Applications of Mathematics

Joost A. A. Opschoor; Christoph Schwab

Exponential expressivity of ReLU^k neural networks on Gevrey classes with point singularities

Applications of Mathematics, Vol. 69 (2024), No. 5, 695–724

Persistent URL: http://dml.cz/dmlcz/152637

Terms of use:

© Institute of Mathematics CAS, 2024

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* http://dml.cz

EXPONENTIAL EXPRESSIVITY OF ReLU^k NEURAL NETWORKS ON GEVREY CLASSES WITH POINT SINGULARITIES

Joost A. A. Opschoor, Christoph Schwab, Zürich

Received March 5, 2024. Published online August 27, 2024.

Dedicated to the memory of Ivo Milan Babuška, with gratitude and highest esteem

Abstract. We analyze deep Neural Network emulation rates of smooth functions with point singularities in bounded, polytopal domains $\mathrm{D} \subset \mathbb{R}^d$, $d = 2, 3$. We prove exponential emulation rates in Sobolev spaces in terms of the number of neurons and in terms of the number of nonzero coefficients for Gevrey-regular solution classes defined in terms of weighted Sobolev scales in D, comprising the countably-normed spaces of I. M. Babuška and B. Q. Guo.

As intermediate result, we prove that continuous, piecewise polynomial high order ("p-version") finite elements with elementwise polynomial degree $p \in \mathbb{N}$ on arbitrary, regular, simplicial partitions of polyhedral domains $\mathrm{D} \subset \mathbb{R}^d$, $d \geqslant 2$, can be *exactly emulated* by neural networks combining ReLU and ReLU^2 activations.

On shape-regular, simplicial partitions of polytopal domains D, both the number of neurons and the number of nonzero parameters are proportional to the number of degrees of freedom of the hp finite element space of I. M. Babuška and B. Q. Guo.

Keywords: neural network; hp-finite element method; singularities; Gevrey regularity; exponential convergence

MSC 2020: 65N30, 41A25

1. Introduction

Recent years have seen the emergence of machine learning in scientific computing. One key component are *Deep Neural Network* (DNN for short) based numerical approximations of functions and operators. DNNs appear to impinge on nearly all applications of computation in science and engineering, including the numerical solution of Partial Differential Equations (PDEs). It is therefore of interest to explore,

Open access funding provided by Swiss Federal Institute of Technology Zürich.

DOI: 10.21136/AM.2024.0052-24

© The author(s) 2024.

mathematically, the consequences of this development for established methodologies in numerical analysis and scientific computing. The present paper addresses DNNs in the context of the *hp*-version of the Finite Element Method, as initiated by Babuška and Szabó [47] and coworkers. It will, in particular, develop *exact DNN emulations of hp-finite element (FE for short) spaces in polygonal and polyhedral domains* D. As a natural consequence, *existing exponential convergence rate results* for *hp*-FE approximation of elliptic boundary value problems (e.g., [5], [3], [46], [37], [47], [38] and the references there) will imply corresponding DNN approximation error bounds.

1.1. Previous work. In a precursor to this manuscript, [40], we constructed, in particular, deep ReLU neural network (NN for short) emulations of univariate *hp*-FE spaces (referred to also as "variable degree, variable knot splines"). In [41], an improved NN architecture to express high-order polynomials by ReLU NNs based on the emulation of Chebyšev polynomials was developed, implying sharper stability and expression rate bounds in some of the results in [40].

In [26], ReLU DNN emulations of first order, Lagrangian FE spaces ("Courant Finite Elements") on particular, regular triangulations of polytopal domains where constructed, and first bounds on size and depth of the NNs were proved. Admissible simplicial partitions in [26] were subject to certain patch conditions, which were essential in the DNN architecture.

The first ReLU NN emulation of first order, Lagrangian FE spaces on general regular simplicial partitions in any space dimension were constructed in [33], Section 4. The present results build, in part, on the constructions in [33].

Accordingly, the recent [27] is closely related to the present work. There, using the construction in [33], Section 4, DNN emulations of arbitrary order, Lagrangian FE spaces on general, regular simplicial partitions are developed, via DNNs which are ReLU and ReLU^2 activated. The constructions in [27] furnish exact DNN representations of Lagrangian FE spaces on regular, simplicial partitions of D, by emulating in the feature space of the DNNs the nodal, Lagrangian basis of arbitrary, fixed polynomial degree $p \geqslant 1$.

1.1.1. Finite elements. Founded in the mid-20th century, finite element discretizations are nowadays well-established discretization methodologies, in particular for elliptic and parabolic PDEs. Their mathematical underpinning (see, e.g., the text [17] and the references there) has, to some extent, an accepted set of terminology and notation, which we recapitulate here.

On a polytopal domain D of dimension $d \geqslant 2$, we consider regular, simplicial partitions $\mathcal{M}$. We start from the ReLU neural network emulation of continuous, piecewise linear functions, forming the classical "Courant finite elements", whose space is de-

noted by $\mathrm{S}_1(\mathcal{M},\mathrm{D})$, and consider the exact emulation of high-order, Lagrangian finite element spaces, for polynomial degree $p \in \mathbb{N} = \{1,2,3,\ldots\}$ defined as

$$\mathrm{S}_p(\mathcal{M},\mathrm{D}) = \{v \in C^0(\mathrm{D})\colon\ v|_K \in \mathbb{P}_p(K)\ \forall\, K \in \mathcal{M}\}.$$

We recall from [33], Section 1.4.1, the notation for the used partitions. For $k \in \{0,\ldots,d\}$ we define a k-simplex K by $K = \mathrm{conv}(\{a_0,\ldots,a_k\}) \subset \mathbb{R}^d$, for some $a_0,\ldots,a_k \in \mathbb{R}^d$ which do not all lie in one affine subspace of dimension $k-1$, and where

$$\mathrm{conv}(Y) := \left\{ x = \sum_{y\in Y} \lambda_y y\colon\ \lambda_y \geqslant 0 \ \text{ and } \ \sum_{y\in Y} \lambda_y = 1 \right\}$$

denotes the closed convex hull.[1] By $|K|$ we denote the k-dimensional Lebesgue measure of the k-simplex K. We consider a simplicial partition $\mathcal{M}$ of a polytopal, bounded domain $\mathrm{D} \subset \mathbb{R}^d$ into d-simplices, i.e., $\overline{\mathrm{D}} = \bigcup_{K\in\mathcal{M}} K$ and $\mathrm{int}\, K \cap \mathrm{int}\, K' = \emptyset$ for all $K \neq K'$. We assume that $\mathcal{M}$ is a *regular* partition, i.e., for all distinct $K, K' \in \mathcal{M}$ it holds that $K \cap K'$ is a k-subsimplex of K for some $k \in \{0,\ldots,d-1\}$, i.e., there exist $a_0,\ldots,a_d \in \overline{\mathrm{D}}$ such that $K = \mathrm{conv}(\{a_0,\ldots,a_d\})$ and $K \cap K' = \mathrm{conv}(\{a_0,\ldots,a_k\})$.

Recall that the *shape-regularity constant* $\kappa_{\mathrm{sh}}(\mathcal{M})$ of a simplicial partition $\mathcal{M}$ of D is $\kappa_{\mathrm{sh}} := \max_{K\in\mathcal{M}} h_K/r_K > 1$. Here $h_K := \mathrm{diam}(K)$ and r_K is the radius of the largest ball contained in K. Let $\mathcal{V}$ be the set of vertices of $\mathcal{M}$. For all $i \in \mathcal{V}$ we denote by $s(i) := |\{K \in \mathcal{M}\colon\ i \in K\}|$ the number of elements of $\mathcal{M}$ sharing the vertex i, and define $\mathfrak{s}(\mathcal{V}) := \max_{i\in\mathcal{V}} s(i)$. Throughout, we will use the notation $|S|$ for the cardinality of a finite set S.

R e m a r k 1.1. The constant $\mathfrak{s}(\mathcal{V})$ can be bounded in terms of the shape regularity constant $\kappa_{\mathrm{sh}}(\mathcal{M})$ (see [17], Remark 11.5 and Proposition 11.6, which generalize verbatim to dimension $d > 3$).

1.1.2. Exact NN emulation of Lagrangian, nodal finite elements. For a polytopal domain D and a regular, simplicial partition $\mathcal{M}$, an exact emulation of $\mathrm{S}_1(\mathcal{M},\mathrm{D})$ by ReLU NNs is provided in [1], but without efficient bounds on the NN depth and size. Such bounds were first obtained in [26], Section 3, under the assumption that the mesh has "convex patches", i.e., it was assumed that for all $i \in \mathcal{V}$ the set $\{K \in \mathcal{M}\colon\ i \in K\}$ is convex. It was shown that functions in $\mathrm{S}_1(\mathcal{M},\mathrm{D})$ can be emulated exactly by a ReLU NN of depth independent of $N = \dim(\mathrm{S}_1(\mathcal{M},\mathrm{D}))$ and size growing at most linearly in N, with constants depending on d and the shape

[1] In [33], conv denotes the *open convex hull*. As a result, in [33] simplices are open by definition. Here, simplices are closed by definition.

regularity of the mesh. By a geometrical construction in which possibly nonconvex patches are re-written as the finite union of convex patches, the results from [26], Section 3, were extended to general regular, simplicial meshes, without assuming convexity of patches, in [33]. There, it was shown that also without assuming convexity of patches the depth is independent of $N = \dim(\mathrm{S}_1(\mathcal{M}, \mathrm{D}))$ and the size grows at most linearly in N, with constants depending on d and the shape regularity.

For the exact emulation of high order finite elements $\mathrm{S}_p(\mathcal{M}, \mathrm{D})$ for $p \in \mathbb{N}$, a first result using ReLU, ReLU^2 and *binary step unit* (BiSU) activations was given in [33], Section 7.1. The ReLU, ReLU^2 and BiSU activation functions $\mathbb{R} \to \mathbb{R}$ are defined by $x \mapsto \max\{0, x\}$, $x \mapsto \max\{0, x\}^2$ and $x \mapsto 1$ for $x > 0$ and $x \mapsto 0$ for $x \leqslant 0$, respectively. The networks constructed in [33], Section 7.1 have depth bounded by $Cd\log_2(p+1)$ and size bounded by $C|\mathcal{M}|(p+1)^d$, which is proportional to the number of degrees of freedom (i.e., the dimension) of the FE space. Although this emulation result is efficient, it is not completely satisfactory, as the use of discontinuous BiSUactivations for the emulation of continuous FE basis functions could be considered somewhat unnatural. The use of ReLU^2 activations for the exact emulation of high order finite elements is natural, as it allows the exact emulation of products (cf. Proposition 3.8).

The exact emulation of $\mathrm{S}_p(\mathcal{M}, \mathrm{D})$ by NNs with only ReLU and ReLU^2 activations was first given in [27]. The main insight is that on an element $K \in \mathcal{M}$, the "hat" basis functions of $\mathrm{S}_1(\mathcal{T}, \mathrm{D})$, which equal 1 in one vertex $i \in \mathcal{V}$ and vanish in all other vertices in $\mathcal{V}$, coincide with the barycentric coordinates on K. Elements of $\mathrm{S}_p(\mathcal{M}, \mathrm{D})$ can therefore be written as sums of products of "hat" basis functions. In [27], the NNs from [33] which exactly emulate $\mathrm{S}_1(\mathcal{T}, \mathrm{D})$ are combined with product subnetworks to obtain an exact emulation of elements in $\mathrm{S}_p(\mathcal{M}, \mathrm{D})$. The size of the networks constructed in [27] is larger than the number of degrees of freedom $\dim(\mathrm{S}_p(\mathcal{M}, \mathrm{D}))$, which is suboptimal, as we will show (see Remark 4.5). A detailed discussion of the results in [27] and the complexity of the NNs constructed there is the topic of Section 4.1.

1.1.3. NN emulation of *hp*-finite elements. The literature on exponential convergence of NN approximations of functions with point singularities based on the NN approximation of *hp*-finite elements goes back to [40], where it is shown that there exist exponentially convergent ReLU NN approximations of univariate weighted Gevrey regular functions (see Section 2.1 for a definition of weighted Gevrey regular functions). These functions, which are defined on a bounded interval, are smooth everywhere except of a finite set of singular points. In that work, only ReLU activations were used, i.e., multiplications could not be realized exactly (as ReLU NNs realize continuous, piecewise linear functions) and were approximated using networks introduced in [49] whose size depends logarithmically on their accuracy.

Approximating in this way all products which are needed for the approximation of high order finite elements results in a NN whose size grows more quickly with the polynomial degree than the number of degrees of freedom of the *hp*-finite elements which it approximates.

For $d = 2, 3$, for weighted analytic functions on polygons with point singularities and on polyhedra with point- and edge singularities, the existence of exponentially convergent ReLU NN approximations was shown in [35]. Again, multiplications were approximated by ReLU subnetworks, leading to an inexact approximation of *hp*-finite elements and a network size that is larger than the number of degrees of freedom. For the approximation of weighted analytic functions on $(0, 1)^d$, the networks constructed in [35] are based on tensor product *hp*-finite elements on tensor product meshes with rectangular elements. To approximate weighted analytic functions on domains which do not have a tensor product structure, multiple such approximations are combined using a ReLU partition of unity.

While in [35], exponentially convergent networks are shown to exist, it is not clear how network parameters (weights and biases) that realize this exponential convergence can be computed based on a finite number of function evaluations. This is the topic of [39]. There, exponentially convergent ReLU NNs are constructed which approximate weighted analytic functions on polygons that have point singularities in the vertices of the domain. The NN construction in [39] is similar to, but different from that in [35]. The main difference is that local polynomial approximations on rectangular elements in tensor product partitions of $(0, 1)^d$, which make up the *hp*-finite elements on $(0, 1)^d$, are computed by nodal interpolation, rather than the tensor product H^1 projection used in [35]. This means that the network parameters can be computed explicitly based on a finite number of queries of the approximated function, whose number grows at most polylogarithmically with the accuracy. Also, it means that we can use existing bounds on the Lebesgue constants for polynomial interpolation to show stability of the NN approximations.

