. Such a function can be approximated on $[-1, 1]^d$ by multivariate Legendre expansions, with the error decaying uniformly like $\exp(-\beta N^{1/d})$ for some $\beta > 0$ and in terms of the dimension N of the approximation space. This statement is made precise in Theorem 3.5 below.

Remark 3.1. *Suppose that $u : [-1, 1]^d \to \mathbb{R}$ is (real) analytic. Then it allows a complex holomorphic extension to some open set $O \subseteq \mathbb{C}^d$ containing $[-1, 1]^d$. Since for $\rho > 1$ close to 1, the maximal distance of a point in $\mathcal{E}_\rho$ to the interval $[-1, 1]$ becomes arbitrarily small, there always exists $\rho > 1$ such that u allows a holomorphic extension to $\bigtimes_{j=1}^d \mathcal{E}_\rho$.*

For the proof of the theorem we shall use the following result mentioned in [38].

Lemma 3.2. *Let $(a_j)_{j=1}^d \in (0, \infty)^d$. Then, with $a := \sum_{j=1}^d 1/a_j$*

$$\left| \left\{ \boldsymbol{\nu} \in \mathbb{N}_0^d \,:\, \sum_{j=1}^d \frac{\nu_j}{a_j} \leq 1 \right\} \right| \leq \frac{1}{d!} (1 + a)^d \prod_{j=1}^d a_j. \tag{3.2}$$

The lemma is proved by computing (as an upper bound of the left-hand side in (3.2)) the volume of the set $\{(x_j)_{j=1}^d \in \mathbb{R}_+^d \,:\, \sum_{j=1}^d \frac{(x_j - 1)}{a_j} \leq 1\}$, which equals the right-hand side in (3.2). The significance of this result is, that it provides an upper bound for the size of multiindex sets of the type

$$\Lambda_\varepsilon := \{\boldsymbol{\nu} \in \mathbb{N}_0^d \,:\, \boldsymbol{\rho}^{-\boldsymbol{\nu}} \geq \varepsilon\}, \qquad \varepsilon \in (0, 1). \tag{3.3}$$

To see this, note that due to $\log(\boldsymbol{\rho}^{-\boldsymbol{\nu}}) = -\sum_{j=1}^d \nu_j \log(\rho_j)$, for any $\varepsilon \in (0,1)$ we have

$$\Lambda_\varepsilon = \left\{ \boldsymbol{\nu} \in \mathbb{N}_0^d \,:\, \sum_{j=1}^d \nu_j \log(\rho_j) \le \log(1/\varepsilon) \right\}.$$

Applying Lemma 3.2 with $a_j = \log(1/\varepsilon)/\log(\rho_j)$ we thus get (see also [2, Lemma 4.2]):

Lemma 3.3. *It holds*

$$|\Lambda_\varepsilon| \le \frac{1}{d!} \left(\log(1/\varepsilon) + \sum_{j=1}^d \log(\rho_j) \right)^d \prod_{j=1}^d \frac{1}{\log(\rho_j)}. \tag{3.4}$$

Remark 3.4. *Note that*

$$\left\{ \boldsymbol{\nu} \in \mathbb{N}_0^d \,:\, 0 \le \nu_j \le \frac{-\log(\varepsilon)}{d\log(\rho_j)} \ \forall j \right\} \subseteq \Lambda_\varepsilon \subseteq \left\{ \boldsymbol{\nu} \in \mathbb{N}_0^d \,:\, 0 \le \nu_j \le \frac{-\log(\varepsilon)}{\log(\rho_j)} \ \forall j \right\}. \tag{3.5}$$

This implies the existence of a constant C (depending on $\boldsymbol{\rho}$ but independent of d) such that for all $\varepsilon \in (0,1)$ with $\rho_{\min} := \min_{j=1,\dots,d} \rho_j$ and $\rho_{\max} := \max_{j=1,\dots,d} \rho_j$ (cp. (2.24))

$$\begin{aligned} m(\Lambda_\varepsilon) &= \max\{|\boldsymbol{\nu}|_1 \,:\, \boldsymbol{\rho}^{-\boldsymbol{\nu}} \ge \varepsilon\} = \max\{n \in \mathbb{N}_0 \,:\, \rho_{\min}^{-n} \ge \varepsilon\} \\ &= \max\left\{ n \in \mathbb{N}_0 \,:\, n \le \frac{-\log(\varepsilon)}{\log(\rho_{\min})} \right\} \le d \frac{\log(\rho_{\max})}{\log(\rho_{\min})} \left(\prod_{j=1}^d \frac{-\log(\varepsilon)}{d\log(\rho_j)} \right)^{1/d} \\ &\le C d |\Lambda_\varepsilon|^{1/d}. \end{aligned} \tag{3.6}$$

We are now in position to prove the following theorem, variations of which can be considered as classical.

Theorem 3.5. *Let $d \in \mathbb{N}$ and $\boldsymbol{\rho} = (\rho_j)_{j=1}^d \in (1,\infty)^d$. Let $u : \mathcal{E}_{\boldsymbol{\rho}} \to \mathbb{C}$ be holomorphic. Then, for all $k \in \mathbb{N}_0$ and for any $\beta > 0$ such that*

$$\beta < \left(d! \prod_{j=1}^d \log(\rho_j) \right)^{1/d} \tag{3.7}$$

there exists $C > 0$ (depending on d, $\boldsymbol{\rho}$, k, β and u) such that with

$$l_{\boldsymbol{\nu}} := \int_{[-1,1]^d} u(\boldsymbol{y}) L_{\boldsymbol{\nu}}(\boldsymbol{y}) \mathrm{d}\mu_d(\boldsymbol{y}), \qquad \boldsymbol{\nu} \in \mathbb{N}_0^d \tag{3.8}$$

and Λ_ε in (3.3) it holds for all $\varepsilon \in (0,1)$

$$\left\| u - \sum_{\boldsymbol{\nu} \in \Lambda_\varepsilon} l_{\boldsymbol{\nu}} L_{\boldsymbol{\nu}} \right\|_{W^{k,\infty}([-1,1]^d)} \le C\, \mathrm{e}^{-\beta |\Lambda_\varepsilon|^{1/d}}.$$

Proof. Due to the holomorphy of u on $\mathcal{E}_{\boldsymbol{\rho}}$, for a constant $C > 0$ depending on d and $\boldsymbol{\rho}$, $l_{\boldsymbol{\nu}} \in \mathbb{R}$ satisfies the bound

$$|l_{\boldsymbol{\nu}}| \le C \|u\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}})} \boldsymbol{\rho}^{-\boldsymbol{\nu}} \prod_{j=1}^d (1+2\nu_j)^{1/2}, \qquad \boldsymbol{\nu} \in \mathbb{N}_0^d. \tag{3.9}$$

For $d = 1$ a proof can be found in [8, Chapter 12]. For general $d \in \mathbb{N}$ the bound follows by application of the one dimensional result in each variable. For more details we refer for instance to [5, Equations (2.14) and (2.16)] or [39, Corollary B.2.7].

Since $(L_{\boldsymbol{\nu}})_{\boldsymbol{\nu}\in\mathbb{N}_0^d}$ forms an orthonormal basis of (the Hilbert space) $L^2([-1,1]^d, \mu_d)$ we have

$$u(\boldsymbol{y}) = \sum_{\boldsymbol{\nu}\in\mathbb{N}_0^d} l_{\boldsymbol{\nu}} L_{\boldsymbol{\nu}} \tag{3.10}$$

converging in $L^2([-1,1]^d, \mu_d)$. Furthermore, with (3.9) and (2.23), for $k \in \mathbb{N}_0$ and every $\boldsymbol{\nu} \in \mathbb{N}_0^d$

$$|l_{\boldsymbol{\nu}}| \|L_{\boldsymbol{\nu}}\|_{W^{k,\infty}([-1,1]^d)} \le C\|u\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}})} \boldsymbol{\rho}^{-\boldsymbol{\nu}} \prod_{j=1}^d (1+2\nu_j)^{1+2k}. \tag{3.11}$$

Using [42, Lemma 3.13] (which is a variation of [6, Lemma 7.11]) $\sum_{\boldsymbol{\nu}\in\mathbb{N}_0^d} |l_{\boldsymbol{\nu}}| \|L_{\boldsymbol{\nu}}\|_{W^{k,\infty}([-1,1]^d)} < \infty$, and thus (3.10) also converges in $W^{k,\infty}([-1,1]^d)$.

Next, for $j \in \{1,\dots,d\}$ let $\mathbf{e}_j := (\delta_{ij})_{i=1}^d$ and introduce

$$A_\varepsilon := \{\boldsymbol{\nu} \in \mathbb{N}_0^d \,:\, \boldsymbol{\rho}^{-\boldsymbol{\nu}} < \varepsilon,\ \exists\, j \in \operatorname{supp}\boldsymbol{\nu} \text{ s.t. } \boldsymbol{\rho}^{-(\boldsymbol{\nu}-\mathbf{e}_j)} \ge \varepsilon\}.$$

Note that for $\varepsilon \in (0,1)$

$$\{\boldsymbol{\nu} \in \mathbb{N}_0^d \,:\, \boldsymbol{\rho}^{-\boldsymbol{\nu}} < \varepsilon\} = \{\boldsymbol{\mu}+\boldsymbol{\eta} \,:\, \boldsymbol{\mu} \in A_\varepsilon,\ \boldsymbol{\eta} \in \mathbb{N}_0^d\}. \tag{3.12}$$

Furthermore, since for every $\boldsymbol{\nu} \in A_\varepsilon$ there exists $j \in \operatorname{supp}\boldsymbol{\nu} \subseteq \{1,\dots,d\}$ such that $\boldsymbol{\rho}^{-(\boldsymbol{\nu}-\mathbf{e}_j)} \ge \varepsilon$ and therefore $\boldsymbol{\nu}-\mathbf{e}_j \in \Lambda_\varepsilon$, we find with (3.4) that there exists a constant C depending on d and $\boldsymbol{\rho}$ but independent of $\varepsilon \in (0,1)$ such that for all $\varepsilon \in (0,1)$

$$|A_\varepsilon| \le d|\Lambda_\varepsilon| \le C(1+\log(1/\varepsilon))^d. \tag{3.13}$$

Furthermore, for such $\boldsymbol{\nu} \in A_\varepsilon$ and $j \in \operatorname{supp}\boldsymbol{\nu} \subseteq \{1,\dots,d\}$ with $\rho_{\min} := \min_{i\in\{1,\dots,d\}} \rho_i$ we get

$$\rho_{\min}^{-|\boldsymbol{\nu}|_1+1} = \rho_{\min}^{-|\boldsymbol{\nu}-\mathbf{e}_j|_1} \ge \boldsymbol{\rho}^{-(\boldsymbol{\nu}-\mathbf{e}_j)} \ge \varepsilon$$

and therefore

$$|\boldsymbol{\nu}|_1 - 1 \le \frac{\log(1/\varepsilon)}{\log(\rho_{\min})}. \tag{3.14}$$

