Technische Universität Berlin

Solving Parametric PDEs with Neural Networks: Unfavorable Structure vs. Expressive Power

vorgelegt von

MONES KONSTANTIN RASLAN (M.Sc.)

an der Fakultät II – Mathematik und Naturwissenschaften
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
– Dr. rer. nat. –

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender:	Prof. Dr. Wilhelm Stannat
Gutachter:	Prof. Dr. Volker Mehrmann
Gutachter:	Prof. Dr. Philipp Grohs
Gutachter:	Prof. Dr. Helmut Bölcskei

Tag der wissenschaftlichen Aussprache: 23. März 2021

Berlin 2021

Abstract

This cumulative dissertation extends the theory of neural networks (NNs).

In the first part of this thesis, [PRV20] in Appendix A, we provide a general analysis of the hypothesis class of NNs from a structural point of view. Here, we examine the algebraic and topological properties of the set of NNs with fixed architecture. We establish that this set is never convex, hardly ever closed in classical function spaces and that the parametrization of NNs is not inverse stable. These observations could, in practice, lead to highly undesirable phenomena such as diverging weights or slow convergence of the underlying training algorithm.

The second part of this thesis deals with the concrete application of solving *parametric* partial differential equations (PDEs) by NNs. In typical modeling tasks, it is required to solve some PDE for different characterizing parameters such as the shape of the domain, the boundary conditions, or the right-hand side. In this context, the development of algorithms that are able to efficiently and accurately compute the solution for a new input is imperative. A large variety of reduced order models, taking into account the low-dimensionality of the solution set, have been developed in the past. Moving away from model-based techniques and motivated by their success in applications, in this thesis we focus on a data-driven approach based on NNs for the solution of parametric PDEs. A factor in favor of their use is their ability to calculate a new solution with little computational effort after training, when compared to the cost of the training phase. The focus of this part of the thesis lies on an examination of the expressive power of NNs for solutions of parametric PDEs. We first derive in [GR21] (see Appendix B) almost optimal approximation rates for smooth functions by NNs with encodable weights, measured with respect to Sobolev norms. These results continue a long avenue of research and provide a consolidating proof strategy for deriving expressivity results based on the regularity of the target function. However, if we apply the results from [GR21] to the approximation of the solution map of parametric PDEs, we might end up with sub-optimal rates. In fact, the upper bounds from [GR21] completely ignore the low-dimensionality of the solution manifold. A remedy to overcome this drawback is our theoretical analysis [KPRS21] (see Appendix C) that establishes improved upper complexity bounds for the approximation of the solution map based on the intrinsic dimension of the solution set. Although theoretical approximation results of the above flavor give some intuition for the success of NNs, they can at most partially explain why NNs work so well in practice. Moreover, it is not clear, to which extent purely asymptotic approximation rates are visible in practice. In our last contribution to this thesis, [GPRSK20] in Appendix D, we provide a comprehensive and systematic numerical study for the practical observation of approximation rates. We concentrate on a large variety of parametrizations of the Poisson equation. We derive numerical complexity bounds for the approximation of the solution map by NNs that do not suffer from the curse of dimensionality and only weakly depend on the dimension of the parameter space.

Concluding, we observe that NNs, despite their unfavorable structure, possess a huge potential for their application within the framework of parametric PDEs.

Zusammenfassung in deutscher Sprache

Die vorliegende Dissertation erweitert die Theorie neuronaler Netze (NNe).

Zunächst untersuchen wir im ersten Teil dieser Arbeit, [PRV20] in Appendix A, die strukturellen Merkmale der Menge der NNe mit fester Architektur. Dabei legen wir einen besonderen Fokus auf ihre algebraischen und topologischen Eigenschaften. Wir stellen fest, dass die Menge nie konvex ist, selten abgeschlossen in klassischen Funktionenräumen ist, und dass die Parametrisierung NNe nicht invers stabil ist. Die praktisch denkbaren Konsequenzen dieser Resultate sind unerwünschte Phänomene wie explodierende Gewichte oder die langsame Konvergenz des zugrundeliegenden Trainingsalgorithmus'.

Der zweite Teil der Arbeit beschäftigt sich mit der konkreten Anwendung der Lösung von parametrischen partiellen Differentialgleichungen (parametrischen PDGen) durch NNe. Eine Vielzahl von Anwendungen erfordert die Lösung einer PDG für verschiedene Parameter, die die Gleichung charakterisieren. Dazu zählen z.B. die Form des zugrundeliegenden Gebiets, die rechte Seite oder die Randbedingungen. In solch einem Zusammenhang ist die Entwicklung effizienter und genauer Algorithmen notwendig, welche eine Approximation der tatsächlichen Lösung für einen neuen Parameter schnell ermitteln können. In der Vergangenheit wurde eine Vielzahl von Methoden zur Reduktion der Modellordnung entwickelt, welche auf der Niedrigdimensionalität der Lösungsmenge basieren. Im Gegensatz zu klassischen, modell-basierten Methoden und motiviert durch ihren Erfolg in vielen praktischen Anwendungen beschäftigen wir uns in der vorliegenden Arbeit mit der Lösung von parametrischen PDGs durch NNe, welche mit einem datenbasierten Ansatz trainiert werden. Diese sind nach dem Training in der Lage, eine neue Lösung mit vergleichsweise kleinem Rechenaufwand zu bestimmen. Die Expressivität von NNen für die Lösung von parametrischen PDGen wird den Fokus des zweiten Teils der Dissertation bilden. Wir beginnen in [GR21] (siehe Appendix B) damit, quasi-optimale Raten für die Approximation von hinreichend glatten Funktionen durch NNe mit kodierbaren Gewichten in Sobolevräumen herzuleiten. Diese Ergebnisse setzen eine seit langem bestehende Forschungslinie fort, welche Raten basierend auf der Glattheit der Zielfunktion beinhalten. Wenn wir die in [GR21] gezeigten Resultate allerdings für die Approximation der Lösungsabbildung von parametrischen PDGen benutzen wollen, so erhalten wir u.U. suboptimale Raten. Dies ist der Tatsache geschuldet, dass die oberen Schranken aus [GR21] die Niedrigdimensionalität der Lösungsmannigfaltigkeit außer Acht lassen. In der Arbeit [KPRS21] (siehe Appendix C) leiten wir verbesserte obere Schranken für die Annäherung der Lösungsabbildung her. Diese fußen auf der Dimension der Lösungsmenge. Obwohl die soeben beschriebenen, approximations-theoretischen Überlegungen einen Einblick in den Erfolg von NNen in Anwendungen geben, so können sie diesen nur teilweise erklären. Weiterhin ist nicht klar, inwiefern die bewiesenen asymptotischen Raten auch in der Praxis sichtbar sind. In dem letzten Teil der Dissertation, [GPRSK20] in Appendix D, führen wir eine umfangreiche und systematische numerische Studie für die Beobachtung praktisch relevanter Approximationsraten durch. Wir konzentrieren uns hierbei auf eine Vielzahl von Parametrisierungen der Poissongleichung. Wir leiten numerische Schranken

für die Annäherung der Lösungsabbildung durch NNe her, welche nicht unter dem Fluch der Dimensionalität leiden und nur schwach von der Dimension des Parameterraums abhängen.

Abschließend stellen wir fest, dass NNe, trotz ihrer ungünstigen Struktur, ein hohes Potential für die Anwendung im Kontext von parametrischen PDGen besitzen.

List of Publications

The results of this cumulative dissertation have been previously published by the author and his collaborators. They are based on the following **four publications**:[1]

[PRV20] see App. A — P. Petersen, M. Raslan, and F. Voigtlaender. **Topological Properties of the Set of Functions Generated by Neural Networks of Fixed Size.** *Foundations of Computational Mathematics*, Online First (2020), 1–70. https://doi.org/10.1007/s10208-020-09461-0

[GR21] see App. B — I. Gühring and M. Raslan. **Approximation Rates for Neural Networks with Encodable Weights in Smoothness Spaces.** *Neural Networks*, 134 (2021), 107–130. https://doi.org/10.1016/j.neunet.2020.11.010

[KPRS21] see App. C — G. Kutyniok, P. Petersen, M. Raslan, and R. Schneider. **A Theoretical Analysis of Deep Neural Networks and Parametric PDEs.** *Constructive Approximation*, in Press (2021).

[GPRSK20] see App. D — M. Geist, P. Petersen, M. Raslan, R. Schneider, and G. Kutyniok. **Numerical Solution of the Parametric Diffusion Equation by Deep Neural Networks.** Preprint [arXiv:2004.12131v1], 2020.

Further publications by the author that are **not directly included** in this thesis are:

Survey Articles

[GRK20] — I. Gühring, M. Raslan, and G. Kutyniok. **Expressivity of Deep Neural Networks.** Preprint [arXiv:2007.04759], 2020.
This review paper will appear as a book chapter in "Theory of Deep Learning" by Cambridge University Press.

Journal Publications

[PR19] — P. Petersen and M. Raslan. **Approximation properties of hybrid shearlet-wavelet frames for Sobolev spaces.** *Advances in Computational Mathematics*, 45.3 (2019), 1581–1606.

[GKMPR20] — P. Grohs, G. Kutyniok, J. Ma, P. Petersen, and M. Raslan. **Anisotropic multiscale systems on bounded domains.** *Advances in Computational Mathematics*, 46.39 (2020).

Conference Proceedings

[PRV19a] — P. Petersen, M. Raslan, and F. Voigtlaender. **Unfavorable structural properties of the set of neural networks with fixed architecture .** *2019 13th International conference on Sampling Theory and Applications (SampTA)*, Bordeaux, France (2019).

[PRV19b] — P. Petersen, M. Raslan, and F. Voigtlaender. **The structure of spaces of neural network functions.** *Proceedings Volume 11138, Wavelets and Sparsity XVIII; 111380F*, San Diego, USA (2019).

[1] [KPRS21] Accepted and soon to be published; A Theoretical Analysis of Deep Neural Networks and Parametric PDEs, *Constructive Approximation*, Springer
[GPRSK20]: Accepted and soon to be published; Numerical Solution of the Parametric Diffusion Equation by Deep Neural Networks, *Journal of Scientific Computing*, Springer / Version included in this thesis: Preprint [arXiv:2004.12131v1]

Acknowledgments

This thesis would not have been possible without a number of people who accompanied me throughout these past years.

First of all, I am indebted to Philipp Petersen for his guidance and support. It has been a joy working with him on many fruitful projects and I am immensely grateful for the advice he gave me.

I would profoundly like to thank Volker Mehrmann, Philipp Grohs, and Helmut Bölcskei for agreeing to review this thesis, which makes me feel very honored. I am very grateful to Wilhelm Stannat for chairing the doctoral committee.

I wish to express my deepest gratitude to my further co-authors Moritz Geist, Ingo Gühring, Reinhold Schneider, and Felix Voigtlaender for the fruitful collaborations that lead to the contents of this thesis. Additionally, I would like to thank Gitta Kutyniok for the opportunities she gave me.

I would like to thank all current and former members of the Applied Functional Analysis Group at the Institute of Mathematics at TU Berlin. It was great to have been a part of this group. I am immensely thankful to my (long-term office) mate Maximilian März as well as for the friendships I formed during that time, especially with Katharina Eller, Ingo Gühring, Martin Genzel, Sandra Keiper, Jan Macdonald, Philipp Petersen, and Patrick Winkert.

I am thankful to Moritz Geist, Ingo Gühring, Jan Macdonald, and Maximilian März for proofreading the main body of this thesis.

I would also like to thank the Berlin Mathematical School a part of which I have been.

On the personal side, I am immensely grateful to my parents, Veronika and Tarek, and to my sister, Lena, for their constant support over the last years. Finally, I owe my gratitude to all of my friends and to those who are dear to me. Here, I would like to pay my special regards to Enis Arkat, and to Dominique Sertel, also for the best singing lessons imaginable.

Contents

List of Figures

1

Introduction

This cumulative dissertation advances the mathematical theory of artificial neural networks, with a particular focus on the application of *parametric partial differential equations*. Artificial neural networks (henceforth referred to as NNs) were first introduced in [MP43] and were originally motivated by biological considerations. They constitute the hypothesis class in the subset of machine learning algorithms coined *deep learning*, e.g., see [LBH15; Sch15; GBC16]. The resulting data-driven methods have lead to performance breakthroughs in numerous applications including image classification [KSH12], translation tasks [Jac+19], autonomous driving [TPJR18], or fraud detection [FDPZP19]. In the last decades, they have also begun to influence classical research areas of (applied) mathematics. Notable instances are inverse problems [JMFU17], optimal control problems [MA01], and partial differential equations (PDEs) [LLF98; EY18]. Oftentimes in these applications, sophisticated methods are developed by combining preexisting knowledge about the underlying problem with a learning-based approach. Despite promising and sometimes even spectacular results, many theoretical facets of deep learning remain unclear, e.g., see [EMWW20].

The purpose of this thesis is to obtain a better understanding of NNs and their applicability to solving parametric PDEs by studying the following two subjects in detail.

Subject A. Structural Properties of NNs with Fixed Architecture: The typical workflow of a deep learning application includes the fixation of an NN *architecture* and an *activation function*. Afterwards, an optimization algorithm (such as stochastic gradient descent or a modification thereof) aims at finding a weight configuration among all NNs with the prescribed architecture and activation function such that a specific problem (e.g., a classification task or a PDE) is solved sufficiently well. During training, there frequently happen undesirable phenomena. These include slow convergence of the optimization procedure or exploding network weights. By studying the hypothesis class of NNs from a structural point of view, we aim at identifying potential causes for these occurrences. This brings us to the first set of questions that is examined in this thesis:

Which algebraic and analytical properties does the set of NNs
with fixed architecture and activation function have?
Are there practical consequences that can be concluded from these properties?

Subject B. Expressivity of Deep NNs for Solving Parametric Elliptic PDEs: When real-world-phenomena are modeled by PDEs, a multitude of *parameters* such as the underlying domain, the right-hand side, boundary conditions, or multiplicative coefficients are fixed beforehand. In a *multi-query context*, one solves a PDE for a large number of parameters [QMN16]. In this framework, classical finite-element-solvers are not suitable from a computational point of view, since they devour too many resources. A well-established

remedy are *reduced order modeling techniques*, which rely on the inherent low-dimensionality of the solution manifold. Compared to many of these techniques, NNs are fully *non-intrusive*, since they do not require knowledge of the underlying PDE and are able to act as a black-box-solver. Furthermore, the computation of the solution of a PDE with new parameters after training is very efficient, since it only consists of a feedforward pass. Despite these obvious merits, it is not clear to which extent NNs are capable of solving parametric PDEs. We examine this topic by studying the following approximation-theoretical questions in detail:

How many weights are sufficient for an NN to approximate the parameter-to-solution map?
Which approximation rates can be observed numerically?

Outline: In the remainder of this introduction, we motivate the aforementioned questions by embedding them into the basic framework of *statistical learning theory*. Section 1.1.1 is devoted to an introduction of abstract statistical learning problems, whereas we introduce parametric elliptic PDEs and their interpretation within the framework of machine learning, in Section 1.2. Section 1.3 contains a concise and high-level, non-mathematical overview of the findings of this thesis. Chapter 2 presents and discusses the results of Subject A. The corresponding publication [PRV20] can be found in Appendix A. Chapter 3 is devoted to Subject B. In particular, Section 3.1 is concerned with the paper [GR21] of Appendix B, Section 3.2 deals with [KPRS21] of Appendix C and Section 3.3 focuses on the results of [GPRSK20] in Appendix D. We conclude this thesis in Chapter 4, where we reflect upon our results and depict potential future research directions. All parts of the main body include extended related work sections.

A Short Note on this Thesis' Structure: Chapters 1–4 contain an almost self-contained overview of the papers in Appendices A–D. We establish a common thread that connects these four works. Additionally, we expound related works in more detail than in the publications. Section 1.1 provides the reader with the basics of machine learning which are sufficient for understanding [PRV20]. Reading Section 1.2 additionally helps to comprehend [GPRSK20; GR21; KPRS21].

1.1 Statistical Learning Problems and NNs: Basic Concepts

We start by giving a concise introduction to statistical learning problems and typical, associated questions in Section 1.1.1. In Section 1.1.2, we formally introduce NNs.

1.1.1 Statistical Learning Problems

In its most basic form[1], the ingredients of a *(statistical) learning problem*, e.g., see [CS02; CZ07; LS11; GRK20], are an *input space* $\mathcal{X}$, a *target space* $\mathcal{Z}$, a *loss function* $\mathcal{L} : \mathcal{Z} \times \mathcal{Z} \to$

[1] Later on, we will interpret parametric PDEs in the framework of machine learning. For this purpose, let us already mention at this point that many modifications of the problem described here are possible. For instance, instead of the risk functional being defined on $\mathcal{Z}^{\mathcal{X}}$, one could choose a normed subspace of $\mathcal{Z}^{\mathcal{X}}$ as its domain of definition.

$[0,\infty]$ and a *(generally unknown probability) measure* σ on $\mathcal{X} \times \mathcal{Z}$. One then seeks to find a minimizer $\hat{f}$ (also called *target function*) of the *risk functional*[2]

$$\mathbf{F} : \mathcal{Z}^{\mathcal{X}} \to [0,\infty],\ f \mapsto \int_{\mathcal{X}\times\mathcal{Z}} \mathcal{L}\left(f(\mathbf{x}), \mathbf{z}\right)\ d\sigma(\mathbf{x}, \mathbf{z}),$$

with the convention that $\mathbf{F}(f) = \infty$ if the integral is not well-defined. In most applications, solving a learning problem in the form described above is not possible for the following reasons[3]:

- Searching over the whole set $\mathcal{Z}^{\mathcal{X}}$ is infeasible in general. Hence, one confines oneself to a smaller, structured *hypothesis* or *model space* $\mathcal{M} \subset \mathcal{Z}^{\mathcal{X}}$, and either hopes to determine

  $$\hat{f}_{\mathcal{M}} \in \operatorname{argmin}\left\{\mathbf{F}(f) :\ f \in \mathcal{M}\right\},$$

  or an approximate minimizer. Depending on the structure of $\mathcal{X}$ and $\mathcal{Z}$, a multitude of hypothesis spaces have been deemed relevant for practical applications. Notable examples include *linear functions*, higher-order *polynomials*, or *reproducing kernel Hilbert spaces*, e.g., see [CZ07, Section 2].

  The focus of this dissertation is on standard *feedforward NNs* as the choice of $\mathcal{M}$. For mathematically precise notions, we refer to Section 1.1.2.

  The quantity $\left|\mathbf{F}(\hat{f}) - \mathbf{F}(\hat{f}_{\mathcal{M}})\right|$ is called *approximation error*. It measures the *expressivity* of the hypothesis space $\mathcal{M}$, i.e., its ability to approximate functions $\hat{f} \in \mathcal{Z}^{\mathcal{X}}$.

- A calculation of $\mathbf{F}(f)$ for a given function f is impossible if σ is unknown. Even if σ was known, calculating $\hat{f}_{\mathcal{M}}$ is an infeasible task in general. In the context of *supervised learning*, only $m \in \mathbb{N}$ *training samples* $(\mathbf{x}_i, \mathbf{z}_i)_{i=1}^m \subset \mathcal{X} \times \mathcal{Z}$ (drawn i.i.d. with respect to σ) are accessible, e.g., see [HTF09]. One then aims at determining a minimizer of the *empirical risk* $\mathbf{F}^m_{\text{emp}}(f) = \frac{1}{m}\sum_{i=1}^m \mathcal{L}\left(f(\mathbf{x}_i), \mathbf{z}_i\right)$, i.e.,

  $$\hat{f}^m_{\mathcal{M},\text{emp}} \in \operatorname{argmin}\left\{\mathbf{F}^m_{\text{emp}}(f) :\ f \in \mathcal{M}\right\}.$$

  The quantity $\left|\mathbf{F}(\hat{f}_{\mathcal{M}}) - \mathbf{F}(\hat{f}^m_{\mathcal{M},\text{emp}})\right|$ is referred to as the *estimation* or *generalization error*.

- In order to find $\hat{f}^m_{\mathcal{M},\text{emp}}$, one needs to solve a potentially non-convex optimization problem. The resulting algorithm ideally returns an approximation $\hat{f}^{m,*}_{\mathcal{M},\text{emp}}$ of $\hat{f}^m_{\mathcal{M},\text{emp}}$. The quantity $\left|\mathbf{F}(\hat{f}^m_{\mathcal{M},\text{emp}}) - \mathbf{F}(\hat{f}^{m,*}_{\mathcal{M},\text{emp}})\right|$ is called *training error*.

Thus, the *overall error* $\left|\mathbf{F}(\hat{f}) - \mathbf{F}(\hat{f}^{m,*}_{\mathcal{M},\text{emp}})\right|$ can be estimated by the sum of approximation error, generalization error and training error (see Figure 1.1 for a schematic visualization).

In theoretical considerations, the hardness of a learning algorithm is often measured by suitable estimates on the three errors introduced above. Typical questions include:

[2] $\mathcal{Z}^{\mathcal{X}}$ denotes the set of all functions $\mathcal{X} \to \mathcal{Z}$.

[3] As every model, a statistical learning problem is an idealization of a real-world application and hence itself flawed [DNR14]. The resulting *modeling error* is not examined in classical works on statistical learning.

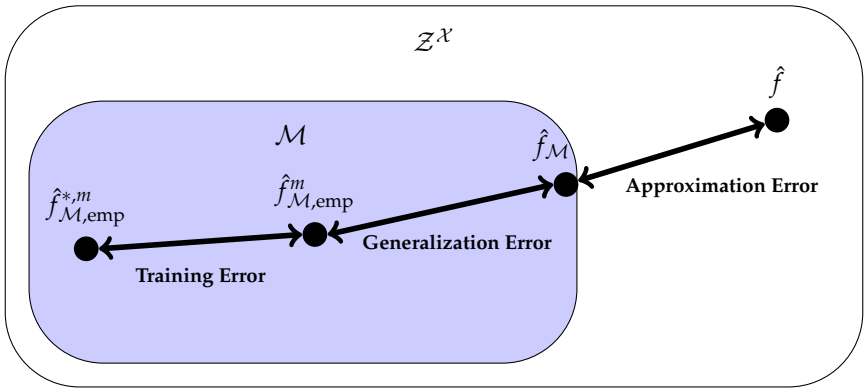


Figure 1.1: Decomposition of the overall error into approximation error, estimation error and training error. This figure is an adaptation of [GRK20, Figure 1].

- For a given target function $\hat{f}$ and decreasing approximation error ε, is it possible to quantify how large the hypothesis space $\mathcal{M} = \mathcal{M}(\varepsilon)$ needs to be in order to obtain $|\mathbf{F}(\hat{f}) - \mathbf{F}(\hat{f}_{\mathcal{M}})| \leq \varepsilon$? Conversely, for a given hypothesis space $\mathcal{M}$, how can we estimate $|\mathbf{F}(\hat{f}) - \mathbf{F}(\hat{f}_{\mathcal{M}})|$?
- How fast does m need to grow in order for the generalization error to converge to 0 with high probability?
- What are the properties of $\mathbf{F}|_{\mathcal{M}}$ and $\mathbf{F}^m_{\text{emp}}$? In case these functions are non-convex, does their *loss landscape* encompass sub-optimal local minima, saddle points, global minima? How do the valleys around minima look like? How effective are (iterative) optimization procedures such as *stochastic gradient descent* or more sophisticated modifications thereof (e.g., batch gradient descent based on *Adam* [KB15]) in finding good local or global minima? How fast does the training error decrease for a *given training set* $(\mathbf{x}_i, \mathbf{z}_i)_{i=1}^m$, as the number of iterations increases?

In many situations, the examination of the aforementioned questions is a complex task, for instance due to the unknown nature of σ. However, even from an abstract point of view, it becomes clear that in order to understand the complete picture, it is imperative to balance the analyses of the three errors described above against each other. For instance, a mere enlargement of the hypothesis space in order to decrease the approximation error might result in a larger generalization error. This phenomenon has been manifested by the *bias-variance* decomposition of the overall error [CS02]. It (at least partially) explains *overfitting* of the training data for a large hypothesis space and *underfitting* in the case of a small hypothesis space. Of course, the hardness of the learning problem is also highly dependent on the choice of the loss function $\mathcal{L}$. Depending on the application at hand, popular choices for loss functions in *classification tasks*[4] are variations of the cross-entropy loss or the hinge loss, whereas popular choices for loss functions in *regression tasks*[5] are the quadratic loss or the absolute loss [GBC16]. In the case of the quadratic loss function, it is at least abstractly possible to write down the target function $\hat{f}$, referred to as the *regression function*[6] *of* σ [CS02].

[4] I.e., when $\mathcal{Z}$ is discrete.
[5] I.e., when $\mathcal{Z}$ is not discrete.
[6] It is given by $\hat{f}(\mathbf{x}) = \int_{\mathcal{Z}} \mathbf{z}\, d\sigma(\mathbf{z}|\mathbf{x})$.

We will see in Section 1.2.4 that it is possible to interpret parametric PDEs as a regression problem with (compared to other applications) significant simplifications: the target function, the measure σ, and a suitable loss function $\mathcal{L}$ are often known a priori.

1.1.2 The Hypothesis Class of NNs: Basic Notions

We now formally introduce standard feedforward NNs. We stick to the terminology introduced in [PV18; PRV20]. Particularly, we denote by an *NN* a structured set of weights, whereas the associated function is called its *realization*.

- *NN:* Let $L, d = N_0, N_1, \dots, N_L \in \mathbb{N}$ be fixed. A family

$$\Phi = \big((\mathbf{A}_\ell, \mathbf{b}_\ell)\big)_{\ell=1}^{L}$$

  of matrix-vector tuples with $\mathbf{A}_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$, $\mathbf{b}_\ell \in \mathbb{R}^{N_\ell}$ is called *neural network* (NN).
  - The *input dimension* of Φ is denoted by d and its *output dimension* by N_L.
  - $L = L(\Phi)$ is the *number of layers* and $(N_0, \dots, N_L)$ is the *architecture* of Φ.
  - We refer to the entries of $\mathbf{A}_\ell, \mathbf{b}_\ell$ as the *weights* of Φ and call[7]

$$M(\Phi) := \sum_{\ell=1}^{L} \left(\|\mathbf{A}_\ell\|_0 + \|\mathbf{b}_\ell\|_0\right)$$

    its *number of (non-zero) weights*. Moreover, we denote by $\|\Phi\|_\infty$ the *maximum absolute value of all weights*.

- *Realization of an NN:* Additionally, fix an *activation function* $\varrho : \mathbb{R} \to \mathbb{R}$ and a set $\mathcal{X} \subset \mathbb{R}^d$. The *$\varrho$-realization of* Φ is the function

$$\begin{aligned} \mathrm{R}_\varrho^{\mathcal{X}}(\Phi) : \mathcal{X} &\to \mathbb{R}^{N_L}, \\ \mathbf{x} &\mapsto \mathbf{A}_L(\varrho(\mathbf{A}_{L-1}\varrho(\dots\varrho(\mathbf{A}_1\mathbf{x} + \mathbf{b}_1)\dots) + \mathbf{b}_{L-1})) + \mathbf{b}_L\,, \end{aligned}$$

  where ϱ is applied componentwise. Throughout this thesis, and for the sake of simplicity, we denote by *(ϱ-)NNs* both NNs and their ϱ-realizations as long as their is no ambiguity about their meaning.

- *Sets of NNs and their Realizations:* Let $S := (N_0, \dots, N_L)$. We call $\mathcal{NN}(S)$ the *set of NNs with architecture S* and define

$$\mathcal{RNN}_\varrho^{\mathcal{X}}(S) := \{\mathrm{R}_\varrho^{\mathcal{X}}(\Phi) \colon \Phi \in \mathcal{NN}(S)\}.$$

A wide range of activation functions has been used in practical applications. For the sake of brevity, we refer to Table 1 of [PRV20] and to Table 1 of [GR21] for comprehensive lists of activation functions. These functions will be frequently referenced.

[7] For a matrix $\mathbf{A}$, we denote by $\|\mathbf{A}\|_0$ its number of non-zero entries.

1.2 Parametric Partial Differential Equations: A Short Introduction

After having established the learning-theoretical background, we now turn to the introduction of *parametric* PDEs. We exclusively focus on the elliptic case. Parametric PDEs appear in various applications, such as uncertainty quantification, e.g., see [Sul15, Chapter 12], or optimal control problems, e.g., see [MA01]. A PDE is typically considered to be an idealized model of a physical phenomenon. Within such a context, one is often interested in the *solution* of the PDE under certain characterizing parameters.[8] These parameters are not necessarily fixed but allowed to vary within a certain range. Among them are boundary conditions, the shape of the underlying domain, the right-hand side, multiplicative coefficients, or the order of the PDE. Mathematically speaking, and in an idealized setting, one ends up with a family of well-posed operator equations of the form

$$B(y)u(y) = g(y), \tag{1.1}$$

where, for every y in the *parameter set* $\mathcal{Y} \subset \mathbb{R}^p$, $p \in \mathbb{N} \cup \{\infty\}$, $B(y) : X(y) \to X(y)^*$ is a continuous map with continuous inverse between a (possibly parameter-dependent) Banach space $X(y)$ and its topological dual space $X(y)^*$, $g(y) \in X(y)^*$ is the parametric right-hand side and $u(y) \in X(y)$ solves (1.1). Parametric PDEs are particularly relevant within a *multi-query context*. There, one needs to compute $u(y)$ for a large amount of parameters y and it is imperative to develop efficient solvers, which enable the quick computation of solutions for new input parameters. *Reduced order models* (ROMs) deal with the underlying parametric PDE very efficiently by significantly lowering the dimension of the problem (see Section 3.2.1 for a more detailed overview).

The goal of this part is to provide an overview of the mathematical background of parametric PDEs. In particular, we introduce the concrete setup of operator equations we consider throughout this thesis in Section 1.2.1. Afterwards, we describe the notions behind high-fidelity discretizations in Section 1.2.2 and the main objectives within multi-query applications in Section 1.2.3. Finally, we demonstrate in Section 1.2.4 that parametric problems yield a natural interpretation within statistical learning theory, thereby enabling us to build a bridge to NN-based techniques.

1.2.1 Setup of Operator Equations

In a classical modeling context (e.g., see [CD15]), one typically distinguishes between the case of *finite* parametrizations (where $p < \infty$) and *infinite* parametrizations (where $p = \infty$ and where the components of $(y_1, y_2, \dots) \in \mathcal{Y}$ fulfill certain decay conditions). Moreover, the parameters are either *deterministic* or *random*. In this thesis, we restrict ourselves to $p < \infty$. We consider parametric elliptic PDEs, in their weak formulation given by

$$b(y)(u(y), v) = g(y)(v), \qquad \text{for all } v \in \mathcal{H}. \tag{1.2}$$

Here, for every y in the *compact parameter set* $\mathcal{Y} \subset \mathbb{R}^p$, the *symmetric, elliptic, bounded*[9] *bilinear form* $b(y)$ is defined on $\mathcal{H} \times \mathcal{H}$ for a (parameter-independent) Hilbert space $\mathcal{H}$,

[8] An associated topic of interest we are not going to discuss in this thesis is the converse question of determining the underlying parameters from observed solutions, e.g., see [CDMN20].

[9] With ellipticity, boundedness constants independent of y.

and $g(y) \in \mathcal{H}^*$ is the *parametric right hand side*[10]. The Lax-Milgram lemma ensures that Equation (1.2) has, for every $y \in \mathcal{Y}$, a uniquely determined solution $u(y) \in \mathcal{H}$ [QMN16, Lemma 2.1]. We call

$$\mathbf{S} : \mathcal{Y} \to \mathcal{H}, \ \ y \mapsto u(y), \qquad \mathbf{S}(\mathcal{Y}) = \{u(y) : y \in \mathcal{Y}\}$$

the *parameter-to-solution map* and *solution manifold*, respectively. An important observation is that if the *forward maps* $y \mapsto b(y)(u, v)$ and $y \mapsto g(y)(v)$ are of a certain smoothness, then so is the solution map $\mathbf{S}$ (e.g., see [QMN16, Section 5.3.2] and the references therein). In particular, analytic parametrizations yield that $\mathbf{S}$ is analytic.

A Representative Example - The Parametric Poisson Equation: For illustrative purposes, and since this example plays an important role in our numerical analysis [GPRSK20], one could always bear the *parametric Poisson equation with homogeneous boundary conditions* in mind:

$$-\nabla \cdot (a(y)(\mathbf{x})\nabla u(y)(\mathbf{x})) = g(\mathbf{x}), \quad \text{on } K, \qquad \text{and} \qquad u = 0, \quad \text{on } \partial K.$$

Here, $K \subset \mathbb{R}^d$ is a Lipschitz domain, $g \in L^2(K)$ and, for every y in the compact parameter set $\mathcal{Y} \subset \mathbb{R}^p$, the function $a(y) \in L^\infty(K)$ is a *diffusion coefficient* with $0 < r \leq a(y) \leq R$ for absolute constants $r, R > 0$. The Hilbert space $\mathcal{H}$ is given by the Sobolev space $W_0^{1,2}(K)$ (see [Ada75, Section 3] for precise definitions) and we end up with the weak formulation

$$b(y)(u, v) := \int_K a(y)(\mathbf{x})\nabla u(\mathbf{x})\nabla v(\mathbf{x})\, d\mathbf{x} = \int_K g(\mathbf{x}) \cdot v(\mathbf{x})\, d\mathbf{x}, \qquad \text{for all } v \in W_0^{1,2}(K).$$

An important subclass of parametric PDEs are *affine* problems that, in the special case of the parametric Poisson equation, are of the form

$$a(y) = \sum_{i=1}^{p} \theta_i(y)\kappa_i, \tag{1.3}$$

for functions $\theta_i : \mathcal{Y} \to \mathbb{R}$ and $\kappa_i : K \to \mathbb{R}$. Depending on the application at hand, typical choices for the functions κ_i are (trigonometric) polynomials, B-splines, characteristic functions of sub-domains on K, wavelets, or, in the stochastic setting, functions from the Karhunen-Loeve basis (e.g., see [CD15, Section 1.2] or [BCM17, Section 4.3]). If the functions θ_i are analytic (in particular, this includes linearly parametrized problems), then the nonlinear solution map is analytic as well (e.g., see [CDS11; OR16]).

1.2.2 High-Fidelity Discretizations

A PDE, in its continuous formulation, is often considered to be an idealized version of a discrete physical system, typically derived from *finite-, but high-dimensional discretizations*. In particular, instead of the infinite-dimensional space $\mathcal{H}$ one considers a discretization[11] $U^{\mathrm{h}} = \mathrm{span}((\varphi_i)_{i=1}^D)$. The basis vectors $(\varphi_i)_{i=1}^D$ are, for instance, functions obtained by finite element discretizations of the underlying domain (e.g., see [Cia02]). Afterwards,

[10] With operator norm bounded independently of y.

[11] For simplicity of exposition, we do not distinguish between $\mathcal{H}$ and U^{h} any further. However, in a precise analysis, one needs to take the discretization error into account.

for $y \in \mathcal{Y}$, one determines the *Galerkin projection*[12] $u^{\mathrm{h}}(y) = \sum_{i=1}^{D} \mathbf{u}^{\mathrm{h}}(y)_i \varphi_i$, where the coefficient vector $\mathbf{u}^{\mathrm{h}}(y) \in \mathbb{R}^D$ can be determined by solving a high-dimensional system of linear equations. The map

$$\mathbf{S}^{\mathrm{dis}} : \mathcal{Y} \to \mathbb{R}^D,\ y \mapsto \mathbf{u}^{\mathrm{h}}(y) \tag{1.4}$$

is called the *discretized parameter-to-solution map* and will be our main object of interest.

1.2.3 Goal in Multi-Query Applications: Efficient Calculation of the Parameter-to-Solution Map

In applications where one needs to determine $u^{\mathrm{h}}(y)$ for a multitude of parameters and only has limited computational resources, it is imperative to develop computationally fast algorithms. Towards that goal, several types of ROMs have been established. These efficiently capture the intrinsic properties of the parameter-to-solution map $\mathbf{S}$ and of the solution manifold $\mathbf{S}(\mathcal{Y})$ (see Section 3.2.1 for a more detailed depiction). On an abstract level, the two main questions in this framework can be summarized as follows:

- Is it possible to prove the existence of specifically structured approximations $\widetilde{u}(y) \in \mathcal{H}$ of $u(y) \approx u^{\mathrm{h}}(y)$ or $\widetilde{\mathbf{u}}^{\mathrm{h}}(y) \in \mathbb{R}^D$ of $\mathbf{u}^{\mathrm{h}}(y)$ that have a comparatively low complexity? In the literature, this is referred to as *representational complexity* [BCD18]. Popular quantities for measuring the approximation quality are:
  - in the deterministic setting, the *uniform* error

$$\sup_{y\in\mathcal{Y}} \|u(y) - \widetilde{u}(y)\|_{\mathcal{H}} \quad \text{or} \quad \sup_{y\in\mathcal{Y}} \left\| u^{\mathrm{h}}(y) - \sum_{i=1}^{D} (\widetilde{\mathbf{u}}^{\mathrm{h}}(y))_i \varphi_i \right\|_{\mathcal{H}};$$

  - in the stochastic setting, for a probability measure $\sigma_{\mathcal{Y}}$ on $\mathcal{Y}$, and for $r \in (0, \infty)$, the *r-average* error

$$\left(\int_K \|u(y) - \widetilde{u}(y)\|_{\mathcal{H}}^r \, d\sigma_{\mathcal{Y}} \right)^{1/r} \quad \text{or} \quad \left(\int_K \left\| u^{\mathrm{h}}(y) - \sum_{i=1}^{D} (\widetilde{\mathbf{u}}^{\mathrm{h}}(y))_i \varphi_i \right\|_{\mathcal{H}}^r d\sigma_{\mathcal{Y}} \right)^{1/r}.$$

- The question of how to actually compute $\widetilde{u}(y)$ in practice is referred to as the *computational complexity* of the problem.

1.2.4 Interpretation of (Parametric) PDEs as a Learning Problem

One can naturally formulate the task of solving (parametric) PDEs within statistical learning theory. Several different interpretations within this context have been studied:

Fixed Parameters (not the main focus of this thesis): An important line of research that has been established in the literature is to keep the PDE model fixed and then interpret the solution of a *single* PDE defined on a domain $K \subset \mathbb{R}^d$ as a learning problem. This is *not* the main focus of this thesis and we only reiterate it in order to delineate it from

[12] Which is a quasi-best approximation of $u(y)$ in U^{h} by Cea's Lemma, e.g., see [QMN16, Lemma 2.2].

our setup. Nevertheless, we can interpret the bounds we derive in [GR21] within such a framework. Here, the input space is the physical domain $\mathcal{X} = K$, the output space is $\mathcal{Z} = \mathbb{R}^k$ for some $k \in \mathbb{N}$, and the target function is given by the solution $u : K \to \mathbb{R}^k$. This avenue of research in the context of deep NNs has been (numerically) studied, e.g., in [LLF98; RPK17a; EY18; SS18; YP18; LMMK19; BGJ20; Sam+20]. Here, in particular two different points of view on the loss function and the risk functional have been developed:

- The point of view taken in [COJSP17] has not been directly associated within the context of PDEs but has a natural interpretation therein. In the case of the Poisson equation with fixed diffusion coefficient $a(y) \equiv a$, one would end up with minimizing the risk functional

$$\mathbf{F} : W_0^{n,r}(K) \to \mathbb{R},\ f \mapsto \|f - u\|_{W^{n,r}(K)},$$

  where $r \in (0, \infty]$ and $n \in \mathbb{N}_0$. The training data in this setup are observed or computed evaluations of $(\mathbf{x}_i, u(\mathbf{x}_i))$ at spatial points $\mathbf{x}_i \in K$. This approach is feasible even if one does not have access to the concrete PDE model;

- Contrary to the setup above, the papers [RPK17a; EY18; LMMK19] require knowledge about the underlying PDE[13]. On the other hand, training NNs in such a framework may be more efficient, since it is under no circumstances required to compute actual solutions of the PDE during the training phase. The problem formulation incorporates the law prescribed by the PDE. For instance, [EY18], for fixed y, aims at solving the Poisson equation by minimizing the risk functional

$$\mathbf{F} : W_0^{1,2}(K)\ f \mapsto \int_K \left(\frac{1}{2} |a(\mathbf{x}) \nabla f(\mathbf{x})| - g(\mathbf{x}) f(\mathbf{x}) \right)\ d\mathbf{x}.$$

Variable Parameters (the main focus of this thesis): In thesis, we take a different point of view, which has already been proposed in [ESTW19] in the case of random parametrizations. Instead of using NNs as a surrogate for specific solutions of a PDE (with the typically low-dimensional spatial variable as its input), we aim at the approximation of the discretized parameter-to-solution-map $\mathbf{S}^{\text{dis}}$ which typically acts between the two high-dimensional spaces $\mathbb{R}^p$ and $\mathbb{R}^D$. Recalling the notions from Section 1.1.1, we particularly concentrate on the following interpretation:

- $\mathcal{X} = \mathcal{Y}$ is the input space and $\mathcal{Z} = \mathbb{R}^D$ is the target space;
- We distinguish between the following two setups:
  - In the deterministic setting, a popular loss function is given by

$$\mathcal{L} : \mathbb{R}^D \times \mathbb{R}^D \to [0, \infty),\ (\mathbf{c}^1, \mathbf{c}^2) \mapsto \left\| \sum_{i=1}^{D} \left(\mathbf{c}_i^1 - \mathbf{c}_i^2 \right)_i \varphi_i \right\|_{\mathcal{H}}$$

    and the associated risk functional is the uniform error

$$\mathbf{F} : (\mathbb{R}^D)^{\mathcal{Y}} \to [0, \infty),\ \mathbf{c} \mapsto \sup_{y \in \mathcal{Y}} \mathcal{L}(\mathbf{c}(y), \mathbf{u}^{\text{h}}(y));$$

[13] The paper [RPK17a] speaks about *physics-informed deep learning*.

– In the stochastic setting, we assume that $\sigma_{\mathcal{Y}}$ is a probability measure on $\mathcal{Y}$. For $r \in (0, \infty)$, consider the loss function

$$\mathcal{L} : \mathbb{R}^D \times \mathbb{R}^D \to [0, \infty), \ (\mathbf{c}^1, \mathbf{c}^2) \mapsto \left\| \sum_{i=1}^{D} \left(\mathbf{c}_i^1 - \mathbf{c}_i^2 \right)_i \varphi_i \right\|_{\mathcal{H}}^r$$

with associated risk functional induced by the r-average error

$$\mathbf{F} : (\mathbb{R}^D)^{\mathcal{Y}} \to [0, \infty), \ \mathbf{c} \mapsto \left(\int_{\mathcal{Y}} \mathcal{L}(\mathbf{c}(y), \mathbf{u}^{\mathrm{h}}(y)) \, d\sigma_{\mathcal{Y}}(y) \right)^{1/r}.$$

- The target function $\hat{f}$ that minimizes the risk functional (with $\mathbf{F}(\hat{f}) = 0$), both in the deterministic as well as in the stochastic setting, is given by $\hat{f} = \mathbf{u}^{\mathrm{h}}$.

In this regard, the learning problem induced by parametric elliptic PDEs is significantly different from other problems considered in statistical learning theory:

- the probability distribution $\sigma_{\mathcal{Y}}$ is often known a priori;
- the target function can be written down explicitly;
- the risk functional has at least one minimizer with zero risk.

1.3 Main Objectives and Findings of This Thesis

After having provided the necessary background, we are now returning to the two subjects described at the beginning and demonstrate how they are addressed in this thesis.

Main Objectives within Subject A: In view of the questions raised within Subject A, we provide in [PRV20] (see Appendix A) the first comprehensive study of the intrinsic structural properties of the hypothesis class $\mathcal{RNN}_{\varrho}^{K}(S)$ for all practically used activation functions ϱ from Table 1 of [PRV20] and a fixed NN architecture S. A summary and interpretation of the results can be found in Chapter 2. This study provides a new point of view on NNs and can be regarded as a research object that stands independently from the three fundamental pillars of learning theory that examine the approximation, generalization and training error. It turns out that the set $\mathcal{RNN}_{\varrho}^{K}(S)$ is:

non-convex, not closed in classical function spaces and that the realization map $\Phi \mapsto \mathrm{R}_{\varrho}^{K}(\Phi)$ is not inverse stable[14].

Although the properties described above are not dependent on a specific learning problem, they yield interpretations within learning theory. In particular:

- the non-convexity confirms the common observation that the training of NNs is a highly non-trivial optimization problem;
- from the non-closedness we conclude that for certain regression tasks, the weights of the approximating networks explode as the approximation error decreases;

[14] In the sense that for two functions in $\mathcal{RNN}_{\varrho}^{K}(S)$ which are close in some norm, there do not always exist corresponding weight configurations in the finite-dimensional space $\mathcal{NN}(S)$ that are close.

- from the missing inverse stability of the NN parametrization, we conclude that for specific regression problems, the training algorithm could converge either very slowly or not at all.

Overall, we establish that:

$$\mathcal{RNN}_{\varrho}^{K}(S) \textbf{ has unfavorable structural properties},$$

which, from a purely theoretical point of view, at least raise questions about their suitability as a hypothesis class in machine learning.

Of course, despite their disadvantageous intrinsic properties, NNs have shown remarkable success in applications and many of the phenomena described above can be avoided in practice. One reason lies in an abundance of well-established (albeit not always well-understood) techniques that suitably regularize the learning algorithm. Another reason lies in the flexibility of NNs with respect to their capabilities for the efficient approximation of large classes of target functions. This leads us to the

Main Objectives within Subject B: In Chapter 3, we examine the approximation capabilities of NNs for functions which occur in the context of (parametric) PDEs. We first theoretically examine how many parameters of an NN are sufficient to approximate specific function classes to achieve a certain accuracy. Here, we aim at identifying key properties of the target function that enable these rates. Concretely, we concentrate on the following setups:

- In [GR21] (see Appendix B), we provide lower and almost optimal upper bounds for function approximation in Sobolev spaces that hold for many activation functions. We assume that the weights of the approximating NNs are *encodable* in the sense that they can be represented by bit strings of moderate length. Here, the

  target functions are assumed to be sufficiently smooth.

  Concretely, if ε is the approximation accuracy, and M_ε is the uniform bound of parameters (in terms of the number of non-zero weights) needed to achieve approximation error ε, we show the expected relation

  $$M_\varepsilon \in \Theta(\varepsilon^{-d/(n-k)}), \quad \text{as } \varepsilon \to 0.$$

  Here, d is the input dimension, n is the Sobolev regularity (see [Ada75, Section 3] for precise definitions) of the target function, k denotes the degree of smoothness in the approximation norm, and Θ is the standard Landau symbol which expresses that M_ε grows asymptotically like $\varepsilon^{-d/(n-k)}$. We discuss these results and their proof strategy in Section 3.1. They form a continuation of a long line of research, showing that under the sole assumption of smoothness of the target function, the approximation capabilities of NNs are as limited as that of other approximation schemes.

- In [KPRS21] (see Appendix C), we turn our attention to parametric elliptic PDEs and the approximation of the discretized parameter-to-solution map $\mathbf{S}^{\text{dis}}$ in the deterministic, uniform setting. Of course, if this map between the two high-dimensional spaces $\mathbb{R}^p$ and $\mathbb{R}^D$ is sufficiently smooth (lets say, in $W^{n,\infty}$ for some $n \in \mathbb{N}_0$), then we obtain rates of order $\mathcal{O}(D\varepsilon^{-p/n})$ by using classical approximation results. However,

these rates may still be highly sub-optimal, since the term D is multiplied with the unfavorable scaling $\varepsilon^{-p/n}$. On the other hand, a key observation within classical ROM techniques is that, in many cases,

the solution manifold $\mathbf{S}(\mathcal{Y})$ is intrinsically low-dimensional.

In fact, if $\mathbf{S}(\mathcal{Y})$ is approximable by d-dimensional linear subspaces with $d \ll D$, then we are, for some $q, r \in \mathbb{N}$, able to derive NN approximation rates of the form

$$M_\varepsilon \in \mathcal{O}(Dd + d^q \log(1/\varepsilon)^r).$$

This is particularly pertinent in scenarios, where d scales more favorably in p than ε^{-p}. Additionally, the result is not confined to frameworks for which $\mathbf{S}^{\text{dis}}$ is sufficiently smooth but could, in principle, also be applied to other problems as long as the solution set has small dimension. The results of [KPRS21] are another instance of the commonly made observation that NNs are able to efficiently identify low-dimensional structures in high-dimensional data. The proof strategy (see Section 3.2) is based on an emulation of the online phase of reduced basis methods.

Due to their asymptotic nature, approximation-theoretical results of the above flavor have limited meaning in practice. Additionally, we have seen in Section 1.1.1 that other factors also contribute to the hardness of a learning problem. It is not clear whether it is numerically possible to observe approximation-theoretical effects.

- In [GPRSK20] (see Appendix D), we perform comprehensive experiments for the solution of the parametric Poisson equation. Here, as the parameter dimension p increases, our goal is to numerically estimate the expressiveness of fixed architecture NNs for the approximation of the discretized parameter-to-solution map $\mathbf{S}^{\text{dis}}$ for a large variety of parametrizations. Towards this goal,

  we numerically isolate the approximation error

  and

  derive approximation rates which do not suffer from the curse of dimensionality and depend on an intrinsic complexity of the underlying parametrization.

  We describe our method and observations in Section 3.3.

Overall, our expressivity results demonstrate the potential of using NNs in the task of solving parametric PDEs, thereby contrasting the results derived within Subject A.

2

The Hypothesis Class of Neural Networks with Fixed Architecture: Structural Properties and Their Consequences

Within the context of a vanilla deep learning algorithm, one fixes an NN architecture[1] $S = (d, N_1, \dots, N_{L-1}, 1)$, $L \in \mathbb{N}_{\geq 2}$, and an activation function ϱ a priori. Afterwards, for some $K \subset \mathbb{R}^d$, $\mathcal{Z} \subset \mathbb{R}$, one proceeds by solving the associated learning problem, which can either be formulated as

$$\operatorname{argmin}_{\Phi \in \mathcal{NN}(S)} \mathbf{F}\left(\mathrm{R}_\varrho^K(\Phi)\right), \tag{2.1}$$

or, sticking more closely to the notions of Section 1.1.1, as

$$\operatorname{argmin}_{f \in \mathcal{RNN}_\varrho^K(S)} \mathbf{F}(f), \tag{2.2}$$

where $\mathbf{F} : \mathcal{Z}^K \to \mathbb{R}$ is a risk functional prescribed by an application of interest. Although a minimizer Φ^* of Problem (2.1) yields a minimizer $\mathrm{R}_\varrho^K(\Phi^*)$ of Problem (2.2), the two optimization problems are fundamentally different from an analytical point of view:

- Problem (2.1) has a linear, finite-dimensional vector space as its hypothesis set, whereas the objective function is given by $\mathbf{F} \circ \mathrm{R}_\varrho^K$.
- Problem (2.2) has the highly complex hypothesis set $\mathcal{RNN}_\varrho^K(S)$ and the objective function is given by $\mathbf{F}$.

Among other characteristics, the structural properties of the hypothesis class determine the hardness of a learning problem. Whereas the structure of $\mathcal{NN}(S)$ is clear and many analyses in the past focused on the properties of $\mathbf{F} \circ \mathrm{R}_\varrho^K$, several aspects concerning the structure of $\mathcal{RNN}_\varrho^K(S)$ have either been unknown or only conjectured to hold. A prominent example is the presumption that $\mathcal{RNN}_\varrho^K(S)$ is not convex for many practically used activation functions, e.g., see [GBC16].

In [PRV20], we have examined the structural properties of the hypothesis set $\mathcal{RNN}_\varrho^K(S)$. This paper is the first comprehensive analysis of the intrinsic properties of this set for a wide range of activation functions. The goal of this section is a summary of the results of that paper, their interpretation as well as the underlying proof strategies. Moreover, we include a more detailed discussion of related works. We have divided the analysis to follow into three topics, which will be examined in the upcoming Sections 2.1–2.3:

[1] For $n \in \mathbb{N}_0$, we define $\mathbb{N}_{\geq n} := \{n, n+1, n+2, \dots\}$, and $\mathbb{N}_{\leq n}$ analogously.

(i) *Non-Convexity:* We examine to which extent the conjecture that $\mathcal{RNN}_\varrho^K(S)$ *is not convex* is true. Also, we establish further algebraic properties of $\mathcal{RNN}_\varrho^K(S)$.

(ii) *(Non)-Closedness:* Here, we additionally fix a function space $\mathcal{D} \supset \mathcal{RNN}_\varrho^K(S)$ and establish to which extent $\mathcal{RNN}_\varrho^K(S)$ is closed in $\mathcal{D}$. It turns out that closedness is a desirable property, since non-closedness potentially leads to the frequently observed occurrence of *divergent network weights*.

(iii) *Relationship between Problem* (2.1) *and Problem* (2.2)*:* Although a minimizer of (2.1) yields a minimizer of (2.2), it is not clear how the loss landscapes of both problems are associated to each other. We examine this topic in more detail below by studying the forward and inverse stability of the realization map. In particular, we pinpoint potential causes for *slow convergence* of the associated learning algorithm.

2.1 Non-Convexity of $\mathcal{RNN}_\varrho^K(S)$

One of the most complicated and, in many parts unresolved, questions within deep learning theory is a proper understanding of the underlying optimization procedure and its convergence properties. Many results have established that training NNs, even in very simple frameworks, is NP-complete in general (e.g., see [Jud87; BR89; BB02; GV15]). A prevalent reason for this fact is that the optimization problem is highly non-convex. Most analyses in the past have examined the degree of non-convexity of the objective function of Problem (2.1). As examples, we mention [BH88; Bac17; FB17; NH17; ZLW17; JGH18; MMN18; RV18; VBB18] and the references therein. Many of these works study the loss surface of the risk functional in different settings and inspect the (non-)existence of sub-optimal local minima or saddle points. Frequently, suitable convexification techniques are introduced and discussed. These results give some insight into the practically observed success of training algorithms by excluding obstructing phenomena that can occur during non-convex optimization. In several cases, after suitable regularization and initialization of the weights, convergence results to global minima of the loss surface of gradient-descent-related approaches can be derived, e.g., see [Jud87; BR89; BB02; GV15].

While the aforementioned works depend on the choice of the loss function, our paper [PRV20] takes the point of view of Problem (2.2) by studying the convexity of the hypothesis class $\mathcal{RNN}_\varrho^K(S)$ instead of the risk functional. This analysis can be performed independently from the underlying learning problem. Although the aforementioned works to some degree implicate that the set $\mathcal{RNN}_\varrho^K(S)$ needs to be non-convex, it is not clear to which extent (i.e., for which NN architectures and activation functions) this can be mathematically verified. In Theorem 2.1 of [PRV20], we show the following general result establishing non-convexity of $\mathcal{RNN}_\varrho^K(S)$ for all practically used activation functions:

Theorem 2.1 *Assume that* K has non-empty interior. *Moreover, let* ϱ *be* locally Lipschitz continuous *and assume that* ϱ *is* not a polynomial. *Then* $\mathcal{RNN}_\varrho^K(S)$ *is* ***not convex****.*

Remark 2.2 We are even in a position to show in Theorem 2.2 of [PRV20] that, for an arbitrary $r \geq 0$, the set $\left\{ f \in C(K) : \inf_{g \in \mathcal{RNN}_\varrho^K(S)} \|f - g\|_{C(K)} \leq r \right\}$, **is not convex**. ◇

We now proceed with depicting the overall proof strategy of Theorem 2.1 which also enables us to conclude several other algebraic properties of $\mathcal{RNN}_\varrho^K(S)$.

- We first establish (see Proposition C.1 of [PRV20]) that:

  $$\mathcal{RNN}_\varrho^K(S) \text{ is } \textbf{scaling- invariant}^2, \text{ hence } \textbf{star-shaped} \text{ with center } 0.$$

- Now recall the basic fact that *a star-shaped subset of a vector space is convex if and only if every element of this subset is a center*. Using that ϱ is *locally Lipschitz continuous*, we show in Lemma C.2, Corollary C.3 and Proposition C.4 of [PRV20] that:

  $$\mathcal{RNN}_\varrho^K(S) \quad \textbf{can only contain finitely many linearly independent centers}.$$

  From this it can be directly concluded (see Corollary C.5 of [PRV20]) that $\mathcal{RNN}_\varrho^K(S)$ is not convex if it encompasses infinitely many linearly independent functions.

- Finally, by using that ϱ is *not a polynomial* as well as the fact that $\mathcal{RNN}_\varrho^K(S)$ is a *translation-invariant*[3] subset of $C(K)$, a classical result from [AK64] shows that:

  $$\mathcal{RNN}_\varrho^K(S) \textbf{ has infinitely many linearly independent functions},$$

  from which the claim follows.

2.2 (Non)-Closedness of $\mathcal{RNN}_\varrho^K(S)$

An important part of past research connected to the topological properties of $\mathcal{RNN}_\varrho^K(S)$ has been done in connection with the examination of the *best approximation property*. If $\mathcal{D}$ is a normed function space with $\mathcal{RNN}_\varrho^K(S) \subset \mathcal{D}$, then $\mathcal{RNN}_\varrho^K(S) \subset \mathcal{D}$ is said to have the best approximation property, if, for every $g \in \mathcal{D}$, there exists $f^* = f^*(g) \in \mathcal{RNN}_\varrho^K(S)$ with

$$\|g - f^*\|_{\mathcal{D}} = \inf_{f \in \mathcal{RNN}_\varrho^K(S)} \|g - f\|_{\mathcal{D}}.$$

The question of the existence of best approximations is motivated by the associated question whether, for a target function g, there exists an NN from the underlying hypothesis class which has minimal distance to g and is the desired output of the learning algorithm.

A necessary condition for a set to have the best approximation property (e.g., see [GP90]) is the *closedness* of $\mathcal{RNN}_\varrho^K(S)$ in $\mathcal{D}$ which is the focus of this section. In this regard, [GP90, Proposition 4.1] shows that for $\varrho = \text{sigmoid}$ (with $\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$), the set $\mathcal{RNN}_\varrho^K((d, N_1, 1))$ is not closed in L^∞ and hence cannot have the best approximation property. The proof idea consists in showing that there exists a function that is the limit of ϱ-realizations of NNs but cannot be represented as a ϱ-realization of an NN with the same architecture.[4] Moreover, in [GP90, Page 171] it has been wrongly claimed that the non-closedness property can be proven for "every other non trivial choice of nonlinear [activation] function and for networks with more than one hidden layer". As we will see below, a result of such generality is not true. E.g., [KKV00] proves that Heaviside NNs *do* possess the best approximation property in $L^r(K)$, $r \in (1, \infty)$. Moreover, in Theorem 2.3.(iii) we demonstrate that ReLU-NNs *are* closed in $C(K)$.

[2] I.e., for all $\lambda \in \mathbb{R}$, $f \in \mathcal{RNN}_\varrho^K(S)$, also $\lambda f \in \mathcal{RNN}_\varrho^K(S)$.

[3] I.e., for every $f \in \mathcal{RNN}_\varrho^K(S)$, $t \in \mathbb{R}^d$, also $f(\cdot - t) \in \mathcal{RNN}_\varrho^K(S)$.

[4] In our proofs we use a similar strategy.

Why is closedness desirable? At this point one might argue that non-closedness is not severe, since in practice it may be sufficient to get arbitrarily close to $\inf_{f \in \mathcal{RNN}_\varrho^K(S)} \|g - f\|_\mathcal{D}$. However, as we have shown in Section 3.3 of [PRV20], in this case, the undesirable occurrence of

divergent network weights needs to occur.

To see this, we first note that it has been demonstrated in different setups[5] that, for all $C > 0$, the set of ϱ-NNs with bounded weights

$$\left\{ \mathrm{R}_\varrho^K(\Phi) : \Phi \in \mathcal{NN}(S),\ \|\Phi\|_\infty \leq C \right\} \textbf{ is compact in } \mathcal{D}.$$

Since compactness implies the best approximation property, we make the following two observations:

- *Observation in the non-discrete setting:* If $g \in \mathcal{D}$ does not have a best approximation in $\mathcal{RNN}_\varrho^K(S)$ and $(\mathrm{R}_\varrho^K(\Phi_n))_{n\in\mathbb{N}} \subset \mathcal{RNN}_\varrho^K(S)$ is *any* sequence with

$$\left\| g - \mathrm{R}_\varrho^K(\Phi_n) \right\|_\mathcal{D} \to \inf_{f \in \mathcal{RNN}_\varrho^K(S)} \|g - f\|_\mathcal{D}, \quad \text{as } n \to \infty,$$

  then $\|\Phi_n\|_\infty \to \infty$.

- *Observation in the discrete setting:* In practice, one usually aims at minimizing the empirical risk $\mathbf{F}_{\text{emp}}^n$ as introduced in Section 1.1.1. If the underlying loss function is the *empirical mean square error*[6] and based on classical results in statistical learning theory from [CS02], we derive in Proposition 3.6 of [PRV20]: If
  - σ is a (potentially unknown) probability Borel measure on $K \times \mathbb{R}$, σ_K is the marginal probability on K;
  - the regression function $\hat{f} \in L^2(K; d\sigma_K)$ of σ does *not* have a best approximation in $\mathcal{RNN}_\varrho^K(S)$,

  then for every random sequence $(\mathrm{R}_\varrho^K(\Phi_n))_{n\in\mathbb{N}}$ which approximately minimizes the empirical risk $(\mathbf{F}_{\text{emp}}^n)_{n\in\mathbb{N}}$, there holds $\|\Phi_n\|_\infty \to \infty$ in probability.

When is $\mathcal{RNN}_\varrho^K(S)$ actually closed? We proceed with examining the closedness of the set of ϱ-NNs with respect to Lebesgue norms, where ϱ is any of the functions from Table 1 of [PRV20]. We summarize the results of Theorem 3.1, Corollary 3.2, Theorem 3.3, Corollary 3.4 and Theorem 3.8 of [PRV20] in a non-technical way:

Theorem 2.3 *The following results hold:*

(i) Let ϱ be any of the activation functions from Table 1 of [PRV20]. *Moreover, let σ_K be* any finite Borel measure on K with uncountable support *and $r \in (0, \infty)$. Then:*

$$\mathcal{RNN}_\varrho^K(S) \textbf{ is not closed } \textit{in } L^r(K; d\sigma_K).$$

[5] In [Kur95] for continuous, bounded ϱ and $\mathcal{D} = L^r(K)$, $r \in [1, \infty]$; in Proposition 3.5 of [PRV20] for continuous ϱ and $\mathcal{D} = L^r(K; d\sigma)$ for finite Borel measures σ_K, $r \in (0, \infty)$ as well as for $\mathcal{D} = C(K)$.

[6] $\mathcal{L} : \mathbb{R}^2 \to \mathbb{R},\ \mathbf{x} \mapsto |\mathbf{x}_1 - \mathbf{x}_2|^2$

(ii) *Let ϱ be* any of the activation functions from Table 1 of [PRV20] except for the (leaky) ReLU. *Then:*

$$\mathcal{RNN}_{\varrho}^{K}(S) \textbf{ is not closed } \textit{in } C(K).$$

(iii) *Let $S = (d, N_1, 1)$. Then:*

$$\mathcal{RNN}_{\text{(leaky) ReLU}}^{K}(S) \textbf{ is closed } \textit{in } C(K).$$

Remark 2.4

- The original statement of (i), (ii) in [PRV20] has been stated in an abstract way by giving technical conditions on the activation functions that imply non-closedness. The proof strategy behind (i), (ii) consists in demonstrating the existence of functions in the $\mathcal{D}$-closure of $\mathcal{RNN}_{\varrho}^{K}(S)$ that cannot be exactly represented by NNs, since they do not belong to the same function class. For instance, in the case of (i), we approximate a discontinuous function by NNs with continuous activation functions (a property which all of the functions from Table 1 of [PRV20] fulfill).
- The proof idea of (iii) is to exploit the non-differentiability of the (leaky) ReLU combined with an analysis of the singularity hyper-planes of the resulting NNs. ◇

The recent paper [MKC20] also establishes non-closedness of NNs in Sobolev spaces for multiple activation functions. The proofs essentially rely on the same ideas as the corresponding statements in [PRV20]. Hence, even a stronger norm does not automatically enforce closedness of $\mathcal{RNN}_{\varrho}^{K}(S)$.

2.3 Relation between Problem (2.1) and Problem (2.2)

As we have already seen in the introduction of this section, a minimizer Φ^* of Problem (2.1) yields a minimizer of Problem (2.2). Additionally, we have established in Proposition 4.1 of [PRV20] the rather obvious fact that, if $\mathcal{D} = C(K)$ or $\mathcal{D} = L^r(K)$, for continuous ϱ, the realization map

$$\mathrm{R}_{\varrho}^{K} : \mathcal{NN}(S) \to \mathcal{D},\ \Phi \mapsto \mathrm{R}_{\varrho}^{K}(\Phi)$$

is *forward stable* (i.e., continuous). This implies that if two NNs are close in some norm on $\mathcal{NN}(S)$, also their associated realizations are close in $\mathcal{D}$. Hence, approximate minimizers of Problem (2.1) yield approximate minimizers of Problem (2.2). Nevertheless, it is not clear to which extent the converse is true:

If $f, g \in \mathcal{RNN}_{\varrho}^{K}(S)$ are two NN realizations that are close, do there always exist $\Phi, \widetilde{\Phi} \in \mathcal{NN}(S)$ that are close to each other and such that $f = \mathrm{R}_{\varrho}^{K}(\Phi)$ and $g = \mathrm{R}_{\varrho}^{K}(\widetilde{\Phi})$?

In particular, this would imply that for an approximate minimizer of Problem (2.2) there exists a parametrization that is an approximate minimizer of Problem (2.1). We examine this question by studying the *inverse stability* of the realization map.

Why is inverse stability of R_{ϱ}^{K} desirable? Assume we want to find an approximate solution $\Phi^* \in \mathcal{NN}(S)$ of (2.1). Then, an iterative training procedure may return $\widetilde{\Phi}^*$ for which $\mathbf{F}(\mathrm{R}_{\varrho}^{K}(\Phi^*)) \approx \mathbf{F}(\mathrm{R}_{\varrho}^{K}(\widetilde{\Phi}^*))$. However, if R_{ϱ}^{K} is *not* inverse stable, every weight configuration $\widetilde{\Phi}^*$ yielding this realization is potentially very far away from Φ^* in $\mathcal{NN}(S)$. In case the loss landscape of $\mathbf{F} \circ \mathrm{R}_{\varrho}^{K}$ has wide valleys,

this would result in very slow or no convergence of the underlying training algorithm.

Is R_ϱ^K inverse stable? The following simplified version of Theorem 4.2 and Corollary 4.3 of [PRV20] establishes that R_ϱ^K is *never* inverse stable in a neighborhood around 0 as long as $\mathcal{D} = C(K)$ or $\mathcal{D} = L^r(K)$:

Theorem 2.5 *Let ϱ be* Lipschitz continuous and not affine linear *and assume that K* has non-empty interior. *Then, there exists a sequence $(\Phi_n)_{n\in\mathbb{N}} \subset \mathcal{NN}(S)$ with:*

- $\left\|\mathrm{R}_\varrho^K(\Phi_n)\right\|_{C(K)} \to 0$;
- any *sequence* $(\Psi_n)_{n\in\mathbb{N}} \subset \mathcal{NN}(S)$ *with* $\mathrm{R}_\varrho^K(\Phi_n) = \mathrm{R}_\varrho^K(\Psi_n)$ *implies* $\|\Psi_n\|_\infty \to \infty$.

In particular:

$$\mathrm{R}_\varrho^K : \mathcal{NN}(S) \to \mathrm{Range}(\mathrm{R}_\varrho^K) \subset C(K) \text{ is } \textbf{\textit{not a quotient map}}^7.$$

In view of Theorem 2.5, one may ask whether inverse stability can be achieved under stricter restrictions, such as:

- a stronger topology than the one on $C(K)$. In such a framework, our proof strategy, relying on the construction of NNs with exploding Lipschitz constants and decreasing amplitude, fails;
- a restriction of the parameter space.

These types of restrictions lie at the heart of the more recent work [EBG19] which examines inverse stability for two-layer ReLU-NNs with respect to first-order Sobolev semi-norms. Despite focusing on a stronger norm, it is shown that inverse stability cannot be achieved in general. However, by suitably confining the parameter space, [EBG19, Theorem 3.3] establishes inverse stability of the restricted realization map. This observation may open up new research directions also in the $C(K)$-setting.

Let us close this part by a depiction of:

Further Related Works Examining the Parametrization of NNs: Another line of research that has gained attention in recent years is the study of *all* parametrizations which yield a *specific* realization. This question of *identifiability* stands in contrast to our analysis which focuses on NN realizations that are only *close* in some norm. The papers [PL20; VB20a; VB20b] (and the references therein) examine the question whether, up to certain symmetries or transformations, a particular output of the realization map uniquely determines the NN (i.e., its architecture, weights and biases) yielding that realization.

[7] A quotient map (see [Lee11]) is a continuous map $q : V \to W$ between two topological spaces V, W such that for any $A \subset W$ the equivalence A *is open in* $W \iff q^{-1}(A)$ *is open in* V holds.

3

Efficient Approximation of Solutions of Parametric PDEs by Neural Networks

We have seen in the last section that the structural properties of NNs are rather unfavorable. In contrast, their approximation capabilities are considered to be quite exceptional. Additionally, compared to many questions related to NN-based techniques, they are well-understood[1] and a plethora of past research has focused on estimates of the approximation error. However, it should be noted that in most applications one does not have *direct* access to the quantity $\left|\mathbf{F}(\hat{f}) - \mathbf{F}(\hat{f}_{\mathcal{M}})\right|$, since the probability distribution σ and the target function $\hat{f}$ are unknown. Instead, (potentially rough) surrogate quantities, such as norm estimates, are considered (see also the related discussion in [GRK20, Page 2]). As we have seen in Section 1.2.4, the application of parametric PDEs, the main focus of this section, is different in this regard. An important reason for the high expressivity of NNs lies in the fact that they are, in theory (disregarding algorithmic considerations), as efficient as a multitude of other schemes of approximation. A notable example is the *universal approximation theorem* (e.g., see [Cyb89; Pin99]), a version of which establishes that *every continuous function on a compact set $K \subset \mathbb{R}^d$ can be uniformly approximated arbitrarily well by shallow NNs with continuous, non-polynomial activation function*. Several proof strategies have been established in the literature. One idea consists of an emulation of polynomials[2] by NNs. In other words, under the assumption of unrestricted width[3], NNs are theoretically at least as powerful as polynomials for function approximation in $C(K)$. The universal approximation theorem does not give any bounds on the *complexity* of the approximating NNs. Subsequent works have measured the number of free parameters (in terms of the number of non-zero weights) in order to quantify the approximation capabilities of NNs. In the second part of this thesis, we continue this line of research by examining NN approximation rates for solutions of (parametric) PDEs. Although they are not adapted for this particular task, our results highlight the potential of NNs for their utilization within the area of parametric PDEs. We study the following aspects:

- In Section 3.2, we depict and discuss the results of [KPRS21] in Appendix C. We provide upper bounds on NN approximation of the discretized parameter-to-solution map $\mathbf{S}^{\text{dis}}$ from Equation (1.4). Contrary to the upper bounds from Section 3.1, which are solely based on the smoothness of the target function, we use the specific structure of $\mathbf{S}^{\text{dis}}$ in order to derive improved rates. The proof is based on the emulation of *reduced bases*, which are a ROM for solving parametric PDEs and a particular instance

[1] For a comprehensive survey on expressivity results of NNs we refer to [GRK20] and the references therein.
[2] These are dense in $C(K)$ by the Stone-Weierstraß Theorem, e.g., see [Rud76, Theorem 7.26].
[3] Denseness of NNs with uniformly bounded width cannot be achieved for practically used activation functions as is shown in Appendix C.3 of [PRV20].

of manifold approximation by linear subspaces. This result shows that using the nonlinear approximation scheme of NNs is essentially at least as powerful as *linear* approximation schemes for the solution of parametric PDEs.

The theoretical results from the papers [GR21; KPRS21] examine, for $\varepsilon > 0$, the sufficient/necessary number $M_\varepsilon \in \mathbb{N}$ of non-zero parameters for an NN such that it yields approximations with error ε with a particular emphasis on the dependence on the input dimension of the target function. From a practical point of view, this notion might be inadequate, since it is inherently asymptotic. Instead, it is often more relevant to ask the converse question: For a *fixed* NN architecture, is it possible to upper bound the resulting approximation error? This leads us to the last part of this section.

- In Section 3.3, we depict and discuss the results of [GPRSK20] in Appendix D. We perform systematic and comprehensive numerical experiments in which we learn the map $\mathbf{S}^{\text{dis}}$ associated to the parametric Poisson equation (see Section 1.2) for parametrizations of varying complexity. We aim at a numerical derivation of approximation rates, dependent on the underlying parametrization and for increasing parameter dimension p. We are particularly interested in a numerical verification of the theoretical observation that the complexity of the solution manifold is a main contributing factor to the hardness of the associated approximation problem.

3.1 Theoretical Approximation of Regular Functions in Higher-Order Smoothness Norms

The first approximation-theoretical results following the universal approximation theorem were mostly concerned with shallow NNs (e.g., see [Pin99] for an overview). In recent years, driven by their success in applications, deeper NNs and the potential gain in expressivity of deeper over shallow architectures have been examined more closely (for an overview of connected works, e.g., see [GRK20, Section 7] or [MLP16; PMRML17]).

Many works have focused on the following question:

For a function space $\mathcal{D}$ on a domain $K \subset \mathbb{R}^d$ and $\mathcal{C} \subset \mathcal{D}$, approximation accuracy $\varepsilon > 0$, and an activation function ϱ, is it possible to upper/lower bound $M_\varepsilon = M_\varepsilon(\mathcal{C}, \mathcal{D}, \varrho)$, $L_\varepsilon = L(\mathcal{C}, \mathcal{D}, \varrho) \in \mathbb{N}$ to obtain

$$\sup_{f \in \mathcal{C}} \inf_{\substack{\Phi \text{ NN with} \\ M(\Phi) \leq M_\varepsilon,\, L(\Phi) \leq L_\varepsilon}} \left\| f - \mathrm{R}_\varrho^K(\Phi) \right\|_{\mathcal{D}} \leq \varepsilon \, ?$$

Before we continue, we emphasize that approximation rates depend on

Conditions Regarding the Choice of Weights:

- Allowing for *arbitrary* weights might lead to better rates than approximation with constrained weights.
- *However:* Examining NNs with unrestricted weights is not feasible from a practical point of view, since computers are only able to store weights with a limited amount of bits. Hence, an assumption of the works [PV18; BGKP19; GPEB19; GKV20] and also the focus of this section, [GR21], is to only allow for encodable weights.

In particular, we require that the lengths of the bit strings of the weights of the approximating NNs grow moderately in ε. In such a framework, an NN typically is interpreted as an encoder within *rate-distortion theory* as introduced in [Don93]. Throughout the remainder of this section, and for the sake of simplicity, we do not provide the precise mathematical notions.

- Another assumption that has been studied in [Pin99; Yar17], is the *continuous* dependence of the parameters on the target function.

In the following, we particularly focus on results connected to classes $\mathcal{C}$ of sufficiently *smooth* functions. For an overview of works connected to *piecewise* smooth functions, we refer, e.g., to [GRK20, Section 5].

3.1.1 Lower Bounds for NN Approximation in General Smoothness Spaces

A variety of lower bounds that limit the capabilities of NN approximation have been established in the past. For classical function spaces, these bounds are subject to the *curse of dimensionality* (in the sense that M_ε grows exponentially in the input dimension as $\varepsilon \to 0$).

- The famous lower bound of [DHM89] essentially states that *any* scheme of approximation (not necessarily NNs) with *continuous* parameter dependence for the $L^r(K)$-approximation of functions in the unit ball of $W^{n,r}(K)$ requires at least

$$\Omega\left(\varepsilon^{-d/n}\right)$$

parameters.

- The slightly more optimistic bound $\Omega(\varepsilon^{-(d-1)/n})$ for function approximation by *shallow* NNs with *unrestricted* weights and *arbitrary* activation function has been obtained in [Mai10].

- In the case of NNs with *more than two layers* and *unrestricted* weights, a result of the above flavor does not hold anymore for *arbitrary* activation functions[4]. However, it is widely assumed that for practically used activation functions one cannot circumvent the curse of dimensionality. For instance, [Yar17; GKP20] show the lower bound

$$\Omega\left(\varepsilon^{-d/(2(n-k))}\right)$$

for approximation by ReLU-NNs in the case $\mathcal{C} = \{f \in W^{n,\infty}(K) : \|f\|_{W^{n,r}} \leq 1\}$, $\mathcal{D} = W^{k,r}(K)$, for $n \in \mathbb{N}_{\geq k+1}$. The proof relies on estimates of the VC-dimension [AB99].

We now turn to the case of NNs with weights that are *encodable by bit strings which moderately grow in the approximation accuracy*. Concretely, we consider weights that are encodable by $\mathcal{O}(\log(1/\varepsilon))$ bits. This information-theoretical viewpoint has first been taken in [BGKP19] and further considered in [PV18; GPEB19]. By imposing the restriction of encodable weights, in our paper [GR21] we derive lower bounds for *arbitrary* choices of activation functions. The main idea is to show that:

[4] For instance, [Pin99, Theorem 4] shows the existence of an exotic activation function ϱ such that ϱ-NNs with three layers and number of weights bounded linearly in d is dense in $C(K)$.

if the *minimax code length*[5] $T_\varepsilon(\mathcal{C}, \mathcal{D})$ can be bounded from below, then the same holds true for M_ε.

The minimax code-length essentially describes the necessary length of bit strings to encode every element in $\mathcal{C}$ up to accuracy (or distortion) ε. We have derived the following informal version of Theorem 3.5 of [GR21]:

Theorem 3.1 *Let ϱ be such that the resulting ϱ-NNs are a subset of $\mathcal{D}$. Assume that $T_\varepsilon(\mathcal{C}, \mathcal{D}) \in \Omega(\varepsilon^{-\gamma})$ for some $\gamma > 0$, as $\varepsilon \to 0$. Then, if we only allow for approximation by NNs* with weights that are encodable by $\mathcal{O}(\log(1/\varepsilon))$ bits, *we have*

$$M_\varepsilon(\mathcal{C}, \mathcal{D}, \varrho) \in \Omega(\varepsilon^{-\gamma} / \log(1/\varepsilon)).$$

Remark 3.2 This statement is a direct generalization of [BGKP19, Proposition 3.6] (where $\mathcal{D} = L^2(K)$) to arbitrary function spaces $\mathcal{D}$ and an immediate generalization of the proof strategy of [PV18, Theorem 4.3] (where $\mathcal{D} = L^r(K)$ and $\varrho(0) = 0$ was assumed). ◇

In particular, it follows from [GHJW18, Remark 5.10] that, if $\mathcal{C}$ is the unit ball of a function space $\widetilde{\mathcal{C}}$, then

$$T_\varepsilon(\mathcal{C}, \mathcal{D}) \geq H_\varepsilon(\widetilde{\mathcal{C}}, \mathcal{D}),$$

where $H_\varepsilon(\widetilde{\mathcal{C}}, \mathcal{D})$ is the *ε-entropy* of $\widetilde{\mathcal{C}}$ in $\mathcal{D}$ (e.g., see [ET96]). Since lower bounds on the ε-entropy are well-studied for a large variety of function spaces (e.g., see [Tri78; ET96]), we immediately obtain the following shortened version of Corollary 3.8 of [GR21] which establishes the following lower bound for NN approximation in Sobolev spaces:[6]

Corollary 3.3 *Let $n, k \in \mathbb{N}_0$ with $n \geq k+1$ and let $r \in (0, \infty]$. Let ϱ be such that the resulting ϱ-NNs are a subset of $\mathcal{D} := W^{k,r}(K)$. If:*

- *$\mathcal{C}$ is the* unit ball in $W^{n,r}(K)$*;*
- *as $\varepsilon \to 0$, we only allow for the approximation by NNs for which* every weight is encodable by $\mathcal{O}(\log(1/\varepsilon))$ bits,

we have

$$M_\varepsilon(\mathcal{C}, \mathcal{D}, \varrho) \in \Omega\left(\varepsilon^{-d/(n-k)} \Big/ \log(1/\varepsilon)\right). \tag{3.1}$$

3.1.2 Almost Optimal Upper Bounds for NN Approximation in Sobolev Spaces

In view of Equation (3.1), we now turn to deriving complementing (and almost tight) upper complexity bounds. We now exclusively concentrate on the case that $\mathcal{C}$ is the unit ball in $W^{n,r}(K)$ for $n \in \mathbb{N}$ and $\mathcal{D} = W^{k,r}(K)$ for $k \in \mathbb{N}_{\leq n-1}$. These bounds can be employed for deriving upper bounds on the approximation of the discretized parameter-to-solution map $\mathbf{S}^{\mathrm{dis}}$, provided it is sufficiently smooth. Additionally, this setup is relevant in the context of the approximation of solutions of PDEs with *fixed* parameters by NNs. These typically have a higher regularity than the elements from the solution space. Consider the

[5] We refer, e.g., to Definition 3.1 of [GR21] for the precise definition.

[6] Similar results can be obtained for Hölder spaces, Besov spaces, Triebel-Lizorkin spaces, and more.

Two Examples: Depending on the regularity of the right-hand side f, of the underlying domain $K \subset \mathbb{R}^d$, and of the diffusion coefficient a,

- typical weak solutions $u_a \in W^{1,2}(K)$ of the second-order *Poisson equation*

$$-\nabla(a(\mathbf{x})\nabla u(\mathbf{x})) = f(\mathbf{x}), \quad \text{on } K \quad + \text{boundary conditions},$$

  are even in $W^{n,2}(K)$, for some $n \geq 2$ (e.g., see [Dob10, Section 7]);

- typical weak solutions $u_a \in W^{2,2}(K)$ of the fourth-order *Dirichlet problem for the biharmonic operator* Δ^2 (e.g., see [Cia02])

$$-\Delta(a(\mathbf{x})\Delta u(\mathbf{x})) = f(\mathbf{x}), \quad \text{on } K \quad + \text{boundary conditions},$$

  are even in $W^{n,2}(K)$, for some $n \geq 3$ (e.g., see [Cia02, Section 6]).

Contrary to the lower bounds of the last section, complementing upper bounds cannot be derived in a straightforward manner for *arbitrary* choices of sufficiently smooth ϱ. In [GR21], we have identified simple conditions that allow for almost optimal approximation of Sobolev functions by ϱ-NNs with encodable weights. Our paper [GR21] can be seen as a generalization of the results and overall proof strategy of the upper bounds of [Yar17; GKP20] for ReLU-NNs to more general activation functions. [GR21] is the first paper that establishes approximation rates for a wide class of activation functions in higher-order smoothness norms. For related works, we refer to the end of this subsection. We now depict the overall idea behind the upper bounds of [Yar17; GKP20], in which the rates

$$M_\varepsilon \in \mathcal{O}(\varepsilon^{-d/(n-k)} \log(1/\varepsilon)), \qquad L_\varepsilon \in \mathcal{O}(\log(1/\varepsilon)), \tag{3.2}$$

have been derived for $k \in [0,1]$, $n \geq k+1$ and $r \in [1,\infty]$. When restricting to ReLU-NNs, approximation with respect to Sobolev norms of order $k \geq 2$ is excluded due to the ReLU's limited regularity. Hence, although ReLU-NNs have gained popularity in applications in recent years, from an approximation-theoretical point of view, they might not be as well suited for applications in which the smoothness of the target function plays a significant role. Before turning to smoother activation functions, let us shortly describe the proof idea behind the upper bounds of [Yar17; GKP20], since this constitutes the starting point of [GR21]. The strategy consists in

the emulation of localized Taylor polynomials

by ReLU-NNs.

(i) First, approximate f by localized polynomials of the form

$$\sum_{\mathrm{b}} \phi_{\mathrm{b}} \cdot p_{f,\mathrm{b}}. \tag{3.3}$$

Here,

- the functions ϕ_{b} are localizing, piecewise linear, compactly supported bump functions within patches $\mathrm{b} \subset K$;
- The functions $(\phi_{\mathrm{b}})_{\mathrm{b}}$ form a *partition of unity (PU)* on K, i.e., $\sum_{\mathrm{b}} \phi_{\mathrm{b}} = 1$;

  - the number of patches scales, for a grid-size $1/R$, $R \in \mathbb{N}$, like $\mathcal{O}(R^d)$;
  - the $p_{f,\mathrm{b}}$ are Taylor polynomials of degree $\leq n-1$ provided by the Bramble-Hilbert Lemma (e.g., see [BS08]). The polynomials approximate f up to a certain accuracy on the patch b. In order to achieve the overall approximation accuracy ε, choose $1/R = 1/R(\varepsilon) \in \mathcal{O}(\varepsilon^{1/(n-k)})$.

(ii) Second, approximate (3.3) by ReLU-NNs. Towards this goal, it is shown that:

  - the bump functions ϕ_{b} can be implemented by ReLU-NNs with constant size;
  - ReLU-NNs approximate the square function $x \mapsto x^2$ with $\mathcal{O}(\log(1/\varepsilon))$ weights and layers. This observation is the foundation for the approximation of the multiplication $(x,y) \mapsto xy = \frac{1}{2}((x+y)^2 - x^2 - y^2)$ by ReLU-NNs. Subsequently, the approximation of arbitrary polynomials as well as the approximation of the product $\phi_{\mathrm{b}} \cdot p_{f,\mathrm{b}}$ can be achieved by ReLU-NNs with $\mathcal{O}(\log(1/\varepsilon))$ weights and layers. Combining this result with the number of patches yields (3.2).

If we want to transfer the proof strategy of [Yar17; GKP20] based on localized polynomials to ϱ-NNs with other activation functions $\varrho \in W^{j,\infty}_{\mathrm{loc}}(\mathbb{R})$ for some $j \in \mathbb{N}$, we identify the following two basic ingredients, from which the approximation of localized polynomials can be deduced from in a straightforward manner (see Section 4.3 of [GR21] for more detailed explanations):

- **Ingredient 1:** Efficient approximation of the square function $x \mapsto x^2$. This is straightforward for all activation functions that are sufficiently smooth and have non-vanishing second derivative at *some* point $x_0 \in \mathbb{R}$.[7] This includes *all* activation functions listed in Table 1 of [GR21] except for the (leaky) ReLU. In particular, the resulting networks have $\mathcal{O}(1)$ non-zero weights and 2 layers. In this sense, they allow for more efficient approximations of the square function than ReLU-NNs. For a more detailed discussion, we refer to Section 4.2 of [GR21].

- **Ingredient 2:** Implementation of localizing bump functions ϕ_{b} by ϱ-NNs. We have not been aware of a strategy for general ϱ.

The main goal of Section 4.1 of [GR21] is to:

provide a general framework for the construction of bump functions by ϱ-NNs

that, together with the efficient approximation of polynomials, allows for almost optimal approximation rates in Sobolev norms. Contrary to ReLU-NNs, we face the following difficulties:

- The implementation of *compactly supported* bump functions is no longer possible in general. In particular, a bump function which is mainly concentrated on one patch does not vanish on other patches. Here, a fast decay outside of a certain patch is required in order to yield meaningful approximation rates;

- In general, we cannot hope for $\sum_{\mathrm{b}} \phi_{\mathrm{b}} = 1$. Instead, we aim at constructions of *approximate* PUs.

[7] For the corresponding statements, e.g., see, [RT18, Proof of Prop. 4.6] for L^∞-approximation or Proposition 4.7 of [GR21] for $W^{k,r}$-approximation.

We have identified the asymptotic behavior of ϱ to be the key factor in the construction of bump functions that vanish sufficiently fast outside of a certain patch. Concretely:

- on a high-level and following the analogous notion in [MM92], for $\tau \in \mathbb{N}_0$, we first require ϱ to be a *sigmoidal function of order* τ in the sense that there exist $a, b \in \mathbb{R}$ with $a \neq b$ and

$$\lim_{x \to -\infty} \frac{\varrho(x)}{x^\tau} = a, \qquad \lim_{x \to \infty} \frac{\varrho(x)}{x^\tau} = b.$$

- we require that ϱ and its weak derivatives approach the asymptotes with a specific speed (see Definition 4.2 of [GR21] for the case $\tau \in \{0, 1\}$). Particularly, we either require that the asymptotes are reached with *polynomial* speed, *exponential* speed or that the asymptotes are reached with *zero error*.

We then show in Lemma 4.5 of [GR21], for the case $\tau \in \{0, 1\}$, the existence of bump functions (implementable by ϱ-NNs with a constant number of weights) which:

either converge to a PU with polynomial speed, exponential speed,
or yield an exact PU.

Remark 3.4 All activation functions from Table 1 of [GR21] are either zero-order sigmoidal (if they are bounded) or first-order sigmoidal functions (if they are unbounded). An important example of a τ-order sigmoidal for $\tau \geq 2$ is given by the *rectified power unit* $\mathrm{RePU}_\tau = \mathrm{ReLU}^\tau$. τ-order sigmoidals allow for the construction of sufficiently concentrated bump functions (see Definition 4.4 for the case $\tau \in \{0, 1\}$ and Remark 4.6 of [GR21] for $\tau \geq 2$). All functions from Table 1 of [GR21] reach their asymptotes at least with polynomial speed. Several (such as, e.g., softsign or arctan) do not exhibit faster decay. Functions that approach their asymptotes with exponential speed are for instance the softplus or the ELU. Functions that reach their asymptotes with zero error are, e.g., (leaky) ReLU or RePU_τ. For a detailed exposition we refer to Table 1 of [GR21]. ◇

Depending on the behavior of the bump functions, we are able to show the following informal version of Proposition 4.8 and Theorem 4.9 of [GR21]:

Theorem 3.5 *We assume the following:*

- *Let* $j, \tau \in \mathbb{N}_0$, $k \in \{0, \dots, j\}$, $n \in \mathbb{N}_{\geq k+1}$, $r \in [1, \infty]$ *and* $\mu > 0$.
- $\varrho \in W^{j,\infty}_{\mathrm{loc}}(\mathbb{R})$ *is a* τ-order sigmoidal *that admits the construction of a* polynomial, exponential *or* exact PU.
- *There exists a neighborhood around* $x_0 \in \mathbb{R}$ *in which* ϱ *is* three times continuously differentiable *with* $\varrho''(x_0) \neq 0$.

Then, as $\varepsilon \to 0$, *for every* f *in the* unit ball of $W^{n,r}(K)$ *there exists an NN* $\Phi_{\varepsilon,f}$ *that has:*

- ***at most***

$$M_\varepsilon \in \begin{cases} \mathcal{O}(\varepsilon^{-d/(n-k)}), & \text{for } k \leq \tau, \text{if polynomial PU,} \\ \mathcal{O}(\varepsilon^{-d/(n-k-\mu\max\{0,k-\tau\})}), & \text{if exponential PU,} \\ \mathcal{O}(\varepsilon^{-d/(n-k)}), & \text{if exact PU,} \end{cases}$$

  non-zero weights *that are* ***encodable by*** $\mathcal{O}(\log(1/\varepsilon))$ ***bits;***

- ***at most*** $L_\varepsilon \in \mathcal{O}(1)$ ***layers,***

such that

$$\left\| f - \mathrm{R}_\varrho^K(\Phi_{f,\varepsilon}) \right\|_{W^{k,r}(K)} \leq \varepsilon.$$

Remark 3.6 The encodability of the weights can be concluded from the fact that only the weights in the last layer of the approximating NNs $\Phi_{f,\varepsilon}$ depend on f. These weights are rounded to a suitable mesh. The other weights can be directly encoded by an abstract coding scheme (see the proof strategy of Theorem 4.9 of [GR21]). This observation can intuitively be transferred to the emulation of other approximation schemes for which only the weights in the last layer of the approximating NNs depend on the target function. $\diamond$

Concluding, we can show:

- (up to a log factor) optimal rates for all activation functions from Table 1 of [GR21] in $W^{k,r}(K)$ for $k \leq \tau$. If the activation function ϱ only admits the construction of a polynomial PU, then we are not able to show optimal approximation rates for $k > \tau$. The reason for this is that, on a high level, the k-th order derivatives of the resulting bumps do not vanish fast enough outside of a certain patch. For a more detailed explanation we refer to Section 4.3 of [GR21];

- (almost) optimal rates for all activation functions that allow for an exact or exponential PU up to their highest order of smoothness;

- constant-depth approximations. This is a slight improvement over the corresponding statement for ReLU-NNs. L_ε only depends on n, d.

We conclude this part by depicting further works that are related to our setup.

Related Works I - Approximation of Dictionaries and Classes of Smooth Functions: Our results can be embedded into a long line of research that shows upper bounds for function approximation in smoothness spaces. These results, based on the emulation of a multitude of other approximation tools, demonstrate the expressive power of NNs. Thereby, many instances of approximation rates have been established that essentially show that NNs are theoretically at least or almost as powerful as other schemes of approximation. Similarly to our proof strategy, in these results one:

- approximates the target function by combinations of elements from a dictionary for which approximation rates are known;

- approximates the elements of the dictionary and combinations thereof by NNs.

Notable instances of dictionaries and simple functions that NNs with practically used activation functions are able to efficiently approximate are:

- polynomials (e.g., see [Mha96; Pin99; Yar17; PV18; RT18; GKP20; Sch20; GR21]);
- rational functions (e.g., see [Tel17]);
- wavelets (e.g., see [SCC18]), and more general affine systems (e.g., see [BGKP19]);
- B-splines (e.g., see [MM92; Mha93; Suz19]);
- finite elements (e.g., see [HLXZ20; OPS20]);
- radial basis functions (e.g., see [CLZ19]);
- sparse grids (e.g., see [MD19]);
- localizing functions (e.g., see [CS13; Yar17; PV18; BGKP19; Lin19; GKP20; GR21]).

Remark 3.7 We note that the constructions of the localizing functions of the works [CS13; Yar17; PV18; Lin19; GKP20] are only given for specific activation functions and do not provide a unifying framework for their construction:

- [BGKP19] proposes the implementation of compactly supported bump functions by activation functions that have compact support on the negative half-axis;
- [PV18] considers approximation in L^r, $r \in (0, \infty)$. In this setup, approximations of indicator functions by NNs with continuous activation function are shown. Dealing with higher-order smoothness spaces is not possible. Similarly, the localizing functions of [Lin19] rely on the mixed application of an indicator function and a sigmoidal function as the choice of activation functions;
- [Yar17; GKP20] consider bump functions implemented by ReLU-NNs. These are only in $W^{1,r}$ and hence not usable for approximation in higher-order Sobolev norms;
- [CS13] propose implementations of approximate bump functions by zero-order sigmoidals. The essential difference to our approach is that the influence of bump functions neighboring a certain patch does not decrease fast enough as $1/R \to 0$. $\diamond$

Based on the emulation of the dictionaries described above, many NN approximation results that prove upper bounds that are similar to ours have been studied. We only mention a few of these works:

- Further rates for function approximation in first-order Sobolev norms by ReLU-NNs, based on an emulation of finite element methods, have been derived in [OPS20]. Such a proof strategy might also work in the setting of higher-order smoothness spaces and for general activation functions;
- Constant-depth approximations of Hölder continuous functions with respect to L^r-norms, $r \in (0, \infty)$, by ReLU-NNs have been derived in [PV18]. The proof strategy is also based on the emulation of localized polynomials with indicator functions as the localizing functions. The constant depth can only be achieved due to $r < \infty$;

- Results based on the approximation of B-splines by ϱ-NNs where $\varrho = \mathrm{ReLU}$ or ϱ is a τ-order sigmoidal function for $\tau \geq 2$ can be found in [MM92; Mha93; Suz19]. These results provide approximation of Besov functions in L^r. Such a proof strategy might also work in the setting of higher-order smoothness spaces and for general activation functions. However, we are not aware of approximation rates for B-splines by NNs with general zero-or first-order sigmoidal activation function;

- [Mha96] shows approximation rates for Sobolev functions in L^r, $r \in [1, \infty]$. The result is based on an emulation of *global* polynomials. To achieve accuracy-independent depth, it is assumed that the activation function has a point with non-vanishing derivatives. This proof strategy can potentially be transferred to our setting of approximation in Sobolev norms. However, since we only assume that the activation function has a non-vanishing *second* derivative, the depth of the resulting NNs needs to grow with the approximation accuracy. This is caused by the unboundedness of the degree of the polynomial as the approximation accuracy increases.