1.2. Contributions. In our main result, Theorem 5.1, we establish, for functions from weighted analytic and weighted Gevrey regular spaces with point singularities in bounded, polytopal domains D in Euclidean space of dimension $d = 2$ or $d = 3$, *exponential expressivity bounds of certain deep neural networks.* The NNs under consideration are deep feedforward NNs which encode so-called *hp* finite element approximations on shape-regular, simplicial partitions of D with geometric refinement(s) towards the singular support of the function to be emulated. We consider singular support sets $\mathcal{S} \subset \partial$D which are contained in the boundary.[2]

[2] The present analysis also covers point singularities in the interior of D, when they are located at a vertex of the used regular, simplicial triangulation of the domain.

Compared to [35], [39], our construction combines both ReLU and ReLU^2 activations, rather than only ReLU activations. This means that products of real numbers can be emulated exactly and allows us to obtain an NN size which is proportional to the number of degrees of freedom in the finite-element space. The use of two different activation functions throughout our networks means that the used architectures are slightly more complex than pure ReLU networks, as for every node in the network, the choice of activation function has to be specified. As in [48], in each layer, the same activation is applied in all positions in that layer. The network consists of several layers with ReLU activation, followed by several layers with ReLU^2 activation. The ReLU layers emulate continuous, piecewise linear hat functions on a regular, simplicial triangulation. The number of such layers depends on the shape regularity of the triangulation. It depends logarithmically on the dimension d of the domain, but it is independent of the polynomial degree p. The number of ReLU^2 layers is independent of the triangulation and its shape regularity, but depends logarithmically on p and on d.

Our construction works with arbitrary regular, simplicial partitions, which makes it simpler than the partition of unity-*hp*-constructions in [35], [39]. We do not provide exponential rate bounds to approximate edge singularities in three space dimensions, but extend previous convergence rate bounds which only hold for weighted analytic functions to weighted Gevrey classes with isolated point singularities. Isolated point singularities arise in space dimension $d = 2$ as corner and interface singularities in solutions to elliptic BVPs with analytic data in polygons [2], [22], [25], and in space dimension $d = 3$ in a number of applications [10], [34], [20], [45].

Our networks are constructed by emulating continuous, piecewise polynomial nodal Lagrangian basis functions (see Remark 4.8). Therefore, as in [29], [39], nodal interpolation could be used to construct NN approximations and explicitly compute their parameters (weights and biases), provided that point values of the function to be approximated can be queried. We could compute all NN parameters based on only a finite number of function evaluations, whose number equals the number of degrees of freedom of the underlying *hp* finite elements and grows polylogarithmically with the accuracy.

The NNs constructed in Section 4.2, which exactly emulate high order finite elements, are of independent interest, also beyond the current context of *hp*-finite elements. We provide exact neural network emulations for any continuous, piecewise polynomial function of total degree $p \in \mathbb{N}$ on a regular, simplicial partition $\mathcal{M}$ of a polytopal domain $\mathrm{D} \subset \mathbb{R}^d$, $d \geqslant 2$. We show that on regular partitions $\mathcal{M}$ of D into simplices, the neural network size is proportional to the number of degrees of freedom, see Proposition 4.6 and its discussion in Remark 4.7. Here, we improve

upon [33] in the sense that we obtain (assuming exact arithmetic) exact emulations with the same order of computational complexity, but without using discontinuous BiSU activations. As compared to [27], the presently developed NN architectures exactly realize *hp*-finite element feature spaces on regular, simplicial partitions $\mathcal{M}$ of D with smaller network sizes.

1.3. Layout. The structure of this paper is as follows. In Section 2.1, we recall weighted function classes for characterizing the smoothness of functions with point singularities and introduce classes of weighted Gevrey regular functions, which include as a special case ($\delta = 1$) weighted analytic functions. In Section 2.2, we recall *hp*-approximations of such functions and their exponential convergence. In Section 3, we fix notation and recall basic terminology and results from deep neural networks, as needed subsequently. The topic of Section 4 is the exact emulation of high order finite elements by neural networks. In Section 4.1, we discuss previous results from [27]. Based on an observation from [27], a new and more efficient NN emulation is constructed in Section 4.2. This is used in Section 5 to obtain exponentially convergent NN approximations of Gevrey regular functions in polygonal or polyhedral domains with point singularities. Section 6 concludes the paper with an overview of the main results and a discussion of the conditioning of the used Lagrangian finite element basis.

2. Regularity and *hp*-approximation of point singularities

We review definitions and results from weighted, countably-normed function classes, as used in *hp*-FE approximation theory, and in the corresponding elliptic regularity. These corner-weighted, analytic classes were introduced in the late 80ies by Babuška and Guo and by Bolley, Camus and Dauge in their pioneering works [21], [9], [2], [4], [22], [5] and the references there in space dimension $d = 2$, and in space dimension $d = 3$ in [23], [24], [14].

Throughout, $\mathrm{D} \subset \mathbb{R}^d$ denotes an open, bounded, polytopal domain in Euclidean space of dimension $d = 2, 3$. We denote by $\mathcal{S} \subset \partial\mathrm{D}$ a finite set of singular points. We consider solutions $u \in H^1(\mathrm{D})$ which are smooth in $\overline{\mathrm{D}} \setminus \mathcal{S}$ so that the singular support of u coincides with $\mathcal{S}$. This allows us to determine in D a collection of $|\mathcal{S}|$ many disjoint open sets $\omega_c \subset \mathrm{D}$ containing exactly one singularity $c \in \mathcal{S}$ in their closure. We denote $\mathrm{D}_0 := \mathrm{D} \setminus \overline{\bigcup_{c\in\mathcal{S}} \omega_c}$.

2.1. Weighted Sobolev spaces and Gevrey classes. We characterize analytic regularity of singular solutions by weighted Sobolev spaces. To define these, we

follow [19], Section 2.1, and introduce distance functions to a corner point $c \in \mathcal{S}$:

$$r_c(x) = \operatorname{dist}(x, c), \quad x \in \mathrm{D}, \quad c \in \mathcal{S}. \tag{2.1}$$

For each conical point $c \in \mathcal{S}$, a *singular exponent* $\beta_c \in \mathbb{R}$ quantifies the allowed strength of the singularity at c. We collect all singular exponents in the "multi-weight-exponent"

$$\underline{\beta} = \{\beta_c \colon\ c \in \mathcal{S}\} \in \mathbb{R}^{|\mathcal{S}|}. \tag{2.2}$$

We assume for $d = 3$ ($\underline{\beta} > s$ and $\underline{\beta} \pm s$ being understood componentwise for $s \in \mathbb{R}$)

$$\underline{b} := -1 - \underline{\beta} \in (0, 1/2), \text{ i.e., } -3/2 < \underline{\beta} < -1. \tag{2.3}$$

For $d = 2$, we assume for some $\varepsilon > 0$ that

$$\underline{b} := -1 - \underline{\beta} \in (0, \varepsilon), \text{ i.e., } -1 - \varepsilon < \underline{\beta} < -1. \tag{2.4}$$

We consider the *inhomogeneous, corner-weighted semi-norms* $|u|_{N^k_{\underline{\beta}}(\mathrm{D})}$ given by [14], Definition 6.2 and (6.9), [22] and [23],

$$|u|^2_{N^k_{\underline{\beta}}(\mathrm{D})} = |u|^2_{H^k(\mathrm{D}_0)} + \sum_{c \in \mathcal{S}} \sum_{\substack{\alpha \in \mathbb{N}_0^d \\ |\alpha| = k}} \|r_c^{\max\{\beta_c + |\alpha|, 0\}} D^\alpha u\|^2_{L^2(\omega_c)}, \quad k \in \mathbb{N}_0. \tag{2.5}$$

We define the *inhomogeneous weighted norm* $\|u\|_{N^m_{\underline{\beta}}(\mathrm{D})}$ by $\|u\|^2_{N^m_{\underline{\beta}}(\mathrm{D})} = \sum_{k=0}^{m} \|u\|^2_{N^k_{\underline{\beta}}(\mathrm{D})}$. Here, $|u|_{H^m(\mathrm{D}_0)}$ signifies the Hilbertian Sobolev semi-norm of integer order m on D_0, and D^α denotes the weak partial derivative of order $\alpha \in \mathbb{N}_0^d$. The space $N^m_{\underline{\beta}}(\mathrm{D})$ is the weighted Sobolev space obtained as the closure of $C_0^\infty(\mathrm{D})$ with respect to the norm $\|\cdot\|_{N^m_{\underline{\beta}}(\mathrm{D})}$.

R e m a r k 2.1. The weighted spaces are related, for particular ranges of the weight parameters, to various other corner-weighted scales of Sobolev spaces. We refer to [2], [24], [23], [13], [14] and references there and the discussion in [19] for details.

With $N^k_{\underline{\beta}}(\mathrm{D})$ as defined in (2.5), for $\delta > 0$ we define the *$\underline{\beta}$-weighted δ-Gevrey regular class of functions with point singularities at $\mathcal{S}$* by

$$\mathcal{G}^\delta_{\underline{\beta}}(\mathcal{S}; \mathrm{D}) = \Big\{ u \in \bigcap_{k \geqslant 0} N^k_{\underline{\beta}}(\mathrm{D}) \colon\ \exists\, C_u > 0 \text{ s.t. } |u|_{N^k_{\underline{\beta}}(\mathrm{D})} \leqslant C_u^{k+1} (k!)^\delta\ \forall\, k \in \mathbb{N}_0 \Big\}. \tag{2.6}$$

A wide range of partial differential equations is known to admit singular solutions in weighted Gevrey classes, for example, nonlinear Schrödinger equations in electron structure calculations [8], [10], [20], nonlinear parabolic PDEs with critical growth [31], [45], incompressible Euler equations [11], linear elasticity [22], station-

ary Stokes [25] and stationary, incompressible Navier Stokes [36], [28]; see also the references of the cited papers. We refer to [19], Section 2.2, for more examples and a more detailed exposition, which includes the standard example of a linear, elliptic, second order PDE with weighted analytic solutions, in a polygon.

2.2. ***hp*-Approximation.** It is well-known (e.g., [19] and the references there) that functions $u \in \mathcal{G}^{\delta}_{\vec{\beta}}(\mathcal{S};\mathrm{D})$ admit approximations from systems of continuous, piecewise polynomial functions at exponential rates in terms of the number N of degrees of freedom defining the approximations. In these "*hp*-" or "variable mesh and degree" approximations, geometric subdivisions towards the singular support of u are coupled to an increase in polynomial degree. We recapitulate from [19], Section 3, the construction of these approximations, and the corresponding exponential approximation rate bounds. These comprise corresponding results first obtained by Babuška and his coworkers in the analytic case (where $\delta = 1$ in (2.6)), see, e.g., [21], [5], [3], and will form the basis for corresponding DNN emulation rate bounds.

2.2.1. Geometric meshes. We recall from [19], Section 3.1, the notion of *geometric mesh sequences* $\mathfrak{M}_{\kappa,\sigma} = \{\mathcal{M}^{(l)}\}_{l\geqslant 1}$ in D. Such mesh families in D constitute an essential ingredient in the exponential convergence analysis of *hp*-approximations [21], [5], [3], [46], [19]. Specifically, geometric mesh sequences are sequences of regular, simplicial partitions of D for which there exist two parameters $\sigma \in (0,1)$ and $\kappa > 1$ with the following properties:

(i) All elements $K \in \mathcal{M}^{(l)}$, $l = 1, 2, \ldots$, are uniformly κ-shape regular, i.e., there exists a constant $\kappa > 1$ such that $\sup_{l\in\mathbb{N}} \kappa_{\mathrm{sh}}(\mathcal{M}^{(l)}) \leqslant \kappa$.

(ii) The partitions $\mathcal{M}^{(l)} \in \mathfrak{M}_{\kappa,\sigma}$ are σ-geometric, i.e., for every $K \in \mathcal{M}^{(l)}\colon\ K \cap \mathcal{S} = \emptyset$, $l = 1, 2, \ldots$, the following inequalities hold:

$$0 < \sigma < \frac{\mathrm{diam}(K)}{\mathrm{dist}(K,\mathcal{S})} < \frac{1}{\sigma}. \tag{2.7}$$

It was shown in [19], Proposition 1, that (κ,σ)-geometric mesh sequences $\mathfrak{M}_{\kappa,\sigma}$ in D have the following geometric properties: for every $l > d$ all elements $K \in \mathcal{M}^{(l)}$ can be grouped in *mesh layers*; there exists a partition

$$\mathcal{M}^{(l)} = \mathfrak{O}^{(l)} \dot\cup \mathfrak{T}^{(l)}, \quad \text{where } |\mathfrak{T}^{(l)}| \leqslant C_{\mathfrak{T}}(\kappa,\sigma), \tag{2.8}$$

and, for $k \simeq l\log(2)/|\log(\sigma)|$, there are partitions

$$\mathfrak{O}^{(l)} = \mathfrak{L}^{(l)}_1 \dot\cup \mathfrak{L}^{(l)}_2 \dot\cup \ldots \dot\cup \mathfrak{L}^{(l)}_k, \tag{2.9}$$

such that there exists $c_{\mathfrak{T}} > 0$ such that for all l it holds that

$$\mathcal{S} \subset \bigcup_{K \in \mathfrak{T}^{(l)}} K, \quad \operatorname{dist}(\mathcal{S}, \mathfrak{O}^{(l)}) \geqslant c_{\mathfrak{T}} \sigma^k. \tag{2.10}$$

Furthermore, there exists a constant $c(\mathfrak{M}_{\kappa,\sigma}) \geqslant 1$ with

$$\forall\, j = 1, \dots, k\colon\ |\mathfrak{L}_j^{(l)}| \leqslant c(\mathfrak{M}_{\kappa,\sigma}) \tag{2.11}$$

and such that for every $j = 1, \dots, k$ and for every $K \in \mathfrak{L}_j^{(l)}$,

$$0 < \frac{1}{c(\mathfrak{M}_{\kappa,\sigma})} \leqslant \frac{\operatorname{diam}(K)}{\sigma^j} \leqslant c(\mathfrak{M}_{\kappa,\sigma}). \tag{2.12}$$

We will frequently use that (2.8), (2.9) and (2.11) imply

$$|\mathcal{M}^{(l)}| \simeq k \simeq l. \tag{2.13}$$

As shown in [19], Proposition 1, for a given polytopal domain D, singular set $\mathcal{S}$ and regular, simplicial initial triangulation $\mathcal{M}^{(0)}$, [19], Algorithm 1 provides an explicit construction of a κ-shape-regular and σ-geometrically refined mesh sequence $\mathfrak{M}_{\kappa,\sigma}$.