Using (3.12), there is $C > 0$ depending on d, $\boldsymbol{\rho}$, k but independent of $\varepsilon \in (0,1)$, with

$$\begin{aligned}
\left\| u - \sum_{\boldsymbol{\nu}\in\Lambda_\varepsilon} l_{\boldsymbol{\nu}} L_{\boldsymbol{\nu}} \right\|_{W^{k,\infty}([-1,1]^d)} &\le \sum_{\{\boldsymbol{\nu}\in\mathbb{N}_0^d \,:\, \boldsymbol{\rho}^{-\boldsymbol{\nu}}<\varepsilon\}} |l_{\boldsymbol{\nu}}| \|L_{\boldsymbol{\nu}}\|_{W^{k,\infty}([-1,1]^d)} \\
&\le \sum_{\{\boldsymbol{\nu},\boldsymbol{\mu} \,:\, \boldsymbol{\nu}\in A_\varepsilon,\ \boldsymbol{\mu}\in\mathbb{N}_0^d\}} C\|u\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}})} \boldsymbol{\rho}^{-(\boldsymbol{\nu}+\boldsymbol{\mu})} \prod_{j=1}^d (1+2(\nu_j+\mu_j))^{1+2k} \\
&\le C\|u\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}})} \sum_{\{\boldsymbol{\nu},\boldsymbol{\mu} \,:\, \boldsymbol{\nu}\in A_\varepsilon,\ \boldsymbol{\mu}\in\mathbb{N}_0^d\}} \boldsymbol{\rho}^{-\boldsymbol{\nu}} \boldsymbol{\rho}^{-\boldsymbol{\mu}} \prod_{j=1}^d \left((1+2\nu_j)(1+2\mu_j)\right)^{1+2k} \\
&\le C\|u\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}})} \varepsilon \left(\sum_{\boldsymbol{\nu}\in A_\varepsilon} \prod_{j=1}^d (1+2\nu_j)^{1+2k} \right) \left(\sum_{\boldsymbol{\mu}\in\mathbb{N}_0^d} \boldsymbol{\rho}^{-\boldsymbol{\mu}} \prod_{j=1}^d (1+2\mu_j)^{1+2k} \right).
\end{aligned}$$

The sum in the second brackets is finite independent of ε by [42, Lemma 3.13]. The sum in the first brackets can be bounded using (3.13) and (3.14) to obtain a constant $C > 0$ depending on u, d, $\boldsymbol{\rho}$ and k such that for all $\varepsilon \in (0,1)$

$$\left\| u - \sum_{\boldsymbol{\nu} \in \Lambda_\varepsilon} l_{\boldsymbol{\nu}} L_{\boldsymbol{\nu}} \right\|_{W^{k,\infty}([-1,1]^d)} \leq C\varepsilon |A_\varepsilon| \max_{\boldsymbol{\nu} \in A_\varepsilon} \prod_{j=1}^{d} (1+2\nu_j)^{1+2k} \leq C\varepsilon (1+\log(1/\varepsilon))^{2d+2dk}.$$

To finish the proof, note that our above calculation shows that for any $\tau \in (0,1)$ there exists $C_\tau > 0$ depending on u, d, $\boldsymbol{\rho}$ and k such that $\left\| u - \sum_{\boldsymbol{\nu} \in \Lambda_\varepsilon} l_{\boldsymbol{\nu}} L_{\boldsymbol{\nu}} \right\|_{W^{k,\infty}([-1,1]^d)} \leq C_\tau \varepsilon^\tau$ for all $\varepsilon \in (0,1)$. Moreover, (3.4) implies

$$\sum_{j=1}^{d} \log(\rho_j) - \left(|\Lambda_\varepsilon| d! \prod_{j=1}^{d} \log(\rho_j) \right)^{1/d} \geq \log(\varepsilon). \tag{3.15}$$

Hence for all $\varepsilon \in (0,1)$

$$\begin{aligned} \left\| u - \sum_{\boldsymbol{\nu} \in \Lambda_\varepsilon} l_{\boldsymbol{\nu}} L_{\boldsymbol{\nu}} \right\|_{W^{k,\infty}([-1,1]^d)} &\leq C_\tau \varepsilon^\tau \leq C_\tau \exp\left(\tau \left(\sum_{j=1}^{d} \log(\rho_j) - \left(|\Lambda_\varepsilon| d! \prod_{j=1}^{d} \log(\rho_j) \right)^{1/d} \right) \right) \\ &= C \exp\left(-\beta |\Lambda_\varepsilon|^{1/d} \right) \end{aligned}$$

where $C := C_\tau \exp(\tau \sum_{j=1}^{d} \log(\rho_j))$, $\beta := \tau (d! \prod_{j=1}^{d} \log(\rho_j))^{1/d}$ and where $\tau \in (0,1)$ can be arbitrarily close to 1. □

For later use, we note that the right-hand side of (3.7) can be estimated by Stirling's inequality, with $\rho_{\min} = \min_{j=1}^{d} \rho_j$ and $\rho_{\max} = \max_{j=1}^{d} \rho_j$:

$$(d/e) \log(\rho_{\min}) \leq \left(d! \prod_{j=1}^{d} \log(\rho_j) \right)^{1/d} \leq (d/e)(e^2 d)^{1/(2d)} \log(\rho_{\max}). \tag{3.16}$$

3.2 ReLU DNN approximation

We now come to the main result, concerning the approximation of holomorphic functions on bounded intervals by ReLU networks.

Theorem 3.6. *Fix $d \in \mathbb{N}$ and let $\boldsymbol{\rho} = (\rho_j)_{j=1}^{d} \in (1,\infty)^d$. Assume that $u : [-1,1]^d \to \mathbb{R}$ admits a holomorphic extension to $\mathcal{E}_{\boldsymbol{\rho}}$.*

Then, there exist constants $\beta' = \beta'(\boldsymbol{\rho}, d) > 0$ and $C = C(u, \boldsymbol{\rho}, d) > 0$, and for every $\mathcal{N} \in \mathbb{N}$ there exists a σ_1-NN $\tilde{u}_\mathcal{N} : [-1,1]^d \to \mathbb{R}$ satisfying

$$\operatorname{size}(\tilde{u}_\mathcal{N}) \leq \mathcal{N}, \qquad \operatorname{depth}(\tilde{u}_\mathcal{N}) \leq C \mathcal{N}^{\frac{1}{d+1}} \log_2(\mathcal{N}) \tag{3.17}$$

and the error bound

$$\| u(\cdot) - \tilde{u}_\mathcal{N}(\cdot) \|_{W^{1,\infty}([-1,1]^d)} \leq C \exp\left(-\beta' \mathcal{N}^{\frac{1}{d+1}} \right). \tag{3.18}$$

Proof. Throughout this proof, let $\beta > 0$ be fixed such that (3.7) holds. We proceed in three steps: In Step 1, we introduce a NN approximation of u, whose error, network depth and size we estimate in Step 2. Based on these estimates, we show Equations (3.17) – (3.18) in Step 3.

Step 1. Let $d \in \mathbb{N}$. In this step, for any $\varepsilon \in (0,1)$ we introduce a network $\hat{u}_\varepsilon$ approximating u (with increasing accuracy as $\varepsilon \to 0$).

Fix $\varepsilon \in (0,1)$ arbitrary, let $\Lambda_\varepsilon \subseteq \mathbb{N}_0^d$ be as in (3.3) and set $u_\varepsilon := \sum_{\boldsymbol{\nu} \in \Lambda_\varepsilon} l_{\boldsymbol{\nu}} L_{\boldsymbol{\nu}}$ with the Legendre coefficients $l_{\boldsymbol{\nu}}$ of u as in (3.8).

Let Affine_u be a NN of depth 0, with input dimension $|\Lambda_\varepsilon|$, output dimension 1 and size at most $|\Lambda_\varepsilon|$ which implements the affine transformation $\mathbb{R}^{|\Lambda_\varepsilon|} \to \mathbb{R} : (z_{\boldsymbol{\nu}})_{\boldsymbol{\nu} \in \Lambda} \mapsto \sum_{\boldsymbol{\nu} \in \Lambda_\varepsilon} l_{\boldsymbol{\nu}} z_{\boldsymbol{\nu}}$. Furthermore, let $\boldsymbol{f}_{\Lambda_\varepsilon, \delta}$ be the network from Proposition 2.13, emulating approximations to all multivariate Legendre polynomials $(L_{\boldsymbol{\nu}})_{\boldsymbol{\nu} \in \Lambda_\varepsilon}$. We define a NN

$$\hat{u}_\varepsilon := \mathrm{Affine}_u \circ \boldsymbol{f}_{\Lambda_\varepsilon, \delta}.$$

Then

$$\hat{u}_\varepsilon(\boldsymbol{y}) = \sum_{\boldsymbol{\nu} \in \Lambda_\varepsilon} l_{\boldsymbol{\nu}} \tilde{L}_{\boldsymbol{\nu}, \delta}(\boldsymbol{y}), \qquad \boldsymbol{y} \in [-1,1]^d,$$

where (with $\beta > 0$ as in (3.7)) the accuracy $\delta > 0$ of the σ_1-NN approximations of the tensor product Legendre polynomials is chosen as

$$\delta := \exp\left(-\beta |\Lambda_\varepsilon|^{1/d}\right).$$

Step 2. For the NN $\hat{u}_\varepsilon$ we obtain the error estimate

$$\|u_\varepsilon - \hat{u}_\varepsilon\|_{W^{1,\infty}([-1,1]^d)} \leq \sum_{\boldsymbol{\nu} \in \Lambda_\varepsilon} |l_{\boldsymbol{\nu}}| \, \|L_{\boldsymbol{\nu}} - \tilde{L}_{\boldsymbol{\nu}, \delta}\|_{W^{1,\infty}([-1,1]^d)} \leq \sum_{\boldsymbol{\nu} \in \Lambda_\varepsilon} |l_{\boldsymbol{\nu}}| \, \delta = \sum_{\boldsymbol{\nu} \in \Lambda_\varepsilon} |l_{\boldsymbol{\nu}}| \exp\left(-\beta |\Lambda_\varepsilon|^{1/d}\right).$$

With Theorem 3.5 this yields the existence of a constant $C > 0$ (depending on d, $\boldsymbol{\rho}$, β and u) such that

$$\|u - \hat{u}_\varepsilon\|_{W^{1,\infty}([-1,1]^d)} \leq C \exp\left(-\beta |\Lambda_\varepsilon|^{1/d}\right). \tag{3.19}$$