Related Works II - Improved Rates by Further Structural Constraints: The aforementioned results suffer from the curse of dimensionality. This is due to the fact that the *sole* assumption on the target function is to be sufficiently smooth. The established lower bounds show that under such a weak constraint, no notable improvement is possible. However, significantly better approximation rates can be derived for functions with additional structural constraints. We refer to [GRK20, Section 6] for a more comprehensive overview. Here, we only shortly describe four lines of research in this direction:

- A famous example of a class of functions for which approximation rates can be derived that only weakly depend on the input dimension are functions with a

  finite Fourier moment,

  sometimes referred to as the *Barron class* (e.g., see [Bar93; Bar94; CPV20]).

- If the target function is

  compositional,

  the hierarchical arrangement of NN helps to gain significant expressive power over other approximation schemes (e.g., see [MLP16; PMRML17; MD19]).

- If the target function is the

  solution of a particular PDE,

  then it is also sometimes possible to break the curse of dimensionality (e.g., see [EHJ17; BBGJJ18; EGJS18; GHJW18; HJE18; JSW18; BEJ19; BGJ20; HJKN20]). A common strategy in this framework is the interpretation of the underlying PDE in a statistical context combined with an emulation of Monte Carlo methods by NNs.

- Another structural constraint leading to a massive gain in expressivity is the assumption that the target function is

  intrinsically low-dimensional.

This assumption which has, e.g., been considered in [PV18; SCC18; Sch19], mirrors the common observation that NNs are able to efficiently capture low-dimensional structures in high-dimensional data. The aforementioned works show that the approximation rate in this setting mainly depends on the dimension of the low-dimensional structure and only weakly on the ambient dimension.

3.2 Theoretical Approximation of the Discretized Parameter-to-Solution Map

We now turn to the approximation of the discretized parameter-to-solution map $\mathbf{S}^{\mathrm{dis}}$ of (1.4). Of course, as long as this map is sufficiently smooth, we can directly infer approximation rates of the type which have been discussed in the last section (see also Section 5 of [KPRS21] for a related discussion). However, as we have pointed out above, such results might lead to highly sub-optimal approximation rates, since they do not take the specific properties of the solution map into account. An important observation which lead to the emergence of classical ROMs in order to deal with solutions of parametric PDEs is that the *solution manifold* $\mathbf{S}(\mathcal{Y})$ *is low-dimensional* in many cases. We will exploit this property later on to derive approximation rates for $\mathbf{S}^{\mathrm{dis}}$ that are highly superior compared to classical rates. Our results heavily depend on the existence of small *reduced bases*, a linear approximation scheme. Since their theory plays an important role in our results and to give the reader a better overview of related, classical ROM techniques, we first proceed by a short survey of these techniques. Afterwards, we turn to NN-based approaches. We stick to the notation of Section 1.2.

3.2.1 Preliminaries: Classical Reduced Order Modeling Techniques

Linear ROM approaches aim at finding a linear subspace[8] $U^{\mathrm{red}} = \operatorname{span}\left((\psi_i)_{i=1}^N\right)$ of $\mathcal{H}$ with $N \ll D$ such that, for $y \in \mathcal{Y}$, there exist coefficient vectors $\mathbf{u}^{\mathrm{red}}(y) \in \mathbb{R}^N$ with

$$u(y) \approx \sum_{i=1}^{N} \mathbf{u}_i^{\mathrm{red}}(y)\psi_i.$$

This approach can be seen as a separation ansatz, in which the functions ψ_i only depend on the spatial variable, whereas the coefficient functions $\mathbf{u}_i^{\mathrm{red}}$ only depend on the parametric variable. In general, a small representational complexity (i.e., when N can be chosen to be comparatively small) does *not* necessarily imply small computational complexity. Due to this reason, most dimension-reducing algorithms have an *offline-online decomposition*.

- In the computationally expensive offline phase, the dimension of the problem is reduced by capturing the intrinsic properties of the solution manifold. In particular, the functions ψ_i are computed. In this phase, one has plenty of computational resources at ones disposal.
- The online phase, in which one only has access to few computational resources, uses the previously computed ROM to efficiently calculate new solutions.

[8] In the trivial case of a PDE with fixed parameters, i.e., when $\mathcal{Y}$ consists of one element, one can simply choose the space spanned by the Galerkin solution as the optimal reduced space.

A lower bound on the approximative capabilities of the space U^{red} is given by the *Kolmogorov N-width* $W_N(\mathbf{S}(\mathcal{Y}))$ (e.g., see [CD15]) which measures the best theoretically feasible uniform approximation of $\mathbf{S}(\mathcal{Y})$ by an *arbitrary*, at most N-dimensional subspace of $\mathcal{H}$ or U^{h}. A fast decay of $W_N(\mathbf{S}(\mathcal{Y}))$ indicates the existence of spaces U^{red} with small dimension, implying small representational complexity of $\mathbf{S}(\mathcal{Y})$. Standard estimates of $W_N(\mathbf{S}(\mathcal{Y}))$ often rely on the smoothness of the solution map $\mathbf{S}$. In case it is analytic, it is possible to expand $u(y)$ in a Taylor series of the form

$$u(y) = \sum_{0 \leq |\alpha| < \infty} y^{\alpha} \psi_{\alpha}, \tag{3.4}$$

with Taylor coefficients $\psi_{\alpha} \in \mathcal{H}$, [CDS11, Theorem 1.3]. Such an expansion gives rise to:

sparse polynomial decompositions (SPDs),

where the infinite series (3.4) is approximated by a truncated partial sum $\sum_{i=1}^{N} \mathbf{u}_i^{\mathrm{red}}(y)\psi_{\alpha_i} = \sum_{i=1}^{N} y^{\alpha_i}\psi_{\alpha_i}$. For problems with affine parametrizations (1.3), it is possible to choose $N \in \mathcal{O}(\log(1/\varepsilon)^p)$ to achieve uniform approximation error ε, e.g., see [OR16, Theorem 3.1] or [BC17, Equation 3.6]. This constitutes an upper bound on the Kolmogorov N-width which is the only available bound for affine elliptic problems with general functions[9] κ_i. The size of the resulting reduced space $\operatorname{span}((\psi_{\alpha_i})_{i=1}^{N})$ is subject to a mitigated curse of dimensionality. Apart from Taylor expansions, popular choices of SPDs in the stochastic setting are truncations of orthonormal *Legendre* polynomials [CD15].

Since for SPDs the structure of $\mathbf{u}^{\mathrm{red}}(y)$ is prescribed, the structure of the associated ψ_{α_i} is also predetermined. In practice, it remains to find a configuration $\psi_{\alpha_1}, \dots, \psi_{\alpha_N}$ such that the resulting polynomial approximates $u(y)$. There exists an abundance of numerical algorithms that identify the multi-indices $\alpha_1, \dots, \alpha_N$ and subsequently calculate $\psi_{\alpha_1}, \dots, \psi_{\alpha_N}$ by solving a high-fidelity equation. Notable examples are *interpolation algorithms* (e.g., see [CD15, Section 6]), *greedy algorithms* (e.g., see [CD15, Section 7.2]) or approaches based on *compressed sensing* (e.g., see [RS17; CDTW18; DTW19]). The computational complexity of these methods in the offline phase is manageable[10]. After having found a configuration $\psi_{\alpha_1}, \dots, \psi_{\alpha_N}$, computing a solution for a new parameter in the online phase is trivial, since it only requires the evaluation of the polynomials and the assembly of the resulting sum[11].

A more general approach than SPDs which has gained attention in recent years is the structural assumption that the coefficients $\mathbf{u}^{\mathrm{red}}(y)$ possess a (e.g., see [EPS17; BCD18])

hierarchical tensor decomposition.

Since in both approaches described above, one a priori prescribes the structure of the coefficient vectors $\mathbf{u}^{\mathrm{red}}(y)$, the flexibility in choosing the functions $(\psi_i)_{i=1}^{N}$ is limited. This might ultimately lead to the construction of a sub-optimal reduced space. In order to overcome this issue and if one is interested in finding almost optimal approximations with respect to the 2-average error in the stochastic setting, methods based on

[9] Sharper estimates, which completely overcome the curse of dimensionality, are only known in very special cases and for small parameter dimensions p, e.g., see [CD15, Section 4.4], [BC17, Sections 2 & 4], [BCM17, Section 4] or [BCD18, Section 6].

[10] For instance, the algorithms proposed in [CD15, Sections 6 & 7] require a number of operations which grow of low polynomial order in D, N, see [CD15, Equations 6.110 & 7.99].

[11] This results in $\mathcal{O}(ND)$ operations.

proper orthogonal decompositions (PODs)

have been developed. Here, the basis $(\psi_i)_{i=1}^N$ is constructed via an approximation of a *covariance operator* associated to the problem at hand (e.g., see [QMN16, Section 6] and the references therein). Another approach, which has gained a lot of attention in recent years and aims at approximations with respect to the *uniform* error, are

reduced basis methods (RBMs),

which we will discuss now in more detail (e.g., see [RHP08; CD15; Dah15; QMN16] for more comprehensive overviews). The basic idea is to find a configuration $y_1, \dots, y_N \in \mathcal{Y}$ such that the associated *snapshots* $\psi_1 \coloneqq u^{\mathrm{h}}(y_1), \dots, \psi_N \coloneqq u^{\mathrm{h}}(y_N) \in U^{\mathrm{h}}$ yield (almost) optimal approximations in the sense that

$$\sup_{y \in \mathcal{Y}} \inf_{v \in \operatorname{span}((u^{\mathrm{h}}(y_i))_{i=1}^N)} \|u(y) - v\|_{\mathcal{H}}$$

exhibits similar decay as $W_N(\mathbf{S}(\mathcal{Y}))$ for $N \to \infty$. In theory, [Bin+11, Theorem 4.1] establishes the existence of such bases in the case that $W_N(\mathbf{S}(\mathcal{Y}))$ either decays polynomially or exponentially. In practice, *(weak) greedy algorithms* are able compute such a basis (e.g., see [Bin+11; BMPPT12; DPW13; CD15; Dah15]). Concerning the cost of the offline-decomposition, the following observations can be made:

- Compared to the computation of SPDs, the offline phase of RBMs is very costly and based on a sufficiently dense sampling of the parameter set $\mathcal{Y} \subset \mathbb{R}^p$. The size of the sampling set is subject to the curse of dimensionality[12] as it grows exponentially in p. Moreover, for every sample, one needs to solve a D-dimensional Galerkin scheme. Hence, when choosing an algorithm, one carefully needs to weigh a (potentially small) gain in approximation accuracy against a (potentially massive) increase in computational complexity.

- After having found the reduced basis $(\psi_i)_{i=1}^N = \left(\sum_{j=1}^D \mathbf{V}_{ji}\varphi_j\right)_{i=1}^N$, for a transformation matrix $\mathbf{V} \in \mathbb{R}^{D \times N}$, in the online phase and for a new parameter y, one:
  1. assembles the associated stiffness matrix $\mathbf{B}^{\mathrm{red}}(y) \coloneqq \left(b(y)(\psi_j, \psi_i)\right)_{i,j=1}^N$ and discretized right-hand-side $\mathbf{g}^{\mathrm{red}}(y) \coloneqq (g(y)(\psi_i))_{i=1}^N$;
  2. solves the N-dimensional system of linear equations $\mathbf{B}^{\mathrm{red}}(y)\mathbf{u}^{\mathrm{red}}(y) = \mathbf{g}^{\mathrm{red}}(y)$ to compute the coefficient vector $\mathbf{u}^{\mathrm{red}}(y)$;
  3. recovers the approximative solution $u^{\mathrm{red}}(y) = \sum_{j=1}^D (\mathbf{V}\mathbf{u}^{\mathrm{red}}(y))\varphi_j$, expanded with respect to the high-fidelity basis.

  Compared to the online phase of SPDs, the online phase of RBMs is more costly, since one needs to assemble and solve a system of linear equations.

Typical results ensuring the efficiency of RBMs is the affine parameter-dependence (1.3). Many RB solvers for non-affine problems depend on a reduction step which substitutes

[12] This observation is only valid if one strives for convergence results that hold with certainty, as the recent paper [CDDN20] shows. If one only aims at an algorithm which outputs a reduced basis that has the required approximation accuracy *with high probability*, then it is sufficient to use a random sample set the size of which, asymptotically, does not depend on p.

the problem by an affine one (see [QMN16, Section 10] and the references therein for detailed explanations). Other approaches for dealing with non-affine equations do exist. E.g., we mention the study [BCDM17] which provides sparse polynomial approximation for parametric PDEs with *lognormal coefficients*.

As we have already mentioned above, the efficiency of *linear* ROMs is limited by the decay of the Kolmogorov N-width of $\mathbf{S}(\mathcal{Y})$. In contrast, *nonlinear* methods of approximation are often able to simultaneously provide low representational and computational complexity. For a comprehensive overview of general nonlinear approximation schemes we refer to [DeV98]. In the context of parametric PDEs, nonlinear techniques are the subject of [Bon+20]. Instead of examining approximation by one linear space, it establishes approximation properties for finite unions of affine spaces generated by (with respect to y) *local* polynomial chaos expansions. This method, referred to as

library approximation,

gives significant gains, both from a representational as well as from a computational point of view when compared to standard SPDs or RBMs.

Let us close this part with the following short comments:

Other Types of parametric PDEs: In this thesis, we solely focus on *uniformly elliptic, symmetric parametric PDEs* with small intrinsic dimension of $\mathbf{S}(\mathcal{Y})$. RBMs based on *Petrov-Galerkin schemes* for non-symmetric or not uniformly elliptic parametric PDEs have been developed, e.g., see [Dah15; QMN16]. Here, the ansatz and the test space are potentially different. Linear ROMs for linear, parabolic problems have been considered, e.g., in [CD15]. For certain nonlinear equations or optimal control problems, linear ROMs exist as well, e.g., see [QMN16, Sections 11 & 12]. However, linear techniques fail to work well for parametric PDEs where $W_N(\mathbf{S}(\mathcal{Y}))$ exhibits slow decay. A prominent example are linear transport equations, e.g., see [DPW14; OR16]. Here, the sub-polynomial decay of $W_N(\mathbf{S}(\mathcal{Y}))$ prohibits the construction of small subspaces U^{red} and only nonlinear schemes enable the development of sufficiently efficient algorithms.

Other Solution Techniques for Parametric PDEs: After solving the reduced problem in the online phase of RBMs, it is still required to transform the reduced solution into the high-dimensional space $\mathbb{R}^D$ by multiplying $\mathbf{V}$. Hence, the complexity of the online phase, additionally to the complexity of solving the low-dimensional system of linear equations, depends linearly on D. On the other hand, it is known that *multigrid methods* [Yse93] are able to compute the high-fidelity solution (e.g., induced by a full or sparse grid) with complexity $\mathcal{O}(D)$ and are therefore asymptotically optimal. Hence, in a multi-query context, one might come to the conclusion that it would be favorable to use the fully non-intrusive multigrid algorithm (which does not rely on an offline/online decomposition) for the computation of the high-fidelity solution associated to a new parameter y. However, from a practical point of view, it has been pointed out in [Peh13] that the online phase of an RBM is computationally less involved, since the recursive nature of the multigrid algorithm as well as the repeated need for the assembly of high-dimensional matrices hampers its speed. Nevertheless, the high cost of the offline phase of RBMs can only be compensated by the computationally less expensive online phase in the case that one needs to compute the solutions for a large variety of parameters and if $\mathbf{S}(\mathcal{Y})$ is small.

3.2.2 NN Approximation Rates Based on the Low-Dimensionality of $\mathbf{S}(\mathcal{Y})$

The aforementioned ROM techniques can be seen as a special instance of *dictionary learning*. We now turn to NNs as the choice of the dictionary. Deep learning algorithms are similar to ROM techniques, since they also admit an offline-online decomposition:

- the offline phase corresponds to the training of the NN;
- the online phase primarily consists in a feedforward pass.

The cost of the online phase of a deep learning algorithm is proportional to the resulting NN's number of non-zero weights and the size of its architecture.[13]

We now turn to an examination of the representational complexity of NNs for the solution of parametric PDEs. First of all, we have seen before that NNs are at least as expressive as as polynomials. This fact has been used in [SZ19] where NN-approximation rates for the solution map $\mathbf{S}$ with infinite parameters have been derived. [HSZ20] focuses on the case of finite parameters and analytic solution map. An approach based on dimension reduction by a POD can be found in [BHKS20]. In [KPRS21], we

emulate the online phase of RBMs

to obtain approximation rates for the map $\mathbf{S}^{\mathrm{dis}}$. Particularly, we provide a framework for the *solution of systems of linear equations with NNs*. Although $\mathbf{S}^{\mathrm{dis}}$ is a map between two potentially high-dimensional spaces ($\mathbb{R}^p \to \mathbb{R}^D$), our rates are only weakly dependent on the ambient dimensions (see also Section 5 of [KPRS21] for a more detailed discussion). The governing factor is the dimension of the solution manifold. We exclusively use the

existence of a small reduced basis, without knowing its concrete shape,

to derive bounds on the size of the approximating NNs which are superior to classical rates. Contrary to RBMs, we do not require the reduced basis to consist of snapshots of the solution manifold, but of *arbitrary* linear combinations of the high-fidelity basis vectors $(\varphi_i)_{i=1}^D$. Apart from this property, we only assume that the *forward maps* $y \mapsto \mathbf{B}^{\mathrm{red}}(y)$, $y \mapsto \mathbf{g}^{\mathrm{red}}(y)$ can be efficiently approximated by NNs $\Phi^{\mathbf{B}}, \Phi^{\mathbf{g}}$, respectively.[14] The following informal version of Theorem 4.3 of [KPRS21] is true:

Theorem 3.8 *Let $\varrho =$ (leaky) ReLU. As $\varepsilon \to 0$, there exist NNs $\Phi_\varepsilon^{\mathbf{u}}$ and $q \in \mathbb{N}$ with:*

- ***at most*** $\mathcal{O}(DN + N^3 \cdot \log^q(1/\varepsilon) + M(\Phi^{\mathbf{B}}) + M(\Phi^{\mathbf{g}}))$ ***non-zero weights;***
- ***at most*** $\mathcal{O}(\log(1/\varepsilon) + \max\{L(\Phi^{\mathbf{B}}), L(\Phi^{\mathbf{g}})\})$ ***layers,***

such that

$$\sup_{y\in\mathcal{Y}} \left\| u^{\mathrm{h}}(y) - \sum_{j=1}^{D} (\mathrm{R}_\varrho^{\mathcal{Y}}(\Phi_\varepsilon^{\mathbf{u}}))_j \varphi_j \right\|_{\mathcal{H}} \leq \varepsilon.$$

[13] It is straightforward to see that, for an NN Φ with architecture $(N_0, \dots, N_L)$, a forward pass requires $\mathcal{O}(M(\Phi) + \sum_{\ell=1}^{L-1} N_i)$ operations. The first term stems from the calculation of the involved affine-linear maps, whereas the second term comes from the application of the activation function.

[14] This is the case for many parametrizations. Particularly, for linearly parametrized problems one can choose $\Phi^{\mathbf{B}}, \Phi^{\mathbf{g}}$ to be one-layer NNs with $p + D^2$ non-zero weights (see Section 4.2.1of [KPRS21]).

Remark 3.9 If ϱ is another activation function than the (leaky) ReLU, similarly to the strategy in [GR21], one can derive slightly improved approximation rates. ◇

The proof strategy (see Section 1.2.2 of [KPRS21] for a detailed exposition) is based on an approximation of the following compositional map by NNs:

$$\mathbb{R}^p \supset \mathcal{Y} \ni y \mapsto \mathbf{V}\mathbf{u}^{\mathrm{red}}(y) = \mathbf{V}(\mathbf{B}^{\mathrm{red}}(y))^{-1}\mathbf{g}^{\mathrm{red}}(y) \in \mathbb{R}^D.$$

We have identified the efficient approximation of the matrix inversion map

$$\mathbb{R}^{N\times N} \ni \mathbf{A} \mapsto \mathbf{A}^{-1} = \sum_{k=0}^{\infty} (\mathbf{Id}_N - \mathbf{A})^k \in \mathbf{R}^{N\times N}$$

to be the crucial task in this regard. Hence, our approach can be seen as a special case of an operator inversion based on NNs. Towards this goal, we

emulate matrix polynomials

by NNs. Besides the efficient approximate matrix multiplication (see Proposition 3.7 of [KPRS21]), the proof of Theorem 3.8 of [KPRS21] regarding the approximation of the Neumann series heavily uses the intrinsic hierarchical structure of NNs to control the resulting number of non-zero weights and layers.

3.3 Numerical Approximation of the Discretized Parameter-to-Solution Map

Our results from the last section state that, from a theoretical point of view, the main governing quantity in the approximation of $\mathbf{S}^{\mathrm{dis}}$ is an intrinsic complexity of the solution manifold. However, although rates of the above type are the main focus of NN approximation theory, they have several drawbacks from a practical point of view (see also [AD20] and Section 1.1.1 of [GPRSK20] for more comprehensive discussions):

- The upper bounds are *pure existence* results. It is not clear to which extent deep learning algorithms are actually able to find a theoretically existing NN.
- The bounds are *asymptotic*, implying that for increasing accuracy the approximating NNs need to grow. The rates additionally include implicit constants which may depend very unfavorably on the problem at hand and the ambient dimension.
- The upper bounds are based on the emulation of a linear approximation scheme. Hence, they are very likely *sub-optimal*. This would also hamper the computational efficiency of the NNs in the online phase, since a forward pass would require a potentially unfavorable number of computations (comparable to the computation of a new solution in the online phase of RBMs).

In our comprehensive numerical study [GPRSK20], we aim at addressing these issues by providing systematic experiments in the case of the parametric Poisson equation. Our goals are two-fold:

- **Goal 1:** We have briefly discussed in the introduction that it is in general not possible to understand a statistical learning problem from a purely approximation-theoretical point of view, since the training algorithm itself or the potential overfitting of the training data heavily influence the outcome. In an experiment, we aim at:

numerically isolating the approximation-theoretical properties

of NNs by eliminating any obfuscating phenomena that happen during learning. Such an analysis could also be of independent interest.

- **Goal 2:** We consider a large class of parametrizations of different complexities of the parametric Poisson equation on the unit square $K = [0,1]^2$ with a first-order finite element discretization U^{h} as the solution space. For a fixed NN architecture with varying input dimension, we want to:

  determine approximation rates for increasing parameter dimension p that correspond to the intuitive complexity of the parametrization and do not suffer from the curse of dimensionality.

  Additionally, we hope that these results are more refined than the theoretical considerations from Section 3.2 for which concrete bounds are tied to concrete and potentially coarse upper bounds on $W_N(\mathbf{S}(\mathcal{Y}))$.

In contrast to the deterministic theoretical results of the last section, we consider random parameters distributed according to the uniform probability measure on $\mathcal{Y} = [a,b]^p$. The performance of the resulting NN is measured with respect to the *relative* 1-average error.[15] We shortly describe our test cases, method and outcome of the experiment.

Design of Test Cases: We consider four different test cases of parametric diffusion coefficients $a(y)$ which contain subcases of varying complexity (see Section 4.2 of [GPRSK20] for a detailed exposition):

- trigonometric polynomials with *affine* parameter dependence. Every function $a(y)$ is *analytic* with respect to the spatial variable. This case is divided into subcases of different complexity where a hyper-parameter controls the influence of higher-frequency components: the higher the influence of the high-frequency components, the harder the problem at hand;

- indicator functions on $[0,1]^2$ based on a checkerboard partition with *affine* parameter dependence. The functions $a(y)$ are *discontinuous* with respect to the spatial variable. This case is divided into subcases of different complexity where a shift hyper-parameter controls the ellipticity of the problem: a lower ellipticity constant increases the hardness of the problem;

- indicator functions of cookies on $[0,1]^2$. This case contains the *affine subcase* of cookies with fixed radii and the *non-affine subcase* of cookies with variable radii. In both cases, the functions $a(y)$ are *discontinuous*;

- clipped polynomials with *non-affine parameter dependence*. The functions $a(y)$ are only *continuous*, but not in $W^{k,\infty}(K)$ for $k \geq 2$.

Method: In order to isolate the approximation-theoretical properties, we keep our experiment as simple as possible. To be more precise, we:

[15] We use the relative error in order enforce comparability among different test cases.

- choose one feedforward NN architecture (see Appendix A.1 of [GPRSK20]) and one activation function (leaky ReLU) for all test cases. Only the input dimension is allowed to vary. In this sense, we ensure that the impact of the choice of the NN architecture on the training procedure is essentially equal over all test cases. Certainly, more refined NN architectures might lead to better performance;
- use a simple training algorithm (see Section 4.2.2 of [GPRSK20]) with few adjustable hyper-parameters. In our case, we employ batch gradient descent combined with Adam (e.g., see [KB15]) with the same hyper-parameters for all test cases. In particular, we keep the batch sizes and the number of epochs fixed.
- fix the size of the training set independently of the test case. Additionally, we ensure (see Appendix A.2 of [GPRSK20]) that different sizes of the training set do not have a decisive impact on the observed approximation rates;
- establish a posteriori (see Appendix A.3 of [GPRSK20]) that:
  - the training algorithm always converges by examining the training error;
  - no overfitting occurs by studying the gap between generalization and training error.

Observations: For all test cases, we observe (see Sections 4.3 and 4.4 of [GPRSK20]), in accordance with Goal 2,

approximation rates of the types $\mathcal{O}(1), \mathcal{O}(\log(p))$ **or** $\mathcal{O}(p^q)$, for some $q \in \mathbb{N}$, **as** $p \to \infty$.

These rates do not suffer from the curse of dimensionality and correspond to the intuitive hardness of the problem. Additionally, we cannot observe a fundamentally different behavior between affinely decomposed and non-affinely decomposed problems. Such an observation is particularly useful in comparison with RBMs for which an affine parameter decomposition is often crucial to allow for an offline-online decomposition.

Before we close this section, let us briefly comment on:

Related Numerical Approaches for the Solution of Parametric PDEs by NNs: The papers [HU18; WHR19] combine RBMs by POD for nonlinear problems with few parameters with NNs. Concretely, instead of learning the high-dimensional solution map $\mathbf{S}^{\text{dis}}$ directly, a reduced basis is computed in the offline phase. Afterwards, an NN is trained to learn the coefficients with respect to this reduced (instead of the high-fidelity) basis. In this regard, the offline phase might be more costly, since it relies on two highly complex optimization procedures. A similar approach has been employed in [DDP20]. Here, an RB solver computed in the offline phase is directly incorporated into the NN. In [KLY20], convolutional NNs are trained to find a Quantity-of-Interest stemming from the parametric Poisson equation in two dimensions. This approach is fundamentally different from our setup, since it takes a fully discretized diffusion coefficient, evaluated at specific spatial grid points as an input. The paper [TB18] is of a similar flavor. The inverse problem of identifying the parameters of a PDE from observed data has, e.g., been examined in [RPK17b; Rai18; BN19]. The paper [LC20] proposes the use of a convolutional auto-encoder as a ROM technique for the solution of parametric PDEs of low parameter dimension. The works [EHJ17; BBGJJ18] propose a framework for the solution of stochastic PDEs.

4

Conclusion and Outlook

In this thesis, we have examined several aspects of NNs. Reflecting upon our results, we have found disadvantageous structural properties which stand in contrast to approximation-theoretical considerations that demonstrate the potential of NNs for the application of parametric PDEs.

I. We have provided in Chapter 2 and within Subject A the first general and fundamental mathematical study of the intrinsic structure of the set of NNs with fixed architecture. Mirroring the commonly made observation that optimizing over NNs is a highly non-trivial task, we have shown that their algebraic and topological properties are inconvenient. In particular, we confirm that the set is not convex for all practically used activation functions and that its structure facilitates exploding weights or slow convergence in certain situations. Whereas the non-convexity is prevalent in general frameworks, the topological properties have been mostly examined in the context of function spaces. While the findings of [PRV20] provide some insight into the hardness of deep learning problems, a practitioner might find this setting to be overly artificial. Studying our main questions of interest with respect to discrete norms might make our analysis more meaningful in the context of real-world applications. Also, in practice one does not optimize over NNs with unrestricted parameters. Similarly to our approximation-theoretical considerations, it would be interesting how the algebraic and topological properties behave under constrained weight configurations. The compactness of NNs with uniformly bounded parameters and the inverse stability results proven in [EBG19] are indicators that in less arbitrary setups, the structure of NNs with fixed architecture seems to be more advantageous. This is also reflected by many works (see the associated references in Chapter 2) which alleviate the non-convexity of the learning algorithm or suitably regularize it to ensure stability.

II. In the second part of this thesis, we focused on the application of parametric PDEs.

    First of all, recalling the interpretation of parametric PDEs within statistical learning theory from Section 1.2.4, we can naturally regard this application in the framework of our structural inquiry. The continuous formulation of the problem shows that examining topological properties with respect to function spaces is relevant in such a context. In particular, when computing approximations of the parameter-to-solution map, one ideally aims at norm estimates. Here, it would be interesting to examine whether in this concrete setup the phenomena described in [PRV20] are bound to occur in theory.

    The main focus of Chapter 3 and Subject B was on the analysis of the approximation-theoretical power of NNs. The theoretical works [GR21; KPRS21] derive bounds on

the sizes of NNs for function approximation relevant in the context of (parametric) PDEs. Both works exploit different properties of the target function - its smoothness in [GR21], its intrinsic low-dimensionality in [KPRS21]. The contribution of both works becomes clear when focusing on their proof strategies.

- The lower and upper bounds of [GR21] generalize and unify several approaches introduced in previous works (see Section 3.1 for the related discussion). Our main contribution consists in providing an extensive framework for the construction of localized functions by NNs with general activation function. This enables the emulation of localized polynomials by NNs and the derivation of upper bounds for approximation of functions based on their smoothness in Sobolev norms. We believe that the plug-and-play approach presented in this work is transferable to almost every practically used activation function.
- The proof of the upper bounds of [KPRS21] essentially demonstrates how to solve systems of linear equations with NNs. This general ansatz can be employed in contexts different from the application of parametric PDEs. We believe that several improvements of our estimates are possible. In particular, our results are based on the naive emulation of Richardson iterations and more sophisticated schemes from numerical linear algebra might yield sharper bounds in specific contexts. E.g., one could examine (algebraic) multigrid methods [Yse93] or the solution of systems of linear equations induced by hierarchical matrices [Hac15]. The connection to parametric PDEs is made by combining the just described strategy with the intrinsic low-dimensionality of the solution manifold. We obtain rates for the approximation of the parameter-to-solution map that are primarily dependent on the dimension of the solution set instead of the ambient dimension. Similarly to [ESTW19; BGJ20], we think that it is feasible to obtain bounds on the generalization error in specific setups by using standard tools from statistical learning theory.

Our works are two instances of approximation-theoretical considerations that demonstrate the high flexibility of NNs in multiple contexts they were not originally designed for. However, despite having a solid mathematical foundation, their practical implications are often considered to be limited [AD20]. Our comprehensive numerical analysis [GPRSK20] can be seen as one of the first studies that aims at isolating the approximation error of NNs. On the one hand, our findings indicate that typical solution maps of parametric PDEs are indeed approximable without curse of dimensionality in the probabilistic setting. On the other hand, we believe that a study of this flavor shows high potential for its utilization within other areas of deep learning: by excluding clouding occurrences such as overfitting or the non-convergence of the training procedure, a systematic heuristic could indeed yield practically relevant estimates on the approximation error. This might help practitioners to obtain reasonable bounds on the number of free parameters needed to achieve a certain accuracy. Nonetheless, a lot of work needs to be done in this direction in order to make such an approach viable. Also, from a computational point of view, it is not clear whether NNs are indeed competitive against classical reduced order algorithms for the solution of parametric PDEs. Particularly the cost of the offline phase in order to achieve certain bounds on the training error needs to be studied in detail.

Bibliography

[Ada75] R. A. Adams. *Sobolev Spaces*. Academic Press, 1975, 320.

[AD20] B. Adcock and N. Dexter. „The gap between theory and practice in function approximation with deep neural networks“. Preprint [arXiv:2001.07523]. 2020.

[AK64] P. M. Anselone and J. Korevaar. „Translation invariant subspaces of finite dimension“. *Proc. Amer. Math. Soc.* 15 (1964), 747–752.

[AB99] M. Anthony and P. L. Bartlett. *Neural network learning: theoretical foundations*. Cambridge University Press, Cambridge, 1999, xiv+389.

[Bac17] F. Bach. „Breaking the curse of dimensionality with convex neural networks“. *J. Mach. Learn. Res.* 18.1 (2017), 629–681.

[BC17] M. Bachmayr and A. Cohen. „Kolmogorov widths and low-rank approximations of parametric elliptic PDEs“. *Math. Comp.* 86.304 (2017), 701–724.

[BCD18] M. Bachmayr, A. Cohen, and W. Dahmen. „Parametric PDEs: sparse or low-rank approximations?“ *IMA J. Numer. Anal.* 38.4 (2018), 1661–1708.

[BCM17] M. Bachmayr, A. Cohen, and G. Migliorati. „Sparse polynomial approximation of parametric elliptic PDEs. Part I: Affine coefficients“. *ESAIM Math. Model. Numer. Anal.* 51.1 (2017), 321–339.

[BCDM17] M. Bachmayr, A. Cohen, R. DeVore, and G. Migliorati. „Sparse polynomial approximation of parametric elliptic PDEs. Part II: lognormal coefficients“. *ESAIM: M2AN* 51.1 (2017), 341–363.

[BH88] P. Baldi and K. Hornik. „Neural Networks and Principal Component Analysis: Learning from Examples Without Local Minima“. *Neural Netw.* 2 (1988).

[Bar93] A. Barron. „Universal Approximation Bounds for Superpositions of a Sigmoidal Function“. *IEEE Trans. Inf. Theory* 39.3 (1993), 930–945.

[Bar94] A. Barron. „Approximation and Estimation Bounds for Artificial Neural Networks“. *Mach. Learn.* 14.1 (1994), 115–133.

[BB02] P. L. Bartlett and S. Ben-David. „Hardness results for neural network approximation problems“. *Theor. Comput. Sci.* 284.1 (2002), 53–66.

[BBGJJ18] C. Beck, S. Becker, P. Grohs, N. Jaafari, and A. Jentzen. „Solving stochastic differential equations and Kolmogorov equations by means of deep learning“. Preprint [arXiv:1806.00421]. 2018.

[BEJ19] C. Beck, W. E, and A. Jentzen. „Machine Learning Approximation Algorithms for High-Dimensional Fully Nonlinear Partial Differential Equations and Second-order Backward Stochastic Differential Equations“. *J. Nonlinear Sci.* 29 (2019), 1563–1619.

[BN19] J. Berg and K. Nyström. „Data-driven discovery of PDEs in complex datasets“. *J. Comput. Phys.* 384 (2019), 239–252.

[BGJ20] J. Berner, P. Grohs, and A. Jentzen. „Analysis of the Generalization Error: Empirical Risk Minimization over Deep Artificial Neural Networks Overcomes the Curse of Dimensionality in the Numerical Approximation of Black-Scholes Partial Differential Equations“. *SIAM J. Math. Data Sci.* 2.3 (2020), 631–657.

[BHKS20] K. Bhattacharya, B. Hosseini, N. B. Kovachki, and A. M. Stuart. „Model Reduction and Neural Networks for Parametric PDEs“. Preprint [arXiv: 2005.03180]. 2020.

[Bin+11] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk. „Convergence rates for greedy algorithms in reduced basis methods“. *SIAM J. Math. Anal.* 43.3 (2011), 1457–1472.

[BR89] A. Blum and R. Rivest. „Training a 3-node neural network is NP-complete“. *Adv. Neural Inf. Process. Syst.* 1989, 494–501.

[BGKP19] H. Bölcskei, P. Grohs, G. Kutyniok, and P. C. Petersen. „Optimal approximation with sparsely connected deep neural networks“. *SIAM J. Math. Data Sci.* 1.1 (2019), 8–45.

[Bon+20] A. Bonito, A. Cohen, R. Devore, D. Guignard, P. Jantsch, and G. Petrova. „Nonlinear Methods for Model Reduction“. Preprint [arXiv:2005.02565]. 2020.

[BS08] S. Brenner and R. Scott. *The Mathematical Theory of Finite Element Methods*. 3rd Edition. Vol. 15. Texts in Applied Mathematics. Springer, 2008, xviii+ 400.

[BMPPT12] A. Buffa, Y. Maday, A. T. Patera, C. Prud'homme, and G. Turinici. „A priori convergence of the greedy algorithm for the parametrized reduced basis method“. *ESAIM Math. Model. Numer. Anal.* 46.3 (2012), 595–603.

[CPV20] A. Caragea, P. C. Petersen, and F. Voigtlaender. „Neural network approximation and estimation of classifiers with classification boundary in a Barron class“. Preprint [arXiv:2011.09363]. 2020.

[CDTW18] A. Chkifa, N. Dexter, H. Tran, and C. G. Webster. „Polynomial approximation via compressed sensing of high-dimensional functions on lower sets“. *Math. Comp.* 87.311 (2018), 1415–1450.

[CLZ19] C. K. Chui, S.-B. Lin, and D.-X. Zhou. „Deep neural networks for rotation-invariance approximation and learning“. *Anal. and Appl.* 17.05 (2019), 737–772.

[Cia02] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. Society for Industrial and Applied Mathematics, 2002, xxiii+529.

[CDMN20] A. Cohen, W. Dahmen, O. Mula, and J. Nichols. „Nonlinear reduced models for state and parameter estimation“. Preprint [arXiv:2009.02687]. 2020.

[CD15] A. Cohen and R. DeVore. „Approximation of high-dimensional parametric PDEs“. *Acta Numer.* 24 (2015), 1–159.

[CDDN20] A. Cohen, W. Dahmen, R. DeVore, and J. Nichols. „Reduced basis greedy selection using random training sets“. *ESAIM Math. Model. Numer. Anal.* 54.5 (2020), 1509–1524.

[CDS11] A. Cohen, R. Devore, and C. Schwab. „Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDEs“. *Anal. Appl. (Singap.)* 09.01 (2011), 11–47.

[CS13] D. Costarelli and R. Spigler. „Approximation results for neural network operators activated by sigmoidal functions“. *Neural Netw.* 44 (2013), 101–106.

[CS02] F. Cucker and S. Smale. „On the mathematical foundations of learning“. *Bull. Am. Math. Soc.* 39 (2002), 1–49.

[CZ07] F. Cucker and D.-X. Zhou. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2007, 224.

[Cyb89] G. Cybenko. „Approximation by superpositions of a sigmoidal function“. *Math. Control Signal* 2.4 (1989), 303–314.

[COJSP17] W. M. Czarnecki, S. Osindero, M. Jaderberg, G. Swirszcz, and R. Pascanu. „Sobolev Training for Neural Networks“. *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017, 4278–4287.

[Dah15] W. Dahmen. „How to best sample a solution manifold?“ *Sampling theory, a renaissance*. Appl. Numer. Harmon. Anal. Birkhäuser/Springer, Cham, 2015, 403–435.

[DPW14] W. Dahmen, C. Plesken, and G. Welper. „Double greedy algorithms: Reduced basis methods for transport dominated problems“. *ESAIM Math. Model. Numer. Anal.* 48 (2014), 623–663.

[DDP20] N. Dal Santo, S. Deparis, and L. Pegolotti. „Data driven approximation of parametrized PDEs by reduced basis and neural networks“. *J. Comput. Phys.* 416 (2020), 109550.

[DPW13] R. DeVore, G. Petrova, and P. Wojtaszczyk. „Greedy algorithms for reduced bases in Banach spaces“. *Constr. Approx.* 37.3 (2013), 455–466.

[DeV98] R. DeVore. „Nonlinear approximation“. *Acta Numer.* 7 (1998), 51–150.

[DHM89] R. DeVore, R. Howard, and C. Micchelli. „Optimal nonlinear approximation“. *Manuscripta Math.* 63.4 (1989), 469–478.

[DTW19] N. Dexter, H. Tran, and C. Webster. „A mixed ℓ_1 regularization approach for sparse simultaneous approximation of parameterized PDEs“. *ESAIM Math. Model. Numer. Anal.* 53.6 (2019), 2025–2045.

[DNR14] J. Dobaczewski, W. Nazarewicz, and P.-G. Reinhard. „Error estimates of theoretical models: a guide“. *J. Phys. G Nucl. Partic.* 41.7 (2014), 074001.

[Dob10] M. Dobrowolski. *Applied functional analysis. Functional analysis, Sobolev spaces and elliptic differential equations. 2nd revised and extended ed*. Springer-Lehrbuch Masterclass. Berlin: Springer. xi, 284 p., 2010.

[Don93] D. Donoho. „Unconditional Bases are Optimal Bases for Data Compression and for Statistical Estimation“. *Appl. Comput. Harmon. Anal.* 1 (1993), 100–115.

[EMWW20] W. E, C. Ma, S. Wojtowytsch, and L. Wu. „Towards a Mathematical Understanding of Neural Network-Based Machine Learning: what we know and what we don't“. Preprint [arXiv:2009.10713]. 2020.

[EHJ17] W. E, J. Han, and A. Jentzen. „Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations“. *Commun. Math. Stat.* 5.4 (2017), 349–380.

[EY18] W. E and B. Yu. „The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems“. *Commun. Math. Stat.* 6.1 (2018), 1–12.

[ET96] D. E. Edmunds and H. Triebel. *Function Spaces, Entropy Numbers, Differential Operators*. Cambridge Tracts in Mathematics. Cambridge University Press, 1996, xii+252.

[EPS17] M. Eigel, M. Pfeffer, and R. Schneider. „Adaptive Stochastic Galerkin FEM with Hierarchical Tensor Representations“. *Numer. Math.* 136.3 (2017), 765–803.

[ESTW19] M. Eigel, R. Schneider, P. Trunschke, and S. Wolf. „Variational Monte Carlo-Bridging Concepts of Machine Learning and High Dimensional Partial Differential Equations“. *Adv. Comp. Math.* 45 (2019), 2503–2532.

[EGJS18] D. Elbrächter, P. Grohs, A. Jentzen, and C. Schwab. „DNN Expression Rate Analysis of High-dimensional PDEs: Application to Option Pricing“. Preprint [arXiv:1809.07669]. 2018.

[EBG19] D. Elbrächter, J. Berner, and P. Grohs. „How degenerate is the parametrization of neural networks with the ReLU activation function?“ *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8–14 December 2019, Vancouver, BC, Canada*. 2019, 7788–7799.

[FDPZP19] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri. „Using generative adversarial networks for improving classification effectiveness in credit card fraud detection“. *Inf. Sci.* 479 (2019), 448–455.

[FB17] C. D. Freeman and J. Bruna. „Topology and Geometry of Half-Rectified Network Optimization“. *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017.

[GPRSK20] M. Geist, P. Petersen, M. Raslan, R. Schneider, and G. Kutyniok. „Numerical Solution of the Parametric Diffusion Equation by Deep Neural Networks“. Preprint [arXiv:2004.12131]. 2020.

[GP90] F. Girosi and T. Poggio. „Networks and the best approximation property“. *Biol. Cybern.* 63.3 (1990), 169–176.

[GBC16] I. J. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[GV15] I. J. Goodfellow and O. Vinyals. „Qualitatively characterizing neural network optimization problems“. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015.

[GKV20] P. Grohs, A. Klotz, and F. Voigtlaender. „Phase Transitions in Rate Distortion Theory and Deep Learning “. Preprint [arXiv:2008.01011]. 2020.

[GHJW18] P. Grohs, F. Hornung, A. Jentzen, and P. von Wurstemberger. „A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations“. Preprint [arXiv:1809.02362]. 2018.

[GKMPR20] P. Grohs, G. Kutyniok, J. Ma, P. C. Petersen, and M. Raslan. „Anisotropic multiscale systems on bounded domains“. *Adv. Comput. Math.* 46.2 (2020).

[GPEB19] P. Grohs, D. Perekrestenko, D. Elbrächter, and H. Bölcskei. „Deep Neural Network Approximation Theory“. Preprint [arXiv:1901.02220]. 2019.

[GR21] I. Gühring and M. Raslan. „Approximation Rates for Neural Networks with Encodable Weights in Smoothness Spaces“. *Neural Netw.* 134 (2021), 107–130.

[GRK20] I. Gühring, M. Raslan, and G. Kutyniok. „Expressivity of Deep Neural Networks “. Preprint [arXiv:2007.04759]. 2020.

[GKP20] I. Gühring, G. Kutyniok, and P. Petersen. „Error bounds for approximations with deep ReLU neural networks in Ws,p norms“. *Anal. Appl. (Singapore)* 18.05 (2020), 803–859.

[Hac15] W. Hackbusch. *Hierarchical Matrices: Algorithms and Analysis*. 1st Edition. Springer Publishing Company, Incorporated, 2015, xxv+511.

[HJE18] J. Han, A. Jentzen, and W. E. „Solving high-dimensional partial differential equations using deep learning“. *Proc. Natl. Acad. Sci U.S.A.* 115.34 (2018), 8505–8510.

[HTF09] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. 2nd Edition. Springer Series in Statistics. New York, USA: Springer New York Inc., 2009, xxii+745.

[HLXZ20] J. He, L. Li, J. Xu, and C. Zheng. „ReLU Deep Neural Networks and Linear Finite Elements“. *J. Comput. Math.* 38.3 (2020), 502–527.

[HSZ20] L. Herrmann, C. Schwab, and J. Zech. *Exponential ReLU DNN expression of holomorphic maps in high dimension*. Tech. rep. 2019–35. Switzerland: Seminar for Applied Mathematics, ETH Zürich, 2020.

[HU18] J. S. Hesthaven and S. Ubbiali. „Non-intrusive reduced order modeling of nonlinear problems using neural networks“. *J. Comput. Phys.* 363 (2018), 55–78.

[HJKN20] M. Hutzenthaler, A. Jentzen, T. Kruse, and T. Nguyen. „A proof that rectified deep neural networks overcome the curse of dimensionality in the numerical approximation of semilinear heat equations“. *SN Partial Differ. Equ. Appl.* 1.10 (2020).

[Jac+19] J. Jackson, A. Kuriyama, A. Anton, A. Choi, J.-P. Fournier, A.-K. Geier, F. Jacquerioz, D. Kogan, C. Scholcoff, and R. Sun. „The Accuracy of Google Translate for Abstracting Data From Non–English-Language Trials for Systematic Reviews“. *Ann. Intern. Med.* 171.9 (2019), 677–679.

[JGH18] A. Jacot, F. Gabriel, and C. Hongler. „Neural tangent kernel: Convergence and generalization in neural networks“. *Adv. Neural Inf. Process. Syst.* 2018, 8571–8580.

[JSW18] A. Jentzen, D. Salimova, and T. Welti. „A proof that deep artificial neural networks overcome the curse of dimensionality in the numerical approximation of Kolmogorov partial differential equations with constant diffusion and nonlinear drift coefficients“. Preprint [arXiv:1809.07321]. 2018.

[JMFU17] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. „Deep Convolutional Neural Network for Inverse Problems in Imaging“. *IEEE Trans. Image Process.* 26.9 (2017), 4509–4522.

[Jud87] J. Judd. „Learning in networks is hard“. *Proc. of IEEE International Conference on Neural Networks, 1987*. Vol. 2. 1987, 685–692.

[KKV00] P. Kainen, V. Kurková, and A. Vogt. „Best approximation by Heaviside perceptron networks“. *Neural Netw.* 13.7 (2000), 695–697.

[KLY20] Y. Khoo, J. Lu, and L. Ying. „Solving parametric PDE problems with artificial neural networks“. *Eur. J. Appl. Math.* (2020), 1–15.

[KB15] D. P. Kingma and J. Ba. „Adam: A Method for Stochastic Optimization“. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015.

[KSH12] A. Krizhevsky, I. Sutskever, and G. Hinton. „ImageNet Classification with Deep Convolutional Neural Networks“. *Adv. Neural Inf. Process. Syst. 25*. Curran Associates, Inc., 2012, 1097–1105.

[Kur95] V. Kurkova. „Approximation of functions by perceptron networks with bounded number of hidden units“. *Neural Netw.* 8.5 (1995), 745–750.

[KPRS21] G. Kutyniok, P. C. Petersen, M. Raslan, and R. Schneider. „A Theoretical Analysis of Deep Neural Networks and Parametric PDEs“. *Constr. Approx.*, in Press (2021).

[LLF98] I. E. Lagaris, A. Likas, and D. I. Fotiadis. „Artificial neural networks for solving ordinary and partial differential equations“. *IEEE Trans. Neural Netw.* 9.5 (1998), 987–1000.

[LBH15] Y. LeCun, Y. Bengio, and G. Hinton. „Deep learning“. *Nature* 521.7553 (2015), 436–444.

[Lee11] J. M. Lee. *Introduction to topological manifolds*. 2nd Edition. Vol. 202. Graduate Texts in Mathematics. Springer, New York, 2011, xviii+433.

[LC20] K. Lee and K. Carlberg. „Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders“. *J. Comput. Phys.* 404 (2020), 108973.

[Lin19] S.-B. Lin. „Generalization and Expressivity for Deep Nets “. *IEEE T. Neur. Net. Lear.* 30.5 (2019), 1392–1406.

[LMMK19] L. Lu, X. Meng, Z. Mao, and G. Karniadakis. „DeepXDE: A deep learning library for solving differential equations“. Preprint [arXiv:1907.04502]. 2019.

[LS11] U. von Luxburg and B. Schölkopf. „Statistical Learning Theory: Models, Concepts, and Results“. *Handbook of the History of Logic, Vol. 10: Inductive Logic*. Vol. 10. Amsterdam, Netherlands: Elsevier North Holland, May 2011, 651–706.

[MKC20] S. Mahan, E. King, and A. Cloninger. „Nonclosedness of the Set of Neural Networks in Sobolev Space “. Preprint [arXiv:2007.11730]. 2020.

[Mai10] V. Maiorov. „Best approximation by ridge functions in L_p-spaces“. *Ukraïn. Mat. Zh.* 62.3 (2010), 396–408.

[MA01] H. Maurer and D. Augustin. „Sensitivity Analysis and Real-Time Control of Parametric Optimal Control Problems Using Boundary Value Methods“. *Online Optimization of Large Scale Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, 17–55.

[MP43] W. McCulloch and W. Pitts. „A Logical Calculus of Ideas Immanent in Nervous Activity“. *Bull. Math. Biophys.* 5 (1943), 115–133.

[MMN18] S. Mei, A. Montanari, and P.-M. Nguyen. „A mean field view of the landscape of two-layer neural networks“. *Proc. Natl. Acad. Sci. USA* 115.33 (2018), E7665–E7671.

[Mha93] H. N. Mhaskar. „Approximation properties of a multilayered feedforward artificial neural network“. *Adv. Comput. Math.* 1.1 (1993), 61–80.

[MM92] H. N. Mhaskar and C. Micchelli. „Approximation by superposition of sigmoidal and radial basis functions“. *Adv. Appl. Math.* 13.3 (1992), 350–373.

[Mha96] H. Mhaskar. „Neural Networks for Optimal Approximation of Smooth and Analytic Functions“. *Neural Comput.* 8.1 (1996), 164–177.

[MLP16] H. N. Mhaskar, Q. Liao, and T. Poggio. „Learning functions: when is deep better than shallow“. Preprint [arXiv:1603.00988]. 2016.

[MD19] H. Montanelli and Q. Du. „New error bounds for deep ReLU networks using sparse grids“. *SIAM J. Math. Data Sci.* 1.1 (2019), 78–92.

[NH17] Q. Nguyen and M. Hein. „The loss surface of deep and wide neural networks“. *Proc. of the 34th International Conference on Machine Learning-Volume 70*. 2017, 2603–2612.

[OR16] M. Ohlberger and S. Rave. „Reduced Basis Methods: Success, Limitations and Future Challenges“. *Proceedings of the Conference Algoritmy* (2016), 1–12.

[OPS20] J. A. A. Opschoor, P. C. Petersen, and C. Schwab. „Deep ReLU networks and high-order finite element methods“. *Anal. Appl. (Singap.)* (2020), 1–56.

[Peh13] B. Peherstorfer. „Model Order Reduction of Parametrized Systems with Sparse Grid Learning Techniques“. Dissertation. München: Technische Universität München, 2013.

[PRV19a] P. C. Petersen, M. Raslan, and F. Voigtlaender. „Unfavorable structural properties of the set of neural networks with fixed architecture“. *2019 13th International conference on Sampling Theory and Applications (SampTA)*. 2019, 1–4.

[PRV20] P. C. Petersen, M. Raslan, and F. Voigtlaender. „Topological Properties of the Set of Functions Generated by Neural Networks of Fixed Size“. *Found. of Comp. Math.* Online first (2020), 1–70.

[PRV19b] P. C. Petersen, M. Raslan, and F. Voigtlaender. „The structure of spaces of neural network functions“. *Wavelets and Sparsity XVIII*. Vol. 11138. SPIE, 2019, 144–151.

[PV18] P. C. Petersen and F. Voigtlaender. „Optimal approximation of piecewise smooth functions using deep ReLU neural networks“. *Neural Netw.* 108 (2018), 296–330.

[PR19] P. C. Petersen and M. Raslan. „Approximation properties of hybrid shearlet-wavelet frames for Sobolev spaces“. *Adv. Comput. Math.* 45.3 (2019), 1581–1606.

[PL20] M. Phuong and C. H. Lampert. „Functional vs. parametric equivalence of ReLU networks“. *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. 2020.

[Pin99] A. Pinkus. „Approximation theory of the MLP model in neural networks“. *Acta Numer.* 8 (1999), 143–195.

[PMRML17] T. Poggio, H. N. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao. „Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review“. *Int. J. Autom. Comput.* 14.5 (2017), 503–519.

[QMN16] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced basis methods for partial differential equations*. Vol. 92. Unitext. An introduction, La Matematica per il 3+2. Springer, Cham, 2016, xi+296.

[Rai18] M. Raissi. „Deep Hidden Physics Models: Deep Learning of Nonlinear Partial Differential Equations“. *J. Mach. Learn. Res.* 19.1 (2018), 932–955.

[RPK17a] M. Raissi, P. Perdikaris, and G. E. Karniadakis. „Physics informed deep learning (part I): Data-driven solutions of nonlinear partial differential equations“. Preprint [arXiv:1711.10561]. 2017.

[RPK17b] M. Raissi, P. Perdikaris, and G. E. Karniadakis. „Physics informed deep learning (part II): Data-driven discovery of nonlinear partial differential equations.“ Preprint [arXiv:1711.10561]. 2017.

[RS17] H. Rauhut and C. Schwab. „Compressive sensing Petrov-Galerkin approximation of high-dimensional parametric operator equations“. *Math. Comp.* 86.304 (2017), 661–700.

[RT18] D. Rolnick and M. Tegmark. „The power of deeper networks for expressing natural functions“. *International Conference on Learning Representations*. 2018.

[RV18] G. M. Rotskoff and E. Vanden-Eijnden. „Neural Networks as Interacting Particle Systems: Asymptotic Convexity of the Loss Landscape and Universal Scaling of the Approximation Error“. Preprint [arXiv:1805.00915]. 2018.

[RHP08] G. Rozza, D. B. P. Huynh, and A. T. Patera. „Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics“. *Arch. Comput. Methods Eng.* 15.3 (2008), 229–275.

[Rud76] W. Rudin. *Principles of Mathematical Analysis*. 3rd Edition. International Series in Pure and Applied Mathematics. McGraw-Hill Science/ Engineering/ Math, 1976, x+342.

[Sam+20] E. Samaniego, C. Anitescu, S. Goswami, V. Nguyen-Thanh, H. Guo, K. Hamdia, X. Zhuang, and T. Rabczuk. „An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications“. *Comput. Methods Appl. Mech. Eng.* 362 (2020), 112790.

[Sch15] J. Schmidhuber. „Deep learning in neural networks: An overview“. *Neural Netw.* 61 (2015), 85–117.

[Sch19] J. Schmidt-Hieber. „Deep ReLU network approximation of functions on a manifold“. Preprint [arXiv:1908.00695]. 2019.

[Sch20] J. Schmidt-Hieber. „Nonparametric regression using deep neural networks with ReLU activation function“. *Ann. Statist.* 48.4 (Aug. 2020), 1875–1897.

[SZ19] 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 (2019), 19–55.

[SCC18] U. Shaham, A. Cloninger, and R. R. Coifman. „Provable approximation properties for deep neural networks“. *Appl. Comput. Harmon. Anal.* 44.3 (2018), 537–557.

[SS18] J. Sirignano and K. Spiliopoulos. „DGM: A deep learning algorithm for solving partial differential equations“. *J. Comput. Phys.* 375 (2018), 1339–1364.

[Sul15] T. J. Sullivan. *Introduction to Uncertainty Quantification*. Vol. 63 of *Texts in Applied Mathematics*. Springer, 2015, xii + 342.

[Suz19] T. Suzuki. „Adaptivity of deep ReLU network for learning in Besov and mixed smooth Besov spaces: optimal rate and curse of dimensionality“. *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. 2019.

[Tel17] M. Telgarsky. „Neural networks and rational functions“. English (US). *34th International Conference on Machine Learning, ICML 2017*. Vol. 7. International Machine Learning Society (IMLS), 2017, 5195–5210.

[TPJR18] Y. Tian, K. Pei, S. Jana, and B. Ray. „DeepTest: Automated Testing of Deep-Neural-Network-Driven Autonomous Cars“. *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*. 2018, 303–314.

[Tri78] H. Triebel. *Interpolation Theory, Function Spaces, Differential Operators*. North-Holland Publishing Company, 1978, 528.

[TB18] R. Tripathy and I. Bilionis. „Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification“. *J. Comput. Phys.* 375 (2018).

[VBB18] L. Venturi, A. S. Bandeira, and J. Bruna. „Neural Networks with Finite Intrinsic Dimension have no Spurious Valleys“. Preprint [arXiv:1802.06384]. 2018.

[VB20a] V. Vlačić and H. Bölcskei. „Affine symmetries and neural network identifiability“. *Adv. Math.* (2020), 107485.

[VB20b] V. Vlačić and H. Bölcskei. „Neural Network Identifiability for a Family of Sigmoidal Nonlinearities “. Preprint [arXiv:1906.06994]. 2020.

[WHR19] Q. Wang, J. S. Hesthaven, and D. Ray. „Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem“. *J. Comput. Phys.* 384 (2019), 289–307.

[YP18] Y. Yang and P. Perdikaris. „Physics-informed deep generative models“. Preprint [arXiv:1812.03511]. 2018.

[Yar17] D. Yarotsky. „Error bounds for approximations with deep ReLU networks“. *Neural Netw.* 94 (2017), 103–114.

[Yse93] H. Yserentant. „Old and new convergence proofs for multigrid methods“. *Acta Numer.* 2 (1993), 285–326.

[ZLW17] Y. Zhang, P. Liang, and M. J. Wainwright. „Convexified convolutional neural networks“. *Proc. of the 34th International Conference on Machine Learning-Volume 70*. 2017, 4044–4053.

Foundations of Computational Mathematics
https://doi.org/10.1007/s10208-020-09461-0

FOUNDATIONS OF COMPUTATIONAL MATHEMATICS
The Journal of the Society for the Foundations of Computational Mathematics

Topological Properties of the Set of Functions Generated by Neural Networks of Fixed Size

Philipp Petersen[1] · Mones Raslan[2] · Felix Voigtlaender[3]

Received: 8 November 2018 / Revised: 23 January 2020 / Accepted: 30 March 2020

Abstract
We analyze the topological properties of the set of functions that can be implemented by neural networks of a fixed size. Surprisingly, this set has many undesirable properties. It is highly non-convex, except possibly for a few exotic activation functions. Moreover, the set is not closed with respect to L^p-norms, $0 < p < \infty$, for all practically used activation functions, and also not closed with respect to the L^∞-norm for all practically used activation functions except for the ReLU and the parametric ReLU. Finally, the function that maps a family of weights to the function computed by the associated network is not inverse stable for every practically used activation function. In other words, if f_1, f_2 are two functions realized by neural networks and if f_1, f_2 are close in the sense that $\|f_1 - f_2\|_{L^\infty} \leq \varepsilon$ for $\varepsilon > 0$, it is, regardless of the size of ε, usually not possible to find weights w_1, w_2 close together such that each f_i is realized by a neural network with weights w_i. Overall, our findings identify potential causes for issues in the training procedure of deep learning such as no guaranteed convergence, explosion of parameters, and slow convergence.

Keywords Neural networks · General topology · Learning · Convexity · Closedness

Mathematics Subject Classification 54H99 · 68T05 · 52A30

1 Introduction

Neural networks, introduced in 1943 by McCulloch and Pitts [49], are the basis of every modern machine learning algorithm based on *deep learning* [30,43,63]. The term *deep learning* describes a variety of methods that are based on the data-driven manipulation of the weights of a neural network. Since these methods perform spectac-

Communicated by Francis Bach.

All authors have contributed equally to this work.

Extended author information available on the last page of the article

Published online: 14 May 2020

ularly well in practice, they have become the state-of-the-art technology for a host of applications including image classification [36,41,65], speech recognition [22,34,69], game intelligence [64,66,70], and many more.

This success of deep learning has encouraged many scientists to pick up research in the area of neural networks after the field had gone dormant for decades. In particular, quite a few mathematicians have recently investigated the properties of different neural network architectures, hoping that this can explain the effectiveness of deep learning techniques. In this context, mathematical analysis has mainly been conducted in the context of statistical learning theory [20], where the overall success of a learning method is determined by the approximation properties of the underlying function class, the feasibility of optimizing over this class, and the generalization capabilities of the class, when only training with finitely many samples.

In the *approximation theoretical* part of deep learning research, one analyzes the expressiveness of deep neural network architectures. The universal approximation theorem [21,35,45] demonstrates that neural networks can approximate *any* continuous function, as long as one uses networks of increasing complexity for the approximation. If one is interested in approximating more specific function classes than the class of all continuous functions, then one can often quantify more precisely how large the networks have to be to achieve a given approximation accuracy for functions from the restricted class. Examples of such results are [7,14,51,52,57,71]. Some articles [18,54,57,62,72] study in particular in which sense *deep* networks have a superior expressiveness compared to their shallow counterparts, thereby partially explaining the efficiency of networks with many layers in deep learning.

Another line of research studies the training procedures employed in deep learning. Given a set of training samples, the training process is an *optimization problem* over the parameters of a neural network, where a loss function is minimized. The loss function is typically a nonlinear, non-convex function of the weights of the network, rendering the optimization of this function highly challenging [8,13,38]. Nonetheless, in applications, neural networks are often trained successfully through a variation of stochastic gradient descent. In this regard, the energy landscape of the problem was studied and found to allow convergence to a global optimum, if the problem is sufficiently overparametrized; see [1,16,27,56,67].

The third large area of mathematical research on deep neural networks is analyzing the so-called *generalization* error of deep learning. In the framework of statistical learning theory [20,53], the discrepancy between the *empirical* loss and the *expected* loss of a classifier is called the generalization error. Specific bounds for this error for the class of deep neural networks were analyzed for instance in [4,11], and in more specific settings for instance in [9,10].

In this work, we study neural networks from a different point of view. Specifically, we study the *structure of the set of functions implemented by neural networks of fixed size*. These sets are naturally (nonlinear) subspaces of classical function spaces like $L^p(\Omega)$ and $C(\Omega)$ for compact sets Ω.

Due to the size of the networks being fixed, our analysis is inherently non-asymptotic. Therefore, our viewpoint is fundamentally different from the analysis in the framework of statistical learning theory. Indeed, in approximation theory, the expressive power of networks growing in size is analyzed. In optimization, one stud-

ies the convergence properties of iterative algorithms—usually that of some form of stochastic gradient descent. Finally, when considering the generalization capabilities of deep neural networks, one mainly studies how and with which probability the empirical loss of a classifier converges to the expected loss, for increasing numbers of random training samples and depending on the sizes of the underlying networks.

Given this strong delineation to the classical fields, we will see that our point of view yields interpretable results describing phenomena in deep learning that are not directly explained by the classical approaches. We will describe these results and their interpretations in detail in Sects. 1.1–1.3.

We will use standard notation throughout most of the paper without explicitly introducing it. We do, however, collect a list of used symbols and notions in Appendix A. To not interrupt the flow of reading, we have deferred several auxiliary results to Appendix B and all proofs and related statements to Appendices C–E.

Before we continue, we formally introduce the notion of spaces of neural networks of fixed size.

Neural Networks of Fixed Size: Basic Terminology

To state our results, it will be necessary to distinguish between a *neural network* as a set of weights and the associated function implemented by the network, which we call its *realization*. To explain this distinction, let us fix numbers $L, N_0, N_1, \dots, N_L \in \mathbb{N}$. We say that a family $\Phi = \big((A_\ell, b_\ell)\big)_{\ell=1}^{L}$ of matrix-vector tuples of the form $A_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and $b_\ell \in \mathbb{R}^{N_\ell}$ is a **neural network**. We call $S{:=}(N_0, N_1, \dots, N_L)$ the **architecture** of Φ; furthermore $N(S){:=}\sum_{\ell=0}^{L} N_\ell$ is called the **number of neurons of** S and $L = L(S)$ is the **number of layers of** S. We call $d{:=}N_0$ the **input dimension** of Φ, and throughout this introduction we assume that the **output dimension** N_L of the networks is equal to one. For a given architecture S, we denote by $\mathcal{NN}(S)$ the **set of neural networks with architecture** S.

Defining the realization of such a network $\Phi = \big((A_\ell, b_\ell)\big)_{\ell=1}^{L}$ requires two additional ingredients: a so-called **activation function** $\varrho : \mathbb{R} \to \mathbb{R}$, and a domain of definition $\Omega \subset \mathbb{R}^{N_0}$. Given these, the **realization of the network** $\Phi = \big((A_\ell, b_\ell)\big)_{\ell=1}^{L}$ is the function

$$\mathrm{R}_\varrho^\Omega(\Phi) : \Omega \to \mathbb{R}, \quad x \mapsto x_L\,,$$

where x_L results from the following scheme:

$$\begin{aligned} &x_0{:=}x,\\ &x_\ell{:=}\varrho(A_\ell\, x_{\ell-1} + b_\ell), \quad \text{for } \ell = 1, \dots, L-1,\\ &x_L{:=}A_L\, x_{L-1} + b_L, \end{aligned}$$

and where ϱ acts componentwise; that is, $\varrho(x_1, \dots, x_d) := (\varrho(x_1), \dots, \varrho(x_d))$. In what follows, we study topological properties of sets of realizations of neural networks with a *fixed size*. Naturally, there are multiple conventions to specify the size of a network. We will study the **set of realizations of networks with a given architecture**

S **and activation function** ϱ; that is, the set $\mathcal{RNN}_{\varrho}^{\Omega}(S) := \{\mathrm{R}_{\varrho}^{\Omega}(\Phi) : \Phi \in \mathcal{NN}(S)\}$. In the context of machine learning, this point of view is natural, since one usually prescribes the network architecture, and during training only adapts the weights of the network.

Before we continue, let us note that the set $\mathcal{NN}(S)$ of all neural networks (that is, the network weights) with a fixed architecture forms a finite-dimensional vector space, which we equip with the norm

$$\|\Phi\|_{\mathcal{NN}(S)} := \|\Phi\|_{\mathrm{scaling}} + \max_{\ell=1,\dots,L} \|b_\ell\|_{\max}, \text{ for } \Phi = \big((A_\ell, b_\ell)\big)_{\ell=1}^{L} \in \mathcal{NN}(S),$$

where $\|\Phi\|_{\mathrm{scaling}} := \max_{\ell=1,\dots,L} \|A_\ell\|_{\max}$. If the specific architecture of Φ does not matter, we simply write $\|\Phi\|_{\mathrm{total}} := \|\Phi\|_{\mathcal{NN}(S)}$. In addition, if ϱ is continuous, we denote the **realization map** by

$$\mathrm{R}_{\varrho}^{\Omega} : \mathcal{NN}(S) \to C(\Omega; \mathbb{R}^{N_L}), \ \Phi \mapsto \mathrm{R}_{\varrho}^{\Omega}(\Phi). \tag{1.1}$$

While the activation function ϱ can in principle be chosen arbitrarily, a couple of particularly useful activation functions have been established in the literature. We proceed by listing some of the most common activation functions, a few of their properties, as well as references to articles using these functions in the context of deep learning. We note that all activation functions listed below are non-constant, monotonically increasing, *globally* Lipschitz continuous functions. This property is much stronger than the assumption of *local* Lipschitz continuity that we will require in many of our results. Furthermore, all functions listed below belong to the class $C^{\infty}(\mathbb{R} \setminus \{0\})$.

In the remainder of this introduction, we discuss our results concerning the topological properties of the sets of realizations of neural networks with fixed architecture and their interpretation in the context of deep learning. Then, we give an overview of related work. We note at this point that it is straightforward to generalize all of the results in this paper to neural networks for which one only prescribes the total number of neurons and layers and not the specific architecture.

For simplicity, we will always assume in the remainder of this introduction that $\Omega \subset \mathbb{R}^{N_0}$ is compact with non-empty interior.

1.1 Non-convexity of the Set of Realizations

We will show in Sect. 2 (Theorem 2.1) that, for a given architecture S, the set $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ *is not convex*, except possibly when the activation function is a polynomial, which is clearly not the case for any of the activation functions that are commonly used in practice.

In fact, for a large class of activation functions (including the ReLU and the standard sigmoid activation function), the set $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ turns out to be *highly non-convex* in the sense that for every $r \in [0, \infty)$, the set of functions having uniform distance at most r to any function in $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ is not convex. We prove this result in Theorem 2.2 and Remark 2.3.

This non-convexity is undesirable, since for non-convex sets, there do not necessarily exist well-defined projection operators onto them. In classical statistical learning theory [20], the property that the so-called regression function can be *uniquely* projected onto a convex (and compact) hypothesis space greatly simplifies the learning problem; see [20, Sect. 7]. Furthermore, in applications where the realization of a network— rather than its set of weights—is the quantity of interest (for example when a network is used as an Ansatz for the solution of a PDE, as in [24,42]), our results show that the Ansatz space is non-convex. This non-convexity is inconvenient if one aims for a convergence proof of the underlying optimization algorithm, since one cannot apply convexity-based fixed-point theorems. Concretely, if a neural network is optimized by stochastic gradient descent so as to satisfy a certain PDE, then it is interesting to see if there even exists a network so that the iteration stops. In other words, one might ask whether gradient descent on the set of neural networks (potentially with bounded weights) has a fixed point. If the space of neural networks were convex and compact, then the fixed-point theorem of Schauder would guarantee the existence of such a fixed point.

1.2 (Non-)Closedness of the Set of Realizations

For any fixed architecture S, we show in Sect. 3 (Theorem 3.1) that the set $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ *is not a closed subset of* $L^p(\mu)$ *for* $0 < p < \infty$, under very mild assumptions on the measure μ and the activation function ϱ. The assumptions concerning ϱ are satisfied for all activation functions used in practice.

For the case $p = \infty$, the situation is more involved: For all activation functions that are commonly used in practice—*except for the (parametric) ReLU*— the associated sets $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ *are non-closed also with respect to the uniform norm*; see Theorem 3.3. For the (parametric) ReLU, however, the question of closedness of the sets $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ remains mostly open. Nonetheless, in two special cases, we prove in Sect. 3.4 that the sets $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ are closed. In particular, for neural network architectures with two layers only, Theorem 3.8 establishes the *closedness of* $\mathcal{RNN}_{\varrho}^{\Omega}(S)$, *where* ϱ *is the (parametric) ReLU*.

A practical consequence of the observation of non-closedness can be identified with the help of the following argument that is made precise in Sect. 3.3: We show that the set

$$\left\{ \mathrm{R}_{\varrho}^{\Omega}(\Phi) \,:\, \Phi = ((A_\ell, b_\ell))_{\ell=1}^{L} \text{ has architecture } S \text{ with } \|A_\ell\| + \|b_\ell\| \leq C \right\}$$

of realizations of neural networks with a fixed architecture and all affine linear maps bounded in a suitable norm, *is* always closed. As a consequence, we observe the following phenomenon of exploding weights: If a function f is such that it does not have a best approximation in $\mathcal{RNN}_{\varrho}^{\Omega}(S)$, that is, if there does not exist $f^* \in \mathcal{RNN}_{\varrho}^{\Omega}(S)$ such that

$$\|f^* - f\|_{L^p(\mu)} = \tau_f := \inf_{g \in \mathcal{RNN}_{\varrho}^{\Omega}(S)} \|f - g\|_{L^p(\mu)},$$

then for any sequence of networks $(\Phi_n)_{n\in\mathbb{N}}$ with architecture S which satisfies $\|f - \mathrm{R}_\varrho^\Omega(\Phi_n)\|_{L^p(\mu)} \to \tau_f$, the weights of the networks Φ_n cannot remain uniformly bounded as $n \to \infty$. In words, if f does not have a best approximation in the set of neural networks of fixed size, then every sequence of realizations approximately minimizing the distance to f will have exploding weights. Since $\mathcal{RNN}_\varrho^\Omega(S)$ is not closed, there do exist functions f which do not have a best approximation in $\mathcal{RNN}_\varrho^\Omega(S)$.

Certainly, the presence of large coefficients will make the numerical optimization increasingly unstable. Thus, exploding weights in the sense described above are highly undesirable in practice.

The argument above discusses an approximation problem in an L^p-norm. In practice, one usually only minimizes "empirical norms". We will demonstrate in Proposition 3.6 that also in this situation, for increasing numbers of samples, the weights of the neural networks that minimize the empirical norms necessarily explode under certain assumptions. Note that the setup of having a fixed architecture and a potentially unbounded number of training samples is common in applications where neural networks are trained to solve partial differential equations. There, training samples are generated during the training process [25,42].

1.3 Failure of Inverse Stability of the Realization Map

As our final result, we study (in Sect. 4) the stability of the realization map $\mathrm{R}_\varrho^\Omega$ introduced in Eq. (1.1), which maps a family of weights to its realization. Even though this map will turn out to be continuous from the finite dimensional parameter space to $L^p(\Omega)$ for any $p \in (0, \infty]$, we will show that it is *not* inverse stable. In other words, for two realizations that are very close in the uniform norm, there do not always exist network weights associated with these realizations that have a small distance. In fact, Theorem 4.2 even shows that there exists a sequence of realizations of networks converging uniformly to 0, but such that every sequence of weights with these realizations is necessarily unbounded.

For both of these results—continuity and no inverse stability—we only need to assume that the activation function ϱ is Lipschitz continuous and not constant.

These properties of the realization map pinpoint a potential problem that can occur when training a neural network: Let us consider a regression problem, where a network is iteratively updated by a (stochastic) gradient descent algorithm trying to minimize a loss function. It is then possible that at some iterate the loss function exhibits a very small error, even though the associated network *parameters* have a large distance to the optimal parameters. This issue is especially severe since a small error term leads to small steps if gradient descent methods are used in the optimization. Consequently, convergence to the very distant optimal weights will be slow even if the energy landscape of the optimization problem happens to be free of spurious local minima.

1.4 Related Work

Structural properties

The aforementioned properties of non-convexity and non-closedness have, to some extent, been studied before. Classical results analyze the spaces of shallow neural networks, that is, of $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ for $S = (d, N_0, 1)$, so that $L = 2$. For such sets of shallow networks, a property that has been extensively studied is to what extent $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ has the *best approximation property*. Here, we say that $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ has the best approximation property, if for every function $f \in L^p(\Omega)$, $1 \leq p \leq \infty$, there exists a function $F(f) \in \mathcal{RNN}_{\varrho}^{\Omega}(S)$ such that $\|f - F(f)\|_{L^p} = \inf_{g \in \mathcal{RNN}_{\varrho}^{\Omega}(S)} \|f - g\|_{L^p}$. In [40] it was shown that even if a minimizer always exists, the map $f \mapsto F(f)$ is necessarily discontinuous. Furthermore, at least for the Heaviside activation function, there does exist a (non-unique) best approximation; see [39].

Additionally, [28, Proposition 4.1] demonstrates, for shallow networks as before, that for the logistic activation function $\varrho(x) = (1 + e^{-x})^{-1}$, the set $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ does not have the best approximation property in $C(\Omega)$. In the proof of this statement, it was also shown that $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ is not closed. Furthermore, it is claimed that this result should hold for every nonlinear activation function. The previously mentioned result of [39] and Theorem 3.8 below disprove this conjecture for the Heaviside and ReLU activation functions, respectively.

Other notions of (non-)convexity

In deep learning, one chooses a loss function $\mathcal{L} : C(\Omega) \to [0, \infty)$, which is then minimized over the set of neural networks $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ with fixed architecture S. A typical loss function is the empirical square loss, that is,

$$E_N(f) := \frac{1}{N} \sum_{i=1}^{N} |f(x_i) - y_i|^2,$$

where $(x_i, y_i)_{i=1}^{N} \subset \Omega \times \mathbb{R}$, $N \in \mathbb{N}$. In practice, one solves the minimization problem *over the weights of the network*; that is, one attempts to minimize the function $\mathcal{L} \circ \mathrm{R}_{\varrho}^{\Omega} : \mathcal{NN}(S) \to [0, \infty)$. In this context, to assess the hardness of this optimization problem, one studies whether $\mathcal{L} \circ \mathrm{R}_{\varrho}^{\Omega}$ is convex, the degree to which it is non-convex, and if one can find remedies to alleviate the problem of non-convexity, see for instance [5,6,27,37,50,56,59,67,73].

It is important to emphasize that this notion of non-convexity describes properties *of the loss function*, in contrast to the non-convexity *of the sets of functions* that we analyze in this work.

2 Non-convexity of the Set of Realizations

In this section, we analyze the convexity of the set of all neural network realizations. In particular, we will show that this set is highly non-convex for all practically

used activation functions listed in Table 1. First, we examine the convexity of the set $\mathcal{RNN}_{\varrho}^{\Omega}(S)$:

Theorem 2.1 *Let $S = (d, N_1, \dots, N_L)$ be a neural network architecture with $L \in \mathbb{N}_{\geq 2}$ and let $\Omega \subset \mathbb{R}^d$ with non-empty interior. Moreover, let $\varrho : \mathbb{R} \to \mathbb{R}$ be locally Lipschitz continuous.*

If $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ is convex, then ϱ is a polynomial.

Remark (1) It is easy to see that all of the activation functions in Table 1 are locally Lipschitz continuous, and that none of them is a polynomial. Thus, the associated sets of realizations are never convex.

(2) In the case where ϱ is a polynomial, the set $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ might or might not be convex. Indeed, if $S = (1, N, 1)$ and $\varrho(x) = x^m$, then it is not hard to see that $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ is convex if and only if $N \geq m$.

Proof The detailed proof of Theorem 2.1 is the subject of "Appendix C.1". Let us briefly outline the proof strategy:

1. We first show in Proposition C.1 that $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ is *closed under scalar multiplication*, hence *star-shaped with respect to the origin, i.e., 0 is a center.* [1]
2. Next, using the local Lipschitz continuity of ϱ, we establish in Proposition C.4 that the maximal *number of linearly independent centers* of the set $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ is finite. Precisely, it is bounded by the number of parameters of the underlying neural networks, given by $\sum_{\ell=1}^{L}(N_{\ell-1}+1)N_{\ell}$.
3. A direct consequence of Step 2 is that if $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ is convex, then it can only contain a finite number of linearly independent functions; see Corollary C.5.
4. Finally, using that $\mathcal{RNN}_{\varrho}^{\mathbb{R}^d}(S)$ is a *translation-invariant subset of* $C(\mathbb{R}^d)$, we show in Proposition C.6 that $\mathcal{RNN}_{\varrho}^{\mathbb{R}^d}(S)$ (and hence also $\mathcal{RNN}_{\varrho}^{\Omega}(S)$) contains *infinitely many linearly independent functions*, if ϱ is not a polynomial.

□

In applications, the non-convexity of $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ might not be as problematic as it first seems. If, for instance, the set $\mathcal{RNN}_{\varrho}^{\Omega}(S) + B_{\delta}(0)$ of functions that can be approximated up to error $\delta > 0$ by a neural network with architecture S was convex, then one could argue that the non-convexity of $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ was not severe. Indeed, in practice, neural networks are only trained to minimize a certain empirical loss function, with resulting bounds on the generalization error which are typically of size $\varepsilon = \mathcal{O}(m^{-1/2})$, with m denoting the number of training samples. In this setting, one is not really interested in "completely minimizing" the (empirical) loss function, but would be content with finding a function for which the empirical loss is ε-close to the global minimum. Hence, one could argue that one is effectively working with a hypothesis space of the form $\mathcal{RNN}_{\varrho}^{\Omega}(S) + B_{\delta}(0)$, containing all functions that can be represented up to an error of δ by neural networks of architecture S.

[1] A subset A of some vector space V is called **star-shaped**, if there exists some $f \in A$ such that for all $g \in A$, also $\{\lambda f + (1-\lambda)g : \lambda \in [0,1]\} \subset A$. The vector f is called a **center of** A.

Table 1 Commonly used activation functions and their properties

Name	Given by	Smoothness/ boundedness	References
Rectified linear unit (ReLU)	$\max\{0, x\}$	$C(\mathbb{R})$/unbounded	[55]
Parametric ReLU	$\max\{ax, x\}$ for some $a \geq 0, a \neq 1$	$C(\mathbb{R})$/unbounded	[33]
Exponential linear unit	$x \cdot \chi_{x\geq 0}(x) + (\exp(x) - 1) \cdot \chi_{x<0}(x)$	$C^1(\mathbb{R})$/unbounded	[17]
Softsign	$\frac{x}{1+\lvert x\rvert}$	$C^1(\mathbb{R})$/bounded	[12]
Inverse square root linear unit	$x \cdot \chi_{x\geq 0}(x) + \frac{x}{\sqrt{1+ax^2}} \cdot \chi_{x<0}(x)$ for $a > 0$	$C^2(\mathbb{R})$/unbounded	[15]
Inverse square root unit	$\frac{x}{\sqrt{1+ax^2}}$ for $a > 0$	Analytic/bounded	[15]
sigmoid/logistic	$\frac{1}{1+\exp(-x)}$	Analytic/bounded	[32]
tanh	$\frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}$	Analytic/bounded	[47]
arctan	$\arctan(x)$	Analytic/bounded	[46]
Softplus	$\ln(1 + \exp(x))$	Analytic/unbounded	[29]

To quantify this potentially more relevant notion of convexity of neural networks, we define, for a subset A of a vector space $\mathcal{Y}$, the **convex hull of** A as

$$\operatorname{co}(A) := \bigcap_{B \subset \mathcal{Y} \text{ convex and } B \supset A} B \,.$$

For $\varepsilon > 0$, we say that a subset A of a normed vector space $\mathcal{Y}$ is **ε-convex in** $(\mathcal{Y}, \|\cdot\|_{\mathcal{Y}})$, if

$$\operatorname{co}(A) \subset A + B_\varepsilon(0) \,.$$

Hence, the notion of ε-convexity asks whether the convex hull of a set is contained in an enlargement of this set. Note that if $\mathcal{RNN}_\varrho^\Omega(S)$ is dense in $C(\Omega)$, then its closure is trivially ε-convex for all $\varepsilon > 0$. Our main result regarding the ε-convexity of neural network sets shows that this is the only case in which $\overline{\mathcal{RNN}_\varrho^\Omega(S)}$ is ε-convex for any $\varepsilon > 0$.

Theorem 2.2 *Let $S = (d, N_1, \dots, N_{L-1}, 1)$ be a neural network architecture with $L \geq 2$, and let $\Omega \subset \mathbb{R}^d$ be compact. Let $\varrho : \mathbb{R} \to \mathbb{R}$ be continuous but not a polynomial, and such that $\varrho'(x_0) \neq 0$ for some $x_0 \in \mathbb{R}$.*

Assume that $\mathcal{RNN}_\varrho^\Omega(S)$ is not *dense in $C(\Omega)$. Then there does not exist* any *$\varepsilon > 0$ such that $\overline{\mathcal{RNN}_\varrho^\Omega(S)}$ is ε-convex in $\big(C(\Omega), \|\cdot\|_{\sup}\big)$.*

Remark All closures in the theorem are taken in $C(\Omega)$.

Proof The proof of this theorem is the subject of "Appendix C.2". It is based on showing that if $\overline{\mathcal{RNN}_\varrho^\Omega(S)}$ is ε-convex for some $\varepsilon > 0$, then in fact $\overline{\mathcal{RNN}_\varrho^\Omega(S)}$ is convex, which we then use to show that $\overline{\mathcal{RNN}_\varrho^\Omega(S)}$ contains all realizations of two-layer neural networks with activation function ϱ. As shown in [45], this implies that $\mathcal{RNN}_\varrho^\Omega(S)$ is dense in $C(\Omega)$, since ϱ is not a polynomial. □

Remark 2.3 While it is certainly natural to expect that $\overline{\mathcal{RNN}_\varrho^\Omega(S)} \neq C(\Omega)$ should hold for most activation functions ϱ, giving a reference including large classes of activation functions such that the claim holds is not straightforward. We study this problem more closely in "Appendix C.3".

To be more precise, from Proposition C.10 it follows that the ReLU, the parametric ReLU, the exponential linear unit, the softsign, the sigmoid, and the tanh yield realization sets $\mathcal{RNN}_\varrho^\Omega(S)$ which are *not* dense in $L^p(\Omega)$ and in $C(\Omega)$.

The only activation functions listed in Table 1 for which we do *not* know whether any of the realization sets $\mathcal{RNN}_\varrho^\Omega(S)$ is dense in $L^p(\Omega)$ or in $C(\Omega)$ are: the inverse square root linear unit, the inverse square root unit, the softplus, and the arctan function. Of course, we expect that also for these activation functions, the resulting sets of realizations are never dense in $L^p(\Omega)$ or in $C(\Omega)$.

Finally, we would like to mention that if $\Omega \subset \mathbb{R}^d$ has non-empty interior and if the input dimension satisfies $d \geq 2$, then it follows from the results in [48] that if $S = (d, N_1, 1)$ is a *two-layer architecture*, then $\mathcal{RNN}_\varrho^\Omega(S)$ is *not* dense in $C(\Omega)$ or $L^p(\Omega)$.

3 (Non-)Closedness of the Set of Realizations

Let $\varnothing \neq \Omega \subset \mathbb{R}^d$ be compact with non-empty interior. In the present section, we analyze whether the neural network realization set $\mathcal{RNN}_\varrho^\Omega(S)$ with $S = (d, N_1, \ldots, N_{L-1}, 1)$ is closed in $C(\Omega)$, or in $L^p(\mu)$, for $p \in (0, \infty)$ and any measure μ satisfying a mild "non-atomicness" condition. For the L^p-spaces, the answer is simple: Under very mild assumptions on the activation function ϱ, we will see that $\mathcal{RNN}_\varrho^\Omega(S)$ is never closed in $L^p(\mu)$. In particular, this holds for *all* of the activation functions listed in Table 1. Closedness in $C(\Omega)$, however, is more subtle: For this setting, we will identify several different classes of activation functions for which the set $\mathcal{RNN}_\varrho^\Omega(S)$ is *not* closed in $C(\Omega)$. As we will see, these classes of activation functions cover all those functions listed in Table 1, *except for the ReLU and the parametric ReLU*. For these two activation functions, we were unable to determine whether the set $\mathcal{RNN}_\varrho^\Omega(S)$ is closed in $C(\Omega)$ in general, but we conjecture this to be true. Only for the case $L = 2$, we could show that these sets are indeed closed.

Closedness of $\mathcal{RNN}_\varrho^\Omega(S)$ is a highly desirable property as we will demonstrate in Sect. 3.3. Indeed, we establish that if $X = L^p(\mu)$ or $X = C(\Omega)$, then, for all functions $f \in X$ *that do not possess a best approximation within* $\mathcal{R} = \mathcal{RNN}_\varrho^\Omega(S)$, the weights of approximating networks necessarily explode. In other words, if $(\mathrm{R}_\varrho^\Omega(\Phi_n))_{n\in\mathbb{N}} \subset \mathcal{R}$ is such that $\|f - \mathrm{R}_\varrho^\Omega(\Phi_n)\|_X$ converges to $\inf_{g\in\mathcal{R}} \|f - g\|_X$ for $n \to \infty$, then $\|\Phi_n\|_{\mathrm{total}} \to \infty$. Such functions without a best approximation in $\mathcal{R}$ necessarily exist if $\mathcal{R}$ is not closed. Moreover, even in practical applications, where empirical error terms instead of $L^p(\mu)$ norms are minimized, the absence of closedness implies exploding weights as we show in Proposition 3.6.

Finally, we note that for simplicity, all "non-closedness" results in this section are formulated for compact rectangles of the form $\Omega = [-B, B]^d$ only; but our arguments easily generalize to any compact set $\Omega \subset \mathbb{R}^d$ with non-empty interior.

3.1 Non-closedness in $L^p(\mu)$

We start by examining the closedness with respect to L^p-norms for $p \in (0, \infty)$. In fact, for all $B > 0$ and all widely used activation functions (including all activation functions presented in Table 1), the set $\mathcal{RNN}_\varrho^{[-B,B]^d}(S)$ is *not* closed in $L^p(\mu)$, for any $p \in (0, \infty)$ and any "sufficiently non-atomic" measure μ on $[-B, B]^d$. To be more precise, the following is true:

Theorem 3.1 *Let $S = (d, N_1, \ldots, N_{L-1}, 1)$ be a neural network architecture with $L \in \mathbb{N}_{\geq 2}$. Let $\varrho : \mathbb{R} \to \mathbb{R}$ be a function satisfying the following conditions:*

(i) *ϱ is continuous and increasing;*
(ii) *There is some $x_0 \in \mathbb{R}$ such that ϱ is differentiable at x_0 with $\varrho'(x_0) \neq 0$;*
(iii) *There is some $r > 0$ such that $\varrho|_{(-\infty,-r)\cup(r,\infty)}$ is differentiable;*
(iv) At least one *of the following two conditions is satisfied:*

(a) *There are $\lambda, \lambda' \geq 0$ with $\lambda \neq \lambda'$ such that $\varrho'(x) \to \lambda$ as $x \to \infty$, and $\varrho'(x) \to \lambda'$ as $x \to -\infty$, and we have $N_{L-1} \geq 2$.*

(b) ϱ is bounded.

Finally, let $B > 0$ and let μ be a finite Borel measure on $[-B, B]^d$ for which the support $\mathrm{supp}\mu$ is uncountable. Then the set $\mathcal{RNN}_{\varrho}^{[-B,B]^d}(S)$ is not *closed in $L^p(\mu)$ for any $p \in (0, \infty)$. More precisely, there is a function $f \in L^\infty(\mu)$ which satisfies $f \in \overline{\mathcal{RNN}_{\varrho}^{[-B,B]^d}(S)} \setminus \mathcal{RNN}_{\varrho}^{[-B,B]^d}(S)$ for all $p \in (0, \infty)$, where the closure is taken in $L^p(\mu)$.*

Remark If $\mathrm{supp}\mu$ is countable, then $\mu = \sum_{x \in \mathrm{supp}\mu} \mu(\{x\})\, \delta_x$ is a countable sum of Dirac measures, meaning that μ is *purely atomic*. In particular, if μ is *non-atomic* (meaning that $\mu(\{x\}) = 0$ for all $x \in [-B, B]^d$), then $\mathrm{supp}\mu$ is uncountable and the theorem is applicable.

Proof For the proof of the theorem, we refer to "Appendix D.1". The main proof idea consists in the approximation of a (discontinuous) step function which cannot be represented by a neural network with continuous activation function. □

Corollary 3.2 *The assumptions concerning the activation function ϱ in Theorem 3.1 are satisfied for all of the activation functions listed in Table 1. In any case where ϱ is bounded, one can take $N_{L-1} = 1$; otherwise, one can take $N_{L-1} = 2$.*

Proof For a proof of this statement, we refer to "Appendix D.2". □

3.2 Non-closedness in $C([-B, B]^d)$ for Many Widely Used Activation Functions

We have seen in Theorem 3.1 that under reasonably mild assumptions on the activation function ϱ—which are satisfied for all commonly used activation functions—the set $\mathcal{RNN}_{\varrho}^{[-B,B]^d}(S)$ is not closed in any L^p-space where $p \in (0, \infty)$. However, the argument of the proof of Theorem 3.1 breaks down if one considers closedness with respect to the $\|\cdot\|_{\sup}$-norm. Therefore, we will analyze this setting more closely in this section. More precisely, in Theorem 3.3, we present several criteria regarding the activation function ϱ which imply that the set $\mathcal{RNN}_{\varrho}^{[-B,B]^d}(S)$ is *not* closed in $C([-B, B]^d)$. We remark that in all these results, ϱ will be assumed to be at least C^1. Developing similar criteria for non-differentiable functions is an interesting topic for future research.

Before we formulate Theorem 3.3, we need the following notion: We say that a function $f : \mathbb{R} \to \mathbb{R}$ is **approximately homogeneous of order** $(r, q) \in \mathbb{N}_0^2$ if there exists $s > 0$ such that $|f(x) - x^r| \leq s$ for all $x \geq 0$ and $|f(x) - x^q| \leq s$ for all $x \leq 0$. Now the following theorem holds:

Theorem 3.3 *Let $S = (d, N_1, \ldots, N_{L-1}, 1)$ be a neural network architecture with $L \in \mathbb{N}_{\geq 2}$, let $B > 0$, and let $\varrho : \mathbb{R} \to \mathbb{R}$. Assume that* at least one *of the following three conditions is satisfied:*

(i) *$N_{L-1} \geq 2$ and $\varrho \in C^1(\mathbb{R}) \setminus C^\infty(\mathbb{R})$.*
(ii) *$N_{L-1} \geq 2$ and ϱ is bounded, analytic, and not constant.*
(iii) *ϱ is approximately homogeneous of order (r, q) for certain $r, q \in \mathbb{N}_0$ with $r \neq q$, and $\varrho \in C^{\max\{r,q\}}(\mathbb{R})$.*

Then the set $\mathcal{RNN}_{\varrho}^{[-B,B]^d}(S)$ *is* not *closed in the space* $C([-B, B]^d)$.