2.2.2. Exponential convergence. Based on the geometric mesh sequences $\mathfrak{M}_{\kappa,\sigma}$ in D, for $u \in \mathcal{G}^{\delta}_{\underline{\beta}}(\mathcal{S}; \mathrm{D})$, there exist sequences of continuous, piecewise polynomial (on $\mathcal{M}^{(l)} \in \mathfrak{M}_{\kappa,\sigma}$) functions in D approximating u at an exponential rate.

Proposition 2.2 ([19], Theorem 1)**.** *In a bounded polytope* $\mathrm{D} \subset \mathbb{R}^d$, $d = 2, 3$, *with plane sides or plane faces, suppose given a weight vector* $\underline{\beta}$ *as in* (2.3) *if* $d = 3$ *or* (2.4) *if* $d = 2$.

Then for every sequence $\mathfrak{M}_{\kappa,\sigma}(\mathcal{S})$ *of nested, regular simplicial meshes in* D *which are* σ*-geometrically refined towards* $\mathcal{S}$ *and which are* κ*-shape regular, there exist continuous projectors* $\Pi^p_{\kappa,\sigma}\colon N^2_{\underline{\beta}}(\mathrm{D}) \to S^p(\mathcal{M}^{(l)}, \mathrm{D})$ *with* $l \simeq p^{1/\delta}$ *and for every* $u \in \mathcal{G}^{\delta}_{\underline{\beta}}(\mathcal{S}; \mathrm{D})$ *there exist constants* $b, C > 0$ *(depending on* κ, C_u, d_u *in* (2.6) *and on* σ*) such that the following error bound holds:*

$$\|u - \Pi^p_{\kappa,\sigma} u\|_{H^1(\mathrm{D})} \leqslant C \begin{cases} \exp(-bN^{1/(1+\delta d)}), & \delta \geqslant 1, \\ (\Gamma(N^{1/(1+\delta d)}))^{-b(1-\delta)}, & 0 < \delta < 1. \end{cases} \tag{2.14}$$

Here,

$$N = \dim(S^p(\mathcal{M}^{(l)}, \mathrm{D})) \simeq |\mathcal{M}^{(l)}| p^d \simeq l p^d \simeq p^{d+1/\delta}.$$

If, additionally, $u|_{\partial \mathrm{D}} = 0$, *then* $(\Pi^p_{\kappa,\sigma} u)|_{\partial \mathrm{D}} = 0$ *and* (2.14) *holds.*

3. Neural networks

The continuous, piecewise polynomial approximations in $S^p(\mathcal{M}^{(l)})$ of $u \in \mathcal{G}^{\delta}_{\underline{\beta}}(\mathcal{S};\mathrm{D})$ imply the existence of deep neural networks with corresponding exponential approximation rates. In Sections 3–4, we develop a rigorous statement of this fact.

Section 3.1 introduces basic notation and NN terminology, from [33] and the references there. Section 3.2 recalls, from [33], a result on the exact representation of continuous, piecewise linear (CPwL) functions on the geometric partitions $\mathcal{M}^{(l)}$ in Proposition 2.2, by means of a ReLU-activated NN in D.

The main results on NN approximation of $u \in \mathcal{G}^{\delta}_{\underline{\beta}}(\mathcal{S};\mathrm{D})$ are then developed in Section 4.2 ahead.

3.1. Neural network definitions.

Definition 3.1 (Neural Network [33], Section 2.1)**.** For $d, L \in \mathbb{N}$, a *neural network* Φ with input dimension $d \geqslant 1$ and number of layers $L \geqslant 1$, comprises a finite collection of activation functions[3] $\boldsymbol{\varrho} = \{\varrho_l\}_{l=1}^{L}$ and a finite sequence of matrix-vector tuples, i.e.,

$$\Phi = ((A_1, b_1, \varrho_1), (A_2, b_2, \varrho_2), \ldots, (A_L, b_L, \varrho_L)).$$

For $N_0 := d$ and *numbers of neurons* $N_1, \ldots, N_L \in \mathbb{N}$ *per layer*, for all $l = 1, \ldots, L$ it holds that $A_l \in \mathbb{R}^{N_l \times N_{l-1}}$ and $b_l \in \mathbb{R}^{N_l}$, and that ϱ_l is a list of length N_l of *activation functions* $(\varrho_l)_i \colon \mathbb{R} \to \mathbb{R}$, $i = 1, \ldots, N_l$, acting on node i in layer l.

The *realization* of Φ as a map is the function

$$\mathrm{R}(\Phi) \colon \mathbb{R}^d \to \mathbb{R}^{N_L} \colon x \to x_L,$$

where

$$x_0 := x, \quad x_l := \varrho_l(A_l x_{l-1} + b_l) \text{ for } l = 1, \ldots, L.$$

Here, for $l = 1, \ldots, L$, the list of activation functions ϱ_l of length N_l is effected componentwise: for $y = (y_1, \ldots, y_{N_l}) \in \mathbb{R}^{N_l}$ we denote $\varrho_l(y) = ((\varrho_l)_1(y_1), \ldots, (\varrho_l)_{N_l}(y_{N_l}))$, i.e., $(\varrho_l)_i$ is the activation function applied in position i of layer l.

We call the layers indexed by $l = 1, \ldots, L-1$ *hidden layers*; in those layers activation functions are applied. We fix the activation function in the last layer of the NN to be the identity, i.e., $\varrho_L := \mathrm{Id}_{\mathbb{R}^{N_L}}$.

We refer to $L(\Phi) := L$ as the *depth* of Φ. For $l = 1, \ldots, L$ we denote by $M_l(\Phi) := \|A_l\|_0 + \|b_l\|_0$ the *size of layer* l, which is the number of nonzero components in the weight matrix A_l and the bias vector b_l, and call $M(\Phi) := \sum_{l=1}^{L} M_l(\Phi)$ the *size* of Φ.

[3] No activation is applied in the output layer L. We introduce ϱ_L only for consistency of notation, and define it to be equal to the identity function.

Furthermore, we call d and N_L the *input dimension* and the *output dimension*, and denote by $M_{\text{in}}(\Phi) := M_1(\Phi)$ and $M_{\text{out}}(\Phi) := M_L(\Phi)$ the size of the first and the last layer, respectively.

Our networks will use two different activation functions. Firstly, we use the *Rectified Linear Unit* (ReLU) activation

$$\varrho(x) = \max\{0, x\}. \tag{3.1}$$

Networks which only contain ReLU activations realize continuous, piecewise linear functions. By ReLU *NNs* we refer to NNs which only have ReLU activations, including networks of depth 1, which do not have hidden layers and realize affine transformations. Secondly, for the emulation of high-order finite element methods, we use the ReLU^2 activation

$$\varrho^2(x) = \max\{0, x\}^2. \tag{3.2}$$

R e m a r k 3.2. Upper bounds on the network size also provide upper bounds on the number of neurons. Without loss of generality, the network size is bounded from below by $\sum_{j=1}^{L} N_j$, see [44], Lemma G.1. There, it was proved formally that each neuron to which no nonzero weights are associated can be omitted.

In the following sections, we will construct NNs from smaller networks using a *calculus of NNs*, which we now recall from [44]. The results cited from [44] were derived for NNs which only use the ReLU activation function, but they also hold for networks with multiple activation functions without modification.

Proposition 3.3 (Parallelization of NNs [44], Definition 2.7)**.** *For $d, L \in \mathbb{N}$ let*

$$\Phi^1 = ((A_1^{(1)}, b_1^{(1)}, \varrho_1^{(1)}), \dots, (A_L^{(1)}, b_L^{(1)}, \varrho_L^{(1)}))$$

and

$$\Phi^2 = ((A_1^{(2)}, b_1^{(2)}, \varrho_1^{(2)}), \dots, (A_L^{(2)}, b_L^{(2)}, \varrho_L^{(2)}))$$

be two NNs *with input dimension d and depth L. Let the parallelization $\mathrm{P}(\Phi^1, \Phi^2)$ of Φ^1 and Φ^2 be defined by*

$$\begin{gathered}
\mathrm{P}(\Phi^1, \Phi^2) := ((A_1, b_1, \varrho_1), \dots, (A_L, b_L, \varrho_L)), \\
A_1 = \begin{pmatrix} A_1^{(1)} \\ A_1^{(2)} \end{pmatrix}, \quad A_l = \begin{pmatrix} A_l^{(1)} & 0 \\ 0 & A_l^{(2)} \end{pmatrix} \quad \text{for } l = 2, \dots, L, \\
b_l = \begin{pmatrix} b_l^{(1)} \\ b_l^{(2)} \end{pmatrix}, \quad \varrho_l = \begin{pmatrix} \varrho_l^{(1)} \\ \varrho_l^{(2)} \end{pmatrix} \quad \text{for } l = 1, \dots, L.
\end{gathered}$$

Then

$$\mathrm{R}(\mathrm{P}(\Phi^1,\Phi^2))(x) = (\mathrm{R}(\Phi^1)(x),\mathrm{R}(\Phi^2)(x)) \quad \forall\, x \in \mathbb{R}^d,$$
$$L(\mathrm{P}(\Phi^1,\Phi^2)) = L, \quad M(\mathrm{P}(\Phi^1,\Phi^2)) = M(\Phi^1) + M(\Phi^2).$$

The parallelization of more than two NNs is handled by repeated application of Proposition 3.3.

We will also use the parallelization of networks which do not have the same inputs.

Proposition 3.4 (Full parallelization of NNs [16], Setting 5.2)**.** *For $L \in \mathbb{N}$ let $\Phi^1 = ((A_1^{(1)}, b_1^{(1)}), \dots, (A_L^{(1)}, b_L^{(1)}))$ and $\Phi^2 = ((A_1^{(2)}, b_1^{(2)}), \dots, (A_L^{(2)}, b_L^{(2)}))$ be two NNs with the same depth L, with input dimensions $N_0^1 = d_1$ and $N_0^2 = d_2$, respectively. Let the full parallelization of Φ^1 and Φ^2 be defined by*

$$\mathrm{FP}(\Phi^1,\Phi^2) := ((A_1, b_1, \varrho_1), \dots, (A_L, b_L, \varrho_L)),$$
$$A_l = \begin{pmatrix} A_l^{(1)} & 0 \\ 0 & A_l^{(2)} \end{pmatrix}, \quad b_l = \begin{pmatrix} b_l^{(1)} \\ b_l^{(2)} \end{pmatrix}, \quad \varrho_l = \begin{pmatrix} \varrho_l^{(1)} \\ \varrho_l^{(2)} \end{pmatrix} \quad \text{for } l = 1, \dots, L.$$

Then it has $d = d_1 + d_2$-dimensional input, depth L and size $M(\mathrm{FP}(\Phi^1,\Phi^2)) = M(\Phi^1) + M(\Phi^2)$. It satisfies for all $x = (x_1, x_2) \in \mathbb{R}^d$ with $x_i \in \mathbb{R}^{d_i}$, $i = 1, 2$,

$$\mathrm{R}(\mathrm{FP}(\Phi^1,\Phi^2))(x_1,x_2) = (\mathrm{R}(\Phi^1)(x_1),\mathrm{R}(\Phi^2)(x_2)).$$

Next, we define the concatenation of two NNs.

Definition 3.5 (Concatenation of NNs [44], Definition 2.2)**.** For $L^{(1)}$, $L^{(2)} \in \mathbb{N}$, let

$$\Phi^1 = ((A_1^{(1)}, b_1^{(1)}, \varrho_1^{(1)}), \dots, (A_{L^{(1)}}^{(1)}, b_{L^{(1)}}^{(1)}, \varrho_{L^{(1)}}^{(1)}))$$

and

$$\Phi^2 = ((A_1^{(2)}, b_1^{(2)}, \varrho_1^{(2)}), \dots, (A_{L^{(2)}}^{(2)}, b_{L^{(2)}}^{(2)}, \varrho_{L^{(2)}}^{(2)}))$$

be two NNs such that the input dimension of Φ^1 equals the output dimension of Φ^2. Then the *concatenation* of Φ^1 and Φ^2 is the NN of depth $L := L^{(1)} + L^{(2)} - 1$ defined as

$$\begin{aligned}
&\Phi^1 \bullet \Phi^2 := ((A_1, b_1, \varrho_1), \dots, (A_L, b_L, \varrho_L)),\\
&(A_l, b_l, \varrho_l) = (A_l^{(2)}, b_l^{(2)}, \varrho_l^{(2)}) \quad \text{for } l = 1, \dots, L^{(2)} - 1,\\
&\qquad A_{L^{(2)}} = A_1^{(1)} A_{L^{(2)}}^{(2)}, \quad b_{L^{(2)}} = A_1^{(1)} b_{L^{(2)}}^{(2)} + b_1^{(1)}, \quad \varrho_{L^{(2)}} = \varrho_1^{(1)},\\
&(A_l, b_l, \varrho_l) = (A_{l-L^{(2)}+1}^{(1)}, b_{l-L^{(2)}+1}^{(1)}, \varrho_{l-L^{(2)}+1}^{(1)}) \quad \text{for } l = L^{(2)} + 1, \dots, L^{(1)} + L^{(2)} - 1.
\end{aligned}$$

It follows immediately from this definition that $\mathrm{R}(\Phi^1 \bullet \Phi^2) = \mathrm{R}(\Phi^1) \circ \mathrm{R}(\Phi^2)$. In addition, we define the sparse concatenation of two NNs based on the ReLU^2 activation, which also exactly realizes the composition of the realizations of the two networks. The sparse concatenation is a construction which allows to bound the size of the concatenation as a constant times the sum of the sizes of the individual networks. This bound does not hold if we combine the affine transformation of the output layer of Φ^2 with the affine transformation of the input layer of Φ^1, as we did in Definition 3.5.