We now bound the depth and the size of $\hat{u}_\varepsilon$. Using Proposition 2.13 and (3.6), we obtain

$$\begin{aligned}
\mathrm{depth}(\hat{u}_\varepsilon) &\leq \mathrm{depth}(\mathrm{Affine}_u) + 1 + \mathrm{depth}\left(\boldsymbol{f}_{\Lambda_\varepsilon, \delta}\right) \\
&\leq C(1 + d \log d)(1 + \log_2 m(\Lambda_\varepsilon))\big(m(\Lambda_\varepsilon) + \log_2(1/\delta)\big) \\
&\leq C(1 + d \log d)(1 + \log_2(d) + \log_2 |\Lambda_\varepsilon|)\big(Cd|\Lambda_\varepsilon|^{1/d} + \beta |\Lambda_\varepsilon|^{1/d}\big) \\
&\leq C(1+\beta)(1 + d^2 (\log d)^2)(1 + |\Lambda_\varepsilon|^{1/d} \log_2 |\Lambda_\varepsilon|)
\end{aligned} \tag{3.20}$$

for $C > 0$ depending on $\boldsymbol{\rho}$. To bound the NN size, Proposition 2.13 and (3.6) give

$$\begin{aligned}
\mathrm{size}(\hat{u}_\varepsilon) &\leq 2\,\mathrm{size}(\mathrm{Affine}_u) + 2\,\mathrm{size}(\boldsymbol{f}_{\Lambda_\varepsilon, \delta}) \\
&\leq 2|\Lambda_\varepsilon| + 2Cd^2 m(\Lambda)^2 + 2Cdm(\Lambda)\log_2(1/\delta) + 2Cd^2|\Lambda_\varepsilon|\big(1 + \log_2 m(\Lambda_\varepsilon) + \log_2(1/\delta)\big) \\
&\leq 2|\Lambda_\varepsilon| + Cd^4(|\Lambda_\varepsilon|^{1/d})^2 + Cd^2|\Lambda_\varepsilon|^{1/d}\beta|\Lambda_\varepsilon|^{1/d} \\
&\quad + Cd^2|\Lambda_\varepsilon|\big(1 + \log(d) + \log_2|\Lambda_\varepsilon| + \beta|\Lambda_\varepsilon|^{1/d}\big) \\
&\leq C(1+\beta)d^4|\Lambda_\varepsilon|^{2/d} + C(1+\beta)(1 + d^2 \log d)|\Lambda_\varepsilon|^{1+1/d} \leq C_2(1+\beta)d^4|\Lambda_\varepsilon|^{1+1/d}
\end{aligned} \tag{3.21}$$

for a constant $C_2 > 0$ which depends on $\boldsymbol{\rho}$, but is independent of d, β, u and of $\varepsilon \in (0,1)$.

Step 3. Finally, we define $\tilde{u}_{\mathcal{N}}$. Fix $\beta > 0$ satisfying (3.7) and $\mathcal{N} \in \mathbb{N}$ such that $\mathcal{N} > \mathcal{N}_0 := C_2(1+\beta)d^4$, with the constant C_2 as in (3.21). Set

$$\widehat{\mathcal{N}} := \left(\frac{\mathcal{N}}{\mathcal{N}_0}\right)^{d/(d+1)} \in \mathbb{R}. \tag{3.22}$$

Next, let $\varepsilon \in (0,1)$ be such that

$$\widehat{\mathcal{N}} = \prod_{j=1}^{d} \left(\frac{\log(1/\varepsilon)}{\log(\rho_j)} + 1 \right), \tag{3.23}$$

which is possible since $\widehat{\mathcal{N}} > 1$ due to the assumption $\mathcal{N} > \mathcal{N}_0 = C_2(1+\beta)d^4$. Define $\tilde{u}_{\mathcal{N}} := \hat{u}_\varepsilon$.

First let us estimate the size of $\tilde{u}_{\mathcal{N}}$. By (3.5)

$$\widehat{\mathcal{N}} \geq \prod_{j=1}^{d} \left(\left\lfloor \frac{\log(1/\varepsilon)}{\log(\rho_j)} \right\rfloor + 1 \right) = \left| \left\{ \boldsymbol{\nu} \in \mathbb{N}_0^d \,:\, 0 \leq \nu_j \leq \frac{\log(1/\varepsilon)}{\log(\rho_j)} \ \forall j \right\} \right| \geq |\Lambda_\varepsilon|.$$

Hence (3.21) and the definition of $\widehat{\mathcal{N}}$ imply

$$\operatorname{size}(\tilde{u}_{\mathcal{N}}) = \operatorname{size}(\hat{u}_\varepsilon) \leq C_2(1+\beta)d^4|\Lambda_\varepsilon|^{1+1/d} \leq C_2(1+\beta)d^4\widehat{\mathcal{N}}^{1+1/d} \leq \mathcal{N}.$$

Similarly one obtains the bound on the depth of $\tilde{u}_{\mathcal{N}}$ by using (3.20). This shows (3.17).

Next we estimate the error $\|u - \tilde{u}_{\mathcal{N}}\|_{W^{1,\infty}([-1,1]^d)}$. By (3.5)

$$\begin{aligned}\widehat{\mathcal{N}} &\leq \prod_{j=1}^{d} \left(d \left\lfloor \frac{\log(1/\varepsilon)}{d\log(\rho_j)} \right\rfloor + d + 1 \right) = \prod_{j=1}^{d} \left(\left\lfloor \frac{\log(1/\varepsilon)}{d\log(\rho_j)} \right\rfloor + 1 \right) \prod_{j=1}^{d} \left(d + \frac{1}{\left\lfloor \frac{\log(1/\varepsilon)}{d\log(\rho_j)} \right\rfloor + 1} \right) \\ &\leq |\Lambda_\varepsilon|(d+1)^d.\end{aligned}$$

Thus (3.19) gives

$$\|u - \tilde{u}_{\mathcal{N}}\|_{W^{1,\infty}([-1,1]^d)} \leq C \exp\left(-\beta|\Lambda_\varepsilon|^{1/d}\right) \leq C \exp\left(-\beta(d+1)^{-1}\widehat{\mathcal{N}}^{1/d}\right).$$

By (3.22) this is (3.18) for any $\mathcal{N} > \mathcal{N}_0$ and with

$$\beta' = \beta(d+1)^{-1}(C_2(1+\beta)d^4)^{-1/(d+1)} \tag{3.24}$$

for C_2 as in (3.21) (independent of d, β and u). With $\tilde{u}_{\mathcal{N}} := 0$ (i.e. a trivial NN giving the constant value 0) for all (finitely many) $\mathcal{N} \leq \mathcal{N}_0$, we conclude that (3.18) holds for all $\mathcal{N} \in \mathbb{N}$ (by increasing $C > 0$ in (3.18) if necessary). □

Remark 3.7 (Fully connected networks)**.** *In the proof of Thm. 3.6 we explicitly constructed a sparsely connected DNN to approximate u. In practice, it might be tedious to implement this type of architecture. Instead one can set up a fully connected network, containing our sparse architecture. We shortly discuss the implications of Thm. 3.6 in this case.*

The width $w \in \mathbb{N}$ of a neural network ϕ (i.e. the maximum number of nodes in one of its layer) is trivially bounded by $\operatorname{size}(\phi)$*. For a fully connected network of width w, the weight matrix connecting two layers may have w^2 nonzero weights. Denote now by $\hat{u}_{\mathcal{N}}$ a fully connected σ_1-NN of width $w = \mathcal{N}$ and depth* $\operatorname{depth}(\hat{u}_{\mathcal{N}}) \leq C\mathcal{N}^{1/(d+1)}\log_2(\mathcal{N})$ *(with C as in* (3.17)*) realizing the function $\tilde{u}_{\mathcal{N}}$ from Thm. 3.6. The existence of $\hat{u}_{\mathcal{N}}$ is an immediate consequence of the depth and size bounds given in Thm. 3.6. Then by* (3.17)*, denoting its total number of weights, also counting vanishing weights, by* $\#\text{weights}(\hat{u}_{\mathcal{N}})$*,*

$$\#\text{weights}(\hat{u}_{\mathcal{N}}) \leq C\mathcal{N}^{2+\frac{1}{d+1}}\log_2(\mathcal{N}) = C\mathcal{N}^{\frac{2d+3}{d+1}}\log_2(\mathcal{N}), \qquad \operatorname{depth}(\hat{u}_{\mathcal{N}}) \leq C\mathcal{N}^{\frac{1}{d+1}}\log_2(\mathcal{N})$$

and by (3.18)

$$\|u - \hat{u}_{\mathcal{N}}\|_{W^{1,\infty}([-1,1]^d)} \leq C \exp\left(-\beta'\mathcal{N}^{\frac{1}{d+1}}\right).$$

This yields the error bound

$$\|u - \hat{u}_{\mathcal{N}}\|_{W^{1,\infty}([-1,1]^d)} \leq C \exp\left(-\beta' \mathcal{N}^{\frac{1}{d+1}}\right) \leq \exp\left(-\hat{\beta} \frac{(\#\text{weights}(\hat{u}_{\mathcal{N}}))^{\frac{1}{2d+3}}}{\log(\#\text{weights}(\hat{u}_{\mathcal{N}}))}\right),$$

for fully connected networks, and where $\hat{\beta} > 0$ is some constant independent of $\mathcal{N}$. Hence, the exponent in the error estimate has (up to logarithmic terms) decreased from $\frac{1}{d+1}$ for the sparsely connected network in Thm. 3.6 to $\frac{1}{2d+3}$ for the fully connected network.

Remark 3.8. *Note that in Step 2 of the proof, the network $\hat{u}_\varepsilon$ depends on u only via the Legendre coefficients $\{l_{\boldsymbol{\nu}}\}_{\boldsymbol{\nu}\in\Lambda_\varepsilon}$, appearing only as weights in the output layer. In particular, the weights and biases of $\hat{u}_\varepsilon$ continuously depend on u with respect to the $L^2([-1,1]^d, \mu_d)$-norm, because the Legendre coefficients do so. Finally, the $L^2([-1,1]^d, \mu_d)$-norm is bounded by the $L^\infty([-1,1]^d)$-norm.*

Remark 3.9. *The same result does not follow if we approximate the basis of multivariate polynomials by applying Proposition 2.6 to approximate the product of $m(\Lambda_\varepsilon)$ linear factors. With $\delta := \exp(-\beta|\Lambda_\varepsilon|^{1/d})$, each basis polynomial would have a network size of the order $O(m(\Lambda_\varepsilon)\log(1/\delta)) = O(m(\Lambda_\varepsilon)^2) = O(|\Lambda_\varepsilon|^{2/d})$, hence the total network size would be of the order $O(|\Lambda_\varepsilon|^{1+2/d})$, corresponding to $C\exp(-\beta'\mathcal{N}^{1/(d+2)})$ in the right-hand side of* (3.18).

3.3 RePU DNN approximation

For RePU approximations, with activation $\sigma_r(x)$ for integer $r \geq 2$, we may combine Proposition 2.14 (which is almost identical to [17, Theorem 4.1]) and Theorem 3.5 to infer the following result. Note that the decay of the provided upper bound of the error in (3.25) in terms of the network size $\mathcal{N}$ is slightly faster than the one we obtained for ReLU approximations in (3.18).