Proof For the proof of the statement, we refer to "Appendix D.3". In particular, the proof of the statement under Condition (i) can be found in "Appendix D.3.1". Its main idea consists of the uniform approximation of ϱ' (which cannot be represented by neural networks with activation function ϱ, due to its lack of sufficient regularity) by neural networks. For the proof of the statement under Condition (ii), we refer to "Appendix D.3.2". The main proof idea consists in the uniform approximation of an unbounded analytic function which cannot be represented by a neural network with activation function ϱ, since ϱ itself is bounded. Finally, the proof of the statement under Condition (iii) can be found in "Appendix D.3.3". Its main idea consists in the approximation of the function $x \mapsto (x)_+^{\max\{r,q\}} \notin C^{\max\{r,q\}}$. □

Corollary 3.4 *Theorem 3.3 applies to all activation functions listed in Table 1* except *for the ReLU and the parametric ReLU. To be more precise,*

(1) *Condition (i) is fulfilled by the function* $x \mapsto \max\{0, x\}^k$ *for* $k \geq 2$*, and by the exponential linear unit, the softsign function, and the inverse square root linear unit.*
(2) *Condition (ii) is fulfilled by the inverse square root unit, the sigmoid function, the* tanh *function, and the* arctan *function.*
(3) *Condition (iii) (with* $r = 1$ *and* $q = 0$*) is fulfilled by the softplus function.*

Proof For the proof of this statement, we refer to "Appendix D.4". In particular, for the proof of (1), we refer to "Appendix D.4.1", the proof of (2) is clear and for the proof of (3), we refer to "Appendix D.4.2". □

3.3 The Phenomenon of Exploding Weights

We have just seen that the realization set $\mathcal{RNN}_{\varrho}^{[-B,B]^d}(S)$ is not closed in $L^p(\mu)$ for any $p \in (0, \infty)$ and every practically relevant activation function. Furthermore, for a variety of activation functions, we have seen that $\mathcal{RNN}_{\varrho}^{[-B,B]^d}(S)$ is not closed in $C([-B, B]^d)$. The situation is substantially different if the weights are taken from a compact subset:

Proposition 3.5 *Let* $S = (d, N_1, \ldots, N_L)$ *be a neural network architecture, let* $\Omega \subset \mathbb{R}^d$ *be compact, let furthermore* $p \in (0, \infty)$*, and let* $\varrho : \mathbb{R} \to \mathbb{R}$ *be continuous. For* $C > 0$*, let*

$$\Theta_C := \left\{\Phi \in \mathcal{NN}(S) : \|\Phi\|_{\text{total}} \leq C\right\}.$$

Then the set $\mathrm{R}_{\varrho}^{\Omega}(\Theta_C)$ *is compact in* $C(\Omega)$ *as well as in* $L^p(\mu)$*, for any finite Borel measure* μ *on* Ω *and any* $p \in (0, \infty)$.

Proof The proof of this statement is based on the continuity of the realization map and can be found in "Appendix D.5". □

Proposition 3.5 helps to explain the phenomenon of exploding network weights that is sometimes observed during the training of neural networks. Indeed, let us assume that $\mathcal{R} := \mathcal{RNN}_{\varrho}^{[-B,B]^d}(S)$ is *not* closed in $\mathcal{Y}$, where $\mathcal{Y} := L^p(\mu)$ for a Borel measure μ on $[-B, B]^d$, or $\mathcal{Y} := C([-B, B]^d)$; as seen in Sects. 3.1 and 3.2, this is true under mild assumptions on ϱ. Then, it follows that there exists a function $f \in \mathcal{Y}$ which *does not have a best approximation in* $\mathcal{R}$, meaning that there does not exist any $g \in \mathcal{R}$ satisfying

$$\|f - g\|_{\mathcal{Y}} = \inf_{h \in \mathcal{R}} \|f - h\|_{\mathcal{Y}} =: M\,;$$

in fact, one can take any $f \in \overline{\mathcal{R}} \setminus \mathcal{R}$. Next, recall from Proposition 3.5 that the subset of $\mathcal{R}$ that contains only realizations of networks with uniformly bounded weights is compact.

Hence, we conclude the following: For every sequence

$$(f_n)_{n\in\mathbb{N}} = \left(\mathrm{R}_{\varrho}^{[-B,B]^d}(\Phi_n)\right)_{n\in\mathbb{N}} \subset \mathcal{R}$$

satisfying $\|f - f_n\|_{\mathcal{Y}} \to M$, we must have $\|\Phi_n\|_{\text{total}} \to \infty$, since otherwise, by compactness, $(f_n)_{n\in\mathbb{N}}$ would have a subsequence that converges to some $h \in \mathrm{R}_{\varrho}^{\Omega}(\Theta_C) \subset \mathcal{R}$. In other words, *the weights of the networks* Φ_n *necessarily explode*.

The argument above only deals with the approximation problem in the space $C([-B, B]^d)$ or in $L^p(\mu)$ for $p \in (0, \infty)$. In practice, one is often not concerned with these norms, but instead wants to minimize an *empirical loss function* over $\mathcal{R}$. For the empirical square loss, this loss function takes the form

$$E_N(f) := \frac{1}{N} \sum_{i=1}^{N} |f(x_i) - y_i|^2,$$

for $\left((x_i, y_i)\right)_{i=1}^{N} \subset \Omega \times \mathbb{R}$ drawn i.i.d. according to a probability distribution σ on $\Omega \times \mathbb{R}$. By the strong law of large numbers, for each fixed measurable function f, the empirical loss function converges almost surely to the *expected loss*

$$\mathcal{E}_\sigma(f) := \int_{\Omega\times\mathbb{R}} |f(x) - y|^2 \,\mathrm{d}\sigma(x, y). \tag{3.1}$$

This expected loss is related to an L^2 minimization problem. Indeed, [20, Proposition 1] shows that there is a measurable function $f_\sigma : \Omega \to \mathbb{R}$—called the *regression function*—such that the expected risk from Eq. (3.1) satisfies

$$\mathcal{E}_\sigma(f) = \mathcal{E}_\sigma(f_\sigma) + \int_{\Omega} |f(x) - f_\sigma(x)|^2 \,\mathrm{d}\sigma_\Omega(x) \text{ for each measurable } f : \Omega \to \mathbb{R}. \tag{3.2}$$

Here, σ_Ω is the marginal probability distribution of σ on Ω, and $\mathcal{E}_\sigma(f_\sigma)$ is called the *Bayes risk*; it is the minimal expected loss achievable by any (measurable) function.

In this context of a statistical learning problem, we have the following result regarding exploding weights:

Proposition 3.6 *Let $d \in \mathbb{N}$ and $B, K > 0$. Let $\Omega := [-B, B]^d$. Moreover, let σ be a Borel probability measure on $\Omega \times [-K, K]$. Further, let $S = (d, N_1, \dots, N_{L-1}, 1)$ be a neural network architecture and $\varrho : \mathbb{R} \to \mathbb{R}$ be locally Lipschitz continuous. Assume that the regression function f_σ is such that there does not exist a best approximation of f_σ in $\mathcal{RNN}_\varrho^\Omega(S)$ with respect to $\|\cdot\|_{L^2(\sigma_\Omega)}$. Let $\big((x_i, y_i)\big)_{i\in\mathbb{N}} \overset{\text{i.i.d.}}{\sim} \sigma$; all probabilities below will be with respect to this family of random variables.*

If $(\Phi_N)_{N\in\mathbb{N}} \subset \mathcal{NN}(S)$ is a random sequence of neural networks (depending on $\big((x_i, y_i)\big)_{i\in\mathbb{N}}$) that satisfies

$$\mathbb{P}\left(E_N\left(\mathrm{R}_\varrho^\Omega(\Phi_N)\right) - \inf_{f\in\mathcal{RNN}_\varrho^\Omega(S)} E_N(f) > \varepsilon\right) \to 0 \text{ as } N \to \infty \text{ for all } \varepsilon > 0, \tag{3.3}$$

then $\|\Phi_N\|_{\text{total}} \to \infty$ in probability as $N \to \infty$.

Remark A compact way of stating Proposition 3.6 is that, if f_σ has no best approximation in $\mathcal{RNN}_\varrho^\Omega(S)$ with respect to $\|\cdot\|_{L^2(\sigma_\Omega)}$, then the weights of the minimizers (or approximate minimizers) of the empirical square loss explode for increasing numbers of samples.

Since σ is unknown in applications, it is indeed possible that f_σ has no best approximation in the set of neural networks. As just one example, this is true if σ_Ω is any Borel probability measure on Ω and if σ is the distribution of $(X, g(X))$, where $X \sim \sigma_\Omega$ and $g \in L^2(\sigma_\Omega)$ is bounded and satisfies $g \in \overline{\mathcal{RNN}_\varrho^\Omega(S)} \setminus \mathcal{RNN}_\varrho^\Omega(S)$, with the closure taken in $L^2(\sigma_\Omega)$. The existence of such a function g is guaranteed by Theorem 3.1 if $\mathrm{supp}\sigma_\Omega$ is uncountable.

Proof For the proof of Proposition 3.6, we refer to "Appendix D.6". The proof is based on classical arguments of statistical learning theory as given in [20]. □

3.4 Closedness of ReLU Networks in $C([-B, B]^d)$

In this subsection, we analyze the closedness of sets of realizations of neural networks with respect to the ReLU or the parametric ReLU activation function in $C(\Omega)$, mostly for the case $\Omega = [-B, B]^d$. We conjecture that the set of (realizations of) ReLU networks of a fixed complexity is closed in $C(\Omega)$, but were not able to prove such a result in full generality. In two special cases, namely when the networks have only two layers, or when at least the *scaling* weights are bounded, we can show that the associated set of ReLU realizations is closed in $C(\Omega)$; see below.

We begin by analyzing the set of realizations with uniformly bounded scaling weights and possibly unbounded biases, before proceeding with the analysis of two layer ReLU networks.

For $\Phi = \big((A_1, b_1), \dots, (A_L, b_L)\big) \in \mathcal{NN}(S)$ satisfying $\|\Phi\|_{\text{scaling}} \leq C$ for some $C > 0$, we say that the network Φ has C**-bounded scaling weights**. Note that this does *not* require the biases b_ℓ of the network to satisfy $|b_\ell| \leq C$.

Our first goal in this subsection is to show that if ϱ denotes the ReLU, if $S = (d, N_1, \dots, N_L)$, if $C > 0$, and if $\Omega \subset \mathbb{R}^d$ is measurable and bounded, then the set

$$\mathcal{RNN}_\varrho^{\Omega, C}(S) := \left\{ \mathrm{R}_\varrho^\Omega(\Phi) \,:\, \Phi \in \mathcal{NN}(S) \text{ with } \|\Phi\|_{\text{scaling}} \leq C \right\}$$

is closed in $C(\Omega; \mathbb{R}^{N_L})$ and in $L^p(\mu; \mathbb{R}^{N_L})$ for arbitrary $p \in [1, \infty]$. Here, and in the remainder of the paper, we use the norm $\|f\|_{L^p(\mu; \mathbb{R}^{N_L})} = \| |f| \|_{L^p(\mu)}$ for vector-valued L^p-spaces. The norm on $C(\Omega; \mathbb{R}^{N_L})$ is defined similarly. The difference between the following proposition and Proposition 3.5 is that in the following proposition, the "shift weights" (the biases) of the networks can be potentially unbounded. Therefore, the resulting set is merely closed, not compact.

Proposition 3.7 *Let $S = (d, N_1, \dots, N_L)$ be a neural network architecture, let $C > 0$, and let $\Omega \subset \mathbb{R}^d$ be Borel measurable and bounded. Finally, let $\varrho : \mathbb{R} \to \mathbb{R}, x \mapsto \max\{0, x\}$ denote the ReLU function.*

Then the set $\mathcal{RNN}_\varrho^{\Omega, C}(S)$ is closed in $L^p(\mu; \mathbb{R}^{N_L})$ for every $p \in [1, \infty]$ and any finite Borel measure μ on Ω. If Ω is compact, then $\mathcal{RNN}_\varrho^{\Omega, C}(S)$ is also closed in $C(\Omega; \mathbb{R}^{N_L})$.

Remark In fact, the proof shows that each subset of $\mathcal{RNN}_\varrho^{\Omega, C}(S)$ which is bounded in $L^1(\mu; \mathbb{R}^{N_L})$ (when $\mu(\Omega) > 0$) is precompact in $L^p(\mu; \mathbb{R}^{N_L})$ and in $C(\Omega; \mathbb{R}^{N_L})$.

Proof For the proof of the statement, we refer to "Appendix D.7". The main idea is to show that for every sequence $(\Phi_n)_{n \in \mathbb{N}} \subset \mathcal{NN}(S)$ of neural networks with uniformly bounded *scaling* weights and with $\|\mathrm{R}_\varrho^\Omega(\Phi_n)\|_{L^1(\mu)} \leq M$, there exist a subsequence $(\Phi_{n_k})_{k \in \mathbb{N}}$ of $(\Phi_n)_{n \in \mathbb{N}}$ and neural networks $(\widetilde{\Phi}_{n_k})_{k \in \mathbb{N}}$ with uniformly bounded scaling weights *and* biases such that $\mathrm{R}_\varrho^\Omega\big(\widetilde{\Phi}_{n_k}\big) = \mathrm{R}_\varrho^\Omega\big(\Phi_{n_k}\big)$. The rest then follows from Proposition 3.5. □

As our second result in this section, we show that the set of realizations of *two-layer* neural networks with arbitrary scaling weights and biases is closed in $C([-B, B]^d)$, if the activation is the parametric ReLU. It is a fascinating question for further research whether this also holds for deeper networks.

Theorem 3.8 *Let $d, N_0 \in \mathbb{N}$, let $B > 0$, and let $a \geq 0$. Let $\varrho_a : \mathbb{R} \to \mathbb{R}, x \mapsto \max\{x, ax\}$ be the parametric ReLU. Then $\mathcal{RNN}_{\varrho_a}^{[-B,B]^d}((d, N_0, 1))$ is closed in $C([-B, B]^d)$.*

Proof For the proof of the statement, we refer to "Appendix D.8"; here we only sketch the main idea: First, note that each $f \in \mathcal{RNN}_{\varrho_a}^{[-B,B]^d}((d, N_0, 1))$ is of the form $f(x) = c + \sum_{i=1}^{N_0} \varrho_a(\langle \alpha_i, x \rangle + \beta_i)$. The proof is based on a careful—and quite technical—analysis of the *singularity hyperplanes* of the functions $\varrho_a(\langle \alpha_i, x \rangle + \beta_i)$,

that is, the hyperplanes $\langle \alpha_i, x \rangle + \beta_i = 0$ on which these functions are not differentiable. More precisely, given a *uniformly convergent* sequence $(f_n)_{n\in\mathbb{N}} \subset \mathcal{RNN}_{\varrho a}^{[-B,B]^d}((d, N_0, 1))$, we analyze how the singularity hyperplanes of the functions f_n behave as $n \to \infty$, in order to show that the limit is again of the same form as the f_n. For more details, we refer to the actual proof. □

4 Failure of Inverse Instability of the Realization Map

In this section, we study the properties of the realization map $\mathrm{R}_\varrho^\Omega$. First of all, we observe that the realization map is continuous.

Proposition 4.1 *Let $\Omega \subset \mathbb{R}^d$ be compact and let $S = (d, N_1, \dots, N_L)$ be a neural network architecture. If the activation function $\varrho : \mathbb{R} \to \mathbb{R}$ is continuous, then the realization map from Eq. (1.1) is continuous. If ϱ is locally Lipschitz continuous, then so is $\mathrm{R}_\varrho^\Omega$.*

Finally, if ϱ is globally Lipschitz continuous, then there is a constant $C = C(\varrho, S) > 0$ such that

$$\mathrm{Lip}\big(\mathrm{R}_\varrho^\Omega(\Phi)\big) \leq C \cdot \|\Phi\|_{\text{scaling}}^L \qquad \text{for all } \Phi \in \mathcal{NN}(S)\,.$$

Proof For the proof of this statement, we refer to "Appendix E.1". □

In general, the realization map is not injective; that is, there can be networks $\Phi \neq \Psi$ but such that $\mathrm{R}_\varrho^\Omega(\Phi) = \mathrm{R}_\varrho^\Omega(\Psi)$; in fact, if for instance

$$\Phi = \big((A_1, b_1), \dots, (A_{L-1}, b_{L-1}), (0, 0)\big),$$

and

$$\Psi = \big((B_1, c_1), \dots, (B_{L-1}, c_{L-1}), (0, 0)\big),$$

then the realizations of Φ, Ψ are identical.

In this section, our main goal is to determine whether, up to the failure of injectivity, the realization map is a homeomorphism onto its range; mathematically, this means that we want to determine whether the realization map is a *quotient map*. We will see that this is *not* the case.

To this end, we will prove for fixed Φ that even if $\mathrm{R}_\varrho^\Omega(\Psi)$ is very close to $\mathrm{R}_\varrho^\Omega(\Phi)$, it is *not* true in general that $\mathrm{R}_\varrho^\Omega(\Psi) = \mathrm{R}_\varrho^\Omega(\widetilde{\Psi})$ for network weights $\widetilde{\Psi}$ close to Φ. Precisely, this follows from the following theorem for $\Phi = 0$ and $\Psi = \Phi_n$.

Theorem 4.2 *Let $\varrho : \mathbb{R} \to \mathbb{R}$ be Lipschitz continuous, but not affine-linear. Let $S = (N_0, \dots, N_{L-1}, 1)$ be a network architecture with $L \geq 2$, with $N_0 = d$, and $N_1 \geq 3$. Let $\Omega \subset \mathbb{R}^d$ be bounded with nonempty interior.*

Then there is a sequence $(\Phi_n)_{n\in\mathbb{N}}$ of networks with architecture S and the following properties:

1. *We have* $\mathrm{R}_\varrho^\Omega(\Phi_n) \to 0$ *uniformly on* Ω.
2. *We have* $\mathrm{Lip}(\mathrm{R}_\varrho^\Omega(\Phi_n)) \to \infty$ *as* $n \to \infty$.

Finally, if $(\Phi_n)_{n\in\mathbb{N}}$ *is a sequence of networks with architecture* S *and the preceding two properties, then the following holds: For each sequence of networks* $(\Psi_n)_{n\in\mathbb{N}}$ *with architecture* S *and* $\mathrm{R}_\varrho^\Omega(\Psi_n) = \mathrm{R}_\varrho^\Omega(\Phi_n)$, *we have* $\|\Psi_n\|_{\mathrm{scaling}} \to \infty$.

Proof For the proof of the statement, we refer to "Appendix E.2". The proof is based on the fact that the Lipschitz constant of the realization of a network essentially yields a lower bound on the $\|\cdot\|_{\mathrm{scaling}}$ norm of every neural network with this realization. We construct neural networks Φ_n the realizations of which have small amplitude but high Lipschitz constants. The associated realizations uniformly converge to 0, but every associated neural network must have exploding weights. □

We finally rephrase the preceding result in more topological terms:

Corollary 4.3 *Under the assumptions of Theorem 4.2, the realization map* $\mathrm{R}_\varrho^\Omega$ *from Eq. (1.1) is* not *a quotient map when considered as a map onto its range.*

Proof For the proof of the statement, we refer to "Appendix E.3". □

Acknowledgements Open access funding provided by University of Vienna. P.P. and M.R. were supported by the DFG Collaborative Research Center TRR 109 "Discretization in Geometry and Dynamics". P.P was supported by a DFG Research Fellowship "Shearlet-based energy functionals for anisotropic phase-field methods". M.R. is supported by the Berlin Mathematical School. F.V. acknowledges support from the European Commission through DEDALE (contract no. 665044) within the H2020 Framework Program.

Appendix A: Notation

The symbol $\mathbb{N}$ denotes the **natural numbers** $\mathbb{N} = \{1, 2, 3, \dots\}$, whereas $\mathbb{N}_0 = \{0\} \cup \mathbb{N}$ stands for the natural numbers including zero. Moreover, we set $\mathbb{N}_{\geq d} := \{n \in \mathbb{N} : n \geq d\}$ for $d \in \mathbb{N}$. The number of elements of a set M will be denoted by $|M| \in \mathbb{N}_0 \cup \{\infty\}$. Furthermore, we write $\underline{n} := \{k \in \mathbb{N} : k \leq n\}$ for $n \in \mathbb{N}_0$. In particular, $\underline{0} = \varnothing$.

For two sets A, B, a map $f : A \to B$, and $C \subset A$, we write $f|_C$ for the **restriction of** f **to** C. For a set A, we denote by $\chi_A = \mathbb{1}_A$ the **indicator function of** A, so that $\chi_A(x) = 1$ if $x \in A$ and $\chi_A(x) = 0$ otherwise. For any $\mathbb{R}$-vector space $\mathcal{Y}$ we write $A + B := \{a + b : a \in A, b \in B\}$ and $\lambda A := \{\lambda a : a \in A\}$, for $\lambda \in \mathbb{R}$ and subsets $A, B \subset \mathcal{Y}$.

The **algebraic dual space** of a $\mathbb{K}$-vector space $\mathcal{Y}$ (with $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$), that is the space of all linear functions $\varphi : \mathcal{Y} \to \mathbb{K}$, will be denoted by $\mathcal{Y}^*$. In contrast, if $\mathcal{Y}$ is

a *topological* vector space, we denote by $\mathcal{Y}'$ the **topological dual space** of $\mathcal{Y}$, which consists of all functions $\varphi \in \mathcal{Y}^*$ that are continuous.

Given functions $(f_i)_{i\in\underline{n}}$ with $f_i : X_i \to Y_i$, we consider three different types of products between these maps: The **Cartesian product** of $f_1, \dots, f_n$ is

$$f_1 \times \cdots \times f_n : X_1 \times \cdots \times X_n \to Y_1 \times \cdots \times Y_n, \quad (x_1, \dots, x_n) \mapsto \big(f_1(x_1), \dots, f_n(x_n)\big) .$$

The **tensor product** of $f_1, \dots, f_n$ is defined if $Y_1, \dots, Y_n \subset \mathbb{C}$, and is then given by

$$f_1 \otimes \cdots \otimes f_n : X_1 \times \cdots \times X_n \to \mathbb{C}, \quad (x_1, \dots, x_n) \mapsto f_1(x_1) \cdots f_n(x_n) .$$

Finally, the **direct sum** of $f_1, \dots, f_n$ is defined if $X_1 = \cdots = X_n$, and given by

$$f_1 \oplus \cdots \oplus f_n : X_1 \to Y_1 \times \cdots \times Y_n, \quad x \mapsto \big(f_1(x), \dots, f_n(x)\big) .$$

The **closure** of a subset A of a topological space will be denoted by $\overline{A}$, while the **interior** of A is denoted by A°. For a metric space $(\mathcal{U}, d)$, we write $B_\varepsilon(x) := \{y \in \mathcal{U} : d(x, y) < \varepsilon\}$ for the ε-**ball around** x, where $x \in \mathcal{U}$ and $\varepsilon > 0$. Furthermore, for a Lipschitz continuous function $f : \mathcal{U}_1 \to \mathcal{U}_2$ between two metric spaces $\mathcal{U}_1$ and $\mathcal{U}_2$, we denote by $\mathrm{Lip}(f)$ the smallest possible **Lipschitz constant** for f.

For $d \in \mathbb{N}$ and a function $f : A \to \mathbb{R}^d$ or a vector $v \in \mathbb{R}^d$, we denote for $j \in \{1, \dots, d\}$ the j-**th component** of f or v by $(f)_j$ or v_j, respectively. As an example, the **Euclidean scalar product** on $\mathbb{R}^d$ is given by $\langle x, y\rangle = \sum_{i=1}^d x_i\, y_i$. We denote the **Euclidean norm** by $|x| := \sqrt{\langle x, x\rangle}$ for $x \in \mathbb{R}^d$. For a matrix $A \in \mathbb{R}^{n\times d}$, let $\|A\|_{\max} := \max_{i=1,\dots,n} \max_{j=1,\dots,d} |A_{i,j}|$. The **transpose** of a matrix $A \in \mathbb{R}^{n\times d}$ will be denoted by $A^T \in \mathbb{R}^{d\times n}$. For $A \in \mathbb{R}^{n\times d}$, $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, d\}$, we denote by $A_{i,-} \in \mathbb{R}^d$ the i-th row of A and by $A_{-,j} \in \mathbb{R}^n$ the j-th column of A. The Euclidean **unit sphere** in $\mathbb{R}^d$ will be denoted by $S^{d-1} \subset \mathbb{R}^d$.

For $n \in \mathbb{N}$ and $\varnothing \neq \Omega \subset \mathbb{R}^d$, we denote by $C(\Omega; \mathbb{R}^n)$ the space of all **continuous functions defined on** Ω **with values in** $\mathbb{R}^n$. If Ω is compact, then $(C(\Omega; \mathbb{R}^n), \|\cdot\|_{\sup})$ denotes the Banach space of $\mathbb{R}^n$- valued continuous functions equipped with the supremum norm, where we use the Euclidean norm on $\mathbb{R}^n$. If $n = 1$, then we shorten the notation to $C(\Omega)$.

We note that on $C(\Omega)$, the supremum norm coincides with the $L^\infty(\Omega)$-norm, if for all $x \in \Omega$ and for all $\varepsilon > 0$ we have that $\lambda(\Omega \cap B_\varepsilon(x)) > 0$, where λ denotes the Lebesgue measure on $\mathbb{R}^d$. For any nonempty set $U \subset \mathbb{R}$, we say that a function $f : U \to \mathbb{R}$ is **increasing** if $f(x) \leq f(y)$ for every $x, y \in U$ with $x < y$. If even $f(x) < f(y)$ for all such x, y, we say that f is **strictly increasing**. The terms "decreasing" and "strictly decreasing" are defined analogously.

The **Schwartz space** will be denoted by $\mathcal{S}(\mathbb{R}^d)$ and the space of **tempered distributions** by $\mathcal{S}'(\mathbb{R}^d)$. The associated bilinear **dual pairing** will be denoted by $\langle\cdot,\cdot\rangle_{\mathcal{S}',\mathcal{S}}$. We refer to [26, Sects. 8.1–8.3 and 9.2] for more details on the spaces $\mathcal{S}(\mathbb{R}^d)$ and $\mathcal{S}'(\mathbb{R}^d)$. Finally, the **Dirac delta distribution** δ_x at $x \in \mathbb{R}^d$ is given by $\delta_x : C(\mathbb{R}^d) \to \mathbb{R}$, $f \mapsto f(x)$.

Appendix B: Auxiliary Results: Operations with Neural Networks

This part of the appendix is devoted to auxiliary results that are connected with basic operations one can perform with neural networks and which we will frequently make use of in the proofs below.

We start by showing that one can "enlarge" a given neural network in such a way that the realizations of the original network and the enlarged network coincide. To be more precise, the following holds:

Lemma B.1 *Let $d, L \in \mathbb{N}$, $\Omega \subset \mathbb{R}^d$, and $\varrho : \mathbb{R} \to \mathbb{R}$. Also, let $\Phi = \big((A_1, b_1), \dots, (A_L, b_L)\big)$ be a neural network with architecture $(d, N_1, \dots, N_L)$ and let $\widetilde{N}_1, \dots, \widetilde{N}_{L-1} \in \mathbb{N}$ such that $\widetilde{N}_\ell \geq N_\ell$ for all $\ell = 1, \dots, L-1$. Then, there exists a neural network $\widetilde{\Phi}$ with architecture $(d, \widetilde{N}_1, \dots, \widetilde{N}_{L-1}, N_L)$ and such that $\mathrm{R}_\varrho^\Omega(\Phi) = \mathrm{R}_\varrho^\Omega(\widetilde{\Phi})$.*

Proof Setting $N_0{:=}\widetilde{N}_0{:=}d$, and $\widetilde{N}_L{:=}N_L$, we define $\widetilde{\Phi}{:=}\big((\widetilde{A}_1, \widetilde{b}_1), \dots, (\widetilde{A}_L, \widetilde{b}_L)\big)$ by

$$\widetilde{A}_\ell{:=}\begin{pmatrix} A_\ell & 0_{N_\ell \times (\widetilde{N}_{\ell-1} - N_{\ell-1})} \\ 0_{(\widetilde{N}_\ell - N_\ell)\times N_{\ell-1}} & 0_{(\widetilde{N}_\ell - N_\ell)\times(\widetilde{N}_{\ell-1} - N_{\ell-1})} \end{pmatrix} \in \mathbb{R}^{\widetilde{N}_\ell \times \widetilde{N}_{\ell-1}},$$

and

$$\widetilde{b}_\ell{:=}\begin{pmatrix} b_\ell \\ 0_{\widetilde{N}_\ell - N_\ell} \end{pmatrix} \in \mathbb{R}^{\widetilde{N}_\ell},$$

for $\ell = 1, \dots, L$. Here, $0_{m_1 \times m_2}$ and 0_k denote the zero-matrix in $\mathbb{R}^{m_1 \times m_2}$ and the zero vector in $\mathbb{R}^k$, respectively. Clearly, $\mathrm{R}_\varrho^\Omega(\widetilde{\Phi}) = \mathrm{R}_\varrho^\Omega(\Phi)$. This yields the claim. □

Another operation that we can perform with networks is *concatenation*, as given in the following definition.

Definition B.2 Let $L_1, L_2 \in \mathbb{N}$ and let

$$\Phi^1 = \big((A_1^1, b_1^1), \dots, (A_{L_1}^1, b_{L_1}^1)\big),\ \Phi^2 = \big((A_1^2, b_1^2), \dots, (A_{L_2}^2, b_{L_2}^2)\big)$$

be two neural networks such that the input layer of Φ^1 has the same dimension as the output layer of Φ^2. Then, $\Phi^1 \bullet \Phi^2$ denotes the following $L_1 + L_2 - 1$ layer network:

$$\begin{aligned}\Phi^1 \bullet \Phi^2{:=}\big(&(A_1^2, b_1^2), \dots, (A_{L_2-1}^2, b_{L_2-1}^2), \\ &(A_1^1 A_{L_2}^2,\ A_1^1 b_{L_2}^2 + b_1^1), (A_2^1, b_2^1), \dots, (A_{L_1}^1, b_{L_1}^1)\big).\end{aligned}$$

Then, we call $\Phi^1 \bullet \Phi^2$ the **concatenation of Φ^1 and Φ^2**.

One directly verifies that for every $\varrho : \mathbb{R} \to \mathbb{R}$ the definition of concatenation is reasonable, that is, if d_i is the dimension of the input layer of Φ^i, $i = 1, 2$, and if $\Omega \subset \mathbb{R}^{d_2}$, then $\mathrm{R}_\varrho^\Omega(\Phi^1 \bullet \Phi^2) = \mathrm{R}_\varrho^{\mathbb{R}^{d_1}}(\Phi^1) \circ \mathrm{R}_\varrho^\Omega(\Phi^2)$. If Φ^2 has architecture

$(d, N_1, \dots, N_{L_2})$ and Φ^1 has architecture $(N_{L_2}, \widetilde{N}_1, \dots, \widetilde{N}_{L_1-1}, \widetilde{N}_{L_1})$, then the neural network $\Phi^1 \bullet \Phi^2$ has architecture $(d, N_1, \dots, N_{L_2-1}, \widetilde{N}_1, \dots, \widetilde{N}_{L_1})$. Therefore, $N(\Phi^1 \bullet \Phi^2) = N(\Phi^1) + N(\Phi^2) - 2N_{L_2}$.

We close this section by showing that under mild assumptions on ϱ—which are always satisfied in practice—and on the network architecture, one can construct a neural network which locally approximates the identity mapping $\mathrm{id}_{\mathbb{R}^d}$ to arbitrary accuracy. Similarly, one can obtain a neural network the realization of which approximates the projection onto the i-th coordinate. The main ingredient of the proof is the approximation $x \approx \frac{\varrho(x_0+x)-\varrho(x_0)}{\varrho'(x_0)}$, which holds for $|x|$ small enough and where x_0 is chosen such that $\varrho'(x_0) \neq 0$.

Proposition B.3 *Let $\varrho : \mathbb{R} \to \mathbb{R}$ be continuous, and assume that there exists $x_0 \in \mathbb{R}$ such that ϱ is differentiable at x_0 with $\varrho'(x_0) \neq 0$. Then, for every $\varepsilon > 0$, $d \in \mathbb{N}$, $B > 0$ and every $L \in \mathbb{N}$ there exists a neural network $\Phi_\varepsilon^B \in \mathcal{NN}((d, d, \dots, d))$ with L layers such that*

- $\left|\mathrm{R}_\varrho^{[-B,B]^d}(\Phi_\varepsilon^B)(x) - x\right| \leq \varepsilon$ *for all* $x \in [-B, B]^d$;
- $\mathrm{R}_\varrho^{[-B,B]^d}(\Phi_\varepsilon^B)(0) = 0$;
- $\mathrm{R}_\varrho^{[-B,B]^d}(\Phi_\varepsilon^B)$ *is totally differentiable at* $x = 0$ *and its Jacobian matrix fulfills* $D\big(\mathrm{R}_\varrho^{[-B,B]^d}(\Phi_\varepsilon^B)\big)(0) = \mathrm{id}_{\mathbb{R}^d}$;
- *for* $j \in \{1, \dots, d\}$, $\left(\mathrm{R}_\varrho^{[-B,B]^d}(\Phi_\varepsilon^B)\right)_j$ *is constant in all but the* j*-th coordinate.*

Furthermore, for every $d, L \in \mathbb{N}$, $\varepsilon > 0$, $B > 0$ and every $i \in \{1, \dots, d\}$, one can construct a neural network $\widetilde{\Phi}_{\varepsilon,i}^B \in \mathcal{NN}((d, 1, \dots, 1))$ with L layers such that

- $\left|\mathrm{R}_\varrho^{[-B,B]^d}(\widetilde{\Phi}_{\varepsilon,i}^B)(x) - x_i\right| \leq \varepsilon$ *for all* $x \in [-B, B]^d$;
- $\mathrm{R}_\varrho^{[-B,B]^d}(\widetilde{\Phi}_{\varepsilon,i}^B)(0) = 0$;
- $\mathrm{R}_\varrho^{[-B,B]^d}(\Phi_{\varepsilon,i}^B)$ *is partially differentiable at* $x = 0$, *with* $\frac{\partial}{\partial x_i}\Big|_{x=0} \mathrm{R}_\varrho^{[-B,B]^d}(\widetilde{\Phi}_{\varepsilon,i}^B)(x) = 1$; *and*
- $\mathrm{R}_\varrho^{[-B,B]^d}(\widetilde{\Phi}_{\varepsilon,i}^B)$ *is constant in all but the* i*-th coordinate.*

Finally, if ϱ is increasing, then $\left(\mathrm{R}_\varrho^{[-B,B]^d}(\Phi_\varepsilon^B)\right)_j$ and $\mathrm{R}_\varrho^{[-B,B]^d}(\widetilde{\Phi}_{\varepsilon,i}^B)$ are monotonically increasing in every coordinate and for all $j \in \{1, \dots, d\}$.

Proof We first consider the special case $L = 1$. Here, we can take $\Phi_\varepsilon^B := ((\mathrm{id}_{\mathbb{R}^d}, 0))$ and $\Phi_{\varepsilon,i}^B := ((e_i, 0))$, with $e_i \in \mathbb{R}^{1\times d}$ denoting the i-th standard basis vector in $\mathbb{R}^d \cong \mathbb{R}^{1\times d}$. In this case, $\mathrm{R}_\varrho^{[-B,B]^d}(\Phi_\varepsilon^B) = \mathrm{id}_{\mathbb{R}^d}$ and $\mathrm{R}_\varrho^{[-B,B]^d}(\Phi_{\varepsilon,i}^B)(x) = x_i$ for all $x \in [-B, B]^d$, which implies that all claimed properties are satisfied. Thus, we can assume in the following that $L \geq 2$.

Without loss of generality, we only consider the case $\varepsilon \leq 1$. Define $\varepsilon' := \varepsilon/(dL)$. Let $x_0 \in \mathbb{R}$ be such that ϱ is differentiable at x_0 with $\varrho'(x_0) \neq 0$.

We set $r_0 := \varrho(x_0)$ and $s_0 := \varrho'(x_0)$. Next, for $C > 0$, we define

$$\varrho_C : [-B - L\varepsilon, B + L\varepsilon] \to \mathbb{R}, \quad x \mapsto \frac{C}{s_0} \cdot \varrho\left(\frac{x}{C} + x_0\right) - \frac{Cr_0}{s_0}.$$

We claim that there is some $C_0 > 0$ such that $|\varrho_C(x) - x| \leq \varepsilon'$ for all $x \in [-B - L\varepsilon, B + L\varepsilon]$ and all $C \geq C_0$. To see this, first note by definition of the derivative that there is some $\delta > 0$ with

$$|\varrho(t + x_0) - r_0 - s_0 t| \leq \frac{|s_0| \cdot \varepsilon'}{1 + B + L} \cdot |t| \qquad \text{for all } t \in \mathbb{R} \text{ with } |t| \leq \delta.$$

Here we implicitly used that $s_0 = \varrho'(x_0) \neq 0$ to ensure that the right-hand side is a *positive* multiple of $|t|$. Now, set $C_0 := (B + L)/\delta$, and let $C \geq C_0$ be arbitrary. Note because of $\varepsilon' \leq \varepsilon \leq 1$ that every $x \in [-B - L\varepsilon, B + L\varepsilon]$ satisfies $|x| \leq B + L$. Hence, if we set $t := x/C$, then $|t| \leq \delta$. Therefore,

$$|\varrho_C(x) - x| = \left|\frac{C}{s_0}\right| \cdot \left|\varrho(t + x_0) - r_0 - s_0 t\right| \leq \left|\frac{C}{s_0}\right| \cdot \frac{|s_0| \cdot \varepsilon'}{1 + B + L} \cdot \left|\frac{x}{C}\right| \leq \varepsilon'.$$

Note that ϱ_C is differentiable at 0 with derivative $\varrho_C'(0) = \frac{C}{s_0}\varrho'(x_0)\frac{1}{C} = 1$, thanks to the chain rule.

Using these preliminary observations, we now construct the neural networks Φ_ε^B and $\Phi_{\varepsilon,i}^B$. Define $\Phi_0^C := \big((A_1, b_1), (A_2, b_2)\big)$, where

$$A_1 := \frac{1}{C} \cdot \mathrm{id}_{\mathbb{R}^d} \in \mathbb{R}^{d \times d}, \quad b_1 := x_0 \cdot (1, \dots, 1)^T \in \mathbb{R}^d,$$

and

$$A_2 := \frac{C}{s_0} \cdot \mathrm{id}_{\mathbb{R}^d} \in \mathbb{R}^{d \times d}, \quad b_2 := -\frac{C r_0}{s_0} \cdot (1, \dots, 1)^T \in \mathbb{R}^d.$$

Note $\Phi_0^C \in \mathcal{NN}((d, d, d))$. To shorten the notation, let $\Omega := [-B, B]^d$ and $J = [-B, B]$. It is not hard to see that $\mathrm{R}_\varrho^\Omega(\Phi_0^C) = \varrho_C|_J \times \cdots \times \varrho_C|_J$, where the Cartesian product has d factors. We define $\Phi_C := \Phi_0^C \bullet \Phi_0^C \bullet \cdots \bullet \Phi_0^C$, where we take $L - 2$ concatenations (meaning $L - 1$ factors, so that $\Phi_C = \Phi_0^C$ if $L = 2$). We obtain $\Phi_C \in \mathcal{NN}((d, \dots, d))$ (with L layers) and

$$\mathrm{R}_\varrho^\Omega(\Phi_C)(x) = \big(\varrho_C \circ \varrho_C \circ \cdots \circ \varrho_C(x_i)\big)_{i=1,\dots,d} \quad \text{for all} \quad x \in \Omega, \tag{B.1}$$

where ϱ_C is applied $L - 1$ times.

Since $|\varrho_C(x) - x| \leq \varepsilon' \leq \varepsilon$ for all $x \in [-B - L\varepsilon, B + L\varepsilon]$, it is not hard to see by induction that

$$|(\varrho_C \circ \cdots \circ \varrho_C)(x) - x| \leq t \cdot \varepsilon' \leq t \cdot \varepsilon \qquad \text{for all } x \in [-B, B],$$

where ϱ_C is applied $t \leq L$ times. Therefore, since $\varepsilon' = \varepsilon/(dL)$, we conclude for $C \geq C_0$ that

$$\left|\mathrm{R}_\varrho^\Omega(\Phi_C)(x) - x\right| \leq \varepsilon \quad \text{for all} \quad x \in \Omega.$$

As we saw above, ϱ_C is differentiable at 0 with $\varrho_C(0) = 0$ and $\varrho_C'(0) = 1$. By induction, we thus get $\frac{\mathrm{d}}{\mathrm{d}x}\big|_{x=0}(\varrho_C \circ \cdots \circ \varrho_C)(x) = 1$, where the composition has at most L factors. Thanks to Eq. (B.1), this shows that $\mathrm{R}_\varrho^\Omega(\Phi_C)$ is totally differentiable at 0, with $D(\mathrm{R}_\varrho^\Omega(\Phi_C))(0) = \mathrm{id}_{\mathbb{R}^d}$, as claimed.

Also by Eq. (B.1), we see that for every $j \in \{1, \dots, d\}$, $\big(\mathrm{R}_\varrho^\Omega(\Phi_C)(x)\big)_j$ is constant in all but the j-th coordinate. Additionally, if ϱ is increasing, then $s_0 > 0$, so that ϱ_C is also increasing, and hence $\big(\mathrm{R}_\varrho^\Omega(\Phi_C)\big)_j$ is increasing in the j-th coordinate, since compositions of increasing functions are increasing. Hence, $\Phi_\varepsilon^B := \Phi_C$ satisfies the desired properties.

We proceed with the second part of the proposition. We first prove the statement for $i = 1$. Let $\widetilde{\Phi}_1^C := \big((A_1', b_1'), (A_2', b_2')\big)$, where

$$A_1' := \begin{pmatrix} \frac{1}{C} & 0 \cdots 0 \end{pmatrix} \in \mathbb{R}^{1\times d}, \quad b_1' := x_0 \in \mathbb{R}^1, \quad A_2' := \frac{C}{s_0} \in \mathbb{R}^{1\times 1}, \quad b_2' := -\frac{Cr_0}{s_0} \in \mathbb{R}^1.$$

We have $\widetilde{\Phi}_1^C \in \mathcal{NN}((d, 1, 1))$. Next, define $\widetilde{\Phi}_2^C := \big((A_1'', b_1''), (A_2'', b_2'')\big)$, where

$$A_1'' := \frac{1}{C} \in \mathbb{R}^{1\times 1}, \quad b_1'' := x_0 \in \mathbb{R}^1, \quad A_2'' := \frac{C}{s_0} \in \mathbb{R}^{1\times 1}, \quad b_2'' := -\frac{Cr_0}{s_0} \in \mathbb{R}^1.$$

We have $\widetilde{\Phi}_2^C \in \mathcal{NN}((1, 1, 1))$. Setting $\widetilde{\Phi}_C := \widetilde{\Phi}_2^C \bullet \ldots \bullet \widetilde{\Phi}_2^C \bullet \widetilde{\Phi}_1^C$, where we take $L - 2$ concatenations (meaning $L - 1$ factors), yields a neural network $\widetilde{\Phi}_C \in \mathcal{NN}((d, 1, \dots, 1))$ (with L layers) such that

$$\mathrm{R}_\varrho^\Omega(\widetilde{\Phi}_C)(x) := \big(\varrho_C \circ \varrho_C \circ \cdots \circ \varrho_C\big)(x_1) \quad \text{for all} \quad x \in \Omega,$$

where ϱ_C is applied $L - 1$ times. Exactly as in the proof of the first part, this implies for $C \geq C_0$ that

$$\left|\mathrm{R}_\varrho^\Omega(\widetilde{\Phi}_C)(x) - x_1\right| \leq \varepsilon \quad \text{for all} \quad x \in \Omega.$$

Setting $\widetilde{\Phi}_{\varepsilon,1}^B := \widetilde{\Phi}_C$ and repeating the previous arguments yields the claim for $i = 1$. Permuting the columns of A_1' yields the result for arbitrary $i \in \{1, \dots, d\}$.

Now, let ϱ be increasing. Then, $s_0 > 0$, and thus ϱ_C is increasing for every $C > 0$. Since $\mathrm{R}_\varrho^\Omega(\widetilde{\Phi}_C)$ is the composition of componentwise monotonically increasing functions, the claim regarding the monotonicity follows. □

Appendix C: Proofs and Results Connected to Sect. 2

C.1. Proof of Theorem 2.1

We first establish the star-shapedness of the set of all realizations of neural networks, which is a direct consequence of the fact that the set is invariant under scalar multiplication. The following proposition provides the details.

Proposition C.1 *Let $S = (d, N_1, \dots, N_L)$ be a neural network architecture, let $\Omega \subset \mathbb{R}^d$, and let $\varrho : \mathbb{R} \to \mathbb{R}$. Then, the set $\mathcal{RNN}_\varrho^\Omega(S)$ is closed under scalar multiplication and is star-shaped with respect to the origin.*

Proof Let $f \in \mathcal{RNN}_\varrho^\Omega(S)$ and choose $\Phi := \big((A_1, b_1), \dots, (A_L, b_L)\big) \in \mathcal{NN}(S)$ satisfying $f = \mathrm{R}_\varrho^\Omega(\Phi)$. For $\lambda \in \mathbb{R}$, define $\widetilde{\Phi} := \big((A_1, b_1), \dots, (A_{L-1}, b_{L-1}), (\lambda A_L, \lambda b_L)\big)$ and observe that $\widetilde{\Phi} \in \mathcal{NN}(S)$ and furthermore $\lambda f = \mathrm{R}_\varrho^\Omega(\widetilde{\Phi}) \in \mathcal{RNN}_\varrho^\Omega(S)$. This establishes the closedness of $\mathcal{RNN}_\varrho^\Omega(S)$ under scalar multiplication.

We can choose $\lambda = 0$ in the argument above and obtain $0 \in \mathcal{RNN}_\varrho^\Omega(S)$. For every $f \in \mathcal{RNN}_\varrho^\Omega(S)$ the line $\{\lambda f : \lambda \in [0, 1]\}$ between 0 and f is contained in $\mathcal{RNN}_\varrho^\Omega(S)$, since $\mathcal{RNN}_\varrho^\Omega(S)$ is closed under scalar multiplication. We conclude that $\mathcal{RNN}_\varrho^\Omega(S)$ is star-shaped with respect to the origin. □

Our next goal is to show that $\mathcal{RNN}_\varrho^\Omega(S)$ cannot contain infinitely many linearly independent centers.

As a preparation, we prove two related results which show that the class $\mathcal{RNN}_\varrho^\Omega(S)$ is "small". The main assumption for guaranteeing this is that the activation function should be locally Lipschitz continuous.

Lemma C.2 *Let $S = (d, N_1, \dots, N_L)$ be a neural network architecture, set $N_0 := d$, and let $M \in \mathbb{N}$. Let $\varrho : \mathbb{R} \to \mathbb{R}$ be locally Lipschitz continuous. Let $\Omega \subset \mathbb{R}^d$ be compact, and let $\Lambda : C(\Omega; \mathbb{R}^{N_L}) \to \mathbb{R}^M$ be locally Lipschitz continuous, with respect to the uniform norm on $C(\Omega; \mathbb{R}^{N_L})$.*

If $M > \sum_{\ell=1}^{L} (N_{\ell-1} + 1) N_\ell$, then $\Lambda(\mathcal{RNN}_\varrho^\Omega(S)) \subset \mathbb{R}^M$ is a set of Lebesgue measure zero.

Proof Since ϱ is locally Lipschitz continuous, Proposition 4.1 (which will be proved completely independently) shows that the realization map

$$\mathrm{R}_\varrho^\Omega : \big(\mathcal{NN}(S), \|\cdot\|_{\mathcal{NN}(S)}\big) \to \big(C(\Omega; \mathbb{R}^{N_L}), \|\cdot\|_{\sup}\big)$$

is locally Lipschitz continuous. Here, the normed vector space $\mathcal{NN}(S)$ is per definition isomorphic to $\prod_{\ell=1}^{L} \big(\mathbb{R}^{N_{\ell-1} \times N_\ell} \times \mathbb{R}^{N_\ell}\big)$ and thus has dimension $D := \sum_{\ell=1}^{L} (N_{\ell-1} + 1) N_\ell$, so that there is an isomorphism $J : \mathbb{R}^D \to \mathcal{NN}(S)$.

As a composition of locally Lipschitz continuous functions, the map

$$\Gamma : \mathbb{R}^M \to \mathbb{R}^M, (x_1, \dots, x_M) \mapsto \Lambda\Big(\mathrm{R}_\varrho^\Omega\big(J(x_1, \dots, x_D)\big)\Big)$$

is locally Lipschitz continuous, and satisfies $\Lambda\big(\mathcal{RNN}_\varrho^\Omega(S)\big) = \mathrm{ran}(\Gamma) = \Gamma(\mathbb{R}^D \times \{0\}^{M-D})$. But it is well known (see for instance [2]Theorem 5.9), that a locally Lipschitz continuous function between Euclidean spaces of the same dimension maps sets of Lebesgue measure zero to sets of Lebesgue measure zero. Hence, $\Lambda(\mathcal{RNN}_\varrho^\Omega(S)) \subset \mathbb{R}^M$ is a set of Lebesgue measure zero. □

As a corollary, we can now show that the class of neural network realizations cannot contain a subspace of large dimension.

Corollary C.3 *Let $S = (d, N_1, \dots, N_L)$ be a neural network architecture, set $N_0 := d$, and let $\varrho : \mathbb{R} \to \mathbb{R}$ be locally Lipschitz continuous.*

Let $\varnothing \neq \Omega \subset \mathbb{R}^d$ be arbitrary. If $V \subset C(\Omega; \mathbb{R}^{N_L})$ is a vector space with $V \subset \mathcal{RNN}_\varrho^\Omega(S)$, then $\dim V \leq \sum_{\ell=1}^{L} (N_{\ell-1} + 1) N_\ell$.

Proof Let $D := \sum_{\ell=1}^{L} (N_{\ell-1} + 1) N_\ell$. Assume toward a contradiction that the claim of the corollary does not hold; then there exists a subspace $V \subset C(\Omega; \mathbb{R}^{N_L})$ of dimension $\dim V = D + 1$ with $V \subset \mathcal{RNN}_\varrho^\Omega(S)$. For $x \in \Omega$ and $\ell \in \underline{N_L}$, let $\delta_x^{(\ell)} : C(\Omega; \mathbb{R}^{N_L}) \to \mathbb{R}, f \mapsto \big(f(x)\big)_\ell$. Define $W := \operatorname{span}\big\{\delta_x^{(\ell)}|_V : x \in \Omega, \ell \in \underline{N_L}\big\}$, and note that W is a subspace of the finite-dimensional algebraic dual space V^* of V. In particular, $\dim W \leq \dim V^* = \dim V = D + 1$, so that there are $(x_1, \ell_1), \dots, (x_{D+1}, \ell_{D+1}) \in \Omega \times \underline{N_L}$ such that $W = \operatorname{span}\big\{\delta_{x_k}^{(\ell_k)} : k \in \underline{D+1}\big\}$.

We claim that the linear map

$$\Lambda_0 : V \to \mathbb{R}^{D+1}, f \mapsto \big([f(x_k)]_{\ell_k}\big)_{k \in \underline{D+1}}$$

is surjective. Since $\dim V = D + 1 = \dim \mathbb{R}^{D+1}$, it suffices to show that Λ_0 is injective. But if $\Lambda_0 f = 0$ for some $f \in V \subset C(\Omega; \mathbb{R}^{N_L})$, and if $x \in \Omega$ and $\ell \in \underline{N_L}$ are arbitrary, then $\delta_x^{(\ell)} = \sum_{k=1}^{D+1} a_k \, \delta_{x_k}^{(\ell_k)}$ for certain $a_1, \dots, a_{D+1} \in \mathbb{R}$. Hence, $[f(x)]_\ell = \sum_{k=1}^{D+1} a_k [f(x_k)]_{\ell_k} = 0$. Since $x \in \Omega$ and $\ell \in \underline{N_L}$ were arbitrary, this means $f \equiv 0$. Therefore, Λ_0 is injective and thus surjective.

Now, let us define $\Omega' := \{x_1, \dots, x_{D+1}\}$, and note that $\Omega' \subset \mathbb{R}^d$ is compact. Set $M := D + 1$, and define

$$\Lambda : C(\Omega', \mathbb{R}^{N_L}) \to \mathbb{R}^M, f \mapsto \big([f(x_k)]_{\ell_k}\big)_{k \in \underline{D+1}}.$$

It is straightforward to verify that Λ is Lipschitz continuous. Therefore, Lemma C.2 shows that the set $\Lambda(\mathcal{RNN}_\varrho^{\Omega'}(S)) \subset \mathbb{R}^M$ is a null-set. However,

$$\Lambda\big(\mathcal{RNN}_\varrho^{\Omega'}(S)\big) = \Lambda\big(\{f|_{\Omega'} : f \in \mathcal{RNN}_\varrho^\Omega(S)\}\big) \supset \Lambda\big(\{f|_{\Omega'} : f \in V\}\big) = \Lambda_0(V) = \mathbb{R}^M.$$

This yields the desired contradiction. □

Now, the announced estimate for the number of linearly independent centers of the set of all network realizations of a fixed size is a direct consequence.

Proposition C.4 *Let $S = (d, N_1, \dots, N_L)$ be a neural network architecture, let $\Omega \subset \mathbb{R}^d$, and let $\varrho : \mathbb{R} \to \mathbb{R}$ be locally Lipschitz continuous. Then, $\mathcal{RNN}_\varrho^\Omega(S)$ contains at most $\sum_{\ell=1}^{L} (N_{\ell-1} + 1) N_\ell$ linearly independent centers, where $N_0 = d$. That is, the number of linearly independent centers is bounded by the total number of parameters of the underlying neural networks.*

Proof Let us set $D := \sum_{\ell=1}^{L} (N_{\ell-1} + 1) N_\ell$, and assume toward a contradiction that $\mathcal{RNN}_\varrho^\Omega(S)$ contains $M := D + 1$ linearly independent centers $\mathrm{R}_\varrho^\Omega(\Phi_1), \dots, \mathrm{R}_\varrho^\Omega(\Phi_M)$.

Since $\mathcal{RNN}_\varrho^\Omega(S)$ is closed under multiplication with scalars, this implies

$$V := \operatorname{span}\left\{\mathrm{R}_\varrho^\Omega(\Phi_1), \dots, \mathrm{R}_\varrho^\Omega(\Phi_M)\right\} \subset \mathcal{RNN}_\varrho^\Omega(S).$$

Indeed, this follows by induction on M, using the following observation: If V is a vector space contained in a set A, if A is closed under multiplication with scalars, and if $x_0 \in A$ is a center for A, then $V + \operatorname{span}\{x_0\} \subset A$. To see this, let $\mu \in \mathbb{R}$ and $v \in V$. There is some $\varepsilon \in \{1, -1\}$ such that $\varepsilon\mu = |\mu|$. Now set $x := \varepsilon v \in V \subset A$ and $\lambda := |\mu|/(1 + |\mu|) \in [0, 1]$. Then,

$$\begin{aligned} v + \mu\, x_0 &= \varepsilon \cdot (\varepsilon v + |\mu| x_0) = \varepsilon \cdot (1 + |\mu|) \cdot \left(\frac{1}{1 + |\mu|} x + \frac{|\mu|}{1 + |\mu|} x_0\right) \\ &= \varepsilon \cdot (1 + |\mu|) \cdot \big(\lambda x_0 + (1 - \lambda) x\big) \in A. \end{aligned}$$

Since the family $\left(\mathrm{R}_\varrho^\Omega(\Phi_k)\right)_{k \in \underline{M}}$ is linearly independent, we see

$$\dim V = M > D = \sum_{\ell=1}^{L} (N_{\ell-1} + 1) N_\ell.$$

In view of Corollary C.3, this yields the desired contradiction. □

Next, we analyze the convexity of $\mathcal{RNN}_\varrho^\Omega(S)$. As a direct consequence of Proposition C.4, we see that $\mathcal{RNN}_\varrho^\Omega(S)$ is never convex if $\mathcal{RNN}_\varrho^\Omega(S)$ contains more than a certain number of linearly independent functions.

Corollary C.5 *Let $S = (d, N_1, \dots, N_L)$ be a neural network architecture and let $N_0 := d$. Let $\Omega \subset \mathbb{R}^d$, and let $\varrho : \mathbb{R} \to \mathbb{R}$ be locally Lipschitz continuous.*

If $\mathcal{RNN}_\varrho^\Omega(S)$ contains more than $\sum_{\ell=1}^{L} (N_{\ell-1} + 1) N_\ell$ linearly independent functions, then $\mathcal{RNN}_\varrho^\Omega(S)$ is not convex.

Proof Every element of a convex set is a center. Thus the result follows directly from Proposition C.4. □

Corollary C.5 claims that if a set of realizations of neural networks with fixed size contains more than a fixed number of linearly independent functions, then it cannot be convex. Since $\mathcal{RNN}_\varrho^{\mathbb{R}^d}(S)$ is translation invariant, it is very likely that $\mathcal{RNN}_\varrho^{\mathbb{R}^d}(S)$ (and hence also $\mathcal{RNN}_\varrho^\Omega(S)$) contains *infinitely* many linearly independent functions. In fact, our next result shows under minor regularity assumptions on ϱ that if the set $\mathcal{RNN}_\varrho^\Omega(S)$ *does not* contain infinitely many linearly independent functions, then ϱ is necessarily a polynomial.

Proposition C.6 *Let $S = (d, N_1, \dots, N_L)$ be a neural network architecture with $L \in \mathbb{N}_{\geq 2}$. Moreover, let $\varrho : \mathbb{R} \to \mathbb{R}$ be continuous. Assume that there exists $x_0 \in \mathbb{R}$ such that ϱ is differentiable at x_0 with $\varrho'(x_0) \neq 0$.*

Further assume that $\Omega \subset \mathbb{R}^d$ has nonempty interior, and that $\mathcal{RNN}_\varrho^\Omega(S)$ does not *contain infinitely many linearly independent functions. Then, ϱ is a polynomial.*

Proof *Step 1* Set $S' := (d, N_1, \dots, N_{L-1}, 1)$. We first show that $\mathcal{RNN}_{\varrho}^{\Omega}(S')$ does not contain infinitely many linearly independent functions. To see this, first note that the map

$$\Theta : \mathcal{RNN}_{\varrho}^{\Omega}(S) \to \mathcal{RNN}_{\varrho}^{\Omega}(S'),\ f \mapsto f_1,$$

which maps an $\mathbb{R}^{N_L}$-valued function to its first component, is linear, well-defined, and *surjective*.

Hence, if there were infinitely many linearly independent functions $(f_n)_{n\in\mathbb{N}}$ in the set $\mathcal{RNN}_{\varrho}^{\Omega}(S')$, then we could find $(g_n)_{n\in\mathbb{N}}$ in $\mathcal{RNN}_{\varrho}^{\Omega}(S)$ such that $f_n = \Theta\, g_n$. But then the $(g_n)_{n\in\mathbb{N}}$ are necessarily linearly independent, contradicting the hypothesis of the theorem.

Step 2 We show that $\mathcal{G} := \mathcal{RNN}_{\varrho}^{\mathbb{R}^d}(S')$ does not contain infinitely many linearly independent functions.

To see this, first note that since $\mathcal{F} := \mathcal{RNN}_{\varrho}^{\Omega}(S')$ does not contain infinitely many linearly independent functions (Step 1), elementary linear algebra shows that there is a finite-dimensional subspace $V \subset C(\Omega;\mathbb{R})$ satisfying $\mathcal{F} \subset V$. Let $D := \dim V$, and assume toward a contradiction that there are $D+1$ linearly independent functions $f_1, \dots, f_{D+1} \in \mathcal{G}$, and set $W := \operatorname{span}\{f_1, \dots, f_{D+1}\} \subset C(\mathbb{R}^d;\mathbb{R})$. The space $\Gamma := \operatorname{span}\{\delta_x|_W : x \in \mathbb{R}^d\} \subset W^*$ spanned by the point evaluation functionals $\delta_x : C(\mathbb{R}^d;\mathbb{R}) \to \mathbb{R},\ f \mapsto f(x)$ is finite-dimensional with $\dim \Gamma \leq \dim W^* = \dim W = D+1$. Hence, there are $x_1, \dots, x_{D+1} \in \mathbb{R}^d$ such that $\Gamma = \operatorname{span}\{\delta_{x_1}|_W, \dots, \delta_{x_{D+1}}|_W\}$.

We claim that the map

$$\Theta : W \to \mathbb{R}^{D+1},\ f \mapsto \big(f(x_\ell)\big)_{\ell\in\underline{D+1}}$$

is surjective. Since $\dim W = D+1$, it suffices to show that Θ is injective. If this was not true, there would be some $f \in W \subset C(\mathbb{R}^d;\mathbb{R})$, $f \not\equiv 0$ such that $\Theta f = 0$. But since $f \not\equiv 0$, there is some $x_0 \in \mathbb{R}^d$ satisfying $f(x_0) \neq 0$. Because of $\delta_{x_0}|_W \in \Gamma$, we have $\delta_{x_0}|_W = \sum_{\ell=1}^{D+1} a_\ell\, \delta_{x_\ell}|_W$ for certain $a_1, \dots, a_{D+1} \in \mathbb{R}$. Hence, $0 \neq f(x_0) = \delta_{x_0}|_W(f) = \sum_{\ell=1}^{D+1} a_\ell\, \delta_{x_\ell}|_W(f) = 0$, since $f(x_\ell) = \big(\Theta(f)\big)_\ell = 0$ for all $\ell \in \underline{D+1}$. This contradiction shows that Θ is injective, and hence surjective.

Now, since Ω has nonempty interior, there is some $b \in \Omega$ and some $r > 0$ such that $y_\ell := b + r\, x_\ell \in \Omega$ for all $\ell \in \underline{D+1}$. Define

$$g_\ell : \mathbb{R}^d \to \mathbb{R},\ y \mapsto f_\ell\left(\frac{y}{r} - \frac{b}{r}\right) \quad \text{for} \quad \ell \in \underline{D+1}.$$

It is not hard to see $g_\ell \in \mathcal{G}$, and hence $g_\ell|_\Omega \in \mathcal{F} \subset V$ for all $\ell \in \underline{D+1}$. Now, define the linear operator $\Lambda : V \to \mathbb{R}^{D+1},\ f \mapsto \big(f(y_\ell)\big)_{\ell\in\underline{D+1}}$, and note that $\Lambda(g_\ell) = \big(g_\ell(y_k)\big)_{k\in\underline{D+1}} = \big(f_\ell(x_k)\big)_{k\in\underline{D+1}} = \Theta(f_\ell)$, because of $y_\ell/r - b/r = x_\ell$. Since the functions $f_1, \dots, f_{D+1}$ span the space W, this implies $\Lambda(V) \supset \Theta(W) = \mathbb{R}^{D+1}$, in contradiction to Λ being linear and $\dim V = D < D+1$. This contradiction shows that $\mathcal{G}$ does not contain infinitely many linearly independent functions.

Step 3 From the previous step, we know that $\mathcal{G} = \mathcal{RNN}_{\varrho}^{\mathbb{R}^d}(S')$ does not contain infinitely many linearly independent functions. In this step, we show that this implies that the activation function ϱ is a polynomial.

To this end, define

$$\mathcal{RNN}_{S',\varrho}^{*} := \left\{ f : \mathbb{R} \to \mathbb{R} \;\middle|\; \begin{array}{l} \text{there is some } g \in \mathcal{RNN}_{\varrho}^{\mathbb{R}^d}(S') \\ \text{with } f(x) = g(x, 0, \dots, 0) \;\text{ for all }\; x \in \mathbb{R} \end{array} \right\}.$$

Clearly, $\mathcal{RNN}_{S',\varrho}^{*}$ is dilation- and translation invariant; that is, if $f \in \mathcal{RNN}_{S',\varrho}^{*}$, then also $f(a\,\cdot) \in \mathcal{RNN}_{S',\varrho}^{*}$ and $f(\cdot - x) \in \mathcal{RNN}_{S',\varrho}^{*}$ for arbitrary $a > 0$ and $x \in \mathbb{R}$. Furthermore, by Step 2, we see that $\mathcal{RNN}_{S',\varrho}^{*}$ does not contain infinitely many linearly independent functions. Therefore, $V := \operatorname{span} \mathcal{RNN}_{S',\varrho}^{*}$ is a finite-dimensional translation- and dilation invariant subspace of $C(\mathbb{R})$. Thanks to the translation invariance, it follows from [3] that there exists some $r \in \mathbb{N}$, and certain $\lambda_j \in \mathbb{C}$, $k_j \in \mathbb{N}_0$ for $j = 1, \dots, r$ such that

$$\mathcal{RNN}_{S',\varrho}^{*} \subset V \subset \operatorname{span}_{\mathbb{C}} \left\{ x \mapsto x^{k_j} e^{\lambda_j x} : j = 1, \dots, r \right\}, \tag{C.1}$$

where $\operatorname{span}_{\mathbb{C}}$ denotes the linear span, with $\mathbb{C}$ as the underlying field. Clearly, we can assume $(k_j, \lambda_j) \neq (k_\ell, \lambda_\ell)$ for $j \neq \ell$.

Step 4 Let $N := \max_{j \in \{1,\dots,r\}} k_j$. We now claim that V is contained in the space $\mathbb{C}_{\deg \leq N}[X]$ of (complex) polynomials of degree at most N.

Indeed, suppose toward a contradiction that there is some $f \in V \setminus \mathbb{C}_{\deg \leq N}[X]$. Thanks to (C.1), we can write $f = \sum_{j=1}^{r} a_j\, x^{k_j}\, e^{\lambda_j x}$ with $a_1, \dots, a_r \in \mathbb{C}$. Because of $f \notin \mathbb{C}_{\deg \leq N}[X]$, there is some $\ell \in \{1, \dots, r\}$ such that $a_\ell \neq 0$ and $\lambda_\ell \neq 0$. Now, choose $\beta > 0$ such that $|\beta \lambda_\ell| > |\lambda_j|$ for all $j \in \{1, \dots, r\}$, and note that $f(\beta\,\cdot) \in V$, so that Eq. (C.1) yields coefficients $b_1, \dots, b_r \in \mathbb{C}$ such that $f(\beta\, x) = \sum_{j=1}^{r} b_j\, x^{k_j}\, e^{\lambda_j x}$. By subtracting the two different representations for $f(\beta\, x)$, we thus see

$$0 \equiv f(\beta x) - f(\beta x) = \sum_{j=1}^{r} a_j\, \beta^{k_j}\, x^{k_j}\, e^{\beta \lambda_j x} - \sum_{j=1}^{r} b_j\, x^{k_j}\, e^{\lambda_j x},$$

and hence

$$x^{k_\ell} e^{\beta \lambda_\ell x} = \frac{1}{a_\ell\, \beta^{k_\ell}} \cdot \Big(\sum_{j=1}^{r} b_j\, x^{k_j}\, e^{\lambda_j x} - \sum_{j \in \{1,\dots,r\} \setminus \{\ell\}} a_j\, \beta^{k_j}\, x^{k_j}\, e^{\beta \lambda_j x} \Big). \tag{C.2}$$

Note, however, that $|\beta \lambda_\ell| > |\lambda_j|$ and hence $(k_\ell, \beta\lambda_\ell) \neq (k_j, \lambda_j)$ for all $j \in \{1, \dots, r\}$, and furthermore that $(k_\ell, \beta\lambda_\ell) \neq (k_j, \beta\lambda_j)$ for $j \in \{1, \dots, r\} \setminus \{\ell\}$. Thus, Lemma C.7 below shows that Eq. (C.2) cannot be true. This is the desired contradiction.

Step 5 In this step, we complete the proof, by first showing for arbitrary $B > 0$ that $\varrho|_{[-B,B]}$ is a polynomial of degree at most N.

Let $\varepsilon, B > 0$ be arbitrary. Since ϱ is continuous, it is uniformly continuous on $[-B-1, B+1]$, that is, there is some $\delta \in (0, 1)$ such that $|\varrho(x) - \varrho(y)| \leq \varepsilon$ for all

$x, y \in [-B-1, B+1]$ with $|x-y| \leq \delta$. Since $\varrho'(x_0) \neq 0$ and $L \geq 2$, Proposition B.3 and Lemma B.1 imply existence of a neural network $\widetilde{\Phi}_{\varepsilon,B} \in \mathcal{NN}((d, N_1, \dots, N_{L-1}))$ such that

$$\left| \left[\mathrm{R}_{\varrho}^{[-B,B]^d}(\widetilde{\Phi}_{\varepsilon,B})(x) \right]_1 - x_1 \right| \leq \delta, \text{ for all } x \in [-B, B]^d.$$

In particular, this implies because of $\delta \leq 1$ that $\left[\mathrm{R}_{\varrho}^{[-B,B]^d}(\widetilde{\Phi}_{\varepsilon,B})(x) \right]_1 \in [-B-1, B+1]$ for all $x \in [-B, B]^d$. We conclude that

$$\left| \left[\varrho\left(\mathrm{R}_{\varrho}^{[-B,B]^d}(\widetilde{\Phi}_{\varepsilon,B})(x) \right) \right]_1 - \varrho(x_1) \right| \leq \varepsilon, \quad \text{for all} \quad x \in [-B, B]^d, \tag{C.3}$$

with ϱ acting componentwise. By (C.3), it follows that there is some $\Phi_{\varepsilon,B} \in \mathcal{NN}(S')$ satisfying

$$\left| \mathrm{R}_{\varrho}^{[-B,B]^d}(\Phi_{\varepsilon,B})(x_1, 0, \dots, 0) - \varrho(x_1) \right| \leq \varepsilon \quad \text{for all } x_1 \in [-B, B]. \tag{C.4}$$

From (C.4) and Step 4, we thus see

$$\varrho|_{[-B,B]} \in \overline{\left\{ f|_{[-B,B]} : f \in \mathcal{RNN}^*_{S',\varrho} \right\}} \subset \{ p|_{[-B,B]} : p \in \mathbb{C}_{\deg \leq N}[X] \},$$

where the closure is taken with respect to the sup norm, and where we implicitly used that the space on the right-hand side is a closed subspace of $C([-B, B])$, since it is a finite dimensional subspace.

Since $\varrho|_{[-B,B]}$ is a polynomial of degree at most N; we see that the $N+1$-th derivative of ϱ satisfies $\varrho^{(N+1)} \equiv 0$ on $(-B, B)$, for arbitrary $B > 0$. Thus, $\varrho^{(N+1)} \equiv 0$, meaning that ϱ is a polynomial. □

In the above proof, we used the following elementary lemma, whose proof we provide for completeness.

Lemma C.7 *For $k \in \mathbb{N}_0$ and $\lambda \in \mathbb{C}$, define $f_{k,\lambda} : \mathbb{R} \to \mathbb{C}, x \mapsto x^k e^{\lambda x}$.*

Let $N \in \mathbb{N}$, and let $(k_1, \lambda_1), \dots, (k_N, \lambda_N) \in \mathbb{N}_0 \times \mathbb{C}$ satisfy $(k_\ell, \lambda_\ell) \neq (k_j, \lambda_j)$ for $\ell \neq j$. Then, the family $(f_{k_\ell,\lambda_\ell})_{\ell=1,\dots,N}$ is linearly independent over $\mathbb{C}$.

Proof Let us assume toward a contradiction that

$$0 \equiv \sum_{\ell=1}^{N} a_\ell \, f_{k_\ell,\lambda_\ell}(x) = \sum_{\ell=1}^{N} a_\ell \, x^{k_\ell} \, e^{\lambda_\ell x} \tag{C.5}$$

for some coefficient vector $(a_1, \dots, a_N) \in \mathbb{C}^N \setminus \{0\}$. By dropping those terms for which $a_\ell = 0$, we can assume that $a_\ell \neq 0$ for all $\ell \in \{1, \dots, N\}$.

Let $\Lambda := \{\lambda_i : i \in \{1, \dots, N\}\}$. In the case where $|\Lambda| = 1$, it follows that $k_j \neq k_\ell$ for $j \neq \ell$. Furthermore, multiplying Eq. (C.5) by $e^{-\lambda_1 x}$, we see that $0 \equiv \sum_{\ell=1}^{N} a_\ell \, x^{k_\ell}$, which is impossible since the monomials $(x^k)_{k \in \mathbb{N}_0}$ are linearly independent. Thus, we only need to consider the case that $|\Lambda| > 1$.

Define $M := \max\{k_\ell : \ell \in \{1, \dots, N\}\}$ and

$$I := \big\{\ell \in \{1, \dots, N\} : \lambda_\ell = \lambda_1\big\}, \qquad \text{and choose } j \in I \text{ satisfying} \qquad k_j = \max_{\ell \in I} k_\ell.$$

Note that this implies $k_\ell < k_j$ for all $\ell \in I \setminus \{j\}$, since $(k_\ell, \lambda_\ell) \neq (k_j, \lambda_j)$ and hence $k_\ell \neq k_j$ for $\ell \in I \setminus \{j\}$.

Consider the differential operator

$$T := \prod_{\lambda \in \Lambda \setminus \{\lambda_1\}} \Big(\frac{d}{dx} - \lambda \,\mathrm{id}\Big)^{M+1}$$

Note that $\big(\frac{\mathrm{d}}{\mathrm{d}x} - \lambda\,\mathrm{id}\big)(x^k\, e^{\mu x}) = (\mu - \lambda) x^k\, e^{\mu x} + k\, x^{k-1}\, e^{\mu x}$. Using this identity, it is easy to see that if $\lambda \in \Lambda \setminus \{\lambda_1\}$ and $k \in \mathbb{N}_0$ satisfies $k \leq M$, then $T(x^k\, e^{\lambda x}) \equiv 0$. Furthermore, for each $k \in \mathbb{N}_0$ with $k \leq M$, there exist a constant $c_k \in \mathbb{C} \setminus \{0\}$ and a polynomial $p_k \in \mathbb{C}[X]$ with $\deg p_k < k$ satisfying $T(x^k\, e^{\lambda_1 x}) = e^{\lambda_1 x} \cdot (c_k x^k + p_k(x))$. Overall, Eq. (C.5) implies that

$$\begin{aligned} 0 \equiv e^{-\lambda_1 x} \cdot T\Big(\sum_{\ell=1}^{N} a_\ell\, x^{k_\ell}\, e^{\lambda_\ell x}\Big) &= a_j c_{k_j} x^{k_j} + a_j\, p_{k_j}(x) \\ &\quad + \sum_{\ell \in I \setminus \{j\}} \big[a_\ell \cdot (c_{k_\ell} x^{k_\ell} + p_{k_\ell}(x))\big] \\ &=: a_j c_{k_j} x^{k_j} + q(x), \end{aligned}$$

where $a_j c_{k_j} \neq 0$ and where $\deg q < k_j$, since $k_j > k_\ell$ for all $\ell \in I \setminus \{j\}$. This is the desired contradiction. □

As our final ingredient for the proof of Theorem 2.1, we show that every non-constant locally Lipschitz function ϱ satisfies the technical assumptions of Proposition C.6.

Lemma C.8 *Let $\varrho : \mathbb{R} \to \mathbb{R}$ be locally Lipschitz continuous and not constant. Then, there exists some $x_0 \in \mathbb{R}$ such that ϱ is differentiable at x_0 with $\varrho'(x_0) \neq 0$.*

Proof Since ϱ is not constant, there is some $B > 0$ such that $\varrho|_{[-B,B]}$ is not constant. By assumption, ϱ is Lipschitz continuous on $[-B, B]$. Thus, $\varrho|_{[-B,B]}$ is absolutely continuous; see for instance [60, Definition 7.17]. Thanks to the fundamental theorem of calculus for the Lebesgue integral (see [60, Theorem 7.20]), this implies that $\varrho|_{[-B,B]}$ is differentiable almost everywhere on $(-B, B)$ and satisfies $\varrho(y) - \varrho(x) = \int_x^y \varrho'(t)\, dt$ for $-B \leq x < y \leq B$, where $\varrho'(t) := 0$ if ϱ is not differentiable at t.

Since $\varrho|_{[-B,B]}$ is not constant, the preceding formula shows that there has to be some $x_0 \in (-B, B)$ such that $\varrho'(x_0) \neq 0$; in particular, this means that ϱ is differentiable at x_0. □

Now, a combination of Corollary C.5, Proposition C.6, and Lemma C.8 proves Theorem 2.1. For the application of Lemma C.8, note that if ϱ is constant, then ϱ is a polynomial, so that the conclusion of Theorem 2.1 also holds in this case.

C.2. Proof of Theorem 2.2

We first show in the following lemma that if $\overline{\mathcal{RNN}_\varrho^\Omega(S)}$ is convex, then $\mathcal{RNN}_\varrho^\Omega(S)$ is dense in $C(\Omega)$. The proof of Theorem 2.2 is given thereafter.

Lemma C.9 *Let $S = (d, N_1, \dots, N_{L-1}, 1)$ be a neural network architecture with $L \geq 2$. Let $\Omega \subset \mathbb{R}^d$ be compact and let $\varrho : \mathbb{R} \to \mathbb{R}$ be continuous but not a polynomial. Finally, assume that there is some $x_0 \in \mathbb{R}$ such that ϱ is differentiable at x_0 with $\varrho'(x_0) \neq 0$.*

If $\overline{\mathcal{RNN}_\varrho^\Omega(S)}$ is convex, then $\mathcal{RNN}_\varrho^\Omega(S)$ is dense in $C(\Omega)$.

Proof Since $\overline{\mathcal{RNN}_\varrho^\Omega(S)}$ is convex and closed under scalar multiplication, $\overline{\mathcal{RNN}_\varrho^\Omega(S)}$ forms a closed linear subspace of $C(\Omega)$. Below, we will show that $\big(\Omega \to \mathbb{R}, x \mapsto \varrho(\langle a, x\rangle + b)\big) \in \overline{\mathcal{RNN}_\varrho^\Omega(S)}$ for arbitrary $a \in \mathbb{R}^d$ and $b \in \mathbb{R}$. Once we prove this, it follows that $\big(\Omega \to \mathbb{R}, x \mapsto \sum_{i=1}^N c_i\, \varrho(b_i + \langle a_i, x\rangle)\big) \in \overline{\mathcal{RNN}_\varrho^\Omega(S)}$ for arbitrary $N \in \mathbb{N}$, $a_i \in \mathbb{R}^d$, and $b_i, c_i \in \mathbb{R}$. As shown in [45], this then entails that $\overline{\mathcal{RNN}_\varrho^\Omega(S)}$ (and hence also $\mathcal{RNN}_\varrho^\Omega(S)$) is dense in $C(\Omega)$, since ϱ is not a polynomial.

Thus, let $a \in \mathbb{R}^d$, $b \in \mathbb{R}$, and $\varepsilon > 0$ be arbitrary, and define $g : \Omega \to \mathbb{R}, x \mapsto \varrho(b + \langle a, x\rangle)$ and $\Psi := \big((a^T, b), (1, 1)\big) \in \mathcal{NN}((d, 1, 1))$, noting that $\mathrm{R}_\varrho^\Omega(\Psi) = g$. Since g is continuous on the compact set Ω, we have $|g(x)| \leq B$ for all $x \in \Omega$ and some $B > 0$. By Proposition B.3 and since $\varrho'(x_0) \neq 0$ and $L \geq 2$, there exists a neural network $\Phi_\varepsilon \in \mathcal{NN}((1, \dots, 1))$ (with $L - 1$ layers) such that $|\mathrm{R}_\varrho^{[-B,B]}(\Phi_\varepsilon) - x| \leq \varepsilon$ for all $x \in [-B, B]$. This easily shows $\|\mathrm{R}_\varrho^\Omega(\Phi_\varepsilon \bullet \Psi) - g\|_{\sup} \leq \varepsilon$, while $\mathrm{R}_\varrho^\Omega(\Phi_\varepsilon \bullet \Psi) \in \mathcal{RNN}_\varrho^\Omega((d, 1, \dots, 1)) \subset \mathcal{RNN}_\varrho^\Omega(S)$ by Lemma B.1. Therefore, $g \in \overline{\mathcal{RNN}_\varrho^\Omega(S)}$, which completes the proof. □

Now we are ready to prove Theorem 2.2. By assumption, $\mathcal{RNN}_\varrho^\Omega(S)$ is *not* dense in $C(\Omega)$. We start by proving that there exists at least one $\varepsilon > 0$ such that the set $\overline{\mathcal{RNN}_\varrho^\Omega(S)}$ is not ε-convex. Suppose toward a contradiction that this is not true, so that $\overline{\mathcal{RNN}_\varrho^\Omega(S)}$ is ε-convex for *all* $\varepsilon > 0$. This implies

$$\mathrm{co}\left(\overline{\mathcal{RNN}_\varrho^\Omega(S)}\right) \subset \bigcap_{\varepsilon > 0} \left(\overline{\mathcal{RNN}_\varrho^\Omega(S)} + B_\varepsilon(0)\right) = \overline{\mathcal{RNN}_\varrho^\Omega(S)}, \tag{C.6}$$

where the last identity holds true, since if $\widetilde{f} \notin \overline{\mathcal{RNN}_\varrho^\Omega(S)}$, there exists $\varepsilon' > 0$ such that $\|\widetilde{f} - f\|_{\sup} > \varepsilon'$ for all $f \in \overline{\mathcal{RNN}_\varrho^\Omega(S)}$. Equation (C.6) shows that $\overline{\mathcal{RNN}_\varrho^\Omega(S)}$ is convex, which by Lemma C.9 implies that $\mathcal{RNN}_\varrho^\Omega(S) \subset C(\Omega)$ is dense, in contradiction to the assumptions of Theorem 2.2. This is the desired contradiction, showing that $\overline{\mathcal{RNN}_\varrho^\Omega(S)}$ is not ε-convex for *some* $\varepsilon > 0$.

Thus, there exists $g \in \mathrm{co}\big(\overline{\mathcal{RNN}_\varrho^\Omega(S)}\big)$ such that $\|g - f\|_{\sup} \geq \varepsilon_0$ for all $f \in \overline{\mathcal{RNN}_\varrho^\Omega(S)}$. Now, let $\varepsilon > 0$ be arbitrary. Then, $\frac{\varepsilon}{\varepsilon_0} g \in \mathrm{co}(\overline{\mathcal{RNN}_\varrho^\Omega(S)})$, since

$\mathcal{RNN}_\varrho^\Omega(S)$ is closed under scalar multiplication. Moreover,

$$\left\| \frac{\varepsilon}{\varepsilon_0} g - f \right\|_{\sup} \geq \varepsilon \quad \text{for all} \quad f \in \overline{\mathcal{RNN}_\varrho^\Omega(S)},$$

again due to the closedness under scalar multiplication of $\mathcal{RNN}_\varrho^\Omega(S)$. This shows that $\overline{\mathcal{RNN}_\varrho^\Omega(S)}$ is not ε-convex for any $\varepsilon > 0$. □

C.3. Non-dense Network Sets

In this section, we review criteria on ϱ which ensure that $\overline{\mathcal{RNN}_\varrho^\Omega(S)} \neq C(\Omega)$.

Precisely, we will show that this is true if $\varrho : \mathbb{R} \to \mathbb{R}$ is **computable by elementary operations**, which means that there is some $N \in \mathbb{N}$ and an algorithm that takes $x \in \mathbb{R}$ as input and returns $\varrho(x)$ after no more than N of the following operations:

- applying the exponential function $\exp : \mathbb{R} \to \mathbb{R}$;
- applying one of the arithmetic operations $+, -, \times$, and $/$ on real numbers;
- jumps conditioned on comparisons of real numbers using the following operators: $<, >, \leq, \geq, =, \neq$.

Then, a combination of [4, Theorem 14.1] with [4, Theorem 8.14] shows that if ϱ is computable by elementary operations, then the *pseudo-dimension* of each of the function classes $\mathcal{RNN}_\varrho^{\mathbb{R}^d}(S)$ is finite. Here, the pseudo-dimension $\operatorname{Pdim}(\mathcal{F})$ of a function class $\mathcal{F} \subset \mathbb{R}^X$ is defined as follows (see [4, Sect. 11.2]):

$$\operatorname{Pdim}(\mathcal{F}) := \sup \left\{ |K| : K \subset X \text{ finite and pseudo-shattered by } \mathcal{F} \right\} \in \mathbb{N} \cup \{\infty\}.$$

Here, a finite set $K = \{x_1, \ldots, x_m\} \subset X$ (with pairwise distinct x_i) is *pseudo-shattered* by $\mathcal{F}$ if there are $r_1, \ldots, r_m \in \mathbb{R}$ such that for each $b \in \{0,1\}^m$ there is a function $f_b \in \mathcal{F}$ with $\mathbb{1}_{[0,\infty)}\big(f_b(x_i) - r_i\big) = b_i$ for all $i \in \{1, \ldots, m\}$.

Using this result, we can now show that the realization sets of networks with activation functions that are computable by elementary operations are never dense in $L^p(\Omega)$ or $C(\Omega)$.

Proposition C.10 *Let $\varrho : \mathbb{R} \to \mathbb{R}$ be continuous and computable by elementary operations. Moreover, let $S = (d, N_1, \ldots, N_{L-1}, 1)$ be a neural network architecture. Let $\Omega \subset \mathbb{R}^d$ be any measurable set with nonempty interior, and let $\mathcal{Y}$ denote either $L^p(\Omega)$ (for some $p \in [1, \infty)$), or $C(\Omega)$. In case of $\mathcal{Y} = C(\Omega)$, assume additionally that Ω is compact.*

Then. we have $\overline{\mathcal{Y} \cap \mathcal{RNN}_\varrho^\Omega(S)} \subsetneq \mathcal{Y}$.

Proof The considerations from before the statement of the proposition show that

$$\operatorname{Pdim}\big(\mathcal{Y} \cap \mathcal{RNN}_\varrho^\Omega(S)\big) \leq \operatorname{Pdim}\big(\mathcal{RNN}_\varrho^{\mathbb{R}^d}(S)\big) < \infty.$$

Therefore, all we need to show is that if $\mathcal{F} \subset C(\Omega)$ is a function class for which $\mathcal{F} \cap \mathcal{Y}$ is dense in $\mathcal{Y}$, then $\operatorname{Pdim}(\mathcal{F}) = \infty$.

For $\mathcal{Y} = C(\Omega)$, this is easy: Let $m \in \mathbb{N}$ be arbitrary, choose distinct points $x_1, \dots, x_m \in \Omega$, and note that for each $b \in \{0,1\}^m$, there is $g_b \in C(\Omega)$ satisfying $g_b(x_j) = b_j$ for all $j \in \underline{m}$. By density, for each $b \in \{0,1\}^m$, there is $f_b \in \mathcal{F}$ such that $\|f_b - g_b\|_{\sup} < \frac{1}{2}$. In particular, $f_b(x_j) > \frac{1}{2}$ if $b_j = 1$ and $f_b(x_j) < \frac{1}{2}$ if $b_j = 0$. Thus, if we set $r_1 := \ldots := r_m := \frac{1}{2}$, then $\mathbb{1}_{[0,\infty)}(f_b(x_j) - r_j) = b_j$ for all $j \in \underline{m}$. Hence, $S = \{x_1, \dots, x_m\}$ is pseudo-shattered by $\mathcal{F}$, so that $\operatorname{Pdim}(\mathcal{F}) \geq m$. Since $m \in \mathbb{N}$ was arbitrary, $\operatorname{Pdim}(\mathcal{F}) = \infty$.

For $\mathcal{Y} = L^p(\Omega)$, one can modify this argument as follows: Since Ω has nonempty interior, there are $x_0 \in \Omega$ and $r > 0$ such that $x_0 + r[0,1]^d \subset \Omega$. Let $m \in \mathbb{N}$ be arbitrary, and for $j \in \underline{m}$ define $M_j := x_0 + r\big[(\frac{j-1}{m}, \frac{j}{m}) \times [0,1]^{d-1}\big]$. Furthermore, for $b \in \{0,1\}^m$, let $g_b := \sum_{j \in \underline{m} \text{ with } b_j = 1} \mathbb{1}_{M_j}$, and note $g_b \in L^p(\Omega)$.

Since $L^p(\Omega) \cap \mathcal{F} \subset L^p(\Omega)$ is dense, there is for each $b \in \{0,1\}^m$ some $f_b \in \mathcal{F} \cap L^p(\Omega)$ such that $\|f_b - g_b\|_{L^p}^p \leq r^d/(2^{1+p} \cdot m \cdot 2^m)$. If we set $\Omega_b := \{x \in \Omega \colon |f_b(x) - g_b(x)| \geq 1/2\}$, then $\mathbb{1}_{\Omega_b} \leq 2^p \cdot |f_b - g_b|^p$, and hence

$$\lambda(\Omega_b) \leq 2^p \, \|f_b - g_b\|_{L^p}^p \leq \frac{r^d}{2 \cdot m \cdot 2^m},$$

and thus $\lambda(\bigcup_{b \in \{0,1\}^m} \Omega_b) \leq \frac{r^d}{2m}$, where λ is the Lebesgue measure. Hence,

$$\lambda(M_j \setminus \bigcup_{b \in \{0,1\}^m} \Omega_b) \geq \frac{r^d}{2m} > 0,$$

so that we can choose for each $j \in \underline{m}$ some $x_j \in M_j \setminus \bigcup_{b \in \{0,1\}^m} \Omega_b$. We then have

$$|f_b(x_j) - \delta_{b_j,1}| = |f_b(x_j) - g_b(x_j)| < 1/2,$$

and hence $f_b(x_j) > 1/2$ if $b_j = 1$ and $f_b(x_j) < 1/2$ otherwise. Thus, if we set $r_1 := \ldots := r_m := \frac{1}{2}$, then we have as above that $\mathbb{1}_{[0,\infty)}\big(f_b(x_j) - r_j\big) = b_j$ for all $j \in \underline{m}$ and $b \in \{0,1\}^m$. The remainder of the proof is as for $\mathcal{Y} = C(\Omega)$. □

Note that the following activation functions are computable by elementary operations: any piecewise polynomial function (in particular, the ReLU and the parametric ReLU), the exponential linear unit, the softsign (since the absolute value can be computed using a case distinction), the sigmoid, and the tanh. Thus, the preceding proposition applies to each of these activation functions.

Appendix D: Proofs of the Results in Sect. 3

D.1. Proof of Theorem 3.1

The proof of Theorem 3.1 is crucially based on the following lemma:

Lemma D.1 *Let μ be a finite Borel measure on $[-B, B]^d$ with uncountable support $\mathrm{supp}\mu$. For $x^*, v \in \mathbb{R}^d$ with $v \neq 0$, define*

$$H_\pm(x^*, v) := x^* + H_\pm(v)$$

where

$$H_+(v) := \{x \in \mathbb{R}^d : \langle x, v\rangle > 0\} \quad \text{and} \quad H_-(v) := \{x \in \mathbb{R}^d : \langle x, v\rangle < 0\}.$$

Then, there are $x^* \in [-B, B]^d$ and $v \in S^{d-1}$ such that if $f : [-B, B]^d \to \mathbb{R}$ satisfies

$$f(x) = c \ \text{ for } x \in H_+(x^*, v) \quad \text{and} \quad f(x) = c' \ \text{ for } x \in H_-(x^*, v) \quad \text{with } c \neq c', \tag{D.1}$$

then there is no continuous $g : [-B, B]^d \to \mathbb{R}$ satisfying $f = g$ μ-almost everywhere.

Proof *Step 1* Let $K := \mathrm{supp}\mu \subset [-B, B]^d$. In this step, we show that there is some $x^* \in K$ and some $v \in S^{d-1}$ such that $x^* \in \overline{K \cap H_+(x^*, v)} \cap \overline{K \cap H_-(x^*, v)}$. This follows from a result in [68], where the following is shown: For $x^* \in \mathbb{R}^d$ and $v \in S^{d-1}$, as well as $\delta, \eta > 0$, write

$$C(v; \delta, \eta) := \big\{r\,\xi : 0 < r < \delta \text{ and } \xi \in S^{d-1} \text{ with } |\xi - v| < \eta\big\},$$

and

$$C(x^*, v; \delta, \eta) := x^* + C(v; \delta, \eta).$$

Then, for each uncountable set $E \subset \mathbb{R}^d$ and for all but countably many $x^* \in E$, there is some $v \in S^{d-1}$ such that $E \cap C(x^*, v; \delta, \eta)$ and $E \cap C(x^*, -v; \delta, \eta)$ are both uncountable for all $\delta, \eta > 0$.

Now, if $\eta < 1$, then any $r\xi \in C(v; \delta, \eta)$ with $|\xi - v| < \eta$ satisfies $\langle v, \xi\rangle = \langle v, v\rangle + \langle v, \xi - v\rangle \geq 1 - |\xi - v| > 0$, so that $C(x^*, v; \delta, \eta) \subset B_\delta(x^*) \cap H_+(x^*, v)$ and $C(x^*, -v; \delta, \eta) \subset B_\delta(x^*) \cap H_-(x^*, v)$. From this it is easy to see that if x^*, v are as provided by the result in [68] (for $E = K$), then indeed $x^* \in \overline{K \cap H_+(x^*, v)} \cap \overline{K \cap H_-(x^*, v)}$.

We remark that strictly speaking, the proof in [68] is only provided for $E \subset \mathbb{R}^3$, but the proof extends almost verbatim to $\mathbb{R}^d$. A direct proof of the existence of x^*, v can be found in [58].

Step 2 We show that if x^*, v are as in Step 1 and if $f : [-B, B]^d \to \mathbb{R}$ satisfies (D.1), then there is no continuous $g : [-B, B]^d \to \mathbb{R}$ satisfying $f = g$ μ-almost everywhere.

Assume toward a contradiction that such a continuous function g exists. Recall (see for instance [19, Sect. 7.4]) that the support of μ is defined as

$$\mathrm{supp}\mu = [-B, B]^d \setminus \bigcup \{U : U \subset [-B, B]^d \text{ open and } \mu(U) = 0\}.$$

In particular, if $U \subset [-B, B]^d$ is open with $U \cap \mathrm{supp}\mu \neq \varnothing$, then $\mu(U) > 0$.

For each $n \in \mathbb{N}$, set $U_{n,+} := B_{1/n}(x^*) \cap [-B, B]^d \cap H_+(x^*, v)$ and $U_{n,-} := B_{1/n}(x^*) \cap [-B, B]^d \cap H_-(x^*, v)$, and note that $U_{n,\pm}$ are both open (as subsets of $[-B, B]^d$) with $K \cap U_{n,\pm} \neq \varnothing$, since $x^* \in \overline{K \cap H_+(x^*, v)}$ and $x^* \in \overline{K \cap H_-(x^*, v)}$. Hence, $\mu(U_{n,\pm}) > 0$. Since $f = g$ μ-almost everywhere, there exist $x_{n,\pm} \in U_{n,\pm}$ with $f(x_{n,\pm}) = g(x_{n,\pm})$. This implies $g(x_{n,+}) = c$ and $g(x_{n,-}) = c'$. But since $x_{n,\pm} \in B_{1/n}(x^*)$, we have $x_{n,\pm} \to x^*$, so that the continuity of g implies $g(x^*) = \lim_n g(x_{n,+}) = c$ and $g(x^*) = \lim_n g(x_{n,-}) = c'$, in contradiction to $c \neq c'$. □

We now prove Theorem 3.1. Set $\Omega := [-B, B]^d$ and define $\widetilde{N}_i := 1$ for $i = 1, \dots, L-2$ and $\widetilde{N}_{L-1} := 2$ if ϱ is unbounded, while $\widetilde{N}_{L-1} := 1$ otherwise. We show that for ϱ as given in the statement of the theorem there exists a sequence of functions in the set $\mathcal{RNN}_\varrho^\Omega((d, \widetilde{N}_1, \dots, \widetilde{N}_{L-1}, 1)) \subset \mathcal{RNN}_\varrho^\Omega(S)$ such that the sequence converges (in $L^p(\mu)$) to a *bounded, discontinuous limit* $f \in L^\infty(\mu)$, meaning that f does not have a continuous representative, even after possibly changing it on a μ-null-set. Since $\mathcal{RNN}_\varrho^\Omega(S) \subset C(\Omega)$, this will show that $f \in \overline{\mathcal{RNN}_\varrho^\Omega(S)} \setminus \mathcal{RNN}_\varrho^\Omega(S)$.