Proposition 3.6 (Sparse Concatenation of NNs based on ReLU^2 [43], Proposition 2.4)**.** *For $L^{(1)}, L^{(2)} \in \mathbb{N}$, let Φ^1 and Φ^2 be two* NNs *with depths $L^{(1)}$ and $L^{(2)}$, respectively, such that $N^{(2)}_{L^{(2)}} = N^{(1)}_0$, i.e., the output dimension of Φ^2 equals the input dimension of Φ^1. Then there exists an* NN $\Phi^1 \odot_{\varrho^2} \Phi^2$ *of depth $L := L^{(1)} + L^{(2)}$ to which we shall refer as sparse* ReLU^2*-based concatenation of Φ^1 and Φ^2 which has* ReLU^2 *activations, as well as those from Φ^1 and Φ^2. It satisfies*

$$\mathrm{R}(\Phi^1 \odot_{\varrho^2} \Phi^2) = \mathrm{R}(\Phi^1) \circ \mathrm{R}(\Phi^2), \quad L(\Phi^1 \odot_{\varrho^2} \Phi^2) = L^{(1)} + L^{(2)},$$
$$M(\Phi^1 \odot_{\varrho^2} \Phi^2) \leqslant M(\Phi^1) + 4M_{\mathrm{in}}(\Phi^1) + 7M_{\mathrm{out}}(\Phi^2) + M(\Phi^2) \leqslant 5M(\Phi^1) + 8M(\Phi^2).$$

Proposition 3.3 and 3.4 only apply to networks of equal depth. To parallelize two networks of unequal depth, the shallower of the two can be concatenated with a network that emulates the identity using Proposition 3.6.

Next, we recall the exact emulation of the identity and of products by ReLU^2 NNs.

Proposition 3.7 (ReLU^2 NN emulation of $\mathrm{Id}_{\mathbb{R}^d}$ [43], Proposition 2.3)**.** *For all $d \in \mathbb{N}$ and $L \in \mathbb{N}$, there exists a* ReLU^2 NN $\Phi^{\mathrm{Id},\varrho^2}_{d,L}$ *such that $\mathrm{R}(\Phi^{\mathrm{Id},\varrho^2}_{d,L})(x) = x$ for all $x \in \mathbb{R}^d$. Its* NN *depth and size satisfy $L(\Phi^{\mathrm{Id},\varrho^2}_{d,L}) = L$ and $M(\Phi^{\mathrm{Id},\varrho^2}_{d,L}) \leqslant CdL$, for $C > 0$ independent of d and L.*

Proposition 3.8 (ReLU^2 NN emulation of products)**.** *For all $d \in \mathbb{N}$, $d \geqslant 2$, there exists a* ReLU^2 NN Φ^{prod}_d *such that $\mathrm{R}(\Phi^{\mathrm{prod}}_d)(x_1, \ldots, x_d) = \prod_{j=1}^{d} x_j$ for all $x_1, \ldots, x_d \in \mathbb{R}$. Its* NN *depth and size satisfy $L(\Phi^{\mathrm{prod}}_d) \leqslant C\lceil \log_2(d) \rceil$ and $M(\Phi^{\mathrm{prod}}_d) \leqslant Cd$, for $C > 0$ independent of d.*

P r o o f. This proof is given in three steps. In Step 1, we construct Φ^{prod}_8 and analyze its depth and size. In Step 2, we construct and analyze Φ^{prod}_d for $d > 8$ satisfying $d \in 8^{\mathbb{N}}$ as an octree of Φ^{prod}_8 NNs. Here, $8^{\mathbb{N}} = \{8^k \colon k \in \mathbb{N}\}$ denotes the set of positive integer powers of 8. The reason for considering an octree of

Φ_8^{prod} NNs is that we need 8 to be bigger than the number 5 from the estimate $M(\Phi^1 \odot_{\varrho^2} \Phi^2) \leqslant 5M(\Phi^1) + 8M(\Phi^2)$ from Proposition 3.6. Finally, Step 3 considers $d \geqslant 2$ satisfying $d \notin 8^{\mathbb{N}}$.

Step 1. For the product of two numbers, Φ_2^{prod} is given in [32], Lemma 2.1. It satisfies $\mathrm{R}(\Phi_2^{\mathrm{prod}})(x,y) = xy$ for all $x, y \in \mathbb{R}$, $L(\Phi_2^{\mathrm{prod}}) = 2$, $N_0 = 2$, $N_1 = 4$, $N_2 = 1$ and $M(\Phi_2^{\mathrm{prod}}) = 12$. For $d = 8$, we define

$$\Phi_8^{\mathrm{prod}} := \Phi_2^{\mathrm{prod}} \odot_{\varrho^2} \mathrm{FP}(\Phi_2^{\mathrm{prod}}, \Phi_2^{\mathrm{prod}}) \odot_{\varrho^2} \mathrm{FP}(\Phi_2^{\mathrm{prod}}, \Phi_2^{\mathrm{prod}}, \Phi_2^{\mathrm{prod}}, \Phi_2^{\mathrm{prod}}).$$

By Proposition 3.6 it satisfies $L(\Phi_8^{\mathrm{prod}}) = 2 + 2 + 2 = 6$, and we denote its size by $M_8 := M(\Phi_8^{\mathrm{prod}})$. The network Φ_2^{prod} exactly emulates the product of two numbers, and no errors are incurred in the sparse concatenation $\odot_{\varrho^2}$, thus $\mathrm{R}(\Phi_8^{\mathrm{prod}})\colon \mathbb{R}^8 \to \mathbb{R}\colon (x_1, \ldots, x_8) \mapsto \prod_{i=1}^{8} x_i$.

Step 2. For $d > 8$ satisfying $d \in 8^{\mathbb{N}}$, we define

$$\Phi_d^{\mathrm{prod}} := \Phi_{d/8}^{\mathrm{prod}} \odot_{\varrho^2} \mathrm{FP}(\Phi_8^{\mathrm{prod}}, \ldots, \Phi_8^{\mathrm{prod}}),$$

where the full parallelization contains $d/8$ product networks. Based on Proposition 3.6 and the depth and size bounds for the $d = 8$ product network, we inductively obtain that for all $d \in 8^{\mathbb{N}}$ satisfying $d > 8$, $L(\Phi_d^{\mathrm{prod}}) = 2\lceil \log_2(d) \rceil$ and $M(\Phi_d^{\mathrm{prod}}) \leqslant 8dM_8$ holds:

$$\begin{aligned} L(\Phi_d^{\mathrm{prod}}) &= L(\Phi_{d/8}^{\mathrm{prod}}) + L(\Phi_8^{\mathrm{prod}}) = 2\lceil \log_2(d/8) \rceil + 6 = 2\lceil \log_2(d) \rceil, \\ M(\Phi_d^{\mathrm{prod}}) &\leqslant 5M(\Phi_{d/8}^{\mathrm{prod}}) + 8(d/8)M(\Phi_8^{\mathrm{prod}}) \leqslant 5 \cdot 8(d/8)M_8 + 8(d/8)M_8 \\ &= (5 \cdot 8 + 8)(d/8)M_8 \leqslant (8 \cdot 8)(d/8)M_8 = 8dM_8. \end{aligned}$$

The fact that $\mathrm{R}(\Phi_d^{\mathrm{prod}})\colon \mathbb{R}^d \to \mathbb{R}\colon (x_1, \ldots, x_d) \mapsto \prod_{i=1}^{d} x_i$ follows from the exactness of $\mathrm{R}(\Phi_{d/8}^{\mathrm{prod}})$ and $\mathrm{R}(\Phi_8^{\mathrm{prod}})$ and the fact that no error is incurred in the sparse concatenation $\odot_{\varrho^2}$.

Step 3. Finally, for $d \geqslant 2$ satisfying $d \notin 8^{\mathbb{N}}$, let $\tilde{d} = \min\{8^k \colon k \in \mathbb{N}, 8^k \geqslant d\} \in 8^{\mathbb{N}}$. By definition, it holds that $\tilde{d} \leqslant 8d$. We obtain Φ_d^{prod} from $\Phi_{\tilde{d}}^{\mathrm{prod}}$ by setting the last $\tilde{d} - d$ inputs of the NN to 1, through the biases in the first layer. This gives the desired realization and increases the network size by at most $\tilde{d} - d$. We thus obtain that

$$\begin{aligned} L(\Phi_d^{\mathrm{prod}}) &= L(\Phi_{\tilde{d}}^{\mathrm{prod}}) = 2\lceil \log_2(\tilde{d}) \rceil \leqslant 2\lceil \log_2(d) \rceil + 6 \leqslant C\lceil \log_2(d) \rceil, \\ M(\Phi_d^{\mathrm{prod}}) &\leqslant M(\Phi_{\tilde{d}}^{\mathrm{prod}}) + (\tilde{d} - d) \leqslant 8\tilde{d}M_8 + \tilde{d} \leqslant Cd. \end{aligned}$$

□

Finally, we recall the exact emulation of polynomials by ReLU^2 networks. We need the result from [43], which holds for multivariate polynomials, only for the special case of univariate polynomials.

Proposition 3.9 (ReLU^2 emulation of univariate polynomials [43], Proposition 2.14)**.** *For all $p \in \mathbb{N}$ and $w \in \mathbb{P}_p$ there exists an* ReLU^2 NN Φ^w *such that* $\mathrm{R}(\Phi^w) = w$, $L(\Phi^w) \leqslant C \log_2(p+1)$ *and* $M(\Phi^w) \leqslant Cp$, *for $C > 0$ independent of p and w.*

3.2. NN emulations of continuous, piecewise linear functions on simplicial meshes. We recall from [33] *exact NN emulations* of CPwL functions on regular, simplicial partitions $\mathcal{M}$ of polytopal domains $\mathrm{D} \subset \mathbb{R}^d$.

We have by [33], Section 5 a vector space of NNs,

$$\mathcal{NN}(CPwL; \mathcal{M}, \mathrm{D}) = \{\Phi^{CPwL,v} \colon\ v \in \mathrm{S}_1(\mathcal{M}, \mathrm{D})\}$$

such that the realization of each NN $\Phi^{CPwL,v}$ equals v everywhere in $\overline{\mathrm{D}}$.

Proposition 3.10 ([33], Proposition 5.7)**.** *Let $\mathrm{D} \subset \mathbb{R}^d$, $d \geqslant 2$, be a bounded, polytopal domain. For every regular, simplicial triangulation $\mathcal{M}$ of* D *there exists an* NN $\Phi^{CPwL} := \Phi^{CPwL(\mathcal{M},\mathrm{D})}$ *with only* ReLU *activations, which in parallel emulates the shape functions $\{\theta_i^{\mathrm{S}_1}\}_{i\in\mathcal{V}}$, which are defined by $\theta_i^{\mathrm{S}_1}(i) = 1$ and $\theta_i^{\mathrm{S}_1}(j) = 0$ for all other $j \in \mathcal{V}$. That is,* $\mathrm{R}(\Phi^{CPwL})\colon\ \mathrm{D} \to \mathbb{R}^{|\mathcal{V}|}$ *satisfies*

$$\mathrm{R}(\Phi^{CPwL})_i(x) = \theta_i^{\mathrm{S}_1}(x) \quad \forall\, x \in \mathrm{D} \text{ and all } i \in \mathcal{V}.$$

There exists $C > 0$ independent of d and $\mathcal{M}$ such that

$$\begin{aligned} L(\Phi^{CPwL}) &\leqslant 8 + \log_2(\mathfrak{s}(\mathcal{V})) + \log_2(d+1), \\ M(\Phi^{CPwL}) &\leqslant C|\mathcal{V}| \log_2(\mathfrak{s}(\mathcal{V})) + Cd^2 \sum_{i\in\mathcal{V}} s(i) \leqslant Cd^2 \mathfrak{s}(\mathcal{V}) \dim(\mathrm{S}_1(\mathcal{M}, \mathrm{D})). \end{aligned}$$

For all $v = \sum_{i\in\mathcal{V}} v_i \theta_i^{\mathrm{S}_1} \in \mathrm{S}_1(\mathcal{M}, \mathrm{D})$, there exists an NN $\Phi^{CPwL,v} := \Phi^{CPwL(\mathcal{M},\mathrm{D}),v}$ with only ReLU *activations, such that for a constant $C > 0$ independent of d and $\mathcal{M}$,*

$$\begin{aligned} \mathrm{R}(\Phi^{CPwL,v})(x) &= v(x) \quad \forall\, x \in \mathrm{D}, \\ L(\Phi^{CPwL,v}) &\leqslant 8 + \log_2(\mathfrak{s}(\mathcal{V})) + \log_2(d+1), \\ M(\Phi^{CPwL,v}) &\leqslant C|\mathcal{V}| \log_2(\mathfrak{s}(\mathcal{V})) + Cd^2 \sum_{i\in\mathcal{V}} s(i) \leqslant Cd^2 \mathfrak{s}(\mathcal{V}) \dim(\mathrm{S}_1(\mathcal{M}, \mathrm{D})). \end{aligned}$$

The layer dimensions and the lists of activation functions of Φ^{CPwL} *and* $\Phi^{CPwL,v}$ *are independent of* v *and only depend on* $\mathcal{M}$ *through*

$$\{s(i)\}_{i\in\mathcal{V}} \quad \textit{and} \quad |\mathcal{V}| = \dim(\mathrm{S}_1(\mathcal{M},\mathrm{D})).$$

The set $\mathcal{NN}(CPwL;\mathcal{M},\mathrm{D}) := \{\Phi^{CPwL,v}\colon\ v\in\mathrm{S}_1(\mathcal{M},\mathrm{D})\}$ *together with the linear operation* $\Phi^{CPwL,v}\widehat{+}\lambda\Phi^{CPwL,w} := \Phi^{CPwL,v+\lambda w}$ *for all* $v,w\in\mathrm{S}_1(\mathcal{M},\mathrm{D})$ *and all* $\lambda\in\mathbb{R}$ *is a vector space. The realization map* $\mathrm{R}(\cdot)\colon\ \mathcal{NN}(CPwL;\mathcal{M},\mathrm{D})\to\mathrm{S}_1(\mathcal{M},\mathrm{D})$ *is a linear isomorphism.*

For a proof, we refer to [33], Section 5.