Theorem 3.10. *Fix $d \in \mathbb{N}$, $k \in \mathbb{N}_0$ and $r \in \mathbb{N}$, $r \geq 2$. Let $\boldsymbol{\rho} = (\rho_j)_{j=1}^d \in (1,\infty)^d$. Assume that $u : [-1,1]^d \to \mathbb{R}$ admits a holomorphic extension to $\mathcal{E}_{\boldsymbol{\rho}}$.*

Then, there exists $C > 0$ and a constant $C_1 > 0$ which only depends on r such that with β as in (3.7)*, for every $\mathcal{N} \in \mathbb{N}$, there exists a σ_r-NN $\tilde{u}_{\mathcal{N}} : [-1,1]^d \to \mathbb{R}$ satisfying*

$$\text{size}(\tilde{u}_{\mathcal{N}}) \leq C_1 \mathcal{N}, \qquad \text{depth}(\tilde{u}_{\mathcal{N}}) \leq C_1 \log_2(\mathcal{N}) \tag{3.25}$$

and, with $\beta' := \beta/(d+1)$,

$$\|u(\boldsymbol{y}) - \tilde{u}_{\mathcal{N}}(\boldsymbol{y})\|_{W^{k,\infty}([-1,1]^d)} \leq C \exp\left(-\beta' \mathcal{N}^{\frac{1}{d}}\right). \tag{3.26}$$

Here, we can consider the $W^{k,\infty}([-1,1]^d)$-norm of $(u - \tilde{u}_{\mathcal{N}})$ for $k \in \mathbb{N}$ independent of r, because u is holomorphic on $[-1,1]^d$, and $\tilde{u}_{\mathcal{N}}$ is a polynomial by construction. Also, we note with (3.16) that $\beta' = \log(\rho_{\min})/(2e)$ is attainable for all $d \in \mathbb{N}$.

Proof. For $\varepsilon \in (0,1)$ let Λ_ε be as in (3.3). This set is finite and downward closed. Hence, by Proposition 2.14 there exists a σ_r-NN $\hat{u}_\varepsilon$ such that $\hat{u}_\varepsilon(\boldsymbol{y}) = \sum_{\boldsymbol{\nu}\in\Lambda_\varepsilon} l_{\boldsymbol{\nu}} L_{\boldsymbol{\nu}}(\boldsymbol{y})$ for all $\boldsymbol{y} \in [-1,1]^d$. According to this proposition, the NN $\hat{u}_\varepsilon$ satisfies $\text{size}(\hat{u}_\varepsilon) \leq C_1|\Lambda_\varepsilon|$ and $\text{depth}(\hat{u}_\varepsilon) \leq C_1 \log|\Lambda_\varepsilon|$. This is (3.25) for $\mathcal{N} := |\Lambda_\varepsilon|$. By Theorem 3.5, it holds (3.26) for such $\mathcal{N}$, with $\beta' = \beta$.

For general $\mathcal{N} > 1$, it follows as in Step 3 of the proof of Theorem 3.6 (with $\mathcal{N}$ taking the role of $\widehat{\mathcal{N}}$ in (3.23)) that there exists $\varepsilon \in (0,1)$ such that $(d+1)^{-d}\mathcal{N} \leq |\Lambda_\varepsilon| \leq \mathcal{N}$. This implies that (3.26) holds for any $\mathcal{N} \in \mathbb{N}$ with $\beta' := \beta/(d+1)$ and a constant C depending on d. □

Remark 3.11 (Fully connected networks)**.** *A similar statement as in Rmk. 3.7 also holds for σ_r-NNs with $r \geq 2$. By the same arguments, we obtain an error bound the type*

$$\|u(\boldsymbol{y}) - \hat{u}_{\mathcal{N}}(\boldsymbol{y})\|_{W^{k,\infty}([-1,1]^d)} \leq C \exp\left(-\hat{\beta} \frac{\#\text{weights}(\hat{u}_{\mathcal{N}})^{\frac{1}{2d}}}{\log(\#\text{weights}(\hat{u}_{\mathcal{N}}))}\right)$$

for a fully connected σ_r-NN $\hat{u}_{\mathcal{N}}$, whose total number of weights, also counting vanishing weights, we denote by $\#\text{weights}(\hat{u}_{\mathcal{N}})$. Here $k \in \mathbb{N}$ is arbitrary but fixed, and $\hat{\beta} > 0$ is a constant independent of $\mathcal{N}$.

Remark 3.12. *It follows from the proof of Proposition 2.14 that the weights of $\tilde{u}_{\mathcal{N}}$ depend continuously on the Legendre coefficients of u, which themselves depend continuously on u w.r.t. the $L^2([-1,1]^d, \mu_d)$-norm, which is bounded by the $L^\infty([-1,1]^d)$-norm.*

Remark 3.13. *A similar result as in Theorem 3.10 was obtained in [21, Theorem 3.3]. It assumed a different class of activation functions, termed "sigmoidal functions of order $k \geq 2$" (see Remark 2.1). The $L^\infty([-1,1]^d)$ error bound provided in [21, Theorem 3.3] is, in our notation, of the type $\exp(-b\mathcal{N}^{1/d})$ for a suitable constant $b > 0$ and a DNN of size $\mathcal{N}\log(\mathcal{N})$. This is slightly worse than Theorem 3.10. Also note that in [21] the number of neurons is used as measure for the NN size, which may be smaller but not larger than the number of nonzero weights if all neurons have at least one nonzero weight.*

4 Conclusion

We review in Section 4.1 the main results obtained in the previous sections. In Section 4.2, we relate these results to results which appeared in the literature. In Section 4.3, we discuss several novel implications of the main results, which could be of interest in various applications. We point out that although the present analysis is developed in detail for DNNs with ReLU activation, as explained in Remarks 2.1 and 2.15, all DNN expression error bounds proved up to this point, and also in the ensuing remarks remain valid (possibly even with slightly better estimates for the DNN sizes) for smoother activation functions, such as sigmoidal, tanh, or softmax activations.

4.1 Main Results

We have established for analytic maps $u : [-1,1]^d \to \mathbb{R}$ exponential expression rate bounds in $W^{k,\infty}([-1,1]^d)$ in terms of the DNN size for the ReLU activation (for $k = 0, 1$) and for the RePU activations σ_r, $r \geq 2$ (for $k \in \mathbb{N}_0$). The present analysis improves earlier results in that the NN sizes are slightly reduced and we obtain exponential convergence of ReLU and RePU DNNs for general d-variate analytic functions, without assuming the Taylor expansion of u around $0 \in \mathbb{R}^d$ to converge on $[-1,1]^d$. We also point out that by a simple scaling argument our main results in Theorem 3.6 and Theorem 3.10 imply corresponding expression rate results for analytic functions defined on an arbitrary cartesian product of finite intervals $\times_{j=1}^d [a_j, b_j]$, where $-\infty < a_j < b_j < \infty$ for all $j \in \{1, \ldots, d\}$.

4.2 Related Results

We already commented on [10] where ReLU NN expression rates for multivariate, holomorphic functions u were obtained. Assumptions in [10, Theorem 2.6] included absolute convergence of Taylor expansions of u about the origin with convergence radius sufficiently large to contain the unit cube $[-1,1]^d$, implying existence of a complex holomorphic extension to $(B_1^{\mathbb{C}})^d$. Under those assumptions $L^\infty([-(1-\delta),(1-\delta)]^d)$-error bounds were obtained for any $\delta \in (0,1)$. With

a linear coordinate transformation, error bounds on $[-1,1]^d$ follow under the assumption that the Taylor expansion converges absolutely on $[-(1-\delta)^{-1},(1-\delta)^{-1}]^d$. The presently proposed argument being based on (classical) Bernstein ellipses is admissible for functions u which are real analytic merely in $[-1,1]^d$ (cp. Remark 3.1). We also mention the recent work [9] which addresses similar questions as in the present paper; the results in that reference address, however, only L^∞ errors and obtain slightly larger NN sizes. Our proofs are constructive, with constructions being based on ReLU NN emulations of Legendre polynomials, drawing on [25]. In [34], alternative constructions of so-called RePU NNs are proposed which are based on NN emulation of univariate Chebyšev polynomials. It is argued in [34] (and verified in numerical experiments) that the numerical size of NN weights scales more favorably than the weights in the presently proposed emulations. "Chebyšev" versions of the present proofs could also be facilitated, resulting in the same scalings of NN sizes and depths, however, as are obtained here.

4.3 Applications and generalizations

4.3.1 Solution manifolds of PDEs

One possible application of our results concerns the approximation of (quantities of interest) of *solution manifolds of parametric PDEs* depending on a d-dimensional parameter $\boldsymbol{y} \in [-1,1]^d$. Such a situation arises in particular in Uncertainty Quantification (UQ). There, a mathematical model is described by a PDE depending on the parameters $\boldsymbol{y}$, which in turn can for instance determine boundary conditions, forcing terms or diffusion coefficients. It is known for a wide range of linear and nonlinear PDE models (see e.g. [5]), that parametric PDE solutions depend analytically on the parameters. In addition, for these models usually one has precise knowledge on the domain of holomorphic extension of the objective function u, i.e. knowledge of the constants $(\rho_j)_{j=1}^d$ in Thm. 3.5. These constants determine the sets of multiindices Λ_ε in (3.3). As our proofs are constructive and based on the sets Λ_ε, such information can be leveraged to a priori guide the identification of suitable network architectures.

4.3.2 ReLU DNN expression of Data-to-QoI maps for Bayesian PDE Inversion

The exponential σ_1-DNN expression rate bound, Theorem 3.6, implies exponential expressivity of data-to-quantity of interest maps in Bayesian PDE inversion, as is shown in [14]. Here, the assumption of *centered, additive Gaussian observation noise* in the data model underlying the Bayesian inverse theory implies holomorphy of the data to prediction map in the Bayesian theory as we show [14]. This, combined with the present results in Theorems 3.6 and 3.10 implies exponential expressivity of σ_r DNNs for this map, for all $r \geq 1$.

4.3.3 Infinite-dimensional ($d = \infty$) case

The expression rate analysis becomes more involved, if the objective function u depends on an infinite dimensional parameter (i.e., a parameter sequence) $\boldsymbol{y} \in [-1,1]^{\mathbb{N}}$. Such functions occur in UQ for instance if the uncertainty is described by a Karhunen-Loeve expansion. Under certain circumstances, u can be expressed by a so-called *generalized polynomial chaos (gpc) expansion*. Reapproximating truncated gpc expansions by NNs leads to expression rate results for the approximation of infinite dimensional functions, as we showed in [33]. One drawback of [33] is however, that the proofs crucially relied on the assumption that u is holomorphic on certain polydiscs containing $[-1,1]^{\mathbb{N}}$. This criterion is not always met in practice [5]. To overcome this restriction, we will generalize the expression rate results of [33] in the forthcoming paper [26],

by basing the analysis on the present results for the approximation of d-variate functions which are merely assumed to be analytic in some (possibly small) neighborhood of $[-1,1]^d$.