For the construction of the sequence, let $x^* \in \operatorname{supp}\mu$ and $v \in S^{d-1}$ as provided by Lemma D.1. Extend v to an orthonormal basis $(v, w_1, \dots, w_{d-1})$ of $\mathbb{R}^d$, and define $A := O^T$ for $O := (v, w_1, \dots, w_{d-1}) \in \mathbb{R}^{d \times d}$. Note that $x^* \in \Omega \subset \overline{B}_{dB}(0)$ and hence $A(\Omega - x^*) \subset \overline{B}_{2dB}(0) \subset [-2dB, 2dB]^d =: \Omega'$. Define $B' := 2dB$.

Next, using Proposition B.3, choose a neural network $\Psi \in \mathcal{NN}((d, 1, \dots, 1))$ with $L-1$ layers such that

(1) $\mathrm{R}_\varrho^{\Omega'}(\Psi)(0) = 0$;

(2) $\mathrm{R}_\varrho^{\Omega'}(\Psi)$ is differentiable at 0 and $\frac{\partial \mathrm{R}_\varrho^{\Omega'}(\Psi)}{\partial x_1}(0) = 1$;

(3) $\mathrm{R}_\varrho^{\Omega'}(\Psi)$ is constant in all but the x_1-direction; and

(4) $\mathrm{R}_\varrho^{\Omega'}(\Psi)$ is increasing with respect to each variable (with the remaining variables fixed).

Let $J_0 := \mathrm{R}_\varrho^{\Omega'}(\Psi)$. Since $J_0(0) = 0$ and $\frac{\partial J_0}{\partial x_1}(0) = 1$, we see directly from the definition of the partial derivative that for each $\delta \in (0, B')$, there are $x_\delta \in (-\delta, 0)$ and $y_\delta \in (0, \delta)$ such that $J_0(x_\delta, 0, \dots, 0) < J_0(0) = 0$ and $J_0(y_\delta, 0, \dots, 0) > J_0(0) = 0$. Furthermore, Properties (3) and (4) from above show that $J_0(x)$ only depends on x_1 and that $t \mapsto J_0(t, 0, \dots, 0)$ is increasing. In combination, these observations imply that

$$\begin{aligned} &J_0(x) < 0 \quad \text{for all } x \in \Omega' \text{ with } x_1 < 0, \\ &\text{and} \quad J_0(x) > 0 \quad \text{for all } x \in \Omega' \text{ with } x_1 > 0. \end{aligned} \tag{D.2}$$

Finally, with $\Psi = \big((A_1, b_1), \dots, (A_{L-1}, b_{L-1})\big)$, define

$$\Phi := \big((A_1 A, b_1 - A_1 A x^*), (A_2, b_2), \dots, (A_{L-1}, b_{L-1})\big),$$

and note that $\Phi \in \mathcal{NN}((d, 1, \dots, 1))$ with $L-1$ layers, and $\mathrm{R}_\varrho^{\mathbb{R}^d}(\Phi)(x) = \mathrm{R}_\varrho^{\mathbb{R}^d}(\Psi)(A(x - x^*))$ for all $x \in \mathbb{R}^d$. Combining this with the definition of A and

with Eq. (D.2), and noting that $A(x - x^*) \in \Omega'$ for $x \in \Omega$, we see that $J := \mathrm{R}_\varrho^\Omega(\Phi)$ satisfies

$$\begin{cases} J(x) < 0, & \text{for } x \in \Omega \cap H_-(x^*, v), \\ J(x) > 0, & \text{for } x \in \Omega \cap H_+(x^*, v), \\ J(x) = 0, & \text{for } x \in \Omega \cap H_0(x^*, v), \end{cases} \tag{D.3}$$

where $H_0(x^*, v) := \mathbb{R}^d \setminus (H_-(x^*, v) \cup H_+(x^*, v))$.

We now distinguish the cases given in Assumption (iv)(a) and (b) of Theorem 3.1.

Case 1 ϱ is unbounded, so that necessarily Assumption (iv)(a) of Theorem 3.1 holds, and $\widetilde{N}_{L-1} = 2$. For $n \in \mathbb{N}$ let $\Phi_n = \big((A_1^n, b_1^n), (A_2^n, b_2^n)\big) \in \mathcal{NN}((1, 2, 1))$ be given by

$$A_1^n = \begin{pmatrix} n \\ n \end{pmatrix} \in \mathbb{R}^{2\times 1}, \quad b_1^n = \begin{pmatrix} 0 \\ -1 \end{pmatrix} \in \mathbb{R}^2, \quad A_2^n = \begin{pmatrix} 1 & -1 \end{pmatrix} \in \mathbb{R}^{1\times 2}, \quad b_2^n = 0 \in \mathbb{R}^1.$$

Then, $\Phi_n \bullet \Phi \in \mathcal{NN}((d, \widetilde{N}_1, \dots, \widetilde{N}_{L-1}, 1))$. Now, let us define

$$h_n := \mathrm{R}_\varrho^\Omega(\Phi_n \bullet \Phi), \quad \text{and note} \quad h_n(x) = \varrho(nJ(x)) - \varrho(nJ(x) - 1) \quad \text{for } x \in \Omega.$$

Then, since h_n is continuous and hence bounded on the compact set Ω, we see that $h_n \in L^p(\mu)$ for every $n \in \mathbb{N}$ and all $p \in (0, \infty]$.

We now show that $(h_n)_{n\in\mathbb{N}}$ converges to a discontinuous limit. To see this, first consider $x \in \Omega \cap H_+(x^*, v)$. Since $J(x) > 0$ by (D.3), there exists some $N_x \in \mathbb{N}$ such that for all $n \geq N_x$, the estimate $nJ(x) - 1 > r$ holds, where $r > 0$ is as in Assumption (iii) of Theorem 3.1. Hence, by the mean value theorem, there exists some $\xi_n^x \in [nJ(x) - 1, nJ(x)]$ such that

$$\lim_{n\to\infty} h_n(x) = \lim_{n\to\infty} \varrho'(\xi_n^x) = \lambda,$$

since $\xi_n^x \to \infty$ as $n \to \infty$, $n \geq N_x$. Analogously, it follows for $x \in \Omega \cap H_-(x^*, v)$ that $\lim_{n\to\infty} h_n(x) = \lambda'$. Hence, setting $\gamma := \varrho(0) - \varrho(-1)$, we see for each $x \in \Omega$ that

$$\lim_{n\to\infty} h_n(x) = \left(\lambda \cdot \mathbb{1}_{H_+(x^*,v)} + \gamma \cdot \mathbb{1}_{H_0(x^*,v)} + \lambda' \cdot \mathbb{1}_{H_-(x^*,v)}\right)(x) =: h(x).$$

We now claim that there is some $M > 0$ such that $|\varrho(x) - \varrho(x - 1)| \leq M$ for all $x \in \mathbb{R}$. To see this, note because of $\varrho'(x) \to \lambda$ as $x \to \infty$ and because of $\varrho'(x) \to \lambda'$ as $x \to -\infty$ that there are $M_0 > 0$ and $R > r$ with $|\varrho'(x)| \leq M_0$ for all $x \in \mathbb{R}$ with $|x| \geq R$. Hence, ϱ is M_0-Lipschitz on $(-\infty, -R]$ and on $[R, \infty)$, so that $|\varrho(x) - \varrho(x - 1)| \leq M_0$ for all $x \in \mathbb{R}$ with $|x| \geq R + 1$. But by continuity and compactness, we also have $|\varrho(x) - \varrho(x - 1)| \leq M_1$ for all $|x| \leq R + 1$ and some constant $M_1 > 0$. Thus, we can simply choose $M := \max\{M_0, M_1\}$.

By what was shown in the preceding paragraph, we get $|h_n| \leq M$ and hence also $|h| \leq M$ for all $n \in \mathbb{N}$. Hence, by the dominated convergence theorem, we see for any $p \in (0, \infty)$ that $\lim_{n\to\infty} \|h_n - h\|_{L^p(\mu)} = 0$. But since $\lambda \neq \lambda'$, Lemma D.1 shows that h doesn't have a continuous representative, even after changing it on a μ-null-set.

This yields the required non-continuity of a limit point as discussed at the beginning of the proof.

Case 2

ϱ is bounded, so that $\widetilde{N}_{L-1} = 1$. Since ϱ is monotonically increasing, there exist $c, c' \in \mathbb{R}$ such that

$$\lim_{x\to\infty} \varrho(x) = c \quad \text{and} \quad \lim_{x\to-\infty} \varrho(x) = c'.$$

By the monotonicity and since ϱ is not constant (because of $\varrho'(x_0) \neq 0$), we have $c > c'$.

For each $n \in \mathbb{N}$, we now consider the neural network $\widetilde{\Phi}_n = \big((\widetilde{A}_1^n, \widetilde{b}_1^n), (\widetilde{A}_2^n, \widetilde{b}_2^n)\big) \in \mathcal{NN}((1,1,1))$ given by

$$\widetilde{A}_1^n = n \in \mathbb{R}^{1\times 1}, \quad \widetilde{b}_1^n = 0 \in \mathbb{R}^1, \quad \widetilde{A}_2^n = 1 \in \mathbb{R}^{1\times 1}, \quad \widetilde{b}_2^n = 0 \in \mathbb{R}^1.$$

Then, $\widetilde{\Phi}_n \bullet \Phi \in \mathcal{NN}((d, \widetilde{N}_1, \dots, \widetilde{N}_{L-1}, 1))$. Now, let us define

$$\widetilde{h}_n := \mathrm{R}_\varrho^\Omega(\widetilde{\Phi}_n \bullet \Phi) \quad \text{and note} \quad \widetilde{h}_n(x) = \varrho(nJ(x)) \quad \text{for all } x \in \Omega.$$

Since each of the $\widetilde{h}_n$ is continuous and Ω is compact, we have $\widetilde{h}_n \in L^p(\mu)$ for all $p \in (0,\infty]$. Equation (D.3) implies that $J(x) > 0$ for all $x \in \Omega \cap H_+(x^*, v)$. This in turn yields that

$$\lim_{n\to\infty} \widetilde{h}_n(x) = c \qquad \text{for all } x \in \Omega \cap H_+(x^*, v). \tag{D.4}$$

Similarly, the fact that $J(x) < 0$ for all $x \in \Omega \cap H_-(x^*, v)$ yields

$$\lim_{n\to\infty} \widetilde{h}_n(x) = c' \qquad \text{for all } x \in \Omega \cap H_-(x^*, v). \tag{D.5}$$

Combining (D.4) with (D.5) yields for all $x \in \Omega$ that

$$\lim_{n\to\infty} \widetilde{h}_n(x) = \big(c \cdot \mathbb{1}_{H_+(x_*,v)} + \varrho(0) \cdot \mathbb{1}_{H_0(x^*,v)} + c' \cdot \mathbb{1}_{H_-(x^*,v)}\big)(x) =: \widetilde{h}(x).$$

By the boundedness of ϱ, we get $|\widetilde{h}_n(x)| \leq C$ for all $n \in \mathbb{N}$ and $x \in \Omega$ and a suitable $C > 0$, so that also $\widetilde{h}$ is bounded. Together with the dominated convergence theorem, this implies for any $p \in (0,\infty)$ that $\lim_{n\to\infty} \big\|\widetilde{h}_n - \widetilde{h}\big\|_{L^p(\mu)} = 0$. Since $c \neq c'$, Lemma D.1 shows that $\widetilde{h}$ does not have a continuous representative (with respect to equality μ-almost everywhere). This yields the required non-continuity of a limit point as discussed at the beginning of the proof. □

D.2. Proof of Corollary 3.2

It is not hard to verify that all functions listed in Table 1 are continuous and increasing. Furthermore, each activation function ϱ listed in Table 1 is not constant and satisfies

$\varrho|_{\mathbb{R}\setminus\{0\}} \in C^\infty(\mathbb{R}\setminus\{0\})$. This shows that $\varrho|_{(-\infty,-r)\cup(r,\infty)}$ is differentiable for any $r > 0$, and that there is some $x_0 = x_0(\varrho) \in \mathbb{R}$ such that $\varrho'(x_0) \neq 0$.

Next, the softsign, the inverse square root unit, the sigmoid, the tanh, and the arctan function are all bounded, and thus satisfy condition (iv)(b) of Theorem 3.1. Thus, all that remains is to verify condition (iv)(a) of Theorem 3.1 for the remaining activation functions:

1. For the ReLU $\varrho(x) = \max\{0, x\}$, condition (iv)(a) is satisfied with $\lambda = 1$ and $\lambda' = 0 \neq \lambda$.
2. For the parametric ReLU $\varrho(x) = \max\{ax, x\}$ (with $a \geq 0$, $a \neq 1$), Condition (iv)(a) is satisfied with $\lambda = \max\{1, a\}$ and $\lambda' = \min\{1, a\}$, where $\lambda \neq \lambda'$ since $a \neq 1$.
3. For the exponential linear unit $\varrho(x) = x\mathbb{1}_{[0,\infty)}(x) + (e^x - 1)\mathbb{1}_{(-\infty,0)}(x)$, Condition (iv)(a) is satisfied for $\lambda = 1$ and $\lambda' = \lim_{x\to-\infty} e^x = 0 \neq \lambda$.
4. For the inverse square root linear unit $\varrho(x) = x\mathbb{1}_{[0,\infty)}(x) + \frac{x}{\sqrt{1+ax^2}}\mathbb{1}_{(-\infty,0)}(x)$, the quotient rule shows that for $x < 0$ we have

$$\begin{aligned}\varrho'(x) &= \frac{\sqrt{1+ax^2} - x\cdot\frac{1}{2}(1+ax^2)^{-1/2}2ax}{1+ax^2} \\ &= \frac{(1+ax^2) - ax^2}{(1+ax^2)^{3/2}} = (1+ax^2)^{-3/2}. \end{aligned} \tag{D.6}$$

   Therefore, Condition (iv)(a) is satisfied for $\lambda = 1$ and $\lambda' = \lim_{x\to-\infty} \varrho'(x) = 0 \neq \lambda$.
5. For the softplus function $\varrho(x) = \ln(1 + e^x)$, Condition (iv)(a) is satisfied for

$$\lambda = \lim_{x\to\infty} \frac{e^x}{1+e^x} = 1 \quad \text{and} \quad \lambda' = \lim_{x\to-\infty} \frac{e^x}{1+e^x} = 0 \neq \lambda.$$

□

D.3. Proof of Theorem 3.3

D.3.1. Proof of Theorem 3.3 Under Condition (i)

Let $\Omega := [-B, B]^d$. Let $m \in \mathbb{N}$ be maximal with $\varrho \in C^m(\mathbb{R})$; this is possible since $\varrho \in C^1(\mathbb{R})\setminus C^\infty(\mathbb{R})$. Note that $\varrho \in C^m(\mathbb{R})\setminus C^{m+1}(\mathbb{R})$. This easily implies $\mathcal{RNN}_\varrho^\Omega(S) \subset C^m(\Omega)$.

We now show for the architecture $S' := (d, \widetilde{N}_1, \ldots, \widetilde{N}_{L-2}, 2, 1)$, where $\widetilde{N}_i := 1$ for all $i = 1, \ldots, L-2$, that the set $\mathcal{RNN}_\varrho^\Omega(S')$ is not closed in $C(\Omega)$. If we had $\varrho \in C^{m+1}([-C, C])$ for all $C > 0$, this would imply $\varrho \in C^{m+1}(\mathbb{R})$; thus, there is $C > 0$ such that $\varrho \notin C^{m+1}([-C, C])$. Now, choose $\lambda > C/B$, so that $\lambda[-B, B] \supset [-C, C]$. This entails that $\varrho(\lambda\cdot) \in C^m([-B, B])\setminus C^{m+1}([-B, B])$. Next, since the continuous derivative $\frac{\mathrm{d}}{\mathrm{d}x}\varrho(\lambda x) = \lambda\varrho'(\lambda x)$ is bounded on the compact set $[-B, B]$, we see that $\varrho(\lambda\cdot)$ is Lipschitz continuous on $[-B, B]$, and we set $M_1 := \mathrm{Lip}(\varrho(\lambda\cdot))$. Next, by the

uniform continuity of $\lambda \cdot \varrho'(\lambda \cdot)$ on $[-(B+1), B+1]$, if we set

$$\varepsilon_n := \sup_{\substack{x,y \in [-(B+1),B+1] \\ \text{with } |x-y| \leq 1/n}} |\lambda \cdot \varrho'(\lambda x) - \lambda \cdot \varrho'(\lambda y)|,$$

then $\varepsilon_n \to 0$ as $n \to \infty$.

For $n \in \mathbb{N}$, let $\Phi_n^1 = \big((A_1^n, b_1^n), (A_2^n, b_2^n)\big) \in \mathcal{NN}((1,2,1))$ be given by

$$A_1^n = \begin{pmatrix} \lambda \\ \lambda \end{pmatrix} \in \mathbb{R}^{2\times 1}, \quad b_1^n = \begin{pmatrix} \lambda/n \\ 0 \end{pmatrix} \in \mathbb{R}^2, \quad A_2^n = \begin{pmatrix} n & -n \end{pmatrix} \in \mathbb{R}^{1\times 2}, \quad b_2^n = 0 \in \mathbb{R}^1.$$

Note that there is some $x^* \in \mathbb{R}$ such that $\varrho'(x^*) \neq 0$, since otherwise $\varrho' \equiv 0$ and hence $\varrho \in C^\infty(\mathbb{R})$. Thus, for each $n \in \mathbb{N}$, Proposition B.3 yields the existence of a neural network $\Phi_n^2 \in \mathcal{NN}((d, 1, \dots, 1))$ with $L-1$ layers such that

$$\left| \mathrm{R}_\varrho^\Omega(\Phi_n^2)(x) - x_1 \right| \leq \frac{1}{2n^2} \text{ for all } x \in \Omega. \tag{D.7}$$

We set $\Phi_n := \Phi_n^1 \bullet \Phi_n^2 \in \mathcal{NN}(S')$ and $f_n := \mathrm{R}_\varrho^\Omega(\Phi_n)$. For $x \in \Omega$, we then have

$$\begin{aligned} |f_n(x) - \lambda \varrho'(\lambda x_1)| = \Big| n \cdot \Big(\varrho\Big(\lambda \mathrm{R}_\varrho^\Omega(\Phi_n^2)(x) + \lambda \cdot n^{-1} \Big) \\ - \varrho(\lambda \mathrm{R}_\varrho^\Omega(\Phi_n^2)(x)) \Big) - \lambda \varrho'(\lambda x_1) \Big|. \end{aligned}$$

Now, by the Lipschitz continuity of $\varrho(\lambda \cdot)$ and Eq. (D.7), we conclude that

$$\begin{aligned} &\Big| n \cdot \Big(\varrho\Big(\lambda \mathrm{R}_\varrho^\Omega(\Phi_n^2)(x) + \lambda \cdot n^{-1} \Big) \\ &\quad - \varrho(\lambda \mathrm{R}_\varrho^\Omega(\Phi_n^2)(x)) \Big) - n \cdot \Big(\varrho\Big(\lambda \Big(x_1 + n^{-1} \Big) \Big) - \varrho(\lambda x_1) \Big) \Big| \\ &\leq \frac{M_1 \lambda}{n}. \end{aligned}$$

This implies for every $x \in \Omega$ that

$$\begin{aligned} |f_n(x) - \lambda \varrho'(\lambda x_1)| &\leq \Big| n \Big(\varrho\Big(\lambda \Big(x_1 + n^{-1} \Big) \Big) - \varrho(\lambda x_1) \Big) - \lambda \varrho'(\lambda x_1) \Big| + \frac{M_1 \lambda}{n} \\ &\overset{\text{(mean value theorem, } \xi_n^x \in (x_1, x_1 + n^{-1}))}{=} \big| \lambda \cdot \varrho'(\lambda \cdot \xi_n^x) - \lambda \varrho'(\lambda x_1) \big| \\ &\quad + \frac{M_1 \lambda}{n} \\ &\leq \varepsilon_n + \frac{M_1 \lambda}{n}. \end{aligned}$$

Here, the last step used that $|\xi_n^x - x_1| \leq n^{-1} \leq 1$, so that $x_1, \xi_n^x \in [-(B+1), B+1]$.

Overall, we established the existence of a sequence $(f_n)_{n\in\mathbb{N}}$ in $\mathcal{RNN}_\varrho^\Omega(S')$ which converges uniformly to the function $\Omega \to \mathbb{R}$, $x \mapsto \varrho_\lambda(x) := \lambda \, \varrho'(\lambda x_1)$. By our choice

of λ, we have $\varrho_\lambda \notin C^m(\Omega)$. Because of $\mathcal{RNN}_\varrho^\Omega(S') \subset C^m(\Omega)$, we thus see that $\varrho_\lambda \notin \mathcal{RNN}_\varrho^\Omega(S')$, so that $\mathcal{RNN}_\varrho^\Omega(S')$ is not closed in $C(\Omega)$.

Finally, note by Lemma B.1 that

$$f_n \in \mathcal{RNN}_\varrho^\Omega(S') \subset \mathcal{RNN}_\varrho^\Omega(S) \quad \text{for all} \quad n \in \mathbb{N}.$$

Since $f_n \to \varrho_\lambda$ uniformly, where $\varrho_\lambda \notin C^m(\Omega)$, and hence $\varrho_\lambda \notin \mathcal{RNN}_\varrho^\Omega(S)$, we thus see that $\mathcal{RNN}_\varrho^\Omega(S)$ is not closed in $C(\Omega)$. □

D.3.2. Proof of Theorem 3.3 Under Condition (ii)

Let $\Omega := [-B, B]^d$. We first show that if we set $S' := (d, \widetilde{N}_1, \dots, \widetilde{N}_{L-2}, 2, 1)$, where $\widetilde{N}_i := 1$ for all $i = 1, \dots, L-2$, then there exists a limit point of $\mathcal{RNN}_\varrho^\Omega(S')$ which is the restriction $f|_\Omega$ of an *unbounded* analytic function $f : \mathbb{R} \to \mathbb{R}$.

Since ϱ is not constant, there is some $x^* \in \mathbb{R}$ such that $\varrho'(x^*) \neq 0$. For $n \in \mathbb{N}$, let us define $\Phi_n^1 := \big((A_1^n, b_1^n), (A_2^n, b_2^n)\big) \in \mathcal{NN}((1, 2, 1))$ by

$$\begin{aligned} A_1^n &:= \begin{pmatrix} 1 \\ 1/n \end{pmatrix} \in \mathbb{R}^{2\times 1}, \quad b_1^n := \begin{pmatrix} 0 \\ x^* \end{pmatrix} \in \mathbb{R}^2, \quad A_2^n \\ &:= \begin{pmatrix} 1 & n \end{pmatrix} \in \mathbb{R}^{1\times 2}, \quad b_2^n := -\varrho(x^*)n \in \mathbb{R}^1. \end{aligned}$$

With this choice, we have

$$\mathrm{R}_\varrho^{\mathbb{R}}(\Phi_n^1)(x) = \varrho(x) + n \cdot \big(\varrho(x/n + x^*) - \varrho(x^*)\big) \quad \text{for all } x \in \mathbb{R}.$$

For any $x \in \mathbb{R}$, the mean-value theorem yields $\widetilde{x}$ between x^* and $x^* + \frac{x}{n}$ satisfying $\varrho(x^* + \frac{x}{n}) - \varrho(x^*) = \frac{x}{n} \cdot \varrho'(\widetilde{x})$. Therefore, if $B > 0$ and $x \in [-B, B]$, then

$$\begin{aligned} &\Big|\mathrm{R}_\varrho^{\mathbb{R}}(\Phi_n^1)(x) - \big(\varrho(x) + \varrho'(x^*)x\big)\Big| \le B|\varrho'(\widetilde{x}) \\ &-\varrho'(x^*)| \quad \text{for some} \quad \widetilde{x} \in [x^* - B/n, x^* + B/n]. \end{aligned}$$

Since ϱ' is continuous, we conclude that

$$\sup_{x\in[-B,B]} \Big|\mathrm{R}_\varrho^{\mathbb{R}}(\Phi_n^1)(x) - \big(\varrho(x) + \varrho'(x^*)x\big)\Big| \xrightarrow[n\to\infty]{} 0. \tag{D.8}$$

Moreover, note that $\frac{\mathrm{d}}{\mathrm{d}x}\mathrm{R}_\varrho^{\mathbb{R}}(\Phi_n^1)(x) = \varrho'(x) + \varrho'(x^* + n^{-1} \cdot x)$ is bounded on $[-(B+1), B+1]$, uniformly with respect to $n \in \mathbb{N}$. Hence, $\mathrm{R}_\varrho^{\mathbb{R}}(\Phi_n^1)$ is Lipschitz continuous on $[-(B+1), B+1]$, with Lipschitz constant $C' > 0$ independent of $n \in \mathbb{N}$.

Next, for each $n \in \mathbb{N}$, Proposition B.3 yields a neural network $\Phi_n^2 \in \mathcal{NN}((d, 1, \dots, 1))$ with $L-1$ layers such that

$$\Big|\mathrm{R}_\varrho^\Omega(\Phi_n^2)(x) - x_1\Big| \le \frac{1}{n}, \text{ for all } x \in \Omega. \tag{D.9}$$

We set $\Phi_n := \Phi_n^1 \bullet \Phi_n^2 \in \mathcal{NN}(S')$ and note for all $x \in \Omega$ that

$$\left|\mathrm{R}_\varrho^\Omega(\Phi_n)(x) - \big(\varrho(x_1) + \varrho'(x^*)x_1\big)\right| = \left|\mathrm{R}_\varrho^{\mathbb{R}}(\Phi_n^1)\big(\mathrm{R}_\varrho^\Omega(\Phi_n^2)(x)\big) - \big(\varrho(x_1) + \varrho'(x^*)x_1\big)\right|.$$

By the Lipschitz continuity of $\mathrm{R}_\varrho^{\mathbb{R}}(\Phi_n^1)$ on $[-(B+1), B+1]$, and using (D.9), we conclude that

$$\left|\mathrm{R}_\varrho^\Omega(\Phi_n)(x) - \big(\varrho(x_1) + \varrho'(x^*)x_1\big)\right| \leq \left|\mathrm{R}_\varrho^{\mathbb{R}}(\Phi_n^1)(x_1) - \big(\varrho(x_1) + \varrho'(x^*)x_1\big)\right| + \frac{C'}{n},$$

so that an application of (D.8) yields

$$\sup_{x\in\Omega}\left|\mathrm{R}_\varrho^\Omega(\Phi_n)(x) - \big(\varrho(x_1) + \varrho'(x^*)x_1\big)\right| \xrightarrow[n\to\infty]{} 0.$$

Now, to show that $\mathcal{RNN}_\varrho^\Omega(S) \subset C(\Omega)$ is not closed, due to the fact that there holds $\mathrm{R}_\varrho^\Omega(\Phi_n) \in \mathcal{RNN}_\varrho^\Omega(S') \subset \mathcal{RNN}_\varrho^\Omega(S)$, it is sufficient to show that with

$$F : \mathbb{R}^d \to \mathbb{R}, \quad x \mapsto \varrho(x_1) + \varrho'(x^*)x_1,$$

$F|_\Omega$ is not an element of $\mathcal{RNN}_\varrho^\Omega(S)$. This is accomplished, once we show that there do not exist any $\widehat{N}_1, \dots, \widehat{N}_{L-1} \in \mathbb{N}$ such that $F|_\Omega$ is an element of $\mathcal{RNN}_\varrho^\Omega((d, \widehat{N}_1, \dots, \widehat{N}_{L-1}, 1))$.

Toward a contradiction, we assume that there exist $\widehat{N}_1, \dots, \widehat{N}_{L-1} \in \mathbb{N}$ such that $F|_\Omega = \mathrm{R}_\varrho^\Omega(\Phi^3)$ for a network $\Phi^3 \in \mathcal{NN}((d, \widehat{N}_1, \dots, \widehat{N}_{L-1}, 1))$. Since F and $\mathrm{R}_\varrho^{\mathbb{R}^d}(\Phi^3)$ are both analytic functions that coincide on $\Omega = [-B, B]^d$, they must be equal on all of $\mathbb{R}^d$. However, F is unbounded (since ϱ is bounded, and since $\varrho'(x^*) \neq 0$), while $\mathrm{R}_\varrho^{\mathbb{R}^d}(\Phi^3)$ is bounded as a consequence of ϱ being bounded. This produces the desired contradiction. □

D.3.3. Proof of Theorem 3.3 Under Condition (iii)

Let $\varrho \in C^{\max\{r,q\}}(\mathbb{R})$ be approximately homogeneous of order (r, q) with $r \neq q$. For simplicity, let us assume that $r > q$; we will briefly comment on the case $q > r$ at the end of the proof.

Note that $r \geq 1$, since $r, q \in \mathbb{N}_0$ with $r > q$. Let $(x)_+ := \max\{x, 0\}$ for $x \in \mathbb{R}$. We start by showing that

$$k^{-r}\varrho(k\cdot) \xrightarrow[k\to\infty]{\text{uniformly on } [-B,B]} (\cdot)_+^r. \tag{D.10}$$

To see this, let $s > 0$ such that $|\varrho(x) - x^r| \leq s$ for all $x > 0$ and $|\varrho(x) - x^q| \leq s$ for all $x < 0$. For any $k \in \mathbb{N}$ and $x \in [-B, 0]$, we have

$$\begin{aligned} |k^{-r}\,\varrho(kx) - (x)_+^r| = |k^{-r}\,\varrho(kx)| &\leq k^{-r} \cdot \big(|\varrho(kx) - (kx)^q| + |(kx)^q|\big) \\ &\leq k^{-r} \cdot (s + k^q B^q) \leq c_0 \cdot k^{-1} \end{aligned}$$

for a constant $c_0 = c_0(B, s, r, q) > 0$. Moreover, for $x \in [0, B]$, we have

$$|k^{-r}\,\varrho(kx) - (x)_+^r| = k^{-r}|\varrho(kx) - (kx)^r| \leq s \cdot k^{-r}.$$

Overall, we conclude that

$$\sup_{x \in [-B,B]} |k^{-r}\varrho(kx) - (x)_+^r| \leq \max\{c_0, s\} \cdot k^{-1},$$

which implies (D.10).

We observe that $\big(x \mapsto (x)_+^r\big) \notin C^r([-B, B])$. Additionally, since $\varrho \in C^{\max\{r,q\}}(\mathbb{R}) = C^r(\mathbb{R})$, we have $\mathcal{RNN}_\varrho^{[-B,B]^d}(S) \subset C^r([-B, B]^d)$. Hence, the proof is complete if we can construct a sequence $(\Phi_n)_{n\in\mathbb{N}}$ of neural networks in $\mathcal{NN}((d, 1, \ldots, 1))$ (with L layers) such that the ϱ-realizations $\mathrm{R}_\varrho^\Omega(\Phi_n)$ converge uniformly to the function $[-B, B]^d \to \mathbb{R}, x \mapsto (x_1)_+^r$. By the preceding considerations, this is clearly possible, as can be seen by the same arguments used in the proofs of the previous results. For invoking these arguments, note that $\max\{r, q\} \geq 1$, so that $\varrho \in C^1(\mathbb{R})$. Also, since ϱ is approximately homogeneous of order (r, q) with $r \neq q$, ϱ cannot be constant, and hence $\varrho'(x_0) \neq 0$ for some $x_0 \in \mathbb{R}$.

For completeness, let us briefly consider the case where $q > r$ that was omitted at the beginning of the proof. In this case, $(-k)^{-q}\,\varrho(-k\cdot) \to (\cdot)_+^q$ with uniform convergence on $[-B, B]$. Indeed, for $x \in [0, B]$, we have $|(-k)^{-q}\,\varrho(-kx) - (x)_+^q| = k^{-q}|\varrho(-kx) - (-kx)^q| \leq k^{-q} \cdot s \leq s \cdot k^{-1}$. Similarly, for $x \in [-B, 0]$, we get $|(-k)^{-q}\,\varrho(-kx) - (x)_+^q| \leq k^{-q}\big(|\varrho(-kx) - (-kx)^r| + |(-kx)^r|\big) \leq k^{-q}(s + B^r k^r) \leq c_1 \cdot k^{-1}$ for some constant $c_1 = c_1(B, s, r, q) > 0$. Here, we used that $q - r \geq 1$, since $r, q \in \mathbb{N}_0$ with $q > r$. Now, the proof proceeds as before, noting that $\big(x \mapsto (x)_+^q\big) \notin C^q([-B, B])$, while $\varrho \in C^{\max\{r,q\}}(\mathbb{R}) \subset C^q(\mathbb{R})$. □

D.4. Proof of Corollary 3.4

D.4.1. Proof of Corollary 3.4.(1)

Powers of ReLUs: For $k \in \mathbb{N}$, let $\mathrm{ReLU}_k : \mathbb{R} \to \mathbb{R}, x \mapsto \max\{0, x\}^k$, and note that this is a continuous function. On $\mathbb{R} \setminus \{0\}$, ReLU_k is differentiable with $\mathrm{ReLU}_k' = k \cdot \mathrm{ReLU}_{k-1}$. Furthermore, if $k \geq 2$, then $|h^{-1}(\mathrm{ReLU}_k(h) - \mathrm{ReLU}_k(0))| \leq |h|^{k-1} \to 0$ as $h \to 0$. Thus, if $k \geq 2$, then ReLU_k is continuously differentiable with derivative $\mathrm{ReLU}_k' = k \cdot \mathrm{ReLU}_{k-1}$. Finally, ReLU_1 is *not* differentiable at $x = 0$. Overall, this shows $\mathrm{ReLU}_k \in C^1(\mathbb{R}) \setminus C^\infty(\mathbb{R})$ for all $k \geq 2$.

The exponential linear unit: We have $\frac{d^k}{dx^k}(e^x - 1) = e^x$ for all $k \in \mathbb{N}$. Therefore, the exponential linear unit $\varrho : \mathbb{R} \to \mathbb{R}, x \mapsto x\mathbb{1}_{[0,\infty)}(x) + (e^x - 1)\mathbb{1}_{(-\infty,0)}(x)$ satisfies for $k \in \mathbb{N}_0$ that

$$\lim_{x \downarrow 0} \varrho^{(k)}(x) = \delta_{k,1} \qquad \text{and} \qquad \lim_{x \uparrow 0} \varrho^{(k)}(x) = \begin{cases} \lim_{x \uparrow 0}(e^x - 1) = 0, & \text{if } k = 0, \\ \lim_{x \uparrow 0} e^x = 1, & \text{if } k \neq 0. \end{cases}$$

By standard results in real analysis (see for instance [23, Problem 2 in Chapter VIII.6]), this implies that $\varrho \in C^1(\mathbb{R}) \setminus C^2(\mathbb{R})$.

The softsign function: On $(-1, \infty)$, we have $\frac{\mathrm{d}}{\mathrm{d}x}\frac{x}{1+x} = (1 + x)^{-2}$ and $\frac{d^2}{dx^2}\frac{x}{1+x} = -2(1 + x)^{-3}$. Furthermore, if $x < 0$, then $\operatorname{softsign}(x) = \frac{x}{1+|x|} = -\frac{-x}{1+(-x)} = -\operatorname{softsign}(-x)$. Therefore, the softsign function is C^∞ on $\mathbb{R} \setminus \{0\}$, and satisfies

$$\lim_{x \downarrow 0} \operatorname{softsign}'(x) = \lim_{x \downarrow 0}(1 + x)^{-2} = 1 = \lim_{x \uparrow 0}(1 - x)^{-2} = \lim_{x \uparrow 0} \operatorname{softsign}'(x).$$

By standard results in real analysis (see for instance [23, Problem 2 in Chapter VIII.6]), this implies that $\operatorname{softsign} \in C^1(\mathbb{R})$. However, since

$$\lim_{x \downarrow 0} \operatorname{softsign}''(x) = \lim_{x \downarrow 0} -2(1 + x)^{-3} = -2 \quad \text{and} \quad \lim_{x \uparrow 0} \operatorname{softsign}''(x) = \lim_{x \uparrow 0} 2(1 - x)^{-3} = 2,$$

we have $\operatorname{softsign} \notin C^2(\mathbb{R})$.

The inverse square root linear unit: Let

$$\varrho : \mathbb{R} \to \mathbb{R}, x \mapsto x\mathbb{1}_{[0,\infty)}(x) + \frac{x}{(1 + ax^2)^{1/2}}\mathbb{1}_{(-\infty,0)}(x)$$

denote the inverse square root linear unit with parameter $a > 0$, and note $\varrho|_{\mathbb{R}\setminus\{0\}} \in C^\infty(\mathbb{R} \setminus \{0\})$. As we saw in Eq. (D.6), we have $\frac{\mathrm{d}}{\mathrm{d}x}\frac{x}{(1+ax^2)^{1/2}} = (1 + ax^2)^{-3/2}$, and thus $\frac{d^2}{dx^2}\frac{x}{(1+ax^2)^{1/2}} = -3ax \cdot (1 + ax^2)^{-5/2}$, and finally $\frac{d^3}{dx^3}\frac{x}{(1+ax^2)^{1/2}} = -3a(1 + ax^2)^{-5/2} + 15a^2x^2(1 + ax^2)^{-7/2}$. These calculations imply

$$\lim_{x \uparrow 0} \varrho'(x) = \lim_{x \uparrow 0}(1 + ax^2)^{-3/2} = 1 = \lim_{x \downarrow 0} \varrho'(x)$$

and

$$\lim_{x \uparrow 0} \varrho''(x) = \lim_{x \uparrow 0} -3ax \cdot (1 + ax^2)^{-5/2} = 0 = \lim_{x \downarrow 0} \varrho''(x),$$

but also

$$\begin{aligned}\lim_{x\uparrow 0}\varrho'''(x) &= \lim_{x\uparrow 0}\big[-3a(1+ax^2)^{-5/2}+15a^2x^2(1+ax^2)^{-7/2}\big]\\ &= -3a \neq 0 = \lim_{x\downarrow 0}\varrho'''(x).\end{aligned}$$

By standard results in real analysis (see for instance [23, Problem 2 in Chapter VIII.6]), this implies that $\varrho \in C^2(\mathbb{R}) \setminus C^3(\mathbb{R})$. □

D.4.2. Proof of Corollary 3.4.(3)

The softplus function Clearly, softplus $\in C^\infty(\mathbb{R}) \subset C^{\max\{1,0\}}(\mathbb{R})$. Furthermore, the softplus function is approximately homogeneous of order $(1, 0)$. Indeed, for $x \geq 0$, we have

$$|\ln(1+e^x) - x| = \left|\ln\left(\frac{1+e^x}{e^x}\right)\right| = \ln(1+e^{-x}) \leq \ln(2),$$

and for $x \leq 0$, we have $|\ln(1+e^x) - x^0| \leq 1 + \ln(2)$. □

D.5. Proof of Proposition 3.5

The set Θ_C is closed and bounded in the normed space $\big(\mathcal{NN}(S), \|\cdot\|_{\mathcal{NN}(S)}\big)$. Thus, the Heine-Borel Theorem implies the compactness of Θ_C. By Proposition 4.1 (which will be proved independently), the map

$$\mathrm{R}_\varrho^\Omega : \big(\mathcal{NN}(S), \|\cdot\|_{\mathcal{NN}(S)}\big) \to \big(C(\Omega), \|\cdot\|_{\sup}\big)$$

is continuous. As a consequence, the set $\mathrm{R}_\varrho^\Omega(\Theta_C)$ is compact in $C(\Omega)$. Because of the compactness of Ω, $C(\Omega)$ is continuously embedded into $L^p(\mu)$ for every $p \in (0, \infty)$ and any finite Borel measure μ on Ω. This implies that the set $\mathrm{R}_\varrho^\Omega(\Theta_C)$ is compact in $L^p(\mu)$ as well. □

D.6. Proof of Proposition 3.6

With $(\Phi_N)_{N\in\mathbb{N}}$ as in the statement of Proposition 3.6, we want to show that $\|\Phi_N\|_{\text{total}} \to \infty$ in probability. By definition, this means that for each fixed $C > 0$, and letting Ω_N denote the event where $\|\Phi_N\| \geq C$, we want to show that $\mathbb{P}(\Omega_N) \to 1$ as $N \to \infty$. For brevity, let us write $\mathcal{R}^Z := \mathrm{R}_\varrho^\Omega(\Theta_Z)$ for Θ_Z, $Z > 0$ as in Proposition 3.5.

By compactness of $\mathcal{R}^C$, we can choose $g \in \mathcal{R}^C$ satisfying

$$\|f_\sigma - g\|^2_{L^2(\sigma_\Omega)} = \inf_{h\in\mathcal{R}^C} \|f_\sigma - h\|^2_{L^2(\sigma_\Omega)}.$$

Define $M := \inf_{h\in\mathcal{RNN}_\varrho^\Omega(S)} \|f_\sigma - h\|^2_{L^2(\sigma_\Omega)}$. Since by assumption the infimum defining M is not attained, we have $\|f_\sigma - g\|^2_{L^2(\sigma_\Omega)} > M$, so that there are

$h \in \mathcal{RNN}_{\varrho}^{\Omega}(S)$ and $\delta > 0$ with $\|f_\sigma - g\|_{L^2(\sigma_\Omega)}^2 \geq 2\delta + \|f_\sigma - h\|_{L^2(\sigma_\Omega)}^2$. Let $C' > 0$ with $h \in \mathcal{R}^{C'}$. For $N \in \mathbb{N}$ and $\varepsilon > 0$, let us denote by $\Omega_{N,\varepsilon}^{(1)}$ the event where $\sup_{f \in \mathcal{R}^{C'}} |\mathcal{E}_\sigma(f) - E_N(f)| > \varepsilon$. Since $\mathcal{R}^{C'}$ is compact, [20]Theorem B shows for arbitrary $\varepsilon > 0$ that $\mathbb{P}(\Omega_{N,\varepsilon}^{(1)}) \xrightarrow[N\to\infty]{} 0$ for each fixed $\varepsilon > 0$. Similarly, denoting by $\Omega_{N,\varepsilon}^{(2)}$ the event where $E_N(\mathrm{R}_{\varrho}^{\Omega}(\Phi_N)) - \inf_{f \in \mathcal{RNN}_{\varrho}^{\Omega}(S)} E_N(f) > \varepsilon$, we have by assumption (3.3) that $\mathbb{P}(\Omega_{N,\varepsilon}^{(2)}) \xrightarrow[N\to\infty]{} 0$, for each fixed $\varepsilon > 0$.

We now claim that $\Omega_N^c \subset \Omega_{N,\delta/3}^{(1)} \cup \Omega_{N,\delta/3}^{(2)}$. Once we prove this, we get

$$0 \leq \mathbb{P}(\Omega_N^c) \leq \mathbb{P}\big(\Omega_{N,\delta/3}^{(1)}\big) + \mathbb{P}\big(\Omega_{N,\delta/3}^{(2)}\big) \xrightarrow[N\to\infty]{} 0,$$

and hence $\mathbb{P}(\Omega_N) \to 1$, as desired.

To prove $\Omega_N^c \subset \Omega_{N,\delta/3}^{(1)} \cup \Omega_{N,\delta/3}^{(2)}$, assume toward a contradiction that there exists a training sample $\omega := \big((x_i, y_i)\big)_{i\in\mathbb{N}} \in \Omega_N^c \setminus \big(\Omega_{N,\delta/3}^{(1)} \cup \Omega_{N,\delta/3}^{(2)}\big)$. Thus, $\|\Phi_N\|_{\text{total}} < C$, meaning $f_N := \mathrm{R}_{\varrho}^{\Omega}(\Phi_N) \in \mathcal{R}^C \subset \mathcal{R}^{C'}$. Using the decomposition of the expected loss from Eq. (3.2), we thus see

$$\begin{aligned}
&\|g - f_\sigma\|_{L^2(\sigma_\Omega)}^2 + \mathcal{E}_\sigma(f_\sigma) \\
(\text{by choice of } g \text{ and since } f_N \in \mathcal{R}^C) \quad &\leq \|f_N - f_\sigma\|_{L^2(\sigma_\Omega)}^2 + \mathcal{E}_\sigma(f_\sigma) = \mathcal{E}_\sigma\big(f_N\big) \\
(\text{since } \omega \notin \Omega_{N,\delta/3}^{(1)} \text{ and } f_N \in \mathcal{R}^{C'}) \quad &\leq E_N(f_N) + \frac{\delta}{3} \\
(\text{since } \omega \notin \Omega_{N,\delta/3}^{(2)}) \quad &\leq \frac{2}{3}\delta + \inf_{f \in \mathcal{RNN}_{\varrho}^{\Omega}(S)} E_N(f) \leq \frac{2}{3}\delta + E_N(h) \\
(\text{since } h \in \mathcal{R}^{C'} \text{ and } \omega \notin \Omega_{N,\delta/3}^{(1)}) \quad &\leq \mathcal{E}_\sigma(h) + \delta = \mathcal{E}_\sigma(f_\sigma) + \|h - f_\sigma\|_{L^2(\sigma_\Omega)}^2 + \delta.
\end{aligned}$$

By rearranging and recalling the choice of h and δ, we finally see

$$\|f_\sigma - h\|_{L^2(\sigma_\Omega)}^2 + 2\delta \leq \|f_\sigma - g\|_{L^2(\sigma_\Omega)} \leq \|f_\sigma - h\|_{L^2(\sigma_\Omega)}^2 + \delta,$$

which is the desired contradiction. □

D.7. Proof of Proposition 3.7

The main ingredient of the proof will be to show that one can replace a given sequence of networks with C-bounded scaling weights by another sequence with C-bounded scaling weights that also has bounded biases. Then one can apply Proposition 3.5.

Lemma D.2 *Let $S = (d, N_1, \ldots, N_L)$ be a neural network architecture, let $C > 0$ and let $\Omega \subset \mathbb{R}^d$ be measurable and bounded. Let μ be a finite Borel measure on Ω with $\mu(\Omega) > 0$. Finally, let $\varrho : \mathbb{R} \to \mathbb{R},\ x \mapsto \max\{0, x\}$ denote the ReLU activation function.*

Let $(\Phi_n)_{n\in\mathbb{N}}$ be a sequence of networks in $\mathcal{NN}(S)$ with C-bounded scaling weights and such that there exists some $M > 0$ with $\|\mathrm{R}_\varrho^\Omega(\Phi_n)\|_{L^1(\mu)} \leq M$ for all $n \in \mathbb{N}$.

Then, there is an infinite set $I \subset \mathbb{N}$ and a family of networks $(\Psi_n)_{n\in I} \subset \mathcal{NN}(S)$ with C-bounded scaling weights which satisfies $\mathrm{R}_\varrho^\Omega(\Phi_n) = \mathrm{R}_\varrho^\Omega(\Psi_n)$ for $n \in I$ and such that $\|\Psi_n\|_{\mathrm{total}} \leq C'$ for all $n \in I$ and a suitable constant $C' > 0$.

Proof Set $N_0 := d$. Since Ω is bounded, there is some $R > 0$ with $\|x\|_{\ell^\infty} \leq R$ for all $x \in \Omega$. In the following, we will use without further comment the estimate $\|Ax\|_{\ell^\infty} \leq k \cdot \|A\|_{\max} \cdot \|x\|_{\ell^\infty}$ which is valid for $A \in \mathbb{R}^{n\times k}$ and $x \in \mathbb{R}^k$.

Below, we will show by induction on $m \in \{0, \dots, L-1\}$ that for all $m \in \{0, \dots, L-1\}$, there is an infinite subset $I_m \subset \mathbb{N}$, and a family of networks $\left(\Psi_n^{(m)}\right)_{n\in I_m} \subset \mathcal{NN}(S)$ of the form

$$\Psi_n^{(m)} = \left((B_1^{(n,m)}, c_1^{(n,m)}), \dots, (B_L^{(n,m)}, c_L^{(n,m)})\right) \tag{D.11}$$

with the following properties:

(A) We have $\mathrm{R}_\varrho^\Omega\left(\Psi_n^{(m)}\right) = \mathrm{R}_\varrho^\Omega(\Phi_n)$ for all $n \in I_m$;

(B) each network $\Psi_n^{(m)}$, $n \in I_m$, has C-bounded scaling weights;

(C) there is a constant $C_m > 0$ with $\left\|c_\ell^{(n,m)}\right\|_{\ell^\infty} \leq C_m$ for all $n \in I_m$ and all $\ell \in \{1, \dots, m\}$.

Once this is shown, we set $I := I_{L-1}$ and $\Psi_n := \Psi_n^{(L-1)}$ for $n \in I$. Clearly, Ψ_n has C-bounded scaling weights and satisfies $\mathrm{R}_\varrho^\Omega(\Psi_n) = \mathrm{R}_\varrho^\Omega(\Phi_n)$, so that it remains to show $\|\Psi_n\|_{\mathrm{total}} \leq C'$, for which it suffices to show $\|c_L^{(n,L-1)}\|_{\ell^\infty} \leq C''$ for some $C'' > 0$ and all $n \in I$, since we have $\|c_\ell^{(n,L-1)}\|_{\ell^\infty} \leq C_{L-1}$ for all $\ell \in \{1, \dots, L-1\}$.

Now, note for $\ell \in \{1, \dots, L-1\}$ and $x \in \mathbb{R}^{N_{\ell-1}}$ that $T_\ell^{(n,L-1)}(x) := B_\ell^{(n,L-1)} x + c_\ell^{(n,L-1)}$ satisfies

$$\left\|T_\ell^{(n,L-1)}(x)\right\|_{\ell^\infty} \leq N_{\ell-1} \cdot C \cdot \|x\|_{\ell^\infty} + C_{L-1}.$$

Since Ω is bounded, and since $|\varrho(x)| \leq |x|$ for all $x \in \mathbb{R}$, there is thus a constant $C'_{L-1} > 0$ such that if we set

$$\beta^{(n)}(x) := \left(\varrho \circ T_{L-1}^{(n,L-1)} \circ \cdots \circ \varrho \circ T_1^{(n,L-1)}\right)(x) \quad \text{for} \quad x \in \Omega,$$

then $\|\beta^{(n)}(x)\|_{\ell^\infty} \leq C'_{L-1}$ for all $x \in \Omega$ and all $n \in I$.

For arbitrary $i \in \{1, \dots, N_L\}$ and $x \in \Omega$, this implies

$$\begin{aligned}
\left|[\mathrm{R}_\varrho^\Omega(\Phi_n)(x)]_i\right| = \left|[\mathrm{R}_\varrho^\Omega(\Psi_n^{(L-1)})(x)]_i\right| &= \left|\left\langle \left(B_L^{(n,L-1)}\right)_i, \beta^{(n)}(x)\right\rangle + (c_L^{(n,L-1)})_i\right| \\
&\geq \left|(c_L^{(n,L-1)})_i\right| - \left|\left\langle \left(B_L^{(n,L-1)}\right)_i, \beta^{(n)}(x)\right\rangle\right| \\
&\geq \left|(c_L^{(n,L-1)})_i\right| - N_{L-1} \cdot C \cdot \|\beta^{(n)}(x)\|_{\ell^\infty} \\
&\geq \left|(c_L^{(n,L-1)})_i\right| - N_{L-1} \cdot C \cdot C'_{L-1}.
\end{aligned}$$

Since by assumption $\|\mathrm{R}_{\varrho}^{\Omega}(\Phi_n)\|_{L^1(\mu)} \leq M$ and $\mu(\Omega) > 0$, we see that $\left(c_L^{(n,L-1)}\right)_{n\in I}$ must be a bounded sequence.

Thus, it remains to construct the networks $\Psi_n^{(m)}$ for $n \in I_m$ (and the sets I_m) for $m \in \{0, \dots, L-1\}$ with the properties (A)–(C) from above.

For the start of the induction ($m = 0$), we can simply take $I_0 := \mathbb{N}$, $\Psi_n^{(0)} := \Phi_n$, and $C_0 > 0$ arbitrary, since condition (C) is void in this case.

Now, assume that a family of networks $(\Psi_n^{(m)})_{n\in I_m}$ as in Eq. (D.11) with an infinite subset $I_m \subset \mathbb{N}$ and satisfying conditions (A)–(C) has been constructed for some $m \in \{0, \dots, L-2\}$. In particular, $L \geq 2$.

For brevity, set $T_\ell^{(n)} : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell},\ x \mapsto B_\ell^{(n,m)} x + c_\ell^{(n,m)}$ for $\ell \in \{1, \dots, L\}$, and $\varrho_L := \mathrm{id}_{\mathbb{R}^{N_L}}$, and let $\varrho_\ell := \varrho \times \cdots \times \varrho$ denote the N_ℓ-fold Cartesian product of ϱ for $\ell \in \{1, \dots, L-1\}$. Furthermore, let us define $\beta_n := \varrho_m \circ T_m^{(n)} \circ \cdots \circ \varrho_1 \circ T_1^{(n)} : \mathbb{R}^d \to \mathbb{R}^{N_m}$. Note $\|\varrho_\ell(x)\|_{\ell^\infty} \leq \|x\|_{\ell^\infty}$ for all $x \in \mathbb{R}^{N_\ell}$. Additionally, observe for $n \in I_m$, $\ell \in \{1, \dots, m\}$ and $x \in \mathbb{R}^{N_{\ell-1}}$ that

$$\left\| T_\ell^{(n)}(x) \right\|_{\ell^\infty} = \left\| B_\ell^{(n,m)} x + c_\ell^{(n,m)} \right\|_{\ell^\infty} \leq N_{\ell-1} \cdot C \cdot \|x\|_{\ell^\infty} + C_m.$$

Combining these observations, and recalling that Ω is bounded, we easily see that there is some $R' > 0$ with $\|\beta_n(x)\|_{\ell^\infty} \leq R'$ for all $x \in \Omega$ and $n \in I_m$.

Next, since $\left(c_{m+1}^{(n,m)}\right)_{n\in I_m}$ is an infinite family in $\mathbb{R}^{N_{m+1}} \subset [-\infty, \infty]^{N_{m+1}}$, we can find (by compactness) an infinite subset $I_m^{(0)} \subset I_m$ such that $c_{m+1}^{(n,m)} \to c_{m+1} \in [-\infty, \infty]^{N_{m+1}}$ as $n \to \infty$ in the set $I_m^{(0)}$.

Our goal is to construct vectors $d^{(n)}, e^{(n)} \in \mathbb{R}^{N_{m+1}}$, matrices $C^{(n)} \in \mathbb{R}^{N_{m+1}\times N_m}$, and an infinite subset $I_{m+1} \subset I_m^{(0)}$ such that $\|C^{(n)}\|_{\max} \leq C$ for all $n \in I_{m+1}$, such that $\left(d^{(n)}\right)_{n\in I_{m+1}}$ is a bounded family, and such that we have

$$\varrho_{m+1}\left(T_{m+1}^{(n)}(x)\right) = \varrho_{m+1}\left(C^{(n)} x + d^{(n)}\right) + e^{(n)} \quad \text{for all } x \in \mathbb{R}^{N_m} \text{ with } \|x\|_{\ell^\infty} \leq R', \tag{D.12}$$

for all $n \in I_{m+1}$.

Once $d^{(n)}, e^{(n)}, C^{(n)}$ are constructed, we can choose $\Psi_n^{(m+1)}$ as in Eq. (D.11), where we define

$$B_\ell^{(n,m+1)} := B_\ell^{(n,m)}$$

and

$$c_\ell^{(n,m+1)} := c_\ell^{(n,m)}$$

for $\ell \in \{1, \dots, L\} \setminus \{m+1, m+2\}$, and finally

$$B_{m+1}^{(n,m+1)} := C^{(n)}, \quad B_{m+2}^{(n,m+1)} := B_{m+2}^{(n,m)}, \quad c_{m+1}^{(n,m+1)} := d^{(n)},$$

and

$$c_{m+2}^{(n,m+1)} := c_{m+2}^{(n,m)} + B_{m+2}^{(n,m+1)} e^{(n)}$$

for $n \in I_{m+1}$. Indeed, these choices clearly ensure $\left\| B_\ell^{(n,m+1)} \right\|_{\max} \leq C$ for all $\ell \in \{1, \dots, L\}$, as well as $\left\| c_\ell^{(n,m+1)} \right\|_{\ell^\infty} \leq C_{m+1}$ for all $\ell \in \{1, \dots, m+1\}$ and $n \in I_{m+1}$, for a suitable constant $C_{m+1} > 0$.

Finally, since $\|\beta_n(x)\|_{\ell^\infty} \leq R'$ for all $x \in \Omega$ and $n \in I_m$, Eq. (D.12) implies

$$\begin{aligned}
&T_{m+2}^{(n)} \left(\varrho_{m+1} \left(T_{m+1}^{(n)} \left(\beta_n(x) \right) \right) \right) \\
&= T_{m+2}^{(n)} \left(\varrho_{m+1} \left(C^{(n)} \beta_n(x) + d^{(n)} \right) + e^{(n)} \right) \\
&= B_{m+2}^{(n,m)} \left(\varrho_{m+1} \left(B_{m+1}^{(n,m+1)} \beta_n(x) + c_{m+1}^{(n,m+1)} \right) + e^{(n)} \right) + c_{m+2}^{(n,m)} \\
&= B_{m+2}^{(n,m+1)} \left(\varrho_{m+1} \left(B_{m+1}^{(n,m+1)} \beta_n(x) + c_{m+1}^{(n,m+1)} \right) \right) + c_{m+2}^{(n,m+1)}
\end{aligned}$$

for all $x \in \Omega$ and $n \in I_{m+1}$. By recalling the definition of β_n, and by noting that $B_\ell^{(n,m+1)}, c_\ell^{(n,m+1)}$ are identical to $B_\ell^{(n,m)}, c_\ell^{(n,m)}$ for $\ell \in \{1, \dots, L\} \setminus \{m+1, m+2\}$, this easily yields

$$\mathrm{R}_\varrho^\Omega \left(\Psi_n^{(m+1)} \right) = \mathrm{R}_\varrho^\Omega \left(\Psi_n^{(m)} \right) = \mathrm{R}_\varrho^\Omega (\Phi_n) \qquad \text{for all } n \in I_{m+1}.$$

Thus, it remains to construct $d^{(n)}, e^{(n)}, C^{(n)}$ for $n \in I_{m+1}$ (and the set I_{m+1} itself) as described around Eq. (D.12). To this end, for $n \in I_m^{(0)}$ and $k \in \{1, \dots, N_{m+1}\}$, define

$$d_k^{(n)} := \begin{cases} R' \cdot N_m C, & \text{if } (c_{m+1})_k = \infty, \\ 0, & \text{if } (c_{m+1})_k = -\infty, \\ \left(c_{m+1}^{(n,m)} \right)_k, & \text{if } (c_{m+1})_k \in \mathbb{R}, \end{cases},$$

and

$$e_k^{(n)} := \begin{cases} \left(c_{m+1}^{(n,m)} \right)_k - R' \cdot N_m C, & \text{if } (c_{m+1})_k = \infty, \\ 0, & \text{if } (c_{m+1})_k = -\infty, \\ 0, & \text{if } (c_{m+1})_k \in \mathbb{R}, \end{cases}$$

as well as

$$C_{k,-}^{(n)} := \begin{cases} \left(B_{m+1}^{(n,m)} \right)_{k,-}, & \text{if } (c_{m+1})_k = \infty, \\ 0 \in \mathbb{R}^{N_m}, & \text{if } (c_{m+1})_k = -\infty, \\ \left(B_{m+1}^{(n,m)} \right)_{k,-}, & \text{if } (c_{m+1})_k \in \mathbb{R}. \end{cases}$$

To see that these choices indeed fulfill the conditions outlined around Eq. (D.12) for a suitable choice of $I_{m+1} \subset I_m^{(0)}$, first note that $\left(d^{(n)}\right)_{n \in I_m^{(0)}}$ is indeed a bounded family. Furthermore, $\left|C_{k,i}^{(n)}\right| \leq \left|(B_{m+1}^{(n,m)})_{k,i}\right|$ for all $k \in \{1, \dots, N_{m+1}\}$ and $i \in \{1, \dots, N_m\}$, which easily implies $\|C^{(n)}\|_{\max} \leq \|B_{m+1}^{(n,m)}\|_{\max} \leq C$ for all $n \in I_m^{(0)}$. Thus, it remains to verify Eq. (D.12) itself. But the estimate $\|B_{m+1}^{(n,m)}\|_{\max} \leq C$ also implies

$$\left|\left(B_{m+1}^{(n,m)}\, x\right)_k\right| \leq N_m\, C \cdot \|x\|_{\ell^\infty} \leq N_m\, C \cdot R', \tag{D.13}$$

for all $k \in \underline{N_{m+1}}$ and all $x \in \mathbb{R}^{N_m}$ with $\|x\|_{\ell^\infty} \leq R'$. As a final preparation, note that $\varrho_{m+1} = \varrho \times \cdots \times \varrho$ is a Cartesian product of ReLU functions, since $m \leq L-2$. Now, for $k \in \{1, \dots, N_{m+1}\}$ there are three cases:

Case 1: We have $(c_{m+1})_k = \infty$. Thus, there is some $n_k \in \mathbb{N}$ such that $\left(c_{m+1}^{(n,m)}\right)_k \geq R' \cdot N_m C$ for all $n \in I_m^{(0)}$ with $n \geq n_k$. In view of Eq. (D.13), this implies $\left(T_{m+1}^{(n)}(x)\right)_k = \left(B_{m+1}^{(n,m)} x + c_{m+1}^{(n,m)}\right)_k \geq 0$, and hence

$$\left[\varrho_{m+1}\left(\, T_{m+1}^{(n)}(x)\,\right)\right]_k = \left(B_{m+1}^{(n,m)}\, x + c_{m+1}^{(n,m)}\right)_k = \left(\varrho_{m+1}\left(C^{(n)}\, x + d^{(n)}\right) + e^{(n)}\right)_k,$$

where the last step used our choice of $d^{(n)}$, $e^{(n)}$, $C^{(n)}$, and the fact that $\left(C^{(n)} x + d^{(n)}\right)_k \geq 0$ by Eq. (D.13).

Case 2: We have $(c_{m+1})_k = -\infty$. This implies that there is some $n_k \in \mathbb{N}$ with $\left(c_{m+1}^{(n,m)}\right)_k \leq -R' \cdot N_m C$ for all $n \in I_m^{(0)}$ with $n \geq n_k$. Because of Eq. (D.13), this yields $\left(T_{m+1}^{(n)}(x)\right)_k = \left(B_{m+1}^{(n,m)} x + c_{m+1}^{(n,m)}\right)_k \leq 0$, and hence

$$\left[\varrho_{m+1}\left(T_{m+1}^{(n)}(x)\right)\right]_k = 0 = \left[\varrho_{m+1}\left(C^{(n)} x + d^{(n)}\right) + e^{(n)}\right]_k,$$

where the last step used our choice of $d^{(n)}$, $e^{(n)}$, $C^{(n)}$.

Case 3: We have $(c_{m+1})_k \in \mathbb{R}$. In this case, set $n_k := 1$, and note by our choice of $d^{(n)}$, $e^{(n)}$, $C^{(n)}$ for $n \in I_m^{(0)}$ with $n \geq 1 = n_k$ that

$$\left[\varrho_{m+1}\left(C^{(n)} x + d^{(n)}\right) + e^{(n)}\right]_k = \left[\varrho_{m+1}\left(B_{m+1}^{(n,m)} x + c_{m+1}^{(n,m)}\right)\right]_k = \left[\varrho_{m+1}\left(T_{m+1}^{(n)}(x)\right)\right]_k.$$

Overall, we have thus shown that Eq. (D.12) is satisfied for all $n \in I_{m+1}$, where

$$I_{m+1} := \left\{ n \in I_m^{(0)} \,:\, n \geq \max\left\{n_k \,:\, k \in \{1, \dots, N_{m+1}\}\right\} \right\}$$

is clearly an infinite set, since $I_m^{(0)}$ is. □

Using Lemma D.2, we can now easily show that the set $\mathcal{RNN}_\varrho^{\Omega,C}(S)$ is closed in $L^p(\mu; \mathbb{R}^{N_L})$ and in $C(\Omega; \mathbb{R}^{N_L})$: Let $\mathcal{Y}$ denote either $L^p(\mu; \mathbb{R}^{N_L})$ for some $p \in [1, \infty]$ and some finite Borel measure μ on Ω, or $C(\Omega; \mathbb{R}^{N_L})$, where we assume in the latter case that Ω is compact and set $\mu = \delta_{x_0}$ for a fixed $x_0 \in \Omega$. Note that we can

assume $\mu(\Omega) > 0$, since otherwise the claim is trivial. Let $(f_n)_{n\in\mathbb{N}}$ be a sequence in $\mathcal{RNN}_\varrho^{\Omega,C}(S)$ which satisfies $f_n \to f$ for some $f \in \mathcal{Y}$, with convergence in $\mathcal{Y}$. Thus, $f_n = \mathrm{R}_\varrho^\Omega(\Phi_n)$ for a suitable sequence $(\Phi_n)_{n\in\mathbb{N}}$ in $\mathcal{NN}(S)$ with C-bounded scaling weights.

Since $(f_n)_{n\in\mathbb{N}} = \left(\mathrm{R}_\varrho^\Omega(\Phi_n)\right)_{n\in\mathbb{N}}$ is convergent in $\mathcal{Y}$, it is also bounded in $\mathcal{Y}$. But since Ω is bounded and μ is a finite measure, it is not hard to see $\mathcal{Y} \hookrightarrow L^1(\mu)$, so that we get $\|\mathrm{R}_\varrho^\Omega(\Phi_n)\|_{L^1(\mu)} \leq M$ for all $n \in \mathbb{N}$ and a suitable constant $M > 0$.

Therefore, Lemma D.2 yields an infinite set $I \subset \mathbb{N}$ and networks $(\Psi_n)_{n\in I} \subset \mathcal{NN}(S)$ with C-bounded scaling weights such that $f_n = \mathrm{R}_\varrho^\Omega(\Psi_n)$ and $\|\Psi_n\|_{\mathrm{total}} \leq C'$ for all $n \in I$ and a suitable $C' > 0$.

Hence, $(\Psi_n)_{n\in I}$ is a bounded, infinite family in the finite dimensional vector space $\mathcal{NN}(S)$. Thus, there is a further infinite set $I_1 \subset I$ such that $\Psi_n \to \Psi \in \mathcal{NN}(S)$ as $n \to \infty$ in I_1.

But since Ω is bounded, say $\Omega \subset [-R, R]^d$, the realization map

$$\mathrm{R}_\varrho^{[-R,R]^d} : \mathcal{NN}(S) \to C([-R, R]^d; \mathbb{R}^{N_L}), \Phi \mapsto \mathrm{R}_\varrho^{[-R,R]^d}(\Phi)$$

is continuous (even locally Lipschitz continuous); see Proposition 4.1, which will be proved independently. Hence, $\mathrm{R}_\varrho^{[-R,R]^d}(\Psi_n) \to \mathrm{R}_\varrho^{[-R,R]^d}(\Psi)$ as $n \to \infty$ in I_1, with uniform convergence. This easily implies $f_n = \mathrm{R}_\varrho^\Omega(\Psi_n) \to \mathrm{R}_\varrho^\Omega(\Psi)$, with convergence in $\mathcal{Y}$ as $n \to \infty$ in I_1. Hence, $f = \mathrm{R}_\varrho^\Omega(\Psi) \in \mathcal{RNN}_\varrho^{C,\Omega}(S)$. □

D.8. Proof of Theorem 3.8

For the proof of Theorem 3.8, we will use a careful analysis of the **singularity hyperplanes** of functions of the form $x \mapsto \varrho_a(\langle\alpha, x\rangle + \beta)$, that is, the hyperplane on which this function is not differentiable. To simplify this analysis, we first introduce a convenient terminology and collect quite a few auxiliary results.

Definition D.3 For $\alpha, \widetilde{\alpha} \in S^{d-1}$ and $\beta, \widetilde{\beta} \in \mathbb{R}$, we write $(\alpha, \beta) \sim (\widetilde{\alpha}, \widetilde{\beta})$ if there is a $\varepsilon \in \{\pm 1\}$ such that $(\alpha, \beta) = \varepsilon \cdot (\widetilde{\alpha}, \widetilde{\beta})$.

Furthermore, for $a \geq 0$ and with $\varrho_a : \mathbb{R} \to \mathbb{R}, x \mapsto \max\{x, ax\}$ denoting the parametric ReLU, we set

$$S_{\alpha,\beta} := \left\{x \in \mathbb{R}^d : \langle\alpha, x\rangle + \beta = 0\right\} \quad \text{and} \quad h_{\alpha,\beta}^{(a)} : \mathbb{R}^d \to \mathbb{R}, x \mapsto \varrho_a(\langle\alpha, x\rangle + \beta).$$

Moreover, we define

$$W_{\alpha,\beta}^+ := \{x \in \mathbb{R}^d : \langle\alpha, x\rangle + \beta > 0\} \quad \text{and} \quad W_{\alpha,\beta}^- := \{x \in \mathbb{R}^d : \langle\alpha, x\rangle + \beta < 0\},$$

and finally

$$\begin{aligned} U_{\alpha,\beta}^{(\varepsilon)} &:= \left\{x \in \mathbb{R}^d : |\langle\alpha, x\rangle + \beta| \geq \varepsilon\right\}, \quad U_{\alpha,\beta}^{(\varepsilon,+)} \\ &:= U_{\alpha,\beta}^{(\varepsilon)} \cap W_{\alpha,\beta}^+ \quad \text{and} \quad U_{\alpha,\beta}^{(\varepsilon,-)} := U_{\alpha,\beta}^{(\varepsilon)} \cap W_{\alpha,\beta}^-, \end{aligned}$$

for $\epsilon > 0$

Lemma D.3 *Let $(\alpha, \beta) \in S^{d-1} \times \mathbb{R}$ and $x_0 \in S_{\alpha,\beta}$. Also, let $(\alpha_1, \beta_1), \ldots, (\alpha_N, \beta_N) \in S^{d-1} \times \mathbb{R}$ with $(\alpha_\ell, \beta_\ell) \nsim (\alpha, \beta)$ for all $\ell \in \underline{N}$. Then, there exists $z \in \mathbb{R}^d$ satisfying*

$$\langle z, \alpha \rangle = 0 \quad \text{and} \quad \langle z, \alpha_j \rangle \neq 0 \ \ \forall\, j \in \underline{N} \text{ with } x_0 \in S_{\alpha_j,\beta_j}.$$

Proof By discarding those (α_j, β_j) for which $x_0 \notin S_{\alpha_j,\beta_j}$, we can assume that $x_0 \in S_{\alpha_j,\beta_j}$ for all $j \in \underline{N}$.

Assume toward a contradiction that the claim of the lemma is false; that is,

$$\alpha^\perp = \bigcup_{j=1}^{N} \left\{ z \in \alpha^\perp \,:\, \langle z, \alpha_j \rangle = 0 \right\}, \tag{D.14}$$

where $\alpha^\perp := \{z \in \mathbb{R}^d \,:\, \langle z, \alpha \rangle = 0\}$. Since $\alpha^\perp$ is a closed subset of $\mathbb{R}^d$ and thus a complete metric space, and since the right-hand side of (D.14) is a countable (in fact, finite) union of closed sets, the Baire category theorem (see [26, Theorem 5.9]) shows that there are $j \in \underline{N}$ and $\varepsilon > 0$ such that

$$V := \left\{ z \in \alpha^\perp \,:\, \langle z, \alpha_j \rangle = 0 \right\} \supset B_\varepsilon(x) \cap \alpha^\perp \qquad \text{for some} \quad x \in V.$$

But since V is a vector space, this easily implies $V = \alpha^\perp$, that is, $\langle z, \alpha_j \rangle = 0$ for all $z \in \alpha^\perp$. In other words, $\alpha^\perp \subset \alpha_j^\perp$, and then $\alpha^\perp = \alpha_j^\perp$ by a dimension argument, since $\alpha, \alpha_j \neq 0$.

Hence, $\operatorname{span} \alpha = (\alpha^\perp)^\perp = (\alpha_j^\perp)^\perp = \operatorname{span} \alpha_j$. Because of $|\alpha| = |\alpha_j| = 1$, we thus see $\alpha = \varepsilon\, \alpha_j$ for some $\varepsilon \in \{\pm 1\}$. Finally, since $x_0 \in S_{\alpha,\beta} \cap S_{\alpha_j,\beta_j}$, we see $\beta = -\langle \alpha, x_0 \rangle = -\varepsilon \langle \alpha_j, x_0 \rangle = \varepsilon \beta_j$, and thus $(\alpha, \beta) = \varepsilon(\alpha_j, \beta_j)$, in contradiction to $(\alpha, \beta) \nsim (\alpha_j, \beta_j)$. □

Lemma D.4 *Let $(\alpha, \beta) \in S^{d-1} \times \mathbb{R}$ and $(\alpha_1, \beta_1), \ldots, (\alpha_N, \beta_N) \in S^{d-1} \times \mathbb{R}$ with $(\alpha_i, \beta_i) \nsim (\alpha, \beta)$ for all $i \in \underline{N}$. Furthermore, let $U \subset \mathbb{R}^d$ be open with $S_{\alpha,\beta} \cap U \neq \varnothing$.*

Then, there is $\varepsilon > 0$ satisfying

$$U \cap S_{\alpha,\beta} \cap \bigcap_{j=1}^{N} U^{(\varepsilon)}_{\alpha_j,\beta_j} \neq \varnothing.$$

Proof By assumption, there exists $x_0 \in U \cap S_{\alpha,\beta}$. Next, Lemma D.3 yields $z \in \mathbb{R}^d$ such that $\langle z, \alpha \rangle = 0$ and $\langle z, \alpha_j \rangle \neq 0$ for all $j \in \underline{N}$ with $x_0 \in S_{\alpha_j,\beta_j}$. Note that this implies $\langle \alpha, x_0 + tz \rangle + \beta = \langle \alpha, x_0 \rangle + \beta = 0$ and hence $x_0 + tz \in S_{\alpha,\beta}$ for all $t \in \mathbb{R}$.

Next, let $J := \{j \in \underline{N} \,:\, x_0 \notin S_{\alpha_j,\beta_j}\}$, so that $\langle \alpha_j, x_0 \rangle + \beta_j \neq 0$ for all $j \in J$. Thus, there are $\varepsilon_1, \delta > 0$ with $|\langle \alpha_j, x_0 + tz \rangle + \beta_j| \geq \varepsilon_1$ (that is, $x_0 + tz \in U^{(\varepsilon_1)}_{\alpha_j,\beta_j}$) for all $t \in \mathbb{R}$ with $|t| \leq \delta$ and all $j \in J$. Since U is open with $x_0 \in U$, we can shrink δ so that $x_0 + tz \in U$ for all $|t| \leq \delta$. Let $t := \delta$.

We claim that there is some $\varepsilon > 0$ such that $x := x_0 + tz \in U \cap S_{\alpha,\beta} \cap \bigcap_{j=1}^{N} U^{(\varepsilon)}_{\alpha_j,\beta_j}$. To see this, note for $j \in \underline{N} \setminus J$ that $x_0 \in S_{\alpha_j,\beta_j}$, and hence

$$|\langle x_0 + tz, \alpha_j\rangle + \beta_j| = |t| \cdot |\langle z, \alpha_j\rangle| \geq \delta \cdot \min_{\ell \in \underline{N}\setminus J} |\langle z, \alpha_\ell\rangle| =: \varepsilon_2 > 0,$$

since $\langle z, \alpha_j\rangle \neq 0$ for all $j \in \underline{N} \setminus J$, by choice of z. By combining all our observations, we see that $x_0 + tz \in U \cap S_{\alpha,\beta} \cap \bigcap_{j=1}^{N} U^{(\varepsilon)}_{\alpha_j,\beta_j}$ for $\varepsilon := \min\{\varepsilon_1, \varepsilon_2\} > 0$. □

Lemma D.5 *If $0 \leq a < 1$ and $(\alpha, \beta) \in S^{d-1} \times \mathbb{R}$, then $h^{(a)}_{\alpha,\beta}$ is* not *differentiable at any $x_0 \in S_{\alpha,\beta}$.*

Proof Assume toward a contradiction that $h^{(a)}_{\alpha,\beta}$ is differentiable at some $x_0 \in S_{\alpha,\beta}$. Then, the function $f : \mathbb{R} \to \mathbb{R}, t \mapsto h^{(a)}_{\alpha,\beta}(x_0 + t\alpha)$ is differentiable at $t = 0$. But since $x_0 \in S_{\alpha,\beta}$ and $\|\alpha\|_{\ell^2} = 1$, we have

$$f(t) = \varrho_a\big(\langle \alpha, x_0 + t\alpha\rangle + \beta\big) = \varrho_a(t) = \begin{cases} t, & \text{if } t \geq 0, \\ a\,t, & \text{if } t < 0, \end{cases}$$

for all $t \in \mathbb{R}$. This easily shows that f is not differentiable at $t = 0$, since the right-sided derivative is 1, while the left-sided derivative is $a \neq 1$. This is the desired contradiction. □

Lemma D.6 *Let $0 \leq a < 1$, and let $(\alpha_1, \beta_1), \ldots, (\alpha_N, \beta_N) \in S^{d-1} \times \mathbb{R}$ with $(\alpha_i, \beta_i) \nsim (\alpha_j, \beta_j)$ for $j \neq i$. Furthermore, let $U \subset \mathbb{R}^d$ be open with $U \cap S_{\alpha_i,\beta_i} \neq \varnothing$ for all $i \in \underline{N}$. Finally, set $h_i := h^{(a)}_{\alpha_i,\beta_i}|_U$ for $i \in \underline{N}$ with $h^{(a)}_{\alpha_i,\beta_i}$ as in Definition D.3, and let $h_{N+1} : U \to \mathbb{R}, x \mapsto 1$.*

Then, the family $(h_i)_{i=1,\ldots,N+1}$ is linearly independent.

Proof Assume toward a contradiction that $0 = \sum_{i=1}^{N+1} \gamma_i\, h_i$ for certain $\gamma_1, \ldots, \gamma_{N+1} \in \mathbb{R}$ with $\gamma_\ell \neq 0$ for some $\ell \in \underline{N+1}$. Note that if we had $\gamma_i = 0$ for all $i \in \underline{N}$, we would get $0 = \gamma_{N+1}\, h_{N+1} \equiv \gamma_{N+1}$, and thus $\gamma_i = 0$ for all $i \in \underline{N+1}$, a contradiction. Hence, there is some $j \in \underline{N}$ with $\gamma_j \neq 0$.

By Lemma D.4 there is some $\varepsilon > 0$ such that there exists

$$x_0 \in U \cap S_{\alpha_j,\beta_j} \cap \bigcap_{i \in \underline{N}\setminus\{j\}} U^{(\varepsilon)}_{\alpha_i,\beta_i}.$$

Therefore, $x_0 \in U \cap S_{\alpha_j,\beta_j} \cap V$ for the open set $V := \bigcap_{i \in \underline{N}\setminus\{j\}} \big(\mathbb{R}^d \setminus S_{\alpha_i,\beta_i}\big)$.

Because of $x_0 \in U \cap S_{\alpha_j,\beta_j}$, Lemma D.5 shows that $h^{(a)}_{\alpha_j,\beta_j}|_U$ is *not* differentiable at x_0. On the other hand, we have

$$h^{(a)}_{\alpha_j,\beta_j}|_U = h_j = -\gamma_j^{-1} \cdot \Big(\gamma_{N+1}\, h_{N+1} + \sum_{i \in \underline{N}\setminus\{j\}} \gamma_i\, h^{(a)}_{\alpha_i,\beta_i}|_U\Big),$$

where the right-hand side is differentiable at x_0, since each summand is easily seen to be differentiable on the open set V, with $x_0 \in V \cap U$. □

Lemma D.7 *Let $(\alpha, \beta) \in S^{d-1} \times \mathbb{R}$. If $\Omega \subset \mathbb{R}^d$ is compact with $\Omega \cap S_{\alpha,\beta} = \varnothing$, then there is some $\varepsilon > 0$ such that $\Omega \subset U^{(\varepsilon)}_{\alpha,\beta}$.*

Proof The continuous function $\Omega \to (0, \infty)$, $x \mapsto |\langle \alpha, x\rangle + \beta|$, which is well-defined by assumption, attains a minimum $\varepsilon = \min_{x\in\Omega} |\langle \alpha, x\rangle + \beta| > 0$. □

Lemma D.8 *Let $0 \leq a < 1$, let $(\alpha, \beta) \in S^{d-1} \times \mathbb{R}$, and let $U \subset \mathbb{R}^d$ be open with $U \cap S_{\alpha,\beta} \neq \varnothing$. Finally, let $f : U \to \mathbb{R}$ be continuous, and assume that f is affine-linear on $U \cap W^+_{\alpha,\beta}$ and on $U \cap W^-_{\alpha,\beta}$.*

Then, there are $c, \kappa \in \mathbb{R}$ and $\zeta \in \mathbb{R}^d$ such that

$$f(x) = c \cdot \varrho_a(\langle \alpha, x\rangle + \beta) + \langle \zeta, x\rangle + \kappa \qquad \text{for all } x \in U.$$

Proof By assumption, there are $\xi_1, \xi_2 \in \mathbb{R}^d$ and $\omega_1, \omega_2 \in \mathbb{R}$ satisfying

$$f(x) = \langle \xi_1, x\rangle + \omega_1 \quad \text{for } x \in U \cap W^+_{\alpha,\beta} \qquad \text{and} \qquad f(x) = \langle \xi_2, x\rangle + \omega_2 \quad \text{for } x \in U \cap W^-_{\alpha,\beta}.$$

Step 1: We claim that $U \cap S_{\alpha,\beta} \subset \overline{U \cap W^{\pm}_{\alpha,\beta}}$. Indeed, for arbitrary $x \in U \cap S_{\alpha,\beta}$, we have $x + t\alpha \in U$ for $t \in (-\varepsilon, \varepsilon)$ for a suitable $\varepsilon > 0$, since U is open. But since $x \in S_{\alpha,\beta}$ and $\|\alpha\|_{\ell^2} = 1$, we have $\langle x + t\alpha, \alpha\rangle + \beta = t$. Hence, $x + t\alpha \in U \cap W^+_{\alpha,\beta}$ for $t \in (0, \varepsilon)$ and $x + t\alpha \in U \cap W^-_{\alpha,\beta}$ for $t \in (-\varepsilon, 0)$. This easily implies the claim of this step.
Step 2: We claim that $\xi_1 - \xi_2 \in \operatorname{span} \alpha$. To see this, consider the modified function

$$\widetilde{f} : U \to \mathbb{R}, x \mapsto f(x) - (\langle \xi_2, x\rangle + \omega_2),$$

which is continuous and satisfies $\widetilde{f} \equiv 0$ on $U \cap W^-_{\alpha,\beta}$ and $\widetilde{f}(x) = \langle \theta, x\rangle + \omega$ on $U \cap W^+_{\alpha,\beta}$, where we defined $\theta := \xi_1 - \xi_2$ and $\omega := \omega_1 - \omega_2$.

Since we saw in Step 1 that $U \cap S_{\alpha,\beta} \subset \overline{U \cap W^{\pm}_{\alpha,\beta}}$, we thus get by continuity of $\widetilde{f}$ that

$$0 = \widetilde{f}(x) = \langle \theta, x\rangle + \omega \qquad \forall\, x \in U \cap S_{\alpha,\beta}.$$

But by assumption on U, there is some $x_0 \in U \cap S_{\alpha,\beta}$. For arbitrary $v \in \alpha^\perp$, we then have $x_0 + tv \in U \cap S_{\alpha,\beta}$ for all $t \in (-\varepsilon, \varepsilon)$ and a suitable $\varepsilon = \varepsilon(v) > 0$, since U is open. Hence, $0 = \langle \theta, x_0 + tv\rangle + \omega = t \cdot \langle \theta, v\rangle$ for all $t \in (-\varepsilon, \varepsilon)$, and thus $v \in \theta^\perp$. In other words, $\alpha^\perp \subset \theta^\perp$, and thus $\operatorname{span} \alpha = (\alpha^\perp)^\perp \supset (\theta^\perp)^\perp \ni \theta = \xi_1 - \xi_2$, as claimed in this step.
Step 3: In this step, we complete the proof. As seen in the previous step, there is some $c \in \mathbb{R}$ satisfying $c\alpha = (\xi_1 - \xi_2)/(1 - a)$. Now, set $\zeta := (\xi_2 - a\xi_1)/(1 - a)$ and

$\kappa := f(x_0) - \langle \zeta, x_0 \rangle$, where $x_0 \in U \cap S_{\alpha,\beta}$ is arbitrary. Finally, define

$$g : \mathbb{R}^d \to \mathbb{R}, x \mapsto c \cdot \varrho_a(\langle \alpha, x \rangle + \beta) + \langle \zeta, x \rangle + \kappa.$$

Because of $x_0 \in S_{\alpha,\beta}$, we then have $g(x_0) = \langle \zeta, x_0 \rangle + \kappa = f(x_0)$. Furthermore, since $\varrho_a(x) = x$ for $x \geq 0$, we see for all $x \in U \cap W^+_{\alpha,\beta}$ that

$$\begin{aligned} g(x) - f(x_0) = g(x) - g(x_0) &= c \cdot (\langle \alpha, x \rangle + \beta) + \langle \zeta, x - x_0 \rangle \\ (x_0 \in S_{\alpha,\beta}, \text{ i.e., } \langle \alpha, x_0 \rangle + \beta = 0) &= c \cdot \langle \alpha, x - x_0 \rangle + \langle \zeta, x - x_0 \rangle \\ &= \left\langle \frac{\xi_1 - \xi_2}{1-a} + \frac{\xi_2 - a\,\xi_1}{1-a}\,,\, x - x_0 \right\rangle \\ &= \langle \xi_1, x - x_0 \rangle = f(x) - f(x_0). \end{aligned} \tag{D.15}$$

Here, the last step used that $f(x) = \langle \xi_1, x \rangle + \omega_1$ for $x \in U \cap W^+_{\alpha,\beta}$, and that $x_0 \in U \cap S_{\alpha,\beta} \subset \overline{U \cap W^+_{\alpha,\beta}}$ by Step 1, so that we get $f(x_0) = \langle \xi_1, x_0 \rangle + \omega_1$ as well.

Likewise, since $\varrho_a(t) = a\,t$ for $t < 0$, we see for $x \in U \cap W^-_{\alpha,\beta}$ that

$$\begin{aligned} g(x) - f(x_0) = g(x) - g(x_0) &= ac \cdot (\langle \alpha, x \rangle + \beta) + \langle \zeta, x - x_0 \rangle \\ (\text{since } x_0 \in S_{\alpha,\beta}, \text{ i.e., } \langle \alpha, x_0 \rangle + \beta = 0) &= ac \langle \alpha, x - x_0 \rangle + \langle \zeta, x - x_0 \rangle \\ &= \left\langle a\frac{\xi_1 - \xi_2}{1-a} + \frac{\xi_2 - a\xi_1}{1-a}\,,\, x - x_0 \right\rangle \\ &= \langle \xi_2, x - x_0 \rangle = f(x) - f(x_0). \end{aligned} \tag{D.16}$$

In combination, Eqs. (D.15) and (D.16) show $f(x) = g(x)$ for all $x \in U \cap (W^+_{\alpha,\beta} \cup W^-_{\alpha,\beta})$. Since this set is dense in U by Step 1, we are done. □

With all of these preparations, we can finally prove Theorem 3.8.

Proof of Theorem 3.8 Since $\varrho_1 = \mathrm{id}_{\mathbb{R}}$, the result is trivial for $a = 1$, since the set $\mathcal{RNN}^{[-B,B]^d}_{\varrho_1}((d, N_0, 1))$ is just the set of all affine-linear maps $[-B, B]^d \to \mathbb{R}$. Furthermore, if $a > 1$, then

$$\varrho_a(x) = \max\{x, ax\} = a\,\varrho_{a^{-1}}(x),$$

and hence $\mathcal{RNN}^{[-B,B]^d}_{\varrho_a}((d, N_0, 1)) = \mathcal{RNN}^{[-B,B]^d}_{\varrho_{a^{-1}}}((d, N_0, 1))$. Therefore, we can assume $a < 1$ in the sequel. For brevity, let $\Omega := [-B, B]^d$. Then, each $\Phi \in \mathcal{NN}((d, N_0, 1))$ is of the form $\Phi = \big((A_1, b_1), (A_2, b_2)\big)$ with $A_1 \in \mathbb{R}^{N_0 \times d}$, $A_2 \in \mathbb{R}^{1 \times N_0}$, and $b_1 \in \mathbb{R}^{N_0}$, $b_2 \in \mathbb{R}^1$.

Let $(\Phi_n)_{n \in \mathbb{N}} \subset \mathcal{NN}((d, N_0, 1))$ with

$$\Phi_n = \big((\widetilde{A}^n_1, \widetilde{b}^n_1), (\widetilde{A}^n_2, \widetilde{b}^n_2)\big)$$

be such that $f_n := \mathrm{R}^{\Omega}_{\varrho_a}(\Phi_n)$ converges uniformly to some $f \in C(\Omega)$. Our goal is to prove $f \in \mathcal{RNN}^{\Omega}_{\varrho_a}((d, N_0, 1))$. The proof of this is divided into seven steps.

Step 1 (Normalizing the rows of the first layer): Our first goal is to normalize the rows of the matrices $\widetilde{A}^n_1$; that is, we want to change the parametrization of the network such that $\|(\widetilde{A}^n_1)_{i,-}\|_{\ell^2} = 1$ for all $i \in \underline{N_0}$. To see that this is possible, consider arbitrary $A \in \mathbb{R}^{M_1 \times M_2} \neq 0$ and $b \in \mathbb{R}^{M_1}$; then we obtain by the positive homogeneity of ϱ_a for all $C > 0$ that

$$\varrho_a(Ax + b) = C \cdot \varrho_a\left(\frac{A}{C}\, x + \frac{b}{C}\right) \text{ for all } x \in \mathbb{R}^{M_2}.$$

This identity shows that for each $n \in \mathbb{N}$, we can find a network

$$\widetilde{\Phi}_n = \left(\left(A^n_1, b^n_1\right), \left(A^n_2, b^n_2\right)\right) \in \mathcal{NN}((d, N_0, 1)),$$

such that the rows of A^n_1 are normalized, that is, $\|(A^n_1)_{i,-}\|_{\ell^2} = 1$ for all $i \in \underline{N_0}$, and such that

$$\mathrm{R}^{\Omega}_{\varrho_a}\left(\widetilde{\Phi}_n\right) = \mathrm{R}^{\Omega}_{\varrho_a}(\Phi_n) = f_n \quad \text{for all } n \in \mathbb{N}.$$

Step 2 (Extracting a partially convergent subsequence): By the Theorem of Bolzano-Weierstraß, there is a common subsequence of $(A^n_1)_{n\in\mathbb{N}}$ and $(b^n_1)_{n\in\mathbb{N}}$, denoted by $(A^{n_k}_1)_{k\in\mathbb{N}}$ and $(b^{n_k}_1)_{k\in\mathbb{N}}$, converging to $A_1 \in \mathbb{R}^{N_0 \times d}$ and $b_1 \in [-\infty, \infty]^{N_0}$, respectively.

For $j \in \underline{N_0}$, let $a_{k,j} \in \mathbb{R}^d$ denote the j-th row of $A^{n_k}_1$, and let $a_j \in \mathbb{R}^d$ denote the j-th row of A_1. Note that $\|a_{k,j}\|_{\ell^2} = \|a_j\|_{\ell^2} = 1$ for all $j \in \underline{N_0}$ and $k \in \mathbb{N}$. Next, let

$$J := \left\{ j \in \underline{N_0} \,:\, (b_1)_j \in \{\pm\infty\} \text{ or } \left[(b_1)_j \in \mathbb{R} \text{ and } S_{a_j,(b_1)_j} \cap \Omega^\circ = \varnothing\right]\right\},$$

where $\Omega^\circ = (-B, B)^d$ denotes the interior of Ω. Additionally, let $J^c := \underline{N_0} \setminus J$, and for $j, \ell \in J^c$ write $j \simeq \ell$ iff $(a_j, (b_1)_j) \sim (a_\ell, (b_1)_\ell)$, with the relation $\sim$ introduced in Definition D.3. Note that this makes sense, since $(b_1)_j \in \mathbb{R}$ if $j \in J^c$. Clearly, the relation $\simeq$ is an equivalence relation on J^c. Let $(J_i)_{i=1,\dots,r}$ denote the equivalence classes of the relation $\simeq$. For each $i \in \underline{r}$, choose $\alpha^{(i)} \in S^{d-1}$ and $\beta^{(i)} \in \mathbb{R}$ such that for each $j \in J_i$ there is a (unique) $\sigma_j \in \{\pm 1\}$ with $(a_j, (b_1)_j) = \sigma_j \cdot (\alpha^{(i)}, \beta^{(i)})$.

Step 3 (Handling the case of distinct singularity hyperplanes): Note that $r \leq |J^c| \leq N_0$. Before we continue with the general case, let us consider the special case where equality occurs, that is, where $r = N_0$. This means that $J = \varnothing$ (and hence $(b_1)_j \in \mathbb{R}$ and $\Omega^\circ \cap S_{a_j,(b_1)_j} \neq \varnothing$ for all $j \in \underline{N_0}$), and that each equivalence class J_i has precisely one element; that is, $(a_j, (b_1)_j) \nsim (a_\ell, (b_1)_\ell)$ for $j, \ell \in \underline{N_0}$ with $j \neq \ell$.

Therefore, Lemma D.6 shows that the functions $(h_j|_{\Omega^\circ})_{j=1,\dots,N_0+1}$, where we define $h_j := h^{(a)}_{a_j,(b_1)_j}|_\Omega$ for $j \in \underline{N_0}$ and $h_{N_0+1} : \Omega \to \mathbb{R}, x \mapsto 1$, are linearly independent. In particular, these functions are linearly independent when considered on all of

Ω. Thus, we can define a norm $\|\cdot\|_*$ on $\mathbb{R}^{N_0+1}$ by virtue of

$$\|c\|_* := \left\| c_{N_0+1} + \sum_{j=1}^{N_0} c_j \, h^{(a)}_{a_j,(b_1)_j} \right\|_{L^\infty(\Omega)} \qquad \text{for } c = (c_j)_{j=1,\dots,N_0+1} \in \mathbb{R}^{N_0+1}.$$

Since all norms on the finite dimensional vector space $\mathbb{R}^{N_0+1}$ are equivalent, there is some $\tau > 0$ with $\|c\|_* \geq \tau \cdot \|c\|_{\ell^1}$ for all $c \in \mathbb{R}^{N_0+1}$.