R e m a r k 3.11 ([33], Remark 5.2). It was shown in [33], Remark 5.2 that $|\mathcal{V}|\leqslant\sum\limits_{i\in\mathcal{V}} s(i)\leqslant c(\mathcal{V},d)|\mathcal{M}|$, where $c(\mathcal{V},d)=d+1$ is the number of vertices of a d-simplex. We obtain this inequality by observing that $s(i)\geqslant 1$ and that each element $K\in\mathcal{M}$ contributes $+1$ to $c(\mathcal{V},d)$ terms $s(i)$. Therefore, we also have the bound $M(\Phi^{CPwL})\leqslant C|\mathcal{V}|\log_2(\mathfrak{s}(\mathcal{V}))+Cd^2c(\mathcal{V},d)|\mathcal{M}|\leqslant Cd^2c(\mathcal{V},d)\log_2(\mathfrak{s}(\mathcal{V}))|\mathcal{M}|$. The same bound holds for $M(\Phi^{CPwL,v})$.

4. NN emulation of high-order, Lagrangian finite element spaces

We state and prove the main result of this paper: a *mathematically exact emulation of Lagrangian FE spaces of any order* $p\geqslant 1$ on general regular, simplicial triangulations in a polytopal domain $\mathrm{D}\subset\mathbb{R}^d$, in space dimension $d\geqslant 2$, with ReLU and ReLU^2 activations, and with both the number of neurons and the DNN size (i.e., the number of nonzero NN parameters, called weights and biases) in the NN required to realize the emulation being bounded by a constant (which depends on the domain D and on the mesh connectivity) times the dimension of the corresponding FE space. This result comprises in particular the *hp*-finite element spaces $S^p(\mathcal{M}^{(l)})$ on families $\mathfrak{M}_{\kappa,\sigma}(\mathcal{S})$ of nested, regular simplicial partitions $\mathcal{M}^{(l)}$ in D which are σ-geometrically refined towards the set $\mathcal{S}$ of singular support points and which are κ-shape regular. For these DNNs, the exponential approximation rate result (2.14) in Proposition 2.2 holds. This implies exponential rates for the DNN approximation error for various DNN-based PDE approximation methods.

4.1. Existing results. The exact ReLU NN emulation of continuous, piecewise linear functions on general regular, simplicial partitions of polytopal domains $\mathrm{D}\subset\mathbb{R}^d$, $d\in\mathbb{N}$, was achieved in [33], Proposition 5.7, as stated above in Proposition 3.10. In [27], it was observed that on each simplex $K\in\mathcal{M}$, each of the

barycentric coordinates equals one of the "hat" basis functions $\theta_q^{\mathrm{S}_1} \in \mathrm{S}_1(\mathcal{M}, \mathrm{D})$ for a vertex q of K. For $q \in \mathcal{V}$, these are defined by $\theta_q^{\mathrm{S}_1}(q) = 1$ and $\theta_q^{\mathrm{S}_1}(q') = 0$ for all other vertices $q' \in \mathcal{V}$. Based on an expression of the local polynomial space $\mathbb{P}_p(K)$ for $p \in \mathbb{N}$ in terms of barycentric coordinates, an explicit formula in terms of these hat functions for a global basis of high-order finite elements was given in [27]. We recall it in Proposition 4.1 below. To state the result, on each subsimplex K' of $\mathcal{M}$ of dimension $m \in \{0, \dots, d\}$ with vertices $a_0, \dots, a_m$, for all $p \in \mathbb{N}$ we define the set of (equispaced in simplex K') interpolation points

$$\mathcal{N}_p(K') := \Big\{ \sum_{k=0}^{m} \frac{\alpha_k a_k}{p} \colon \alpha_k \in \mathbb{N}_0 \text{ and } \sum_{k=0}^{m} \alpha_k = p \Big\}.$$

As nodal basis for $\mathbb{P}_p(K')$ we consider the Lagrange polynomials $\{v_i\}_{i \in \mathcal{N}_p(K')}$ defined by $v_i(i) = 1$ and $v_i(j) = 0$ for all other $j \in \mathcal{N}_p(K')$. We denote the set of all interpolation points by $\mathcal{N} := \mathcal{N}_p(\mathcal{M}) := \bigcup_{K \in \mathcal{M}} \mathcal{N}_p(K)$ and consider the global Lagrangian basis functions $\{\theta_i^{\mathrm{S}_p}\}_{i \in \mathcal{N}}$ of $\mathrm{S}_p(\mathcal{M}, \mathrm{D})$ defined by $\theta_i^{\mathrm{S}_p}(i) = 1$ and $\theta_i^{\mathrm{S}_p}(j) = 0$ for all other $j \in \mathcal{N}$.

Proposition 4.1 ([27], Theorem 3.2)**.** *For all $p \in \mathbb{N}$ and a polytopal domain $\mathrm{D} \subset \mathbb{R}^d$, for $d \in \mathbb{N}$, let $\mathcal{M}$ be a regular, simplicial partition of D. For each interpolation node $i \in \mathcal{N}$, if i is a vertex of $\mathcal{M}$, let $K' = i$ and $m = 0$, and else, let K' be the subsimplex of $\mathcal{M}$ satisfying $i \in \operatorname{int} K'$ and let m denote the dimension of K'. Denoting the vertices of K' by $a_0, \dots, a_m$, let $\alpha_0, \dots, \alpha_m \in \mathbb{N}_0$ be such that $i = \sum_{k=0}^{m} \alpha_k a_k / p$. Then*

$$\theta_i^{\mathrm{S}_p}(x) = \prod_{k=0}^{m} \tfrac{1}{\alpha_k!} \prod_{j=0}^{\alpha_k - 1} (p\theta_{a_k}^{\mathrm{S}_1}(x) - j) = \prod_{k=0}^{m} \prod_{j=0}^{\alpha_k - 1} \Big(\frac{p}{j+1} \theta_{a_k}^{\mathrm{S}_1}(x) - \frac{j}{j+1} \Big). \tag{4.1}$$

From this, the following ReLU NN emulation rate result is obtained in [27].

Proposition 4.2 ([27], Theorem 4.2)**.** *For all $p \in \mathbb{N}$ and a polytopal domain $\mathrm{D} \subset \mathbb{R}^d$, for $d \in \mathbb{N}$, let $\mathcal{M}$ be a regular, simplicial partition of D.*

For all $v \in \mathrm{S}_p(\mathcal{M}, \mathrm{D})$, there exists an NN *$\Phi_v$ with* ReLU *and* ReLU^2 *activation, such that $\mathrm{R}(\Phi_v) = v$. The depth $L = L(\Phi_v)$ and the numbers of neurons $N_1, \dots, N_{L-1}$ in the hidden layers satisfy for some $C > 0$ independent of $\mathcal{M}$*

$$\begin{aligned} L(\Phi_v) &\leqslant (\lceil \log_2(\mathfrak{s}(\mathcal{V})) \rceil + \lceil \log_2(d+1) \rceil + \lceil \log_2(p) \rceil + 7)\, |\mathcal{N}|, \\ \max_{l=1}^{L-1} N_l &\leqslant C \max\{p, (d+1) \min\{d+1, p+1\} \mathfrak{s}(\mathcal{V})\}. \end{aligned}$$

R e m a r k 4.3. The proof of [27], Theorem 4.2 is based on [27], Lemma 4.2, in which single Lagrangian basis functions are emulated. For each $p \in \mathbb{N}$ and each interpolation point $i \in \mathcal{N}$, let K', m, $a_0, \ldots, a_m$ and $\alpha_0, \ldots, \alpha_m$ be as in Proposition 4.1.

The first part of the network in parallel emulates the CPwL "hat" functions $\{\theta^{\mathrm{S}_1}_{a_k}\}_{k=0}^m$ by subnetworks of depth $L_k \leqslant \lceil \log_2(s(a_k)) \rceil + \lceil \log_2(d+1) \rceil + 7$ and numbers of neurons per layer $N_j \leqslant C(d+1)s(a_k)2^{-j}$ for $j = 1, \ldots, L_k$ for a constant $C > 0$ independent of $\mathcal{M}$.[4] The parallelization of these $m+1$ networks has depth $L \leqslant \max\limits_{k=0}^{m} \lceil \log_2(s(a_k)) \rceil + \lceil \log_2(d+1) \rceil + 7$ and numbers of neurons per layer $N_j \leqslant C(d+1)2^{-j} \sum\limits_{k=0}^{m} s(a_k)$ for $j = 1, \ldots, L$.

The second part of the network emulates the product of p factors stated in (4.1). In [27], Lemma 4.2, this is realized with $\lceil \log_2(p) \rceil$ layers. Using a binary tree of product subnetworks, the numbers N_j of neurons per layer of this binary tree are bounded as[5]

$$N_j \leqslant Cp2^{-j} \quad \text{for} \quad j = 1, \ldots, \lceil \log_2(p) \rceil.$$

The concatenation of both parts has depth at most

$$\max_{k=0}^{m} \lceil \log_2(s(a_k)) \rceil + \lceil \log_2(d+1) \rceil + \lceil \log_2(p) \rceil + 7$$

and numbers of neurons bounded by

$$N_j \leqslant C(d+1)2^{-j} \sum_{k=0}^{m} s(a_k) \quad \text{for} \quad j = 1, \ldots, \max_{k=0}^{m} \lceil \log_2(s(a_k)) \rceil + \lceil \log_2(d+1) \rceil + 7$$

and for the last $\lceil \log_2(p) \rceil$ layers

$$N_{L - \lceil \log_2(p) \rceil + j} \leqslant Cp2^{-j} \quad \text{for} \quad j = 1, \ldots, \lceil \log_2(p) \rceil,$$

i.e. $\max\limits_{j=1}^{L} N_j \leqslant C \max\Big\{p, (d+1) \sum\limits_{k=0}^{m} s(a_k)\Big\}$. For $i \in \mathcal{V}$ it holds that $m = 0 \leqslant p$, and for interpolation points that are not vertices, by definition of K' it holds that $i \in \operatorname{int} K'$, thus, $i = \sum\limits_{k=0}^{m} \alpha_k a_k / p$ for $\alpha_k \in \mathbb{N}$, i.e., none of the α_k vanishes. From $\sum\limits_{k=0}^{m} \alpha_k = p$ it thus follows that $m \leqslant p$. By definition, it also holds that $m \leqslant d$, thus, we can estimate $\sum\limits_{k=0}^{m} s(a_k) \leqslant (m+1)\mathfrak{s}(\mathcal{V}) \leqslant \min\{d+1, p+1\}\mathfrak{s}(\mathcal{V})$.

[4] The value $C = 2^3$ stated in [27], Lemma 4.2 gives $N_{L_k} < 1$, which appears inconsistent.

[5] In [27], Lemma 4.2, the formula for the numbers of neurons per layer is incorrect, as it does not take into account the numbers of neurons in these last $\lceil \log_2(p) \rceil$ layers. In fact, for those layers, [27], (4.11) states that $N_j < 1$.

To obtain an approximation of $v \in \mathrm{S}_p(\mathcal{M}, \mathrm{D})$, it remains to take the linear combination of the Lagrangian basis functions for all interpolation points $i \in \mathcal{N}$. In [27], Theorem 4.2, this is done by placing these subnetworks in subsequent groups of layers, increasing the depth and not the width. In each layer, the numbers of neurons per layer is increased by $2d+1$ in order to forward the inputs to the hidden layers and to keep track of the partial sums of the outputs of the subnetworks, see [27], Properties 4.2. This gives the bounds on the width and depth in Proposition 4.2.

R e m a r k 4.4. For fixed $\mathcal{M}$ and $p \to \infty$, it holds that $|\mathcal{N}| \simeq p^d$, thus the total number of neurons is $\sum_{j=0}^{L} N_j = O(p^{d+1}\log(p))$, which is larger than the number of degrees of freedom, which equals $|\mathcal{N}| \simeq p^d$. The network size may be even larger, as w.l.o.g. it is bounded from below by $\sum_{j=1}^{L} N_j$, see Remark 3.2.

4.2. DNN emulation of hp-FE spaces on regular triangulations. The NN emulation results in [27] focused on the NN size bounds for Lagrangian finite elements with nodal bases, at fixed, uniform, polynomial degree $p \geqslant 1$. Here, we focus on NN size bounds which are explicit in terms of $p \geqslant 1$ and tighter than those in [27]. This is achieved by a suitably modified NN architecture for the numerical realization of the shape functions.