4.3.4 Gevrey functions

The use of DNN approximations of tensor product Legendre polynomials constructed in Section 2 can be used more generally than for the approximation of holomorphic functions by truncated Legendre expansions. We consider as an example, for $d \in \mathbb{N}$, the approximation of non-holomorphic, Gevrey regular functions (see, e.g., [29] and the references there for definitions and properties of such functions). Here, for some $\delta \geq 1$ we consider maps $u : [-1,1]^d \to \mathbb{R}$ that satisfy, for constants $C, A > 0$ depending on u, the bound

$$\forall \boldsymbol{\nu} \in \mathbb{N}_0^d : \qquad \left\| \frac{\partial^{|\boldsymbol{\nu}|_1} u}{\partial_{x_1}^{\nu_1} \cdots \partial_{x_d}^{\nu_d}} \right\|_{L^\infty([-1,1]^d)} \leq C A^{|\boldsymbol{\nu}|_1} (\boldsymbol{\nu}!)^\delta. \tag{4.1}$$

We write $u \in \mathcal{G}^\delta([-1,1]^d, C, A)$ for u satisfying (4.1). Evidently, $\mathcal{G}^\delta([-1,1]^d, C, A) \subset C^\infty([-1,1]^d)$. These maps are analytic when $\delta = 1$, but possibly non-analytic when $\delta > 1$.

Proposition 4.1. *For dimension $d \in \mathbb{N}$, and for constants $C, A > 0$, for $u \in \mathcal{G}^\delta([-1,1]^d, C, A)$ exist $C'(d,\delta,u) > 0$ and $\beta'(d,\delta,u) > 0$, and for every $\mathcal{N} \in \mathbb{N}$ there exists a ReLU DNN $\tilde{u}_\mathcal{N}$ such that*

$$\begin{aligned} &\mathrm{size}(\tilde{u}_\mathcal{N}) \leq \mathcal{N}, \qquad \mathrm{depth}(\tilde{u}_\mathcal{N}) \leq C' \mathcal{N}^{\min\left\{\frac{1}{2}, \frac{1}{d+1/\delta}\right\}} \log(\mathcal{N}), \\ &\|u - \tilde{u}_\mathcal{N}\|_{W^{1,\infty}([-1,1]^d)} \leq C' \exp\left(-\beta' \mathcal{N}^{\min\left\{\frac{1}{2\delta}, \frac{1}{d\delta+1}\right\}}\right). \end{aligned}$$

In the proof, which is provided in Appendix B, we furthermore show that there exist constants $C', \beta' > 0$ such that for every $p \in \mathbb{N}$ holds

$$\forall u \in \mathcal{G}^\delta([-1,1]^d, C, A) : \inf_{v_p \in \otimes_{j=1}^d \mathbb{P}_p([-1,1])} \|u - v_p\|_{W^{1,\infty}([-1,1]^d)} \leq C' \exp(-\beta' N^{1/(\delta d)}) \,. \tag{4.2}$$

Here, $N = \dim(\otimes_{j=1}^d \mathbb{P}_p([-1,1])) = (p+1)^d$ denotes the dimension of the space of all d-variate polynomials of degree at most p in each variable.

4.3.5 ReLU expression of non-smooth maps by composition

The results were based on the quantified holomorphy of the map $u : [-1,1]^d \to \mathbb{C}$. While this could be perceived as a strong requirement (and, consequently, limitation) of the present results, by composition the present deep ReLU NN emulation rate bounds cover considerably more general situations. The key observation is that deep ReLU NNs are closed under concatenation (or under composition of realizations) as we explained in Section 2.2.3.

Let us give a specific example from high-dimensional integration, where the task is to evaluate the integral

$$\int_{[-1,1]^d} u(\boldsymbol{y}) \pi(\boldsymbol{y}) \mathrm{d}\boldsymbol{y} \,. \tag{4.3}$$

Here, $u : [-1,1]^d \to \mathbb{R}$ is a function which is holomorphic in a polyellipse $\mathcal{E}_{\boldsymbol{\rho}}$ as in (3.1) and π denotes an a-priori given probability density on the coordinates $y_1, ..., y_d$ w.r.t. the measure μ_d (i.e. $\pi : [-1,1]^d \to [0,\infty)$ is measurable and satisfies $\int_{[-1,1]^d} \pi(\boldsymbol{x}) \mathrm{d}\mu_d(\boldsymbol{x}) = 1$). Assuming that the coordinates are independent, the density π factors, i.e. $\pi = \bigotimes_{j=1}^d \pi_j$ with certain marginal probability densities π_j which we assume to be absolutely continuous w.r. to the Lebesgue measure, i.e. $\int_{-1}^1 \pi_j(\xi) \mathrm{d}\xi = 2$. In the case that the *marginals $\pi_j > 0$ are simple functions* for

example on finite partitions $\mathcal{T}_j$ of $[-1,1]$ (as e.g. if π_j is a histogram for the law of y_j estimated from empirical data), the changes of coordinates in (4.3)

$$T_j(y_j) := -1 + \int_{-1}^{y_j} \pi_j(\xi_j)\mathrm{d}\xi_j : [-1,1] \to [-1,1]\,, \quad j = 1, ..., d \tag{4.4}$$

are bijective. Furthermore, in this case each component map $T_j : [-1,1] \to [-1,1]$ is bijective, continuous and piecewise affine, and can therefore *be exactly represented by a σ_1-NN of depth* 1 *and width proportional to $\#(\mathcal{T}_j)$*.

Denote by $T = (T_1, ..., T_d)^\top$ the d-variate diagonal transformation, and let $T^{-1} : [-1,1]^d \to [-1,1]^d$ denote its inverse (which is also continuous, piecewise linear). Denoting by $dT^{-1}(\boldsymbol{x})$ the Jacobian matrix of T^{-1} at $\boldsymbol{x} \in [-1,1]^d$ we may then rewrite (4.3) as

$$\int_{[-1,1]^d} u(\boldsymbol{y})\pi(\boldsymbol{y})\mathrm{d}\boldsymbol{y} = \int_{[-1,1]^d} u(T^{-1}(\boldsymbol{x}))\pi(T^{-1}(\boldsymbol{x}))\det dT^{-1}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \int_{[-1,1]^d} g(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \tag{4.5}$$

where $g = u \circ T^{-1}$ is not continuously differentiable. Here we have used that $dT^{-1}(T(\boldsymbol{y})) = (dT(\boldsymbol{y}))^{-1}$ and $\det(dT(\boldsymbol{y})) = \pi(\boldsymbol{y})$, i.e. $\det dT^{-1}(\boldsymbol{x}) = \pi(T^{-1}(\boldsymbol{x}))^{-1}$.

Now, the function $\tilde{g}_\mathcal{N} := \tilde{u}_\mathcal{N} \circ T^{-1}$ with the σ_1-NN $\tilde{u}_\mathcal{N}$ constructed in Theorem 3.6 is a σ_1-NN which still affords the error bound (3.18): Denote for $n \in \mathbb{N}$ and $f \in W^{1,\infty}([-1,1]^d, \mathbb{R}^n)$

$$|f|_{W^{1,\infty}([-1,1]^d,\mathbb{R}^n)} := \sup_{\boldsymbol{x}\neq\boldsymbol{y}\in[-1,1]^d} \frac{\|f(\boldsymbol{x}) - f(\boldsymbol{y})\|}{\|\boldsymbol{x} - \boldsymbol{y}\|},$$

where $\|\cdot\|$ is the Euclidean norm on $\mathbb{R}^n$ resp. on $\mathbb{R}^d$. As usual, for $n = 1$ we write $|f|_{W^{1,\infty}([-1,1]^d)} := |f|_{W^{1,\infty}([-1,1]^d,\mathbb{R})}$ instead. With these conventions, it holds

$$\begin{aligned}
\|g(\cdot) - \tilde{g}_\mathcal{N}(\cdot)\|_{W^{1,\infty}([-1,1]^d)} &= \|u \circ T^{-1}(\cdot) - \tilde{u}_\mathcal{N} \circ T^{-1}(\cdot)\|_{W^{1,\infty}([-1,1]^d)} \\
&= \|u \circ T^{-1}(\cdot) - \tilde{u}_\mathcal{N} \circ T^{-1}(\cdot)\|_{L^\infty([-1,1]^d)} + |u \circ T^{-1}(\cdot) - \tilde{u}_\mathcal{N} \circ T^{-1}(\cdot)|_{W^{1,\infty}([-1,1]^d)} \\
&\leq \|u(\cdot) - \tilde{u}_\mathcal{N}(\cdot)\|_{L^\infty([-1,1]^d)} + |u(\cdot) - \tilde{u}_\mathcal{N}(\cdot)|_{W^{1,\infty}([-1,1]^d)}|T^{-1}|_{W^{1,\infty}([-1,1]^d,\mathbb{R}^d)} \\
&\leq C \exp\left(-\beta' \mathcal{N}^{\frac{1}{d+1}}\right)
\end{aligned} \tag{4.6}$$

for a constant C which now additionally depends on $|T^{-1}(\cdot)|_{W^{1,\infty}([-1,1]^d,\mathbb{R}^d)}$. The approximation of the integral (4.3) can thus be reduced to the problem of approximating the integral of the surrogate $\tilde{g}_\mathcal{N}$, which can be efficiently represented by a σ_1-NN. In the case that u is merely assumed Gevrey regular as in Sec. 4.3.4, a similar calculation leads to a bound of the type (4.6), but with $\exp(-\beta'\mathcal{N}^{\frac{1}{d+1}})$ replaced by $\exp(-\beta'\mathcal{N}^{\min\{\frac{1}{2\delta}, \frac{1}{d\delta+1}\}})$.

More generally, if $\pi : [-1,1]^d \to (0,\infty)$ is for example a continuous density function (not necessarily a product of its marginals) there exists a bijective transport $T : [-1,1]^d \to [-1,1]^d$ such that analogous to (4.5) it holds $\int_{[-1,1]^d} u(\boldsymbol{y})\pi(\boldsymbol{y})\mathrm{d}\boldsymbol{y} = \int_{[-1,1]^d} u(T^{-1}(\boldsymbol{x}))\mathrm{d}\boldsymbol{x}$ (contrary to the situation above, this transformation T is not diagonal in general). One explicit representation of such a transport is provided by the Knothe-Rosenblatt transport, see, e.g. [31, Section 2.3]. It has the property that T inherits the smoothness of π, cp. [31, Remark 2.19]. In case T^{-1} can be realized without error by a σ_1 (or σ_r) network, we find again an estimate of the type (3.18). If T^{-1} does not allow an explicit representation by a NN however, we may still approximate T^{-1} by a NN $\tilde{S}_\mathcal{N}$ to obtain a NN $\tilde{g}_\mathcal{N} := \tilde{u}_\mathcal{N} \circ \tilde{S}_\mathcal{N}$ approximating $g = u \circ T^{-1}$. This will introduce an additional error in (4.6) due to the approximation of T^{-1}. We refer to [41, Section 4.3] and the references there.

A Proof of Proposition 2.14

Proof. The proof consists of 2 steps. In Step 1, we define subnetworks, similar to those in [25, Lemma 4.5], to emulate all monomials $\boldsymbol{x}^{\boldsymbol{\nu}}$ for $\boldsymbol{\nu} \in \Lambda$ of order $2^{k-1} \leq |\boldsymbol{\nu}|_1 \leq 2^k$. In Step 2, we use them to construct $\tilde{p}$.