Now, recall that $a_{k,j} \to a_j$ and $b_1^{n_k} \to b_1 \in \mathbb{R}^{N_0}$ as $k \to \infty$. Since Ω is bounded, this implies for arbitrary $j \in \underline{N_0}$ and $h_j^{(k)} := h^{(a)}_{a_{k,j},(b_1^{n_k})_j}$ that $h_j^{(k)} \to h^{(a)}_{a_j,(b_1)_j}$ as $k \to \infty$, with uniform convergence on Ω. Thus, there is some $N_1 \in \mathbb{N}$ such that $\left\| h_j^{(k)} - h^{(a)}_{a_j,(b_1)_j} \right\|_{L^\infty(\Omega)} \leq \tau/2$ for all $k \geq N_1$ and $j \in \underline{N_0}$. Therefore, if $k \geq N_1$, we have

$$\begin{aligned}
&\left\| c_{N_0+1} + \sum_{j=1}^{N_0} c_j \, h_j^{(k)} \right\|_{L^\infty(\Omega)} \\
&\geq \left\| c_{N_0+1} + \sum_{j=1}^{N_0} c_j \, h^{(a)}_{a_j,(b_1)_j} \right\|_{L^\infty(\Omega)} - \left\| \sum_{j=1}^{N_0} c_j \left(h^{(a)}_{a_j,(b_1)_j} - h_j^{(k)} \right) \right\|_{L^\infty(\Omega)} \\
&\geq \tau \cdot \|c\|_{\ell^1} - \sum_{j=1}^{N_0} |c_j| \cdot \left\| h^{(a)}_{a_j,(b_1)_j} - h_j^{(k)} \right\|_{L^\infty(\Omega)} \\
&\geq \left(\tau - \frac{\tau}{2} \right) \cdot \|c\|_{\ell^1} = \frac{\tau}{2} \cdot \|c\|_{\ell^1} \qquad \text{for all } c = (c_j)_{j=1,\dots,N_0+1} \in \mathbb{R}^{N_0+1}.
\end{aligned}$$

Since $f_{n_k} = \mathrm{R}^{\Omega}_{\varrho a}\big(\widetilde{\Phi}_{n_k}\big) = b_2^{n_k} + \sum_{j=1}^{N_0} \left(A_2^{n_k}\right)_{1,j} h_j^{(k)}$ converges uniformly on Ω, we thus see that the sequence consisting of $(A_2^{n_k}, b_2^{n_k}) \in \mathbb{R}^{1\times N_0} \times \mathbb{R} \cong \mathbb{R}^{N_0+1}$ is bounded. Thus, there is a further subsequence $(n_{k_\ell})_{\ell\in\mathbb{N}}$ such that $A_2^{n_{k_\ell}} \to A_2 \in \mathbb{R}^{1\times N_0}$ and $b_2^{n_{k_\ell}} \to b_2 \in \mathbb{R}$ as $\ell \to \infty$. But this implies as desired that

$$\begin{aligned}
f = \lim_{\ell\to\infty} f_{n_{k_\ell}} &= \lim_{\ell\to\infty} \left[b_2^{n_{k_\ell}} + \sum_{j=1}^{N_0} \left(A_2^{n_{k_\ell}}\right)_{1,j} h_j^{(k_\ell)}\big|_\Omega \right] \\
&= b_2 + \sum_{j=1}^{N_0} (A_2)_{1,j} \, h^{(a)}_{a_j,(b_1)_j}|_\Omega \in \mathcal{RNN}^{\Omega}_{\varrho a}((d, N_0, 1)).
\end{aligned}$$

Step 4 (Showing that the j-th neuron is eventually affine-linear, for $j \in J$): Since Step 3 shows that the claim holds in case of $r = N_0$, we will from now on consider only the case where $r < N_0$.

For $j \in J$, there are two cases: In case of $(b_1)_j \in [0, \infty]$, define

$$\phi_j^{(k)} : \mathbb{R}^d \to \mathbb{R}, x \mapsto (A_2^{n_k})_{1,j} \cdot \left[\langle a_{k,j}, x \rangle + (b_1^{n_k})_j \right] \qquad \text{for all } k \in \mathbb{N}.$$

If otherwise $(b_1)_j \in [-\infty, 0)$, define

$$\phi_j^{(k)} : \mathbb{R}^d \to \mathbb{R}, x \mapsto a \cdot (A_2^{n_k})_{1,j} \cdot \left[\langle a_{k,j}, x\rangle + (b_1^{n_k})_j\right] \quad \text{for all } k \in \mathbb{N}.$$

Next, for arbitrary $0 < \delta < B$, we define $\Omega_\delta := [-(B-\delta), B-\delta]^d$. Note that since $S_{\alpha^{(i)},\beta^{(i)}} \cap \Omega^\circ \neq \varnothing$ for all $i \in \underline{r}$, there is some $\delta_0 > 0$ such that $S_{\alpha^{(i)},\beta^{(i)}} \cap (-(B-\delta), B-\delta)^d \neq \varnothing$ for all $i \in \underline{r}$ and all $0 < \delta \leq \delta_0$. For the remainder of this step, we will consider a fixed $\delta \in (0, \delta_0]$, and we claim that there is some $N_2 = N_2(\delta) > 0$ such that

$$(b_1)_j \neq 0 \quad \text{and} \quad \operatorname{sign}\left(\langle a_{k,j}, x\rangle + (b_1^{n_k})_j\right) = \operatorname{sign}\left((b_1^{n_k})_j\right) \neq 0, \tag{D.17}$$

for all $j \in J$, $k \geq N_2$ and $x \in \Omega_\delta$, where $\operatorname{sign} x = 1$ if $x > 0$, $\operatorname{sign} x = -1$ if $x < 0$, and $\operatorname{sign} 0 = 0$. Note that once this is shown, it is not hard to see that there is some $N_3 = N_3(\delta) \in \mathbb{N}$ such that

$$(A_2^{n_k})_{1,j} \cdot \varrho_a\left(\langle a_{k,j}, x\rangle + (b_1^{n_k})_j\right) = \phi_j^{(k)}(x) \quad \text{for all } j \in J,\ k \geq N_3, \text{ and } x \in \Omega_\delta,$$

simply because $(b_1^{n_k})_j \to (b_1)_j$ and $\varrho_a(x) = x$ if $x \geq 0$, and $\varrho_a(x) = ax$ if $x < 0$. Therefore, for the affine-linear function

$$\begin{aligned} g_{r+1}^{(k)} &:= b_2^{n_k} + \sum_{j \in J} \phi_j^{(k)} : \mathbb{R}^d \to \mathbb{R} \\ &\text{we have } g_{r+1}^{(k)}(x) = b_2^{n_k} + \sum_{j \in J} (A_2^{n_k})_{1,j}\, \varrho_a\left(\langle a_{k,j}, x\rangle + (b_1^{n_k})_j\right) \text{ for all } k \\ &\geq N_3(\delta) \text{ and } x \in \Omega_\delta. \end{aligned} \tag{D.18}$$

To prove Eq. (D.17), we distinguish two cases for each $j \in J$; by definition of J, these are the only two possible cases:
Case 1: We have $(b_1)_j \in \{\pm\infty\}$. In this case, the first part of Eq. (D.17) is trivially satisfied. To prove the second part, note that because of $(b_1^{n_k})_j \to (b_1)_j \in \{-\infty, \infty\}$, there is some $k_j \in \mathbb{N}$ with $|(b_1^{n_k})_j| \geq 2d \cdot B$ for all $k \geq k_j$. Since we have $\|a_{k,j}\|_{\ell^2} = 1$ and $\|x\|_{\ell^2} \leq \sqrt{d}B \leq dB$ for $x \in \Omega$, this implies

$$|\langle a_{k,j}, x\rangle + (b_1^{n_k})_j| \geq |(b_1^{n_k})_j| - |\langle a_{k_j}, x\rangle| \geq 2d \cdot B - \|x\|_{\ell^2} \geq dB > 0,$$

for all $x \in \Omega = [-B, B]^d$ and $k \geq k_j$. Now, since the function $x \mapsto \langle a_{k,j}, x\rangle + (b_1^{n_k})_j$ is continuous, since Ω is connected (in fact convex), and since $0 \in \Omega$, this implies $\operatorname{sign}(\langle a_{k,j}, x\rangle + (b_1^{n_k})_j) = \operatorname{sign}(b_1^{n_k})_j$ for all $x \in \Omega$ and $k \geq k_j$.
Case 2: We have $(b_1)_j \in \mathbb{R}$, but $S_{a_j,(b_1)_j} \cap \Omega^\circ = \varnothing$, and hence $S_{a_j,(b_1)_j} \cap \Omega_\delta = \varnothing$. In view of Lemma D.7, there is thus some $\varepsilon_{j,\delta} > 0$ satisfying $\Omega_\delta \subset U_{a_j,(b_1)_j}^{(\varepsilon_{j,\delta})}$; that is, $|\langle a_j, x\rangle + (b_1)_j| \geq \varepsilon_{j,\delta} > 0$ for all $x \in \Omega_\delta$. In particular, since $0 \in \Omega_\delta$, this implies $|(b_1)_j| \geq \varepsilon_{j,\delta} > 0$ and hence $(b_1)_j \neq 0$, as claimed in the first part of Eq. (D.17).

To prove the second part, note that because of $a_{k,j} \to a_j$ and $(b_1^{n_k})_j \to (b_1)_j$ as $k \to \infty$, there is some $k_j = k_j(\varepsilon_{j,\delta}) = k_j(\delta) \in \mathbb{N}$ such that $\|a_{k,j} - a_j\|_{\ell^2} \leq \varepsilon_{j,\delta}/(4dB)$ and $|(b_1^{n_k})_j - (b_1)_j| \leq \varepsilon_{j,\delta}/4$ for all $k \geq k_j$. Therefore,

$$\begin{aligned}|\langle a_{k,j}, x\rangle + (b_1^{n_k})_j| &\geq |\langle a_j, x\rangle + (b_1)_j| - |\langle a_j - a_{k,j}, x\rangle + (b_1)_j - (b_1^{n_k})_j| \\ &\geq \varepsilon_{j,\delta} - \|a_j - a_{k,j}\|_{\ell^2} \cdot \|x\|_{\ell^2} - |(b_1)_j - (b_1^{n_k})_j| \\ &\geq \varepsilon_{j,\delta} - \frac{\varepsilon_{j,\delta}}{4dB} \cdot dB - \frac{\varepsilon_{j,\delta}}{4} = \frac{\varepsilon_{j,\delta}}{2} > 0 \quad \text{for all } x \in \Omega_\delta \text{ and } k \geq k_j.\end{aligned}$$

With the same argument as at the end of Case 1, we thus see $\operatorname{sign}(\langle a_{k,j}, x\rangle + (b_1^{n_k})_j) = \operatorname{sign}(b_1^{n_k})_j$ for all $x \in \Omega_\delta$ and $k \geq k_j(\delta)$.

Together, the two cases prove that Eq. (D.17) holds for $N_2(\delta) := \max_{j \in J} k_j(\delta)$.

Step 5 (Showing that the j-th neuron is affine-linear on $U^{(\varepsilon,+)}_{\alpha^{(i)},\beta^{(i)}}$ **and on** *$U^{(\varepsilon,-)}_{\alpha^{(i)},\beta^{(i)}}$* **for** *$j \in J_i$)*: In the following, we write $U^{(\varepsilon,\pm)}_{\alpha^{(i)},\beta^{(i)}}$ for one of the two sets $U^{(\varepsilon,+)}_{\alpha^{(i)},\beta^{(i)}}$ or $U^{(\varepsilon,-)}_{\alpha^{(i)},\beta^{(i)}}$. We claim that for each $\varepsilon > 0$, there is some $N_4(\varepsilon) \in \mathbb{N}$ such that:

$$\begin{aligned}&\text{If } i \in \underline{r},\ j \in J_i,\ k \geq N_4(\varepsilon), \\ &\text{then } v_j^{(k)} := \varrho_a\big(\langle a_{k,j}, \cdot\rangle + (b_1^{n_k})_j\big) \text{ is affine-linear on } \Omega \cap U^{(\varepsilon,\pm)}_{\alpha^{(i)},\beta^{(i)}}.\end{aligned}$$

To see this, let $\varepsilon > 0$ be arbitrary, and recall $J^c = \bigcup_{i=1}^r J_i$. By definition of J_i, and by choice of $\alpha^{(i)}$ and $\beta^{(i)}$, there is for each $i \in \underline{r}$ and $j \in J_i$ some $\sigma_j \in \{\pm 1\}$ satisfying

$$\big(a_{k,j}, (b_1^{n_k})_j\big) \xrightarrow[k\to\infty]{} \big(a_j, (b_1)_j\big) = \sigma_j \cdot \big(\alpha^{(i)}, \beta^{(i)}\big).$$

Thus, there is some $k^{(j)}(\varepsilon) \in \mathbb{N}$ such that $\|a_{k,j} - \sigma_j\, \alpha^{(i)}\|_{\ell^2} \leq \varepsilon/(4dB)$ and $|(b_1^{n_k})_j - \sigma_j\, \beta^{(i)}| \leq \varepsilon/4$ for all $k \geq k^{(j)}(\varepsilon)$.

Define $N_4(\varepsilon) := \max_{j \in J^c} k^{(j)}(\varepsilon)$. Then, for $k \geq N_4(\varepsilon)$, $i \in \underline{r}$, $j \in J_i$, and arbitrary $x \in \Omega \cap U^{(\varepsilon,\pm)}_{\alpha^{(i)},\beta^{(i)}}$, we have on the one hand $|\sigma_j \cdot (\langle \alpha^{(i)}, x\rangle + \beta^{(i)})| \geq \varepsilon$, and on the other hand

$$\begin{aligned}&\big|\big(\langle a_{k,j}, x\rangle + (b_1^{n_k})_j\big) - \sigma_j \cdot \big(\langle \alpha^{(i)}, x\rangle + \beta^{(i)}\big)\big| \leq dB \cdot \|a_{k,j} \\ &\quad - \sigma_j\, \alpha^{(i)}\|_{\ell^2} + |(b_1^{n_k})_j - \sigma_j\, \beta^{(i)}| \leq \varepsilon/2,\end{aligned}$$

since $\|x\|_{\ell^2} \leq \sqrt{d} \cdot B \leq dB$. In combination, this shows $|\langle a_{k,j}, x\rangle + (b_1^{n_k})_j| \geq \varepsilon/2 > 0$ for all $x \in \Omega \cap U^{(\varepsilon,\pm)}_{\alpha^{(i)},\beta^{(i)}}$. But since $\Omega \cap U^{(\varepsilon,\pm)}_{\alpha^{(i)},\beta^{(i)}}$ is connected (in fact, convex), and since the function $x \mapsto \langle a_{k,j}, x\rangle + (b_1^{n_k})_j$ is continuous, it must have a constant sign on $\Omega \cap U^{(\varepsilon,\pm)}_{\alpha^{(i)},\beta^{(i)}}$. This easily implies that $v_j^{(k)} = \varrho_a\big(\langle a_{k,j}, \cdot\rangle + (b_1^{n_k})_j\big)$ is indeed affine-linear on $\Omega \cap U^{(\varepsilon,\pm)}_{\alpha^{(i)},\beta^{(i)}}$ for $k \geq N_4(\varepsilon)$.

Step 6 (Proving the "almost convergence" of the sum of all j-th neurons for $j \in J_i$): For $i \in \underline{r}$ define

$$g_i^{(k)} : \mathbb{R}^d \to \mathbb{R}, x \mapsto \sum_{j \in J_i} (A_2^{n_k})_{1,j} \; \varrho_a\big(\langle a_{k,j}, x\rangle + (b_1^{n_k})_j\big) = \sum_{j \in J_i} (A_2^{n_k})_{1,j} \; \nu_j^{(k)}(x).$$

In combination with Eq. (D.18), we see

$$f_{n_k}(x) = \mathrm{R}_{\varrho_a}^{\Omega}(\widetilde{\Phi}_{n_k})(x) = \sum_{\ell=1}^{r+1} g_\ell^{(k)}(x) \qquad \forall\, x \in \Omega_\delta \text{ and } k \geq N_3(\delta), \tag{D.19}$$

with $g_{r+1}^{(k)} : \mathbb{R}^d \to \mathbb{R}$ being affine-linear.

Recall from Step 4 that $\Omega_{\delta_0}^\circ \cap S_{\alpha^{(i)},\beta^{(i)}} \neq \varnothing$ for all $i \in \underline{r}$, by choice of δ_0. Therefore, Lemma D.4 shows (because of $U_{\alpha,\beta}^{(\sigma)} \subset (U_{\alpha,\beta}^{(\varepsilon)})^\circ$ for $\varepsilon < \sigma$) for each $i \in \underline{r}$ that

$$K_i := \Omega_{\delta_0}^\circ \cap S_{\alpha^{(i)},\beta^{(i)}} \cap \bigcap_{\ell \in \underline{r} \setminus \{i\}} \big(U_{\alpha^{(\ell)},\beta^{(\ell)}}^{(\varepsilon_i)}\big)^\circ \neq \varnothing \qquad \text{for a suitable} \quad \varepsilon_i > 0.$$

Let us fix some $x_i \in K_i$ and some $r_i > 0$ such that $\overline{B}_{r_i}(x_i) \subset \Omega_{\delta_0}^\circ \cap \bigcap_{\ell \in \underline{r} \setminus \{i\}} \big(U_{\alpha^{(\ell)},\beta^{(\ell)}}^{(\varepsilon_i)}\big)^\circ$; this is possible, since the set on the right-hand side contains x_i and is open. Now, since $\overline{B}_{r_i}(x_i)$ is connected, we see for each $\ell \in \underline{r} \setminus \{i\}$ that either $\overline{B}_{r_i}(x_i) \subset U_{\alpha^{(\ell)},\beta^{(\ell)}}^{(\varepsilon_i,+)}$ or $\overline{B}_{r_i}(x_i) \subset U_{\alpha^{(\ell)},\beta^{(\ell)}}^{(\varepsilon_i,-)}$. Therefore, as a consequence of the preceding step, we see that there is some $N_5^{(i)} \in \mathbb{N}$ such that $g_\ell^{(k)}$ is affine-linear on $\overline{B}_{r_i}(x_i)$ for all $\ell \in \underline{r} \setminus \{i\}$ and all $k \geq N_5^{(i)}$.

Thus, setting $N_5 := \max\{N_3(\delta_0), \max_{i=1,\dots,r} N_5^{(i)}\}$, we see as a consequence of Eq. (D.19) and because of $\overline{B}_{r_i}(x_i) \subset \Omega_{\delta_0}^\circ$ that for each $i \in \underline{r}$ and any $k \geq N_5$, there is an affine-linear map $q_i^{(k)} : \mathbb{R}^d \to \mathbb{R}$ satisfying

$$f_{n_k}(x) = \sum_{\ell=1}^{k+1} g_\ell^{(k)}(x) = g_i^{(k)}(x) + q_i^{(k)}(x) \quad \text{for all } x \in \overline{B}_{r_i}(x_i) \text{ and } k \geq N_5. \tag{D.20}$$

Next, note that Step 5 implies for arbitrary $\varepsilon > 0$ that for all k large enough (depending on ε), $g_i^{(k)}$ is affine-linear on $B_{r_i}(x_i) \cap U_{\alpha^{(i)},\beta^{(i)}}^{(\varepsilon,\pm)}$. Since $f(x) = \lim_k f_{n_k}(x) = \lim_k g_i^{(k)}(x) + q_i^{(k)}(x)$, we thus see that f is affine-linear on $B_{r_i}(x_i) \cap U_{\alpha^{(i)},\beta^{(i)}}^{(\varepsilon,\pm)}$ for *arbitrary* $\varepsilon > 0$. Therefore, f is affine-linear on $B_{r_i}(x_i) \cap W_{\alpha^{(i)},\beta^{(i)}}^{\pm}$ and continuous on $\Omega \supset B_{r_i}(x_i)$, and we have $x_i \in B_{r_i}(x_i) \cap S_{\alpha^{(i)},\beta^{(i)}} \neq \varnothing$. Thus, Lemma D.8 shows that there are $c_i \in \mathbb{R}$, $\zeta_i \in \mathbb{R}^d$, and $\kappa_i \in \mathbb{R}$ such that

$$\begin{aligned} f(x) \;=\; G_i(x) \quad \forall\, x \in B_{r_i}(x_i), \quad \text{with} \quad G_i : \mathbb{R}^d \to \mathbb{R}, x \\ \mapsto c_i \cdot \varrho_a\big(\langle \alpha^{(i)}, x\rangle + \beta^{(i)}\big) + \langle \zeta_i, x\rangle + \kappa_i. \end{aligned} \tag{D.21}$$

We now intend to make use of the following elementary fact: If $(\psi_k)_{k\in\mathbb{N}}$ is a sequence of maps $\psi_k : \mathbb{R}^d \to \mathbb{R}$, if $\Theta \subset \mathbb{R}^d$ is such that each ψ_k is affine-linear on Θ,

and if $U \subset \Theta$ is a nonempty *open* subset such that $\psi(x) := \lim_{k\to\infty} \psi_k(x) \in \mathbb{R}$ exists for all $x \in U$, then ψ can be uniquely extended to an affine-linear map $\psi : \mathbb{R}^d \to \mathbb{R}$, and we have $\psi_k(x) \to \psi(x)$ for all $x \in \Theta$, even with locally uniform convergence. Essentially, what is used here is that the vector space of affine-linear maps $\mathbb{R}^d \to \mathbb{R}$ is finite-dimensional, so that the (Hausdorff) topology of pointwise convergence on U coincides with that of locally uniform convergence on Θ; see [61, Theorem 1.21].

To use this observation, note that Eq.s (D.20) and (D.21) show that $g_i^{(k)} + q_i^{(k)}$ converges pointwise to G_i on $B_{r_i}(x_i)$. Furthermore, since $x_i \in S_{\alpha^{(i)},\beta^{(i)}}$, it is not hard to see that there is some $\varepsilon_0 > 0$ with $\left(U^{(\varepsilon,\pm)}_{\alpha^{(i)},\beta^{(i)}}\right)^\circ \cap B_{r_i}(x_i) \neq \varnothing$ for all $\varepsilon \in (0, \varepsilon_0)$; for the details, we refer to Step 1 in the proof of Lemma D.8. Finally, as a consequence of Step 5, we see for arbitrary $\varepsilon \in (0, \varepsilon_0)$ that $g_i^{(k)} + q_i^{(k)}$ and G_i are both affine-linear on $U^{(\varepsilon,\pm)}_{\alpha^{(i)},\beta^{(i)}}$, at least for k large enough (depending on ε). Thus, the observation from above (with $\Theta = U^{(\varepsilon,\pm)}_{\alpha^{(i)},\beta^{(i)}}$ and $U = \Theta^\circ \cap B_{r_i}(x_i)$) implies that $g_i^{(k)} + q_i^{(k)} \to G_i$ pointwise on $U^{(\varepsilon,\pm)}_{\alpha^{(i)},\beta^{(i)}}$, for *arbitrary* $\varepsilon \in (0, \varepsilon_0)$.

Because of $\bigcup_{\sigma\in\{\pm\}} \bigcup_{0<\varepsilon<\varepsilon_0} U^{(\varepsilon,\sigma)}_{\alpha^{(i)},\beta^{(i)}} = \mathbb{R}^d \setminus S_{\alpha^{(i)},\beta^{(i)}}$, this implies

$$g_i^{(k)} + q_i^{(k)} \xrightarrow[k\to\infty]{} G_i \quad \text{pointwise on} \quad \mathbb{R}^d \setminus S_{\alpha^{(i)},\beta^{(i)}} \quad \text{for any} \quad i \in \underline{r}. \tag{D.22}$$

Step 7 (Finishing the proof): For arbitrary $\delta \in (0, \delta_0)$, let us set

$$\Lambda_\delta := \Omega_\delta^\circ \setminus \bigcup_{i=1}^{r} S_{\alpha^{(i)},\beta^{(i)}}.$$

Then, Eqs. (D.19) and (D.22) imply for $k \geq N_3(\delta)$ that

$$\begin{aligned} g_{r+1}^{(k)} - \sum_{i=1}^{r} q_i^{(k)} &= \sum_{i=1}^{r+1} g_i^{(k)} - \left(\sum_{i=1}^{r} g_i^{(k)} + q_i^{(k)}\right) \\ &= f_{n_k} - \left(\sum_{i=1}^{r} g_i^{(k)} + q_i^{(k)}\right) \xrightarrow[k\to\infty]{\text{pointwise on } \Lambda_\delta} f - \sum_{i=1}^{r} G_i. \end{aligned}$$

But since $g_{r+1}^{(k)}$ and all $q_i^{(k)}$ are affine-linear, and since Λ_δ is an open set of positive measure, this implies that there is an affine-linear map $\psi : \mathbb{R}^d \to \mathbb{R}, x \mapsto \langle \zeta, x\rangle + \kappa$ satisfying $f - \sum_{i=1}^{r} G_i = \psi$ on Λ_δ, for arbitrary $\delta \in (0, \delta_0)$. Note that ψ is independent of the choice of δ, and thus

$$f = \psi + \sum_{i=1}^{r} G_i \quad \text{on} \quad \bigcup_{0<\delta<\delta_0} \Lambda_\delta = \Omega^\circ \setminus \bigcup_{i=1}^{r} S_{\alpha^{(i)},\beta^{(i)}}.$$

But the latter set is dense in Ω (since its complement is a null-set), and f and $\psi + \sum_{i=1}^{r} G_i$ are continuous on Ω. Hence,

$$f(x) = \psi(x) + \sum_{i=1}^{r} G_i(x) = \Big(\kappa + \sum_{i=1}^{r} \kappa_i\Big) + \Big\langle \zeta + \sum_{i=1}^{r} \zeta_i, x \Big\rangle + \sum_{i=1}^{r} c_i \cdot \varrho_a\big(\langle \alpha^{(i)}, x\rangle + \beta^{(i)}\big) \quad \forall x \in \Omega.$$

Recalling from Steps 3 and 4 that $r < N_0$, this implies $f \in \mathcal{RNN}^{\Omega}_{\varrho_a}((d, r+1, 1)) \subset \mathcal{RNN}^{\Omega}_{\varrho_a}((d, N_0, 1))$, as claimed. Here, we implicitly used that

$$\langle \alpha, x\rangle + \beta = \varrho_a\big(\langle \alpha, x\rangle + dB\,\|\alpha\|_{\ell^2}\big) + \beta - dB\,\|\alpha\|_{\ell^2} \quad \text{for all } x \in \Omega, \text{ arbitrary } \alpha \in \mathbb{R}^d,\ \beta \in \mathbb{R},$$

since $\langle \alpha, x\rangle + dB\,\|\alpha\|_{\ell^2} \geq 0$ for $x \in \Omega = [-B, B]^d$, so that $\varrho_a(\langle \alpha, x\rangle + dB\|\alpha\|_{\ell^2}) = \langle \alpha, x\rangle + dB\|\alpha\|_{\ell^2}$. □

Appendix E: Proofs of the Results in Sect. 4

E.1. Proof of Proposition 4.1

Step 1: We first show that if $(f_n)_{n\in\mathbb{N}}$ and $(g_n)_{n\in\mathbb{N}}$ are sequences of continuous functions $f_n : \mathbb{R}^d \to \mathbb{R}^N$ and $g_n : \mathbb{R}^N \to \mathbb{R}^D$ that satisfy $f_n \to f$ and $g_n \to g$ with locally uniform convergence, then also $g_n \circ f_n \to g \circ f$ locally uniformly.

To see this, let $R, \varepsilon > 0$ be arbitrary. On $\overline{B_R}(0) \subset \mathbb{R}^d$, we then have $f_n \to f$ uniformly. In particular, $C := \sup_{n\in\mathbb{N}} \sup_{|x|\leq R} |f_n(x)| < \infty$; here, we implicitly used that f and all f_n are continuous, and hence bounded on $\overline{B_R}(0)$. But on $\overline{B_C}(0) \subset \mathbb{R}^N$, we have $g_n \to g$ uniformly, so that there is some $n_1 \in \mathbb{N}$ with $|g_n(y) - g(y)| < \varepsilon$ for all $n \geq n_1$ and all $y \in \mathbb{R}^N$ with $|y| \leq C$. Furthermore, g is uniformly continuous on $\overline{B_C}(0)$, so that there is some $\delta > 0$ with $|g(y) - g(z)| < \varepsilon$ for all $y, z \in \overline{B_C}(0)$ with $|y - z| \leq \delta$. Finally, by the uniform convergence of $f_n \to f$ on $\overline{B_R}(0)$, we get some $n_2 \in \mathbb{N}$ with $|f_n(x) - f(x)| \leq \delta$ for all $n \geq n_2$ and all $x \in \mathbb{R}^d$ with $|x| \leq R$.

Overall, these considerations show for $n \geq \max\{n_1, n_2\}$ and $x \in \mathbb{R}^d$ with $|x| \leq R$ that

$$|g_n(f_n(x)) - g(f(x))| \leq |g_n(f_n(x)) - g(f_n(x))| + |g(f_n(x)) - g(f(x))| \leq \varepsilon + \varepsilon.$$

Step 2 We show that $\mathrm{R}^{\Omega}_{\varrho}$ is continuous. Assume that some neural network sequence $(\Phi_n)_{n\in\mathbb{N}} \subset \mathcal{NN}((d, N_1, \ldots, N_L))$ given by $\Phi_n = \big((A_1^{(n)}, b_1^{(n)}), \ldots, (A_L^{(n)}, b_L^{(n)})\big)$ fulfills $\Phi_n \to \Phi = \big((A_1, b_1), \ldots, (A_L, b_L)\big) \in \mathcal{NN}((d, N_1, \ldots, N_L))$. For $\ell \in$

$\{1, \dots, L-1\}$ set

$$\alpha_\ell^{(n)} : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell},\ x \mapsto \varrho_\ell(A_\ell^{(n)} x + b_\ell^{(n)}),$$
$$\alpha_\ell : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell},\ x \mapsto \varrho_\ell(A_\ell x + b_\ell),$$

where $\varrho_\ell := \varrho \times \cdots \times \varrho$ denotes the N_ℓ-fold Cartesian product of ϱ. Likewise, set

$$\alpha_L^{(n)} : \mathbb{R}^{N_{L-1}} \to \mathbb{R}^{N_L},\ x \mapsto A_L^{(n)} x + b_L^{(n)} \quad \text{and} \quad \alpha_L : \mathbb{R}^{N_{L-1}} \to \mathbb{R}^{N_L},\ x \mapsto A_L x + b_L.$$

By what was shown in Step 1, it is not hard to see for every $\ell \in \{1, \dots, L\}$ that $\alpha_\ell^{(n)} \to \alpha_\ell$ locally uniformly as $n \to \infty$. By another (inductive) application of Step 1, this shows

$$\mathrm{R}_\varrho^\Omega(\Phi_n) = \alpha_L^{(n)} \circ \cdots \circ \alpha_1^{(n)} \to \alpha_L \circ \cdots \circ \alpha_1 = \mathrm{R}_\varrho^\Omega(\Phi)$$

with locally uniform convergence. Since Ω is compact, this implies uniform convergence on Ω, and thus completes the proof of the first claim.

Step 3 Let $\varrho_\ell := \varrho \times \cdots \times \varrho$ be the N_ℓ-fold Cartesian product of ϱ in case of $\ell \in \{1, \dots, L-1\}$, and set $\varrho_L := \mathrm{id}_{\mathbb{R}^{N_L}}$. For arbitrary $x \in \Omega$ and $\Phi = \big((A_1, b_1), \dots, (A_L, b_L)\big) \in \mathcal{NN}(S)$, define inductively $\alpha_x^{(0)}(\Phi) := x \in \mathbb{R}^d = \mathbb{R}^{N_0}$, and

$$\alpha_x^{(\ell+1)}(\Phi) := \varrho_{\ell+1}\big(A_{\ell+1}\, \alpha_x^{(\ell)}(\Phi) + b_{\ell+1}\big) \in \mathbb{R}^{N_{\ell+1}} \quad \text{for} \quad \ell \in \{0, \dots, L-1\}.$$

Let $R > 0$ be fixed, but arbitrary. We will prove by induction on $\ell \in \{0, \dots, L\}$ that

$$\|\alpha_x^{(\ell)}(\Phi)\|_{\ell^\infty} \le C_{\ell,R} \quad \text{and} \quad \|\alpha_x^{(\ell)}(\Phi) - \alpha_x^{(\ell)}(\Psi)\|_{\ell^\infty} \le M_{\ell,R} \cdot \|\Phi - \Psi\|_{\text{total}}$$

for suitable $C_{\ell,R}, M_{\ell,R} > 0$, arbitrary $x \in \Omega$ and $\Phi, \Psi \in \mathcal{NN}(S)$ with $\|\Phi\|_{\text{total}}, \|\Psi\|_{\text{total}} \le R$.

This will imply that $\mathrm{R}_\varrho^\Omega$ is locally Lipschitz, since clearly $\mathrm{R}_\varrho^\Omega(\Phi)(x) = \alpha_x^{(L)}(\Phi)$, and hence

$$\|\mathrm{R}_\varrho^\Omega(\Phi) - \mathrm{R}_\varrho^\Omega(\Psi)\|_{\sup} = \sup_{x \in \Omega} |\alpha_x^{(L)}(\Phi) - \alpha_x^{(L)}(\Psi)| \le M_{L,R} \cdot \|\Phi - \Psi\|_{\text{total}}.$$

The case $\ell = 0$ is trivial: On the one hand, $|\alpha_x^{(0)}(\Phi) - \alpha_x^{(0)}(\Psi)| = 0 \le \|\Phi - \Psi\|_{\text{total}}$. On the other hand, since Ω is bounded, we have $|\alpha_x^{(0)}(\Phi)| = |x| \le C_0$ for a suitable constant $C_0 = C_0(\Omega)$.

For the induction step, let us write $\Psi = \big((B_1, c_1), \dots, (B_L, c_L)\big)$, and note that

$$\begin{aligned} \|A_{\ell+1}\, \alpha_x^{(\ell)}(\Phi) + b_{\ell+1}\|_{\ell^\infty} &\le N_\ell \|A_{\ell+1}\|_{\max} \cdot \|\alpha_x^{(\ell)}(\Phi)\|_{\ell^\infty} + \|b_{\ell+1}\|_{\ell^\infty} \\ &\le (1 + N_\ell C_{\ell,R}) \cdot \|\Phi\|_{\text{total}} =: K_{\ell+1,R}. \end{aligned}$$

Clearly, the same estimate holds with $A_{\ell+1}, b_{\ell+1}$ and Φ replaced by $B_{\ell+1}, c_{\ell+1}$ and Ψ, respectively. Next, observe that with ϱ also $\varrho_{\ell+1}$ is locally Lipschitz. Thus, there is $\Gamma_{\ell+1,R} > 0$ with

$$\|\varrho_{\ell+1}(x) - \varrho_{\ell+1}(y)\|_{\ell^\infty} \leq \Gamma_{\ell+1,R} \cdot \|x - y\|_{\ell^\infty}.$$

for all $x, y \in \mathbb{R}^{N_{\ell+1}}$ with $\|x\|_{\ell^\infty}, \|y\|_{\ell^\infty} \leq K_{\ell+1,R}$. On the one hand, this implies

$$\begin{aligned}\|\alpha_x^{(\ell+1)}(\Phi)\|_{\ell^\infty} &\leq \left\|\varrho_{\ell+1}\left(A_{\ell+1}\,\alpha_x^{(\ell)}(\Phi) + b_{\ell+1}\right) - \varrho_{\ell+1}(0)\right\|_{\ell^\infty} + \|\varrho_{\ell+1}(0)\|_{\ell^\infty} \\ &\leq \Gamma_{\ell+1,R}\,\|A_{\ell+1}\,\alpha_x^{(\ell)}(\Phi) + b_{\ell+1}\|_{\ell^\infty} + \|\varrho_{\ell+1}(0)\|_{\ell^\infty} \\ &\leq \Gamma_{\ell+1,R}\,K_{\ell+1,R} + \|\varrho_{\ell+1}(0)\|_{\ell^\infty} =: C_{\ell+1,R}.\end{aligned}$$

On the other hand, we also get

$$\begin{aligned}&\|\alpha_x^{(\ell+1)}(\Phi) - \alpha_x^{(\ell+1)}(\Psi)\|_{\ell^\infty} \\ &\quad= \|\varrho_{\ell+1}(A_{\ell+1}\alpha_x^{(\ell)}(\Phi) + b_{\ell+1}) - \varrho_{\ell+1}(B_{\ell+1}\alpha_x^{(\ell)}(\Psi) + c_{\ell+1})\|_{\ell^\infty} \\ &\quad\leq \Gamma_{\ell+1,R} \cdot \|(A_{\ell+1}\alpha_x^{(\ell)}(\Phi) + b_{\ell+1}) - (B_{\ell+1}\alpha_x^{(\ell)}(\Psi) + c_{\ell+1})\|_{\ell^\infty} \\ &\quad\leq \Gamma_{\ell+1,R} \cdot \Big(\|(A_{\ell+1} - B_{\ell+1})\alpha_x^{(\ell)}(\Phi)\|_{\ell^\infty} + \|B_{\ell+1}(\alpha_x^{(\ell)}(\Phi) - \alpha_x^{(\ell)}(\Psi))\|_{\ell^\infty} \\ &\qquad+ \|b_{\ell+1} - c_{\ell+1}\|_{\ell^\infty}\Big) \\ &\quad\leq \Gamma_{\ell+1,R} \cdot \Big(\|\Phi - \Psi\|_{\text{total}} \cdot (N_\ell \|\alpha_x^{(\ell)}(\Phi)\|_{\ell^\infty} + 1) \\ &\qquad+ N_\ell \cdot \|\Psi\|_{\text{total}} \cdot \|\alpha_x^{(\ell)}(\Phi) - \alpha_x^{(\ell)}(\Psi)\|_{\ell^\infty}\Big) \\ &\quad\leq \Gamma_{\ell+1,R} \cdot (N_\ell C_{\ell,R} + R N_\ell M_{\ell,R} + 1) \cdot \|\Phi - \Psi\|_{\text{total}} =: M_{\ell+1,R} \cdot \|\Phi - \Psi\|_{\text{total}}.\end{aligned}$$

Step 4 Let ϱ be Lipschitz with Lipschitz constant M, where we assume without loss of generality that $M \geq 1$. With the functions ϱ_ℓ from the preceding step, it is not hard to see that each ϱ_ℓ is M-Lipschitz, where we use the $\|\cdot\|_{\ell^\infty}$-norm on $\mathbb{R}^{N_\ell}$.

Let $\Phi = \big((A_1, b_1), \dots, (A_L, b_L)\big) \in \mathcal{NN}(S)$, and $\alpha_\ell : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell}, x \mapsto \varrho_\ell(A_\ell\, x + b_\ell)$ for $\ell \in \{1, \dots, L-1\}$. Then, α_ℓ is Lipschitz with $\operatorname{Lip}(\alpha_\ell) \leq M \cdot \|A_\ell\|_{\ell^\infty \to \ell^\infty} \leq M \cdot N_{\ell-1} \cdot \|A\|_{\max} \leq M N_{\ell-1} \cdot \|\Phi\|_{\text{scaling}}$. Thus, we finally see that $\mathrm{R}_\varrho^\Omega(\Phi) = \alpha_L \circ \cdots \circ \alpha_1$ is Lipschitz with Lipschitz constant $M^L \cdot N_0 \cdots N_{L-1} \cdot \|\Phi\|_{\text{scaling}}^L$. This proves the final claim of the proposition when choosing the ℓ^∞-norm on $\mathbb{R}^d$ and $\mathbb{R}^{N_L}$. Of course, choosing another norm than the ℓ^∞-norm can be done, at the cost of possibly enlarging the constant C in the statement of the proposition. □

E.2. Proof of Theorem 4.2

Step 1 For $a > 0$, define

$$f_a : \mathbb{R} \to \mathbb{R},\ x \mapsto \varrho(x + a) - 2\varrho(x) + \varrho(x - a).$$

Our claim in this step is that there is some $a > 0$ with $f_a \not\equiv \text{const}$.

Let us assume toward a contradiction that this fails; that is, $f_a \equiv c_a$ for all $a > 0$. Since ϱ is Lipschitz continuous, it is at most of linear growth, so that ϱ is a tempered distribution. We will now make use of the *Fourier transform*, which we define by $\widehat{f}(\xi) = \int_{\mathbb{R}} f(x)\, e^{-2\pi i x\xi}\, dx$ for $f \in L^1(\mathbb{R})$, as in [26,31], where it is also explained how the Fourier transform is extended to the space of tempered distributions. Elementary properties of the Fourier transform for tempered distributions (see [31, Proposition 2.3.22]) show

$$c_a \cdot \delta_0 = \widehat{f_a} = \widehat{\varrho} \cdot g_a \qquad \text{with} \qquad g_a : \mathbb{R} \to \mathbb{R},\ \xi \mapsto e^{2\pi i a\xi} - 2 + e^{-2\pi i a\xi}.$$

Next, setting $z(\xi) := e^{2\pi i a\xi} \neq 0$, we observe that

$$\begin{aligned} g_a(\xi) &= z(\xi) - 2 + [z(\xi)]^{-1} = [z(\xi)]^{-1} \cdot (z^2(\xi) - 2z(\xi) + 1) \\ &= [z(\xi)]^{-1} \cdot (z(\xi) - 1)^2 \neq 0, \end{aligned}$$

as long as $z(\xi) \neq 1$, that is, as long as $\xi \notin a^{-1}\mathbb{Z}$.

Let $\varphi \in C_c^\infty(\mathbb{R})$ such that $0 \notin \operatorname{supp}\varphi$ be fixed, but arbitrary. This implies $\operatorname{supp}\varphi \subset \mathbb{R} \setminus a^{-1}\mathbb{Z}$ for some sufficiently small $a > 0$. Since g_a vanishes nowhere on the compact set $\operatorname{supp}\varphi$, it is not hard to see that there is some smooth, compactly supported function h with $h \cdot g_a \equiv 1$ on the support of φ. All in all, we thus get

$$\langle \widehat{\varrho}, \varphi \rangle_{\mathcal{S}',\mathcal{S}} = \langle \widehat{\varrho} \cdot g_a, h \cdot \varphi \rangle_{\mathcal{S}',\mathcal{S}} = \langle \widehat{f_a}, h \cdot \varphi \rangle_{\mathcal{S}',\mathcal{S}} = c_a \cdot h(0) \cdot \varphi(0) = 0.$$

Since $\varphi \in C_c^\infty(\mathbb{R})$ with $0 \notin \operatorname{supp}\varphi$ was arbitrary, we have shown $\operatorname{supp}\widehat{\varrho} \subset \{0\}$. But by [31, Corollary 2.4.2], this implies that ϱ is a polynomial. Since the only globally Lipschitz continuous polynomials are affine-linear, ϱ must be affine-linear, contradicting the prerequisites of the theorem.

Step 2 In this step we construct certain continuous functions $F_n : \mathbb{R}^d \to \mathbb{R}$ which satisfy $\operatorname{Lip}(F_n|_\Omega) \to \infty$ and $F_n \to 0$ uniformly on $\mathbb{R}^d$. We will then use these functions in the next step to construct the desired networks Φ_n.

We first note that each function f_a from Step 1 is bounded. In fact, if ϱ is M-Lipschitz, then

$$|f_a(x)| \leq |\varrho(x+a) - \varrho(x)| + |\varrho(x-a) - \varrho(x)| \leq 2M|a|. \tag{E.1}$$

Next, recall that ϱ is Lipschitz continuous and not affine-linear. Therefore, Lemma C.8 shows that there is some $t_0 \in \mathbb{R}$ such that ϱ is differentiable at t_0 with $\varrho'(t_0) \neq 0$. Therefore, Proposition B.3 shows that there is a neural network $\Phi \in \mathcal{NN}((1, \dots, 1))$ with $L - 1$ layers such that $\psi := \mathrm{R}_\varrho^{\mathbb{R}}(\Phi)$ is differentiable at the origin with $\psi(0) = 0$ and $\psi'(0) = 1$. By definition, this means that there is a function $\delta : \mathbb{R} \to \mathbb{R}$ such that $\psi(x) = x + x \cdot \delta(x)$ and $\delta(x) \to 0 = \delta(0)$ as $x \to 0$.

Next, since Ω has nonempty interior, there exist $x_0 \in \mathbb{R}^d$ and $r > 0$ with $x_0 + [-r, r]^d \subset \Omega$. Let us now choose $a > 0$ with $f_a \not\equiv \text{const}$ (the existence of such an

$a > 0$ is implied by the previous step), and define

$$F_n : \mathbb{R}^d \to \mathbb{R},\ x \mapsto \psi\left(n^{-1} \cdot f_a(n^2 \cdot (x - x_0)_1)\right).$$

Since f_a is not constant, there are $b, c \in \mathbb{R}$ with $b < c$ and $f_a(b) \neq f_a(c)$. Because of $\delta(x) \to 0$ as $x \to 0$, we see that there is some $\kappa > 0$ and some $n_1 \in \mathbb{N}$ with

$$|f_a(b) - f_a(c)| - |f_a(b)| \cdot |\delta(f_a(b)/n)| - |f_a(c)| \cdot |\delta(f_a(c)/n)| \geq \kappa > 0 \qquad \text{for all } n \geq n_1.$$

Let us set $x_n := x_0 + n^{-2} \cdot (b, 0, \dots, 0) \in \mathbb{R}^d$ and $y_n := x_0 + n^{-2} \cdot (c, 0, \dots, 0) \in \mathbb{R}^d$, and observe $x_n, y_n \in \Omega$ for $n \in \mathbb{N}$ large enough. We have $|x_n - y_n| = n^{-2} \cdot |b - c|$. Furthermore, using the expansion $\psi(x) = x + x \cdot \delta(x)$, and noting $f_a(n^2(x_n - x_0)_1) = f_a(b)$ as well as $f_a(n^2(y_n - x_0)_1) = f_a(c)$, we get

$$\begin{aligned}
|F_n(x_n) - F_n(y_n)| &= |\psi(f_a(b)/n) - \psi(f_a(c)/n)| \\
&= \left| \frac{f_a(b)}{n} - \frac{f_a(c)}{n} + \frac{f_a(b)}{n} \cdot \delta\left(\frac{f_a(b)}{n}\right) - \frac{f_a(c)}{n} \cdot \delta\left(\frac{f_a(c)}{n}\right) \right| \\
&\geq \frac{1}{n} \cdot \big(|f_a(b) - f_a(c)| \\
&\quad - |f_a(b)| \cdot |\delta(f_a(b)/n)| - |f_a(c)| \cdot |\delta(f_a(c)/n)| \big) \\
&\geq \kappa / n,
\end{aligned}$$

as long as $n \geq n_1$ is so large that $x_n, y_n \in \Omega$. But this implies

$$\operatorname{Lip}(F_n|_\Omega) \geq \frac{|F_n(x_n) - F_n(y_n)|}{|x_n - y_n|} \geq \frac{\kappa/n}{n^{-2} \cdot |b - c|} = n \cdot \frac{\kappa}{|b - c|} \xrightarrow[n \to \infty]{} \infty.$$

It remains to show $F_n \to 0$ uniformly on $\mathbb{R}^d$. Thus, let $\varepsilon > 0$ be arbitrary. By continuity of ψ at 0, there is some $\delta > 0$ with $|\psi(x)| \leq \varepsilon$ for $|x| \leq \delta$. But Eq. (E.1) shows $|n^{-1} \cdot f_a(n^{-2} \cdot (x - x_0)_1)| \leq n^{-1} \cdot 2M|a| \leq \delta$ for all $x \in \mathbb{R}^d$ and all $n \geq n_0$, with $n_0 = n_0(M, a, \delta) \in \mathbb{N}$ suitable. Hence, $|F_n(x)| \leq \varepsilon$ for all $n \geq n_0$ and $x \in \mathbb{R}^d$.

Step 3 In this step, we construct the networks Φ_n. For $n \in \mathbb{N}$ define

$$A_1^{(n)} := n^2 \cdot \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{3 \times d} \quad \text{and} \quad b_1^{(n)} := \begin{pmatrix} -n^2 \cdot (x_0)_1 + a \\ -n^2 \cdot (x_0)_1 \\ -n^2 \cdot (x_0)_1 - a \end{pmatrix} \in \mathbb{R}^3,$$

as well as $A_2^{(n)} := n^{-1} \cdot (1, -2, 1) \in \mathbb{R}^{1 \times 3}$ and $b_2^{(n)} := 0 \in \mathbb{R}^1$. A direct calculation shows

$$\mathrm{R}_\varrho^{\mathbb{R}^d}(\Phi_n^{(0)})(x) = n^{-1} \cdot f_a(n^2 \cdot (x - x_0)_1),$$

for all $x \in \mathbb{R}^d$, where $\Phi_n^{(0)} := \big((A_1^{(n)}, b_1^{(n)}), (A_2^{(n)}, b_2^{(n)})\big)$. Thus, with the concatenation operation introduced in Definition B.2, the network $\Phi_n^{(1)} := \Phi \bullet \Phi_n^{(0)}$ satisfies $\mathrm{R}_\varrho^\Omega(\Phi_n^{(1)}) = F_n|_\Omega$. Furthermore, it is not hard to see that $\Phi_n^{(1)}$ has L layers and has the architecture $(d, 3, 1, \dots, 1)$. From this and because of $N_1 \geq 3$, by Lemma B.1 there is a network Φ_n with architecture $(d, N_1, \dots, N_{L-1}, 1)$ and $\mathrm{R}_\varrho^\Omega(\Phi_n) = F_n|_\Omega$. By Step 2, this implies $\mathrm{R}_\varrho^\Omega(\Phi_n) = F_n|_\Omega \to 0$ uniformly on Ω, as well as $\mathrm{Lip}(\mathrm{R}_\varrho^\Omega(\Phi_n)) \to \infty$ as $n \to \infty$.

Step 4 In this step, we establish the final property which is stated in the theorem. For this, let us assume toward a contradiction that there is a family of networks $(\Psi_n)_{n\in\mathbb{N}}$ with architecture S and $\mathrm{R}_\varrho^\Omega(\Psi_n) = \mathrm{R}_\varrho^\Omega(\Phi_n)$, some $C > 0$, and a subsequence $(\Psi_{n_r})_{r\in\mathbb{N}}$ with $\|\Psi_{n_r}\|_{\mathrm{scaling}} \leq C$ for all $r \in \mathbb{N}$. In view of the last part of Proposition 4.1, there is a constant $C' = C'(\varrho, S) > 0$ with

$$\mathrm{Lip}\big(\mathrm{R}_\varrho^\Omega(\Phi_{n_r})\big) = \mathrm{Lip}\big(\mathrm{R}_\varrho^\Omega(\Psi_{n_r})\big) \leq C' \cdot \|\Psi_{n_r}\|_{\mathrm{scaling}}^L \leq C' \cdot C^L,$$

in contradiction to $\mathrm{Lip}\big(\mathrm{R}_\varrho^\Omega(\Phi_n)\big) \to \infty$. □

E.3. Proof of Corollary 4.3

Let us denote the range of the realization map by R. By definition (see [44, p. 65]), $\mathrm{R}_\varrho^\Omega$ is a quotient map if and only if

$$\forall\, M \subset R: \qquad M \subset R \text{ open} \iff \left(\mathrm{R}_\varrho^\Omega\right)^{-1}(M) \subset \mathcal{NN}(S) \text{ open}.$$

Clearly, by switching to complements, we can equivalently replace "open" by "closed" everywhere.

Now, choose a sequence of neural networks $(\Phi_n)_{n\in\mathbb{N}}$ as in Theorem 4.2, and set $F_n := \mathrm{R}_\varrho^{\mathbb{R}^d}(\Phi_n)$. Since $\mathrm{Lip}(F_n|_\Omega) \to \infty$, we have $F_n|_\Omega \not\equiv 0$ for all $n \geq n_0$ with $n_0 \in \mathbb{N}$ suitable. Define $M := \{F_n|_\Omega : n \geq n_0\} \subset R$. Note that $M \subset R \subset C(\Omega)$ is *not* closed, since $F_n|_\Omega \to 0$ uniformly, but $0 \in R \setminus M$. Hence, once we show that $\left(\mathrm{R}_\varrho^\Omega\right)^{-1}(M)$ is closed, we will have shown that $\mathrm{R}_\varrho^\Omega$ is not a quotient map.

Thus, let $(\Psi_n)_{n\in\mathbb{N}}$ be a sequence in $\left(\mathrm{R}_\varrho^\Omega\right)^{-1}(M)$ and assume $\Psi_n \to \Psi$ as $n \to \infty$. In particular, $\|\Psi_n\|_{\mathrm{scaling}} \leq C$ for some $C > 0$ and all $n \in \mathbb{N}$. We want to show $\Psi \in \left(\mathrm{R}_\varrho^\Omega\right)^{-1}(M)$ as well. Since $\Psi_n \in \left(\mathrm{R}_\varrho^\Omega\right)^{-1}(M)$, there is for each $n \in \mathbb{N}$ some $r_n \in \mathbb{N}$ with $\mathrm{R}_\varrho^\Omega(\Psi_n) = F_{r_n}|_\Omega$. Now there are two cases:
Case 1: The family $(r_n)_{n\in\mathbb{N}}$ is infinite. But in view of Proposition 4.1, we have

$$\mathrm{Lip}(F_{r_n}|_\Omega) = \mathrm{Lip}(\mathrm{R}_\varrho^\Omega(\Psi_n)) \leq C' \cdot \|\Psi_n\|_{\mathrm{scaling}}^L \leq C' \cdot C^L$$

for a suitable constant $C' = C'(\varrho, S)$, in contradiction to the fact that $\mathrm{Lip}(F_{r_n}|_\Omega) \to \infty$ as $r_n \to \infty$. Thus, this case cannot occur.

Case 2: The family $(r_n)_{n\in\mathbb{N}}$ is finite. Thus, there is some $N \in \mathbb{N}$ with $r_n = N$ for infinitely many $n \in \mathbb{N}$, that is, $\mathrm{R}_\varrho^\Omega(\Psi_n) = F_{r_n}|_\Omega = F_N|_\Omega$ for infinitely many $n \in \mathbb{N}$. But since $\mathrm{R}_\varrho^\Omega(\Psi_n) \to \mathrm{R}_\varrho^\Omega(\Psi)$ as $n \to \infty$ (by the continuity of the realization map), this implies $\mathrm{R}_\varrho^\Omega(\Psi) = F_N|_\Omega \in M$, as desired. □

References

1. Z. Allen-Zhu, Y. Li, and Z. Song, A Convergence Theory for Deep Learning via Over-Parameterization, Proceedings of the 36th International Conference on Machine Learning, 2019, pp. 242–252.
2. H. Amann and J. Escher, Analysis III, Birkhäuser Verlag, Basel, 2009.
3. P. M. Anselone and J. Korevaar, Translation Invariant Subspaces of Finite Dimension, Proc. Amer. Math. Soc. 15 (1964), 747–752.
4. M. Anthony and P. L. Bartlett, Neural Network Learning: Theoretical Foundations, Cambridge University Press, Cambridge, 1999.
5. F. Bach, Breaking the Curse of Dimensionality with Convex Neural Networks, J. Mach. Learn. Res. 18 (2017), no. 1, 629–681.
6. P. Baldi and K. Hornik, Neural Networks and Principal Component Analysis: Learning from Examples Without Local Minima, Neural Netw. 2 (1988), no. 1, 53–58.
7. A.R. Barron, Universal Approximation Bounds for Superpositions of a Sigmoidal Function, IEEE Trans. Inf. Theory 39 (1993), no. 3, 930–945.
8. P. L. Bartlett and S. Ben-David, Hardness Results for Neural Network Approximation Problems, Theor. Comput. Sci. 284 (2002), no. 1, 53–66.
9. P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, Spectrally-normalized margin bounds for neural networks, Adv. Neural Inf. Process. Syst. 30, 2017, pp. 6240–6249.
10. P. L. Bartlett, N. Harvey, C. Liaw, and A. Mehrabian, Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks., J. Mach. Learn. Res. 20 (2019), no. 63, 1–17.
11. P. L. Bartlett and S. Mendelson, Rademacher and Gaussian Complexities: Risk Bounds and Structural Results, J. Mach. Learn. Res. 3 (2002), 463–482.
12. J. Bergstra, G. Desjardins, P. Lamblin, and Y. Bengio, Quadratic Polynomials Learn Better Image Features, Technical Report 1337, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, 2009.
13. A. Blum and R.L. Rivest, Training a 3-node neural network is NP-complete, Adv. Neural Inf. Process. Syst. 2, 1989, pp. 494–501.
14. H. Bölcskei, P. Grohs, G. Kutyniok, and P. C. Petersen, Optimal Approximation with Sparsely Connected Deep Neural Networks, SIAM J. Math. Data Sci. 1 (2019), 8–45.
15. B. Carlile, G. Delamarter, P. Kinney, A. Marti, and B. Whitney, Improving Deep Learn- ing by Inverse Square Root Linear Units (ISRLUs), arXiv preprint arXiv:1710.09967 (2017).
16. L. Chizat and F. Bach, On the Global Convergence of Gradient Descent for Over-parameterized Models using Optimal Transport, Adv. Neural Inf. Process. Syst. 31, 2018, pp. 3036–3046.
17. D. Clevert, T. Unterthiner, and S. Hochreiter, Fast and accurate deep network learn- ing by exponential linear units (elus), 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2–4, 2016, Conference Track Proceedings, 2016.
18. N. Cohen, O. Sharir, and A. Shashua, On the Expressive Power of Deep Learning: A Tensor Analysis, Conference on learning theory, 2016, pp. 698–728.
19. D. L. Cohn, Measure Theory, Birkhäuser Verlag, Basel, 2013.
20. F. Cucker and S. Smale, On the mathematical foundations of learning, Bull. Am. Math. Soc. 39 (2002), 1–49.
21. G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control Signal 2 (1989), no. 4, 303–314.
22. G. E. Dahl, D. Yu, L. Deng, and A. Acero, Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition, IEEE Audio, Speech, Language Process. 20 (2012), no. 1, 30–42.
23. J. Dieudonné, Foundations of Modern Analysis, Pure and Applied Mathematics, Vol. X, Academic Press, New York-London, 1960.

24. W. E, J. Han, and A. Jentzen, Deep learning-based numerical methods for high- dimensional parabolic partial differential equations and backward stochastic differential equations, Commun. Math. Stat. 5 (2017), no. 4, 349–380.
25. W. E and B. Yu, The Deep Ritz Method: A deep learning-based numerical algorithm for solving variational problems, Communications in Mathematics and Statistics 6 (2018), no. 1, 1–12.
26. G. B. Folland, Real Analysis, Pure and Applied Mathematics (New York), Wiley, New York, 1999.
27. C. D. Freeman and J. Bruna, Topology and Geometry of Half-Rectified Network Opti- mization, 5th international conference on learning representations, ICLR 2017, Toulon, France, April 24–26, 2017, conference track proceedings, 2017.
28. F. Girosi and T. Poggio, Networks and the best approximation property, Biol. Cybern. 63 (1990), no. 3, 169–176.
29. X. Glorot, A. Bordes, and Y. Bengio, Deep Sparse Rectifier Neural Networks, Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011, pp. 315–323.
30. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, Cambridge, MA, 2016. http://www.deeplearningbook.org.
31. L. Grafakos, Classical Fourier Analysis, Graduate Texts in Mathematics, vol. 249, Springer, New York, 2008.
32. S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall PTR, Upper Saddle River, 1998.
33. K. He, X. Zhang, S. Ren, and J. Sun, Delving Deep into Rectifiers: Surpassing Human- Level Performance on ImageNet Classification, Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1026–1034.
34. G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups, IEEE Signal Process. Mag. 29 (2012), no. 6, 82–97.
35. K. Hornik, M. Stinchcombe, and H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (1989), no. 5, 359–366.
36. G. Huang, Z. Liu, L. van der Maaten, and K. Q.Weinberger, Densely Connected Convolutional Networks, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017), 2261–2269.
37. A. Jacot, F. Gabriel, and C. Hongler, Neural Tangent Kernel: Convergence and Gen- eralization in Neural Networks, Adv. Neural Inf. Process. Syst. 31, 2018, pp. 8571– 8580.
38. S. Judd, Learning in Networks is Hard, Proceedings of IEEE International Conference on Neural Networks, 1987, pp. 685–692.
39. P. Kainen, V. Kurková, and A. Vogt, Best approximation by Heaviside perceptron networks, Neural Netw. 13 (2000), no. 7, 695–697.
40. P. C. Kainen, V. Kurková, and A. Vogt, Approximation by neural networks is not continuous, Neurocomputing 29 (1999), no. 1-3, 47–56.
41. A. Krizhevsky, I. Sutskever, and G.E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, Adv. Neural Inf. Process. Syst. 25, 2012, pp. 1097–1105.
42. I. E. Lagaris, A. Likas, and D. I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, IEEE Trans. Neural Netw. 9 (1998), no. 5, 987–1000.
43. Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature 521 (2015), no. 7553, 436–444.
44. J. M. Lee, Introduction to Topological Manifolds, Graduate Texts in Mathematics, vol. 202, Springer, New York, 2011.
45. M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, Multilayer feedforward networks with a nonpolynomial activation function can approximate any function, Neural Netw. 6 (1993), no. 6, 861–867.
46. B. Liao, C. Ma, L. Xiao, R. Lu, and L. Ding, An Arctan-Activated WASD Neural Network Approach to the Prediction of Dow Jones Industrial Average, Advances in neural networks - ISNN 2017 - 14th international symposium, ISNN 2017, Sapporo, Hakodate, and Muroran, Hokkaido, Japan, June 21-26, 2017, Proceedings, Part I, 2017, pp. 120–126.
47. A. Maas, Y. Hannun, and A. Ng, Rectifier Nonlinearities Improve Neural Network Acoustic Models, ICML Workshop on Deep Learning for Audio, Speech and Language Processing, 2013.
48. V. E. Maiorov, Best approximation by ridge functions in L_p-spaces, Ukraïn. Mat. Zh. 62 (2010), no. 3, 396–408.
49. W. McCulloch and W. Pitts, A logical calculus of ideas immanent in nervous activity, Bull. Math. Biophys. 5 (1943), 115–133.

50. S. Mei, A. Montanari, and P.-M. Nguyen, A mean field view of the landscape of two-layer neural networks, Proc. Natl. Acad. Sci. USA 115 (2018), no. 33, E7665–E7671.
51. H. N. Mhaskar, Approximation properties of a multilayered feedforward artificial neural network, Adv. Comput. Math. 1 (1993), no. 1, 61–80.
52. H. N. Mhaskar, Neural networks for optimal approximation of smooth and analytic functions, Neural Comput. 8 (1996), no. 1, 164–177.
53. M. Mohri, A. Rostamizadeh, and A. Talwalkar, Foundations of Machine Learning, Adaptive Computation and Machine Learning, MIT Press, Cambridge, MA, 2012.
54. G. Montúfar, R. Pascanu, K. Cho, and Y. Bengio, On the Number of Linear Regions of Deep Neural Networks, Proceedings of the 27th International Conference on Neural Information Processing Systems, 2014, pp. 2924–2932.
55. V. Nair and G. Hinton, Rectified Linear Units Improve Restricted Boltzmann machines, Proceedings of the 27th International Conference on Machine Learning, 2010, pp. 807– 814.
56. Q. Nguyen and M. Hein, The Loss Surface of Deep and Wide Neural Networks, Proceedings of the 34th International Conference on Machine Learning-volume 70, 2017, pp. 2603–2612.
57. P. C. Petersen and F. Voigtlaender, Optimal approximation of piecewise smooth func- tions using deep ReLU neural networks, Neural Netw. 108 (2018), 296–330.
58. PhoemueX (https://math.stackexchange.com/users/151552/phoemuex), Uncountable closed set A, existence of point at which A accumulates "from two sides" of a hyper- plane, 2020. URL:https://math.stackexchange.com/q/3513692 (version: 2020-01-18).
59. G. M. Rotskoff and E. Vanden-Eijnden, Neural Networks as Interacting Particle Sys- tems: Asymptotic Convexity of the Loss Landscape and Universal Scaling of the Ap- proximation Error, arXiv preprint arXiv:1805.00915 (2018).
60. W. Rudin, Real and Complex Analysis, McGraw-Hill Book Co., New York, 1987.
61. W. Rudin, Functional Analysis, International Series in Pure and Applied Mathematics, McGraw-Hill, Inc., New York, 1991.
62. I. Safran and O. Shamir, Depth-Width Tradeoffs in Approximating Natural Functions with Neural Networks, Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 2979–2987.
63. J. Schmidhuber, Deep learning in eural networks: An overview, Neural Netw. 61 (2015), 85–117.
64. D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, Graepel T., T. Lillicrap, K. Simonyan, and D. Hassabis, Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm, arXiv preprint arXiv:1712.01815 (2017).
65. Karen Simonyan and Andrew Zisserman, Very deep convolutional networks for large-scale image recognition, International conference on learning representations, 2015.
66. N. Usunier, G. Synnaeve, Z. Lin, and S. Chintala, Episodic Exploration for Deep Deterministic Policies for StarCraft Micromanagement, 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, 2017.
67. L. Venturi, A. S. Bandeira, and J. Bruna, Neural Networks with Finite Intrinsic Dimension have no Spurious Valleys, arXiv preprint, arXiv:1802.06384 (2018).
68. A. J. Ward, The Structure of Non-Enumerable Sets of Points, J. London Math. Soc. 8 (1933), no. 2, 109–112.
69. C. Wu, P. Karanasou, M. JF. Gales, and K. C. Sim, Stimulated Deep Neural Network for Speech Recognition, University of Cambridge, 2016.
70. G. N. Yannakakis and J. Togelius, Artificial Intelligence and Games, Springer, 2018.
71. D. Yarotsky, Error bounds for approximations with deep ReLU networks, Neural Netw. 94 (2017), 103–114.
72. D. Yarotsky and A. Zhevnerchuk, The phase diagram of approximation rates for deep neural networks, arXiv preprint arXiv:1906.09477 (2019).
73. Y. Zhang, P. Liang, and M. J.Wainwright, Convexified Convolutional Neural Networks, Proceedings of the 34th International Conference on Machine Learning-volume 70, 2017, pp. 4044–4053.

Foundations of Computational Mathematics

Affiliations

Philipp Petersen[1] · Mones Raslan[2] · Felix Voigtlaender[3]

Philipp Petersen
Philipp.Petersen@univie.ac.at

Felix Voigtlaender
felix.voigtlaender@ku.de

Mones Raslan
raslan@math.tu-berlin.de

1 Faculty of Mathematics and Research Platform Data Science @ Uni Vienna, University of Vienna, Oskar Morgenstern Platz 1, 1090 Vienna, Austria

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

3 Department of Scientific Computing, Catholic University of Eichstätt-Ingolstadt, Kollegiengebäude I Bau B, Ostenstraße 26, 85072 Eichstätt, Germany

Neural Networks 134 (2021) 107–130

Contents lists available at ScienceDirect

Neural Networks

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

Approximation rates for neural networks with encodable weights in smoothness spaces

Ingo Gühring [*,1], Mones Raslan [1]

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

ARTICLE INFO

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

Keywords:
Neural networks
Expressivity
Approximation rates
Smoothness spaces
Encodable weights

ABSTRACT

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

1. Introduction

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

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

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

I. General lower bounds based on entropy arguments

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

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

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

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

[1] Both authors contributed equally.

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

https://doi.org/10.1016/j.neunet.2020.11.010
0893-6080/

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

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

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

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

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

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

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

- The *rectified power unit (RePU)* of order $j \in \mathbb{N}_{\geq 2}$, given by ReLU^j, allows for the construction of exact PUs. In this case, for every $k \in \{0,\dots,j\}$, we need at most $C\varepsilon^{-d/(n-k)}$ non-zero weights.
- The *softplus function*, given by $\ln(1+e^x)$, allows for the construction of exponential PUs. In this case, for $k \in \mathbb{N}_0$, and arbitrary $\mu > 0$, we need at most

$$\begin{cases} C\varepsilon^{-d/(n-k)}, & \text{if } k \leq 1, \\ C\varepsilon^{-d/(n-k-\mu)}, & \text{if } k \geq 2, \end{cases}$$

non-zero weights.

- The *inverse square root linear unit*, given by $\mathbb{1}_{[0,\infty)}x + \mathbb{1}_{(-\infty,0)}\frac{x}{\sqrt{1+x^2}}$, allows for the construction of polynomial PUs. In this case, for $k \in \{0,1\}$, we need at most $C\varepsilon^{-d/(n-k)}$ non-zero weights.

  Generally speaking, in the case of polynomial PUs, we are only able to show approximation rates in smoothness norms of a restricted order, depending on the asymptotic behavior of the underlying activation function. We describe the reasons for this issue in more detail in Section 4.3.

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

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

Related work

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

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

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

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

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

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

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

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

Outline

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

2. Neural networks with encodable weights: Terminology

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

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

Encodability

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

To make the notion of encodability more precise, we first introduce coding schemes (see Petersen and Voigtländer (2018)): A *coding scheme (for real numbers)* is a sequence $\mathcal{B} = (B_\ell)_{\ell\in\mathbb{N}}$ of maps $B_\ell : \{0,1\}^\ell \to \mathbb{R}$. Now we define sets of neural networks with weights encodable by a coding scheme. Given an arbitrary coding scheme $\mathcal{B} = (B_\ell)_{\ell\in\mathbb{N}}$, and $d \in \mathbb{N}, \varepsilon, M > 0$, we denote by

$$\mathcal{NN}^{\mathcal{B}}_{M, \lceil C_0 \log_2(1/\varepsilon)\rceil, d} \tag{2.1}$$

the set of all neural networks Φ with d-dimensional input, one-dimensional output and at most M nonzero weights such that *each nonzero weight of Φ is contained in* $\mathrm{Range}(B_{\lceil C_0 \log_2(1/\varepsilon)\rceil})$.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

and

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

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

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

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

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

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

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

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

- The results from Gühring et al. (2020), Yarotsky (2018) combined state:
  For $\mathcal{C} = W^{n,\infty}((0,1)^d)$ and $\mathcal{D} = W^{k,\infty}((0,1)^d)$ with $k = 0, 1$, it holds for the necessary number of nonzero weights M_ε to achieve an ε-approximation in $W^{k,\infty}$ norm that
  $$M_\varepsilon \geq C \varepsilon^{-d/(2n-k)}.$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

4. Upper bounds for general activation functions in Sobolev spaces

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

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

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

(a) Depending on the smoothness j of the activation function, our approximations include $W^{k,p}$ for $k \leq j$ (instead of maximally $W^{1,p}$).
(b) Constructing a PU by neural networks with general activation function is tricky (contrary to ReLU networks) and can, in general, only be done approximately (see Section 4.1 and Fig. 1).
(c) Our polynomial approximations and approximate PUs have depth independent of ε, which results in constant-depth approximations of f.

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

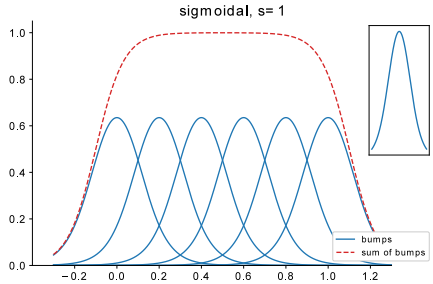
4.1. Ingredient I: (Approximate) partition of unity

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

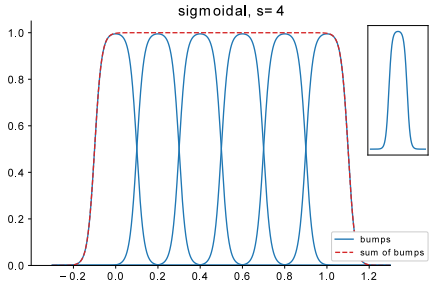
Definition 4.1. Let $d \in \mathbb{N}, j, \tau \in \mathbb{N}_0$. We say that the collection of families of functions $(\Psi^{(j,\tau,N,s)})_{N\in\mathbb{N},s\in\mathbb{R}_{\geq 1}}$, where each $\Psi^{(j,\tau,N,s)} := \{\phi_m^s : m \in \{0,\dots,N\}^d\}$ consists of $(N+1)^d$ functions $\phi_m^s :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

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

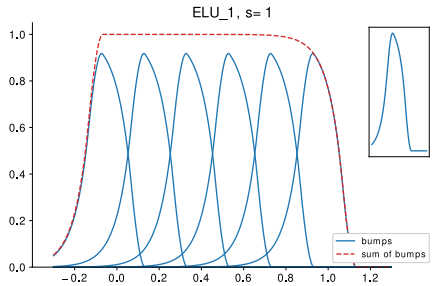
[12] Many results estimating the ε-entropy are only formulated and proven for C^∞-domains for simplicity of exposition. However, as has been described in Triebel (1978, Section 4.10.3) and Edmunds and Triebel (1996, Section 3.5), these results remain valid for function spaces on more general domains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



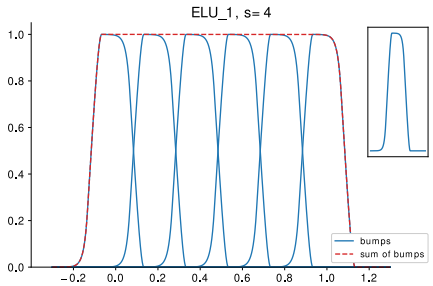
(a) Exponential PU implemented by sigmoid-neural networks with scaling $s = 1$.



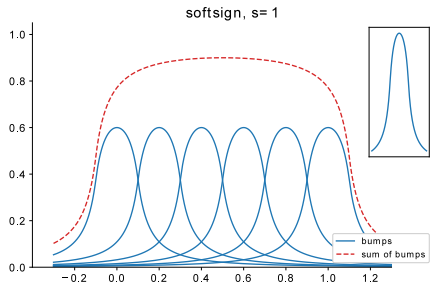
(b) Exponential PU implemented by sigmoid-neural networks with scaling $s = 4$.



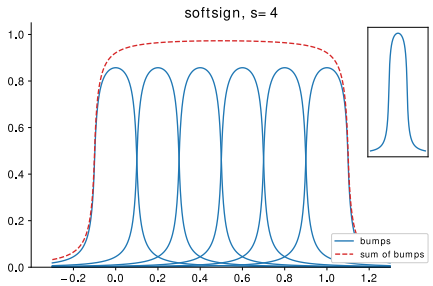
(c) Exponential PU implemented by ELU-neural networks with scaling $s = 1$.



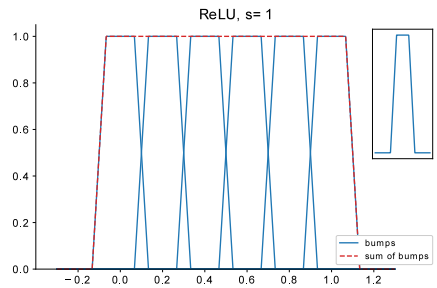
(d) Exponential PU implemented by ELU-neural networks with scaling $s = 4$.



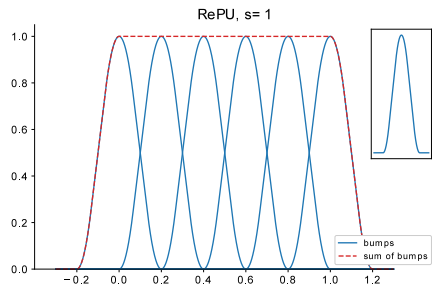
(e) Polynomial PU implemented by softsign-neural networks with scaling $s = 1$.



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



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



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

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

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

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

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

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

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

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

(iii) We have

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Name	Given by	Smoothness Boundedness	PU-Decay (j, τ)	Approximation rates ($k \leq j$)
(leaky) ReLU, $a \in [0, 1)$	$\max\{ax, x\}$	$C(\mathbb{R}) \cap W^{1,\infty}_{\text{loc}}(\mathbb{R})$ Unbounded	Exact $(1, 1)$	$\varepsilon^{-d/(n-k)} \log(1/\varepsilon)$
Exponential linear unit (ELU_a), $a > 0$, $a \neq 1$	$\begin{cases} x, & x \geq 0 \\ a(e^x - 1), & x < 0 \end{cases}$	$C(\mathbb{R}) \cap W^{1,\infty}_{\text{loc}}(\mathbb{R})$ Unbounded	Exponential $(1, 1)$	$\varepsilon^{-d/(n-k)}$
Exponential linear unit (ELU_1)	$\begin{cases} x, & x \geq 0 \\ e^x - 1, & x < 0 \end{cases}$	$C^1(\mathbb{R}) \cap W^{2,\infty}_{\text{loc}}(\mathbb{R})$ Unbounded	Exponential $(2, 1)$	$\varepsilon^{-d/(n-k)}$ for $k \leq 1$, and $\varepsilon^{-d/(n-2-\mu)}$ for $k = 2$
Softsign	$\dfrac{x}{1 + \lvert x \rvert}$	$C^1(\mathbb{R}) \cap W^{2,\infty}(\mathbb{R})$ Bounded	Polynomial $(2, 0)$	$\varepsilon^{-d/(n-k)}$ for $k = 0$
Inverse square root linear unit, $a > 0$	$\begin{cases} x, & x \geq 0 \\ \frac{x}{\sqrt{1+ax^2}}, & x < 0 \end{cases}$	$C^2(\mathbb{R}) \cap W^{3,\infty}_{\text{loc}}(\mathbb{R})$ Unbounded	Polynomial $(3, 1)$	$\varepsilon^{-d/(n-k)}$ for $k \leq 1$
Inverse square root unit, $a > 0$	$\dfrac{x}{\sqrt{1 + ax^2}}$	Analytic Bounded	Polynomial $(j, 0)$ $\forall j \in \mathbb{N}_0$	$\varepsilon^{-d/(n-k)}$ for $k = 0$
Sigmoid/logistic	$\dfrac{1}{1 + e^{-x}}$	Analytic Bounded	Exponential $(j, 0)$ $\forall j \in \mathbb{N}_0$	$\varepsilon^{-d/n}$ for $k = 0$, and $\varepsilon^{-d/(n-k-\mu)}$ for $k \geq 1$
tanh	$\dfrac{e^x - e^{-x}}{e^x + e^{-x}}$	Analytic Bounded	Exponential $(j, 0)$ $\forall j \in \mathbb{N}_0$	$\varepsilon^{-d/n}$ for $k = 0$, and $\varepsilon^{-d/(n-k-\mu)}$ for $k \geq 1$
arctan	$\arctan(x)$	Analytic Bounded	Polynomial $(j, 0)$ $\forall j \in \mathbb{N}_0$	$\varepsilon^{-d/(n-k)}$ for $k = 0$
Softplus	$\ln(1 + e^x)$	Analytic Unbounded	Exponential $(j, 1)$ $\forall j \in \mathbb{N}_0$	$\varepsilon^{-d/n}$ for $k \leq 1$, and $\varepsilon^{-d/(n-k-\mu)}$ for $k \geq 2$
Swish	$\dfrac{x}{1 + e^{-x}}$	Analytic Unbounded	Exponential $(j, 1)$ $\forall j \in \mathbb{N}_0$	$\varepsilon^{-d/n}$ for $k \leq 1$, and $\varepsilon^{-d/(n-k-\mu)}$ for $k \geq 2$
Rectified power unit (RePU), $a \in \mathbb{N}_{\geq 2}$	$\max\{0, x\}^a$	$C^{a-1}(\mathbb{R}) \cap W^{a,\infty}_{\text{loc}}(\mathbb{R})$ Unbounded	Exact (a, a)	$\varepsilon^{-d/(n-k)}$

4.2. Ingredient II: Approximation of polynomials

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

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

(i) $\left\| R_\varrho(\Phi^r_\varepsilon) - x^r \right\|_{C^k([-B,B])} \leq \varepsilon$ *for all* $k = 0, \dots, n$;

(ii) $|R_\varrho(\Phi^r_\varepsilon)|_{W^{k,\infty}([-B,B])} \leq C \frac{r!}{(r-k)!} B^{r-k}$ *for* $k = 0, \dots, r$ *and* $|R_\varrho(\Phi^r_\varepsilon)|_{W^{k,\infty}([-B,B])} \leq \varepsilon$ *for* $k = r + 1, \dots, n$;

(iii) $L\left(\Phi^r_\varepsilon\right) = 2$, *as well as* $M\left(\Phi^r_\varepsilon\right) \leq 3(r + 1)$;

(iv) $\|\Phi^r_\varepsilon\|_{\max} \leq C\varepsilon^{-r}$.

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

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

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

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

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

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

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

[15] For these activation functions see Remark 4.10.

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

4.3. Main results based on ingredients I & II

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Proposition 4.8. *We make the following assumptions:*

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

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

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

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

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

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

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

Theorem 4.9. *We make the following assumptions:*

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

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

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

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

nonzero weights, such that

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

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

- The number of independent weights $|W_\varepsilon|$ is bounded by $C\cdot\varepsilon^{-d/(n-k-\mu_{(k=2)})}$ since the total number of nonzero weights is bounded by this quantity.
- We round the entries of A_f, b_f with a suitable precision ν to the mesh $[-\varepsilon^{-\theta},\varepsilon^{-\theta}]\cap\varepsilon^\nu\mathbb{Z}$, where we also use the fact that the weights of $\Phi_{\varepsilon,f}$ are bounded in absolute value by $C\varepsilon^{-\theta}$.
- The nonzero entries of A_L in the last layer of $\Phi_{\varepsilon,f}$ are in the set $G_{\mathrm{mult}} := \{x_1x_2 : x_1 \in W_\varepsilon, x_2 \in [-\varepsilon^{-\theta},\varepsilon^{-\theta}]\cap\varepsilon^\nu\mathbb{Z}\}$ with cardinality bounded by $\varepsilon^{-\tilde{s}}$ (similar for b_L).

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

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

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

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

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

Declaration of competing interest

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

Acknowledgments

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

Appendix A. Notation and auxiliary results

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Appendix B. Sobolev spaces

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

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

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

and is equipped with the norm

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

Additionally, we set

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

and, consequently

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

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

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

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

from which the statement can easily be deduced.

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

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

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

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

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

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

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

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

and

$$\begin{aligned}
|g \circ f|_{W^{2,\infty}(\Omega_1)} \leq C \left(|g|_{W^{2,\infty}(\Omega_2)}|f|^2_{W^{1,\infty}(\Omega_1;\,\mathbb{R}^m)} \right. \\
\left. + |g|_{W^{1,\infty}(\Omega_2)}|f|_{W^{2,\infty}(\Omega_1;\,\mathbb{R}^m)}\right).
\end{aligned}$$

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

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

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

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

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

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

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

where

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

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

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

and

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

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

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

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

Appendix C. Neural network calculus

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

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

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

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

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

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

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

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

Proof. The neural network

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

with

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

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

fulfills all the desired properties. □

C.1. Approximate monomials and multiplication

We first give the proof of Proposition 4.7:

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

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

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

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

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

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

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

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

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

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

fulfills

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

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

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

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

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

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

This completes the proof. □

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

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

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

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

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

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

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

In detail, we define the neural network

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

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

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

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

$u : [-B, B]^2 \to [-2B, 2B],\ (x, y) \mapsto x + y$ and

$v : [-B, B]^2 \to [-2B, 2B],\ (x, y) \mapsto x - y$,

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

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

and directly see for $k = 0$ that

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

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

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

The composition rule from Corollary B.6 then yields that

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Appendix D. Proof of Proposition 4.8

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

D.1. Approximate partition of unity

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

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

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

$$\begin{aligned}|\phi_0^s(x)| &\leq \frac{1}{s(B-A)}\,|\varrho(3sNx+2s) - \varrho(3sNx+s)| \\ &\quad + \frac{1}{s(B-A)}\,|\varrho(3sNx-s) - \varrho(3sNx-2s)| \\ &\leq 2\frac{\mathrm{Lip}(\varrho)\cdot s}{s(B-A)} = 2\frac{\mathrm{Lip}(\varrho)}{(B-A)}.\end{aligned}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

For the multidimensional case we have,

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

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

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

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

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

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

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

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

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

Now we have that

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

The remainder of the statement can be proven in exactly the same way as the analogous statement for $\tau = 0$.
ad (iv): This immediately follows from the definition of the functions ϕ_m^s. □

D.2. Approximation by localized polynomials

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

for all $N \geq N_1$.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Consequently, we get

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

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

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

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

and, finally, together with Remark B.3

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

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

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

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

D.3. Approximation of localized polynomials by neural networks

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\Psi_{\varepsilon,\Phi_m} = ((A_1',b_1'),(A_2',b_2'),\dots,(A_{L'}',b_{L'}'))$$

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

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

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

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

$$\widetilde{\Psi}_{\varepsilon,\Phi} := \left((A_i',b_i')_{i=1}^{L-1}, \left(\begin{bmatrix} A_L' \\ A_1^{\mathrm{id}} a_L^{(m+1)} \end{bmatrix}, \begin{bmatrix} b_L' \\ A_1^{\mathrm{id}} b_L^{(m+1)} + b_1^{(m+1)} \end{bmatrix}\right), \left(\begin{bmatrix} A_{L+1}' \\ A_2^{\mathrm{id}} \end{bmatrix}, \begin{bmatrix} b_{L+1}' \\ b_2^{\mathrm{id}} \end{bmatrix}\right), \left(\begin{bmatrix} A_{L'}' \\ A_{L'-L+1}^{\mathrm{id}} \end{bmatrix}, \begin{bmatrix} b_{L'}' \\ b_{L'-L+1}^{\mathrm{id}} \end{bmatrix}\right)\right).$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

ad (iii): The estimate

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

can be shown similarly as above.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

and

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

This concludes the proof. □

D.4. Putting everything together

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

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

For the first step, we set

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

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

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

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

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

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

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

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

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

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

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

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

thereby concluding the proof. □

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

We now proceed with the proof of Theorem 4.9.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Clearly,

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

(rounding precision is ε^{ν})

((ii) and (iii) above)

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

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

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

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

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

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

which shows the claim. □

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

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

Exact PUs.

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

Exponential PUs.

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

ELU_1: Here, $j = 2$, $\tau = 1$, $A = 0$ and $B = 1$. Moreover, $R > 0$ can be chosen arbitrarily. Then, for $D = 1$, we have, for all $x > R$, that $|1 - \varrho'(x)| = |1 - 1| = 0$ and, for all $x < -R$ that $|\varrho'(x)| = |e^x| = e^{Dx}$. Moreover, we have for all $|x| > R$ that $|\varrho''(x)| \leq e^{-|x|} = e^{-D|x|}$.

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

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

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

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

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

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

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

Polynomial PUs.

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

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

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

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

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

A Theoretical Analysis of Deep Neural Networks and Parametric PDEs

Gitta Kutyniok [*†‡] Philipp Petersen[§] Mones Raslan[*] Reinhold Schneider[*]

Abstract

We derive upper bounds on the complexity of ReLU neural networks approximating the solution maps of parametric partial differential equations. In particular, without any knowledge of its concrete shape, we use the inherent low-dimensionality of the solution manifold to obtain approximation rates which are significantly superior to those provided by classical neural network approximation results. Concretely, we use the existence of a small reduced basis to construct, for a large variety of parametric partial differential equations, neural networks that yield approximations of the parametric solution maps in such a way that the sizes of these networks essentially only depend on the size of the reduced basis.

Keywords: deep neural networks, parametric PDEs, approximation rates, reduced basis method
Mathematical Subject Classification: 35A35, 35J99, 41A25, 41A46, 68T05, 65N30

1 Introduction

In this work, we analyze the suitability of deep neural networks (DNNs) for the numerical solution of parametric problems. Such problems connect a parameter space with a solution state space via a so-called *parametric map*, [51]. One special case of such a parametric problem arises when the parametric map results from solving a partial differential equation (PDE) and the parameters describe physical or geometrical constraints of the PDE such as, for example, the shape of the physical domain, boundary conditions, or a source term. Applications that lead to these problems include modeling unsteady and steady heat and mass transfer, acoustics, fluid mechanics, or electromagnetics, [33].

Solving a parametric PDE for every point in the parameter space of interest individually, typically leads to two types of problems. First, if the number of parameters of interest is excessive—a scenario coined many-query application—then the associated computational complexity could be unreasonably high. Second, if the computation time is severely limited, such as in real-time applications, then solving even a single PDE might be too costly.

A core assumption to overcome the two issues outlined above is that the solution manifold, i.e., the set of all admissible solutions associated with the parameter space, is inherently low-dimensional. This assumption forms the foundation for the so-called reduced basis method (RBM). A reduced basis discretization is then a (Galerkin) projection on a low-dimensional approximation space that is built from snapshots of the parametrically induced manifold, [60].

Constructing the low-dimensional approximation spaces is typically computationally expensive because it involves solving the PDEs for multiple instances of parameters. These computations take place in a so-called *offline* phase—a step of pre-computation, where one assumes to have access to sufficiently powerful

[*]Institut für Mathematik, Technische Universität Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany, e-mail: {kutyniok, raslan, schneidr}@math.tu-berlin.de
[†]Fakultät für Informatik und Elektrotechnik, Technische Universität Berlin
[‡]Department of Physics and Technology, University of Tromsø
[§]University of Vienna, Faculty of Mathematics and Research Plattform Data Science @ Uni Vienna, Kolingasse 14-16, 1090 Wien, e-mail: philipp.petersen@univie.ac.at

computational resources. Once a suitable low-dimensional space is found, the cost of solving the associated PDEs for a new parameter value is significantly reduced and can be performed quickly and *online*, i.e., with limited resources, [5, 56]. We will give a more thorough introduction to RBMs in Section 2. An extensive survey of works on RBMs, which can be traced back to the seventies and eighties of the last century (see for instance [22, 49, 50]), is beyond the scope of this paper. We refer, for example, to [33, Chapter 1.1], [57, 16, 29] and [12, Chapter 1.9] for (historical) studies of this topic.

In this work, we show that the low-dimensionality of the solution manifold also enables an efficient approximation of the parametric map by DNNs. In this context, the RBM will be, first and foremost, a tool to model this low-dimensionality by acting as a blueprint for the construction of the DNNs.

1.1 Statistical Learning Problems

The motivation to study the approximability of parametric maps by DNNs stems from the following similarities between parametric problems and *statistical learning problems*: Assume that we are given a *domain set* $X \subset \mathbb{R}^n$, $n \in \mathbb{N}$ and a *label set* $Y \subset \mathbb{R}^k$, $k \in \mathbb{N}$. Further assume that there exists an unknown probability distribution ρ on $X \times Y$.

Given a *loss function* $\mathcal{L}\colon Y \times Y \to \mathbb{R}^+$, the goal of a statistical learning problem is to find a function f, which we will call *prediction rule*, from a hypothesis class $H \subset \{h\colon X \to Y\}$ such that the *expected loss* $\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\rho}\mathcal{L}(f(\mathbf{x}),\mathbf{y})$ is minimized, [14]. Since the probability measure ρ is unknown, we have no direct access to the expected loss. Instead, we assume that we are given a set of training data, i.e. pairs $(\mathbf{x}_i,\mathbf{y}_i)_{i=1}^N$, $N \in \mathbb{N}$, which were drawn independently with respect to ρ. Then one finds f by minimizing the so-called *empirical loss*

$$\sum_{i=1}^{N} \mathcal{L}(f(\mathbf{x}_i),\mathbf{y}_i) \tag{1.1}$$

over H. We will call optimizing the empirical loss the *learning procedure*.

In view of PDEs, the approach proposed above can be rephrased in the following way. We are aiming to produce a function from a parameter set to a state space based on a few snapshots only. This function should satisfy the involved PDEs as precisely as possible, and the evaluation of this function should be very efficient even though the construction of it can potentially be computationally expensive.

In the above-described sense, a parametric PDE problem almost perfectly matches the definition of a statistical learning problem. Indeed, the PDEs and the metric on the state space correspond to a (deterministic) distribution ρ and a loss function. Moreover, the snapshots are construed as the training data, and the offline phase mirrors the learning procedure. Finally, the parametric map is the prediction rule.

One of the most efficient learning methods nowadays is deep learning. This method describes a range of learning procedures to solve statistical learning problems where the hypothesis class H is taken to be a set of DNNs, [40, 24]. These methods outperform virtually all classical machine learning techniques in sufficiently complicated tasks from speech recognition to image classification. Strikingly, training DNNs is a computationally very demanding task that is usually performed on highly parallelized machines. Once a DNN is fully trained, however, its application to a given input is orders of magnitudes faster than the training process. This observation again reflects the offline-online phase distinction that is common in RBM approaches.

Based on the overwhelming success of these techniques and the apparent similarities of learning problems and parametric problems it appears natural to apply methods from deep learning to statistical learning problems in the sense of (partly) replacing the parameter-dependent map by a DNN. Very successful advances in this direction have been reported in [39, 34, 42, 68, 58, 17].

1.2 Our Contribution

In the applications [39, 34, 42, 68, 58, 17] mentioned above, the combination of DNNs and parametric problems seems to be remarkably efficient. In this paper, we present a theoretical justification of this approach.

We address the question to what extent the hypothesis class of DNNs is sufficiently broad to approximately and efficiently represent the associated parametric maps. Concretely, we aim at understanding the necessary number of parameters of DNNs required to allow a sufficiently accurate approximation. We will demonstrate that depending on the target accuracy the required number of parameters of DNNs essentially only scales with the intrinsic dimension of the solution manifold, in particular, according to its Kolmogorov N-widths. We outline our results in Subsection 1.2.1. Then, we present a simplified exposition of our argument leading to the main results in Subsection 1.2.2.

1.2.1 Approximation Theoretical Results

The main contributions of this work is given by an approximation result with DNNs based on ReLU activation functions. Here, we aim to learn a variation of the parametric map

$$\mathcal{Y} \ni y \mapsto u_y \in \mathcal{H},$$

where $\mathcal{Y}$ is the parameter space and $\mathcal{H}$ is a Hilbert space. In our case, the parameter space will be a compact subset of $\mathbb{R}^p$ for some fixed, but possibly large $p \in \mathbb{N}$, i.e., we consider the case of finitely supported parameter vectors.

We assume that there exists a basis of a high-fidelity discretization of $\mathcal{H}$ which may potentially be quite large. Let $\mathbf{u}_y$ be the coefficient vector of u_y with respect to the high-fidelity discretization. Moreover, we assume that there exists a RB approximating u_y sufficiently accurately for every $y \in \mathcal{Y}$.

Theorem 4.3 then states that, under some technical assumptions, there exists a DNN that approximates the *discretized solution map*

$$\mathcal{Y} \ni y \mapsto \mathbf{u}_y$$

up to a uniform error of $\epsilon > 0$, while having a size that is polylogarithmical in ϵ, cubic in the size of the reduced basis, and at most linear in the size of the high-fidelity basis.

This result highlights the common observation that, if a low-dimensional structure is present in a problem, then DNNs are able to identify it and use it advantageously. Concretely, our results show that a DNN is sufficiently flexible to benefit from the existence of a reduced basis in the sense that its size in the complex task of solving a parametric PDE does not or only weakly depend on the high-fidelity discretization and mainly on the size of the reduced basis.

The main result is based on four pillars that are described in detail in Subsection 1.2.2: First, we show that DNNs can efficiently solve linear systems, in the sense that, if supplied with a matrix and a right-hand side, a moderately-sized network outputs the solution of the inverse problem. Second, the reduced-basis approach allows reformulating the parametric problem, as a relatively small and parametrized linear system. Third, in many cases, the map that takes the parameters to the stiffness matrices with respect to the reduced basis and right-hand side can be very efficiently represented by DNNs. Finally, the fact that neural networks are naturally compositional allows combining the efficient representation of linear problems with the NN implementing operator inversion.