Specifically, the emulation of all shape functions $\{\theta_i^{\mathrm{S}_p}\}_{i\in\mathcal{N}}$ can be obtained more efficiently when we rearrange the part of the network that computes the products in (4.1). Rather than computing for each Lagrangian basis function the product of p factors from (4.1), we first compute for all nodes $q \in \mathcal{V}$ the real-valued quantities

$$w_\alpha(\theta_q^{\mathrm{S}_1}(x)) := \prod_{j=0}^{\alpha-1} \Big(\frac{p}{j+1}\theta_q^{\mathrm{S}_1}(x) - \tfrac{j}{j+1} \Big) \quad \text{for } \alpha = 1, \dots, p, \tag{4.2}$$

where $w_\alpha \in \mathbb{P}_\alpha \subset \mathbb{P}_p$. For all $i \in \mathcal{N}$, equation (4.1) can then be rewritten as

$$\theta_i^{\mathrm{S}_p}(x) = \prod_{k=0}^{m} w_{\alpha_m}(\theta_{a_m}^{\mathrm{S}_1}(x)). \tag{4.3}$$

R e m a r k 4.5. For all $i \in \mathcal{N}$, the number of factors in (4.3) is $m+1 \leqslant d+1$, one for each of the vertices $a_0, \dots, a_m$. This is independent of p and leads to a network size of the order $O(p^d)$, rather than a number of neurons of the order $O(p^{d+1}\log(p))$ in [27], Theorem 4.2 (see Remark 4.4) and an even larger network size.

Proposition 4.6. *For all $p \in \mathbb{N}$ and a polytopal domain $\mathrm{D} \subset \mathbb{R}^d$, for $2 \leqslant d \in \mathbb{N}$, let $\mathcal{M}$ be a regular, simplicial partition of* D.

There exists an NN $\Phi^{CPwP_p} := \Phi^{CPwP_p(\mathcal{M},\mathrm{D})}$ *with only* ReLU *and* ReLU^2 *activation, which in parallel emulates the shape functions* $\{\theta_i^{\mathrm{S}_p}\}_{i\in\mathcal{N}}$, *i.e.,* $\mathrm{R}(\Phi^{CPwP_p})\colon \mathrm{D} \to \mathbb{R}^{|\mathcal{N}|}$ *satisfies*

$$\mathrm{R}(\Phi^{CPwP_p})_i(x) = \theta_i^{\mathrm{S}_p}(x) \quad \forall\, i \in \mathcal{N} \text{ and } x \in \mathrm{D}.$$

The network depth and size satisfy for some $C > 0$ *independent of* $\mathcal{M}$, d *and* p

$$\begin{aligned} L(\Phi^{CPwP_p}) &\leqslant C(\log_2(p+1) + \log_2(d+1) + \log_2(\mathfrak{s}(\mathcal{V})) + 1), \\ M(\Phi^{CPwP_p}) &\leqslant C\Big(d|\mathcal{N}| + p^2|\mathcal{V}| + |\mathcal{V}|\log_2(\mathfrak{s}(\mathcal{V})) + d^2\sum_{q\in\mathcal{V}} s(q)\Big) \\ &\leqslant C\big(d|\mathcal{N}| + p^2|\mathcal{V}| + d^2\mathfrak{s}(\mathcal{V})|\mathcal{V}|\big). \end{aligned}$$

For all $v = \sum_{i\in\mathcal{N}} v_i\theta_i^{\mathrm{S}_p} \in \mathrm{S}_p(\mathcal{M},\mathrm{D})$, *there exists an NN* $\Phi^{CPwP_p,v} := \Phi^{CPwP_p(\mathcal{M},\mathrm{D}),v}$ *with only* ReLU *and* ReLU^2 *activations, such that for a constant* $C > 0$ *independent of* $\mathcal{M}$, d *and* p

$$\begin{aligned} \mathrm{R}(\Phi^{CPwP_p,v})(x) &= v(x) \quad \forall\, x \in \mathrm{D}, \\ L(\Phi^{CPwP_p,v}) &\leqslant C(\log_2(p+1) + \log_2(d+1) + \log_2(\mathfrak{s}(\mathcal{V})) + 1), \\ M(\Phi^{CPwP_p,v}) &\leqslant C(d|\mathcal{N}| + p^2|\mathcal{V}| + |\mathcal{V}|\log_2(\mathfrak{s}(\mathcal{V})) + d^2\sum_{q\in\mathcal{V}} s(q)) \\ &\leqslant C(d|\mathcal{N}| + p^2|\mathcal{V}| + d^2\mathfrak{s}(\mathcal{V})|\mathcal{V}|). \end{aligned}$$

The layer dimensions and the lists of activation functions of Φ^{CPwP_p} *and* $\Phi^{CPwP_p,v}$ *are independent of* v *and only depend on* p, *and on* $\mathcal{M}$ *through* $\{m(i)\}_{i\in\mathcal{N}}$, $|\mathcal{N}|$, $\{s(q)\}_{q\in\mathcal{V}}$ *and* $|\mathcal{V}|$. *In the first layers, only* ReLU *activation is applied. The number of these* ReLU *layers is at most* $7 + \log_2(\mathfrak{s}(\mathcal{V})) + \log_2(d+1)$. *The remaining hidden layers are strict* ReLU^2 *layers, i.e., only* ReLU^2 *activation is applied. Their number is bounded by* $C(\log_2(d+1) + \log_2(p+1))$ *for an absolute constant* $C > 0$.

The set $\mathcal{NN}(CPwP_p;\mathcal{M},\mathrm{D}) := \{\Phi^{CPwP_p,v}\colon v \in \mathrm{S}_p(\mathcal{M},\mathrm{D})\}$ *together with the linear operation* $\Phi^{CPwP_p,v} \widehat{+} \lambda\Phi^{CPwP_p,w} := \Phi^{CPwP_p,v+\lambda w}$ *for all* $v, w \in \mathrm{S}_p(\mathcal{M},\mathrm{D})$ *and all* $\lambda \in \mathbb{R}$ *is a vector space. The realization map* $\mathrm{R}(\cdot)\colon \mathcal{NN}(CPwP_p;\mathcal{M},\mathrm{D}) \to \mathrm{S}_p(\mathcal{M},\mathrm{D})$ *is a linear isomorphism.*

P r o o f. For each $p \in \mathbb{N}$ and each interpolation point $i \in \mathcal{N}$, let m, $a_0, \ldots, a_m$ and $\alpha_0, \ldots, \alpha_m$ be as in Proposition 4.1. In the remainder of this proof, we will denote them by $m(i)$, $a_0(i), \ldots, a_{m(i)}(i)$ and $\alpha_0(i), \ldots, \alpha_{m(i)}(i)$.

This proof consists of three steps. In Step 1, we construct Φ^{CPwP_p} and prove the formula for its realization. In Step 2, we prove the bounds on the network depth and size of Φ^{CPwP_p}. The remainder of the statements is proved in Step 3.

Step 1. First, we recall several NNs which we will use in our construction. From Proposition 3.10 we recall $\Phi^{CPwL} := \Phi^{CPwL(\mathcal{M},\mathrm{D})}$, whose depth and size are bounded by

$$\begin{aligned} L(\Phi^{CPwL}) &\leqslant 8 + \log_2(\mathfrak{s}(\mathcal{V})) + \log_2(d+1), \\ M(\Phi^{CPwL}) &\leqslant |\mathcal{V}| \log_2(\mathfrak{s}(\mathcal{V})) + d^2 \sum_{q\in\mathcal{V}} s(q). \end{aligned}$$

We consider NN emulations of the polynomials $\{w_\alpha\}_{\alpha=1}^p \in \mathbb{P}_p$ defined in (4.2). By sparsely concatenating these networks $\{\Phi^{w_\alpha}\}_{\alpha=1}^p$ with a ReLU^2 identity network from Proposition 3.7 using Proposition 3.6, we obtain that for all $p \in \mathbb{N}$ and $\alpha = 1, \ldots, p$, there is a ReLU^2 NN $\Phi_p^{w_\alpha}$ such that $\mathrm{R}(\Phi_p^{w_\alpha}) = w$, $L(\Phi_p^{w_\alpha}) = c\lceil \log_2(p+1)\rceil$ and $M(\Phi_p^{w_\alpha}) \leqslant Cp$, for constants $c \in \mathbb{N}$ and $C > 0$ independent of p and $\alpha = 1, \ldots, p$. We now prove this.

Let $c \in \mathbb{N}$ be such that $L(\Phi^{w_\alpha}) < c\lceil \log_2(p+1)\rceil$ for all $\alpha = 1, \ldots, p$. With $L_\alpha := c\lceil \log_2(p+1)\rceil - L(\Phi^{w_\alpha}) \leqslant c\lceil \log_2(p+1)\rceil$ we then define $\Phi_p^{w_\alpha} := \Phi_{1,L_\alpha}^{\mathrm{Id},\varrho^2} \odot_{\varrho^2} \Phi^{w_\alpha}$ for all $\alpha = 1, \ldots, p$. We obtain that $\mathrm{R}(\Phi_p^{w_\alpha}) = \mathrm{R}(\Phi^{w_\alpha}) = w_\alpha$, $L(\Phi_p^{w_\alpha}) = L_\alpha + L(\Phi^{w_\alpha}) = c\lceil \log_2(p+1)\rceil$ and $M(\Phi_p^{w_\alpha}) \leqslant CM(\Phi_{1,L_\alpha}^{\mathrm{Id},\varrho^2}) + CM(\Phi^{w_\alpha}) \leqslant CL_\alpha + C(\alpha+1) \leqslant Cp$, for a constant $C > 0$ independent of α and p.

Analogously, we obtain from Proposition 3.8 the following result. For all $d \in \mathbb{N}$, there exist ReLU^2 NNs $\{\Phi_{l,d+1}^{\mathrm{prod}}\}_{l=1}^{d+1}$ such that $\mathrm{R}(\Phi_{l,d+1}^{\mathrm{prod}})(x_1, \ldots, x_l) = \prod_{j=1}^{l} x_j$ for all $x_1, \ldots, x_l \in \mathbb{R}$, and such that their NN depths and sizes satisfy $L(\Phi_{l,d+1}^{\mathrm{prod}}) = c\lceil \log_2(d+1)\rceil$ and $M(\Phi_{l,d+1}^{\mathrm{prod}}) \leqslant Cd$, for constants $c \in \mathbb{N}$ and $C > 0$ independent of d.

We can now define

$$\begin{aligned} \Phi^{CPwP_p} &:= \Phi_{(1)}^{CPwP_p} \odot_{\varrho^2} \Phi_{(2)}^{CPwP_p} \odot_{\varrho^2} \Phi_{(3)}^{CPwP_p} \odot_{\varrho^2} \Phi_{(4)}^{CPwP_p} \\ &:= \mathrm{FP}(\{\Phi_{m(i)+1,d+1}^{\mathrm{prod}}\}_{i\in\mathcal{N}}) \odot_{\varrho^2} \Phi_{(2)}^{CPwP_p} \\ &\quad \odot_{\varrho^2} \mathrm{FP}(\{\mathrm{P}(\{\Phi_p^{w_\alpha}\}_{\alpha=1}^p)\}_{q\in\mathcal{V}}) \odot_{\varrho^2} \Phi^{CPwL}, \end{aligned}$$

where $\Phi_{(2)}^{CPwP_p}$ is an NN of depth 1 emulating a linear transformation $x \mapsto Ax$. Fixing a bijection $n\colon \mathcal{N} \to \{1, \ldots, |\mathcal{N}|\}$, the weight matrix A of size $\left(\sum_{i\in\mathcal{N}} (m(i)+1)\right) \times p|\mathcal{V}|$ is defined by

$$A_{j_1,j_2} = \begin{cases} 1 & \text{if } j_1 = k+1+\sum_{n'=1}^{n(i)-1} (m(n^{-1}(n'))+1) \text{ for some } k \in \{0, \ldots, m(i)\} \\ & \text{and } w_{\alpha_k(i)}(\theta_{a_k(i)}^{\mathrm{S}_1}(x)) \text{ is the } j_2\text{'th component of} \\ & \mathrm{R}(\Phi_{(3)}^{CPwP_p} \odot_{\varrho^2} \Phi_{(4)}^{CPwP_p}), \\ 0 & \text{else.} \end{cases}$$

Note that each row has precisely one nonzero element. The number of rows is bounded by $(d+1)|\mathcal{N}|$, hence $M(\Phi_{(2)}^{CPwP_p}) \leqslant (d+1)|\mathcal{N}|$. It follows directly from these definitions that for all $x \in \mathbb{R}^d$

$$\begin{aligned}\mathrm{R}(\Phi_{(4)}^{CPwP_p})(x) &= \{\theta_q^{\mathrm{S}_1}(x)\}_{q\in\mathcal{V}},\\ \mathrm{R}(\Phi_{(3)}^{CPwP_p} \odot_{\varrho^2} \Phi_{(4)}^{CPwP_p})(x) &= \{\{w_\alpha(\theta_q^{\mathrm{S}_1}(x))\}_{\alpha=1}^{p}\}_{q\in\mathcal{V}},\end{aligned}$$

and

$$\begin{aligned}\mathrm{R}(\Phi_{(1)}^{CPwP_p} \odot_{\varrho^2} \Phi_{(2)}^{CPwP_p} \odot_{\varrho^2} \Phi_{(3)}^{CPwP_p} \odot_{\varrho^2} \Phi_{(4)}^{CPwP_p})_i(x)\\ = \prod_{k=0}^{m(i)} w_{\alpha_k(i)}(\theta_{a_k(i)}^{\mathrm{S}_1}(x)) = \theta_i^{\mathrm{S}_p}(x).\end{aligned}$$