Step 1. Throughout this proof, we denote the NN input by $\boldsymbol{x} \in \mathbb{R}^d$. For $k \in \mathbb{N}_0$ we define the index sets $\Lambda_k := \{\boldsymbol{\nu} \in \Lambda : |\boldsymbol{\nu}|_1 = k\}$ and $\Delta_k := \{\boldsymbol{\nu} \in \Lambda : 2^{k-1} < |\boldsymbol{\nu}|_1 \leq 2^k\}$. In this first step of the proof, we define subnetworks to emulate $\boldsymbol{x}^{\boldsymbol{\nu}}$ for $\boldsymbol{\nu} \in \Lambda_{2^{k-1}} \cup \Delta_k$.

We will use that there exists a σ_r-NN $\tilde{\times}_r$ of depth 1, with input dimension 2 and output dimension 1, which exactly emulates the product operator $\mathbb{R}^2 \to \mathbb{R} : (x, y) \mapsto xy$. For $r = 2$ this was shown in [17, Lemma 2.1], for $r > 2$ it follows from [17, Theorem 2.5] and the polarization identity $xy = \frac{1}{4}(x+y)^2 - \frac{1}{4}(x-y)^2$, which was used in the proof of [17, Lemma 2.1]. We note that the size of $\tilde{\times}_r$ depends on r.

Next, for all $k \in \mathbb{N}$ such that $\Delta_k \neq \emptyset$ we define the σ_r-NN Ψ_k as

$$\Psi_k := \left(\{\mathrm{Id}_{\mathbb{R}}\}_{j=1}^{|\Lambda_{2^{k-1}}|}, \{\tilde{\times}_r\}_{j=1}^{|\Delta_k|} \right),$$

where the identity networks have depth 1. With the convention that $\Lambda_{1/2} := \emptyset$, we define Ψ_k such that applied to the inputs $\{\boldsymbol{x}^{\boldsymbol{\nu}} : \boldsymbol{\nu} \in \Lambda_{2^{k-2}} \cup \Delta_{k-1}\}$ the identity networks compute the input values $\boldsymbol{x}^{\boldsymbol{\nu}} : \boldsymbol{\nu} \in \Lambda_{2^{k-1}} \subset \Delta_{k-1}$ and the product networks compute $\boldsymbol{x}^{\boldsymbol{\nu}} : \boldsymbol{\nu} \in \Delta_k$. This is possible, because Λ is downward closed: for all $\boldsymbol{\nu} \in \Delta_k$ and all $\boldsymbol{\mu} \leq \boldsymbol{\nu}$ such that $2^{k-2} \leq |\boldsymbol{\mu}|_1 \leq 2^{k-1}$, we assumed that $\boldsymbol{x}^{\boldsymbol{\mu}}$ is part of the input of Ψ_k ($\boldsymbol{\nu} \in \Lambda$ implies $\boldsymbol{\mu} \in \Lambda$, hence $\boldsymbol{\mu} \in \Delta_{k-1}$). In particular, there exists $\boldsymbol{\mu} \in \Delta_{k-1}$ such that $|\boldsymbol{\mu}|_1 = \lceil |\boldsymbol{\nu}|_1/2 \rceil$. This implies that $|\boldsymbol{\nu} - \boldsymbol{\mu}|_1 = \lfloor |\boldsymbol{\nu}|_1/2 \rfloor$ and thus $\boldsymbol{\nu} - \boldsymbol{\mu} \in \Lambda_{2^{k-2}} \cup \Delta_{k-1}$. As a result, $\boldsymbol{x}^{\boldsymbol{\nu}}$ can be computed as $\boldsymbol{x}^{\boldsymbol{\nu}} = \tilde{\times}_r(\boldsymbol{x}^{\boldsymbol{\mu}}, \boldsymbol{x}^{\boldsymbol{\nu}-\boldsymbol{\mu}})$.

Next, we estimate the NN depth and size of Ψ_k. It holds that $\mathrm{depth}(\Psi_k) = 1$,

$$\begin{aligned}
\mathrm{size}(\Psi_k) &\leq |\Lambda_{2^{k-1}}| \,\mathrm{size}(\mathrm{Id}_{\mathbb{R}}) + |\Delta_k| \,\mathrm{size}(\tilde{\times}_r) \leq C(r)(|\Lambda_{2^{k-1}}| + |\Delta_k|) \\
&\leq C(r)(|\Delta_{k-1}| + |\Delta_k|), \\
\mathrm{size}_{\mathrm{in}}(\Psi_k) &\leq C(r)(|\Delta_{k-1}| + |\Delta_k|), \\
\mathrm{size}_{\mathrm{out}}(\Psi_k) &\leq C(r)(|\Delta_{k-1}| + |\Delta_k|).
\end{aligned}$$

Step 2. In this step we construct $\tilde{p}$. Let $m := m(\Lambda)$ as defined in Equation (2.24) and $k := \min\{k \in \mathbb{N} : 2^k \geq m\}$. In addition, we will write $p(\boldsymbol{x}) =: \sum_{\boldsymbol{\nu} \in \Lambda} t_{\boldsymbol{\nu}} \boldsymbol{x}^{\boldsymbol{\nu}}$.

We define $\tilde{p}$ as

$$\tilde{p} := \mathrm{Affine} \circ (\Psi_k, \mathrm{psum}_k) \circ \left(\Psi_{k-1}, \mathrm{psum}_{k-1}\right) \circ \cdots \circ (\Psi_1, \mathrm{psum}_1),$$

where for $j = 1, \ldots, k$

$$\mathrm{psum}_j \left(\{\boldsymbol{x}^{\boldsymbol{\nu}}\}_{\boldsymbol{\nu} \in \Lambda_{2^{j-2}}}, \{\boldsymbol{x}^{\boldsymbol{\nu}}\}_{\boldsymbol{\nu} \in \Delta_{j-1}}, \mathrm{psum}_{j-1} \right) := \mathrm{Id}_{\mathbb{R}} \left(\mathrm{psum}_{j-1} + \sum_{\boldsymbol{\nu} \in \Delta_{j-1}} t_{\boldsymbol{\nu}} \boldsymbol{x}^{\boldsymbol{\nu}} \right),$$

where the σ_r-identity network has depth 1. In addition, denote by $\boldsymbol{\nu}^{(i)}$, $i = 1, \ldots, |\Delta_k|$ any enumeration of Δ_k. Then, Affine is a NN of depth 0, input dimension $|\Lambda_{2^{k-1}}| + |\Delta_k| + 1$, output dimension 1, computing the affine transformation

$$\begin{aligned}
&\mathrm{Affine}(w_1, \ldots, w_{|\Lambda_{2^{k-1}}|}, \boldsymbol{x}^{\boldsymbol{\nu}^{(1)}}, \ldots, \boldsymbol{x}^{\boldsymbol{\nu}^{(|\Delta_k|)}}, w_{|\Lambda_{2^{k-1}}|+|\Delta_k|+1}) \\
&\qquad := t_{\mathbf{0}} + w_{|\Lambda_{2^{k-1}}|+|\Delta_k|+1} + \sum_{j=1}^{|\Delta_k|} \boldsymbol{x}^{\boldsymbol{\nu}^{(j)}} t_{\boldsymbol{\nu}^{(j)}},
\end{aligned}$$

where the constant $t_{\mathbf{0}}$ is a NN bias. Thus, Affine ignores the first $|\Lambda_{2^{k-1}}|$ inputs, takes an affine combination of the then following $|\Delta_k|$ inputs, and adds the last input. As a result, $\tilde{p}(\boldsymbol{x}) = p(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{R}^d$.

To bound the network depth and size, we note that for $j = 1, \dots, k$

$$\begin{aligned}
\operatorname{size}_{\text{in}}(\text{psum}_j) &\leq C(r)(1 + |\Delta_{j-1}|),\\
\operatorname{size}_{\text{out}}(\text{psum}_j) &\leq C(r),\\
\operatorname{size}(\text{psum}_j) &\leq C(r)(1 + |\Delta_{j-1}|),\\
\operatorname{size}(\text{Affine}) = \operatorname{size}_{\text{in}}(\text{Affine}) &= \operatorname{size}_{\text{out}}(\text{Affine}) \leq 2 + |\Delta_k|.
\end{aligned}$$

We obtain the following bounds on the depth and size of $\tilde{p}$: In case $|\Lambda| = 1$, the constant polynomial p can be emulated exactly by a σ_r-NN $\tilde{p}$ of depth 0 and size 1. In case $|\Lambda| \geq 2$, it holds:

$$\begin{aligned}
\operatorname{depth}(\tilde{p}) &\leq \operatorname{depth}(\text{Affine}) + \sum_{j=1}^{k}(1 + \operatorname{depth}(\Psi_j)) = 2k \leq 2 + 2\log_2(m) \leq C\log_2(|\Lambda|),\\
\operatorname{size}(\tilde{p}) &\leq \operatorname{size}(\text{Affine}) + \operatorname{size}_{\text{in}}(\text{Affine}) + \sum_{j=1}^{k}\Big(\operatorname{size}_{\text{out}}(\Psi_j) + \operatorname{size}_{\text{out}}(\text{psum}_j) + \operatorname{size}(\Psi_j)\\
&\quad + \operatorname{size}(\text{psum}_j) + \operatorname{size}_{\text{in}}(\Psi_j) + \operatorname{size}_{\text{in}}(\text{psum}_j)\Big)\\
&\leq (2 + |\Delta_k|) + (2 + |\Delta_k|) + \sum_{j=1}^{k}\Big(C(r)(|\Delta_{j-1}| + |\Delta_j|) + C(r) + C(r)(|\Delta_{j-1}| + |\Delta_j|)\\
&\quad + C(r)(1 + |\Delta_{j-1}|) + C(r)(|\Delta_{j-1}| + |\Delta_j|) + C(r)(1 + |\Delta_{j-1}|)\Big)\\
&\leq C(r)\left(1 + \sum_{j=0}^{k}|\Delta_j|\right) \leq C(r)|\Lambda|,\\
\operatorname{size}_{\text{in}}(\tilde{p}) &\leq \operatorname{size}_{\text{in}}(\Psi_1) + \operatorname{size}_{\text{in}}(\text{psum}_1) \leq C(r)(|\Delta_0| + |\Delta_1|) \leq C(r)|\Lambda|,\\
\operatorname{size}_{\text{out}}(\tilde{p}) &\leq 2\operatorname{size}_{\text{out}}(\text{Affine}) \leq C|\Delta_k| \leq C|\Lambda|,
\end{aligned}$$

where $C, C(r)$ are independent of d.

□

B Proof of Proposition 4.1

Proof. As in the holomorphic case, to approximate functions $u \in \mathcal{G}^{\delta}([-1,1]^d, C, A)$, we first build a tensor product polynomial approximation by H^2-projection to the space $\mathbb{Q}_p$ of polynomials in d variables with coordinatewise degree at most $p \in \mathbb{N}$. Evidently, $\dim(\mathbb{Q}_p) = (p+1)^d$.