In practice, the approximating DNNs that we show to exist need to be found using a learning algorithm. In this work, we will not analyze the feasibility of learning these DNNs. The typical approach here is to apply methods based on stochastic gradient descent. Empirical studies of this procedure in the context of learning deep neural networks were carried out in [39, 34, 42, 68, 58]. In particular, we mention the recent study in [23], which analyzes precisely the set-up described in this work and finds a strong impact of the approximation-theoretical behavior of DNNs on their practical performance.

1.2.2 Simplified Presentation of the Argument

In this section, we present a simplified outline of the arguments leading to the approximation result described in Subsection 1.2.1. In this simplified setup, we think of a ReLU neural network (ReLU NN) as a function

$$\mathbb{R}^n \to \mathbb{R}^k,\ \mathbf{x} \mapsto T_L \varrho(T_{L-1}\varrho(\dots \varrho(T_1(\mathbf{x})))), \tag{1.2}$$

where $L \in \mathbb{N}$, $T_1, \dots, T_L$ are affine maps, and $\varrho : \mathbb{R} \to \mathbb{R}$, $\varrho(x) \coloneqq \max\{0, x\}$ is the ReLU activation function which is applied coordinate-wise in (1.2). We call L the number of layers of the NN. Since T_ℓ are affine linear maps, we have for all $\mathbf{x} \in \operatorname{dom} T_\ell$ that $T_\ell(\mathbf{x}) = \mathbf{A}_\ell(\mathbf{x}) + \mathbf{b}_\ell$ for a matrix $\mathbf{A}_\ell$ and a vector $\mathbf{b}_\ell$. We define the size of the NN as the number of non-zero entries of all $\mathbf{A}_\ell$ and $\mathbf{b}_\ell$ for $\ell \in \{1, \dots, L\}$. This definition will later be sharpened and extended in Definition 3.1.

1. As a first step, we recall the construction of a *scalar multiplication operator by ReLU NNs* due to [69]. This construction is based on two observations. First, defining $g\colon [0,1] \to [0,1]$, $g(x) \coloneqq \min\{2x, 2-2x\}$, we see that g is a hat function. Moreover, multiple compositions of g with itself produce saw-tooth functions. We set, for $s \in \mathbb{N}$, $g_1 \coloneqq g$ and $g_{s+1} \coloneqq g \circ g_s$. It was demonstrated in [69] that

$$x^2 = \lim_{n\to\infty} f_n(x) \coloneqq \lim_{n\to\infty} x - \sum_{s=1}^{n} \frac{g_s(x)}{2^{2s}}, \quad \text{for all } x \in [0,1]. \tag{1.3}$$

   The second observation for establishing an approximation of a scalar multiplication by NNs is that we can write $g(x) = 2\varrho(x) - 4\varrho(x - 1/2) + 2\varrho(x-2)$ and therefore g_s can be exactly represented by a ReLU NN. Given that g_s is bounded by 1, it is not hard to see that f_n converges to the square function exponentially fast for $n \to \infty$. Moreover, f_n can be implemented exactly as a ReLU NN by previous arguments. Finally, the parallelogram identity, $xz = 1/4((x+z)^2 - (x-z)^2)$ for $x, z \in \mathbb{R}$, demonstrates how an approximate realization of the square function by ReLU NNs yields an approximate realization of scalar multiplication by ReLU NNs.

   It is intuitively clear from the exponential convergence in (1.3) and proved in [69, Proposition 3] that the size of a NN approximating the scalar multiplication on $[-1,1]^2$ up to an error of $\epsilon > 0$ is $\mathcal{O}(\log_2(1/\epsilon))$.

2. As a next step, we use the approximate scalar multiplication to approximate a *multiplication operator for matrices by ReLU NNs*. A matrix multiplication of two matrices of size $d \times d$ can be performed using d^3 scalar multiplications. Of course, as famously shown in [64], a more efficient matrix multiplication can also be carried out with less than d^3 multiplications. However, for simplicity, we focus here on the most basic implementation of matrix multiplication. Hence, the approximate multiplication of two matrices with entries bounded by 1 can be performed by NN of size $\mathcal{O}(d^3 \log_2(1/\epsilon))$ with accuracy $\epsilon > 0$. We make this precise in Proposition 3.7. Along the same lines, we can demonstrate how to construct a *NN emulating matrix-vector multiplications*.

3. Concatenating multiple matrix multiplications, we can implement *matrix polynomials by ReLU NNs*. In particular, for $\mathbf{A} \in \mathbb{R}^{d\times d}$ such that $\|\mathbf{A}\|_2 \leq 1 - \delta$ for some $\delta \in (0,1)$, the map $\mathbf{A} \mapsto \sum_{s=0}^{m} \mathbf{A}^s$ can be approximately implemented by a ReLU NN with an accuracy of $\epsilon > 0$ and which has a size of $\mathcal{O}(m \log_2^2(m) d^3 \cdot (\log(1/\epsilon) + \log_2(m))$, where the additional $\log_2$ term in m inside the brackets appears since each of the approximations of the sum needs to be performed with accuracy ϵ/m. It is well known, that the *Neumann series* $\sum_{s=0}^{m} \mathbf{A}^s$ converges exponentially fast to $(\mathbf{Id}_{\mathbb{R}^d} - \mathbf{A})^{-1}$ for $m \to \infty$. Therefore, under suitable conditions on the matrix $\mathbf{A}$, we can construct a NN $\Phi_\epsilon^{\mathrm{inv}}$ that *approximates the inversion operator*, i.e. the map $\mathbf{A} \mapsto \mathbf{A}^{-1}$ up to accuracy $\epsilon > 0$. This NN has size $\mathcal{O}(d^3 \log_2^q(1/\epsilon))$ for a constant $q > 0$. This is made precise in Theorem 3.8.

4. The existence of $\Phi_\epsilon^{\mathrm{inv}}$ and the emulation of approximate matrix-vector multiplications yield that there exists a NN that for a given matrix and right-hand side approximately solves the associated linear system. Next, we make two assumptions that are satisfied in many applications as we demonstrate in Subsection 4.2:

   - The map from the parameters to the associated stiffness matrices of the Galerkin discretization of the parametric PDE with respect to a reduced basis can be well approximated by NNs.
   - The map from the parameters to the right-hand side of the parametric PDEs discretized according to the reduced basis can be well approximated by NNs.

From these assumptions and the existence of $\Phi_\epsilon^{\mathrm{inv}}$ and a ReLU NN emulating a matrix-vector multiplication, it is not hard to see that there is a NN that *approximately implements the map from a parameter to the associated discretized solution with respect to the reduced basis.* If the reduced basis has size d and the implementations of the map yielding the stiffness matrix and the right-hand side are sufficiently efficient then, by the construction of $\Phi_\epsilon^{\mathrm{inv}}$, the resulting NN has size $\mathcal{O}(d^3 \log_2^q(1/\epsilon))$. We call this NN $\Phi_\epsilon^{\mathrm{rb}}$.

5. Finally, we build on the construction of $\Phi_\epsilon^{\mathrm{rb}}$ to establish the result of Section 1.2.1. First of all, let D be the size of the high-fidelity basis. If D is sufficiently large, then every element from the reduced basis can be approximately represented in the high-fidelity basis. Therefore, one can perform an *approximation to a change of bases* by applying a linear map $\mathbf{V} \in \mathbb{R}^{D\times d}$ to a vector with respect to the reduced basis. The first statement of Subsection 1.2.1 now follows directly by considering the NN $\mathbf{V} \circ \Phi_\epsilon^{\mathrm{rb}}$. Through this procedure, the size of the NN is increased to $\mathcal{O}(d^3 \log_2^q(1/\epsilon)) + dD)$. The full argument is presented in the proof of Theorem 4.3.

1.3 Potential Impact and Extensions

We believe that the results of this article have the potential to significantly impact the research on NNs and parametric problems in the following ways:

- *Theoretical foundation:* We offer a theoretical underpinning for the empirical success of NNs for parametric problems which was observed in, e.g., [39, 34, 42, 68, 58]. Indeed, our result, Theorem 4.3, indicates that properly trained NNs are as efficient in solving parametric PDEs as RBMs if the complexity of NNs is measured in terms of free parameters. On a broader level, linking deep learning techniques for parametric PDE problems with approximation theory opens the field up to a new direction of thorough mathematical analysis.

- *Understanding the role of the ambient dimension:* It has been repeatedly observed that NNs seem to offer approximation rates of high-dimensional functions that do not deteriorate exponentially with increasing dimension, [45, 24].

  In this context, it is interesting to identify the key quantity determining the achievable approximation rates of DNNs. Possible explanation for approximation rates that are essentially independent from the ambient dimension have been identified if the functions to be approximated have special structures such as compositionality, [48, 55], or invariances, [45, 53]. In this article, we identify the highly problem-specific notion of the *dimension of the solution manifold* as a key quantity determining the achievable approximation rates by NNs for parametric problems. We discuss the connection between the approximation rates that NNs achieve and the ambient dimension in detail in Section 5.

- *Identifying suitable architectures:* One question in applications is how to choose the right NN architectures for the associated problem. Our results show that NNs of sufficient depth and size are able to produce very efficient approximations. Nonetheless, it needs to be mentioned that our results do not yield a lower bound on the number of layers and thus it is not clear whether deep NNs are indeed necessary.

This work is a step towards establishing a theory of deep learning-based solutions of parametric problems. However, given the complexity of this field, it is clear that many more steps need to follow. We outline a couple of natural further questions of interest below:

- *General parametric problems:* Below we restrict ourselves to coercive, symmetric, and linear parametric problems with finitely many parameters. There exist many extensions to, e.g. noncoercive, nonsymmetric, or nonlinear problems, [67, 25, 10, 38, 11, 70], or to infinite parameter spaces, see e.g. [4, 2]. It would be interesting to see if the methods proposed in this work can be generalized to these more challenging situations.

- *Bounding the number of snapshots:* The interpretation of the parametric problem as a statistical learning problem has the convenient side-effect that various techniques have been established to bound the number of necessary samples N, such that the empirical loss (1.1) is very close to the expected loss. In other words, the generalization error of the minimizer of the learning procedure is small, meaning that the prediction rule performs well on unseen data. (Here, the error is measured in a norm induced by the loss function and the underlying probability distribution.). Using these techniques, it is possible to bound the number of snapshots required for the offline phase to achieve a certain fidelity in the online phase. Estimates of the generalization error in the context of high-dimensional PDEs have been deduced in, e.g., [19, 26, 7, 20, 59].

- *Special NN architectures:* This article studies the feasibility of standard feed-forward NNs. In practice, one often uses special architectures that have proved efficient in applications. First and foremost, almost all NNs used in applications are convolutional neural networks (CNNs), [41]. Hence a relevant question is to what extent the results of this work also hold for such architectures. It was demonstrated in [54] that there is a direct correspondence between the approximation rates of CNNs and that of standard NNs. Thus we expect that the results of this work translate to CNNs.

  Another successful architecture is that of residual neural networks (ResNets), [32]. These neural networks also admit skip-connections, i.e., do not only connect neurons in adjacent layers. This architecture is by design at least as powerful as a standard NN and hence inherits all approximation properties of standard NNs.

- *Necessary properties of neural networks:* In this work, we demonstrate the attainability of certain approximation rates by NNs. It is not clear if the presented results are optimal or if there are specific necessary assumptions on the architectures, such as a minimal depth, a minimal number of parameters, or a minimal number of neurons per layer. For approximation results of classical function spaces such lower bounds on specifications of NNs have been established for example in [9, 28, 53, 69]. It is conceivable that the techniques in these works can be transferred to the approximation tasks described in this work.

- *General matrix polynomials:* As outlined in Subsection 1.2.2, our results are based on the approximate implementation of matrix polynomials. Naturally, this construction can be used to define and construct a ReLU NN based functional calculus. In other words, for any $d \in \mathbb{N}$ and every continuous function f that can be well approximated by polynomials, we can construct a ReLU NN which approximates the map $\mathbf{A} \mapsto f(\mathbf{A})$ for any appropriately bounded matrix $\mathbf{A}$.

  A special instance of such a function of interest is given by $f(\mathbf{A}) \coloneqq e^{t\mathbf{A}},\ t > 0$, which is analytic and plays an important role in the treatment of initial value problems.

- *Numerical studies:* In a practical learning problem, the approximation-theoretical aspect only describes one part of the problem. Two further central factors are the data generation and the optimization process. It is conceivable that in comparison to these issues, approximation theoretical considerations only play a negligible role. To understand the extent to which the result of this paper is relevant for applications, comprehensive studies of the theoretical set-up of this work should be carried out. A first one was published recently in [23].

1.4 Related Work

In this section, we give an extensive overview of works related to this paper. In particular, for completeness, we start by giving a review of approximation theory of NNs without an explicit connection to PDEs. Afterward, we will see how NNs have been employed for the solution of PDEs.

1.4.1 Review of Approximation Theory of Neural Networks

The first and most fundamental results on the approximation capabilities of NNs were universality results. These results claim that NNs with at least one hidden layer can approximate any continuous function on a bounded domain to arbitrary accuracy if they have sufficiently many neurons, [35, 15]. However, these results do not quantify the required sizes of NNs to achieve these rates. One of the first results in this direction was given in [6]. There, a bound on the sufficient size of NNs with sigmoidal activation functions approximating a function with finite Fourier moments is presented. Further results describe approximation rates for various smoothness classes by sigmoidal or even more general activation functions, [47, 46, 43, 44].

For the non-differentiable activation function ReLU, first rates of approximation were identified in [69] for classes of smooth functions, in [53] for piecewise smooth functions, and in [27] for oscillatory functions. Moreover, NNs mirror the approximation rates of various dictionaries such as wavelets, [62], general affine systems, [9], linear finite elements, [31], and higher-order finite elements, [52].

1.4.2 Neural Networks and PDEs

A well-established line of research is that of solving high-dimensional PDEs by NNs assuming that the NN is the solution of the underlying PDE, e.g., [63, 7, 30, 37, 59, 36, 19, 20]. In this regime, it is often possible to bound the size of the involved NNs in a way that does not scale exponentially with the underlying dimension. In that way, these results are quite related to our approaches. Our results do not seek to represent the solution of a PDE as a NN, but a parametric map. Moreover, we analyze the complexity of the solution manifold in terms of Kolmogorov N-widths. Finally, the underlying spatial dimension of the involved PDEs in our case would usually be moderate. However, the dimension of the parameter space could be immense.

One of the first approaches analyzing NN approximation rates for solutions of parametric PDEs was carried out in [61]. In that work, the analyticity of the solution map $y \mapsto u_y$ and polynomial chaos expansions with respect to the parametric variable are used to approximate the map $y \mapsto u_y$ by ReLU NNs of moderate size. Moreover, we mention the works [39, 34, 42, 68, 58] which apply NNs in one way or another to parametric problems. These approaches study the topic of learning a parametric problem but do not offer a theoretical analysis of the required sizes of the involved NNs. These results form our motivation to study the constructions of this paper.

Finally, we mention that the setup of the recent numerical study [23] is closely related to this work.

1.5 Outline

In Section 2, we describe the type of parametric PDEs that we consider in this paper, and we recall the theory of RBs. Section 3 introduces a NN calculus which is the basis for all constructions in this work. There we will also construct the NN that maps a matrix to its approximate inverse in Theorem 3.8. In Section 4, we construct NNs approximating parametric maps. First, in Theorem 4.1, we approximate the parametric maps after a high-fidelity discretization. Afterward, in Subsection 4.2, we list two broad examples where all assumptions which we imposed are satisfied.

We conclude this paper in Section 5 with a discussion of our results in light of the dependence of the underlying NN complexities in terms of the governing quantities.

To not interrupt the flow of reading, we have deferred all auxiliary results and proofs to the appendices.

1.6 Notation

We denote by $\mathbb{N} = \{1, 2, ...\}$ the set of all *natural numbers* and define $\mathbb{N}_0 \coloneqq \mathbb{N} \cup \{0\}$. Moreover, for $a \in \mathbb{R}$ we set $\lfloor a \rfloor \coloneqq \max\{b \in \mathbb{Z} \colon\ b \leq a\}$ and $\lceil a \rceil \coloneqq \min\{b \in \mathbb{Z} \colon\ b \geq a\}$. Let $n, l \in \mathbb{N}$. Let $\mathbf{Id}_{\mathbb{R}^n}$ be the *identity* and $\mathbf{0}_{\mathbb{R}^n}$ be the *zero vector* on $\mathbb{R}^n$. Moreover, for $\mathbf{A} \in \mathbb{R}^{n \times l}$, we denote by $\mathbf{A}^T$ its *transpose,* by $\sigma(\mathbf{A})$ the *spectrum of* $\mathbf{A}$, by $\|\mathbf{A}\|_2$ its *spectral norm* and by $\|\mathbf{A}\|_0 \coloneqq \#\{(i,j) \colon \mathbf{A}_{i,j} \neq 0\}$, where $\#V$ denotes the cardinality of a set V, the *number of non-zero entries* of $\mathbf{A}$. Moreover, for $\mathbf{v} \in \mathbb{R}^n$ we denote by $|\mathbf{v}|$ its *Euclidean norm.* Let V be a vector space. Then we say that $X \subset^{\mathrm{s}} V$, if X is a *linear subspace* of V. Moreover, if $(V, \|\cdot\|_V)$ is a

normed vector space, X is a subset of V and $v \in V$, we denote by $\operatorname{dist}(v, X) \coloneqq \inf\{\|x - v\|_V \colon\ x \in X\}$ the *distance* between v, X and by $(V^*, \|\cdot\|_{V^*})$ the *topological dual space of* V, i.e. the set of all scalar-valued, linear, continuous functions equipped with the *operator norm*. For a compact set $\Omega \subset \mathbb{R}^n$ we denote by $C^r(\Omega),\ r \in \mathbb{N}_0 \cup \{\infty\}$, the spaces of r *times continuously differentiable functions*, by $L^p(\Omega, \mathbb{R}^n), p \in [1, \infty]$ the $\mathbb{R}^n$*-valued Lebesgue spaces*, where we set $L^p(\Omega) \coloneqq L^p(\Omega, \mathbb{R})$ and by $H^1(\Omega) \coloneqq W^{1,2}(\Omega)$ the *first-order Sobolev space*.

2 Parametric PDEs and Reduced Basis Methods

In this section, we introduce the type of parametric problems that we study in this paper. A parametric problem in its most general form is based on a map $\mathcal{P} \colon \mathcal{Y} \to \mathcal{Z}$, where $\mathcal{Y}$ is the *parameter space* and $\mathcal{Z}$ is called *solution state space* $\mathcal{Z}$. In the case of parametric PDEs, $\mathcal{Y}$ describes certain parameters of a partial differential equation, $\mathcal{Z}$ is a function space or a discretization thereof, and $\mathcal{P}(y) \in \mathcal{Z}$ is found by solving a PDE with parameter y.

We will place several assumptions on the PDEs underlying $\mathcal{P}$ and the parameter spaces $\mathcal{Y}$ in Section 2.1. Afterward, we give an abstract overview of Galerkin methods in Section 2.2 before recapitulating some basic facts about RBs in Section 2.3.

2.1 Parametric Partial Differential Equations

In the following, we will consider parameter-dependent equations given in the variational form

$$b_y(u_y, v) = f_y(v), \quad \text{for all } y \in \mathcal{Y},\ v \in \mathcal{H}, \tag{2.1}$$

where

(i) $\mathcal{Y}$ is the *parameter set* specified in Assumption 2.1,
(ii) $\mathcal{H}$ is a Hilbert space,
(iii) $b_y \colon \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ is a *continuous bilinear form*, which fulfills certain well-posedness conditions specified in Assumption 2.1,
(iv) $f_y \in \mathcal{H}^*$ is the *parameter-dependent right-hand side of* (2.1),
(v) $u_y \in \mathcal{H}$ is the *solution of* (2.1).

Assumption 2.1. *Throughout this paper, we impose the following assumptions on Equation* (2.1).

- ***The parameter set*** $\mathcal{Y}$*: We assume that* $\mathcal{Y}$ *is a compact subset of* $\mathbb{R}^p$*, where* $p \in \mathbb{N}$ *is fixed and potentially large.*

  Remark. *In [12, Section 1.2], it has been demonstrated that if* $\mathcal{Y}$ *is a compact subset of some Banach space* V*, then one can describe every element in* $\mathcal{Y}$ *by a sequence of real numbers in an affine way. To be more precise, there exist* $(v_i)_{i=0}^{\infty} \subset V$ *such that for every* $y \in \mathcal{Y}$ *and some coefficient sequence* $\mathbf{c}_y$ *whose elements can be bounded in absolute value by 1 there holds* $y = v_0 + \sum_{i=1}^{\infty} (\mathbf{c}_y)_i v_i$*, implying that* $\mathcal{Y}$ *can be completely described by the collection of sequences* $\mathbf{c}_y$*. In this paper, we assume these sequences* $\mathbf{c}_y$ *to be finite with a fixed, but possibly large support size.*

- ***Symmetry, uniform continuity, and coercivity of the bilinear forms:*** *We assume that for all* $y \in \mathcal{Y}$ *the bilinear forms* b_y *are* symmetric*, i.e.*

  $$b_y(u, v) = b_y(v, u), \qquad \text{for all } u, v \in \mathcal{H}.$$

  Moreover, we assume that the bilinear forms b_y *are* uniformly continuous *in the sense that there exists a constant* $C_{\mathrm{cont}} > 0$ *with*

  $$|b_y(u, v)| \leq C_{\mathrm{cont}} \|u\|_{\mathcal{H}} \|v\|_{\mathcal{H}}, \qquad \text{for all } u \in \mathcal{H}, v \in \mathcal{H}, y \in \mathcal{Y}.$$

Finally, we assume that the involved bilinear forms are uniformly coercive *in the sense that there exists a constant* $C_{\mathrm{coer}} > 0$ *such that*

$$\inf_{u \in \mathcal{H} \setminus \{0\}} \frac{b_y(u,u)}{\|u\|_{\mathcal{H}}^2} \geq C_{\mathrm{coer}}, \qquad \text{for all } u \in \mathcal{H}, y \in \mathcal{Y}.$$

Hence, by the Lax-Milgram lemma (see [57, Lemma 2.1]), Equation (2.1) *is* well-posed*, i.e. for every* $y \in \mathcal{Y}$ *and every* $f_y \in \mathcal{H}^*$ *there exists exactly one* $u_y \in \mathcal{H}$ *such that* (2.1) *is satisfied and* u_y *depends continuously on* f_y*.*

- ***Uniform boundedness of the right-hand side:*** *We assume that there exists a constant* $C_{\mathrm{rhs}} > 0$ *such that*

$$\|f_y\|_{\mathcal{H}^*} \leq C_{\mathrm{rhs}}, \qquad \text{for all } y \in \mathcal{Y}.$$

- ***Compactness of the solution manifold:*** *We assume that the* solution manifold

$$S(\mathcal{Y}) \coloneqq \{u_y \colon u_y \text{ is the solution of } (2.1),\ y \in \mathcal{Y}\}$$

is compact in $\mathcal{H}$*.*

Remark. *The assumption that* $S(\mathcal{Y})$ *is compact follows immediately if the solution map* $y \mapsto u_y$ *is continuous. This condition is true (see [57, Proposition 5.1, Corollary 5.1]), if for all* $u, v \in \mathcal{H}$ *the maps* $y \mapsto b_y(u,v)$ *as well as* $y \to f_y(v)$ *are Lipschitz continuous. In fact, there exists a multitude of parametric PDEs, for which the maps* $y \mapsto b_y(u,v)$ *and* $y \to f_y(v)$ *are even in* C^r *for some* $r \in \mathbb{N} \cup \{\infty\}$*. In this case,* $\{(y, u_y) : y \in \mathcal{Y}\} \subset \mathbb{R}^p \times \mathcal{H}$ *is a* p*-dimensional manifold of class* C^r *(see [57, Proposition 5.2, Remark 5.4]). Moreover, we refer to [57, Remark 5.2] and the references therein for a discussion under which circumstances it is possible to turn a discontinuous parameter dependency into a continuous one ensuring the compactness of* $S(\mathcal{Y})$*.*

2.2 High-Fidelity Approximations

In practice, one cannot hope to solve (2.1) exactly for every $y \in \mathcal{Y}$. Instead, if we assume for the moment that y is fixed, a common approach towards the calculation of an approximate solution of (2.1) is given by the *Galerkin method*, which we will describe shortly following [33, Appendix A] and [57, Chapter 2.4]. In this framework, instead of solving (2.1), one solves a discrete scheme of the form

$$b_y\left(u_y^{\mathrm{disc}}, v\right) = f_y(v) \qquad \text{for all } v \in U^{\mathrm{disc}}, \tag{2.2}$$

where $U^{\mathrm{disc}} \subset^{\mathrm{s}} \mathcal{H}$ is a subspace of $\mathcal{H}$ with $\dim\left(U^{\mathrm{disc}}\right) < \infty$ and $u_y^{\mathrm{disc}} \in U^{\mathrm{disc}}$ is the solution of (2.2). For the solution u_y^{disc} of (2.2) we have that

$$\left\|u_y^{\mathrm{disc}}\right\|_{\mathcal{H}} \leq \frac{1}{C_{\mathrm{coer}}} \|f_y\|_{\mathcal{H}^*}.$$

Moreover, up to a constant, we have that u_y^{disc} is a best approximation of the solution u_y of (2.1) by elements in U^{disc}. To be more precise, by *Cea's Lemma*, [57, Lemma 2.2],

$$\left\|u_y - u_y^{\mathrm{disc}}\right\|_{\mathcal{H}} \leq \frac{C_{\mathrm{cont}}}{C_{\mathrm{coer}}} \inf_{w \in U^{\mathrm{disc}}} \|u_y - w\|_{\mathcal{H}}. \tag{2.3}$$

Let us now assume that U^{disc} is given. Moreover, if $N \coloneqq \dim\left(U^{\mathrm{disc}}\right)$, let $(\varphi_i)_{i=1}^N$ be a basis for U^{disc}. Then the matrix

$$\mathbf{B}_y \coloneqq \left(b_y\left(\varphi_j, \varphi_i\right)\right)_{i,j=1}^N$$

is non-singular and positive definite. The solution u_y^{disc} of (2.2) satisfies

$$u_y^{\text{disc}} = \sum_{i=1}^{N} (\mathbf{u}_y)_i \varphi_i,$$

where

$$\mathbf{u}_y \coloneqq (\mathbf{B}_y)^{-1} \mathbf{f}_y \in \mathbb{R}^N$$

and $\mathbf{f}_y \coloneqq (f_y(\varphi_i))_{i=1}^N \in \mathbb{R}^N$. Typically, one starts with a *high-fidelity discretization* of the space $\mathcal{H}$, i.e. one chooses a finite- but potentially high-dimensional subspace for which the computed discretized solutions are sufficiently accurate for any $y \in \mathcal{Y}$. To be more precise, we postulate the following:

Assumption 2.2. *We assume that there exists a finite dimensional space $U^{\text{h}} \subset^{\text{s}} \mathcal{H}$ with dimension $D < \infty$ and basis $(\varphi_i)_{i=1}^D$. This space is called* high-fidelity discretization. *For $y \in \mathcal{Y}$, denote by $\mathbf{B}_y^{\text{h}} \coloneqq (b_y(\varphi_j, \varphi_i))_{i,j=1}^D \in \mathbb{R}^{D\times D}$ the stiffness matrix of the high-fidelity discretization, by $\mathbf{f}_y^{\text{h}} \coloneqq (f_y(\varphi_i))_{i=1}^D$ the discretized right-hand side, and by $\mathbf{u}_y^{\text{h}} \coloneqq \left(\mathbf{B}_y^{\text{h}}\right)^{-1} \mathbf{f}_y^{\text{h}} \in \mathbb{R}^D$ the coefficient vector of the Galerkin solution with respect to the high-fidelity discretization. Moreover, we denote by $u_y^{\text{h}} \coloneqq \sum_{i=1}^D \left(\mathbf{u}_y^{\text{h}}\right)_i \varphi_i$ the Galerkin solution.*

We assume that, for every $y \in \mathcal{Y}$, $\sup_{y\in\mathcal{Y}} \left\|u_y - u_y^{\text{h}}\right\|_{\mathcal{H}} \leq \hat{\epsilon}$ for an arbitrarily small, but fixed $\hat{\epsilon} > 0$. In the following, similarly as in [16], we will not distinguish between $\mathcal{H}$ and U^{h}, unless such a distinction matters.

In practice, following this approach, one often needs to calculate $u_y^{\text{h}} \approx u_y$ for a variety of parameters $y \in \mathcal{Y}$. This, in general, is a very expensive procedure due to the high-dimensionality of the space U^{h}. In particular, given $(\varphi_i)_{i=1}^D$, one needs to solve high-dimensional systems of linear equations to determine the coefficient vector $\mathbf{u}_y^{\text{h}}$. A well-established remedy to overcome these difficulties is given by methods based on the theory of reduced bases, which we will recapitulate in the upcoming subsection.

Before we proceed, let us fix some notation. We denote by $\mathbf{G} \coloneqq (\langle\varphi_i, \varphi_j\rangle_{\mathcal{H}})_{i,j=1}^D \in \mathbb{R}^{D\times D}$ the symmetric, positive definite *Gram matrix* of the basis vectors $(\varphi_i)_{i=1}^D$. Then, for any $v \in U^{\text{h}}$ with coefficient vector $\mathbf{v}$ with respect to the basis $(\varphi_i)_{i=1}^D$ we have (see [57, Equation 2.41])

$$|\mathbf{v}|_{\mathbf{G}} \coloneqq \left|\mathbf{G}^{1/2}\mathbf{v}\right| = \|v\|_{\mathcal{H}}. \tag{2.4}$$

2.3 Theory of Reduced Bases

In this subsection and unless stated otherwise, we follow [57, Chapter 5] and the references therein. The main motivation behind the theory of RBs lies in the fact that under Assumption 2.1 the solution manifold $S(\mathcal{Y})$ is a compact subset of $\mathcal{H}$. This compactness property allows posing the question whether, for every $\tilde{\epsilon} \geq \hat{\epsilon}$, it is possible to construct a finite-dimensional subspace $U_{\tilde{\epsilon}}^{\text{rb}}$ of $\mathcal{H}$ such that $d(\tilde{\epsilon}) \coloneqq \dim\left(U_{\tilde{\epsilon}}^{\text{rb}}\right) \ll D$ and such that

$$\sup_{y\in\mathcal{Y}} \inf_{w\in U_{\tilde{\epsilon}}^{\text{rb}}} \|u_y - w\|_{\mathcal{H}} \leq \tilde{\epsilon}, \tag{2.5}$$

or, equivalently, if there exist linearly independent vectors $(\psi_i)_{i=1}^{d(\tilde{\epsilon})}$ with the property that

$$\left\|\sum_{i=1}^{d(\tilde{\epsilon})} (\mathbf{c}_y)_i \psi_i - u_y\right\|_{\mathcal{H}} \leq \tilde{\epsilon}, \quad \text{for all } y \in \mathcal{Y} \text{ and some coefficient vector } \mathbf{c}_y \in \mathbb{R}^{d(\tilde{\epsilon})}.$$

The starting point of this theory lies in the concept of the Kolmogorov N-width which is defined as follows.

Definition 2.3 ([16])**.** *For $N \in \mathbb{N}$, the* Kolmogorov N-width *of a bounded subset X of a normed space V is defined by*

$$W_N(X) \coloneqq \inf_{\substack{V_N\subset^{\text{s}}V \\ \dim(V_N)\leq N}} \sup_{x\in X} \operatorname{dist}(x, V_N).$$

This quantity describes the best possible uniform approximation error of X by an at most N-dimensional linear subspace of V.We discuss concrete upper bounds on $W_N(S(\mathcal{Y}))$ in more detail in Section 5. The aim of RBMs is to construct the spaces $U_{\tilde{\epsilon}}^{\mathrm{rb}}$ in such a way that the quantity $\sup_{y\in\mathcal{Y}} \operatorname{dist}\left(u_y, U_{\tilde{\epsilon}}^{\mathrm{rb}}\right)$ is close to $W_{d(\tilde{\epsilon})}\left(S(\mathcal{Y})\right)$.

The identification of the basis vectors $(\psi_i)_{i=1}^{d(\tilde{\epsilon})}$ of $U_{\tilde{\epsilon}}^{\mathrm{rb}}$ usually happens in an *offline phase* in which one has considerable computational resources available and which is usually based on the determination of high-fidelity discretizations of samples of the parameter set $\mathcal{Y}$. The most common methods are based on (weak) greedy procedures (see for instance [57, Chapter 7] and the references therein) or proper orthogonal decompositions (see for instance [57, Chapter 6] and the references therein). In the last step, an orthogonalization procedure (such as a Gram-Schmidt process) is performed to obtain an orthonormal set of basis vectors $(\psi_i)_{i=1}^{d(\tilde{\epsilon})}$.

Afterward, in the *online phase*, one assembles for a given input y the corresponding low-dimensional stiffness matrices and vectors and determines the Galerkin solution by solving a low-dimensional system of linear equations. To ensure an efficient implementation of the online phase, a common assumption which we do *not* require in this paper is the *affine decomposition* of (2.1), which means that there exist $Q_b, Q_f \in \mathbb{N}$, parameter-independent bilinear forms $b^q\colon \mathcal{H}\times\mathcal{H}\to\mathbb{R}$, maps $\theta_q\colon \mathcal{Y}\to\mathbb{R}$ for $q=1,...,Q_b$, parameter-independent $f^{q'}\in\mathcal{H}^*$ as well as maps $\theta^{q'}\colon \mathcal{Y}\to\mathbb{R}$ for $q'=1,...,Q_f$ such that

$$b_y = \sum_{q=1}^{Q_b} \theta_q(y) b^q, \qquad \text{as well as} \qquad f_y = \sum_{q'=1}^{Q_f} \theta^{q'}(y) f^{q'}, \qquad \text{for all } y\in\mathcal{Y}. \tag{2.6}$$

As has been pointed out in [57, Chapter 5.7], in principal three types of reduced bases generated by RBMs have been established in the literature - the *Lagrange reduced basis*, the *Hermite reduced basis* and the *Taylor reduced basis*. While the most common type, the Lagrange RB, consists of orthonormalized versions of high-fidelity *snapshots* $u^{\mathrm{h}}\left(y^1\right)\approx u\left(y^1\right),...,u^{\mathrm{h}}\left(y^n\right)\approx u\left(y^n\right)$, Hermite RBs consist of snapshots $u^{\mathrm{h}}\left(y^1\right)\approx u\left(y^1\right),...,u^{\mathrm{h}}\left(y^n\right)\approx u\left(y^n\right)$, as well as their first partial derivatives $\frac{\partial u^{\mathrm{h}}}{\partial y_i}(y^j)\approx\frac{\partial u}{\partial y_i}(y^j), i=1,...,p,\ j=1,...,n$, whereas Taylor RBs are built of derivatives of the form $\frac{\partial^k u^{\mathrm{h}}}{\partial y_i^k}(\overline{y})\approx\frac{\partial^k u}{\partial y_i^k}(\overline{y}),\ i=1,...,p,\ k=0,...,n-1$ around a given expansion point $\overline{y}\in\mathcal{Y}$. In this paper, we will later assume that there exist small RBs $(\psi_i)_{i=1}^{d(\tilde{\epsilon})}$ generated by *arbitrary* linear combinations of the high-fidelity elements $(\varphi_i)_{i=1}^{D}$. Note that all types of RBs discussed above satisfy this assumption.

The next statement gives a (generally sharp) upper bound which relates the possibility of constructing small snapshot RBs directly to the Kolmogorov N-width.

Theorem 2.4 ([8, Theorem 4.1.])**.** *Let $N\in\mathbb{N}$. For a compact subset X of a normed space V, define the* inner N-width of X *by*

$$\overline{W}_N(X) \coloneqq \inf_{V_N\in\mathcal{M}_N} \sup_{x\in X} \operatorname{dist}\left(x, V_N\right),$$

where $\mathcal{M}_N \coloneqq \left\{V_N \subset^{\mathrm{s}} V\colon\ V_N = \operatorname{span}\left(x_i\right)_{i=1}^N, x_1,...,x_N\in X\right\}$. *Then*

$$\overline{W}_N(X) \le (N+1) W_N(X).$$

Translated into our framework, Theorem 2.4 states that for every $N\in\mathbb{N}$, there exist solutions $u^{\mathrm{h}}(y^i)\approx u(y^i),\ i=1,...,N$ of (2.1) such that

$$\sup_{y\in\mathcal{Y}} \inf_{w\in\operatorname{span}(u^{\mathrm{h}}(y^i))_{i=1}^N} \|u_y - w\|_{\mathcal{H}} \le (N+1) W_N(S(\mathcal{Y})).$$

Remark 2.5. *We note that this bound is sharp for general X, V. However, it is not necessarily optimal for special instances of $S(\mathcal{Y})$. If, for instance, $W_N(S(\mathcal{Y}))$ decays polynomially, then $\overline{W}_N(S(\mathcal{Y}))$ decays with the same rate (see [8, Theorem 3.1.]). Moreover, if $W_N(S(\mathcal{Y}))\le Ce^{-cN^\beta}$ for some $c, C, \beta>0$ then by [18, Corollary 3.3 (iii)] we have $\overline{W}_N(S(\mathcal{Y}))\le \tilde{C}e^{-\tilde{c}N^\beta}$ for some $\tilde{c},\tilde{C}>0$.*

Taking the discussion above as a justification, we assume from now on that for every $\tilde{\epsilon} \geq \hat{\epsilon}$ there exists a RB space $U_{\tilde{\epsilon}}^{\mathrm{rb}} = \operatorname{span}(\psi_i)_{i=1}^{d(\tilde{\epsilon})}$, which fulfills (2.5), where the linearly independent basis vectors $(\psi_i)_{i=1}^{d(\tilde{\epsilon})}$ are linear combinations of the high-fidelity basis vectors $(\varphi_i)_{i=1}^{D}$ in the sense that there exists a transformation matrix $\mathbf{V}_{\tilde{\epsilon}} \in \mathbb{R}^{D \times d(\tilde{\epsilon})}$ such that

$$(\psi_i)_{i=1}^{d(\tilde{\epsilon})} = \left(\sum_{j=1}^{D} (\mathbf{V}_{\tilde{\epsilon}})_{j,i} \varphi_j \right)_{i=1}^{d(\tilde{\epsilon})}$$

and where $d(\tilde{\epsilon}) \ll D$ is chosen to be as small as possible, at least fulfilling $\operatorname{dist}\left(S(\mathcal{Y}), U_{\tilde{\epsilon}}^{\mathrm{rb}}\right) \leq \overline{W}_{d(\tilde{\epsilon})}(S(\mathcal{Y}))$. In addition, we assume that the vectors $(\psi_i)_{i=1}^{d(\tilde{\epsilon})}$ form an orthonormal system in $\mathcal{H}$, which is equivalent to the fact that the columns of $\mathbf{G}^{1/2}\mathbf{V}_{\tilde{\epsilon}}$ are orthonormal (see [57, Remark 4.1]). This in turn implies

$$\left\| \mathbf{G}^{1/2} \mathbf{V}_{\tilde{\epsilon}} \right\|_2 = 1, \quad \text{for all } \tilde{\epsilon} \geq \hat{\epsilon} \tag{2.7}$$

as well as

$$\left\| \sum_{i=1}^{d(\tilde{\epsilon})} \mathbf{c}_i \psi_i \right\|_{\mathcal{H}} = |\mathbf{c}|, \quad \text{for all } \mathbf{c} \in \mathbb{R}^{d(\tilde{\epsilon})}. \tag{2.8}$$

For the underlying discretization matrix, one can demonstrate (see for instance [57, Section 3.4.1]) that

$$\mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}} \coloneqq (b_y(\psi_j, \psi_i))_{i,j=1}^{d(\tilde{\epsilon})} = \mathbf{V}_{\tilde{\epsilon}}^T \mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{h}} \mathbf{V}_{\tilde{\epsilon}} \in \mathbb{R}^{d(\tilde{\epsilon}) \times d(\tilde{\epsilon})}, \quad \text{for all } y \in \mathcal{Y}.$$

Moreover, due to the symmetry and the coercivity of the underlying bilinear forms combined with the orthonormality of the basis vectors $(\psi_i)_{i=1}^{d(\tilde{\epsilon})}$, one can show (see for instance [57, Remark 3.5]) that

$$C_{\mathrm{coer}} \leq \left\| \mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}} \right\|_2 \leq C_{\mathrm{cont}}, \quad \text{as well as} \quad \frac{1}{C_{\mathrm{cont}}} \leq \left\| \left(\mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}} \right)^{-1} \right\|_2 \leq \frac{1}{C_{\mathrm{coer}}}, \quad \text{for all } y \in \mathcal{Y}, \tag{2.9}$$

implying that the condition number of the stiffness matrix with respect to the RB remains bounded independently of y and the dimension $d(\tilde{\epsilon})$. Additionally, the discretized right-hand side with respect to the RB is given by

$$\mathbf{f}_{y,\tilde{\epsilon}}^{\mathrm{rb}} \coloneqq (f_y(\psi_i))_{i=1}^{d(\tilde{\epsilon})} = \mathbf{V}_{\tilde{\epsilon}}^T \mathbf{f}_{y,\tilde{\epsilon}}^{\mathrm{h}} \in \mathbb{R}^{d(\tilde{\epsilon})}$$

and, by the Bessel inequality, we have that $\left| \mathbf{f}_{y,\tilde{\epsilon}}^{\mathrm{rb}} \right| \leq \|f_y\|_{\mathcal{H}^*} \leq C_{\mathrm{rhs}}$. Moreover, let

$$\mathbf{u}_{y,\tilde{\epsilon}}^{\mathrm{rb}} \coloneqq \left(\mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}} \right)^{-1} \mathbf{f}_{y,\tilde{\epsilon}}^{\mathrm{rb}}$$

be the coefficient vector of the Galerkin solution with respect to the RB space. Then, the Galerkin solution $u_{y,\tilde{\epsilon}}^{\mathrm{rb}}$ can be written as

$$u_{y,\tilde{\epsilon}}^{\mathrm{rb}} = \sum_{i=1}^{d(\tilde{\epsilon})} \left(\mathbf{u}_{y,\tilde{\epsilon}}^{\mathrm{rb}} \right)_i \psi_i = \sum_{j=1}^{D} \left(\mathbf{V}_{\tilde{\epsilon}} \mathbf{u}_{y,\tilde{\epsilon}}^{\mathrm{rb}} \right)_j \varphi_j,$$

i.e.

$$\tilde{\mathbf{u}}_{y,\tilde{\epsilon}}^{\mathrm{h}} \coloneqq \mathbf{V}_{\tilde{\epsilon}} \mathbf{u}_{y,\tilde{\epsilon}}^{\mathrm{rb}} \in \mathbb{R}^D$$

is the coefficient vector of the RB solution if expanded with respect to the high-fidelity basis $(\varphi_i)_{i=1}^{D}$. Finally, as in Equation 2.3, we obtain

$$\sup_{y \in \mathcal{Y}} \left\| u_y - u_{y,\tilde{\epsilon}}^{\mathrm{rb}} \right\|_{\mathcal{H}} \leq \sup_{y \in \mathcal{Y}} \frac{C_{\mathrm{cont}}}{C_{\mathrm{coer}}} \inf_{w \in U_{\tilde{\epsilon}}^{\mathrm{rb}}} \|u_y - w\|_{\mathcal{H}} \leq \frac{C_{\mathrm{cont}}}{C_{\mathrm{coer}}} \tilde{\epsilon}.$$

In the following sections, we will emulate RBMs with NNs by showing that we are able to construct NNs which approximate the maps $\mathbf{u}^{\mathrm{rb}}_{\cdot,\tilde{\epsilon}}, \tilde{\mathbf{u}}^{\mathrm{h}}_{\cdot,\tilde{\epsilon}}$ such that their complexity depends only on the size of the reduced basis and at most linearly on D. The key ingredient will be the construction of small NNs implementing an approximate matrix inversion based on Richardson iterations in Section 3. In Section 4, we then proceed with building the NNs the realizations of which approximate the maps $\mathbf{u}^{\mathrm{rb}}_{\cdot,\tilde{\epsilon}}, \tilde{\mathbf{u}}^{\mathrm{h}}_{\cdot,\tilde{\epsilon}}$, respectively.

3 Neural Network Calculus

The goal of this chapter is to emulate the matrix inversion by NNs. In Section 3.1, we introduce some basic notions connected to NNs as well as some basic operations one can perform with these. In Section 3.2, we state a result which shows the existence of NNs the ReLU-realizations of which take a matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$, $\|\mathbf{A}\|_2 < 1$ as their input and calculate an approximation of $(\mathbf{Id}_{\mathbb{R}^d} - \mathbf{A})^{-1}$ based on its Neumann series expansion. The associated proofs can be found in Appendix A.

3.1 Basic Definitions and Operations

We start by introducing a formal definition of a NN. Afterward, we introduce several operations, such as parallelization and concatenation that can be used to assemble complex NNs out of simpler ones. Unless stated otherwise we follow the notion of [53] where most of this formal framework was introduced. First, we introduce a terminology for NNs that allows us to differentiate between a NN as a family of weights and the function implemented by the NN. This implemented function will be called the realization of the NN.

Definition 3.1. *Let $n, L \in \mathbb{N}$. A* NN Φ with input dimension $\dim_{\mathrm{in}}(\Phi) \coloneqq n$ and L layers *is a sequence of matrix-vector tuples*

$$\Phi = \big((\mathbf{A}_1, \mathbf{b}_1), (\mathbf{A}_2, \mathbf{b}_2), \dots, (\mathbf{A}_L, \mathbf{b}_L)\big),$$

where $N_0 = n$ and $N_1, \dots, N_L \in \mathbb{N}$, and where each $\mathbf{A}_\ell$ is an $N_\ell \times N_{\ell-1}$ matrix, and $\mathbf{b}_\ell \in \mathbb{R}^{N_\ell}$.

If Φ is a NN as above, $K \subset \mathbb{R}^n$, and if $\varrho\colon \mathbb{R} \to \mathbb{R}$ is arbitrary, then we define the associated realization of Φ with activation function ϱ over K *(in short, the* ϱ-realization of Φ over K*) as the map* $\mathrm{R}^K_\varrho(\Phi)\colon K \to \mathbb{R}^{N_L}$ *such that*

$$\mathrm{R}^K_\varrho(\Phi)(\mathbf{x}) = \mathbf{x}_L,$$

where $\mathbf{x}_L$ results from the following scheme:

$$\begin{aligned} &\mathbf{x}_0 \coloneqq \mathbf{x}, \\ &\mathbf{x}_\ell \coloneqq \varrho(\mathbf{A}_\ell\, \mathbf{x}_{\ell-1} + \mathbf{b}_\ell), \qquad \text{for } \ell = 1, \dots, L-1, \\ &\mathbf{x}_L \coloneqq \mathbf{A}_L\, \mathbf{x}_{L-1} + \mathbf{b}_L, \end{aligned}$$

and where ϱ acts componentwise, that is, $\varrho(\mathbf{v}) = (\varrho(\mathbf{v}_1), \dots, \varrho(\mathbf{v}_m))$ for any $\mathbf{v} = (\mathbf{v}_1, \dots, \mathbf{v}_m) \in \mathbb{R}^m$.

We call $N(\Phi) \coloneqq n + \sum_{j=1}^{L} N_j$ the number of neurons of the NN Φ *and $L = L(\Phi)$ the* number of layers*. For $\ell \le L$ we call $M_\ell(\Phi) \coloneqq \|\mathbf{A}_\ell\|_0 + \|\mathbf{b}_\ell\|_0$ the* number of weights in the ℓ-th layer *and we define $M(\Phi) \coloneqq \sum_{\ell=1}^{L} M_\ell(\Phi)$, which we call the* number of weights of Φ*. Moreover, we refer to* $\dim_{\mathrm{out}}(\Phi) \coloneqq N_L$ *as* the output dimension *of Φ.*

First of all, we note that it is possible to concatenate two NNs in the following way.

Definition 3.2. *Let $L_1, L_2 \in \mathbb{N}$ and let $\Phi^1 = \big((\mathbf{A}^1_1, \mathbf{b}^1_1), \dots, (\mathbf{A}^1_{L_1}, \mathbf{b}^1_{L_1})\big), \Phi^2 = \big((\mathbf{A}^2_1, \mathbf{b}^2_1), \dots, (\mathbf{A}^2_{L_2}, \mathbf{b}^2_{L_2})\big)$ be two NNs such that the input layer of Φ^1 has the same dimension as the output layer of Φ^2. Then, $\Phi^1 \bullet \Phi^2$ denotes the following $L_1 + L_2 - 1$ layer NN:*

$$\Phi^1 \bullet \Phi^2 \coloneqq \big((\mathbf{A}^2_1, \mathbf{b}^2_1), \dots, (\mathbf{A}^2_{L_2-1}, \mathbf{b}^2_{L_2-1}), (\mathbf{A}^1_1 \mathbf{A}^2_{L_2}, \mathbf{A}^1_1 \mathbf{b}^2_{L_2} + \mathbf{b}^1_1), (\mathbf{A}^1_2, \mathbf{b}^1_2), \dots, (\mathbf{A}^1_{L_1}, \mathbf{b}^1_{L_1})\big).$$

We call $\Phi^1 \bullet \Phi^2$ the concatenation of Φ^1, Φ^2*.*

In general, there is no bound on $M(\Phi^1 \bullet \Phi^2)$ that is linear in $M(\Phi^1)$ and $M(\Phi^2)$. For the remainder of the paper, let ϱ be given by the ReLU activation function, i.e., $\varrho(x) = \max\{x, 0\}$ for $x \in \mathbb{R}$. We will see in the following, that we are able to introduce an alternative concatenation which helps us to control the number of non-zero weights. Towards this goal, we give the following result which shows that we can construct NNs the ReLU-realization of which is the identity function on $\mathbb{R}^n$.

Lemma 3.3. *For any $L \in \mathbb{N}$ there exists a NN $\Phi^{\mathbf{Id}}_{n,L}$ with input dimension n, output dimension n and at most $2nL$ non-zero, $\{-1, 1\}$-valued weights such that*

$$\mathrm{R}^{\mathbb{R}^n}_{\varrho}\left(\Phi^{\mathbf{Id}}_{n,L}\right) = \mathbf{Id}_{\mathbb{R}^n}.$$

We now introduce the sparse concatenation of two NNs.

Definition 3.4. *Let Φ^1, Φ^2 be two NNs such that the output dimension of Φ^2 and the input dimension of Φ^1 equal $n \in \mathbb{N}$. Then the* sparse concatenation of Φ^1 and Φ^2 *is defined as*

$$\Phi^1 \odot \Phi^2 \coloneqq \Phi^1 \bullet \Phi^{\mathbf{Id}}_{n,1} \bullet \Phi^2.$$

We will see later in Lemma 3.6 the properties of the sparse concatenation of NNs. We proceed with the second operation that we can perform with NNs. This operation is called parallelization.

Definition 3.5 ([53, 21])**.** *Let $\Phi^1, ..., \Phi^k$ be NNs which have equal input dimension such that there holds $\Phi^i = \left((\mathbf{A}^i_1, \mathbf{b}^i_1), ..., (\mathbf{A}^i_L, \mathbf{b}^i_L)\right)$ for some $L \in \mathbb{N}$. Then, we define the* parallelization of $\Phi^1, ..., \Phi^k$ *by*

$$\mathrm{P}\left(\Phi^1, ..., \Phi^k\right) \coloneqq \left(\left(\begin{pmatrix} \mathbf{A}^1_1 & & & \\ & \mathbf{A}^2_1 & & \\ & & \ddots & \\ & & & \mathbf{A}^k_1 \end{pmatrix}, \begin{pmatrix} \mathbf{b}^1_1 \\ \mathbf{b}^2_1 \\ \vdots \\ \mathbf{b}^k_1 \end{pmatrix}\right), ..., \left(\begin{pmatrix} \mathbf{A}^1_L & & & \\ & \mathbf{A}^2_L & & \\ & & \ddots & \\ & & & \mathbf{A}^k_L \end{pmatrix}, \begin{pmatrix} \mathbf{b}^1_L \\ \mathbf{b}^2_L \\ \vdots \\ \mathbf{b}^k_L \end{pmatrix}\right)\right).$$

Now, let Φ be a NN and $L \in \mathbb{N}$ such that $L(\Phi) \leq L$. Then, define the NN

$$E_L(\Phi) \coloneqq \begin{cases} \Phi, & \text{if } L(\Phi) = L, \\ \Phi^{\mathbf{Id}}_{\dim_{\mathrm{out}}(\Phi), L - L(\Phi)} \odot \Phi, & \text{if } L(\Phi) < L. \end{cases}$$

Finally, let $\tilde{\Phi}^1, ..., \tilde{\Phi}^k$ be NNs which have the same input dimension and let

$$\tilde{L} \coloneqq \max\left\{L\left(\tilde{\Phi}^1\right), ..., L\left(\tilde{\Phi}^k\right)\right\}.$$

Then, we define

$$\mathrm{P}\left(\tilde{\Phi}^1, ..., \tilde{\Phi}^k\right) \coloneqq \mathrm{P}\left(E_{\tilde{L}}\left(\tilde{\Phi}^1\right), ..., E_{\tilde{L}}\left(\tilde{\Phi}^k\right)\right).$$

We call $\mathrm{P}\left(\tilde{\Phi}^1, ..., \tilde{\Phi}^k\right)$ the parallelization of $\tilde{\Phi}^1, ..., \tilde{\Phi}^k$.

The following lemma was established in [21, Lemma 5.4] and examines the properties of the sparse concatenation as well as of the parallelization of NNs.

Lemma 3.6 ([21])**.** *Let $\Phi^1, ..., \Phi^k$ be NNs.*

(a) If the input dimension of Φ^1, which shall be denoted by n_1, equals the output dimension of Φ^2, and n_2 is the input dimension of Φ^2, then

$$\mathrm{R}^{\mathbb{R}^{n_1}}_{\varrho}\left(\Phi^1\right) \circ \mathrm{R}^{\mathbb{R}^{n_2}}_{\varrho}\left(\Phi^2\right) = \mathrm{R}^{\mathbb{R}^{n_2}}_{\varrho}\left(\Phi^1 \odot \Phi^2\right)$$

and

(i) $L\left(\Phi^1 \odot \Phi^2\right) \leq L\left(\Phi^1\right) + L\left(\Phi^2\right)$,

(ii) $M\left(\Phi^1 \odot \Phi^2\right) \leq M\left(\Phi^1\right) + M\left(\Phi^2\right) + M_1\left(\Phi^1\right) + M_{L(\Phi^2)}\left(\Phi^2\right) \leq 2M\left(\Phi^1\right) + 2M\left(\Phi^2\right)$,

(iii) $M_1\left(\Phi^1 \odot \Phi^2\right) = M_1\left(\Phi^2\right)$,

(iv) $M_{L(\Phi^1 \odot \Phi^2)}\left(\Phi^1 \odot \Phi^2\right) = M_{L(\Phi^1)}\left(\Phi^1\right)$.

(b) If the input dimension of Φ^i, denoted by n, equals the input dimension of Φ^j, for all i, j, then for the NN $\mathrm{P}\left(\Phi^1, \Phi^2, ..., \Phi^k\right)$ and all $\mathbf{x}_1, \ldots \mathbf{x}_k \in \mathbb{R}^n$ we have

$$\mathrm{R}_\varrho^{\mathbb{R}^n}\left(\mathrm{P}\left(\Phi^1, \Phi^2, ..., \Phi^k\right)\right)(\mathbf{x}_1, \ldots, \mathbf{x}_k) = \left(\mathrm{R}_\varrho^{\mathbb{R}^n}\left(\Phi^1\right)(\mathbf{x}_1), \mathrm{R}_\varrho^{\mathbb{R}^n}\left(\Phi^2\right)(\mathbf{x}_2), \ldots, \mathrm{R}_\varrho^{\mathbb{R}^n}\left(\Phi^k\right)(\mathbf{x}_k)\right)$$

as well as

(i) $L\left(\mathrm{P}\left(\Phi_1, \Phi^2, ..., \Phi^k\right)\right) = \max_{i=1,...,k} L\left(\Phi^i\right)$,

(ii) $M\left(\mathrm{P}\left(\Phi^1, \Phi^2, ..., \Phi^k\right)\right) \leq 2\left(\sum_{i=1}^k M\left(\Phi^i\right)\right) + 4\left(\sum_{i=1}^k \dim_{\mathrm{out}}\left(\Phi^i\right)\right) \max_{i=1,...,k} L\left(\Phi^i\right)$,

(iii) $M\left(\mathrm{P}\left(\Phi^1, \Phi^2, ..., \Phi^k\right)\right) = \sum_{i=1}^k M\left(\Phi^i\right)$, *if* $L\left(\Phi^1\right) = L\left(\Phi^2\right) = \ldots = L\left(\Phi^k\right)$,

(iv) $M_1\left(\mathrm{P}\left(\Phi^1, \Phi^2, ..., \Phi^k\right)\right) = \sum_{i=1}^k M_1\left(\Phi^i\right)$,

(v) $M_{L(\mathrm{P}(\Phi^1, \Phi^2, ..., \Phi^k))}\left(\mathrm{P}\left(\Phi^1, \Phi^2, ..., \Phi^k\right)\right) \leq \sum_{i=1}^k \max\left\{2\dim_{\mathrm{out}}\left(\Phi^i\right), M_{L(\Phi^i)}\left(\Phi^i\right)\right\}$,

(vi) $M_{L(\mathrm{P}(\Phi^1, \Phi^2, ..., \Phi^k))}\left(\mathrm{P}\left(\Phi^1, \Phi^2, ..., \Phi^k\right)\right) = \sum_{i=1}^k M_{L(\Phi^i)}\left(\Phi^i\right)$, *if* $L\left(\Phi^1\right) = L\left(\Phi^2\right) = \ldots = L\left(\Phi^k\right)$.

3.2 A Neural Network Based Approach Towards Matrix Inversion

The goal of this subsection is to emulate the inversion of square matrices by realizations of NNs which are comparatively small in size. In particular, Theorem 3.8 shows that, for $d \in \mathbb{N}$, $\epsilon \in (0, 1/4)$, and $\delta \in (0, 1)$, we are able to efficiently construct NNs $\Phi_{\mathrm{inv};\epsilon}^{1-\delta,d}$ the ReLU-realization of which approximates the map

$$\left\{\mathbf{A} \in \mathbb{R}^{d\times d} \colon \|\mathbf{A}\|_2 \leq 1 - \delta\right\} \to \mathbb{R}^{d\times d},\ \mathbf{A} \mapsto \left(\mathbf{Id}_{\mathbb{R}^d} - \mathbf{A}\right)^{-1} = \sum_{k=0}^{\infty} \mathbf{A}^k$$

up to an $\|\cdot\|_2$- error of ϵ.

To stay in the classical NN setting, we employ vectorized matrices in the remainder of this paper. Let $\mathbf{A} \in \mathbb{R}^{d\times l}$. We write

$$\mathbf{vec}(\mathbf{A}) \coloneqq (\mathbf{A}_{1,1}, ..., \mathbf{A}_{d,1}, ..., \mathbf{A}_{1,l}, ..., \mathbf{A}_{d,l})^T \in \mathbb{R}^{dl}.$$

Moreover, for a vector $\mathbf{v} = (\mathbf{v}_{1,1}, ..., \mathbf{v}_{d,1}, ..., \mathbf{v}_{1,d}, ..., \mathbf{v}_{d,l})^T \in \mathbb{R}^{dl}$ we set

$$\mathbf{matr}(\mathbf{v}) \coloneqq (\mathbf{v}_{i,j})_{i=1,...,d,\ j=1,...,l} \in \mathbb{R}^{d\times l}.$$

In addition, for $d, n, l \in \mathbb{N}$ and $Z > 0$ we set

$$K_{d,n,l}^Z \coloneqq \left\{(\mathbf{vec}(\mathbf{A}), \mathbf{vec}(\mathbf{B})) \colon (\mathbf{A}, \mathbf{B}) \in \mathbb{R}^{d\times n} \times \mathbb{R}^{n\times l},\ \|\mathbf{A}\|_2, \|\mathbf{B}\|_2 \leq Z\right\}$$

as well as

$$K_d^Z \coloneqq \left\{\mathbf{vec}(\mathbf{A}) \colon \mathbf{A} \in \mathbb{R}^{d\times d},\ \|\mathbf{A}\|_2 \leq Z\right\}.$$

The basic ingredient for the construction of NNs emulating a matrix inversion is the following result about NNs emulating the multiplication of two matrices.

Proposition 3.7. *Let $d, n, l \in \mathbb{N}$, $\epsilon \in (0, 1)$, $Z > 0$. There exists a NN $\Phi_{\mathrm{mult};\epsilon}^{Z,d,n,l}$ with $n \cdot (d + l)$- dimensional input, dl-dimensional output such that, for an absolute constant $C_{\mathrm{mult}} > 0$, the following properties are fulfilled:*

(i) $L\left(\Phi_{\mathrm{mult};\tilde{\epsilon}}^{Z,d,n,l}\right) \leq C_{\mathrm{mult}} \cdot \left(\log_2\left(1/\epsilon\right) + \log_2\left(n\sqrt{dl}\right) + \log_2\left(\max\left\{1, Z\right\}\right)\right)$,

(ii) $M\left(\Phi_{\mathrm{mult};\tilde{\epsilon}}^{Z,d,n,l}\right) \leq C_{\mathrm{mult}} \cdot \left(\log_2\left(1/\epsilon\right) + \log_2\left(n\sqrt{dl}\right) + \log_2\left(\max\left\{1, Z\right\}\right)\right) dnl$,

(iii) $M_1\left(\Phi_{\mathrm{mult};\tilde{\epsilon}}^{Z,d,n,l}\right) \leq C_{\mathrm{mult}} dnl$, *as well as* $M_{L\left(\Phi_{\mathrm{mult};\tilde{\epsilon}}^{Z,d,n,l}\right)}\left(\Phi_{\mathrm{mult};\tilde{\epsilon}}^{Z,d,n,l}\right) \leq C_{\mathrm{mult}} dnl$,

(iv) $\sup_{(\mathbf{vec}(\mathbf{A}),\mathbf{vec}(\mathbf{B}))\in K_{d,n,l}^{Z}} \left\| \mathbf{AB} - \mathbf{matr}\left(\mathrm{R}_{\varrho}^{K_{d,n,l}^{Z}}\left(\Phi_{\mathrm{mult};\epsilon}^{Z,d,n,l}\right)\left(\mathbf{vec}(\mathbf{A}), \mathbf{vec}(\mathbf{B})\right)\right)\right\|_2 \leq \epsilon$,

(v) for any $(\mathbf{vec}(\mathbf{A}), \mathbf{vec}(\mathbf{B})) \in K_{d,n,l}^{Z}$ *we have*

$$\left\|\mathbf{matr}\left(\mathrm{R}_{\varrho}^{K_{d,n,l}^{Z}}\left(\Phi_{\mathrm{mult};\tilde{\epsilon}}^{Z,d,n,l}\right)\left(\mathbf{vec}(\mathbf{A}), \mathbf{vec}(\mathbf{B})\right)\right)\right\|_2 \leq \epsilon + \|\mathbf{A}\|_2\|\mathbf{B}\|_2 \leq \epsilon + Z^2 \leq 1 + Z^2.$$

Based on Proposition 3.7, we construct in Appendix A.2 NNs emulating the map $\mathbf{A} \mapsto \mathbf{A}^k$ for square matrices $\mathbf{A}$ and $k \in \mathbb{N}$. This construction is then used to prove the following result.

Theorem 3.8. *For* $\epsilon, \delta \in (0,1)$ *define*

$$m(\epsilon, \delta) \coloneqq \left\lceil \frac{\log_2\left(0.5\epsilon\delta\right)}{\log_2(1-\delta)} \right\rceil.$$

There exists a universal constant $C_{\mathrm{inv}} > 0$ *such that for every* $d \in \mathbb{N}$, $\epsilon \in (0, 1/4)$ *and every* $\delta \in (0,1)$ *there exists a NN* $\Phi_{\mathrm{inv};\epsilon}^{1-\delta,d}$ *with* d^2*-dimensional input,* d^2*-dimensional output and the following properties:*

(i) $L\left(\Phi_{\mathrm{inv};\epsilon}^{1-\delta,d}\right) \leq C_{\mathrm{inv}} \log_2\left(m(\epsilon,\delta)\right) \cdot \left(\log_2\left(1/\epsilon\right) + \log_2\left(m(\epsilon,\delta)\right) + \log_2(d)\right)$,

(ii) $M\left(\Phi_{\mathrm{inv};\epsilon}^{1-\delta,d}\right) \leq C_{\mathrm{inv}} m(\epsilon,\delta) \log_2^2(m(\epsilon,\delta)) d^3 \cdot \left(\log_2\left(1/\epsilon\right) + \log_2\left(m(\epsilon,\delta)\right) + \log_2(d)\right)$,

(iii) $\sup_{\mathbf{vec}(\mathbf{A})\in K_d^{1-\delta}} \left\| (\mathbf{Id}_{\mathbb{R}^d} - \mathbf{A})^{-1} - \mathbf{matr}\left(\mathrm{R}_{\varrho}^{K_d^{1-\delta}}\left(\Phi_{\mathrm{inv};\epsilon}^{1-\delta,d}\right)\left(\mathbf{vec}(\mathbf{A})\right)\right)\right\|_2 \leq \epsilon$,

(iv) for any $\mathbf{vec}(\mathbf{A}) \in K_d^{1-\delta}$ *we have*

$$\left\|\mathbf{matr}\left(\mathrm{R}_{\varrho}^{K_d^{1-\delta}}\left(\Phi_{\mathrm{inv};\epsilon}^{1-\delta,d}\right)\left(\mathbf{vec}(\mathbf{A})\right)\right)\right\|_2 \leq \epsilon + \left\|(\mathbf{Id}_{\mathbb{R}^d} - \mathbf{A})^{-1}\right\|_2 \leq \epsilon + \frac{1}{1 - \|\mathbf{A}\|_2} \leq \epsilon + \frac{1}{\delta}.$$

Remark 3.9. *In the proof of Theorem 3.8, we approximate the function mapping a matrix to its inverse via the Neumann series and then emulate this construction by NNs. There certainly exist alternative approaches to approximating this inversion function, such as, for example, via Chebyshev matrix polynomials (for an introduction of Chebyshev polynomials, see for instance [65, Chapter 8.2]). In fact, approximation by Chebyshev matrix polynomials is more efficient in terms of the degree of the polynomials required to reach a certain approximation accuracy. However, emulation of Chebyshev matrix polynomials by NNs either requires larger networks than that of monomials or, if they are represented in a monomial basis, coefficients that grow exponentially with the polynomial degree. In the end, the advantage of a smaller degree in the approximation through Chebyshev matrix polynomials does not seem to set off the drawbacks described before.*

4 Neural Networks and Solutions of PDEs Using Reduced Bases

In this section, we invoke the estimates for the approximate matrix inversion from Section 3.2 to approximate the parameter-dependent solution of parametric PDEs by NNs. In other words, for $\tilde{\epsilon} \geq \hat{\epsilon}$, we construct NNs approximating the maps

$$\mathcal{Y} \to \mathbb{R}^D\colon \quad y \mapsto \tilde{\mathbf{u}}^{\mathrm{h}}_{y,\tilde{\epsilon}}, \text{ and } \mathcal{Y} \to \mathbb{R}^{d(\tilde{\epsilon})}\colon \quad y \mapsto \mathbf{u}^{\mathrm{rb}}_{y,\tilde{\epsilon}}.$$

Here, the sizes of the NNs essentially only depend on the approximation fidelity $\tilde{\epsilon}$ and the size $d(\tilde{\epsilon})$ of an appropriate RB, but are independent or at most linear in the dimension of the high-fidelity discretization D.

We start in Section 4.1 by constructing, under some general assumptions on the parametric problem, a NN emulating the maps $\tilde{\mathbf{u}}^{\mathrm{h}}_{\cdot,\tilde{\epsilon}}$ and $\mathbf{u}^{\mathrm{rb}}_{\cdot,\tilde{\epsilon}}$. In Section 4.2, we verify these assumptions on two examples.

4.1 Determining the Coefficients of the Solution

Next, we present constructions of NNs the ReLU-realizations of which approximate the maps $\tilde{\mathbf{u}}^{\mathrm{h}}_{\cdot,\tilde{\epsilon}}$ and $\mathbf{u}^{\mathrm{rb}}_{\cdot,\tilde{\epsilon}}$, respectively. In our main result of this subsection, the approximation error of the NN approximation $\tilde{\mathbf{u}}^{\mathrm{h}}_{\cdot,\tilde{\epsilon}}$ will be measured with respect to the $|\cdot|_{\mathbf{G}}$-norm since we can relate this norm directly to the norm on $\mathcal{H}$ via Equation (2.4). In contrast, the approximation error of the NN approximating $\mathbf{u}^{\mathrm{rb}}_{\cdot,\tilde{\epsilon}}$ will be measured with respect to the $|\cdot|$-norm due to Equation 2.8.

As already indicated earlier, the main ingredient of the following arguments is an application of the NN of Theorem 3.8 to the matrix $\mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}$. As a preparation, we show in Proposition B.1 in the appendix, that we can rescale the matrix $\mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}$ with a constant factor $\alpha \coloneqq (C_{\mathrm{coer}} + C_{\mathrm{cont}})^{-1}$ (in particular, independent of y and $d(\tilde{\epsilon})$) so that with $C_{\mathrm{coer}}\delta \coloneqq \alpha C_{\mathrm{coer}}$

$$\left\|\mathbf{Id}_{\mathbb{R}^{d(\tilde{\epsilon})}} - \alpha \mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}\right\|_2 \leq 1 - \delta < 1.$$

We will fix these values of α and δ for the remainder of the manuscript. Next, we state two abstract assumptions on the approximability of the map $\mathbf{B}^{\mathrm{rb}}_{\cdot,\tilde{\epsilon}}$ which we will later on specify when we consider concrete examples in Subsection 4.2.

Assumption 4.1. *We assume that, for any $\tilde{\epsilon} \geq \hat{\epsilon}, \epsilon > 0$, and for a corresponding RB $(\psi_i)_{i=1}^{d(\tilde{\epsilon})}$, there exists a NN $\Phi^{\mathbf{B}}_{\tilde{\epsilon},\epsilon}$ with p-dimensional input and $d(\tilde{\epsilon})^2$-dimensional output such that*

$$\sup_{y\in\mathcal{Y}} \left\|\alpha \mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}} - \mathbf{matr}\left(\mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{B}}_{\tilde{\epsilon},\epsilon}\right)(y)\right)\right\|_2 \leq \epsilon.$$

We set $B_M(\tilde{\epsilon},\epsilon) \coloneqq M\left(\Phi^{\mathbf{B}}_{\tilde{\epsilon},\epsilon}\right) \in \mathbb{N}$ and $B_L(\tilde{\epsilon},\epsilon) \coloneqq L\left(\Phi^{\mathbf{B}}_{\tilde{\epsilon},\epsilon}\right) \in \mathbb{N}$.

In addition to Assumption 4.1, we state the following assumption on the approximability of the map $\mathbf{f}^{\mathrm{rb}}_{\cdot,\tilde{\epsilon}}$.

Assumption 4.2. *We assume that for every $\tilde{\epsilon} \geq \hat{\epsilon}, \epsilon > 0$, and a corresponding RB $(\psi_i)_{i=1}^{d(\tilde{\epsilon})}$ there exists a NN $\Phi^{\mathbf{f}}_{\tilde{\epsilon},\epsilon}$ with p-dimensional input and $d(\tilde{\epsilon})$-dimensional output such that*

$$\sup_{y\in\mathcal{Y}} \left|\mathbf{f}^{\mathrm{rb}}_{y,\tilde{\epsilon}} - \mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{f}}_{\tilde{\epsilon},\epsilon}\right)(y)\right| \leq \epsilon.$$

We set $F_L(\tilde{\epsilon},\epsilon) \coloneqq L(\Phi^{\mathbf{f}}_{\tilde{\epsilon},\epsilon})$ and $F_M(\tilde{\epsilon},\epsilon) \coloneqq M\left(\Phi^{\mathbf{f}}_{\tilde{\epsilon},\epsilon}\right)$.

Now we are in a position to construct NNs the ReLU-realizations of which approximate the coefficient maps $\tilde{\mathbf{u}}^{\mathrm{h}}_{\cdot,\tilde{\epsilon}}, \mathbf{u}^{\mathrm{rb}}_{\cdot,\tilde{\epsilon}}$.

Theorem 4.3. *Let $\tilde{\epsilon} \geq \hat{\epsilon}$ and $\epsilon \in (0, \alpha/4 \cdot \min\{1, C_{\mathrm{coer}}\})$. Moreover, define $\epsilon' \coloneqq \epsilon/\max\{6, C_{\mathrm{rhs}}\}$, $\epsilon'' \coloneqq \epsilon/3 \cdot C_{\mathrm{coer}}$, $\epsilon''' \coloneqq 3/8 \cdot \epsilon' \alpha C^2_{\mathrm{coer}}$ and $\kappa \coloneqq 2\max\{1, C_{\mathrm{rhs}}, 1/C_{\mathrm{coer}}\}$. Additionally, assume that Assumption 4.1 and Assumption 4.2 hold. Then there exist NNs $\Phi^{\mathbf{u},\mathrm{rb}}_{\tilde{\epsilon},\epsilon}$ and $\Phi^{\mathbf{u},\mathrm{h}}_{\tilde{\epsilon},\epsilon}$ such that the following properties hold:*

(i) $\sup_{y\in\mathcal{Y}}\left|\mathbf{u}^{\mathrm{rb}}_{y,\tilde{\epsilon}}-\mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{u},\mathrm{rb}}_{\tilde{\epsilon},\epsilon}\right)(y)\right|\leq\epsilon$, *and* $\sup_{y\in\mathcal{Y}}\left|\tilde{\mathbf{u}}^{\mathrm{h}}_{y,\tilde{\epsilon}}-\mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{u},\mathrm{h}}_{\tilde{\epsilon},\epsilon}\right)(y)\right|_{\mathbf{G}}\leq\epsilon$,

(ii) there exists a constant $C^{\mathbf{u}}_{L}=C^{\mathbf{u}}_{L}(C_{\mathrm{coer}},C_{\mathrm{cont}},C_{\mathrm{rhs}})>0$ *such that*

$$L\left(\Phi^{\mathbf{u},\mathrm{rb}}_{\tilde{\epsilon},\epsilon}\right)\leq L\left(\Phi^{\mathbf{u},\mathrm{h}}_{\tilde{\epsilon},\epsilon}\right)$$
$$\leq C^{\mathbf{u}}_{L}\max\left\{\log_2(\log_2(1/\epsilon))\left(\log_2(1/\epsilon)+\log_2(\log_2(1/\epsilon))+\log_2(d(\tilde{\epsilon}))\right)+B_L(\tilde{\epsilon},\epsilon'''),F_L\left(\tilde{\epsilon},\epsilon''\right)\right\},$$

(iii) there exists a constant $C^{\mathbf{u}}_{M}=C^{\mathbf{u}}_{M}(C_{\mathrm{coer}},C_{\mathrm{cont}},C_{\mathrm{rhs}})>0$ *such that*

$$M\left(\Phi^{\mathbf{u},\mathrm{rb}}_{\tilde{\epsilon},\epsilon}\right)\leq C^{\mathbf{u}}_{M}d(\tilde{\epsilon})^2\cdot\Big(d(\tilde{\epsilon})\log_2(1/\epsilon)\log_2^2(\log_2(1/\epsilon))\big(\log_2(1/\epsilon)+\log_2(\log_2(1/\epsilon))+\log_2(d(\tilde{\epsilon}))\big)...$$
$$...+B_L(\tilde{\epsilon},\epsilon''')+F_L\left(\tilde{\epsilon},\epsilon''\right)\Big)+2B_M(\tilde{\epsilon},\epsilon''')+F_M\left(\tilde{\epsilon},\epsilon''\right),$$

(iv) $M\left(\Phi^{\mathbf{u},\mathrm{h}}_{\tilde{\epsilon},\epsilon}\right)\leq 2Dd(\tilde{\epsilon})+2M\left(\Phi^{\mathbf{u},\mathrm{rb}}_{\tilde{\epsilon},\epsilon}\right)$,

(v) $\sup_{y\in\mathcal{Y}}\left|\mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{u},\mathrm{rb}}_{\tilde{\epsilon},\epsilon}\right)(y)\right|\leq\kappa^2+\frac{\epsilon}{3}$, *and* $\sup_{y\in\mathcal{Y}}\left|\mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{u},\mathrm{h}}_{\tilde{\epsilon},\epsilon}\right)(y)\right|_{\mathbf{G}}\leq\kappa^2+\frac{\epsilon}{3}$.

Remark 4.4. *In the proof of Theorem 4.3 we construct a NN* $\Phi^{\mathbf{B}}_{\mathrm{inv};\tilde{\epsilon},\epsilon}$ *the ReLU realization of which* ϵ*-approximates* $\left(\mathbf{B}^{\mathrm{rb}}_{y}\right)^{-1}$ *(see Proposition B.3). Then the NNs of Theorem 4.3 can be explicitely constructed as*

$$\Phi^{\mathbf{u},\mathrm{rb}}_{\tilde{\epsilon},\epsilon}:=\Phi^{\kappa,d(\tilde{\epsilon}),d(\tilde{\epsilon}),1}_{\mathrm{mult};\frac{\epsilon}{3}}\odot\mathrm{P}\left(\Phi^{\mathbf{B}}_{\mathrm{inv};\tilde{\epsilon},\epsilon'},\Phi^{\mathbf{f}}_{\tilde{\epsilon},\epsilon''}\right)\bullet\left(\left(\begin{pmatrix}\mathbf{Id}_{\mathbb{R}^P}\\ \mathbf{Id}_{\mathbb{R}^P}\end{pmatrix},\mathbf{0}_{\mathbb{R}^{2P}}\right)\right)\quad\text{and}\quad\Phi^{\mathbf{u},\mathrm{h}}_{\tilde{\epsilon},\epsilon}:=\left(\left(\mathbf{V}_{\tilde{\epsilon}},\mathbf{0}_{\mathbb{R}^D}\right)\right)\odot\Phi^{\mathbf{u},\mathrm{rb}}_{\tilde{\epsilon},\epsilon},$$

Remark 4.5. *It can be checked in the proof of Theorem 4.3, specifically* (B.4) *and* (B.5) *that the constants* $C^{\mathbf{u}}_{L},C^{\mathbf{u}}_{M}$ *depend on the constants* $C_{\mathrm{coer}},C_{\mathrm{cont}},C_{\mathrm{rhs}}$ *in the following way (recall that* $\frac{C_{\mathrm{coer}}}{2C_{\mathrm{cont}}}\leq\delta=\frac{C_{\mathrm{coer}}}{C_{\mathrm{coer}}+C_{\mathrm{cont}}}\leq\frac{1}{2}$*):*

- $C^{\mathbf{u}}_{L}$ *depends affine linearly on*

$$\log_2^2\left(\frac{\log_2(\delta/2)}{\log_2(1-\delta)}\right),\quad\log_2\left(\frac{1}{C_{\mathrm{coer}}+C_{\mathrm{cont}}}\right),\quad\log_2\left(\max\left\{1,C_{\mathrm{rhs}},1/C_{\mathrm{coer}}\right\}\right).$$

- $C^{\mathbf{u}}_{M}$ *depends affine linearly on*

$$\log_2\left(\frac{1}{C_{\mathrm{coer}}+C_{\mathrm{cont}}}\right),\ \frac{\log_2(\delta/2)}{\log_2(1-\delta)}\cdot\log_2^3\left(\frac{\log_2(\delta/2)}{\log_2(1-\delta)}\right),\ \log_2\left(\max\left\{1,C_{\mathrm{rhs}},1/C_{\mathrm{coer}}\right\}\right).$$

Remark 4.6. *Theorem 4.3 guarantees the existence of two moderately sized NNs the realizations of which approximate the discretized solution maps:*

$$\mathcal{Y}\to\mathbb{R}^D\colon\quad y\mapsto\tilde{\mathbf{u}}^{\mathrm{h}}_{y,\tilde{\epsilon}},\ \text{and}\ \mathcal{Y}\to\mathbb{R}^{d(\tilde{\epsilon})}\colon\quad y\mapsto\mathbf{u}^{\mathrm{rb}}_{y,\tilde{\epsilon}}.\tag{4.1}$$

Also of interest is the approximation of the parametrized solution of the PDE, i.e., the map $\mathcal{Y}\times\Omega\to\mathbb{R}\colon\quad(y,x)\mapsto u_y(x)$*, where* Ω *is the domain on which the PDE is defined. Note that, if either the elements of the reduced basis or the elements of the high-fidelity basis can be very efficiently approximated by realizations of NNs, then the representation*

$$u_y(x)\approx\sum_{i=1}^{d(\tilde{\epsilon})}(\mathbf{u}^{\mathrm{rb}}_{y,\tilde{\epsilon}})_i\psi_i(x)=\sum_{i=1}^{D}(\tilde{\mathbf{u}}^{\mathrm{h}}_{y,\tilde{\epsilon}})_i\varphi_i(x)$$

suggests that $(y,x)\mapsto u_y(x)$ *can be approximated with essentially the cost of approximating the respective function in* (4.1)*. Many basis elements that are commonly used for the high-fidelity representation can indeed be approximated very efficiently by realizations of NNs, such as, e.g., polynomials, finite elements, or wavelets [69, 31, 52, 62].*

4.2 Examples of Neural Network Approximation of Parametric Maps

In this subsection, we apply Theorem 4.3 to a variety of concrete examples in which the approximation of the coefficient maps $\mathbf{u}^{\mathrm{rb}}_{\cdot,\tilde{\epsilon}}$, $\tilde{\mathbf{u}}^{\mathrm{h}}_{\cdot,\tilde{\epsilon}}$ can be approximated by comparatively small NNs. We show that the sizes of these NNs depend only on the size of associated reduced bases by verifying Assumption 4.1 and Assumption 4.2, respectively. We will discuss to what extent our results depend on the respective ambient dimensions D, p in Section 5.

We will state the following examples already in their variational formulation and note that they fulfill the requirements of Assumption 2.1. We also remark that the presented examples represent only a small portion of problems to which our theory is applicable.

4.2.1 Example I: Diffusion Equation

We consider a special case of [57, Chapter 2.3.1] which can be interpreted as a generalized version of the heavily used example $-\mathrm{div}(a\nabla u) = f$, where a is a scalar field (see for instance [13, 61] and the references therein). Let $n \in \mathbb{N}$, $\Omega \subset \mathbb{R}^n$, be a Lipschitz domain and $\mathcal{H} \coloneqq H^1_0(\Omega) = \left\{u \in H^1(\Omega) \colon\ u|_{\partial\Omega} = 0\right\}$. We assume that the parameter set is given by a compact set $\mathcal{T} \subset L^\infty(\Omega, \mathbb{R}^{n\times n})$ such that for all $\mathbf{T} \in \mathcal{T}$ and almost all $\mathbf{x} \in \Omega$ the matrix $\mathbf{T}(\mathbf{x})$ is symmetric, positive definite with matrix norm that can be bounded from above and below independently of $\mathbf{T}$ and $\mathbf{x}$. As we have noted in Assumption 2.1, we can assume that there exist some $(\mathbf{T}_i)_{i=0}^\infty \subset L^\infty(\Omega, \mathbb{R}^{n\times n})$ such that for every $\mathbf{T} \in \mathcal{T}$ there exist $(y_i(\mathbf{T}))_{i=1}^\infty \subset [-1,1]$ with $\mathbf{T} = \mathbf{T}_0 + \sum_{i=1}^\infty y_i(\mathbf{T})\mathbf{T}_i$. We restrict ourselves to the case of finitely supported sequences $(y_i)_{i=1}^\infty$. To be more precise, let $p \in \mathbb{N}$ be potentially very high but fixed, let $\mathcal{Y} \coloneqq [-1,1]^p$ and consider for $y \in \mathcal{Y}$ and some fixed $f \in \mathcal{H}^*$ the parametric PDE

$$b_y(u_y, v) \coloneqq \int_\Omega \mathbf{T}_0 \nabla u_y \nabla v \, d\mathbf{x} + \sum_{i=1}^p y_i \int_\Omega \mathbf{T}_i \nabla u_y \nabla v \, d\mathbf{x} = f(v), \qquad \text{for all } v \in \mathcal{H}.$$

Then, the parameter-dependency of the bilinear forms is linear, hence analytic whereas the parameter-dependency of the right-hand side is constant, hence also analytic, implying that $\overline{W}(S(\mathcal{Y}))$ decays exponentially fast. This in turn implies existence of small RBs $(\psi_i)_{i=1}^{d(\tilde{\epsilon})}$ where $d(\tilde{\epsilon})$ depends at most polylogarithmically on $1/\tilde{\epsilon}$. In this case, Assumption 4.1 and Assumption 4.2 are trivially fulfilled: for $\tilde{\epsilon} > 0, \epsilon > 0$ we can construct one-layer NNs $\Phi^{\mathbf{B}}_{\tilde{\epsilon},\epsilon}$ with p-dimensional input and $d(\tilde{\epsilon})^2$-dimensional output as well as $\Phi^{\mathbf{f}}_{\tilde{\epsilon},\epsilon}$ with p-dimensional input and $d(\tilde{\epsilon})$-dimensional output the ReLU-realizations of which exactly implement the maps $y \mapsto \mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}$ and $y \mapsto \mathbf{f}^{\mathrm{rb}}_{y,\tilde{\epsilon}}$, respectively.

In conclusion, in this example, we have, for $\tilde{\epsilon}, \epsilon > 0$,

$$B_L(\tilde{\epsilon}, \epsilon) = 1, \quad F_L(\tilde{\epsilon}, \epsilon) = 1, \quad B_M(\tilde{\epsilon}, \epsilon) \leq pd(\tilde{\epsilon})^2, \quad F_M(\tilde{\epsilon}, \epsilon) \leq pd(\tilde{\epsilon}).$$

Theorem 4.3 hence implies the existence of a NN approximating $\mathbf{u}^{\mathrm{rb}}_{\cdot,\tilde{\epsilon}}$ up to error ϵ with a size that is linear in p, polylogarithmic in $1/\epsilon$, and, up to a log factor, cubic in $d(\tilde{\epsilon})$. Moreover, we have shown the existence of a NN approximating $\tilde{\mathbf{u}}^{\mathrm{h}}_{\cdot,\tilde{\epsilon}}$ with a size that is linear in p, polylogarithmic in $1/\epsilon$, linear in D and, up to a log factor, cubic in $d(\tilde{\epsilon})$.

4.2.2 Example II: Linear Elasticity Equation

Let $\Omega \subset \mathbb{R}^3$ be a Lipschitz domain, $\Gamma_D, \Gamma_{N_1}, \Gamma_{N_2}, \Gamma_{N_3} \subset \partial\Omega$, be disjoint such that $\Gamma_D \cup \Gamma_{N_1} \cup \Gamma_{N_2} \cup \Gamma_{N_3} = \partial\Omega$, $\mathcal{H} \coloneqq [H^1_{\Gamma_D}(\Omega)]^3$, where $H^1_{\Gamma_D}(\Omega) = \left\{u \in H^1(\Omega) \colon\ u|_{\Gamma_D} = 0\right\}$. In variational formulation, this problem can be formulated as an affinely decomposed problem dependent on five parameters, i.e., $p = 5$. Let $\mathcal{Y} \coloneqq [\tilde{y}^{1,1}, \tilde{y}^{2,1}] \times \ldots \times [\tilde{y}^{1,5}, \tilde{y}^{2,5}] \subset \mathbb{R}^5$ such that $[\tilde{y}^{1,2}, \tilde{y}^{2,2}] \subset (-1, 1/2)$ and for $y = (y_1, \ldots, y_5) \in \mathcal{Y}$ we consider the problem

$$b_y(u_y, v) = f_y(v), \qquad \text{for all } v \in \mathcal{H},$$

where

- $b_y(u_y, v) \coloneqq \frac{y_1}{1+y_2} \int_\Omega \operatorname{trace}\left(\left(\nabla u_y + (\nabla(u_y)^T) \cdot \left(\nabla v + (\nabla v)^T\right)^T\right) d\mathbf{x} + \frac{y_1 y_2}{1-2y_2} \int_\Omega \operatorname{div}(u_y) \operatorname{div}(v)\ d\mathbf{x}\right.$,
- $f_y(v) \coloneqq y_3 \int_{\Gamma_1} \mathbf{n} \cdot v\, d\mathbf{x} + y_4 \int_{\Gamma_2} \mathbf{n} \cdot v\, d\mathbf{x} + y_5 \int_{\Gamma_3} \mathbf{n} \cdot v\, d\mathbf{x}$, and where $\mathbf{n}$ denotes the outward unit normal on $\partial\Omega$.

The parameter-dependency of the right-hand side is linear (hence analytic), whereas the parameter-dependency of the bilinear forms is rational, hence (due to the choice of $\tilde{y}^{1,2}, \tilde{y}^{2,2}$) also analytic and $\overline{W}_N(S(\mathcal{Y}))$ decays exponentially fast implying that we can choose $d(\tilde{\epsilon})$ to depend polylogarithmically on $\tilde{\epsilon}$. It is now easy to see that Assumption 4.1 and Assumption 4.2 are fulfilled with NNs the size of which is comparatively small: By [66], for every $\tilde{\epsilon}, \epsilon > 0$ we can find a NN with $\mathcal{O}(\log_2^2(1/\epsilon))$ layers and $\mathcal{O}(d(\tilde{\epsilon})^2 \log_2^3(1/\epsilon))$ non-zero weights the ReLU-realization of which approximates the map $y \mapsto \mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}$ up to an error of ϵ. Moreover, there exists a one-layer NN $\Phi^{\mathbf{f}}_{\tilde{\epsilon},\epsilon}$ with p-dimensional input and $d(\tilde{\epsilon})$-dimensional output the ReLU-realization of which exactly implements the map $y \mapsto \mathbf{f}^{\mathrm{rb}}_{y,\tilde{\epsilon}}$. In other words, in these examples, for $\tilde{\epsilon}, \epsilon > 0$,

$$B_L(\tilde{\epsilon}, \epsilon) \in \mathcal{O}\left(\log_2^2(1/\epsilon)\right), \quad F_L(\tilde{\epsilon}, \epsilon) = 1, \quad B_M(\tilde{\epsilon}, \epsilon) \in \mathcal{O}\left(d(\tilde{\epsilon})^2 \log_2^3(1/\epsilon)\right), \quad F_M(\tilde{\epsilon}, \epsilon) \leq 5d(\tilde{\epsilon}).$$

Thus, Theorem 4.3 implies the existence of NNs approximating $\mathbf{u}^{\mathrm{rb}}_{\cdot,\tilde{\epsilon}}$ up to error ϵ with a size that is polylogarithmic in $1/\epsilon$, and, up to a log factor, cubic in $d(\tilde{\epsilon})$. Moreover, there exist NNs approximating $\tilde{\mathbf{u}}^{\mathrm{h}}_{\cdot,\tilde{\epsilon}}$ up to error ϵ with a size that is linear in D, polylogarithmic in $1/\epsilon$, and, up to a log factor, cubic in $d(\tilde{\epsilon})$.

For a more thorough discussion of this example (a special case of the linear elasticity equation which describes the displacement of some elastic structure under physical stress on its boundaries), we refer to [57, Chapter 2.1.2, Chapter 2.3.2, Chapter 8.6].

5 Discussion: Dependence of Approximation Rates on Involved Dimensions

In this section, we will discuss our results in terms of the dependence on the involved dimensions. We would like to stress that the resulting approximation rates (which can be derived from Theorem 4.3) differ significantly from and are often substantially better than alternative approaches. As described in Section 2, there are three central dimensions that describe the hardness of the problem. These are the dimension D of the high-fidelity discretization space U^{h}, the dimension $d(\tilde{\epsilon})$ of the reduced basis space, and the dimension p of the parameter space $\mathcal{Y}$.

Dependence on D: Examples I and II above establish approximation rates that depend at most linearly on D, in particular, the dependence on D is not coupled to the dependence on ϵ. Another approach to solve these problems would be to directly solve the linear systems from the high-fidelity discretization. Without further assumptions on sparsity properties of the matrices, the resulting complexity would be $\mathcal{O}(D^3)$ plus the cost of assembling the high-fidelity stiffness matrices. Since $D \gg d(\tilde{\epsilon})$, this is significantly worse than the approximation rate provided by Theorem 4.3.

Dependence on $d(\tilde{\epsilon})$: If one assembles and solves the Galerkin scheme for a previously found reduced basis, one typically needs $\mathcal{O}\left(d(\tilde{\epsilon})^3\right)$ operations. By building NNs emulating this method, we achieve essentially the same approximation rate of $\mathcal{O}\left(d(\tilde{\epsilon})^3 \log_2(d(\tilde{\epsilon})) \cdot C(\epsilon)\right)$ where $C(\epsilon)$ depends polylogarithmically on the approximation accuracy ϵ.

Note that, while having comparable complexity, the NN-based approach is more versatile than using a Galerkin scheme and can be applied even when the underlying PPDE is fully unknown as long as sufficiently many snapshots are available.

Dependence on p: We start by comparing our result to naive NN approximation results which are simply based on the smoothness properties of the map $y \mapsto \tilde{\mathbf{u}}^{\mathrm{h}}_{\cdot,\tilde{\epsilon}}$ without using its specific structure. For example, if these maps are analytic, then classical approximation rates with NNs (such as those provided by [69, Theorem 1], [53, Theorem 3.1] or [28, Corollary 4.2]) promise approximations up to an error of ϵ with NNs Φ of size $M(\Phi) \leq c(p,n)D\epsilon^{-p/n}$ for arbitrary $n \in \mathbb{N}$ and a constant $c(p,n)$. In this case, the dependence on D is again linear, but coupled with the potentially quickly growing term $\epsilon^{-p/n}$. Similarly, when approximating the map $y \mapsto \tilde{\mathbf{u}}^{\mathrm{rb}}_{\cdot,\tilde{\epsilon}}$, one would obtain an approximation rate of $\epsilon^{-p/n}$. In addition, our approach is more flexible than the naive approach in the sense that Assumptions 4.1 and 4.2 could even be satisfied if the map $y \mapsto \mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}$ is non-smooth.

Now we analyze the dependence of our result to p in more detail. We recall from Theorem 4.3, that in our approach the sizes of approximating networks to achieve an error of ϵ depend only polylogarithmically on $1/\epsilon$, (up to a log factor) cubically on $d(\tilde{\epsilon})$, are independent from or at worst linear in D, and depend linearly on $B_M(\tilde{\epsilon},\epsilon)$, $B_M(\tilde{\epsilon},\epsilon)$, $F_M(\tilde{\epsilon},\epsilon)$, $F_L(\tilde{\epsilon},\epsilon)$, respectively. First of all, the dependence on p materializes through the quantities $B_M(\tilde{\epsilon},\epsilon)$, $B_L(\tilde{\epsilon},\epsilon)$, $F_M(\tilde{\epsilon},\epsilon)$, $F_L(\tilde{\epsilon},\epsilon)$ from Assumptions 4.1 and 4.2. We have seen that in both examples above, the associated weight quantities $B_M(\tilde{\epsilon},\epsilon)$, $F_M(\tilde{\epsilon},\epsilon)$ scale like $pd(\tilde{\epsilon})^2 \cdot \mathrm{polylog}(1/\epsilon)$, whereas the depth quantities $B_L(\tilde{\epsilon},\epsilon)$, $F_L(\tilde{\epsilon},\epsilon)$ scale polylogarithmically in $1/\epsilon$. Combining this observation with the statement of Theorem 4.3, we can conclude that the governing quantity in the obtained approximation rates is given by the dimension of the solution manifold $d(\tilde{\epsilon})$, derived by bounds on the Kolmogorov N-width (and, consequently, the inner N-width).