Step 2. To obtain an *estimate on the network depth*, we combine the bounds from Step 1 on the depths of the subnetworks using Propositions 3.4 and 3.6, and get

$$\begin{aligned}L(\Phi^{CPwP_p}) &\leqslant L(\Phi_{(1)}^{CPwP_p}) + L(\Phi_{(2)}^{CPwP_p}) + L(\Phi_{(3)}^{CPwP_p}) + L(\Phi_{(4)}^{CPwP_p})\\ &\leqslant c\lceil\log_2(d+1)\rceil + 1 + C\lceil\log_2(p+1)\rceil + (8 + \log_2(\mathfrak{s}(\mathcal{V})) + \log_2(d+1))\\ &\leqslant C(1 + \log_2(\mathfrak{s}(\mathcal{V})) + \log_2(d+1) + \log_2(p+1)).\end{aligned}$$

Similarly, using the same propositions to combine bounds on the sizes of the subnetworks, we obtain as *bound on the network size*

$$\begin{aligned}M(\Phi^{CPwP_p}) &\leqslant CM(\Phi_{(1)}^{CPwP_p}) + CM(\Phi_{(2)}^{CPwP_p}) + CM(\Phi_{(3)}^{CPwP_p}) + CM(\Phi_{(4)}^{CPwP_p})\\ &\leqslant C(d+1)|\mathcal{N}| + C(d+1)|\mathcal{N}| + Cp^2|\mathcal{V}|\\ &\quad + C\bigg(|\mathcal{V}|\log_2(\mathfrak{s}(\mathcal{V})) + d^2\sum_{q\in\mathcal{V}} s(q)\bigg)\\ &\leqslant C\bigg(d|\mathcal{N}| + p^2|\mathcal{V}| + |\mathcal{V}|\log_2(\mathfrak{s}(\mathcal{V})) + d^2\sum_{q\in\mathcal{V}} s(q)\bigg)\\ &\leqslant C(d|\mathcal{N}| + p^2|\mathcal{V}| + d^2\mathfrak{s}(\mathcal{V})|\mathcal{V}|).\end{aligned}$$

The subnetwork $\Phi_{(4)}^{CPwP_p} = \Phi^{CPwL}$ only has ReLU activations and comprises at most $7 + \log_2(\mathfrak{s}(\mathcal{V})) + \log_2(d+1)$ hidden layers, which is one less than the total number of layers. The subnetworks $\Phi_{(1)}^{CPwP_p}$, $\Phi_{(2)}^{CPwP_p}$, $\Phi_{(3)}^{CPwP_p}$ and the sparse concatenations have only ReLU^2 activations. The number of such hidden layers is at most

$$L(\Phi_{(1)}^{CPwP_p}) + L(\Phi_{(2)}^{CPwP_p}) + L(\Phi_{(3)}^{CPwP_p}) \leqslant C(\log_2(d+1) + \log_2(p+1)).$$

Step 3. For all $v \in \mathrm{S}_p(\mathcal{M}, \mathrm{D})$, the NN $\Phi^{CPwP_p,v}$ is defined as $(A', 0, \mathrm{Id}_{\mathbb{R}}) \bullet \Phi^{CPwP_p}$, where $A' \in \mathbb{R}^{1\times|\mathcal{N}|}$ is the row vector containing the weights $A'_{1,i} = v_i$ for all $i \in \mathcal{N}$. It holds that $L(\Phi^{CPwP_p,v}) = 1 - 1 + L(\Phi^{CPwP_p}) = L(\Phi^{CPwP_p})$.

To estimate its size, we observe that the hidden layers of Φ^{CPwP_p} and $\Phi^{CPwP_p,v}$ coincide, as these layers emulate the FE basis functions. The layer dimensions and the lists of activation functions of $\Phi^{CPwP_p}_{(4)}$ depend on $\mathcal{M}$ only through $\{s(q)\}_{q\in\mathcal{V}}$ and $|\mathcal{V}|$. Those of $\Phi^{CPwP_p}_{(3)}$ only depend on p and $|\mathcal{V}|$, and those of $\Phi^{CPwP_p}_{(1)}$ and $\Phi^{CPwP_p}_{(2)}$ only on p, $|\mathcal{V}|$, $|\mathcal{N}|$ and $\{m(i)\}_{i\in\mathcal{N}}$. Furthermore, each weight in the output layer of $\Phi^{CPwP_p,v}$ is the inner product of the row vector A' with a column of the output layer weight matrix of Φ^{CPwP_p}, hence the number of nonzero weights of $\Phi^{CPwP_p,v}$ is at most that of Φ^{CPwP_p}. The same holds for the number of nonzero biases, as the bias of $\Phi^{CPwP_p,v}$ is the inner product of A' with the bias vector of Φ^{CPwP_p}. Thus, it follows that

$$M(\Phi^{CPwP_p,v}) \leqslant M(\Phi^{CPwP_p}).$$

By definition of $\Phi^{CPwP_p,v}$, the realization $\mathrm{R}(\Phi^{CPwP_p,v})$ is a linear combination of the outputs of $\mathrm{R}(\Phi^{CPwP_p})$, which is determined uniquely by the weights in the output layer, $(v_i)_{i\in\mathcal{N}}$ (which coincide, due to (4.1), with the nodal values of v). Therefore, $\mathrm{R}(\cdot)\colon \mathcal{NN}(CPwP_p; \mathcal{M}, \mathrm{D}) \to \mathrm{S}_p(\mathcal{M}, \mathrm{D})$ is a bijection. With the linear operations defined in the proposition, this map is linear by definition, thus a linear isomorphism. □

R e m a r k 4.7. Using that $|\mathcal{V}| \leqslant \sum_{q\in\mathcal{V}} s(q) \leqslant (d+1)|\mathcal{M}|$ by Remark 3.11 and that $|\mathcal{N}| \simeq p^d|\mathcal{M}|$ (with the proportionality constant hidden in $\simeq$ depending on d), we obtain that the network size is bounded by $C(d)(\log_2(\mathfrak{s}(\mathcal{V})) + p^d)|\mathcal{M}|$.

For a fixed triangulation $\mathcal{M}$, for increasing polynomial degree this scales as $O(p^d)$, which is of optimal order and improves the complexity of the networks from [27], Theorem 4.2, cf. Remark 4.5 above.

When we do not consider a single, fixed partition, but rather a family of regular, simplicial partitions of D, which is uniformly κ-shape regular, the value $\mathfrak{s}(\mathcal{V})$ is bounded from above in terms of κ, according to Remark 1.1. Using again that $|\mathcal{V}| \leqslant (d+1)|\mathcal{M}|$ and $|\mathcal{N}| \simeq p^d|\mathcal{M}|$, it follows that

$$\begin{aligned} M(\Phi^{CPwP_p}) &\leqslant C(d|\mathcal{N}| + p^2|\mathcal{V}| + d^2\mathfrak{s}(\mathcal{V})|\mathcal{V}|) \\ &\leqslant C(d,\kappa)(p^d|\mathcal{M}| + p^2|\mathcal{M}| + |\mathcal{M}|) \leqslant C(d,\kappa)|\mathcal{N}|. \end{aligned}$$

R e m a r k 4.8. Because the outputs of Φ^{CPwP_p} are continuous, piecewise polynomial Lagrangian basis functions which vanish in all nodes in $\mathcal{N}$ except one, it follows

that the coefficients $(v_i)_{i\in\mathcal{N}}$ are equal to the function values in $\mathcal{N}$, i.e., $v_i = v(i)$ for all $i \in \mathcal{N}$. In particular, the construction of $\Phi^{CPwP_p,v}$ is well suited for obtaining NN approximations based on high order finite element interpolation.

R e m a r k 4.9 (Univariate case $d = 1$). Although the proposition is stated for $d \geqslant 2$, the NN construction in the proof of Proposition 4.6 also holds for $d = 1$. However, the proposition does not hold as stated in the univariate case $d = 1$, because the parallel emulation of p univariate polynomials of degree p which is used in the proof has network size bound Cp^2.

In the univariate case, when $d = 1$, an exact NN emulation with size $|\mathcal{N}| \simeq p|\mathcal{M}|$ can be constructed as follows. In [42], Appendices A and B, for $p \in \mathbb{N}$ we constructed an NN with one input and p outputs, of depth $\lceil \log_2(p) \rceil + 1$ and size at most Cp, whose outputs approximate the Chebyšev polynomials of degree $1, \ldots, p$. While [42], Proposition A.2 is stated for tanh-NNs, by [42], Remark B.4 it also holds for ReLU^2 NNs. In the proof of [42], Proposition A.2, approximations of the Chebyšev polynomials are constructed using subnetworks, among which subnetworks that approximate the identity with a network of fixed size, independently of the accuracy. Similarly, subnetworks approximating the product of two numbers are used whose size is fixed and independent of the desired accuracy. When we replace those by the product and identity networks from Proposition 3.7 and 3.8, we obtain a ReLU^2 NN which exactly emulates the Chebyšev polynomials of degrees $1, \ldots, p$ with depth $\lceil \log_2(p) \rceil + 1$ and network size at most Cp. This result can be used in the constructions from [41] instead of [41], Proposition 4.8. Following the steps in the proofs of [41], Corollary 4.10 and Proposition 4.11 then gives an exact ReLU- and ReLU^2-activated NN emulation of univariate continuous, piecewise polynomial functions on a partition of a bounded interval into $|\mathcal{M}|$ elements with elementwise polynomial degree p, whose depth is of the order $O(\log(p + 1))$ and with a network size of the order $O(p|\mathcal{M}|)$.

5. NN approximation of weighted Gevrey regular functions

The main result is now obtained as a direct consequence of Proposition 2.2 and Proposition 4.6.

Theorem 5.1. *In a bounded polytope* $\mathrm{D} \subset \mathbb{R}^d$, $d = 2, 3$, *with plane sides (or faces), suppose given a weight vector* $\underline{\beta}$ *as in* (2.3) *if* $d = 3$ *or* (2.4) *if* $d = 2$. *Then for all* $\delta > 0$ *there exist constants* $b, \tilde{b}, C > 0$ *depending on* C_u, d_u *in* (2.6), *such that for every* $u \in \mathcal{G}^{\delta}_{\underline{\beta}}(\mathcal{S}; \mathrm{D})$ *and for every polynomial degree* $p \in \mathbb{N}$, *there exists an* NN

$\Phi^{hp,u,p}$ *with only* ReLU *and* ReLU^2 *activation, which satisfies*

$$
\begin{aligned}
(5.1)\quad & L(\Phi^{hp,u,p}) \leqslant C\log_2(p+1), \quad M(\Phi^{hp,u,p}) \leqslant Cp^{d+1/\delta}, \\
& \|u - \mathrm{R}(\Phi^{hp,u,p})\|_{H^1(\mathrm{D})} \\
& \qquad \leqslant \begin{cases} C\exp(-bp^{1/\delta}) \leqslant C\exp(-\tilde{b}M(\Phi^{hp,u,p})^{1/(1+\delta d)}), & \delta \geqslant 1, \\ C(\Gamma(p^{1/\delta}))^{-b(1-\delta)} \leqslant C(\Gamma(M(\Phi^{hp,u,p})^{1/(1+\delta d)}))^{-\tilde{b}(1-\delta)}, & 0 < \delta < 1. \end{cases}
\end{aligned}
$$

In addition, if $u|_{\partial\mathrm{D}} = 0$, *then* $\mathrm{R}(\Phi^{hp,u,p})|_{\partial\mathrm{D}} = 0$.

P r o o f. For some $0 < \sigma < 1$ and $\kappa > 1$, consider a sequence $\mathfrak{M}_{\kappa,\sigma}(\mathcal{S})$ of nested, regular simplicial meshes in D which are σ-geometrically refined towards $\mathcal{S}$ and which are κ-shape regular, e.g., those constructed in [19], Algorithm 1. By Proposition 2.2, for all $p \in \mathbb{N}$, with $l \simeq p^{1/\delta}$, the continuous, piecewise polynomial function $v := \Pi^p_{\kappa,\sigma} u \in S^p(\mathcal{M}^{(l)})$ satisfies

$$
\|u - v\|_{H^1(\mathrm{D})} \leqslant C \begin{cases} \exp(-bN^{1/(1+\delta d)}), & \delta \geqslant 1, \\ (\Gamma(N^{1/(1+\delta d)}))^{-b(1-\delta)}, & 0 < \delta < 1, \end{cases}
$$

where $N = \dim(S^p(\mathcal{M}^{(l)})) \simeq p^{d+1/\delta}$. If $u|_{\partial\mathrm{D}} = 0$, then $v|_{\partial\mathrm{D}} = 0$.

Now, by Proposition 4.6 there exists $\Phi^{hp,u,p} := \Phi^{CPwP_p(\mathcal{M}^{(l)},\mathrm{D}),v}$ with only ReLU and ReLU^2 activations, which satisfies for a constant $C > 0$ independent of $\mathcal{M}^{(l)}$, d and p

$$
\begin{aligned}
& \mathrm{R}(\Phi^{CPwP_p(\mathcal{M}^{(l)},\mathrm{D}),v})(x) = v(x) \quad \forall\, x \in \mathrm{D}, \\
& \quad L(\Phi^{CPwP_p(\mathcal{M}^{(l)},\mathrm{D}),v}) \leqslant C(\log_2(p+1) + \log_2(d+1) + \log_2(\mathfrak{s}(\mathcal{V})) + 1), \\
& \quad M(\Phi^{CPwP_p(\mathcal{M}^{(l)},\mathrm{D}),v}) \leqslant C(d|\mathcal{N}| + p^2|\mathcal{V}| + d^2\mathfrak{s}(\mathcal{V})|\mathcal{V}|).
\end{aligned}
$$

By Remark 1.1, $\mathfrak{s}(\mathcal{V})$ can be bounded in terms of the shape regularity constant κ, which is bounded independently of p. Also, $d \in \{2,3\}$ is bounded independently of p, which shows that $L(\Phi^{hp,u,p}) \leqslant C\log_2(p+1)$ for $C > 0$ depending on κ. To estimate the network size, we recall from Remark 4.7 that $M(\Phi^{hp,u,p}) = M(\Phi^{CPwP_p(\mathcal{M}^{(l)},\mathrm{D}),v}) \leqslant C|\mathcal{N}| = CN \simeq p^{d+1/\delta}$ for a constant C depending on κ, independent of p. To finish the proof, we substitute this into the error bound. □

R e m a r k 5.2. The treatment of homogeneous Dirichlet boundary conditions on a strict subset $\Gamma_{\mathrm{dir}} \subset \partial\mathrm{D}$ comprising the union of several boundary faces (when $d = 3$) or edges (when $d = 2$) is the topic of [19], Section 4.2.7. Because Proposition 2.2 is based on nodal interpolation, it also holds that $u|_{\Gamma_{\mathrm{dir}}} = 0$ implies that $\Pi^p_{\kappa,\sigma} u|_{\Gamma_{\mathrm{dir}}} = 0$. By the presently developed exact NN emulation, this implies that $\mathrm{R}(\Phi^{hp,u,p})|_{\Gamma_{\mathrm{dir}}} = 0$.