For $d = 1$, we denote by $I_3 : H^2([-1,1], \mu_1) \to \mathbb{P}_3$ the Hermite interpolation operator defined by $I_3 u(\pm 1) = u(\pm 1)$ and $(I_3 u)'(\pm 1) = u'(\pm 1)$ for all $u \in H^2([-1,1], \mu_1)$. For $p \in \mathbb{N}$, $p \geq 3$, denote by $\pi_{p-2,0} : L^2([-1,1], \mu_1) \to \mathbb{P}_{p-2}$ the $L^2([-1,1], \mu_1)$-orthogonal projection to $\mathbb{P}_{p-2}$. For all $v \in L^2([-1,1], \mu_1)$ it holds that $v = \sum_{j=0}^{\infty} l_j L_j \mapsto \sum_{j=0}^{p-2} l_j L_j =: \pi_{p-2,0} v$, where analogous to (3.8) it holds $l_j = \int_{[-1,1]} v L_j \mathrm{d}\mu_1$. Bounds on Legendre coefficients required in DNN emulation expression rate bounds from Section 2.3.3 are implied by stability of $\pi_{p-2,0}$ in $L^2([-1,1], \mu_1)$:

$$\|\pi_{p-2,0} u\|^2_{L^2([-1,1],\mu_1)} = \sum_{j=0}^{p-2}|l_j|^2 \leq \sum_{j=0}^{\infty}|l_j|^2 = \|u\|^2_{L^2([-1,1],\mu_1)}.$$

Based on this projector, we define the $H^2([-1,1],\mu_1)$-projector $\pi_{p,2} : H^2([-1,1],\mu_1) \to \mathbb{P}_p$ by $\pi_{p,2}u(x) = I_3u(x) + \int_{-1}^{x}\int_{-1}^{y_1} \pi_{p-2,0}((u - I_3u)'')(y_2)\mathrm{d}y_2\mathrm{d}y_1$ for all $u \in H^2([-1,1],\mu_1)$, as in [7, Section A.1], where it is shown that $\pi_{p,2}u$ satisfies $\pi_{p,2}u(\pm1) = u(\pm1)$ and $(\pi_{p,2}u)'(\pm1) = u'(\pm1)$.

For general $d \in \mathbb{N}$, for all $p \in \mathbb{N}$, $p \geq 3$ we consider the tensor product projector $\Pi^d_{p,2} := \pi^{(1)}_{p,2}\otimes\cdots\otimes\pi^{(d)}_{p,2}$, where $\pi^{(i)}_{p,2}$ denotes the coordinate-wise projection with respect to x_j, $j = 1,\dots,d$. We recall stability and error bounds in terms of the $H^2_{\mathrm{mix}}([-1,1]^d,\mu_d)$-norm, which is defined as

$$\|u\|^2_{H^2_{\mathrm{mix}}([-1,1]^d,\mu_d)} = \sum_{\boldsymbol{\nu}:|\boldsymbol{\nu}|_\infty\leq 2} \left\| \frac{\partial^{|\boldsymbol{\nu}|_1} u}{\partial^{\nu_1}_{x_1}\cdots\partial^{\nu_d}_{x_d}} \right\|^2_{L^2([-1,1]^d,\mu_d)}.$$

By the continuous embedding $H^2_{\mathrm{mix}}([-1,1]^d,\mu_d) \hookrightarrow W^{1,\infty}([-1,1]^d)$, the bounds below imply error bounds w.r.t. the $W^{1,\infty}([-1,1]^d)$-norm. By [32, Propositions 5.2 and 5.3], the former of which is [7, Theorem A.1 and Proposition A.1], for all $u \in \mathcal{G}^\delta([-1,1]^d,C,A)$ it holds for all $p \in \mathbb{N}$, $p \geq 3$ and for all $s \in \{2,\dots,p-1\}$

$$\left\|\Pi^d_{p,2}u\right\|_{H^2_{\mathrm{mix}}([-1,1]^d,\mu_d)} \leq C(d)\left\|u\right\|_{H^2_{\mathrm{mix}}([-1,1]^d,\mu_d)}, \tag{B.1}$$

$$\begin{aligned}
\left\|u - \Pi^d_{p,2}u\right\|_{H^2_{\mathrm{mix}}([-1,1]^d,\mu_d)} &\leq C(d)\sum_{j=1}^{d}\left\|u - \pi^{(j)}_{p,2}u\right\|_{H^2_{\mathrm{mix}}([-1,1]^d,\mu_d)} \\
&\leq C(d)\sum_{j=1}^{d}\sqrt{\frac{(p-1-s)!}{(p-1+s)!}} \sum_{\substack{\nu_j=s+2,\\ \nu_i\in\{0,1,2\}\forall i\neq j}} \left\| \frac{\partial^{|\boldsymbol{\nu}|_1} u}{\partial^{\nu_1}_{x_1}\cdots\partial^{\nu_d}_{x_d}} \right\|_{L^2([-1,1]^d,\mu_d)} \\
&\leq C(d)\sum_{j=1}^{d}\sqrt{\frac{(p-1-s)!}{(p-1+s)!}} \sum_{\substack{\nu_j=s+2,\\ \nu_i\in\{0,1,2\}\forall i\neq j}} \left\| \frac{\partial^{|\boldsymbol{\nu}|_1} u}{\partial^{\nu_1}_{x_1}\cdots\partial^{\nu_d}_{x_d}} \right\|_{L^\infty([-1,1]^d)} \\
&\leq C(d,u)\max\{A,1\}^{s+2d}\sqrt{\frac{(p-1-s)!}{(p-1+s)!}}((s+2)!2^{d-1})^\delta.
\end{aligned} \tag{B.2}$$

We fix $\alpha = (4\max\{A,1\})^{-1/\delta} \in (0,1)$ and similar to [13, Proposition 4] substitute $s = \max\{2, \lfloor \alpha p^{1/\delta} \rfloor\}$. For sufficiently large $p \in \mathbb{N}$ it holds that $2 \leq \alpha p^{1/\delta}$ and thus $s \leq \alpha p^{1/\delta} \leq s+1$. The estimates that follow are derived under this assumption, but hold for all $p \geq 3$ after possibly increasing multiplicative constants. With Stirling's inequality, $\sqrt{2\pi}\sqrt{n}(n/e)^n \leq n! \leq e\sqrt{n}(n/e)^n$ for $n \in \mathbb{N}$, we estimate the square of the right-hand side of (B.2):

$$\begin{aligned}
&\max\{A,1\}^{2(s+2d)}\frac{(p-1-s)!}{(p-1+s)!}((s+2)!2^{d-1})^{2\delta} \\
\leq\ &\max\{A,1\}^{2(s+2d)}\frac{e^{2s+1}(p-1-s)^{p-1-s+1/2}}{\sqrt{2\pi}(p-1+s)^{p-1+s+1/2}}(s+2)^{4\delta}s^{\delta(2s+1)}e^{2\delta(1-s)}2^{2\delta(d-1)} \\
\leq\ &C(d,\delta,A)\max\{A,1\}^{2s}e^{2(1-\delta)s}\left(\tfrac{p-1-s}{p-1+s}\right)^{p-1-s+1/2}p^{-2s}s^{2\delta s}s^\delta(s+2)^{4\delta},
\end{aligned}$$

where we used $((s+2)!)^{2\delta} \leq (s+2)^{4\delta}(s!)^{2\delta}$. Now, since $1-\delta \leq 0$ and $p-1-s+\frac{1}{2} \geq 0$ for $s < p$,

$$e^{2(1-\delta)s}\left(\tfrac{p-1-s}{p-1+s}\right)^{p-1-s+1/2}p^{-2s} \leq Cp^{-2s} \leq (\alpha/s)^{2\delta s}$$

due to $s \leq \alpha p^{1/\delta}$. Using that there exists $C > 0$ depending on δ such that $s^\delta(s+2)^{4\delta} \leq C2^{2s}$

for all $s \geq 2$, we arrive at

$$\begin{aligned}\max\{A,1\}^{2(s+2d)}\frac{(p-1-s)!}{(p-1+s)!}((s+2)!2^{d-1})^{2\delta} &\leq C(d,\delta,A)\max\{A,1\}^{2s}(\alpha/s)^{2\delta s}s^{2\delta s}s^{\delta}(s+2)^{4\delta}\\ &\leq C(d,\delta,A)\max\{A,1\}^{2s}\alpha^{2\delta s}s^{\delta}(s+2)^{4\delta}\\ &\leq C(d,\delta,A)2^2 2^{-2(s+1)}\\ &\leq C(d,\delta,A)\exp(-2\log(2)\alpha p^{1/\delta}).\end{aligned}$$

Substituting into (B.2) shows that

$$\left\|u-\Pi_{p,2}^d u\right\|_{H^2_{\mathrm{mix}}([-1,1]^d,\mu_d)} \leq C(d,\delta,u)\exp(-\log(2)\alpha p^{1/\delta}).$$

Now, $\Pi_{p,2}^d u$ can be approximated by ReLU DNNs from Proposition 2.13. We set $\Lambda_p := \{\boldsymbol{\nu} : |\boldsymbol{\nu}|_\infty \leq p\}$, and express $\Pi_{p,2}^d u$ in the basis of tensor product Legendre polynomials. The size of the Legendre coefficients $(c_{\boldsymbol{\nu}})_{\boldsymbol{\nu}\in\Lambda_p}$ of $\Pi_{p,2}^d u$ can be estimated crudely by $|c_{\boldsymbol{\nu}}|^2 \leq \sum_{\boldsymbol{\nu}\in\Lambda_p}|c_{\boldsymbol{\nu}}|^2 = \left\|\Pi_{p,2}^d u\right\|^2_{L^2([-1,1]^d,\mu_d)} \leq C(d)\left\|u\right\|^2_{H^2_{\mathrm{mix}}([-1,1]^d,\mu_d)} = C(d,u)^2$ for all $\boldsymbol{\nu}\in\Lambda_p$, and their sum by $\sum_{\boldsymbol{\nu}\in\Lambda_p}|c_{\boldsymbol{\nu}}| \leq C(d,u)(p+1)^d$.