For problems of the type (2.6), where the involved maps θ_q are sufficiently smooth and the right-hand side is parameter-independent, one can show (see for instance [1, Equation 3.17] or [51] that $W_N(S(\mathcal{Y}))$ (and hence also $\overline{W}_N(S(\mathcal{Y}))$) scales like $e^{-cN^{Q_b}}$ for some $c > 0$. This implies for the commonly studied case $Q_b = p$ (such as in Example I of Section 4.2.1) that the dimension $d(\tilde{\epsilon})$ of the reduced basis space scales like $\mathcal{O}(\log_2(1/\tilde{\epsilon})^p)$. This bound (which is based on a Taylor expansion of the solution map) has been improved only in very special cases of Example I (see for instance [1, 3]) for small parameter dimensions p. Hence, by Theorem 4.3, the number of non-zero weights necessary to approximate the parameter-to-solution map $\tilde{\mathbf{u}}^{\mathrm{h}}_{y,\tilde{\epsilon}}$, can be upper bounded by $\mathcal{O}\left(p\log_2^{3p}(1/\tilde{\epsilon})\log_2(\log_2(1/\tilde{\epsilon}))\log_2^2(1/\epsilon)\log_2^2(\log_2(1/\epsilon)) + D\log_2(1/\tilde{\epsilon})^p\right)$ and the number of layers by $\mathcal{O}\left(p\log_2^2(1/\epsilon)\log_2(1/\tilde{\epsilon})\right)$. This implies that in our results there is a (mild form of a) curse of dimensionality which can only be circumvented if the sensitivity of the Kolmogorov N-width with regards to the parameter dimension p can be reduced further.

Acknowledgments

M.R. would like to thank Ingo Gühring for fruitful discussions on the topic.

G. K. acknowledges partial support by the Bundesministerium für Bildung und Forschung (BMBF) through the Berlin Institute for the Foundations of Learning and Data (BIFOLD), Project AP4, RTG DAEDALUS (RTG 2433), Projects P1, P3, and P8, RTG BIOQIC (RTG 2260), Projects P4 and P9, and by the Berlin Mathematics Research Center MATH+, Projects EF1-1 and EF1-4. P.P. was supported by a DFG Research Fellowship "Shearlet-based energy functionals for anisotropic phase-field methods". R.S. acknowledges partial support by the DFG through grant RTG DAEDALUS (RTG 2433), Project P14.

References

[1] M. Bachmayr and A. Cohen. Kolmogorov widths and low-rank approximations of parametric elliptic PDEs. *Math. Comp.*, 86(304):701–724, 2017.

[2] M. Bachmayr, A. Cohen, D. Dũng, and C. Schwab. Fully discrete approximation of parametric and stochastic elliptic PDEs. *SIAM J. Numer. Anal.*, 55(5):2151–2186, 2017.

[3] M. Bachmayr, A. Cohen, and W. Dahmen. Parametric PDEs: sparse or low-rank approximations? *IMA J. Numer. Anal.*, 38(4):1661–1708, 2018.

[4] M. Bachmayr, A. Cohen, and G. Migliorati. Sparse polynomial approximation of parametric elliptic PDEs. Part I: Affine coefficients. *ESAIM Math. Model. Numer. Anal.*, 51(1):321–339, 2017.

[5] E. Balmès. Parametric families of reduced finite element models. theory and applications. *Mech. Syst. Signal Process.*, 10(4):381 – 394, 1996.

[6] A. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inf. Theory*, 39(3):930–945, 1993.

[7] J. Berner, P. Grohs, and A. Jentzen. Analysis of the generalization error: Empirical risk minimization over deep artificial neural networks overcomes the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations. *arXiv preprint arXiv:1809.03062*, 2018.

[8] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk. Convergence rates for greedy algorithms in reduced basis methods. *SIAM J. Math. Anal.*, 43(3):1457–1472, 2011.

[9] H. Bölcskei, P. Grohs, G. Kutyniok, and P. C. Petersen. Optimal approximation with sparsely connected deep neural networks. *SIAM J. Math. Data Sci.*, 1:8–45, 2019.

[10] C. Canuto, T. Tonn, and K. Urban. A posteriori error analysis of the reduced basis method for nonaffine parametrized nonlinear PDEs. *SIAM J. Numer. Anal.*, 47(3):2001–2022, 2009.

[11] A. Chkifa, A. Cohen, and C. Schwab. Breaking the curse of dimensionality in sparse polynomial approximation of parametric PDEs. *J. Math. Pures Appl. (9)*, 103(2):400–428, 2015.

[12] A. Cohen and R. DeVore. Approximation of high-dimensional parametric PDEs. *Acta Numer.*, 24:1–159, 2015.

[13] N. Cohen, O. Sharir, and A. Shashua. On the expressive power of deep learning: A tensor analysis. In *Conference on Learning Theory*, pages 698–728, 2016.

[14] F. Cucker and S. Smale. On the mathematical foundations of learning. *Bull. Am. Math. Soc.*, 39:1–49, 2002.

[15] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Math. Control Signal Systems*, 2(4):303–314, 1989.

[16] W. Dahmen. How to best sample a solution manifold? In *Sampling theory, a renaissance*, Appl. Numer. Harmon. Anal., pages 403–435. Birkhäuser/Springer, Cham, 2015.

[17] N. Dal Santo, S. Deparis, and L. Pegolotti. Data driven approximation of parametrized PDEs by Reduced Basis and Neural Networks. *arXiv preprint arXiv:1904.01514*, 2019.

[18] R. DeVore, G. Petrova, and P. Wojtaszczyk. Greedy algorithms for reduced bases in Banach spaces. *Constr. Approx.*, 37(3):455–466, 2013.

[19] W. E, J. Han, and A. Jentzen. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Commun. Math. Stat.*, 5(4):349–380, 2017.

[20] M. Eigel, R. Schneider, P. Trunschke, and S. Wolf. Variational monte carlo-bridging concepts of machine learning and high dimensional partial differential equations. *Adv. Comp. Math.*, 45:2503–2532, 2019.

[21] D. Elbrächter, P. Grohs, A. Jentzen, and C. Schwab. DNN expression rate analysis of high-dimensional PDEs: Application to option pricing. *arXiv preprint arXiv:1809.07669*, 2018.

[22] R. Fox and H. Miura. An approximate analysis technique for design calculations. *AIAA J.*, 9(1):177–179, 1971.

[23] M. Geist, P. Petersen, M. Raslan, R. Schneider, and G. Kutyniok. Numerical solution of the parametric diffusion equation by deep neural networks. *arXiv preprint arXiv:2004.12131*, 2020.

[24] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[25] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *Esaim Math. Model. Numer. Anal.*, 41(3):575–605, 2007.

[26] P. Grohs, F. Hornung, A. Jentzen, and P. von Wurstemberger. A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations. *arXiv preprint arXiv:1809.02362*, 2018.

[27] P. Grohs, D. Perekrestenko, D. Elbrächter, and H. Bölcskei. Deep neural network approximation theory. *arXiv preprint arXiv:1901.02220*, 2019.

[28] I. Gühring, G. Kutyniok, and P. Petersen. Error bounds for approximations with deep relu neural networks in $W^{s,p}$ norms. *Anal. Appl. (Singap.)*, pages 1–57, 2019.

[29] B. Haasdonk. Reduced basis methods for parametrized PDEs—a tutorial introduction for stationary and instationary problems. In *Model reduction and approximation*, volume 15 of *Comput. Sci. Eng.*, pages 65–136. SIAM, Philadelphia, PA, 2017.

[30] J. Han, A. Jentzen, and W. E. Solving high-dimensional partial differential equations using deep learning. *Proc. Natl. Acad. Sci U.S.A.*, 115(34):8505–8510, 2018.

[31] J. He, L. Li, J. Xu, and C. Zheng. ReLU deep neural networks and linear finite elements. *arXiv preprint arXiv:1807.03973*, 2018.

[32] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[33] J. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer Briefs in Mathematics. Springer, Switzerland, 1 edition, 2015.

[34] J. S. Hesthaven and S. Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *J. Comput. Phys.*, 363:55–78, 2018.

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

[36] M. Hutzenthaler, A. Jentzen, T. Kruse, and T. Nguyen. A proof that rectified deep neural networks overcome the curse of dimensionality in the numerical approximation of semilinear heat equations. *SN Partial Differ. Equ. Appl.*, 1(10), 2020.

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

[38] N. Jung, B. Haasdonk, and D. Kroner. Reduced basis method for quadratically nonlinear transport equations. *Int. J. Appl. Math. Comput. Sci.*, 2(4):334–353, 2009.

[39] Y. Khoo, J. Lu, and L. Ying. Solving parametric PDE problems with artificial neural networks. *arXiv preprint arXiv:1707.03351*, 2017.

[40] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[41] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[42] K. Lee and K. Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *J. Comput. Phys.*, 404:108973, 2020.

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

[44] V. Maiorov and A. Pinkus. Lower bounds for approximation by MLP neural networks. *Neurocomputing*, 25(1-3):81–91, 1999.

[45] S. Mallat. Understanding deep convolutional networks. *Phil. Trans. R. Soc. A*, 374(2065):20150203, 2016.

[46] H. Mhaskar. Approximation properties of a multilayered feedforward artificial neural network. *Adv. Comput. Math.*, 1(1):61–80, 1993.

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

[48] H. Mhaskar, Q. Liao, and T. Poggio. Learning functions: when is deep better than shallow. *arXiv preprint arXiv:1603.00988*, 2016.

[49] A. K. Noor. Recent advances in reduction methods for nonlinear problems. *Comput. Struct.*, 13(1-3):31–44, 1981.

[50] A. K. Noor. On making large nonlinear problems small. *Comput. Methods Appl. Mech. Engrg.*, 34(1-3):955–985, 1982. FENOMECH '81, Part III (Stuttgart, 1981).

[51] M. Ohlberger and S. Rave. Reduced basis methods: Success, limitations and future challenges. *Proceedings of the Conference Algoritmy*, pages 1–12, 2016.

[52] J. A. A. Opschoor, P. C. Petersen, and C. Schwab. Deep relu networks and high-order finite element methods. *Analysis and Applications*, pages 1–56, 2020.

[53] P. C. Petersen and F. Voigtlaender. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *Neural Netw.*, 180:296–330, 2018.

[54] P. C. Petersen and F. Voigtlaender. Equivalence of approximation by convolutional neural networks and fully-connected networks. *Proc. Amer. Math. Soc.*, 148:1567–1581, 2020.

[55] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. *Int. J. Autom. Comput.*, 14(5):503–519, 2017.

[56] C. Prud'Homme, D. Rovas, K. Veroy, L. Machiels, Y. Maday, A. Patera, and G. Turinici. Reduced–basis output bound methods for parametrized partial differential equations. In *Proceedings SMA Symposium*, volume 1, page 1, 2002.

[57] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced basis methods for partial differential equations*, volume 92 of *Unitext*. Springer, Cham, 2016. An introduction, La Matematica per il 3+2.

[58] M. Raissi. Deep Hidden Physics Models: Deep Learning of Nonlinear Partial Differential Equations. *J. Mach. Learn. Res.*, 19(1):932—-955, Jan. 2018.

[59] C. Reisinger and Y. Zhang. Rectified deep neural networks overcome the curse of dimensionality for nonsmooth value functions in zero-sum games of nonlinear stiff systems. *arXiv preprint arXiv:1903.06652*, 2019.

[60] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics. *Arch. Comput. Methods Eng.*, 15(3):229–275, 2008.

[61] 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.

[62] U. Shaham, A. Cloninger, and R. R. Coifman. Provable approximation properties for deep neural networks. *Appl. Comput. Harmon. Anal.*, 44(3):537–557, 2018.

[63] J. Sirignano and K. Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. *J. Comput. Syst. Sci.*, 375:1339–1364, 2018.

[64] V. Strassen. Gaussian elimination is not optimal. *Numer. Math.*, 13(4):354–356, 1969.

[65] T. J. Sullivan. *Introduction to Uncertainty Quantification*, volume 63 of *Texts in Applied Mathematics*. Springer, 2015.

[66] M. Telgarsky. Neural networks and rational functions. In *34th International Conference on Machine Learning, ICML 2017*, volume 7, pages 5195–5210. International Machine Learning Society (IMLS), 1 2017.

[67] K. Veroy, C. Prud'Homme, D. Rovas, and A. Patera. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In *16th AIAA Computational Fluid Dynamics Conference*, page 3847, 2003.

[68] Y. Yang and P. Perdikaris. Physics-informed deep generative models. *arXiv preprint arXiv:1812.03511*, 2018.

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

[70] J. Zech, D. Dũng, and C. Schwab. Multilevel approximation of parametric and stochastic pdes. *Mathematical Models and Methods in Applied Sciences*, 29(09):1753–1817, 2019.

A Proofs of the Results from Section 3

A.1 Proof of Proposition 3.7

In this subsection, we will prove Proposition 3.7. As a preparation, we first prove the following special instance under which $M(\Phi^1 \bullet \Phi^2)$ can be estimated by $\max\{M(\Phi^1), M(\Phi^2)\}$.

Lemma A.1. *Let Φ be a NN with m-dimensional output and d-dimensional input. If $\mathbf{a} \in \mathbb{R}^{1\times m}$, then, for all $\ell = 1, \dots, L(\Phi)$,*

$$M_\ell(((\mathbf{a}, 0)) \bullet \Phi) \leq M_\ell(\Phi).$$

In particular, it holds that $M((\mathbf{a},0)\bullet\Phi)\leq M(\Phi)$. *Moreover, if* $\mathbf{D}\in\mathbb{R}^{d\times n}$ *such that, for every* $k\leq d$ *there is at most one* $l_k\leq n$ *such that* $\mathbf{D}_{k,l_k}\neq 0$, *then, for all* $\ell=1,\dots,L(\Phi)$,

$$M_\ell(\Phi\bullet((\mathbf{D},\mathbf{0}_{\mathbb{R}^d})))\leq M_\ell(\Phi).$$

In particular, it holds that $M(\Phi\bullet((\mathbf{D},\mathbf{0}_{\mathbb{R}^d})))\leq M(\Phi)$.

Proof. Let $\Phi=\big((\mathbf{A}_1,\mathbf{b}_1),\dots,(\mathbf{A}_L,\mathbf{b}_L)\big)$, and $\mathbf{a},\mathbf{D}$ as in the statement of the lemma. Then the result follows if

$$\|\mathbf{a}\mathbf{A}_L\|_0+\|\mathbf{a}\mathbf{b}_L\|_0\leq\|\mathbf{A}_L\|_0+\|\mathbf{b}_L\|_0 \tag{A.1}$$

and

$$\|\mathbf{A}_1\mathbf{D}\|_0\leq\|\mathbf{A}_1\|_0.$$

It is clear that $\|\mathbf{a}\mathbf{A}_L\|_0$ is less than the number of nonzero columns of $\mathbf{A}_L$ which is certainly bounded by $\|\mathbf{A}_L\|_0$. The same argument shows that $\|\mathbf{a}\mathbf{b}_L\|_0\leq\|\mathbf{b}_L\|_0$. This yields (A.1).

We have that for two vectors $\mathbf{p},\mathbf{q}\in\mathbb{R}^k$, $k\in\mathbb{N}$ and for all $\mu,\nu\in\mathbb{R}$

$$\|\mu\mathbf{p}+\nu\mathbf{q}\|_0\leq I(\mu)\|\mathbf{p}\|_0+I(\nu)\|\mathbf{q}\|_0,$$

where $I(\gamma)=0$ if $\gamma=0$ and $I(\gamma)=1$ otherwise. Also,

$$\|\mathbf{A}_1\mathbf{D}\|_0=\left\|\mathbf{D}^T\mathbf{A}_1^T\right\|_0=\sum_{l=1}^n\left\|\left(\mathbf{D}^T\mathbf{A}_1^T\right)_{l,-}\right\|_0,$$

where, for a matrix $\mathbf{G}$, $\mathbf{G}_{l,-}$ denotes the l-th row of $\mathbf{G}$. Moreover, we have that for all $l\leq n$

$$\left(\mathbf{D}^T\mathbf{A}_1^T\right)_{l,-}=\sum_{k=1}^d\left(\mathbf{D}^T\right)_{l,k}\left(\mathbf{A}_1^T\right)_{k,-}=\sum_{k=1}^d\mathbf{D}_{k,l}\left(\mathbf{A}_1^T\right)_{k,-}.$$

As a consequence, we obtain

$$\begin{aligned}\|\mathbf{A}_1\mathbf{D}\|_0&\leq\sum_{l=1}^n\left\|\sum_{k=1}^d\mathbf{D}_{k,l}\left(\mathbf{A}_1^T\right)_{k,-}\right\|_0\leq\sum_{l=1}^n\sum_{k=1}^d I\left(\mathbf{D}_{k,l}\right)\left\|\left(\mathbf{A}_1^T\right)_{k,-}\right\|_0\\&=\sum_{k=1}^d I\left(\mathbf{D}_{k,l_k}\right)\left\|\left(\mathbf{A}_1^T\right)_{k,-}\right\|_0\leq\|\mathbf{A}_1\|_0.\end{aligned}$$

□

Now we are ready to prove Proposition 3.7.

Proof of Proposition 3.7. Without loss of generality, assume that $Z\geq 1$. By [21, Lemma 6.2], there exists a NN $\times_\epsilon^Z$ with input dimension 2, output dimension 1 such that for $\Phi_\epsilon:=\times_\epsilon^Z$

$$L\left(\Phi_\epsilon\right)\leq 0.5\log_2\left(\frac{n\sqrt{dl}}{\epsilon}\right)+\log_2(Z)+6, \tag{A.2}$$

$$M\left(\Phi_\epsilon\right)\leq 90\cdot\left(\log_2\left(\frac{n\sqrt{dl}}{\epsilon}\right)+2\log_2(Z)+6\right), \tag{A.3}$$

$$M_1\left(\Phi_\epsilon\right)\leq 16,\ \text{as well as}\ M_{L(\Phi_\epsilon)}\left(\Phi_\epsilon\right)\leq 3, \tag{A.4}$$

$$\sup_{|a|,|b|\leq Z}\left|ab-\mathrm{R}_\varrho^{\mathbb{R}^2}\left(\Phi_\epsilon\right)(a,b)\right|\leq\frac{\epsilon}{n\sqrt{dl}}. \tag{A.5}$$

Since $\|\mathbf{A}\|_2, \|\mathbf{B}\|_2 \leq Z$ we know that for every $i = 1, ..., d,\ k = 1, ..., n,\ j = 1, ..., l$ we have that $|\mathbf{A}_{i,k}|, |\mathbf{B}_{k,j}| \leq Z$. We define, for $i \in \{1, \ldots, d\}, k \in \{1, \ldots, n\}, j \in \{1, \ldots, l\}$, the matrix $\mathbf{D}_{i,k,j}$ such that, for all $\mathbf{A} \in \mathbb{R}^{d \times n}, \mathbf{B} \in \mathbb{R}^{n \times l}$

$$\mathbf{D}_{i,k,j}(\mathbf{vec}(\mathbf{A}), \mathbf{vec}(\mathbf{B})) = (\mathbf{A}_{i,k}, \mathbf{B}_{k,j}).$$

Moreover, let

$$\Phi^Z_{i,k,j;\epsilon} \coloneqq \times^Z_\epsilon \bullet \left((\mathbf{D}_{i,k,j}, \mathbf{0}_{\mathbb{R}^2})\right).$$

We have, for all $i \in \{1, \ldots, d\}, k \in \{1, \ldots, n\}, j \in \{1, \ldots, l\}$, that $L\left(\Phi^Z_{i,k,j;\epsilon}\right) = L\left(\times^Z_\epsilon\right)$ and by Lemma A.1 that $\Phi^Z_{i,k,j;\epsilon}$ satisfies (A.2), (A.3), (A.4) with $\Phi_\epsilon \coloneqq \Phi^Z_{i,k,j;\epsilon}$. Moreover, we have by (A.5)

$$\sup_{(\mathbf{vec}(\mathbf{A}), \mathbf{vec}(\mathbf{B})) \in K^Z_{d,n,l}} \left| \mathbf{A}_{i,k}\mathbf{B}_{k,j} - \mathrm{R}^{K^Z_{d,n,l}}_\varrho \left(\Phi^Z_{i,j,k;\epsilon}\right)(\mathbf{vec}(\mathbf{A}), \mathbf{vec}(\mathbf{B})) \right| \leq \frac{\epsilon}{n\sqrt{dl}}. \tag{A.6}$$

As a next step, we set, for $\mathbf{1}_{\mathbb{R}^n} \in \mathbb{R}^n$ being a vector with each entry equal to 1,

$$\Phi^Z_{i,j;\epsilon} \coloneqq ((\mathbf{1}_{\mathbb{R}^n}, 0)) \bullet \mathrm{P}\left(\Phi^Z_{i,1,j;\epsilon}, ..., \Phi^Z_{i,n,j;\epsilon}\right) \bullet \left(\left(\begin{pmatrix} \mathbf{Id}_{\mathbb{R}^{n(d+l)}} \\ \vdots \\ \mathbf{Id}_{\mathbb{R}^{n(d+l)}} \end{pmatrix}, \mathbf{0}_{\mathbb{R}^{n^2(d+l)}} \right) \right),$$

which by Lemma 3.6 is a NN with $n \cdot (d + l)$-dimensional input and 1-dimensional output such that (A.2) holds with $\Phi_\epsilon \coloneqq \Phi^Z_{i,j;\epsilon}$. Moreover, by Lemmas A.1 and 3.6 and by (A.3) we have that

$$M\left(\Phi^Z_{i,j;\epsilon}\right) \leq M\left(\mathrm{P}\left(\Phi^Z_{i,1,j;\epsilon}, ..., \Phi^Z_{i,n,j;\epsilon}\right)\right) \leq 90n \cdot \left(\log_2\left(\frac{n\sqrt{dl}}{\epsilon}\right) + 2\log_2(Z) + 6\right). \tag{A.7}$$

Additionally, by Lemmas 3.6 and A.1 and (A.4), we obtain

$$M_1\left(\Phi^Z_{i,j;\epsilon}\right) \leq M_1\left(\mathrm{P}\left(\Phi^Z_{i,1,j;\epsilon}, ..., \Phi^Z_{i,n,j;\epsilon}\right)\right) \leq 16n.$$

and

$$M_{L\left(\Phi^Z_{i,j;\epsilon}\right)}\left(\Phi^Z_{i,j;\epsilon}\right) = M_{L\left(\Phi^Z_{i,j;\epsilon}\right)}\left(\mathrm{P}\left(\Phi^Z_{i,1,j;\epsilon}, ..., \Phi^Z_{i,n,j;\epsilon}\right)\right) \leq 2n. \tag{A.8}$$

By construction it follows that

$$\mathrm{R}^{K^Z_{d,n,l}}_\varrho \left(\Phi^Z_{i,j;\epsilon}\right)(\mathbf{vec}(\mathbf{A}), \mathbf{vec}(\mathbf{B})) = \sum_{k=1}^n \mathrm{R}^{K^Z_{d,n,l}}_\varrho \left(\Phi^Z_{i,k,j;\epsilon}\right)(\mathbf{vec}(\mathbf{A}), \mathbf{vec}(\mathbf{B}))$$

and hence we have, by (A.6),

$$\sup_{(\mathbf{vec}(\mathbf{A}), \mathbf{vec}(\mathbf{B})) \in K^Z_{d,n,l}} \left| \sum_{k=1}^n \mathbf{A}_{i,k}\mathbf{B}_{k,j} - \mathrm{R}^{K^Z_{d,n,l}}_\varrho \left(\Phi^Z_{i,j;\epsilon}\right)(\mathbf{vec}(\mathbf{A}), \mathbf{vec}(\mathbf{B})) \right| \leq \frac{\epsilon}{\sqrt{dl}}.$$

As a final step, we define $\Phi^{Z,d,n,l}_{\mathrm{mult};\tilde{\epsilon}} \coloneqq \mathrm{P}\left(\Phi^Z_{1,1;\epsilon}, ..., \Phi^Z_{d,1;\epsilon}, ..., \Phi^Z_{1,l;\epsilon}, ..., \Phi^Z_{d,l;\epsilon}\right) \bullet \left(\left(\begin{pmatrix} \mathbf{Id}_{\mathbb{R}^{n(d+l)}} \\ \vdots \\ \mathbf{Id}_{\mathbb{R}^{n(d+l)}} \end{pmatrix}, \mathbf{0}_{\mathbb{R}^{dln(d+l)}} \right) \right)$.
Then, by Lemma 3.6, we have that (A.2) is satisfied for $\Phi_\epsilon \coloneqq \Phi^{Z,d,n,l}_{\mathrm{mult};\tilde{\epsilon}}$. This yields (i) of the asserted statement. Moreover, invoking Lemma 3.6, Lemma A.1 and (A.7) yields that

$$M\left(\Phi^{Z,d,n,l}_{\mathrm{mult};\tilde{\epsilon}}\right) \leq 90dln \cdot \left(\log_2\left(\frac{n\sqrt{dl}}{\epsilon}\right) + 2\log_2(Z) + 6\right),$$

which yields (ii) of the result. Moreover, by Lemma 3.6 and (A.8) it follows that

$$M_1\left(\Phi_{\mathrm{mult};\tilde{\epsilon}}^{Z,d,n,l}\right) \leq 16dln \text{ and } M_{L\left(\Phi_{\mathrm{mult};\tilde{\epsilon}}^{Z,d,n,l}\right)}\left(\Phi_{\mathrm{mult};\tilde{\epsilon}}^{Z,d,n,l}\right) \leq 2dln,$$

completing the proof of (iii). By construction and using the fact that for any $\mathbf{N} \in \mathbb{R}^{d\times l}$ there holds

$$\|\mathbf{N}\|_2 \leq \sqrt{dl}\max_{i,j}|\mathbf{N}_{i,j}|,$$

we obtain that

$$\begin{aligned}
&\sup_{(\mathbf{vec}(\mathbf{A}),\mathbf{vec}(\mathbf{B}))\in K_{d,n,l}^Z}\left\|\mathbf{AB}-\mathbf{matr}\left(\mathrm{R}_\varrho^{K_{d,n,l}^Z}\left(\Phi_{\mathrm{mult};\tilde{\epsilon}}^{Z,d,n,l}\right)(\mathbf{vec}(\mathbf{A}),\mathbf{vec}(\mathbf{B}))\right)\right\|_2 \\
&\leq \sqrt{dl}\sup_{(\mathbf{vec}(\mathbf{A}),\mathbf{vec}(\mathbf{B}))\in K_{d,n,l}^Z}\max_{i=1,\ldots,d,\ j=1,\ldots,l}\left|\sum_{k=1}^n \mathbf{A}_{i,k}\mathbf{B}_{k,j}-\mathrm{R}_\varrho^{K_{d,n,l}^Z}\left(\Phi_{i,j;\epsilon}^Z\right)(\mathbf{vec}(\mathbf{A}),\mathbf{vec}(\mathbf{B}))\right| \leq \epsilon. \quad \text{(A.9)}
\end{aligned}$$

Equation (A.9) establishes (iv) of the asserted result. Finally, we have for any $(\mathbf{vec}(\mathbf{A}),\mathbf{vec}(\mathbf{B})) \in K_{d,n,l}^Z$ that

$$\begin{aligned}
&\left\|\mathbf{matr}\left(\mathrm{R}_\varrho^{K_{d,n,l}^Z}\left(\Phi_{\mathrm{mult};\tilde{\epsilon}}^{Z,d,n,l}\right)(\mathbf{vec}(\mathbf{A}),\mathbf{vec}(\mathbf{B}))\right)\right\|_2 \\
&\leq \left\|\mathbf{matr}\left(\mathrm{R}_\varrho^{K_{d,n,l}^Z}\left(\Phi_{\mathrm{mult};\tilde{\epsilon}}^{Z,d,n,l}\right)(\mathbf{vec}(\mathbf{A}),\mathbf{vec}(\mathbf{B}))\right)-\mathbf{AB}\right\|_2+\|\mathbf{AB}\|_2 \\
&\leq \epsilon+\|\mathbf{A}\|_2\|\mathbf{B}\|_2 \leq \epsilon+Z^2 \leq 1+Z^2.
\end{aligned}$$

This demonstrates that (v) holds and thereby finishes the proof. □

A.2 Proof of Theorem 3.8

The objective of this subsection is to prove of Theorem 3.8. Towards this goal, we construct NNs which emulate the map $\mathbf{A}\mapsto\mathbf{A}^k$ for $k\in\mathbb{N}$ and square matrices $\mathbf{A}$. This is done by heavily using Proposition 3.7. First of all, as a direct consequence of Proposition 3.7 we can estimate the sizes of the emulation of the multiplication of two squared matrices. Indeed, there exists a universal constant $C_1>0$ such that for all $d\in\mathbb{N}$, $Z>0$, $\epsilon\in(0,1)$

(i) $L\left(\Phi_{\mathrm{mult};\epsilon}^{Z,d,d,d}\right) \leq C_1\cdot(\log_2(1/\epsilon)+\log_2(d)+\log_2(\max\{1,Z\}))$,

(ii) $M\left(\Phi_{\mathrm{mult};\epsilon}^{Z,d,d,d}\right) \leq C_1\cdot(\log_2(1/\epsilon)+\log_2(d)+\log_2(\max\{1,Z\}))\,d^3$,

(iii) $M_1\left(\Phi_{\mathrm{mult};\epsilon}^{Z,d,d,d}\right) \leq C_1d^3$, as well as $M_{L\left(\Phi_{\mathrm{mult};\epsilon}^{Z,d,d,d}\right)}\left(\Phi_{\mathrm{mult};\epsilon}^{Z,d,d,d}\right) \leq C_1d^3$,

(iv) $\sup_{(\mathbf{vec}(\mathbf{A}),\mathbf{vec}(\mathbf{B}))\in K_{d,d,d}^Z}\left\|\mathbf{AB}-\mathbf{matr}\left(\mathrm{R}_\varrho^{K_{d,d,d}^Z}\left(\Phi_{\mathrm{mult};\epsilon}^{Z,d,d,d}\right)(\mathbf{vec}(\mathbf{A}),\mathbf{vec}(\mathbf{B}))\right)\right\|_2 \leq \epsilon$,

(v) for every $(\mathbf{vec}(\mathbf{A}),\mathbf{vec}(\mathbf{B}))\in K_{d,d,d}^Z$ we have

$$\left\|\mathbf{matr}\left(\mathrm{R}_\varrho^{K_{d,d,d}^Z}\left(\Phi_{\mathrm{mult};\epsilon}^{Z,d,d,d}\right)(\mathbf{vec}(\mathbf{A}),\mathbf{vec}(\mathbf{B}))\right)\right\|_2 \leq \epsilon+\|\mathbf{A}\|_2\|\mathbf{B}\|_2 \leq \epsilon+Z^2 \leq 1+Z^2.$$

One consequence of the ability to emulate the multiplication of matrices is that we can also emulate the squaring of matrices. We make this precise in the following definition.

Definition A.2. *For $d \in \mathbb{N}$, $Z > 0$, and $\epsilon \in (0,1)$ we define the NN*

$$\Phi_{2;\epsilon}^{Z,d} \coloneqq \Phi_{\mathrm{mult};\epsilon}^{Z,d,d,d} \bullet \left(\left(\begin{pmatrix} \mathbf{Id}_{\mathbb{R}^{d^2}} \\ \mathbf{Id}_{\mathbb{R}^{d^2}} \end{pmatrix}, \mathbf{0}_{\mathbb{R}^{2d^2}} \right) \right),$$

which has d^2-dimensional input and d^2-dimensional output. By Lemma 3.6 we have that there exists a constant $C_{\mathrm{sq}} > C_1$ such that for all $d \in \mathbb{N}$, $Z > 0$, $\epsilon \in (0,1)$

(i) $L\left(\Phi_{2;\epsilon}^{Z,d}\right) \leq C_{\mathrm{sq}} \cdot (\log_2(1/\epsilon) + \log_2(d) + \log_2(\max\{1, Z\}))$,

(ii) $M\left(\Phi_{2;\epsilon}^{Z,d}\right) \leq C_{\mathrm{sq}} d^3 \cdot (\log_2(1/\epsilon) + \log_2(d) + \log_2(\max\{1, Z\}))$,

(iii) $M_1\left(\Phi_{2;\epsilon}^{Z,d}\right) \leq C_{\mathrm{sq}} d^3$, *as well as* $M_{L\left(\Phi_{2;\epsilon}^{Z,d}\right)}\left(\Phi_{2;\epsilon}^{Z,d}\right) \leq C_{\mathrm{sq}} d^3$,

(iv) $\sup_{\mathbf{vec}(\mathbf{A}) \in K_d^Z} \left\| \mathbf{A}^2 - \mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^Z}\left(\Phi_{2;\epsilon}^{Z,d}\right)(\mathbf{vec}(\mathbf{A})) \right) \right\|_2 \leq \epsilon$,

(v) for all $\mathbf{vec}(\mathbf{A}) \in K_d^Z$ we have

$$\left\| \mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^Z}\left(\Phi_{2;\epsilon}^{Z,d}\right)(\mathbf{vec}(\mathbf{A})) \right) \right\|_2 \leq \epsilon + \|\mathbf{A}\|^2 \leq \epsilon + Z^2 \leq 1 + Z^2.$$

Our next goal is to approximate the map $\mathbf{A} \mapsto \mathbf{A}^k$ for an arbitrary $k \in \mathbb{N}_0$. We start with the case that k is a power of 2 and for the moment we only consider the set of all matrices the norm of which is bounded by $1/2$.

Proposition A.3. *Let $d \in \mathbb{N}$, $j \in \mathbb{N}$, as well as $\epsilon \in (0, 1/4)$. Then there exists a NN $\Phi_{2^j;\epsilon}^{1/2,d}$ with d^2-dimensional input and d^2-dimensional output with the following properties:*

(i) $L\left(\Phi_{2^j;\epsilon}^{1/2,d}\right) \leq C_{\mathrm{sq}} j \cdot (\log_2(1/\epsilon) + \log_2(d)) + 2C_{\mathrm{sq}} \cdot (j-1)$,

(ii) $M\left(\Phi_{2^j;\epsilon}^{1/2,d}\right) \leq C_{\mathrm{sq}} j d^3 \cdot (\log_2(1/\epsilon) + \log_2(d)) + 4C_{\mathrm{sq}} \cdot (j-1) d^3$,

(iii) $M_1\left(\Phi_{2^j;\epsilon}^{1/2,d}\right) \leq C_{\mathrm{sq}} d^3$, *as well as* $M_{L\left(\Phi_{2^j;\epsilon}^{1/2,d}\right)}\left(\Phi_{2^j;\epsilon}^{1/2,d}\right) \leq C_{\mathrm{sq}} d^3$,

(iv) $\sup_{\mathbf{vec}(\mathbf{A}) \in K_d^{1/2}} \left\| \mathbf{A}^{2^j} - \mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^j;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A})) \right) \right\|_2 \leq \epsilon$,

(v) for every $\mathbf{vec}(\mathbf{A}) \in K_d^{1/2}$ we have

$$\left\| \mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^j;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A})) \right) \right\|_2 \leq \epsilon + \left\| \mathbf{A}^{2^j} \right\|_2 \leq \epsilon + \|\mathbf{A}\|_2^{2^j} \leq \frac{1}{4} + \left(\frac{1}{2}\right)^{2^j} \leq \frac{1}{2}.$$

Proof. We show the statement by induction over $j \in \mathbb{N}$. For $j = 1$, the statement follows by choosing $\Phi_{2;\epsilon}^{1/2,d}$ as in Definition A.2. Assume now, as induction hypothesis, that the claim holds for an arbitrary, but fixed $j \in \mathbb{N}$, i.e., there exists a NN $\Phi_{2^j;\epsilon}^{1/2,d}$ such that

$$\left\| \mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^j;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A})) \right) - \mathbf{A}^{2^j} \right\|_2 \leq \epsilon, \qquad \left\| \mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^j;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A})) \right) \right\|_2 \leq \epsilon + \left(\frac{1}{2}\right)^{2^j} \tag{A.10}$$

and $\Phi_{2^j;\epsilon}^{1/2,d}$ satisfies (i),(ii),(iii). Now we define

$$\Phi_{2^{j+1};\epsilon}^{1/2,d} \coloneqq \Phi_{2;\frac{\epsilon}{4}}^{1,d} \odot \Phi_{2^j;\epsilon}^{1/2,d}.$$

By the triangle inequality, we obtain for any $\mathbf{vec}(\mathbf{A}) \in K_d^{1/2}$

$$\begin{aligned}
&\left\|\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^{j+1};\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right) - \mathbf{A}^{2^{j+1}}\right\|_2 \\
&\leq \left\|\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^{j+1};\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right) - \mathbf{A}^{2^j}\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^j;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right)\right\|_2 \\
&\quad + \left\|\mathbf{A}^{2^j}\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^j;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right) - \left(\mathbf{A}^{2^j}\right)^2\right\|_2.
\end{aligned} \tag{A.11}$$

By construction of $\Phi_{2^{j+1};\epsilon}^{1/2,d}$, we know that

$$\left\|\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^{j+1};\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right) - \left(\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^j;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right)\right)^2\right\|_2 \leq \frac{\epsilon}{4}.$$

Therefore, using the triangle inequality and the fact that $\|\cdot\|_2$ is a submultiplicative operator norm, we derive that

$$\begin{aligned}
&\left\|\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^{j+1};\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right) - \mathbf{A}^{2^j}\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^j;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right)\right\|_2 \\
&\leq \frac{\epsilon}{4} + \left\|\left(\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^j;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right)\right)^2 - \mathbf{A}^{2^j}\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^j;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right)\right\|_2 \\
&\leq \frac{\epsilon}{4} + \left\|\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^j;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right) - \mathbf{A}^{2^j}\right\|_2 \left\|\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^j;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right)\right\|_2 \\
&\leq \frac{\epsilon}{4} + \epsilon \cdot \left(\epsilon + \left(\frac{1}{2}\right)^{2^j}\right) \leq \frac{3}{4}\epsilon,
\end{aligned} \tag{A.12}$$

where the penultimate estimate follows by the induction hypothesis (A.10) and $\epsilon < 1/4$. Hence, since $\|\cdot\|_2$ is a submultiplicative operator norm, we obtain

$$\begin{aligned}
\left\|\mathbf{A}^{2^j}\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^j;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right) - \left(\mathbf{A}^{2^j}\right)^2\right\|_2 &\leq \left\|\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^j;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right) - \mathbf{A}^{2^j}\right\|_2 \left\|\mathbf{A}^{2^j}\right\|_2 \\
&\leq \frac{\epsilon}{4},
\end{aligned} \tag{A.13}$$

where we used $\left\|\mathbf{A}^{2^j}\right\|_2 \leq 1/4$ and the induction hypothesis (A.10). Applying (A.13) and (A.12) to (A.11) yields

$$\left\|\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^{j+1};\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right) - \mathbf{A}^{2^{j+1}}\right\|_2 \leq \epsilon. \tag{A.14}$$

A direct consequence of (A.14) is that

$$\left\|\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^{j+1};\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right)\right\|_2 \leq \epsilon + \left\|\mathbf{A}^{2^{j+1}}\right\|_2 \leq \epsilon + \|\mathbf{A}\|_2^{2^{j+1}}. \tag{A.15}$$

The estimates (A.14) and (A.15) complete the proof of the assertions (iv) and (v) of the proposition statement. Now we estimate the size of $\Phi_{2^{j+1};\epsilon}^{1/2,d}$. By the induction hypothesis and Lemma 3.6(a)(i), we obtain

$$\begin{aligned} L\left(\Phi_{2^{j+1};\epsilon}^{1/2,d}\right) &= L\left(\Phi_{2;\frac{\epsilon}{4}}^{1,d}\right) + L\left(\Phi_{2^j;\epsilon}^{1/2,d}\right) \\ &\leq C_{\mathrm{sq}} \cdot (\log_2(1/\epsilon) + \log_2(d) + \log_2(4) + j\log_2(1/\epsilon) + 2\cdot(j-1) + j\log_2(d)) \\ &= C_{\mathrm{sq}} \cdot ((j+1)\log_2(1/\epsilon) + (j+1)\log_2(d) + 2j)\,, \end{aligned}$$

which implies (i). Moreover, by the induction hypothesis and Lemma 3.6(a)(ii), we conclude that

$$\begin{aligned} M\left(\Phi_{2^{j+1};\epsilon}^{1/2,d}\right) &\leq M\left(\Phi_{2;\frac{\epsilon}{4}}^{1,d}\right) + M\left(\Phi_{2^j;\epsilon}^{1/2,d}\right) + M_1\left(\Phi_{2;\frac{\epsilon}{4}}^{1,d}\right) + M_{L\left(\Phi_{2^j;\epsilon}^{1/2,d}\right)}\left(\Phi_{2^j;\epsilon}^{1/2,d}\right) \\ &\leq C_{\mathrm{sq}} d^3 \cdot (\log_2(1/\epsilon) + \log_2(d) + \log_2(4) + j\log_2(1/\epsilon) + j\log_2(d) + 4\cdot(j-1)) + 2C_{\mathrm{sq}} d^3 \\ &= C_{\mathrm{sq}} d^3 \cdot ((j+1)\log_2(1/\epsilon) + (j+1)\log_2(d) + 4j)\,, \end{aligned}$$

implying (ii). Finally, it follows from Lemma 3.6(a)(iii) in combination with the induction hypothesis as well Lemma 3.6(a)(iv) that

$$M_1\left(\Phi_{2^{j+1};\epsilon}^{1/2,d}\right) = M_1\left(\Phi_{2^j;\epsilon}^{1/2,d}\right) \leq C_{\mathrm{sq}} d^3,$$

as well as

$$M_{L\left(\Phi_{2^{j+1};\epsilon}^{1/2,d}\right)}\left(\Phi_{2^{j+1};\epsilon}^{1/2,d}\right) = M_{L\left(\Phi_{2;\frac{\epsilon}{4}}^{1,d}\right)}\left(\Phi_{2;\frac{\epsilon}{4}}^{1,d}\right) \leq C_{\mathrm{sq}} d^3,$$

which finishes the proof. □

We proceed by demonstrating, how to build a NN that emulates the map $\mathbf{A} \mapsto \mathbf{A}^k$ for an arbitrary $k \in \mathbb{N}_0$. Again, for the moment we only consider the set of all matrices the norms of which are bounded by $1/2$. For the case of the set of all matrices the norms of which are bounded by an arbitrary $Z > 0$, we refer to Corollary A.5.

Proposition A.4. *Let $d \in \mathbb{N}$, $k \in \mathbb{N}_0$, and $\epsilon \in (0, 1/4)$. Then, there exists a NN $\Phi_{k;\epsilon}^{1/2,d}$ with d^2- dimensional input and d^2-dimensional output satisfying the following properties:*

(i)

$$\begin{aligned} L\left(\Phi_{k;\epsilon}^{1/2,d}\right) &\leq \lfloor \log_2(\max\{k,2\})\rfloor\, L\left(\Phi_{\mathrm{mult};\frac{\epsilon}{4}}^{1,d}\right) + L\left(\Phi_{2^{\lfloor \log_2(\max\{k,2\})\rfloor};\epsilon}^{1/2,d}\right) \\ &\leq 2C_{\mathrm{sq}} \lfloor \log_2(\max\{k,2\})\rfloor \cdot (\log_2(1/\epsilon) + \log_2(d) + 2)\,, \end{aligned}$$

(ii) $M\left(\Phi_{k;\epsilon}^{1/2,d}\right) \leq \frac{3}{2} C_{\mathrm{sq}} d^3 \cdot \lfloor \log_2(\max\{k,2\})\rfloor \cdot (\lfloor \log_2(\max\{k,2\})\rfloor + 1) \cdot (\log_2(1/\epsilon) + \log_2(d) + 4)$,

(iii) $M_1\left(\Phi_{k;\epsilon}^{1/2,d}\right) \leq C_{\mathrm{sq}} \cdot (\lfloor \log_2(\max\{k,2\})\rfloor + 1)\, d^3$, *as well as* $M_{L\left(\Phi_{k;\epsilon}^{1/2,d}\right)}\left(\Phi_{k;\epsilon}^{1/2,d}\right) \leq C_{\mathrm{sq}} d^3$,

(iv) $\sup_{\mathbf{vec}(\mathbf{A}) \in K_d^{1/2}} \left\| \mathbf{A}^k - \mathbf{matr}\left(\mathrm{R}_{\varrho}^{K_d^{1/2}}\left(\Phi_{k;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right)\right\|_2 \leq \epsilon$,

(v) for any $\mathbf{vec}(\mathbf{A}) \in K_d^{1/2}$ we have

$$\left\|\mathbf{matr}\left(\mathrm{R}_{\varrho}^{K_d^{1/2}}\left(\Phi_{k;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right)\right\|_2 \leq \epsilon + \|\mathbf{A}^k\|_2 \leq \frac{1}{4} + \|\mathbf{A}\|_2^k \leq \frac{1}{4} + \left(\frac{1}{2}\right)^k.$$

Proof. We prove the result per induction over $k \in \mathbb{N}_0$. The cases $k = 0$ and $k = 1$ hold trivially by defining the NNs

$$\Phi_{0;\epsilon}^{1/2,d} \coloneqq \left(\left(\mathbf{0}_{\mathbb{R}^{d^2}\times\mathbb{R}^{d^2}}, \mathbf{vec}(\mathbf{Id}_{\mathbb{R}^d})\right)\right), \qquad \Phi_{1;\epsilon}^{1/2,d} \coloneqq \left(\left(\mathbf{Id}_{\mathbb{R}^{d^2}}, \mathbf{0}_{\mathbb{R}^{d^2}}\right)\right).$$

For the induction hypothesis, we claim that the result holds true for all $k' \leq k \in \mathbb{N}$. If k is a power of two, then the result holds per Proposition A.3, thus we can assume without loss of generality, that k is not a power of two. We define $j \coloneqq \lfloor \log_2(k) \rfloor$ such that, for $t \coloneqq k - 2^j$, we have that $0 < t < 2^j$. This implies that $A^k = A^{2^j} A^t$. Hence, by Proposition A.3 and by the induction hypothesis, respectively, there exist a NN $\Phi_{2^j;\epsilon}^{1/2,d}$ satisfying (i)-(v) of Proposition A.3 and a NN $\Phi_{t;\epsilon}^{1/2,d}$ satisfying (i)-(v) of the statement of this proposition. We now define the NN

$$\Phi_{k;\epsilon}^{1/2,d} \coloneqq \Phi_{\mathrm{mult};\frac{\epsilon}{4}}^{1,d,d,d} \odot \mathrm{P}\left(\Phi_{2^j;\epsilon}^{1/2,d}, \Phi_{t;\epsilon}^{1/2,d}\right) \bullet \left(\left(\begin{pmatrix}\mathbf{Id}_{\mathbb{R}^{d^2}} \\ \mathbf{Id}_{\mathbb{R}^{d^2}}\end{pmatrix}, \mathbf{0}_{\mathbb{R}^{2d^2}}\right)\right).$$

By construction and Lemma 3.6(a)(iv), we first observe that

$$M_{L\left(\Phi_{k;\epsilon}^{1/2,d}\right)}\left(\Phi_{k;\epsilon}^{1/2,d}\right) = M_{L\left(\Phi_{\mathrm{mult};\frac{\epsilon}{4}}^{1,d,d,d}\right)}\left(\Phi_{\mathrm{mult};\frac{\epsilon}{4}}^{1,d,d,d}\right) \leq C_{\mathrm{sq}} d^3.$$

Moreover, we obtain by the induction hypothesis as well as Lemma 3.6(a)(iii) in combination with Lemma 3.6(b)(iv) that

$$\begin{aligned} M_1\left(\Phi_{k;\epsilon}^{1/2,d}\right) &= M_1\left(\mathrm{P}\left(\Phi_{2^j;\epsilon}^{1/2,d}, \Phi_{t;\epsilon}^{1/2,d}\right)\right) = M_1\left(\Phi_{2^j;\epsilon}^{1/2,d}\right) + M_1\left(\Phi_{t;\epsilon}^{1/2,d}\right) \\ &\leq C_{\mathrm{sq}} d^3 + (j+1) C_{\mathrm{sq}} d^3 = (j+2) C_{\mathrm{sq}} d^3. \end{aligned}$$

This shows (iii). To show (iv), we perform a similar estimate as the one following (A.11). By the triangle inequality,

$$\begin{aligned} &\left\| \mathbf{matr}\left(\mathrm{R}_{\varrho}^{K_d^{1/2}}\left(\Phi_{k;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right) - \mathbf{A}^k \right\|_2 \\ &\leq \left\| \mathbf{matr}\left(\mathrm{R}_{\varrho}^{K_d^{1/2}}\left(\Phi_{k;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right) - \mathbf{A}^{2^j}\mathbf{matr}\left(\mathrm{R}_{\varrho}^{K_d^{1/2}}\left(\Phi_{t;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right) \right\|_2 \\ &\quad + \left\| \mathbf{A}^{2^j}\mathbf{matr}\left(\mathrm{R}_{\varrho}^{K_d^{1/2}}\left(\Phi_{t;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right) - \mathbf{A}^{2^j}\mathbf{A}^t \right\|_2. \end{aligned} \tag{A.16}$$

By the construction of $\Phi_{k;\epsilon}^{1/2,d}$ and the Proposition 3.7, we conclude that

$$\begin{aligned} &\Bigg\| \mathbf{matr}\left(\mathrm{R}_{\varrho}^{K_d^{1/2}}\left(\Phi_{k;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right) \\ &\quad - \mathbf{matr}\left(\mathrm{R}_{\varrho}^{K_d^{1/2}}\left(\Phi_{2^j;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right)\mathbf{matr}\left(\mathrm{R}_{\varrho}^{K_d^{1/2}}\left(\Phi_{t;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right) \Bigg\|_2 \\ &\quad \leq \frac{\epsilon}{4}. \end{aligned}$$

Hence, using (A.16), we can estimate

$$\begin{aligned}
&\left\|\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{k;\epsilon}^{1/2,d}(\mathbf{vec}(\mathbf{A}))\right)\right)-\mathbf{A}^k\right\|_2\\
&\leq \frac{\epsilon}{4}+\left\|\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^j;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right)\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{t;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right)\right.\\
&\quad\left.-\,\mathbf{A}^{2^j}\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{t;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right)\right\|_2\\
&\qquad+\left\|\mathbf{A}^{2^j}\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{t;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right)-\mathbf{A}^k\right\|_2\\
&\leq \frac{\epsilon}{4}+\left\|\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{t;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right)\right\|_2\left\|\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{2^j;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right)-\mathbf{A}^{2^j}\right\|_2\\
&\quad+\left\|\mathbf{A}^{2^j}\right\|_2\left\|\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{t;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right)-\mathbf{A}^t\right\|_2=:\frac{\epsilon}{4}+\mathrm{I}+\mathrm{II}.
\end{aligned}$$

We now consider two cases: If $t=1$, then we know by the construction of $\Phi_{1;\epsilon}^{1/2,d}$ that $\mathrm{II}=0$. Thus

$$\frac{\epsilon}{4}+\mathrm{I}+\mathrm{II}=\frac{\epsilon}{4}+\mathrm{I}\leq\frac{\epsilon}{4}+\|\mathbf{A}\|_2\epsilon\leq\frac{3\epsilon}{4}\leq\epsilon.$$

If $t\geq 2$, then

$$\frac{\epsilon}{4}+\mathrm{I}+\mathrm{II}\leq\frac{\epsilon}{4}+\left(\epsilon+\|\mathbf{A}\|^t+\|\mathbf{A}\|^{2^j}\right)\epsilon\leq\frac{\epsilon}{4}+\left(\frac{1}{4}+\left(\frac{1}{2}\right)^t+\left(\frac{1}{2}\right)^{2^j}\right)\epsilon\leq\frac{\epsilon}{4}+\frac{3\epsilon}{4}=\epsilon,$$

where we have used that $\left(\frac{1}{2}\right)^t\leq\frac{1}{4}$ for $t\geq 2$. This shows (iv). In addition, by an application of the triangle inequality, we have that

$$\left\|\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1/2}}\left(\Phi_{k;\epsilon}^{1/2,d}\right)(\mathbf{vec}(\mathbf{A}))\right)\right\|_2\leq\epsilon+\left\|\mathbf{A}^k\right\|_2\leq\epsilon+\|\mathbf{A}\|_2^k\leq\frac{1}{4}+\left(\frac{1}{2}\right)^k.$$

This shows (v). Now we analyze the size of $\Phi_{k;\epsilon}^{1/2,d}$. We have by Lemma 3.6(a)(i) in combination with Lemma 3.6(b)(i) and by the induction hypothesis that

$$\begin{aligned}
L\left(\Phi_{k;\epsilon}^{1/2,d}\right)&\leq L\left(\Phi_{\mathrm{mult};\frac{\epsilon}{4}}^{1,d,d,d}\right)+\max\left\{L\left(\Phi_{2^j;\epsilon}^{1/2,d}\right),L\left(\Phi_{t;\epsilon}^{1/2,d}\right)\right\}\\
&\leq L\left(\Phi_{\mathrm{mult};\frac{\epsilon}{4}}^{1,d,d,d}\right)+\max\left\{L\left(\Phi_{2^j;\epsilon}^{1/2,d}\right),(j-1)L\left(\Phi_{\mathrm{mult};\frac{\epsilon}{4}}^{1,d,d,d}\right)+L\left(\Phi_{2^{j-1};\epsilon}^{1/2,d}\right)\right\}\\
&\leq L\left(\Phi_{\mathrm{mult};\frac{\epsilon}{4}}^{1,d,d,d}\right)+\max\left\{(j-1)L\left(\Phi_{\mathrm{mult};\frac{\epsilon}{4}}^{1,d,d,d}\right)+L\left(\Phi_{2^j;\epsilon}^{1/2,d}\right),(j-1)L\left(\Phi_{\mathrm{mult};\frac{\epsilon}{4}}^{1,d,d,d}\right)+L\left(\Phi_{2^{j-1};\epsilon}^{1/2,d}\right)\right\}\\
&\leq jL\left(\Phi_{\mathrm{mult};\frac{\epsilon}{4}}^{1,d,d,d}\right)+L\left(\Phi_{2^j;\epsilon}^{1/2,d}\right)\\
&\leq C_{\mathrm{sq}}j\cdot(\log_2(1/\epsilon)+\log_2(d)+2)+C_{\mathrm{sq}}j\cdot(\log_2(1/\epsilon)+\log_2(d))+2C_{\mathrm{sq}}\cdot(j-1)\\
&\leq 2C_{\mathrm{sq}}j\cdot(\log_2(1/\epsilon)+\log_2(d)+2),
\end{aligned}$$

which implies (i). Finally, we address the number of non-zero weights of the resulting NN. We first observe that, by Lemma 3.6(a)(ii),

$$\begin{aligned}
M\left(\Phi_{k;\epsilon}^{1/2,d}\right)&\leq\left(M\left(\Phi_{\mathrm{mult};\frac{\epsilon}{4}}^{1,d,d,d}\right)+M_1\left(\Phi_{\mathrm{mult};\frac{\epsilon}{4}}^{1,d,d,d}\right)\right)+M\left(\mathrm{P}\left(\Phi_{2^j;\epsilon}^{1/2,d},\Phi_{t;\epsilon}^{1/2,d}\right)\right)\\
&\quad+M_{L\left(\mathrm{P}\left(\Phi_{2^j;\epsilon}^{1/2,d},\Phi_{t;\epsilon}^{1/2,d}\right)\right)}\left(\mathrm{P}\left(\Phi_{2^j;\epsilon}^{1/2,d},\Phi_{t;\epsilon}^{1/2,d}\right)\right)\\
&=:\mathrm{I}'+\mathrm{II}'(a)+\mathrm{II}'(b).
\end{aligned}$$

Then, by the properties of the NN $\Phi^{1,d,d,d}_{\mathrm{mult};\frac{\epsilon}{4}}$, we obtain

$$\begin{aligned}\mathrm{I}' = M\left(\Phi^{1,d,d,d}_{\mathrm{mult};\frac{\epsilon}{4}}\right) + M_1\left(\Phi^{1,d,d,d}_{\mathrm{mult};\frac{\epsilon}{4}}\right) &\leq C_{\mathrm{sq}} d^3 \cdot (\log_2(1/\epsilon) + \log_2(d) + 2) + C_{\mathrm{sq}} d^3 \\ &= C_{\mathrm{sq}} d^3 \cdot (\log_2(1/\epsilon) + \log_2(d) + 3)\,.\end{aligned}$$

Next, we estimate

$$\mathrm{II}'(a) + \mathrm{II}'(b) = M\left(\mathrm{P}\left(\Phi^{1/2,d}_{2^j;\epsilon}, \Phi^{1/2,d}_{t;\epsilon}\right)\right) + M_{L\left(\mathrm{P}\left(\Phi^{1/2,d}_{2^j;\epsilon}, \Phi^{1/2,d}_{t;\epsilon}\right)\right)}\left(\mathrm{P}\left(\Phi^{1/2,d}_{2^j;\epsilon}, \Phi^{1/2,d}_{t;\epsilon}\right)\right).$$

Without loss of generality we assume that $L \coloneqq L\left(\Phi^{1/2,d}_{t;\epsilon}\right) - L\left(\Phi^{1/2,d}_{2^j;\epsilon}\right) > 0$. The other cases follow similarly. We have that $L \leq 2C_{\mathrm{sq}} j \cdot (\log_2(1/\epsilon) + \log_2(d) + 2)$ and, by the definition of the parallelization of two NNs with a different number of layers that

$$\begin{aligned}\mathrm{II}'(a) &= M\left(\mathrm{P}\left(\Phi^{1/2,d}_{2^j;\epsilon}, \Phi^{1/2,d}_{t;\epsilon}\right)\right) \\ &= M\left(\mathrm{P}\left(\Phi^{\mathbf{Id}}_{d^2,L} \odot \Phi^{1/2,d}_{2^j;\epsilon}, \Phi^{1/2,d}_{t;\epsilon}\right)\right) \\ &= M\left(\Phi^{\mathbf{Id}}_{d^2,L} \odot \Phi^{1/2,d}_{2^j;\epsilon}\right) + M\left(\Phi^{1/2,d}_{t;\epsilon}\right) \\ &\leq M\left(\Phi^{\mathbf{Id}}_{d^2,L}\right) + M_1\left(\Phi^{\mathbf{Id}}_{d^2,L}\right) + M_{L\left(\Phi^{1/2,d}_{2^j;\epsilon}\right)}\left(\Phi^{1/2,d}_{2^j;\epsilon}\right) + M\left(\Phi^{1/2,d}_{2^j;\epsilon}\right) + M\left(\Phi^{1/2,d}_{t;\epsilon}\right) \\ &\leq 2d^2(L+1) + C_{\mathrm{sq}} d^3 + M\left(\Phi^{1/2,d}_{t;\epsilon}\right) + M\left(\Phi^{1/2,d}_{2^j;\epsilon}\right),\end{aligned}$$

where we have used the definition of the parallelization for the first two equalities, Lemma 3.6(b)(iii) for the third equality, Lemma 3.6(a)(ii) for the fourth inequality as well as the properties of $\Phi^{\mathbf{Id}}_{d^2,L}$ in combination with Proposition A.3(iii) for the last inequality. Moreover, by the definition of the parallelization of two NNs with different numbers of layers, we conclude that

$$\mathrm{II}'(b) = M_{L\left(\mathrm{P}\left(\Phi^{1/2,d}_{2^j;\epsilon}, \Phi^{1/2,d}_{t;\epsilon}\right)\right)}\left(\mathrm{P}\left(\Phi^{1/2,d}_{2^j;\epsilon}, \Phi^{1/2,d}_{d;\epsilon}\right)\right) \leq d^2 + C_{\mathrm{sq}} d^3.$$

Combining the estimates on I', $\mathrm{II}'(a)$, and $\mathrm{II}'(b)$, we obtain by using the induction hypothesis that

$$\begin{aligned}M\left(\Phi^{1/2,d}_{k;\epsilon}\right) &\leq C_{\mathrm{sq}} d^3 \cdot (\log_2(1/\epsilon) + \log_2(d) + 3) + 2d^2 \cdot (L+1) + d^2 + C_{\mathrm{sq}} d^3 + M\left(\Phi^{1/2,d}_{t;\epsilon}\right) + M\left(\Phi^{1/2,d}_{2^j;\epsilon}\right) \\ &\leq C_{\mathrm{sq}} d^3 \cdot (\log_2(1/\epsilon) + \log_2(d) + 4) + 2d^2 \cdot (L+2) + M\left(\Phi^{1/2,d}_{t;\epsilon}\right) + M\left(\Phi^{1/2,d}_{2^j;\epsilon}\right) \\ &\leq C_{\mathrm{sq}} \cdot (j+1) d^3 \cdot (\log_2(1/\epsilon) + \log_2(d) + 4) + 2d^2 \cdot (L+2) + M\left(\Phi^{1/2,d}_{t;\epsilon}\right) \\ &\leq C_{\mathrm{sq}} \cdot (j+1) d^3 \cdot (\log_2(1/\epsilon) + \log_2(d) + 4) + 2C_{\mathrm{sq}} j d^2 \cdot (\log_2(1/\epsilon) + \log_2(d) + 2) \\ &\qquad + 4d^2 + M\left(\Phi^{1/2,d}_{t;\epsilon}\right) \\ &\leq 3C_{\mathrm{sq}} \cdot (j+1) d^3 \cdot (\log_2(1/\epsilon) + \log_2(d) + 4) + M\left(\Phi^{1/2,d}_{t;\epsilon}\right) \\ &\leq 3C_{\mathrm{sq}} d^3 \cdot \left(j + 1 + \frac{j \cdot (j+1)}{2}\right) \cdot (\log_2(1/\epsilon) + \log_2(d) + 4)) \\ &= \frac{3}{2} C_{\mathrm{sq}} \cdot (j+1) \cdot (j+2) d^3 \cdot (\log_2(1/\epsilon) + \log_2(d) + 4)\,.\end{aligned}$$

□

Proposition A.4 only provides a construction of a NN the ReLU-realization of which emulates a power of a matrix $\mathbf{A}$, under the assumption that $\|\mathbf{A}\|_2 \leq 1/2$. We remove this restriction in the following corollary by presenting a construction of a NN $\Phi^{Z,d}_{k;\epsilon}$ the ReLU-realization of which approximates the map $\mathbf{A} \mapsto \mathbf{A}^k$, on the set of all matrices $\mathbf{A}$ the norms of which are bounded by an arbitrary $Z > 0$.

Corollary A.5. *There exists a universal constant $C_{\mathrm{pow}} > C_{\mathrm{sq}}$ such that for all $Z > 0$, $d \in \mathbb{N}$ and $k \in \mathbb{N}_0$, there exists some NN $\Phi_{k;\epsilon}^{Z,d}$ with the following properties:*

(i) $L\left(\Phi_{k;\epsilon}^{Z,d}\right) \leq C_{\mathrm{pow}} \log_2\left(\max\{k,2\}\right) \cdot \left(\log_2(1/\epsilon) + \log_2(d) + k \log_2\left(\max\{1,Z\}\right)\right)$,

(ii) $M\left(\Phi_{k;\epsilon}^{Z,d}\right) \leq C_{\mathrm{pow}} \log_2^2\left(\max\{k,2\}\right) d^3 \cdot \left(\log_2(1/\epsilon) + \log_2(d) + k \log_2\left(\max\{1,Z\}\right)\right)$,

(iii) $M_1\left(\Phi_{k;\epsilon}^{Z,d}\right) \leq C_{\mathrm{pow}} \log_2\left(\max\{k,2\}\right) d^3$, *as well as* $M_{L\left(\Phi_{k;\epsilon}^{Z,d}\right)}\left(\Phi_{k;\epsilon}^{Z}\right) \leq C_{\mathrm{pow}} d^3$,

(iv) $\sup_{\mathbf{vec}(\mathbf{A}) \in K_d^Z} \left\| \mathbf{A}^k - \mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^Z}\left(\Phi_{k;\epsilon}^{Z,d}\right)(\mathbf{vec}(\mathbf{A}))\right)\right\|_2 \leq \epsilon$,

(v) for any $\mathbf{vec}(\mathbf{A}) \in K_d^Z$ *we have*

$$\left\|\mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^Z}\left(\Phi_{k;\epsilon}^{Z,d}\right)(\mathbf{vec}(\mathbf{A}))\right)\right\|_2 \leq \epsilon + \|\mathbf{A}^k\|_2 \leq \epsilon + \|\mathbf{A}\|_2^k.$$

Proof. Let $((\mathbf{A}_1,\mathbf{b}_1),...,(\mathbf{A}_L,\mathbf{b}_L)) \coloneqq \Phi_{k;\frac{\epsilon}{2\max\{1,Z^k\}}}^{1/2,d}$ according to Proposition A.4. Then the NN

$$\Phi_{k;\epsilon}^{Z,d} \coloneqq \left(\left(\frac{1}{2Z}\mathbf{A}_1, \mathbf{b}_1\right), (\mathbf{A}_2,\mathbf{b}_2), ..., (\mathbf{A}_{L-1},\mathbf{b}_{L-1}), \left(2Z^k\mathbf{A}_L, 2Z^k\mathbf{b}_L\right)\right)$$

fulfills all of the desired properties. □

We have seen how to construct a NN that takes a matrix as an input and computes a power of this matrix. With this tool at hand, we are now ready to prove Theorem 3.8.

Proof of Theorem 3.8. By the properties of the partial sums of the Neumann series, for $m \in \mathbb{N}$ and every $\mathbf{vec}(\mathbf{A}) \in K_d^{1-\delta}$, we have that

$$\begin{aligned}\left\|(\mathbf{Id}_{\mathbb{R}^d} - \mathbf{A})^{-1} - \sum_{k=0}^{m} \mathbf{A}^k\right\|_2 &= \left\|(\mathbf{Id}_{\mathbb{R}^d} - \mathbf{A})^{-1}\mathbf{A}^{m+1}\right\|_2 \leq \left\|(\mathbf{Id}_{\mathbb{R}^d} - \mathbf{A})^{-1}\right\|_2 \|\mathbf{A}\|_2^{m+1} \\ &\leq \frac{1}{1-(1-\delta)} \cdot (1-\delta)^{m+1} = \frac{(1-\delta)^{m+1}}{\delta}.\end{aligned}$$

Hence, for

$$m(\epsilon,\delta) = \left\lceil \log_{1-\delta}(2)\log_2\left(\frac{\epsilon\delta}{2}\right)\right\rceil = \left\lceil \frac{\log_2(\epsilon)+\log_2(\delta)-1}{\log_2(1-\delta)}\right\rceil \geq \frac{\log_2(\epsilon)+\log_2(\delta)-1}{\log_2(1-\delta)}$$

we obtain

$$\left\|(\mathbf{Id}_{\mathbb{R}^d} - \mathbf{A})^{-1} - \sum_{k=0}^{m(\epsilon,\delta)} \mathbf{A}^k\right\|_2 \leq \frac{\epsilon}{2}.$$

Let now

$$\begin{aligned}&((\mathbf{A}_1,\mathbf{b}_1),...,(\mathbf{A}_L,\mathbf{b}_L)) \\ &\coloneqq \left(\left(\left(\mathbf{Id}_{\mathbb{R}^{d^2}}|...|\mathbf{Id}_{\mathbb{R}^{d^2}}\right), \mathbf{0}_{\mathbb{R}^{d^2}}\right)\right) \odot \mathrm{P}\left(\Phi_{1;\frac{\epsilon}{2(m(\epsilon,\delta)-1)}}^{1,d}, ..., \Phi_{m(\epsilon,\delta);\frac{\epsilon}{2(m(\epsilon,\delta)-1)}}^{1,d}\right) \bullet \left(\left(\begin{pmatrix}\mathbf{Id}_{\mathbb{R}^{d^2}}\\ \vdots \\ \mathbf{Id}_{\mathbb{R}^{d^2}}\end{pmatrix}, \mathbf{0}_{\mathbb{R}^{2m(\epsilon,\delta)d^2}}\right)\right),\end{aligned}$$

where $\left(\mathbf{Id}_{\mathbb{R}^{d^2}}|...|\mathbf{Id}_{\mathbb{R}^{d^2}}\right) \in \mathbb{R}^{d^2 \times m(\epsilon,\delta)\cdot d^2}$. Then we set

$$\Phi_{\mathrm{inv};\epsilon}^{1-\delta,d} \coloneqq ((\mathbf{A}_1, \mathbf{b}_1), ..., (\mathbf{A}_L, \mathbf{b}_L + \mathbf{vec}\left(\mathbf{Id}_{\mathbb{R}^d}\right))).$$

We have for any $\mathbf{vec}(\mathbf{A}) \in K_d^{1-\delta}$

$$\begin{aligned}
&\left\| (\mathbf{Id}_{\mathbb{R}^d} - \mathbf{A})^{-1} - \mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1-\delta}} \left(\Phi_{\mathrm{inv};\epsilon}^{1-\delta,d} \right) (\mathbf{vec}(\mathbf{A})) \right) \right\|_2 \\
&\leq \left\| (\mathbf{Id}_{\mathbb{R}^d} - \mathbf{A})^{-1} - \sum_{k=0}^{m(\epsilon,\delta)} \mathbf{A}^k \right\|_2 + \left\| \sum_{k=0}^{m(\epsilon,\delta)} \mathbf{A}^k - \mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1-\delta}} \left(\Phi_{\mathrm{inv};\epsilon}^{1-\delta,d} \right) (\mathbf{vec}(\mathbf{A})) \right) \right\|_2 \\
&\leq \frac{\epsilon}{2} + \sum_{k=2}^{m(\epsilon,\delta)} \left\| \mathbf{A}^k - \mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1-\delta}} \left(\Phi_{k;\frac{\epsilon}{2(m(\epsilon,\delta)-1)}}^{1,d} \right) (\mathbf{vec}(\mathbf{A})) \right) \right\|_2 \\
&\leq \frac{\epsilon}{2} + (m(\epsilon,\delta) - 1)\, \frac{\epsilon}{2(m(\epsilon,\delta)-1)} = \epsilon,
\end{aligned}$$

where we have used that

$$\left\| \mathbf{A} - \mathbf{matr}\left(\mathrm{R}_\varrho^{K_d^{1-\delta}} \left(\Phi_{1;\frac{\epsilon}{2(m(\epsilon,\delta)-1)}}^{1,d} \right) (\mathbf{vec}(\mathbf{A})) \right) \right\|_2 = 0.$$

This completes the proof of (iii). Moreover, (iv) is a direct consequence of (iii). Now we analyze the size of the resulting NN. First of all, we have by Lemma 3.6(b)(i) and Corollary A.5 that

$$\begin{aligned}
L\left(\Phi_{\mathrm{inv};\epsilon}^{1-\delta,d}\right) &= \max_{k=1,...,m(\epsilon,\delta)} L\left(\Phi_{k;\frac{\epsilon}{2(m(\epsilon,\delta)-1)}}^{1,d}\right) \\
&\leq C_{\mathrm{pow}} \log_2\left(m(\epsilon,\delta)-1\right) \cdot \left(\log_2\left(1/\epsilon\right) + 1 + \log_2\left(m(\epsilon,\delta)-1\right) + \log_2(d)\right) \\
&\leq C_{\mathrm{pow}} \log_2\left(\frac{\log_2\left(0.5\epsilon\delta\right)}{\log_2(1-\delta)}\right) \cdot \left(\log_2\left(1/\epsilon\right) + 1 + \log_2\left(\frac{\log_2\left(0.5\epsilon\delta\right)}{\log_2(1-\delta)}\right) + \log_2(d)\right),
\end{aligned}$$

which implies (i). Moreover, by Lemma 3.6(b)(ii), Corollary A.5 and the monotonicity of the logarithm, we obtain

$$\begin{aligned}
M\left(\Phi_{\mathrm{inv};\epsilon}^{1-\delta,d}\right) &\leq 3 \cdot \left(\sum_{k=1}^{m(\epsilon,\delta)} M\left(\Phi_{k;\frac{\epsilon}{2(m(\epsilon,\delta)-1)}}^{1,d}\right) \right) \\
&\quad + 4C_{\mathrm{pow}} m(\epsilon,\delta) d^2 \log_2\left(m(\epsilon,\delta)\right) \cdot \left(\log_2\left(1/\epsilon\right) + 1 + \log_2\left(m(\epsilon,\delta)\right) + \log_2(d)\right) \\
&\leq 3C_{\mathrm{pow}} \cdot \left(\sum_{k=1}^{m(\epsilon,\delta)} \log_2^2(\max\{k,2\}) \right) d^3 \cdot \left(\log_2(1/\epsilon) + 1 + \log_2\left(m(\epsilon,\delta)\right) + \log_2(d)\right) \\
&\quad + 5m(\epsilon,\delta) d^2 C_{\mathrm{pow}} \log_2\left(m(\epsilon,\delta)\right) \cdot \left(\log_2\left(1/\epsilon\right) + 1 + \log_2\left(m(\epsilon,\delta)\right) + \log_2(d)\right) =: \mathrm{I}.
\end{aligned}$$

Since $\sum_{k=1}^{m(\epsilon,\delta)} \log_2^2(\max\{k,2\}) \leq m(\epsilon,\delta) \log_2^2(m(\epsilon,\delta))$, we obtain for some constant $C_{\mathrm{inv}} > C_{\mathrm{pow}}$ that

$$\mathrm{I} \leq C_{\mathrm{inv}} m(\epsilon,\delta) \log_2^2(m(\epsilon,\delta)) d^3 \cdot \left(\log_2(1/\epsilon) + \log_2\left(m(\epsilon,\delta)\right) + \log_2(d)\right).$$

This completes the proof. □

B Proof of Theorem 4.3

We start by establishing a bound on $\left\|\mathbf{Id}_{\mathbb{R}^{d(\tilde{\epsilon})}} - \alpha \mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}\right\|_2$.

Proposition B.1. *For any $\alpha \in (0, 1/C_{\mathrm{cont}})$ and $\delta \coloneqq \alpha C_{\mathrm{coer}} \in (0,1)$ there holds*

$$\left\|\mathbf{Id}_{\mathbb{R}^{d(\tilde{\epsilon})}} - \alpha \mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}\right\|_2 \leq 1 - \delta < 1, \quad \text{for all } y \in \mathcal{Y},\ \tilde{\epsilon} > 0.$$

Proof. Since $\mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}$ is symmetric, there holds that

$$\left\|\mathbf{Id}_{\mathbb{R}^{d(\tilde{\epsilon})}} - \alpha \mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}\right\|_2 = \max_{\mu \in \sigma\left(\mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}\right)} |1 - \alpha\mu| \leq \max_{\mu \in [C_{\mathrm{coer}}, C_{\mathrm{cont}}]} |1 - \alpha\mu| = 1 - \alpha C_{\mathrm{coer}} = 1 - \delta < 1,$$

for all $y \in \mathcal{Y},\ \tilde{\epsilon} > 0$. □

With an approximation to the parameter-dependent stiffness matrices with respect to a RB, due to Assumption 4.1, we can next state a construction of a NN the ReLU-realization of which approximates the map $y \mapsto \left(\mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}\right)^{-1}$. As a first step, we observe the following remark.

Remark B.2. *It is not hard to see that if $\left((\mathbf{A}^1_{\tilde{\epsilon},\epsilon}, \mathbf{b}^1_{\tilde{\epsilon},\epsilon}), ..., (\mathbf{A}^L_{\tilde{\epsilon},\epsilon}, \mathbf{b}^L_{\tilde{\epsilon},\epsilon})\right) \coloneqq \Phi^{\mathbf{B}}_{\tilde{\epsilon},\epsilon}$ is the NN of Assumption 4.1, then for*

$$\Phi^{\mathbf{B},\mathbf{Id}}_{\tilde{\epsilon},\epsilon} \coloneqq \left((\mathbf{A}^1_{\tilde{\epsilon},\epsilon}, \mathbf{b}^1_{\tilde{\epsilon},\epsilon}), ..., (-\mathbf{A}^L_{\tilde{\epsilon},\epsilon}, -\mathbf{b}^L_{\tilde{\epsilon},\epsilon} + \mathbf{vec}\left(\mathbf{Id}_{\mathbb{R}^{d(\tilde{\epsilon})}}\right))\right)$$

we have that

$$\sup_{y \in \mathcal{Y}} \left\|\mathbf{Id}_{\mathbb{R}^{d(\tilde{\epsilon})}} - \alpha \mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}} - \mathbf{matr}\left(\mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{B},\mathbf{Id}}_{\tilde{\epsilon},\epsilon}\right)(y)\right)\right\|_2 \leq \epsilon,$$

as well as $M\left(\Phi^{\mathbf{B},\mathbf{Id}}_{\tilde{\epsilon},\epsilon}\right) \leq B_M(\tilde{\epsilon},\epsilon) + d(\tilde{\epsilon})^2$ and $L\left(\Phi^{\mathbf{B},\mathbf{Id}}_{\tilde{\epsilon},\epsilon}\right) = B_L(\tilde{\epsilon},\epsilon)$.

Now we present the construction of the NN emulating $y \mapsto \left(\mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}\right)^{-1}$.

Proposition B.3. *Let $\tilde{\epsilon} \geq \hat{\epsilon}, \epsilon \in (0, \alpha/4 \cdot \min\{1, C_{\mathrm{coer}}\})$ and $\epsilon' \coloneqq 3/8 \cdot \epsilon\alpha C^2_{\mathrm{coer}} < \epsilon$. Assume that Assumption 4.1 holds. We define*

$$\Phi^{\mathbf{B}}_{\mathrm{inv};\tilde{\epsilon},\epsilon} \coloneqq \left((\alpha \mathbf{Id}_{\mathbb{R}^{d(\tilde{\epsilon})}}, \mathbf{0}_{\mathbb{R}^{d(\tilde{\epsilon})}})\right) \bullet \Phi^{1-\delta/2, d(\tilde{\epsilon})}_{\mathrm{inv};\frac{\epsilon}{2\alpha}} \odot \Phi^{\mathbf{B},\mathbf{Id}}_{\tilde{\epsilon},\epsilon'},$$

which has p-dimensional input and $d(\tilde{\epsilon})^2$- dimensional output.

Then, there exists a constant $C_B = C_B(C_{\mathrm{coer}}, C_{\mathrm{cont}}) > 0$ such that

(i) $L\left(\Phi^{\mathbf{B}}_{\mathrm{inv};\tilde{\epsilon},\epsilon}\right) \leq C_B \log_2(\log_2(1/\epsilon))\left(\log_2(1/\epsilon) + \log_2(\log_2(1/\epsilon)) + \log_2(d(\tilde{\epsilon}))\right) + B_L(\tilde{\epsilon},\epsilon')$,

(ii) $M\left(\Phi^{\mathbf{B}}_{\mathrm{inv};\tilde{\epsilon},\epsilon}\right) \leq C_B \log_2(1/\epsilon)\log^2_2(\log_2(1/\epsilon))d(\tilde{\epsilon})^3 \cdot \left(\log_2(1/\epsilon) + \log_2(\log_2(1/\epsilon)) + \log_2(d(\tilde{\epsilon}))\right) + 2B_M\left(\tilde{\epsilon},\epsilon'\right)$,

(iii) $\sup_{y\in\mathcal{Y}} \left\|\left(\mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}\right)^{-1} - \mathbf{matr}\left(\mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{B}}_{\mathrm{inv};\tilde{\epsilon},\epsilon}\right)(y)\right)\right\|_2 \leq \epsilon$,

(iv) $\sup_{y\in\mathcal{Y}} \left\|\mathbf{G}^{1/2}\mathbf{V}_{\tilde{\epsilon}} \cdot \left(\left(\mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}\right)^{-1} - \mathbf{matr}\left(\mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{B}}_{\mathrm{inv};\tilde{\epsilon},\epsilon}\right)(y)\right)\right)\right\|_2 \leq \epsilon$,

(v) $\sup_{y\in\mathcal{Y}} \left\|\mathbf{matr}\left(\mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{B}}_{\mathrm{inv};\tilde{\epsilon},\epsilon}\right)(y)\right)\right\|_2 \leq \epsilon + \frac{1}{C_{\mathrm{coer}}}$,

(vi) $\sup_{y\in\mathcal{Y}} \left\|\mathbf{G}^{1/2}\mathbf{V}_{\tilde{\epsilon}}\mathbf{matr}\left(\mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{B}}_{\mathrm{inv};\tilde{\epsilon},\epsilon}\right)(y)\right)\right\|_2 \leq \epsilon + \frac{1}{C_{\mathrm{coer}}}$.

Proof. First of all, for all $y \in \mathcal{Y}$ the matrix $\mathbf{matr}\left(\mathrm{R}_{\varrho}^{\mathcal{Y}}\left(\Phi_{\tilde{\epsilon},\epsilon'}^{\mathbf{B}}\right)(y)\right)$ is invertible. This can be deduced from the fact that

$$\left\|\alpha\mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}} - \mathbf{matr}\left(\mathrm{R}_{\varrho}^{\mathcal{Y}}\left(\Phi_{\tilde{\epsilon},\epsilon'}^{\mathbf{B}}\right)(y)\right)\right\|_2 \leq \epsilon' < \epsilon \leq \frac{\alpha\min\{1, C_{\mathrm{coer}}\}}{4} \leq \frac{\alpha C_{\mathrm{coer}}}{4}. \tag{B.1}$$

Indeed, we estimate

$$\begin{aligned}
&\min_{\mathbf{z}\in\mathbb{R}^{d(\tilde{\epsilon})}\setminus\{0\}} \frac{\left|\mathbf{matr}\left(\mathrm{R}_{\varrho}^{\mathcal{Y}}\left(\Phi_{\tilde{\epsilon},\epsilon'}^{\mathbf{B}}\right)(y)\right)\mathbf{z}\right|}{|\mathbf{z}|} \\
&\text{[Reverse triangle inequality]} \qquad \geq \min_{\mathbf{z}\in\mathbb{R}^{d(\tilde{\epsilon})}\setminus\{0\}} \frac{\left|\alpha\mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}}\mathbf{z}\right|}{|\mathbf{z}|} - \max_{\mathbf{z}\in\mathbb{R}^{d(\tilde{\epsilon})}\setminus\{0\}} \frac{\left|\alpha\mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}}\mathbf{z} - \mathbf{matr}\left(\mathrm{R}_{\varrho}^{\mathcal{Y}}\left(\Phi_{\tilde{\epsilon},\epsilon'}^{\mathbf{B}}\right)(y)\right)\mathbf{z}\right|}{|\mathbf{z}|} \\
&\text{[Definition of } \|.\|_2\text{]} \qquad \geq \left(\max_{\mathbf{z}\in\mathbb{R}^{d(\tilde{\epsilon})}\setminus\{0\}} \frac{|\mathbf{z}|}{\left|\alpha\mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}}\mathbf{z}\right|}\right)^{-1} - \left\|\alpha\mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}} - \mathbf{matr}\left(\mathrm{R}_{\varrho}^{\mathcal{Y}}\left(\Phi_{\tilde{\epsilon},\epsilon'}^{\mathbf{B}}\right)(y)\right)\right\|_2 \\
&\text{[Set } \tilde{\mathbf{z}} \coloneqq (\alpha\mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}})\mathbf{z}\text{]} \qquad \geq \left(\max_{\tilde{\mathbf{z}}\in\mathbb{R}^{d(\tilde{\epsilon})}\setminus\{0\}} \frac{|(\alpha\mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}})^{-1}\tilde{\mathbf{z}}|}{|\tilde{\mathbf{z}}|}\right)^{-1} - \left\|\alpha\mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}} - \mathbf{matr}\left(\mathrm{R}_{\varrho}^{\mathcal{Y}}\left(\Phi_{\tilde{\epsilon},\epsilon'}^{\mathbf{B}}\right)(y)\right)\right\|_2 \\
&\text{[Definition of } \|\cdot\|_2\text{]} \qquad \geq \left\|\left(\alpha\mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}}\right)^{-1}\right\|_2^{-1} - \left\|\alpha\mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}} - \mathbf{matr}\left(\mathrm{R}_{\varrho}^{\mathcal{Y}}\left(\Phi_{\tilde{\epsilon},\epsilon'}^{\mathbf{B}}\right)(y)\right)\right\|_2 \\
&\text{[By Equations (B.1) and (2.9)]} \qquad \geq \alpha C_{\mathrm{coer}} - \frac{\alpha C_{\mathrm{coer}}}{4} \geq \frac{3}{4}\alpha C_{\mathrm{coer}}.
\end{aligned}$$

Thus it follows that

$$\left\|\left(\mathbf{matr}\left(\mathrm{R}_{\varrho}^{\mathcal{Y}}\left(\Phi_{\tilde{\epsilon},\epsilon'}^{\mathbf{B}}\right)(y)\right)\right)^{-1}\right\|_2 \leq \frac{4}{3}\frac{1}{C_{\mathrm{coer}}\alpha}. \tag{B.2}$$

Then

$$\begin{aligned}
&\left\|\frac{1}{\alpha}\left(\mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}}\right)^{-1} - \mathbf{matr}\left(\mathrm{R}_{\varrho}^{\mathcal{Y}}\left(\Phi_{\mathrm{inv};\frac{\epsilon}{2\alpha}}^{1-\delta/2,d(\tilde{\epsilon})} \odot \Phi_{\tilde{\epsilon},\epsilon'}^{\mathbf{B},\mathbf{Id}}\right)(y)\right)\right\|_2 \\
&\leq \left\|\frac{1}{\alpha}\left(\mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}}\right)^{-1} - \left(\mathbf{matr}\left(\mathrm{R}_{\varrho}^{\mathcal{Y}}\left(\Phi_{\tilde{\epsilon},\epsilon'}^{\mathbf{B}}\right)(y)\right)\right)^{-1}\right\|_2 \\
&\quad + \left\|\left(\mathbf{matr}\left(\mathrm{R}_{\varrho}^{\mathcal{Y}}\left(\Phi_{\tilde{\epsilon},\epsilon'}^{\mathbf{B}}\right)(y)\right)\right)^{-1} - \mathbf{matr}\left(\mathrm{R}_{\varrho}^{\mathcal{Y}}\left(\Phi_{\mathrm{inv};\frac{\epsilon}{2\alpha}}^{1-\delta/2,d(\tilde{\epsilon})} \odot \Phi_{\tilde{\epsilon},\epsilon'}^{\mathbf{B},\mathbf{Id}}\right)(y)\right)\right\|_2 =: \mathrm{I} + \mathrm{II}.
\end{aligned}$$

Due to the fact that for two invertible matrices $\mathbf{M}, \mathbf{N}$,

$$\left\|\mathbf{M}^{-1} - \mathbf{N}^{-1}\right\|_2 = \left\|\mathbf{M}^{-1}(\mathbf{N} - \mathbf{M})\mathbf{N}^{-1}\right\|_2 \leq \|\mathbf{M} - \mathbf{N}\|_2\|\mathbf{M}^{-1}\|_2\|\mathbf{N}^{-1}\|_2,$$

we obtain

$$\begin{aligned}
\mathrm{I} &\leq \left\|\alpha\mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}} - \mathbf{matr}\left(\mathrm{R}_{\varrho}^{\mathcal{Y}}\left(\Phi_{\tilde{\epsilon},\epsilon'}^{\mathbf{B}}\right)(y)\right)\right\|_2 \left\|\left(\alpha\mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}}\right)^{-1}\right\|_2 \left\|\left(\mathbf{matr}\left(\mathrm{R}_{\varrho}^{\mathcal{Y}}\left(\Phi_{\tilde{\epsilon},\epsilon'}^{\mathbf{B}}\right)(y)\right)\right)^{-1}\right\|_2 \\
&\leq \frac{3}{8}\epsilon\alpha C_{\mathrm{coer}}^2 \frac{1}{\alpha C_{\mathrm{coer}}}\frac{4}{3}\frac{1}{C_{\mathrm{coer}}\alpha} = \frac{\epsilon}{2\alpha},
\end{aligned}$$

where we have used Assumption 4.1, Equation (2.9) and Equation (B.2). Now we turn our attention to estimating II. First, observe that for every $y \in \mathcal{Y}$ by the triangle inequality and Remark B.2, that

$$\begin{aligned}
\left\|\mathbf{matr}\left(\mathrm{R}_{\varrho}^{\mathcal{Y}}\left(\Phi_{\tilde{\epsilon},\epsilon'}^{\mathbf{B},\mathbf{Id}}\right)(y)\right)\right\|_2 &\leq \left\|\mathbf{matr}\left(\mathrm{R}_{\varrho}^{\mathcal{Y}}\left(\Phi_{\tilde{\epsilon},\epsilon'}^{\mathbf{B},\mathbf{Id}}\right)(y)\right) - \left(\mathbf{Id}_{\mathbb{R}^{d(\tilde{\epsilon})}} - \alpha\mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}}\right)\right\|_2 + \left\|\mathbf{Id}_{\mathbb{R}^{d(\tilde{\epsilon})}} - \alpha\mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}}\right\|_2 \\
&\leq \epsilon' + 1 - \delta \leq 1 - \delta + \frac{\alpha C_{\mathrm{coer}}}{4} \leq 1 - \delta + \frac{\alpha C_{\mathrm{cont}}}{4} \leq 1 - \delta + \frac{\delta}{2} = 1 - \frac{\delta}{2}.
\end{aligned}$$

Moreover, have that $\epsilon/(2\alpha) \leq \alpha/(8\alpha) < 1/4$. Hence, by Theorem 3.8, we obtain that $\mathrm{II} \leq \epsilon/2\alpha$. Putting everything together yields

$$\sup_{y\in\mathcal{Y}} \left\| \frac{1}{\alpha} \left(\mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}\right)^{-1} - \mathbf{matr}\left(\mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{1-\delta/2,d(\tilde{\epsilon})}_{\mathrm{inv};\frac{\epsilon}{2\alpha}} \odot \Phi^{\mathbf{B},\mathbf{Id}}_{\tilde{\epsilon},\epsilon'}\right)(y)\right)\right\|_2 \leq \mathrm{I} + \mathrm{II} \leq \frac{\epsilon}{\alpha}.$$

Finally, by construction we can conclude that

$$\sup_{y\in\mathcal{Y}} \left\| \left(\mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}\right)^{-1} - \mathbf{matr}\left(\mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{B}}_{\mathrm{inv};\tilde{\epsilon},\epsilon}\right)(y)\right)\right\|_2 \leq \epsilon.$$

This implies (iii) of the assertion. Now, by Equation (2.7) we obtain

$$\sup_{y\in\mathcal{Y}} \left\| \mathbf{G}^{1/2}\mathbf{V}_{\tilde{\epsilon}} \left(\mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}\right)^{-1} - \mathbf{G}^{1/2}\mathbf{V}_{\tilde{\epsilon}}\mathbf{matr}\left(\mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{B}}_{\mathrm{inv};\tilde{\epsilon},\epsilon}\right)(y)\right)\right\|_2 \leq \left\|\mathbf{G}^{1/2}\mathbf{V}_{\tilde{\epsilon}}\right\|_2 \epsilon = \epsilon,$$

completing the proof of (iv). Finally, for all $y \in \mathcal{Y}$ we estimate

$$\begin{aligned}
&\left\|\mathbf{G}^{1/2}\mathbf{V}_{\tilde{\epsilon}}\mathbf{matr}\left(\mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{B}}_{\mathrm{inv};\tilde{\epsilon},\epsilon}\right)(y)\right)\right\|_2 \\
&\leq \left\|\mathbf{G}^{1/2}\mathbf{V}_{\tilde{\epsilon}} \cdot \left(\left(\mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}\right)^{-1} - \mathbf{matr}\left(\mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{B}}_{\mathrm{inv};\tilde{\epsilon},\epsilon}\right)(y)\right)\right)\right\|_2 + \left\|\mathbf{G}^{1/2}\mathbf{V}_{\tilde{\epsilon}}\left(\mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}\right)^{-1}\right\|_2 \\
&\leq \epsilon + \frac{1}{C_{\mathrm{coer}}}.
\end{aligned}$$

This yields (vi). A minor modification of the calculation above yields (v). At last, we show (i) and (ii). First of all, it is clear that $L\left(\Phi^{\mathbf{B}}_{\mathrm{inv};\tilde{\epsilon},\epsilon}\right) = L\left(\Phi^{1-\delta/2,d(\tilde{\epsilon})}_{\mathrm{inv};\frac{\epsilon}{2\alpha}} \odot \Phi^{\mathbf{B},\mathbf{Id}}_{\tilde{\epsilon},\epsilon'}\right)$ and $M\left(\Phi^{\mathbf{B}}_{\mathrm{inv};\tilde{\epsilon},\epsilon}\right) = M\left(\Phi^{1-\delta/2,d(\tilde{\epsilon})}_{\mathrm{inv};\frac{\epsilon}{2\alpha}} \odot \Phi^{\mathbf{B},\mathbf{Id}}_{\tilde{\epsilon},\epsilon'}\right)$. Moreover, by Lemma 3.6(a)(i) in combination with Theorem 3.8 (i) we have

$$L\left(\Phi^{\mathbf{B}}_{\mathrm{inv};\tilde{\epsilon},\epsilon}\right) \leq C_{\mathrm{inv}} \log_2\left(m\left(\epsilon/(2\alpha),\delta/2\right)\right) \cdot \left(\log_2\left(2\alpha/\epsilon\right) + \log_2\left(m\left(\epsilon/(2\alpha),\delta/2\right)\right) + \log_2(d(\tilde{\epsilon}))\right) + B_L(\tilde{\epsilon},\epsilon')$$

and, by Lemma 3.6(a)(ii) in combination with Theorem 3.8(ii), we obtain

$$\begin{aligned}
&M\left(\Phi^{\mathbf{B}}_{\mathrm{inv};\tilde{\epsilon},\epsilon}\right) \\
&\quad\leq 2C_{\mathrm{inv}} m(\epsilon/(2\alpha),\delta/2) \log_2^2\left(m(\epsilon/(2\alpha),\delta/2)\right) d(\tilde{\epsilon})^3 \cdot \left(\log_2\left(2\alpha/\epsilon\right) + \log_2\left(m(\epsilon/(2\alpha),\delta/2)\right) + \log_2(d(\tilde{\epsilon}))\right) \\
&\quad+ 2d(\tilde{\epsilon})^2 + 2B_M(\tilde{\epsilon},\epsilon').
\end{aligned}$$

In addition, by definition of $m(\epsilon,\delta)$ in the statement of Theorem 3.8, for some constant $\tilde{C} > 0$ there holds $m\left(\epsilon/(2\alpha),\delta/2\right) \leq \tilde{C}\log_2(1/\epsilon)$. Hence, the claim follows for a suitably chosen constant $C_B = C_B(C_{\mathrm{coer}}, C_{\mathrm{cont}}) > 0$. □

B.1 Proof of Theorem 4.3

We start with proving (i) by deducing the estimate for $\Phi^{\mathbf{u},\mathrm{h}}_{\tilde{\epsilon},\epsilon}$. The estimate for $\Phi^{\mathbf{u},\mathrm{rb}}_{\tilde{\epsilon},\epsilon}$ follows in a similar, but simpler way. For $y \in \mathcal{Y}$, we have that

$$\begin{aligned}
&\left|\tilde{\mathbf{u}}^{\mathrm{h}}_{y,\tilde{\epsilon}} - \mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{u},\mathrm{h}}_{\tilde{\epsilon},\epsilon}\right)(y)\right|_{\mathbf{G}} \\
&= \left|\mathbf{G}^{1/2} \cdot \left(\mathbf{V}_{\tilde{\epsilon}}\left(\mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}\right)^{-1}\mathbf{f}^{\mathrm{rb}}_{y,\tilde{\epsilon}} - \mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{u},\mathrm{h}}_{\tilde{\epsilon},\epsilon}\right)(y)\right)\right| \\
&\leq \left|\mathbf{G}^{1/2}\mathbf{V}_{\tilde{\epsilon}} \cdot \left(\left(\mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}\right)^{-1}\mathbf{f}^{\mathrm{rb}}_{y,\tilde{\epsilon}} - \left(\mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}\right)^{-1}\mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{f}}_{\tilde{\epsilon},\epsilon''}\right)(y)\right)\right| \\
&\quad+ \left|\mathbf{G}^{1/2}\mathbf{V}_{\tilde{\epsilon}} \cdot \left(\left(\mathbf{B}^{\mathrm{rb}}_{y,\tilde{\epsilon}}\right)^{-1}\mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{f}}_{\tilde{\epsilon},\epsilon''}\right)(y) - \mathbf{matr}\left(\mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{B}}_{\mathrm{inv};\tilde{\epsilon},\epsilon'}\right)(y)\right)\mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{f}}_{\tilde{\epsilon},\epsilon''}\right)(y)\right)\right| \\
&\quad+ \left|\mathbf{G}^{1/2} \cdot \left(\mathbf{V}_{\tilde{\epsilon}}\mathbf{matr}\left(\mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{B}}_{\mathrm{inv};\tilde{\epsilon},\epsilon'}\right)(y)\right)\mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{f}}_{\tilde{\epsilon},\epsilon''}\right)(y) - \mathrm{R}^{\mathcal{Y}}_{\varrho}\left(\Phi^{\mathbf{u},\mathrm{h}}_{\tilde{\epsilon},\epsilon}\right)(y)\right)\right| =: \mathrm{I} + \mathrm{II} + \mathrm{III}.
\end{aligned}$$

We now estimate I, II, III separately. By Equation (2.7), Equation (2.9), Assumption 4.2, and the definition of ϵ'' there holds for $y \in \mathcal{Y}$ that

$$\mathrm{I} \leq \left\| \mathbf{G}^{1/2} \mathbf{V}_{\tilde{\epsilon}} \right\|_2 \left\| \left(\mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}} \right)^{-1} \right\|_2 \left| \mathbf{f}_{y,\tilde{\epsilon}}^{\mathrm{rb}} - \mathrm{R}_{\varrho}^{\mathcal{Y}} \left(\Phi_{\tilde{\epsilon},\epsilon''}^{\mathbf{f}} \right)(y) \right| \leq \frac{1}{C_{\mathrm{coer}}} \frac{\epsilon C_{\mathrm{coer}}}{3} = \frac{\epsilon}{3}.$$

We proceed with estimating II. It is not hard to see from Assumption 4.2 that

$$\sup_{y \in \mathcal{Y}} \left| \mathrm{R}_{\varrho}^{\mathcal{Y}} \left(\Phi_{\tilde{\epsilon},\epsilon}^{\mathbf{f}} \right)(y) \right| \leq \epsilon + C_{\mathrm{rhs}}. \tag{B.3}$$

By definition, $\epsilon' = \epsilon / \max\{6, C_{\mathrm{rhs}}\} \leq \epsilon$. Hence, by Assumption 4.1 and (B.3) in combination with Proposition B.3 (i), we obtain

$$\begin{aligned} \mathrm{II} &\leq \left\| \mathbf{G}^{1/2} \mathbf{V}_{\tilde{\epsilon}} \cdot \left(\left(\mathbf{B}_{y,\tilde{\epsilon}}^{\mathrm{rb}} \right)^{-1} - \mathbf{matr}\left(\mathrm{R}_{\varrho}^{\mathcal{Y}} \left(\Phi_{\mathrm{inv};\tilde{\epsilon},\epsilon'}^{\mathbf{B}} \right)(y) \right) \right) \right\|_2 \left| \mathrm{R}_{\varrho}^{\mathcal{Y}} \left(\Phi_{\tilde{\epsilon},\epsilon''}^{\mathbf{f}} \right)(y) \right| \leq \epsilon' \cdot \left(C_{\mathrm{rhs}} + \frac{\epsilon \cdot C_{\mathrm{coer}}}{3} \right) \\ &\leq \frac{\epsilon}{\max\{6, C_{\mathrm{rhs}}\}} C_{\mathrm{rhs}} + \frac{\epsilon C_{\mathrm{coer}}}{\max\{6, C_{\mathrm{rhs}}\}} \frac{\epsilon}{3} \leq \frac{2\epsilon}{6} = \frac{\epsilon}{3}, \end{aligned}$$

where we have used that $C_{\mathrm{coer}} \epsilon < C_{\mathrm{coer}} \alpha / 4 < 1$. Finally, we estimate III. Per construction, we have that

$$\mathrm{R}_{\varrho}^{\mathcal{Y}} \left(\Phi_{\tilde{\epsilon},\epsilon}^{\mathbf{u},\mathrm{h}} \right)(y) = \mathbf{V}_{\tilde{\epsilon}} \mathrm{R}_{\varrho}^{\mathcal{Y}} \left(\Phi_{\mathrm{mult};\frac{\epsilon}{3}}^{\kappa,d(\tilde{\epsilon}),d(\tilde{\epsilon}),1} \odot \mathrm{P}\left(\Phi_{\mathrm{inv};\tilde{\epsilon},\epsilon'}^{\mathbf{B}}, \Phi_{\tilde{\epsilon},\epsilon''}^{\mathbf{f}} \right) \right)(y, y).$$

Moreover, we have by Proposition B.3(v)

$$\left\| \mathbf{matr}\left(\mathrm{R}_{\varrho}^{\mathcal{Y}} \left(\Phi_{\mathrm{inv};\tilde{\epsilon},\epsilon'}^{\mathbf{B}} \right)(y) \right) \right\|_2 \leq \epsilon + \frac{1}{C_{\mathrm{coer}}} \leq 1 + \frac{1}{C_{\mathrm{coer}}} \leq \kappa$$

and by (B.3) that

$$\left| \mathrm{R}_{\varrho}^{\mathcal{Y}} \left(\Phi_{\tilde{\epsilon},\epsilon''}^{\mathbf{f}} \right)(y) \right| \leq \epsilon'' + C_{\mathrm{rhs}} \leq \epsilon C_{\mathrm{coer}} + C_{\mathrm{rhs}} \leq 1 + C_{\mathrm{rhs}} \leq \kappa.$$

Hence, by the choice of κ and Proposition 3.7 we conclude that $\mathrm{III} \leq \epsilon/3$. Combining the estimates on I, II, and III yields (i) and using (i) implies (v). Now we estimate the size of the NNs. We start with proving (ii). First of all, we have by the definition of $\Phi_{\tilde{\epsilon},\epsilon}^{\mathbf{u},\mathrm{rb}}$ and $\Phi_{\tilde{\epsilon},\epsilon}^{\mathbf{u},\mathrm{h}}$ as well as Lemma 3.6(a)(i) in combination with Proposition 3.7 that

$$\begin{aligned} L\left(\Phi_{\tilde{\epsilon},\epsilon}^{\mathbf{u},\mathrm{rb}} \right) &< L\left(\Phi_{\tilde{\epsilon},\epsilon}^{\mathbf{u},\mathrm{h}} \right) \leq 1 + L\left(\Phi_{\mathrm{mult};\frac{\epsilon}{3}}^{\kappa,d(\tilde{\epsilon}),d(\tilde{\epsilon}),1} \right) + L\left(\mathrm{P}\left(\Phi_{\mathrm{inv};\tilde{\epsilon},\epsilon'}^{\mathbf{B}}, \Phi_{\tilde{\epsilon},\epsilon''}^{\mathbf{f}} \right) \right) \\ &\leq 1 + C_{\mathrm{mult}} \cdot (\log_2(3/\epsilon) + 3/2 \log_2(d(\tilde{\epsilon})) + \log_2(\kappa)) + \max\left\{ L\left(\Phi_{\mathrm{inv};\tilde{\epsilon},\epsilon'}^{\mathbf{B}} \right), F_L\left(\tilde{\epsilon}, \epsilon'' \right) \right\} \\ &\leq C_L^{\mathbf{u}} \max\left\{ \log_2(\log_2(1/\epsilon)) \left(\log_2(1/\epsilon) + \log_2(\log_2(1/\epsilon)) + \log_2(d(\tilde{\epsilon})) \right) + B_L(\tilde{\epsilon}, \epsilon'''), F_L\left(\tilde{\epsilon}, \epsilon'' \right) \right\} \end{aligned} \tag{B.4}$$

where we applied Proposition B.3(i) and chose a suitable constant

$$C_L^{\mathbf{u}} = C_L^{\mathbf{u}}(\kappa, \epsilon', C_B) = C_L^{\mathbf{u}}(C_{\mathrm{rhs}}, C_{\mathrm{coer}}, C_{\mathrm{cont}}) > 0.$$

We now note that if we establish (iii), then (iv) follows immediately by Lemma 3.6(a)(ii). Thus, we proceed with proving (iii). First of all, by Lemma 3.6(a)(ii) in combination with Proposition 3.7 we have

$$\begin{aligned} M\left(\Phi_{\tilde{\epsilon},\epsilon}^{\mathbf{u},\mathrm{rb}} \right) &\leq 2M\left(\Phi_{\mathrm{mult};\frac{\epsilon}{3}}^{\kappa,d(\tilde{\epsilon}),d(\tilde{\epsilon}),1} \right) + 2M\left(\mathrm{P}\left(\Phi_{\mathrm{inv};\tilde{\epsilon},\epsilon'}^{\mathbf{B}}, \Phi_{\tilde{\epsilon},\epsilon''}^{\mathbf{f}} \right) \right) \\ &\leq 2C_{\mathrm{mult}} d(\tilde{\epsilon})^2 \cdot (\log_2(3/\epsilon) + 3/2 \log_2(d(\tilde{\epsilon})) + \log_2(\kappa)) + 2M\left(\mathrm{P}\left(\Phi_{\mathrm{inv};\tilde{\epsilon},\epsilon'}^{\mathbf{B}}, \Phi_{\tilde{\epsilon},\epsilon''}^{\mathbf{f}} \right) \right). \end{aligned}$$

Next, by Lemma 3.6(b)(ii) in combination with Proposition B.3 as well as Assumption 4.1 and Assumption 4.2 we have that

$$
\begin{aligned}
&M\left(\mathrm{P}\left(\Phi^{\mathbf{B}}_{\mathrm{inv};\tilde{\epsilon},\epsilon'}, \Phi^{\mathbf{f}}_{\tilde{\epsilon},\epsilon''}\right)\right) \\
&\leq M\left(\Phi^{\mathbf{B}}_{\mathrm{inv};\tilde{\epsilon},\epsilon'}\right) + M\left(\Phi^{\mathbf{f}}_{\tilde{\epsilon},\epsilon''}\right) \\
&\quad + 8d(\tilde{\epsilon})^2 \max\left\{C^{\mathbf{u}}_L \log_2(\log_2(1/\epsilon'))\left(\log_2(1/\epsilon') + \log_2(\log_2(1/\epsilon')) + \log_2(d(\tilde{\epsilon}))\right) + B_L(\tilde{\epsilon},\epsilon'''), F_L\left(\tilde{\epsilon},\epsilon''\right)\right\} \\
&\leq C_B \log_2(1/\epsilon') \log_2^2(\log_2(1/\epsilon')) d(\tilde{\epsilon})^3 \cdot \left(\log_2(1/\epsilon') + \log_2(\log_2(1/\epsilon')) + \log_2(d(\tilde{\epsilon}))\right) \\
&\quad + 8d(\tilde{\epsilon})^2 \max\left\{C^{\mathbf{u}}_L \log_2(\log_2(1/\epsilon'))\left(\log_2(1/\epsilon') + \log_2(\log_2(1/\epsilon')) + \log_2(d(\tilde{\epsilon}))\right) + B_L(\tilde{\epsilon},\epsilon'''), F_L\left(\tilde{\epsilon},\epsilon''\right)\right\} \\
&\quad + 2B_M\left(\tilde{\epsilon},\epsilon'''\right) + F_M\left(\tilde{\epsilon},\epsilon''\right) \\
&\leq C^{\mathbf{u}}_M d(\tilde{\epsilon})^2 \cdot \Bigg(d(\tilde{\epsilon}) \log_2(1/\epsilon) \log_2^2(\log_2(1/\epsilon))\big(\log_2(1/\epsilon) + \log_2(\log_2(1/\epsilon)) + \log_2(d(\tilde{\epsilon}))\big) \ldots \\
&\qquad\qquad \ldots + B_L(\tilde{\epsilon},\epsilon''') + F_L\left(\tilde{\epsilon},\epsilon''\right) \Bigg) \quad + 2B_M(\tilde{\epsilon},\epsilon''') + F_M\left(\tilde{\epsilon},\epsilon''\right), \quad \text{(B.5)}
\end{aligned}
$$

for a suitably chosen constant $C^{\mathbf{u}}_M = C^{\mathbf{u}}_M(\epsilon', C_B, C^{\mathbf{u}}_L) = C^{\mathbf{u}}_L(C_{\mathrm{rhs}}, C_{\mathrm{coer}}, C_{\mathrm{cont}}) > 0$. This shows the claim.