6. Conclusions

For DNNs with a suitable combination of ReLU and ReLU^2 activations, we established exponential expression rate bounds (5.1) in polytopal domains $\mathrm{D} \subset \mathbb{R}^d$, in dimension $d = 2, 3$. The results built on a general result on DNN emulation of *hp*-FE spaces, Proposition 4.6, which is of independent interest, also allowing for DNN emulation of the so-called *p*-version finite element method, and spectral element methods in D. Proposition 4.6 improves earlier results from [27] by realizing the same finite element with NNs which are smaller, i.e., the network size admits smaller bounds in terms of the polynomial degree p (cf. Remark 4.5). With Proposition 4.6 in place, exponential expression rate bounds for the constructed DNNs on countably normed, corner-weighted function classes (2.6) follow from known results on *hp*-FE approximation e.g., in [19], which go back to the *hp*-FE approximation rates by Babuška and Guo in [21], [3], [5]. The present results also imply further "DNN-versions" of *hp*-FE Methods, such as [6], [7], [18].

The proposed DNN emulation of Lagrangian basis functions of the space of polynomials of total degree $p \geqslant 1$ in equispaced nodes (in barycentric coordinates in simplicial elements $K \in \mathcal{M}$) is, as was shown in the proof of Proposition 4.6, mathematically exact. The well-known conditioning issues of these nodal basis functions may preclude their training in finite-precision arithmetic for very high orders. Here, the *modal bases* such as those proposed by Babuška and coworkers, which are based on antiderivatives of univariate Legendre polynomials, will afford better conditioning of the resulting "feature space", but do not exhibit the separability which we used in the proof of Proposition 4.6. Other polynomial bases, both nodal and modal, with substantially better conditioning for large polynomial orders, are known (e.g., [12], [30]). *Separable, nodal bases* are provided by Dubiner's approach [15], which also yields bases whose conditioning is improved w.r. to the Lagrangian basis (4.2) used in the proof of Proposition 4.6. The NN size bounds for DNN emulation of these bases is the topic of future work.

Open Access. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted

by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

[1] *R. Arora, A. Basu, P. Mianjy, A. Mukherjee*: Understanding deep neural networks with rectified linear units. Available at https://arxiv.org/abs/1611.01491 (2018), 17 pages. doi

[2] *I. Babuška, B. Q. Guo*: Regularity of the solution of elliptic problems with piecewise analytic data. I. Boundary value problems for linear elliptic equations of second order. SIAM J. Math. Anal. *19* (1988), 172–203. zbl MR doi

[3] *I. Babuška, B. Q. Guo*: The *h-p* version of the finite element method for domains with curved boundaries. SIAM J. Numer. Anal. *25* (1988), 837–861. zbl MR doi

[4] *I. Babuška, B. Q. Guo*: Regularity of the solution of elliptic problems with piecewise analytic data. II. The trace spaces and application to the boundary value problems with nonhomogeneous boundary conditions. SIAM J. Math. Anal. *20* (1989), 763–781. zbl MR doi

[5] *I. Babuška, B. Q. Guo*: Approximation properties of the *h-p* version of the finite element method. Comput. Methods Appl. Mech. Eng. *133* (1996), 319–346. zbl MR doi

[6] *L. Banjai, J. M. Melenk, C. Schwab*: Exponential convergence of *hp* FEM for spectral fractional diffusion in polygons. Numer. Math. *153* (2023), 1–47. zbl MR doi

[7] *L. Banjai, J. M. Melenk, C. Schwab*: *hp*-FEM for reaction-diffusion equations. II. Robust exponential convergence for multiple length scales in corner domains. IMA J. Numer. Anal. *43* (2023), 3282–3325. zbl MR doi

[8] *G. Bao, G. Hu, D. Liu*: An *h*-adaptive finite element solver for the calculations of the electronic structures. J. Comput. Phys. *231* (2012), 4967–4979. zbl doi

[9] *P. Bolley, M. Dauge, J. Camus*: Régularité Gevrey pour le problème de Dirichlet dans des domaines à singularités coniques. Commun. Partial Differ. Equations *10* (1985), 391–431. (In French.) zbl MR doi

[10] *E. Cancès, R. Chakir, Y. Maday*: Numerical analysis of nonlinear eigenvalue problems. J. Sci. Comput. *45* (2010), 90–117. zbl MR doi

[11] *J.-Y. Chemin*: Perfect Incompressible Fluids. Oxford Lecture Series in Mathematics and Its Applications 14. Clarendon Press, New York, 1998. zbl MR doi

[12] *Q. Chen, I. Babuška*: The optimal symmetrical points for polynomial interpolation of real functions in the tetrahedron. Comput. Methods Appl. Mech. Eng. *137* (1996), 89–94. zbl MR doi

[13] *M. Costabel, M. Dauge, S. Nicaise*: Singularities of Maxwell interface problems. M2AN, Math. Model. Numer. Anal. *33* (1999), 627–649. zbl MR doi

[14] *M. Costabel, M. Dauge, S. Nicaise*: Analytic regularity for linear elliptic systems in polygons and polyhedra. Math. Models Methods Appl. Sci. *22* (2012), Article ID 1250015, 63 pages. zbl MR doi

[15] *M. Dubiner*: Spectral methods on triangles and other domains. J. Sci. Comput. *6* (1991), 345–390. zbl MR doi

[16] *D. Elbrächter, P. Grohs, A. Jentzen, C. Schwab*: DNN expression rate analysis of high-dimensional PDEs: Application to option pricing. Constr. Approx. *55* (2022), 3–71. zbl MR doi

[17] *A. Ern, J.-L. Guermond*: Finite Elements. I. Approximation and Interpolation. Texts in Applied Mathematics 72. Springer, Cham, 2021. zbl MR doi

[18] *M. Faustmann, C. Marcati, J. M. Melenk, C. Schwab*: Exponential convergence of *hp*-FEM for the integral fractional Laplacian in polygons. SIAM J. Numer. Anal. *61* (2023), 2601–2622. zbl MR doi
[19] *M. Feischl, C. Schwab*: Exponential convergence in H^1 of *hp*-FEM for Gevrey regularity with isotropic singularities. Numer. Math. *144* (2020), 323–346. zbl MR doi
[20] *S. Fournais, M. Hoffmann-Ostenhof, T. Hoffmann-Ostenhof, T. Ø. Sørensen*: Analytic structure of solutions to multiconfiguration equations. J. Phys. A. Math. Theor. *42* (2009), Article ID 315208, 11 pages. zbl MR doi
[21] *B. Guo*: The *h-p* Version of Finite Element Method in Two Dimensions. Mathematical Theory and Computational Experience: Ph.D. Thesis. University of Maryland, College Park, 1985. MR
[22] *B. Q. Guo, I. Babuška*: On the regularity of elasticity problems with piecewise analytic data. Adv. Appl. Math. *14* (1993), 307–347. zbl MR doi
[23] *B. Guo, I. Babuška*: Regularity of the solutions for elliptic problems on nonsmooth domains in $\mathbb{R}^3$. I. Countably normed spaces on polyhedral domains. Proc. R. Soc. Edinb., Sect. A *127* (1997), 77–126. zbl MR doi
[24] *B. Guo, I. Babuška*: Regularity of the solutions for elliptic problems on nonsmooth domains in $\mathbb{R}^3$. II. Regularity in neighbourhoods of edges. Proc. R. Soc. Edinb., Sect. A *127* (1997), 517–545. zbl MR doi
[25] *B. Guo, C. Schwab*: Analytic regularity of Stokes flow on polygonal domains in countably weighted Sobolev spaces. J. Comput. Appl. Math. *190* (2006), 487–519. zbl MR doi
[26] *J. He, L. Li, J. Xu, C. Zheng*: ReLU deep neural networks and linear finite elements. J. Comput. Math. *38* (2020), 502–527. zbl MR doi
[27] *J. He, J. Xu*: Deep neural networks and finite elements of any order on arbitrary dimensions. Available at `https://arxiv.org/abs/2312.14276` (2024), 29 pages. doi
[28] *Y. He, C. Marcati, C. Schwab*: Analytic regularity of solutions to the Navier-Stokes equations with mixed boundary conditions in polygons. SIAM J. Math. Anal. *56* (2024), 2488–2520. zbl MR doi
[29] *L. Herrmann, J. A. A. Opschoor, C. Schwab*: Constructive deep ReLU neural network approximation. J. Sci. Comput. *90* (2022), Article ID 75, 37 pages. zbl MR doi
[30] *J. S. Hesthaven*: From electrostatics to almost optimal nodal sets for polynomial interpolation in a simplex. SIAM J. Numer. Anal. *35* (1998), 655–676. zbl MR doi
[31] *B. Holm, T. P. Wihler*: Continuous and discontinuous Galerkin time stepping methods for nonlinear initial value problems with application to finite time blow-up. Numer. Math. *138* (2018), 767–799. zbl MR doi
[32] *B. Li, S. Tang, H. Yu*: Better approximations of high dimensional smooth functions by deep neural networks with rectified power units. Commun. Comput. Phys. *27* (2020), 379–411. zbl MR doi
[33] *M. Longo, J. A. A. Opschoor, N. Disch, C. Schwab, J. Zech*: De Rham compatible deep neural network FEM. Neural Netw. *165* (2023), 721–739. zbl doi
[34] *Y. Maday, C. Marcati*: Regularity and *hp* discontinuous Galerkin finite element approximation of linear elliptic eigenvalue problems with singular potentials. Math. Models Methods Appl. Sci. *29* (2019), 1585–1617. zbl MR doi
[35] *C. Marcati, J. A. A. Opschoor, P. C. Petersen, C. Schwab*: Exponential ReLU neural network approximation rates for point and edge singularities. Found. Comput. Math. *23* (2023), 1043–1127. zbl MR doi
[36] *C. Marcati, C. Schwab*: Analytic regularity for the incompressible Navier-Stokes equations in polygons. SIAM J. Math. Anal. *52* (2020), 2945–2968. zbl MR doi
[37] *J. M. Melenk*: *hp*-Finite Element Methods for Singular Perturbations. Lecture Notes in Mathematics 1796. Springer, Berlin, 2002. zbl MR doi

[38] *J. M. Melenk, C. Schwab*: *hp* FEM for reaction-diffusion equations. I. Robust exponential convergence. SIAM J. Numer. Anal. *35* (1998), 1520–1557. zbl MR doi
[39] *J. A. A. Opschoor*: Constructive Deep Neural Network Approximations of Weighted Analytic Solutions to Partial Differential Equations in Polygons: Ph.D. Thesis. ETH, Zürich, 2023. doi
[40] *J. A. A. Opschoor, P. C. Petersen, C. Schwab*: Deep ReLU networks and high-order finite element methods. Anal. Appl., Singap. *18* (2020), 715–770. zbl MR doi
[41] *J. A. A. Opschoor, C. Schwab*: Deep ReLU networks and high-order finite element methods. II. Chebyšev emulation. Comput. Math. Appl. *169* (2024), 142–162. zbl MR doi
[42] *J. A. A. Opschoor, C. Schwab, C. Xenophontos*: Neural networks for singular perturbations. Available at `https://arxiv.org/abs/2401.06656` (2024), 39 pages. doi
[43] *J. A. A. Opschoor, C. Schwab, J. Zech*: Exponential ReLU DNN expression of holomorphic maps in high dimension. Constr. Approx. *55* (2022), 537–582. zbl MR doi
[44] *P. Petersen, F. Voigtlaender*: Optimal approximation of piecewise smooth functions using deep ReLU neural networks. Neural Netw. *108* (2018), 296–330. zbl doi
[45] *A. A. Samarskii, V. A. Galaktionov, S. P. Kurdyumov, A. P. Mikhailov*: Blow-Up in Quasilinear Parabolic Equations. de Gruyter Expositions in Mathematics 19. Walter De Gruyter, Berlin, 1995. zbl MR doi
[46] *C. Schwab*: *p*- and *hp*-Finite Element Methods: Theory and Applications in Solid and Fluid Mechanics. Numerical Mathematics and Scientific Computation. Clarendon Press, Oxford, 1998. zbl MR
[47] *B. Szabó, I. Babuška*: Finite Element Analysis. John Wiley & Sons, New York, 1991. zbl MR
[48] *Y. Yang, Y. Wu, H. Yang, Y. Xiang*: Nearly optimal approximation rates for deep super ReLU networks on Sobolev spaces. Available at `https://arxiv.org/abs/2310.10766` (2023), 32 pages. doi
[49] *D. Yarotsky*: Error bounds for approximations with deep ReLU networks. Neural Netw. *94* (2017), 103–114. zbl doi

Authors' address: *Joost A. A. Opschoor*, *Christoph Schwab* (corresponding author), Seminar for Applied Mathematics, ETH Zürich, Rämistrasse 101, CH–8092 Zürich, Switzerland, `joost.opschoor@sam.math.ethz.ch`, `christoph.schwab@math.ethz.ch`.