As in Step 1 in the proof of Theorem 3.6, we reapproximate the polynomial $\Pi_{p,2}^d u$ by $\hat{u}_p := \mathrm{Affine}_u \circ \boldsymbol{f}_{\Lambda_p,\delta}$, for $\boldsymbol{f}_{\Lambda_p,\delta}$ from Proposition 2.13 and $\mathrm{Affine}_u : \mathbb{R}^{|\Lambda_p|} \to \mathbb{R} : (z_{\boldsymbol{\nu}})_{\boldsymbol{\nu}\in\Lambda_p} \mapsto \sum_{\boldsymbol{\nu}\in\Lambda_p} c_{\boldsymbol{\nu}} z_{\boldsymbol{\nu}}$. We take $\delta = (p+1)^{-d}\exp(-\log(2)\alpha p^{1/\delta})$ as the accuracy parameter of $\boldsymbol{f}_{\Lambda_p,\delta}$, so that we obtain

$$\begin{aligned}\left\|\Pi_{p,2}^d u - \hat{u}_p\right\|_{W^{1,\infty}([-1,1]^d)} &\leq \sum_{\boldsymbol{\nu}\in\Lambda_p}|c_{\boldsymbol{\nu}}|\left\|L_{\boldsymbol{\nu}}-\tilde{L}_{\boldsymbol{\nu},\delta}\right\|_{W^{1,\infty}([-1,1]^d)} \leq \sum_{\boldsymbol{\nu}\in\Lambda_p}|c_{\boldsymbol{\nu}}|\,\delta\\ &= \sum_{\boldsymbol{\nu}\in\Lambda_p}|c_{\boldsymbol{\nu}}|\,(p+1)^{-d}\exp\left(-\log(2)\alpha p^{1/\delta}\right) \leq C(d,u)\exp\left(-\log(2)\alpha p^{1/\delta}\right).\end{aligned}$$

Together with the estimate of the polynomial interpolation error, it holds that

$$\left\|u-\hat{u}_p\right\|_{W^{1,\infty}([-1,1]^d)} \leq C(d,\delta,u)\exp\left(-\log(2)\alpha p^{1/\delta}\right).$$

We finally estimate the NN depth and size, using that $|\Lambda_p| = (p+1)^d$ and $m(\Lambda_p) = dp$:

$$\begin{aligned}\mathrm{depth}(\hat{u}_p) &\leq \mathrm{depth}(\mathrm{Affine}_u) + 1 + \mathrm{depth}(\boldsymbol{f}_{\Lambda_p,\delta})\\ &\leq C(1+d\log d)(1+\log_2 m(\Lambda_p))(m(\Lambda_p)+\log_2(1/\delta))\\ &\leq C(1+d\log d)(1+\log(dp))(dp+d\log_2(p+1)+\log(2)\alpha p^{1/\delta})\\ &\leq C(1+d^2\log^2 d)(1+p\log p),\\ \mathrm{size}(\hat{u}_p) &\leq 2\,\mathrm{size}(\mathrm{Affine}_u) + 2\,\mathrm{size}(\boldsymbol{f}_{\Lambda_p,\delta})\\ &\leq 2(p+1)^d + Cd^2m(\Lambda)^2 + Cdm(\Lambda)\log_2(1/\delta) + Cd^2|\Lambda|\big(1+\log_2 m(\Lambda)+\log_2(1/\delta)\big)\\ &\leq C(p+1)^d + Cd^2(dp)^2 + Cd(dp)(d\log_2(p+1)+\log(2)\alpha p^{1/\delta})\\ &\quad + Cd^2(p+1)^d\big(1+\log_2(dp)+d\log_2(p+1)+\log(2)\alpha p^{1/\delta}\big)\\ &\leq C_1(\alpha)d^4((p+1)^2+(p+1)^{d+1/\delta})\end{aligned}$$

for some $C_1(\alpha) > 0$. For all $\mathcal{N}\in\mathbb{N}$ satisfying $\mathcal{N} \geq C_1(\alpha)d^4(4^2+4^{d+1/\delta})$, we choose $p := \max\{p\in\mathbb{N} : C_1(\alpha)d^4((p+1)^2+(p+1)^{d+1/\delta}) \leq \mathcal{N}\}$, so that $p \geq 3$ and

$$C_1(\alpha)d^4((p+1)^2+(p+1)^{d+1/\delta}) \leq \mathcal{N} < C_1(\alpha)d^4((p+2)^2+(p+2)^{d+1/\delta}) \leq C(d,\delta,A)d^4p^{\max\{2,d+1/\delta\}},$$

and define $\tilde{u}_{\mathcal{N}} := \hat{u}_p$, which shows the proposition for such $\mathcal{N}$. For the finitely many $\mathcal{N}\in\mathbb{N}$ satisfying $\mathcal{N} < C_1(\alpha)d^4(4^2+4^{d+1/\delta})$, we define $\tilde{u}_{\mathcal{N}} := 0$, for which the proposition holds after increasing the constants, if necessary. This completes the proof of Proposition 4.1. □

References

[1] H. Bölcskei, P. Grohs, G. Kutyniok, and P. Petersen. Optimal approximation with sparsely connected deep neural networks. *SIAM Journal on Mathematics of Data Science*, 1(1):8–45, 2019. ArXiv:1705.01714.

[2] A. Bonito, R. DeVore, D. Guignard, P. Jantsch, and G. Petrova. Polynomial approximation of anisotropic analytic functions of several variables, 2019. ArXiv:1904.12105.

[3] N. Boullé, Y. Nakatsukasa, and A. Townsend. Rational neural networks, 2020. ArXiv:2004.01902.

[4] C. K. Chui and X. Li. Approximation by ridge functions and neural networks with one hidden layer. *J. Approx. Theory*, 70(2):131–141, 1992.

[5] A. Cohen, A. Chkifa, and C. Schwab. Breaking the curse of dimensionality in sparse polynomial approximation of parametric PDEs. *Journ. Math. Pures et Appliquees*, 103(2):400–428, 2015.

[6] A. Cohen, R. DeVore, and C. Schwab. Convergence rates of best N-term Galerkin approximations for a class of elliptic sPDEs. *Found. Comput. Math.*, 10(6):615–646, 2010.

[7] M. Costabel, M. Dauge, and C. Schwab. Exponential convergence of *hp*-FEM for Maxwell equations with weighted regularization in polygonal domains. *Math. Models Methods Appl. Sci.*, 15(4):575–622, 2005.

[8] P. Davis. *Interpolation and Approximation*. Dover Books on Mathematics. Dover Publications, 1975.

[9] J. J. Daws and C. G. Webster. A Polynomial-Based Approach for Architectural Design and Learning with Deep Neural Networks. *Proc. Machine Learning Research*, 107, 2020.

[10] W. E and Q. Wang. Exponential convergence of the deep neural network approximation for analytic functions. *Sci. China Math.*, 61(10):1733–1740, 2018.

[11] D. Elbrächter, P. Grohs, A. Jentzen, and C. Schwab. DNN Expression Rate Analysis of High-dimensional PDEs: Application to Option Pricing. Technical Report 1809.07669, arXiv, 2018.

[12] A. Erdélyi, W. Magnus, F. Oberhettinger, and F. G. Tricomi. *Higher transcendental functions*, volume 2. McGraw-Hill, New York, 1953. Based on notes left by Harry Bateman.

[13] M. Feischl and C. Schwab. Exponential convergence in H^1 of *hp*-FEM for Gevrey regularity with isotropic singularities. *Numerische Mathematik*, 144(2):323–346, 2020.

[14] L. Herrmann, C. Schwab, and J. Zech. Deep ReLU neural network expression rates for data-to-QoI maps in Bayesian PDE inversion. Technical Report 2020-02, Seminar for Applied Mathematics, ETH Zürich, Switzerland, 2020.

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

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

[17] B. Li, S. Tang, and H. Yu. Better approximations of high dimensional smooth functions by deep neural networks with rectified power units. *Communications in Computational Physics*, 27(2):379–411, 2020.

[18] S. Liang and R. Srikant. Why deep neural networks for function approximation? In *Proc. of ICLR 2017*, pages 1 – 17, 2017. ArXiv:1610.04161.

[19] H. Mhaskar. Neural networks for localized approximation of real functions. In *Neural Networks for Signal Processing III - Proceedings of the 1993 IEEE-SP Workshop*, pages 190–196. IEEE, 1993.

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

[21] H. N. Mhaskar. Approximation properties of a multilayered feedforward artificial neural network. *Advances in Computational Mathematics*, 1(1):61–80, Feb 1993.

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

[23] H. Montanelli, H. Yang, and Q. Du. Deep ReLU networks overcome the curse of dimensionality for bandlimited functions. *arXiv e-prints*, Mar 2019. ArXiv:1903.00735.

[24] F. Nobile, R. Tempone, and C. G. Webster. An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.*, 46(5):2411–2442, 2008.

[25] J. A. A. Opschoor, P. C. Petersen, and C. Schwab. Deep ReLU networks and high-order finite element methods. *Analysis and Applications*, 2020.

[26] J. A. A. Opschoor, C. Schwab, and J. Zech. DNN expression rates for Bayesian PDE inversion. Technical Report 2020-XY, Seminar for Applied Mathematics, ETH Zürich, Switzerland, 2020. (in review).

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

[28] A. Pinkus. Approximation theory of the MLP model in neural networks. In *Acta numerica, 1999*, volume 8 of *Acta Numer.*, pages 143–195. Cambridge Univ. Press, Cambridge, 1999.

[29] L. Rodino. *Linear partial differential operators in Gevrey spaces*. World Scientific Publishing Co., Inc., River Edge, NJ, 1993.

[30] D. Rolnik and M. Tegmark. The power of deeper networks for expressing natural functions. Technical Report 1705.05502v1, ArXiv, 2017.

[31] F. Santambrogio. *Optimal transport for applied mathematicians*, volume 87 of *Progress in Nonlinear Differential Equations and their Applications*. Birkhäuser/Springer, Cham, 2015. Calculus of variations, PDEs, and modeling.

[32] D. Schötzau, C. Schwab, and T. P. Wihler. *hp*-DGFEM for second order elliptic problems in polyhedra II: Exponential convergence. *SIAM J. Numer. Anal.*, 51(4):2005–2035, 2013.

[33] C. Schwab and J. Zech. Deep learning in high dimension: neural network expression rates for generalized polynomial chaos expansions in UQ. *Anal. Appl. (Singap.)*, 17(1):19–55, 2019.

[34] S. Tang, B. Li, and H. Yu. ChebNet: Efficient and stable constructions of deep neural networks with rectified power units using Chebyshev approximations, 2019. ArXiv: 1911.05467.

[35] M. Telgarsky. Neural networks and rational functions. In *34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3387–3393, 2017.

[36] H. Tran, C. G. Webster, and G. Zhang. Analysis of quasi-optimal polynomial approximations for parameterized PDEs with deterministic and stochastic coefficients. *Numer. Math.*, 137(2):451–493, 2017.

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

[38] S. T. Yau and L. Zhang. An upper estimate of integral points in real simplices with an application to singularity theory. *Math. Res. Lett.*, 13(5-6):911–921, 2006.

[39] J. Zech. Sparse-Grid Approximation of High-Dimensional Parametric PDEs. 2018. Dissertation 25683, ETH Zürich.

[40] J. Zech, D. Dung, and C. Schwab. Multilevel approximation of parametric and stochastic pdes. *M3AS*, 29(9):1753–1817, 2019.

[41] J. Zech and Y. Marzouk. Sparse approximation of triangular transports on bounded domains, 2020.

[42] J. Zech and C. Schwab. Convergence rates of high dimensional Smolyak quadrature. *ESAIM: Mathematical Modelling and Numerical Analysis*, 2020.