Numerical Solution of the Parametric Diffusion Equation by Deep Neural Networks

Moritz Geist*† Philipp Petersen*‡ Mones Raslan*†
Reinhold Schneider† Gitta Kutyniok†§¶

Abstract

We perform a comprehensive numerical study of the effect of approximation-theoretical results for neural networks on practical learning problems in the context of numerical analysis. As the underlying model, we study the machine-learning-based solution of parametric partial differential equations. Here, approximation theory predicts that the performance of the model should depend only very mildly on the dimension of the parameter space and is determined by the intrinsic dimension of the solution manifold of the parametric partial differential equation. We use various methods to establish comparability between test-cases by minimizing the effect of the choice of test-cases on the optimization and sampling aspects of the learning problem. We find strong support for the hypothesis that approximation-theoretical effects heavily influence the practical behavior of learning problems in numerical analysis.

Keywords: neural networks, parametric diffusion equation, numerical approximation, neural network capacity

MSC (2010) classification: 35J99, 41A25, 41A30, 68T05, 65N30

1 Introduction

This work studies the problem of numerically solving a specific parametric partial differential equation (PPDE) by training and applying neural networks (NNs). The central goal of the following exposition is to identify those key aspects of a parametric problem that render the problem harder or simpler to solve for methods based on NNs.

The underlying mathematical problem, the solution of PPDEs, is a standard problem in applied sciences and engineering. In this model, certain parts of a PDE such as the boundary conditions, the source terms, or the shape of the domain are controlled through a set of parameters, e.g., [29, 51]. In some applications where PDEs need to be evaluated very often or in real-time, individually solving the underlying PDEs for each choice of parameters becomes computationally infeasible. In this case, it is advisable to invoke methods that leverage on the joint structure of all the individual problems. A typical approach is that of constructing a *reduced basis* associated with the problem. With respect to this basis, the computational complexity of solving the PPDE is then significantly reduced, e.g., [29, 51, 57, 45].

Recently, as an alternative or to augment the reduced basis method, approaches were introduced that attempt to learn the parameter-to-solution map through methods of machine learning. We will provide a comprehensive overview of related approaches in Section 1.4. One approach is to train a NN to fit the discretized parameter-to-solution map, i.e., a map taking a parameter to a finite-element discretization of

*These authors contributed equally.

†Institut für Mathematik, Technische Universität Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany, e-mail: {geist, raslan, schneidr, kutyniok}@math.tu-berlin.de

‡University of Vienna, Faculty of Mathematics and Research Plattform Data Science @ Uni Vienna, Oskar Morgenstern Platz 1, 1090 Vienna, Austria, e-mail: philipp.petersen@univie.ac.at

§Fakultät Elektrotechnik und Informatik, Technische Universität Berlin

¶Department of Physics and Technology, University of Tromsø

the solution of the associated PDEs. This approach has already been analyzed theoretically in [36] where it was shown from an approximation-theoretical point of view that the hardness of representing the parameter-to-solution map by NNs is determined by a highly problem-specific notion of complexity that depends (in some cases) only very mildly on the dimension of the parameter space.

In this work, *we study the problem of learning the discretized parameter-to-solution map in practice.* We hypothesize that the approximation-theoretical capacity of a NN architecture is one of the central factors in determining the difficulty level of the learning problem in practice.

The motivation for this analysis is two-fold: First, we regard this as a general analysis of the feasibility of approximation-theoretical arguments in the study of deep learning. Second, specifically for the problem of numerical solution of PPDEs, we consider it important to identify which characteristics of a parametric problem determine its practical hardness. This is especially relevant to identify in which areas the application of this model is appropriate. We outline these two points of motivation in Section 1.1. The design of the numerical experiment is presented in Section 1.2 and we give a high-level report of our findings in Section 1.3.

1.1 Motivation

As already outlined before, we describe the motivation for this paper in the following two sections.

1.1.1 Understanding of Deep Learning in General

A typical learning problem consists of an unknown data model, a hypothesis class, and an optimization procedure to identify the best fit in the hypothesis class to the observed (sampled) data, e.g., [15, 16]. In a deep learning problem, the hypothesis class is the set of NNs with a specific architecture.

The approximation-theoretical point of view analyzes the trade-off between the capacity of the hypothesis class and the complexity of the data model. In this sense, this point of view describes only one aspect of the learning problem.

In the framework of approximation theory, there are precise ways to assess the hardness of an underlying problem. Concretely, this is done by identifying the rate by which the misfit between the hypothesis class and the data model decreases for sequences of growing hypothesis classes. For example, one common theme in the literature is the observation that for certain function classes, NNs do not admit a curse of dimension, i.e., their approximation rates do not deteriorate exponentially fast with increasing input dimension, e.g., [6, 61, 49]. Another theme is that classes of smooth functions can be approximated more efficiently than classes of rougher functions, e.g., [43, 67, 48, 46].

While these results offer some interpretation of why a certain problem should be harder or simpler, it is not clear how relevant these results are in practice. Indeed, there are at least three issues that call the approximation-theoretical explanation for a practical learning problem into question:

- *Tightness of the upper bounds:* Approximation-theoretical bounds usually describe worst-case error estimates for whole classes of functions. For individual functions or subsets of these function classes, there is no guarantee that one could not achieve a significantly better approximation rate.

- *Optimization and sampling prevent approximation theoretical effect from materializing:* As explained at the beginning of this section, the learning problem consists of multiple aspects, one of which is the ability of the hypothesis class to describe the data. Two further aspects are how well the sampling of the data model describes the true model and how well the optimization procedure performs in finding the best fit to the sampled data. Since the underlying optimization problem of deep learning is in general non-convex, it is conceivable that, while there theoretically exists a very good approximation of a function by a NN, finding it in practice is highly unlikely. Moreover, it is certainly possible that the sampling process does not contain sufficient information to guarantee that the optimization routine will identify the theoretically best approximation.

- *Asymptotic estimates:* All approximation-theoretical results mentioned until here and almost all in the literature describe the capacity of NNs to represent functions approximately with accuracy ε for sufficiently large architectures only in a regime where ε tends to zero and the size of the architecture is sufficiently large. The associated approximation rates may contain arbitrarily large implicit constants, and therefore it is entirely unclear if changes to the trade-off between the complexity of the data model and the size of the architecture have the theoretically predicted impact for moderately-sized practical learning problems.

We believe that, to understand the effect of approximation-theoretical capacities of NNs in practical learning scenarios, the learning problem associated with the parameter-to-solution map in a PPDE occupies a special role: It is, in essence, a high-dimensional approximation problem of a function that has a very strong low-dimensional, but highly non-trivial structure. What is more is that one can, to a certain extent, control the complexity of the problem, as we have seen in [36]. In this context, we can ask ourselves the following questions: Do we observe a curse of dimensionality in the practical solution of the problem? If not, how does the difficulty in practice scale with the parameter dimension? On which characteristics of the problem does the hardness of the practical solution thereof depend?

If we study these questions numerically, then the answers can be compared with the predictions from approximation-theoretical considerations. If the predictions coincide with the observed behavior and other causes, such as artefacts from the optimization and sampling procedure, can be ruled out, then we can view these experiments as a strong support for the practical relevance of approximation-theoretical arguments.

Because of this, we study the aforementioned questions in an extensive numerical experiment that will be described in Section 1.2 below.

1.1.2 Feasibility of the Machine-Learning-Based Solution of Parametric PDEs

The method (as described in [36]) of learning the parameter-to-solution map has at least two major advantages over classical approaches to solve PPDEs: First of all, the setup is completely independent of the underlying PPDE. This versatility of NNs could be quite desirable in an environment where many substantially different PPDEs are treated. Second, because this approach is fully data-driven, we do not require any knowledge of the underlying PDE. Indeed, as long as sufficiently many data points are supplied, for example, from a physical experiment, the approach could be feasible under high uncertainty of the model.

The main drawback of the method is the lack of theoretical understanding thereof. Moreover, for the theoretical results that do exist, we lack any evaluation of how pertinent the theoretical observations are for practical behavior. Most importantly, we do not have an a priori assessment for the practical feasibility of certain problems.

In [36], we observed that the complexity of the solution manifold, i.e., the set of all solutions of the PDE, is a central quantity involved in upper bounding the hardness of approximating the parameter-to-solution map with a NN. In practice, it is unclear to what extent this notion is appropriate and if the complexity of the solution manifold influences the performance of the method at all.

In the numerical experiment described in the next chapter, we explore the performance of the learning approach for various test-cases with different intrinsic complexities and observe the sensitivity of the method to the different setups.

1.2 The Experiment

To analyze the approximation-theoretical effect of the architecture on the overall performance of the learning problem in practice, we train a fully connected NN as considered in [36] on a variety of datasets stemming from different parameter choices of the parametric diffusion equation. The design of the data sets is such that we vary the relationship between the capacity of the architecture and the complexity of the data and report the effect on the overall performance.

In designing such an experiment, we face *three fundamental challenges hindering the comparability between test-cases*:

- *Effect of the optimization procedure:* The effect of the architecture on the optimization procedure is not clear, and this interplay may be a much stronger factor in the performance of the method than the capacity of the architecture to fit the data model. Similarly, the effect of the complexity of the data model could affect the optimization procedure and influence the performance of the learning method stronger than any approximation-theoretical effect.
- *Effect of the sampling procedure:* We train our network based on a finite number of samples of the true solution. The number and choice of samples could have a non-negligible effect on the overall performance and most importantly affect some test-cases more than others.
- *Quantification of the intrinsic complexity:* While we have theoretically established that the complexity of the solution manifold is the main factor in upper-bounding the hardness of the problem in the approximation-theoretical framework, we cannot, in practice, quantify this complexity.

We address these issues in the following four ways:

- *Keeping the architecture fixed:* An approximation-theoretical result on NNs is based on three ingredients. A function class $\mathcal{C}$, a worst-case accuracy $\varepsilon > 0$, and the size of the architecture.

  Whenever one of these hyper-parameters—the function class, the accuracy, or the architecture—is fixed, one can theoretically describe how changing a second parameter influences the last one. For example, for fixed $\mathcal{C}$, an approximation-theoretical statement yields an estimate of the necessary size of the architecture to achieve an accuracy of ε.

  Because of the potentially strong impact of the architecture on the optimization procedure, we expect that the most sensible point of view to test numerically is that where the architecture remains fixed while we vary the function class $\mathcal{C}$ and observe ε. This way, we can guarantee that the influence of the architecture on the optimization procedure is the same between test-cases.
- *Analyzing the convergence behavior a posteriori:* We are not aware of any method to guarantee a priori that the choice of the data model would not influence the convergence behavior. We do, however, analyze the convergence after the experiment to see if there are fundamental differences between our test-cases. This analysis reveals no significant differences between all the setups and therefore indicates that the effect of the data model on the optimization procedure is very similar between test-cases.
- *Establishing independence of sample generation:* We run the experiment multiple times for various numbers of training samples N chosen in the same way—uniformly at random—in every test-case. Between the choices of N, we observe a linear dependence of the achieved accuracy on N. This indicates that the influence of the number of N on the performance of the method is the same for all test-cases.
- *Design of semi-ordered test-cases:* While we are not able to assess the intrinsic complexity exactly, it is straight-forward to construct series of test-cases with increasing complexity. In this sense, we can introduce a semi-ordering of test-cases according to their complexity and observe to what extent the performance of the method follows this ordering.

We present the construction of the test-cases in Section 4 and discuss the measures taken to remove effects caused by the optimization and sampling procedures in greater detail in Appendix A. All of our test-cases consider the following parametric diffusion equation

$$-\nabla \cdot (a_y(\mathbf{x}) \cdot \nabla u_y(\mathbf{x})) = f(\mathbf{x}), \quad \text{on } \Omega = (0,1)^2, \quad u_y|_{\partial\Omega} = 0,$$

where $f \in L^2(\Omega)$ and $a_y \in L^\infty(\Omega)$, is a diffusion coefficient depending on a parameter $y \in \mathcal{Y}$. In our test-cases below, we learn a discretization of the map $\mathbb{R}^p \supset \mathcal{Y} \ni y \mapsto u_y$, where $p \in \mathbb{N}$, for various choices of parametrizations

$$\mathbb{R}^p \supset \mathcal{Y} \ni y \mapsto a_y. \tag{1.1}$$

Concretely, we vary the following characteristics of the parametrizations and observe the effect on the overall performance of the learning problem:

- *Type of parametrization:* We choose test-cases which differ with respect to the following characteristics: First, we study parametrizations (1.1) of various degrees of smoothness. Second, we study test-cases where the parametrization (1.1) is affine-linear and non-linear. Third, we consider cases, where $a_y = \sum_{i=1}^{p} \widetilde{a}_{y_i}$ for $\widetilde{a}_{y_i} \in L^\infty(\Omega)$ and where the supports of $(\widetilde{a}_{y_i})_{i=1}^{p}$ overlap or have various degrees of separation.

- *Dimension of parameter space:* The discretization of our solution space is done on the maximal computationally feasible grid (with respect to our workstation). We have chosen the dimensions p of the parameter spaces in such a way that the resolutions of the parametrized solutions are still meaningful with respect to the underlying discretization.

- *Complexity hyper-parameters:* To generate comparable test-cases with increasing complexities, we include two types of hyper-parametes into the data-generation process. One that directly influences the ellipticity of the problem and another that introduces a weighting of the parameter values.

We expect that these tests yield answers to the following questions: How versatile is the approach? Does it perform well only for special types of parametrizations or is it generally applicable? Do we observe a curse of dimensionality and how much does the performance of the learning method depend on the dimension of the parameter space? How strongly does the performance of the learning method depend on the intrinsic complexity of the data?

1.3 Our Findings

In the numerical experiments, which we report in Section 4.3 and evaluate in Section 4.4, we find that the proposed method is very sensitive to the underlying type of test-case. Indeed, we observe qualitatively different scaling behaviors of the achieved error with the dimension p of the parameter space between different test-cases. Concretely, we observe the following asymptotic behavior of the errors in different test-cases: $\mathcal{O}(1), \mathcal{O}(\log(p))$ and $\mathcal{O}(p^k)$ for $p \to \infty$ and $k > 0$, where k depends on one of the complexity hyper-parameters. Notably, we do not observe a scaling according to the curse of dimensionality, i.e., an error scaling exponentially with p, in any of the test-cases. We also observe that the achieved errors obey the semi-ordering of complexities of the test-cases. This shows that the method is very versatile and can be applied for various settings. Moreover, the complexity of the solution manifold appears to be a sensible predictor for the efficiency of the method.

In addition, we observe that the numerical results agree with the predictions that can be made via approximation-theoretical considerations. By design, we can exclude effects associated with the optimization and sampling procedures. This supports the practical relevance of approximation-theoretical results for this particular problem and for deep learning problems in general.

1.4 Related Works

The practical application of NNs in the context of PDEs dates back to the 1990s [37]. However, in recent years the topic again gained traction in the scientific community driven by the ever-increasing availability of computational power. Much of this research can be condensed into three main directions: Learning the solution of a single PDE, system identification, and goal-oriented approaches. The first of these directions uses NNs to directly model the solution of a (in some cases user-specified) single PDE [19, 53, 66, 40, 58], an SDE [7, 18], or even the joint solution for multiple boundary conditions [62]. These methods mostly rely on the differential operator of the PDE to evaluate the loss, but other approaches do exist [27]. In system identification, one tries to discover an underlying physical law from data by reverse-engineering the PDE. This can be done by attempting to uncover a hidden parameter of a known equation [54], or modeling physical relations [10, 52]. Conversely, goal-oriented approaches, try to infer a quantity of interest stemming

from the solution of an underlying PDE. For example, NNs can be used as a surrogate model to directly learn the quantity of interest and thereby circumvent the necessity of explicitly solving the equation [34]. A practical example for this is given by the ground state energy of a molecule which is derived from the solution of the electronic Schrödinger equation. This task has been efficiently solved by graph NNs [60, 41, 22]. Furthermore, building a surrogate model can be especially useful in uncertainty quantification [63]. NNs can also aid classical methods in solving goal-oriented tasks [13, 42]. In addition to the aforementioned research directions, further work has been done on fusing NNs with classical numerical methods to assist, for example, in model-order reduction, [56, 38].

Our work focuses on PPDEs and more specifically we are interested in learning the mapping from the parameter to the coefficients of the high-fidelity solution. Related but different approaches were analyzed in [30], and [17, 63], where the solution of the PPDE is learned in an already precomputed reduced basis or at point evaluations in fixed spatial coordinates.

On the theoretical side, the majority of works analyzing the power of NNs for the solution of (parametric) PDEs is concerned with an approximation-theoretical approach. Notable examples of such works include [25, 18, 24, 8, 26, 21, 33, 7, 11, 32], in which it is shown that NNs can overcome the curse of dimensionality in the approximative solution of some specific single PDE. In the same framework, it was shown in [11] how estimates on the approximation error imply bounds on the generalization error. Concerning the theoretical analysis of PPDEs, we mention [59, 36, 47, 28]. We will describe the results of the first two works in more detail in Section 3.2. The work [28] is concerned with an efficient approximation of a map that takes a noisy solution of a PDE as an input and returns a quantity of interest.

Additionally, we wish to mention that there exists a multitude of approaches (which are not necessarily directly RBM-or NN-related) that study the approximation of the parameter-to-solution map of PPDEs. These include methods based on sparse polynomials (see for instance [14, 31] and the references therein), tensors (see for instance [5, 20] and the references therein) and compressed sensing (see for instance [55, 65] and the references therein).

Parametric PDEs also appear in the context of stochastic PDEs or PDEs with random coefficients (see for instance [50]) and have been theoretically examined under the perspective of *uncertainty quantification*. For the sake of brevity, we only mention [14] and the references therein.

Finally, we mention that a comprehensive numerical study analyzing to what extent the approximation theoretical findings of NNs (not in the context of PPDEs) are visible in practice has been carried out in [2]. Similarly, in [23], a numerical algorithm that reproduces certain approximation-theoretically established exponential convergence rates of NNs was studied. The approximation rates of [12] were also numerically reproduced in that paper.

1.5 Outline

We start by describing the parametric diffusion equation and how we discretize it in Section 2. Then, we provide a formal introduction to NNs and a review of the approximation-theoretical results of NNs for parameter-to-solution maps in Section 3. In Section 4, we describe our numerical experiment. We start by stating three hypotheses underlying the examples in Subsection 4.1, before describing the set-up of our experiments in Subsection 4.2. After that, we present the results of the experiments in Subsection 4.3. Finally, in Subsection 4.4, we evaluate and interpret the observations. In Appendix A we describe the measures taken to ensure comparability between test-cases.

2 The Parametric Diffusion Equation

In this section, we will introduce the abstract setup and necessary notation that we will consider throughout this paper. First of all, we will introduce the *parameter-dependent diffusion equation* in Section 2.1. Afterwards, in Section 2.2, we recapitulate some basic facts about *high-fidelity discretizations* and introduce the *discretized parameter-to-solution map*.

2.1 The Parametric Diffusion Equation

Throughout this paper, we will consider the *parameter-dependent diffusion equation* with homogeneous Dirichlet boundary conditions

$$-\nabla \cdot (a(\mathbf{x}) \cdot \nabla u_a(\mathbf{x})) = f(\mathbf{x}), \quad \text{on } \Omega = (0,1)^2, \quad u|_{\partial\Omega} = 0, \tag{2.1}$$

where $f \in L^2(\Omega)$ is the parameter-independent right-hand side, $a \in \mathcal{A} \subset L^\infty(\Omega)$, and $\mathcal{A}$ constitutes some compact set of *parametrized diffusion coefficients*. In the following, we will examine different varieties of parametrized diffusion coefficient sets $\mathcal{A}$. Following [14] (by restricting ourselves to the case of finite-dimensional parameter spaces), we will always describe the elements of $\mathcal{A}$ by elements in $\mathbb{R}^p$ for some $p \in \mathbb{N}$. To be more precise, we will assume that

$$\mathcal{A} = \{a_y : \ y \in \mathcal{Y}\}, \tag{2.2}$$

where $\mathcal{Y} \subset \mathbb{R}^p$ is the *compact parameter space*.

A common assumption on the set $\mathcal{A}$, present in the first test-cases which we will describe below and especially convenient for the theoretical analysis of the problem, is given by *affine parametrizations* of the form

$$\mathcal{A} = \left\{ a_y = a_0 + \sum_{i=1}^{p} y_i a_i : y = (y_i)_{i=1}^p \in \mathcal{Y} \right\}, \tag{2.3}$$

where the functions $(a_i)_{i=0}^p \subset L^\infty(\Omega)$ are fixed.

After reparametrization, we consider the following problem, given in its variational formulation:

$$b_y(u_y, v) = \int_\Omega f(\mathbf{x}) v(\mathbf{x}) \, \mathrm{d}\mathbf{x}, \quad \text{for all } y \in \mathcal{Y}, \ v \in \mathcal{H}, \tag{2.4}$$

where

$$b_y : \mathcal{H} \times \mathcal{H} \to \mathbb{R}, \ (u,v) \mapsto \int_\Omega a_y(\mathbf{x}) \nabla u(\mathbf{x}) \nabla v(\mathbf{x}) \, \mathrm{d}\mathbf{x},$$

and $u_y \in \mathcal{H} \coloneqq H_0^1(\Omega)$ is the solution.[1]

We will consider experiments in which the involved bilinear forms are *uniformly continuous* and *uniformly coercive* in the sense that there exist $C_{\mathrm{cont}}, C_{\mathrm{coer}} > 0$ with

$$|b_y(u,v)| \le C_{\mathrm{cont}} \|u\|_{\mathcal{H}} \|v\|_{\mathcal{H}}, \quad \inf_{u \in \mathcal{H} \setminus \{0\}} \frac{b_y(u,u)}{\|u\|_{\mathcal{H}}^2} \ge C_{\mathrm{coer}}, \quad \text{for all } u, v \in \mathcal{H}, \ y \in \mathcal{Y}.$$

By the Lax-Milgram lemma (see [51, Lemma 2.1]), the problem of (2.4) is *well-posed*, i.e., for every $y \in \mathcal{Y}$ there exists exactly one $u_y \in \mathcal{H}$ such that (2.4) is satisfied and u_y depends continuously on f.

2.2 High-Fidelity Discretizations

In practice, one cannot hope to solve (2.4) exactly for every $y \in \mathcal{Y}$. Instead, if we assume for the moment that y is fixed, a common approach towards the calculation of an approximate solution of (2.4) is given by the *Galerkin method*, which we will describe briefly below following [29, Appendix A] and [51, Chapter 2.4]. In this framework, instead of solving (2.4), one solves a discrete scheme of the form

$$b_y\left(u_y^{\mathrm{h}}, v\right) = \int_\Omega f(\mathbf{x}) v(\mathbf{x}) \, \mathrm{d}\mathbf{x} \qquad \text{for all } v \in U^{\mathrm{h}}, \tag{2.5}$$

[1] Throughout this paper, we denote by $\mathcal{H}$ the space $H_0^1(\Omega) \coloneqq \{u \in H^1(\Omega) : \ u|_{\partial\Omega} = 0\}$, where $H^1(\Omega) \coloneqq W^{1,2}(\Omega)$ is the *first-order Sobolev space* and where $\partial\Omega$ denotes the *boundary of* Ω. On this space, we consider the norm $\|u\|_{\mathcal{H}} = \|u\|_{H_0^1(\Omega)} \coloneqq \|u\|_{H^1(\Omega)} = \left(\sum_{|\mathbf{a}| \le 1} \|D^{\mathbf{a}} u\|_{L^2(\Omega)}^2\right)^{1/2}$.

where $U^{\mathrm{h}} \subset \mathcal{H}$ is a subspace of $\mathcal{H}$ with $\dim\left(U^{\mathrm{h}}\right) < \infty$ and $u_y^{\mathrm{h}} \in U^{\mathrm{h}}$ is the solution of (2.5). Let us now assume that U^{h} is given. Moreover, let $D \coloneqq \dim\left(U^{\mathrm{h}}\right)$, and let $(\varphi_i)_{i=1}^D$ be a basis for U^{h}. Then the *stiffness matrix* $\mathbf{B}_y^{\mathrm{h}} \coloneqq (b_y(\varphi_j, \varphi_i))_{i,j=1}^D$ is non-singular and positive definite. The solution u_y^{h} of (2.5) satisfies

$$u_y^{\mathrm{h}} = \sum_{i=1}^{D} (\mathbf{u}_y^{\mathrm{h}})_i \varphi_i,$$

where $\mathbf{u}_y^{\mathrm{h}} \coloneqq (\mathbf{B}_y^{\mathrm{h}})^{-1}\mathbf{f}_y^{\mathrm{h}} \in \mathbb{R}^D$ and $\mathbf{f}_y^{\mathrm{h}} \coloneqq \left(\int_\Omega f(\mathbf{x})\varphi_i(\mathbf{x})\,\mathrm{d}\mathbf{x}\right)_{i=1}^D \in \mathbb{R}^D$. By Cea's Lemma (see [51, Lemma 2.2.]), u_y^{h} is, up to a universal constant, a best approximation of u_y in U^{h}.

In this framework, we can now define the central object of interest which is the map taking an element from the parameter space $\mathcal{Y}$ to the discretized solution $\mathbf{u}_y^{\mathrm{h}}$.

Definition 2.1. *Let $\Omega = (0,1)^2$, $U^{\mathrm{h}} \subset \mathcal{H}$ be a finite dimensional space, $\mathcal{A} \subset L^\infty(\Omega)$ with $\mathcal{Y} \subset \mathbb{R}^p$ for $p \in \mathbb{N}$ be as in* (2.2)*. Then we define the* discretized parameter-to-solution map (DPtSM) *by*

$$\mathcal{P}\colon \mathcal{Y} \to \mathbb{R}^D, \qquad y \mapsto \mathcal{P}(y) \coloneqq \mathbf{u}_y^{\mathrm{h}}.$$

Remark 2.2. *The DPtSM $\mathcal{P}$ is a potentially nonlinear map from a p-dimensional set to a D-dimensional space. Therefore, without using the information that $\mathcal{P}$ has a very specific structure described through $\mathcal{A}$ and the PDE* (2.1)*, a direct approximation of $\mathcal{P}$ as a high-dimensional smooth function will suffer from the curse of dimensionality [9, 44].*

Before we continue, let us introduce some crucial notation. Later, we need to compute the Sobolev norms of functions $v \in \mathcal{H}$. This will be done via a vector representation $\mathbf{v}$ of v with respect to the high-fidelity basis $(\varphi_i)_{i=1}^D$. We denote by $\mathbf{G} \coloneqq (\langle \varphi_i, \varphi_j \rangle_{\mathcal{H}})_{i,j=1}^D \in \mathbb{R}^{D\times D}$ the symmetric, positive definite *Gram matrix* of the basis functions $(\varphi_i)_{i=1}^D$. Then, for any $v \in U^{\mathrm{h}}$ with coefficient vector $\mathbf{v}$ with respect to the basis $(\varphi_i)_{i=1}^D$ we have[2] (see [51, Equation 2.41]) $|\mathbf{v}|_{\mathbf{G}} \coloneqq \left|\mathbf{G}^{1/2}\mathbf{v}\right| = \|v\|_{\mathcal{H}}$. In particular, $\left\|u_y^{\mathrm{h}}\right\|_{\mathcal{H}} = |\mathbf{u}_y^{\mathrm{h}}|_{\mathbf{G}}$, for all $y \in \mathcal{Y}$.

3 Approximation of the Discretized Parameter-to-Solution Map by Realizations of Neural Networks

In this section, we describe the approximation-theoretical motivation for the numerical study performed in this paper. We present a formal definition of NNs below. In Question 3.5, we present the underlying approximation-theoretical question of the considered learning problem. Thereafter, we recall the results of [36] showing that one can upper bound the approximation rates that NNs obtain when approximating the DPtSM through an implicit notion of complexity of the DPtSM.

3.1 Neural Networks

NNs describe functions of compositional form that result from repeatedly applying affine linear maps and a so-called activation function. From an approximation-theoretical point of view, it is sensible to count the number of active parameters of a NN. To associate a meaningful and mathematically precise notion of the number of parameters to a NN, we differentiate here between *neural networks* which are sets of matrices and vectors, essentially describing the parameters of the NN, and *realizations of neural networks* which are the associated functions. Concretely, we make the following definition:

Definition 3.1. *Let $n, L \in \mathbb{N}$. A* neural network Φ with input dimension n and L layers *is a sequence of matrix-vector tuples*

$$\Phi = \big((\mathbf{A}_1, \mathbf{b}_1), (\mathbf{A}_2, \mathbf{b}_2), \dots, (\mathbf{A}_L, \mathbf{b}_L)\big),$$

[2] In this paper, $|\mathbf{x}|$ denotes the *Euclidean norm* of $\mathbf{x} \in \mathbb{R}^n$.

where $N_0 = n$ and $N_1, \dots, N_L \in \mathbb{N}$, and where each $\mathbf{A}_\ell$ is an $N_\ell \times N_{\ell-1}$ matrix, and $\mathbf{b}_\ell \in \mathbb{R}^{N_\ell}$.

If Φ is a NN as above, $K \subset \mathbb{R}^n$, and if $\varrho\colon \mathbb{R} \to \mathbb{R}$ is arbitrary, then we define the associated realization of Φ with activation function ϱ over K *(in short, the ϱ-*realization of Φ over K*) as the map $\mathrm{R}_\varrho^K(\Phi)\colon K \to \mathbb{R}^{N_L}$ such that $\mathrm{R}_\varrho^K(\Phi)(\mathbf{x}) = \mathbf{x}_L$, where $\mathbf{x}_L$ results from the following scheme:*

$$\begin{aligned} \mathbf{x}_0 &\coloneqq \mathbf{x}, \\ \mathbf{x}_\ell &\coloneqq \varrho(\mathbf{A}_\ell\, \mathbf{x}_{\ell-1} + \mathbf{b}_\ell), \qquad \text{for } \ell = 1, \dots, L-1, \\ \mathbf{x}_L &\coloneqq \mathbf{A}_L\, \mathbf{x}_{L-1} + \mathbf{b}_L, \end{aligned}$$

and where ϱ acts componentwise, that is, $\varrho(\mathbf{v}) \coloneqq (\varrho(v_1), \dots, \varrho(v_m))$ for all $\mathbf{v} = (v_1, \dots, v_s) \in \mathbb{R}^s$.

We call $N(\Phi) \coloneqq n + \sum_{j=1}^{L} N_j$ the number of neurons of the NN Φ *and L the* number of layers. *We call $M(\Phi) \coloneqq \sum_{\ell=1}^{L} \|\mathbf{A}_\ell\|_0 + \|\mathbf{b}_\ell\|_0$ the* number of non-zero weights of Φ. *Moreover, we refer to N_L as* the output dimension *of Φ. Finally, we refer to $(N_0, \dots, N_L)$ as the* architecture *of Φ.*

We consider the following family of activation functions:

Definition 3.2. *For $\alpha \in [0, 1)$, we define by $\varrho_\alpha(x) \coloneqq \max\{x, \alpha x\}$ the* α-leaky rectified linear unit (α-LReLU)*. The activation function $\varrho_0 = \max\{x, 0\}$ is called the* rectified linear unit (ReLU)*.*

Remark 3.3. *For every $\alpha \in (0, 1)$ it holds that for all $x \in \mathbb{R}$*

$$\varrho_0(x) = \frac{1}{1-\alpha^2}\left(\varrho_\alpha(x) + \alpha\varrho_\alpha(-x)\right) \text{ and } \varrho_\alpha(x) = \varrho_0(x) - \alpha\varrho_0(-x).$$

Hence, for every $\alpha \in (0, 1)$, we can represent the ReLU as the sum of two rescaled α-LReLUs and vice versa. If we define for $n \in \mathbb{N}$

$$\begin{aligned} \mathbf{P}_n(\mathbf{x}) &\coloneqq (x_1, -x_1, x_2, -x_2, \dots, x_n, -x_n), \text{ for } \mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n, \\ \mathbf{Q}_{n,\alpha}(\mathbf{x}) &\coloneqq (x_1 - \alpha x_2, x_3 - \alpha x_4, \dots, x_{2n-1} - \alpha x_{2n}), \text{ for } \mathbf{x} = (x_1, \dots, x_{2n}) \in \mathbb{R}^{2n}, \\ \mathbf{T}_{n,\alpha}(\mathbf{x}) &\coloneqq \frac{1}{1-\alpha^2}(x_1 + \alpha x_2, x_3 + \alpha x_4, \dots, x_{2n-1} + \alpha x_{2n}), \text{ for } \mathbf{x} = (x_1, \dots, x_{2n}) \in \mathbb{R}^{2n}, \end{aligned}$$

then, for NNs

$$\begin{aligned} \Phi_1 &= \big((\mathbf{A}_1, \mathbf{b}_1), (\mathbf{A}_2, \mathbf{b}_2), \dots, (\mathbf{A}_L, \mathbf{b}_L)\big), \\ \Phi_2 &= \big((\mathbf{P}_{N_1}\mathbf{A}_1, \mathbf{P}_{N_1}\mathbf{b}_1), (\mathbf{P}_{N_2}\mathbf{A}_2\mathbf{Q}_{N_1,\alpha}, \mathbf{P}_{N_2}\mathbf{b}_2), \dots, (\mathbf{P}_{N_{L-1}}\mathbf{A}_{L-1}\mathbf{Q}_{N_{L-2},\alpha}, \mathbf{P}_{N_{L-1}}\mathbf{b}_{L-1}), (\mathbf{A}_L\mathbf{Q}_{N_{L-1},\alpha}, \mathbf{b}_L)\big), \\ \Phi_3 &= \big((\mathbf{P}_{N_1}\mathbf{A}_1, \mathbf{P}_{N_1}\mathbf{b}_1), (\mathbf{P}_{N_2}\mathbf{A}_2\mathbf{T}_{N_1,\alpha}, \mathbf{P}_{N_2}\mathbf{b}_2), \dots, (\mathbf{P}_{N_{L-1}}\mathbf{A}_{L-1}\mathbf{T}_{N_{L-2},\alpha}, \mathbf{P}_{N_{L-1}}\mathbf{b}_{L-1}), (\mathbf{A}_L\mathbf{T}_{N_{L-1},\alpha}, \mathbf{b}_L)\big), \end{aligned}$$

we have that for $K \subset \mathbb{R}^n$ it holds that $\mathrm{R}_{\varrho_0}^K(\Phi_1) = \mathrm{R}_{\varrho_\alpha}^K(\Phi_3)$ and $\mathrm{R}_{\varrho_\alpha}^K(\Phi_1) = \mathrm{R}_{\varrho_0}^K(\Phi_2)$. Moreover, it is not hard to see that $M(\Phi_1) \leq M(\Phi_2), M(\Phi_3)$ and $M(\Phi_2), M(\Phi_3) \leq 4M(\Phi_1)$. Therefore, we have that for every $\alpha_1, \alpha_2 \in [0, 1)$ and every function $f : \mathbb{R}^n \to \mathbb{R}^{N_L}$ of a function space X such that

$$\left\| f - \mathrm{R}_{\varrho_{\alpha_1}}^K(\Phi) \right\|_X \leq \varepsilon$$

for a NN Φ implies that there exists another NN $\widetilde{\Phi}$ with $L(\widetilde{\Phi}) = L(\Phi)$ and $M(\widetilde{\Phi}) \leq 16M(\Phi)$ such that

$$\left\| f - \mathrm{R}_{\varrho_{\alpha_2}}^K(\Phi) \right\|_X \leq \varepsilon.$$

In other words, up to a multiplicative constant the parameter α of the α-LReLU does not influence the approximation properties of realizations of NNs.

Remark 3.4. *While Remark 3.3 shows that all α-LReLUs yield, in principle, the same approximation behavior, these activation functions still display quite different behavior during the training phase of NNs, where a non-vanishing parameter α can help avoid the occurrence of dead neurons.*

3.2 Approximation of the Discretized Parameter-to-Solution Map by Realizations of Neural Networks

We can quantify the capability of NNs to represent the DPtSM by answering the following question:

Question 3.5. *Let $p, D \in \mathbb{N}$, $\alpha \in [0,1)$, $\Omega = (0,1)^2$, $U^{\mathrm{h}} \subset \mathcal{H}$ be a D-dimensional space, $\mathcal{A} = \{a_y : y \in \mathcal{Y}\} \subset L^\infty(\Omega)$ be compact with $\mathcal{Y} \subset \mathbb{R}^p$ as in (2.2). We consider the following equivalent questions:*

- *For $\varepsilon > 0$, how large do $M_\varepsilon, L_\varepsilon \in \mathbb{N}$ need to be to guarantee, that there exists a NN Φ that satisfies*

  (1) $\sup_{y\in\mathcal{Y}} |\mathcal{P}(y) - \mathrm{R}^{\mathcal{Y}}_{\varrho_\alpha}(\Phi)(y)|_{\mathbf{G}} \leq \varepsilon$,
  (2) $M(\Phi), N(\Phi) \leq M_\varepsilon$ *and* $L(\Phi) \leq L_\varepsilon$*?*

- *For $M, L \in \mathbb{N}$, how small can $\varepsilon_{L,M} > 0$ be chosen so that there exists a NN Φ that satisfies*

  (1) $\sup_{y\in\mathcal{Y}} |\mathcal{P}(y) - \mathrm{R}^{\mathcal{Y}}_{\varrho_\alpha}(\Phi)(y)|_{\mathbf{G}} \leq \varepsilon_{L,M}$,
  (2) $M(\Phi), N(\Phi) \leq M$ *and* $L(\Phi) \leq L$*?*

Remark 3.6. *(i) Conditions (1) in both instances of Question 3.5 are trivially equivalent to*

$$\sup_{y\in\mathcal{Y}} \left\| \sum_{i=1}^{D} (\mathcal{P}(y))_j \cdot \varphi_j - \left(\mathrm{R}^{\mathcal{Y}}_{\varrho_\alpha}(\Phi)(y)\right)_j \cdot \varphi_j \right\|_{\mathcal{H}} \leq \varepsilon \quad \text{or} \quad \varepsilon_{L,M}.$$

(ii) The results to follow measure the necessary sizes of the NNs in terms of the numbers of non-zero weights $M(\Phi)$. However, from a practical point of view, we are also interested in the number of necessary neurons $N(\Phi)$. Invoking a variation of [48, Lemma G.1.] shows that similar rates to the ones below are valid for the number of neurons $N(\Phi)$.

If the regularity of $\mathcal{P}$ is known, then a straight-forward bound on M_ε and L_ε can be found in [67]. Indeed, if $\mathcal{P} \in C^s(\mathcal{Y}; \mathbb{R}^D)$ with $\|\mathcal{P}\|_{C^s} \leq 1$, then one can choose

$$M_\varepsilon \in \mathcal{O}(D\varepsilon^{-p/s}) \text{ and } L_\varepsilon \in \mathcal{O}(\log_2(1/\varepsilon)), \text{ for } \varepsilon \to 0. \tag{3.1}$$

In other situations, e.g., if L_ε is permitted to grow faster than $\log_2(1/\varepsilon)$, one can even replace s by $2s$ in (3.1), see [68, 39].

This rate of (3.1) uses the smoothness of $\mathcal{P}$ only and does not take into account the underlying structure stemming from the PDE (2.1) and the choice of $\mathcal{A}$. As a result, we find this rate to be significantly suboptimal.

In [36], it was showed that $\mathcal{P}$ can be approximated in the sense of Question 3.5 with

$$\begin{aligned} M_\varepsilon &\in \mathcal{O}\left(d(\varepsilon)D + \left(d(\varepsilon)^3 \log_2(d(\varepsilon)) + pd(\varepsilon)^2\right) \mathrm{polylog}_2(1/\varepsilon)\right) \\ L_\varepsilon &\in \mathcal{O}\left(\mathrm{polylog}_2(1/\varepsilon)\right), \text{ for } \varepsilon \to 0, \end{aligned} \tag{3.2}$$

where $d(\varepsilon)$ is a certain *intrinsic dimension*[3] of the problem, essentially reflecting the size of a *reduced basis* required to sufficiently approximate $S(\mathcal{Y})$. In many cases, especially those discussed in this manuscript, one can theoretically establish the scaling behavior of $d(\varepsilon)$ for $\varepsilon \to 0$. For instance, if $\mathcal{A}$ is as in (2.3), then (see [4, Equation (3.17)])

$$d(\varepsilon) \in \mathcal{O}(\log_2(1/\varepsilon)^p), \text{ for } \varepsilon \to 0.$$

Applied to (3.2) this yields that

$$M_\varepsilon \in \mathcal{O}\left(D \log_2(1/\varepsilon)^p + p \cdot \log_2(1/\varepsilon)^{cp}\right), \text{ for } \varepsilon \to 0,$$

for some $c \geq 1$. We also mention a similar approximation result, not of the discretized parametric map $\mathcal{P}$ but of the parametrized solution $(y, \mathbf{x}) \mapsto u_{a_y}(\mathbf{x})$, where u_{a_y} is as in (2.1) for $a = a_y$. In this situation, and for

[3] derived from bounds on the Kolmogorov N-width of $S(\mathcal{Y})$

specific parametrizations of $\mathcal{A}$, [59, Theorem 4.8] shows that this map can be approximated by the realization of a NN using the ReLU activation function up to an error of ε with a number of weights that essentially scales like ε^{-r} where r depends on the summability of the (in this case potentially infinite) sequence $(a_i)_{i=1}^{\infty}$ such that $a_y = a_0 + \sum_{i=1}^{\infty} y_i a_i$ for a coefficient vector $y = (y_i)_{i=1}^{\infty}$. Here r can be very small if $\|a_i\|_{L^\infty(\Omega)}$ decays quickly for $i \to \infty$. This leads to very efficient approximations.

While the aforementioned results all examine the approximation-theoretical properties of realizations of NNs with respect to the *uniform* approximation error, they trivially imply the same rates if we examine the *average* errors

$$\left(\int_{\mathcal{Y}} \left| \mathcal{P}(y) - \mathrm{R}^{\mathcal{Y}}_{\varrho_\alpha}(\Phi_\varepsilon)(y) \right|^p_{\mathbf{G}} \, \mathrm{d}\mu(y) \right)^{1/p},$$

which are often used in practice. Here, $1 \leq p < \infty$ and μ is an arbitrary probability measure on $\mathcal{Y}$. In this paper, we examine the discrete counterpart of the *mean relative error*

$$\int_{\mathcal{Y}} \frac{\left| \mathcal{P}(y) - \mathrm{R}^{\mathcal{Y}}_{\varrho_\alpha}(\Phi_\varepsilon)(y) \right|_{\mathbf{G}}}{|\mathcal{P}(y)|_{\mathbf{G}}} \, \mathrm{d}\mu(y),$$

where μ denotes the *uniform probability measure* on $\mathcal{Y}$.

In view of the aforementioned theoretical results, it is clear that a parameter that is not the dimension of the parameter space $\mathcal{Y}$ but a problem-specific notion of complexity determines the hardness of the approximation problem of Question 3.5. To what extent this theoretical observation influences the hardness of the practical learning problem will be analyzed in the numerical experiment presented in the next section.

4 Numerical Survey of Approximability of Discretized Parameter-to-Solution Maps

As outlined in Section 3, the theoretical hardness of the approximation problem of Question 3.5 is determined by an intrinsic notion of complexity that potentially differs substantially from the dimension of the parameter space.

To test how this intrinsic complexity affects the practical machine-learning based solution of (2.1), we perform a comprehensive study where we train NNs to approximate the DPtSM $\mathcal{P}$ for various choices of $\mathcal{A}$. Here, we are especially interested in the performance of the learned approximation of $\mathcal{P}$ for varying complexities of $\mathcal{A}$. In this context, we test the hypotheses listed in the following Subsection 4.1. The remainder of this section is structured as follows: In Subsection 4.2, we introduce the concrete setup of parametrized diffusion coefficient sets, NN architecture, and optimization procedure and explain how the choice of test-cases are related to our hypotheses. Afterwards, in Subsection 4.3, we report the results of our numerical experiments. Subsection 4.4 is devoted to an evaluation and interpretation of these results in view of the hypotheses of Subsection 4.1.

4.1 Hypotheses

[H1] *The performance of learning the DPtSM does not suffer from the curse of dimensionality:*

The theoretical results of [36] show that the dimension of the parameter space p is not the main factor in determining the hardness of the underlying approximation-theoretical problem. As already outlined in the introduction, it is by no means clear that this effect is visible in a practical learning problem.

We expect that after accounting for effects stemming from optimization and sampling to promote comparability between test-cases in a way described in Appendix A, the performance of the learning method will scale only mildly with the dimension of the parameter space.

[H2] *The performance of learning the DPtSM is very sensitive to parametrization:*

We expect that, within the framework of Question 3.5, there are still extreme differences of intrinsic complexities for different choices of parametrizations for the diffusion coefficient sets $\mathcal{A} \subset L^\infty(\Omega)$ as defined in (2.2). However, it is not clear to what extent NNs are capable of resolving the low-dimensional sub-structures generated by various choices of $\mathcal{A} \subset L^\infty(\Omega)$.

Since realizations of NNs are a very versatile function class, we expect the degree to which the performance of a trained NN depends on the number of parameters to vary strongly over the choice of $(a_i)_{i=1}^p$.

[H3] *Learning the DPtSM is efficient also for non-affinely parametrized problems:*

The analysis of PPDEs often relies on affine parametrizations as in (2.3) or smooth variations thereof.

We expect the overall theme that NNs perform according to an intrinsic complexity of the problem depending only weakly on the parameter dimension to hold in more general cases.

4.2 Setup of Experiments

To test the hypotheses **[H1]**, **[H2]**, and **[H3]** of Section 4.1, we consider the following setup.

4.2.1 Parameterized Diffusion Coefficient Sets

We perform training of NNs for different instances of the approximation problem of Question 3.5. Here, we always assume the right-hand side to be fixed as $f(\mathbf{x}) = 20 + 10x_1 - 5x_2$, for $\mathbf{x} = (x_1, x_2) \in \Omega$, and we vary the parametrized diffusion coefficient set $\mathcal{A}$.

We consider four different parametrized diffusion coefficient sets as described in the test-cases **[T1]**-**[T4]** (for a visualization of **[T3]** and **[T4]** see Figure 1 below). **[T1]**, **[T2]** and **[T3-F]** are affinely parametrized whereas the remaining parametrizations are non-affine.

[T1] Trigonometric Polynomials: In this case, the set $\mathcal{A}$ consists of trigonometric polynomials that are weighted according to a scaling coefficient σ. To be more precise, we consider

$$\mathcal{A}^{\mathrm{tp}}(p, \sigma) \coloneqq \left\{ \mu + \sum_{i=1}^{p} y_i \cdot i^\sigma \cdot (1 + a_i) : \ y \in \mathcal{Y} = [0,1]^p \right\},$$

for some fixed shift $\mu > 0$ and a scaling coefficient $\sigma \in \mathbb{R}$. Here $a_i(\mathbf{x}) = \sin\left(\left\lfloor \frac{i+2}{2} \right\rfloor \pi x_1\right) \sin\left(\left\lceil \frac{i+2}{2} \right\rceil \pi x_2\right)$, for $i = 1, \dots, p$.

We analyze the cases $p = 2, 5, 10, 15, 20$ and, for each p, the scaling coefficients $\sigma = -1, 0, 1$. As a shift we always choose $\mu = 1$.

[T2] Chessboard Partition: Here, we assume that $p = s^2$ for some $s \in \mathbb{N}$ and we consider[4]

$$\mathcal{A}^{\mathrm{cb}}(p, \mu) \coloneqq \left\{ \mu + \sum_{i=1}^{p} y_i \mathcal{X}_{\Omega_i} : \ y \in \mathcal{Y} = [0,1]^p \right\},$$

where $(\Omega_i)_{i=1}^p$ forms a $s \times s$ chessboard partition of $(0,1)^2$ and $\mu > 0$ is a fixed shift.

We examine this test-case for the shifts $\mu = 10^{-1}, 10^{-2}, 10^{-3}$, and, for each μ we consider $s = 2, 3, 4, 5$ which yields $p = 4, 9, 16, 25$, respectively.

[T3] Cookies: In this test-case we differentiate between two sub-cases:

[4] $\mathcal{X}_A$ denotes the *indicator function* of A.

[T3-F] Cookies with Fixed Radii: In this setting, we assume that $p = s^2$ for some $s \in \mathbb{N}$ and we consider

$$\mathcal{A}^{\mathrm{cfr}}(p,\mu) \coloneqq \left\{ \mu + \sum_{i=1}^{p} y_i \mathcal{X}_{\Omega_i} \,:\, y \in \mathcal{Y} = [0,1]^p \right\},$$

for some fixed shift $\mu > 0$ where the Ω_i are disks with centers $((2k+1)/(2s), (2\ell-1)/(2s))$, where $i = ks+\ell$ for uniquely determined $k \in \{0, \dots s-1\}$ and $\ell \in \{1, \dots, s\}$. The radius is set to $r/(2s)$ for some fixed $r \in (0,1]$.

We examine this test-case for fixed $\mu = 10^{-4}$, $r = 0.8$ and $s = 2,3,4,5,6$ which yields parameter dimensions $p = 4, 9, 16, 25, 36$, respectively.

[T3-V] Cookies with Variable Radii: Here, we additionally assume that the radii of the involved disks are not fixed anymore. To be more precise, for $s \in \mathbb{N}$ and every $i = 1, \dots, s$, we are given disks $\Omega_{i,y_{i+s^2}}$ with center as before and radius $y_{i+s^2}/(2s)$ for $y_{i+s^2} \in [0.5, 0.9]$, so that $\mathcal{Y} = [0,1]^{s^2} \times [0.5, 0.9]^{s^2} \subset \mathbb{R}^p$ with $p = 2s^2$. We define

$$\mathcal{A}^{\mathrm{cvr}}(p,\mu) \coloneqq \left\{ \mu + \sum_{i=1}^{p} y_i \mathcal{X}_{\Omega_{i,y_{i+s^2}}} \,:\, y \in \mathcal{Y} = [0,1]^p \times [0.5,0.9]^p \right\}.$$

Note that, $\mathcal{A}^{\mathrm{cvr}}(p,\mu)$ is *not* an affine parametrization.

We consider the shifts $\mu = 10^{-4}$ and $\mu = 10^{-1}$, and, for each μ, we consider the cases $s = 2,3,4,5$ which yields the parameter dimensions $p = 8, 18, 32, 50$, respectively.

[T4] Clipped Polynomials: Let

$$\mathcal{A}^{\mathrm{cp}}(p,\mu) \coloneqq \left\{ \max\left\{ \mu, \sum_{i=1}^{p} y_i m_i \right\} \,:\, (y_i)_{i=1}^p \in \mathcal{Y} = [-1,1]^p \right\},$$

where $\mu > 0$ is the fixed clipping value and $(m_i)_{i=1}^p$ is the monomial basis of the space of all two-variate polynomials of degree $\leq k$. Therefore $p = \binom{2+k}{2}$.

We examine this test-case for fixed shift $\mu = 10^{-1}$ and for $k = 2,3,5,8,12$ which yields parameter dimensions $p = 6, 10, 21, 45, 91$, respectively.

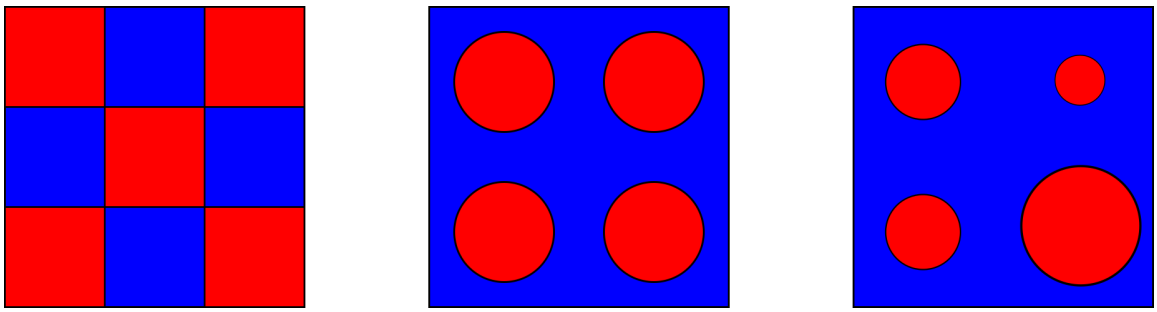


Figure 1: Partition of Ω as in Test-case **[T2]** (left) for $p = 9$, (the red and blue areas indicate the Ω_i), test-case **[T3-F]** (middle) for $p = 4$ (the red areas indicate the Ω_i) and test-case **[T3-V]** (right) for $p = 8$ (the red areas indicate the $\Omega_{i,y_{i+s^2}}$).

4.2.2 Setup of Neural Networks and Training Procedure

Our experiments are implemented using Tensorflow [1] for the learning procedure and FEniCS [3] as FEM solver. The code used for dataset generation of all considered test-cases is made publicly available at `www.github.com/MoGeist/diffusion_PPDE`. To be able to compare different test-cases and remove all effects stemming from the optimization procedure, we train almost the same model for all parameter spaces. The only—to a certain extent inevitable—change that we allow between test-cases is that the input dimension of the NN changes to that of the parameter space. Concretely, we consider the following setup:

(1) The *finite element space* U^{h} resulting from a triangulation of $\Omega = [0,1]^2$ with $101 \times 101 = 10201$ equidistant grid points and first-order Lagrange finite elements. This space shall serve as a discretized version of the space $H^1(\Omega)$. We denote by $D = 10201$ its dimension and by $(\varphi_i)_{i=1}^D$ the corresponding finite element basis.

(2) The (feedforward) *neural network architecture* $S = (p, 300, \dots, 300, 10201)$ with $L = 11$ layers, where p is test-case-dependent and the weights and biases are initialized according to a normal distribution with mean 0 and standard deviation 0.1.

(3) The *activation function* is the 0.2-LReLU of Definition 3.2.

(4) The *loss function* is the relative error on the finite-element discretization of $\mathcal{H}$

$$\mathcal{L} : \mathbb{R}^D \times (\mathbb{R}^D \setminus \{0\}) \to \mathbb{R}, \quad (\mathbf{x}_1, \mathbf{x}_2) \mapsto \frac{|\mathbf{x}_1 - \mathbf{x}_2|_{\mathbf{G}}}{|\mathbf{x}_2|_{\mathbf{G}}}.$$

(5) The *training set* $(y^{i,\mathrm{tr}})_{i=1}^{N_{\mathrm{train}}} \subset \mathcal{Y}$ consists of $N_{\mathrm{train}} \coloneqq 20000$ i.i.d. parameter samples, drawn with respect to the *uniform probability measure* on $\mathcal{Y}$.

(6) The *test set* $(y^{i,\mathrm{ts}})_{i=1}^{N_{\mathrm{test}}} \subset \mathcal{Y}$ consists of $N_{\mathrm{test}} \coloneqq 5000$ i.i.d. parameter samples, drawn with respect to the *uniform probability measure* on $\mathcal{Y}$.

In our experiments, we aim at finding a NN Φ with architecture S such that the *mean relative training error*

$$\frac{1}{N_{\mathrm{train}}} \sum_{i=1}^{N_{\mathrm{train}}} \mathcal{L}\left(\mathrm{R}_\varrho^{\mathcal{Y}}(\Phi)(y^{i,\mathrm{tr}}), \mathbf{u}^{\mathrm{h}}_{y^{i,\mathrm{tr}}}\right) = \frac{1}{N_{\mathrm{train}}} \sum_{i=1}^{N_{\mathrm{train}}} \frac{\left\| \sum_{j=1}^D \left(\mathrm{R}_\varrho^{\mathcal{Y}}(\Phi)(y^{i,\mathrm{tr}})\right)_j \cdot \varphi_j - u^{\mathrm{h}}_{y^{i,\mathrm{tr}}} \right\|_{\mathcal{H}}}{\left\| u^{\mathrm{h}}_{y^{i,\mathrm{tr}}} \right\|_{\mathcal{H}}}$$

is minimized. We then test the accuracy of our NN by computing the *mean relative test error*

$$\frac{1}{N_{\mathrm{test}}} \sum_{i=1}^{N_{\mathrm{test}}} \mathcal{L}\left(\mathrm{R}_\varrho^{\mathcal{Y}}(\Phi)(y^{i,\mathrm{ts}}), \mathbf{u}^{\mathrm{h}}_{y^{i,\mathrm{ts}}}\right) = \frac{1}{N_{\mathrm{test}}} \sum_{i=1}^{N_{\mathrm{test}}} \frac{\left\| \sum_{j=1}^D \left(\mathrm{R}_\varrho^{\mathcal{Y}}(\Phi)(y^{i,\mathrm{ts}})\right)_j \cdot \varphi_j - u^{\mathrm{h}}_{y^{i,\mathrm{ts}}} \right\|_{\mathcal{H}}}{\left\| u^{\mathrm{h}}_{y^{i,\mathrm{ts}}} \right\|_{\mathcal{H}}}.$$

Here, we use the mean *relative* error instead of the mean absolute error in order to establish comparability of our results between different sets $\mathcal{A}$, allowing us to put our results into context.

The optimization is done through *batch gradient descent*. To ensure further comparability between the different setups, the hyper-parameters in the optimization procedure are kept fixed: Training is conducted with batches of size 256 using the ADAM optimizer [35] with hyper-parameters $\alpha = 2.0 \times 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 1.0 \times 10^{-8}$. Training is stopped after reaching 40000 epochs. Having trained the NN, for some new input $y \in \mathcal{Y}$, the computation of the approximate discretized solution $\mathrm{R}_\varrho^{\mathcal{Y}}(\Phi)(y)$ is done by a simple forward pass.

4.2.3 Relation to Hypotheses

The test-cases [**T1**] - [**T4**] are designed to test the hypotheses [**H1**] - [**H3**] in the following way:

Enabling comparability between test-cases: We implement three measures to produce a uniform influence of the optimization and sampling procedure in all test-cases. These are that we only change the architecture in the minimally required way between test-cases to not alter the optimization behavior, we analyze a posteriori the optimization behavior to see if there are qualitative differences between test-cases, and we choose the number of training samples in such a way that neither moderate further increasing or decreasing of the number of training samples affects the outcome of the experiments. We describe these measures in detail in Appendix A.

Relation to Hypothesis [**H1**]: To test if the learning method suffers from the curse of dimensionality or if the prediction of [36] that its complexity is determined only by some intrinsic complexity of the function class holds, we run all test-cases [**T1**]-[**T4**] for various values of the dimension of the parameter space, and study the resulting scaling behavior.

Relation to Hypothesis [**H2**]: To understand the extent to which the NN model is sufficiently versatile to adapt to various types of solution sets, we study four commonly considered parametrized diffusion coefficient sets which also include multiple subproblems described via the hyper-parameters σ and μ. The parametrized sets exhibit the following different characteristics:

[**T1**] The parameter-dependence in this case is affine (i.e. the forward-map $y \mapsto b_y(u,v)$ depends affinely on y for all $u, v \in \mathcal{H}$) whereas the spatial regularity of the functions $(a_i)_{i=1}^p$ is analytic. To vary the difficulty of the problem at hand, we consider different instances of the scaling coefficient σ which put different emphasis on the high-frequency components of the functions $(a_i)_{i=1}^p$. In particular, if $\sigma > 0$, a higher weight is put on the high-frequency components than on the low-frequency ones whereas the opposite is true for $\sigma < 0$.

[**T2**] The parameter-dependence in this case is affine again, whereas the spatial regularity of the $(\mathcal{X}_{\Omega_i})_{i=1}^p$ is very low. To vary the difficulty of the problem, we consider different instances of shifts μ. The higher the shift is, the more elliptic the problem becomes.

[**T3**] [**T3-F**] again exhibits affine parameter-dependence and the same regularity properties as test-case [**T2**]. However, this problem is considered to be easier than test-case [**T2**] since the $\overline{\Omega_i}$ do not intersect each other.

For test-case [**T3-V**], the geometric properties of the domain partition are additionally encoded via a parameter thereby rendering the problem to be non-affine.

[**T4**] In this case, the parameter-dependence is non-affine and has low regularity due to the clipping procedure. Additionally, the spatial regularity of the functions a_y is comparatively low in general.

A visualization highlighting the versatility of our test-cases can be seen when comparing the FE solutions in Figure 2 (test-case [**T2**]) with the FE solutions in Figure 3 (test-case [**T4**]).

Relation to Hypothesis [**H3**]: The test-cases [**T3-V**] and [**T4**] are non-affinely parametrized.

4.3 Numerical Results

In this subsection, we report the results of the test-cases announced in the previous subsection.

[T1] Trigonometric Polynomials

We observe the following mean relative test errors for the sets $\mathcal{A}^{\mathrm{tp}}(p,\sigma)$.

Parameter dimension p	2	5	10	15	20
Mean relative test error (σ = -1)	0.32 %	0.36 %	0.42 %	0.43 %	0.43 %
Mean relative test error ($\sigma = 0$)	0.36 %	0.43 %	0.44 %	0.51 %	0.59 %
Mean relative test error ($\sigma = 1$)	0.39 %	0.84 %	2.05 %	2.45 %	3.85 %

Table 2: Mean relative test error for test-case [**T1**] and different parameter dimensions p, different scaling parameters σ and shift $\mu = 1$.

[T2] Chessboard Partition

We observe the following mean relative test errors for the sets $\mathcal{A}^{\mathrm{cb}}(p, \mu)$.

s	2	3	4	5
Parameter dimension p	4	9	16	25
Mean relative test error ($\mu = 10^{-1}$)	0.57 %	1.06 %	2.19 %	3.22 %
Mean relative test error ($\mu = 10^{-2}$)	0.66 %	1.81 %	4.13 %	6.78 %
Mean relative test error ($\mu = 10^{-3}$)	1.09 %	4.47 %	12.01 %	23.96 %

Table 4: Mean relative test error for test-case [**T2**] and parameter dimensions $p = s^2$.

In Figure 2, we show samples from the test set for different values of μ. Here we always depict one average performing test-case and one with poor performance. These figures offer a potential explanation of why the scaling with p is qualitatively different for different values of μ. This seems to be because for lower μ the effect of the individual parameters on the solution seems to be much more local than for higher μ. This appears to lead to a higher intrinsic dimensionality of the problem.

[T3] Cookies with Fixed and Variable Radii

We start with one experiment where the radii of the cookies are fixed to $0.8/(2s)$:

s	2	3	4	5	6
Parameter dimension p	4	9	16	25	36
Mean relative test error	0.40 %	0.41 %	0.59 %	0.83 %	1.10 %

Table 6: Mean relative test error for test-case [**T3-F**] and different parameter dimensions $p = s^2$ with shift $\mu = 10^{-4}$ and radius $0.8/(2s)$.

Moreover, we find for the sets of cookies with variable radii $\mathcal{A}^{\mathrm{cvr}}(p, \mu)$ the following mean relative test errors:

s	2	3	4	5
Parameter dimension p	8	18	32	50
Mean relative test error ($\mu = 10^{-1}$)	3.30 %	5.44 %	7.81 %	9.09 %
Mean relative test error ($\mu = 10^{-4}$)	6.07 %	9.81 %	12.64 %	14.23 %

Table 8: Mean relative test error for test-case [**T3-V**] and different parameter dimensions $p = 2s^2$.

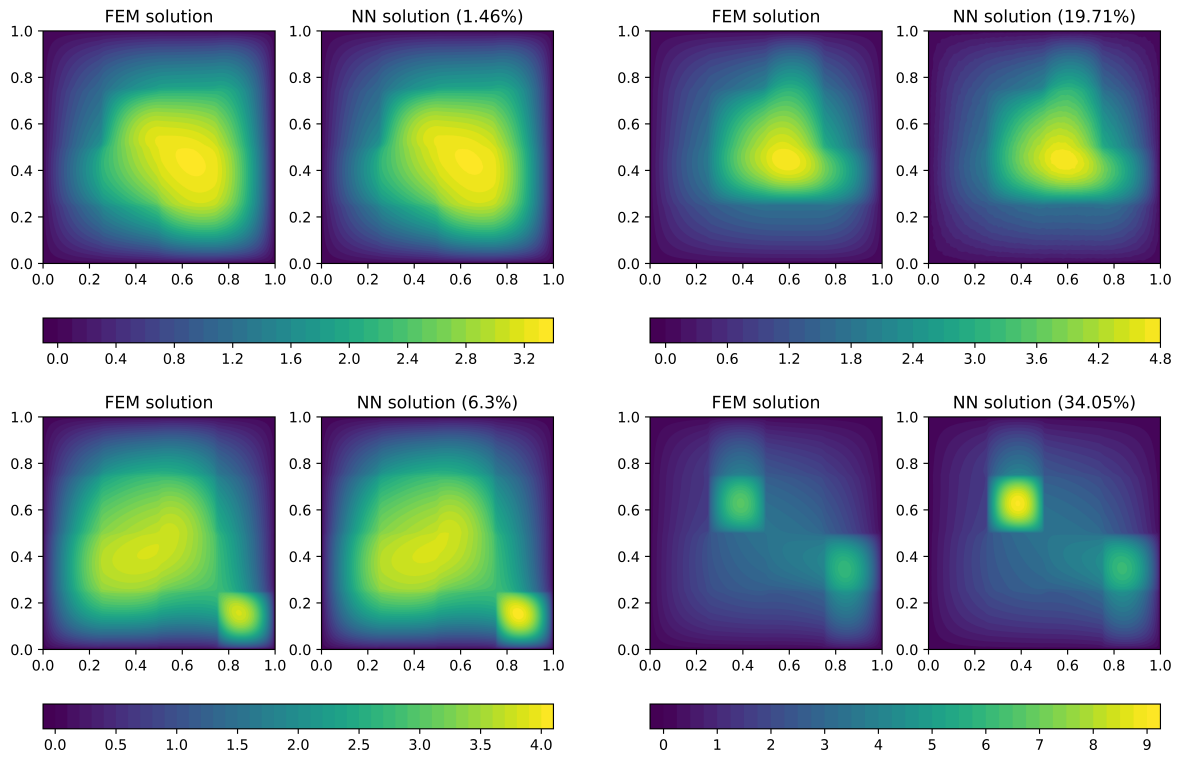


Figure 2: Comparison of the ground truth solution and the one predicted by the NN for an average (left) and a poor performing (right) for $p = 16$ and $\mu = 10^{-1}$ (top) and $\mu = 10^{-3}$ (bottom) for test-case **[T2]**. The percentage in brackets represents the relative test error for this particular sample.

[T4] Clipped Polynomials

For the set $\mathcal{A}^{\text{cp}}(p, 10^{-1})$, we obtain the following mean relative test errors when varying p.

Polynomial Degree k	2	3	5	8	12
Parameter dimension p	6	10	21	45	91
Mean relative test error	1.71 %	2.58 %	3.86 %	6.32 %	7.58 %

Table 10: Mean relative test error for test-case **[T4]** with clipping value $\mu = 10^{-1}$ and different parameter dimensions p.

4.4 Evaluation and Interpretation of Experiments

We make the following observations about the numerical results of Section 4.3.

[O1] Our test-cases show that the error rate achieved by NN approximations for varying parameter sizes *differs strongly and qualitatively between different test-cases.* In Figures 4, 5, 6, and 7 we depict the different scaling behaviors of the test-cases **[T1]**, **[T2]**, **[T3]**, and **[T4]**. For **[T1]** and $\sigma = -1$, *the error appears to be almost independent from* p for $p \to \infty$. In contrast to that, we observe for $\sigma = 1$ a linear scaling in the loglog plot implying a polynomial dependence of the error on p.

For test-case **[T2]**, we observe that the error scales linearly in the loglog scale of Figure 5. We conclude that for $\mathcal{A}^{\text{cb}}(p, \mu)$, *the error scales polynomially with* p.

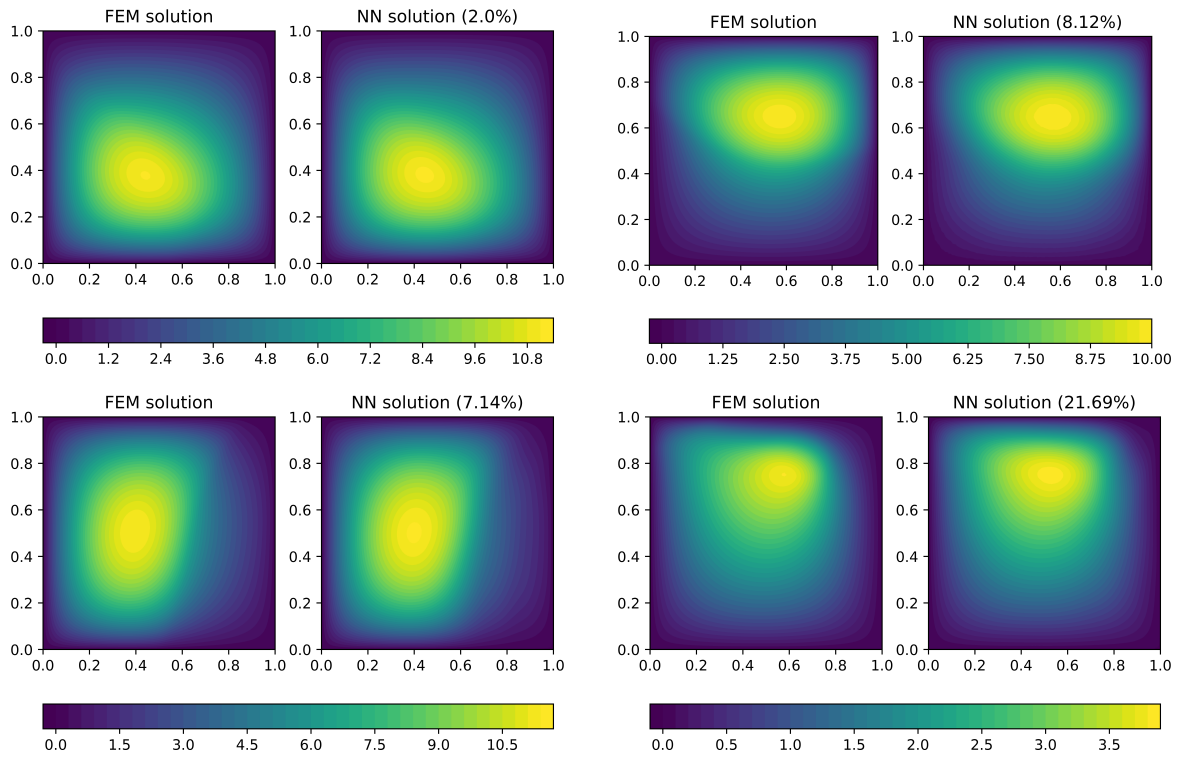


Figure 3: Comparison of the ground truth solution and the one generated by the NN for an average (left) and a poor performing case (right) for $\mu = 10^{-1}$ and $p = 6$ (top) and $p = 91$ (bottom) for test-case **[T4]**. The percentage in brackets represents the relative test error for this particular sample.

The errors of the test-cases associated with **[T3]** seem to scale linearly with p in the loglog scale depicted in Figure 6. This implies that for **[T3]** *the error scales polynomially in p with the same exponent.*

The semilog plot of Figure 7 shows that for test-case **[T4]** with the sets $\mathcal{A}^{\text{cp}}(p, 10^{-4})$, *the growth of the error is logarithmic in p.*

In total, we observed scaling behaviors of $\mathcal{O}(1), \mathcal{O}(\log(p))$ and $\mathcal{O}(p^k)$ for $k > 0$ and for $p \to \infty$. Notably, none of the test-cases exhibited an exponential dependence of the error on p.

[O2] The choice of the hyper-parameters σ and μ in the test-cases **[T1]**, **[T2]**, **[T3]** *influences the scaling behavior according to its effect on the complexity of the parameterized diffusion coefficient set.*

Weighting the parameters using the scaling parameter σ should, in principle, *simplify the parametric problem for smaller values of σ.* This is precisely, what we observe in Table 2 and Figure 4.

The influence of the shift μ is of a somewhat different type. *Higher values of μ make the underlying problem more elliptic.* This can be seen in Figure 2: For a small value of μ, the impacts of the individual values on the chessboard-pieces on the solution appear to be almost completely decoupled. On the other hand, in the more elliptic case, the solution appears more smoothed out, and therefore each parameter value also influences the solution more globally. This implies a stronger coupling of the parameters and at least intuitively indicates a *reduced intrinsic dimensionality for higher values of μ.*

Accordingly, we see in Table 4 and Figure 5 that the parameter μ influences the scaling behavior of the method with p. Indeed, the error scales as $\mathcal{O}(p^k)$, where the exponent in the polynomial dependence on p depends on μ.

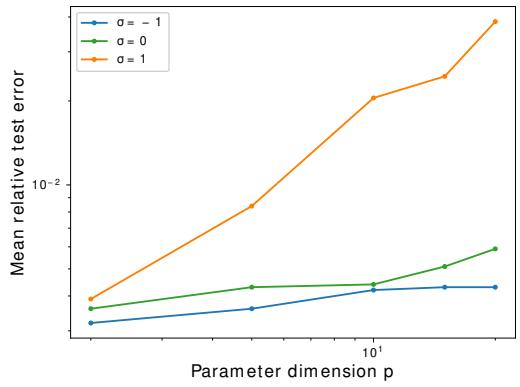


Figure 4: Plot of the mean relative test error for the sets of test-case [**T1**] for different values σ. The horizontal axis follows the dimension of the parameter space p the mean relative test error is shown on the vertical axis. Both axes use a logarithmic scale.

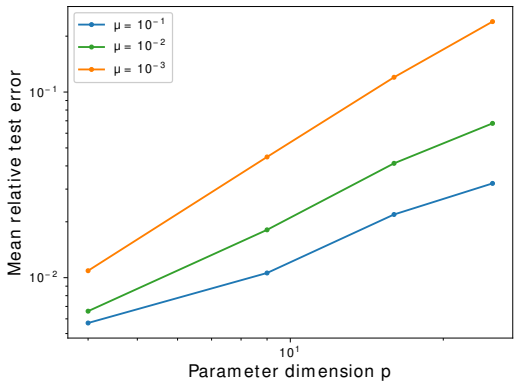


Figure 5: Plot of the mean relative test error for the sets of test-case [**T2**] for different values of μ with p on the horizontal axis and the mean relative test error on the vertical axis. Both axes use a logarithmic scale.

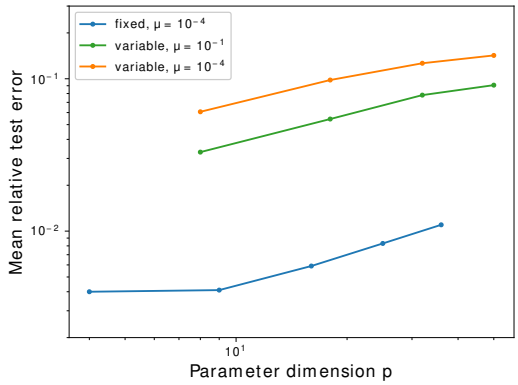


Figure 6: Plot of the mean relative test error for the sets of test-case [**T3**] with p on the horizontal axis and the error on the vertical axis. Both axes use a logarithmic scale.

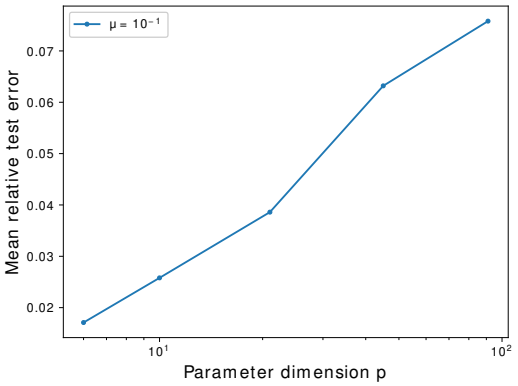


Figure 7: Plot of the mean relative test error for the sets of test-case [**T4**] with p on the horizontal axis and the error on the vertical axis. Only the horizontal axis is scaled logarithmically.

Concluding, we can see that the approximation of the DPtSM by NNs appears to be very sensitive to these parameters, as we observe in Table 4 and Figure 5 as well as in Table 8 and Figure 6.

[O3] We observe *no fundamentally worse scaling behavior for non-affinely parametrized test-cases* compared to test-cases with an affine parameterization. In test-case **[T3]**, we do observe that the non-linearly parametrized problem appears to be more challenging overall, while the scaling behavior is the same as for the affinely parametrized problem. In test-case **[T4]**, which is the test-case with the highest number of parameters p, we observe only a very mild (in fact logarithmic) dependence of the error on p.

From these observations we draw the following conclusions for our hypotheses:

Hypothesis [H1] In observation **[O1]**, we saw that over a wide variety of test-cases multiple types of scaling of the error with the dimension of the parameter space could be observed. None of them admit an exponential scaling. In fact, the behavior of the errors seems to be determined by an intrinsic complexity of the problems.

Hypothesis [H2] Comparing performance both between test-cases (observation **[O1]**) and within test-cases (observation **[O2]**), leads us to conclude that there exist strong differences in the performance of learning the DPtSM. For various test-cases, using NNs with precisely the same architecture, we observed (see **[O2]**) considerably different scaling behaviors of the test-cases **[T1]**-**[T4]** which have the error scale polynomially, logarithmically and being constant with changing parameter dimension p (described in **[O1]**). According to **[O2]**, the overall level of the errors and the type of scaling for increasing p follows the semi-ordering of complexities of test-cases in the sense that more complex parametrized sets yield higher errors whereas simpler sets or spaces with intuitively lower intrinsic dimensionality yield smaller errors (test-cases **[T1]** and **[T2]**).

Therefore, we conclude that the approximation theoretical intrinsic dimension of the parametric problem is a main factor in determining the hardness of learning the DPtSM.

Hypothesis [H3] In support of **[H3]**, we found no fundamental difference of the performance of the NN model for non-affinely parametrized problems (see **[O3]**).

In conclusion, we found support for all the hypotheses **[H1]**-**[H3]**. We consider this result a validation of the importance of approximation-theoretical results for practical learning problems, especially in the application of deep learning to problems of numerical analysis.

It is clear that the results presented in this work only analyze the sensitivity of the performance of the learned DPtSM corresponding to the semi-ordering of complexities. For future work, it would be interesting to identify alternative and more quantitative notions of complexities and test the sensitivity of the learned method with regards to those.

Acknowledgements

M. Geist and M. Raslan would like to thank Philipp Trunschke for fruitful discussions on the topic. This work was made possible by the computational resources provided by the Institute of Mathematics of the TU Berlin. G. Kutyniok acknowledges partial support by the Bundesministerium für Bildung und Forschung (BMBF) through the Berlin Institute for the Foundations of Learning and Data (BIFOLD), Project AP4, RTG DAEDALUS (RTG 2433), Projects P1, P3, and P8, RTG BIOQIC (RTG 2260), Projects P4 and P9, and by the Berlin Mathematics Research Center MATH+, Projects EF1-1 and EF1-4.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. Software available from tensorflow.org.

[2] B. Adcock and N. Dexter. The gap between theory and practice in function approximation with deep neural networks. *arXiv preprint arXiv:2001.07523*, 2020.

[3] M. S. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells. The FEniCS Project Version 1.5. *Archive of Numerical Software*, 3(100), 2015.

[4] M. Bachmayr and A. Cohen. Kolmogorov widths and low-rank approximations of parametric elliptic PDEs. *Math. Comp.*, 86(304):701–724, 2017.

[5] M. Bachmayr, A. Cohen, and W. Dahmen. Parametric PDEs: sparse or low-rank approximations? *IMA J. Numer. Anal.*, 38(4):1661–1708, 2018.

[6] A. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inf. Theory*, 39(3):930–945, 1993.

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

[8] C. Beck, W. E, and A. Jentzen. Machine Learning Approximation Algorithms for High-Dimensional Fully Nonlinear Partial Differential Equations and Second-order Backward Stochastic Differential Equations. *J. Nonlinear Sci.*, 29:1563–1619, 2019.

[9] R. Bellman. On the Theory of Dynamic Programming. *Proc. Natl. Acad. Sci. U.S.A.*, 38(8):716, 1952.

[10] J. Berg and K. Nyström. Data-driven discovery of PDEs in complex datasets. *J. Comput. Phys.*, 384:239–252, May 2019.

[11] J. Berner, P. Grohs, and A. Jentzen. Analysis of the generalization error: Empirical risk minimization over deep artificial neural networks overcomes the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations. *arXiv preprint arXiv:1809.03062*, 2018.

[12] H. Bölcskei, P. Grohs, G. Kutyniok, and P. C. Petersen. Optimal approximation with sparsely connected deep neural networks. *SIAM J. Math. Data Sci.*, 1:8–45, 2019.

[13] I. Brevis, I. Muga, and K. G. van der Zee. Data-driven finite elements methods: Machine learning acceleration of goal-oriented computations. *arXiv preprint arXiv:2003.04485*, 2020.

[14] A. Cohen and R. DeVore. Approximation of high-dimensional parametric PDEs. *Acta Numer.*, 24:1–159, 2015.

[15] F. Cucker and S. Smale. On the mathematical foundations of learning. *Bull. Am. Math. Soc.*, 39:1–49, 2002.

[16] F. Cucker and D.-X. Zhou. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2007.

[17] N. Dal Santo, S. Deparis, and L. Pegolotti. Data driven approximation of parametrized PDEs by Reduced Basis and Neural Networks. *arXiv preprint arXiv:1904.01514*, 2019.

[18] W. E, J. Han, and A. Jentzen. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Commun. Math. Stat.*, 5(4):349–380, 2017.

[19] W. E and B. Yu. The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, 2018.

[20] M. Eigel, R. Schneider, P. Trunschke, and S. Wolf. Variational monte carlo-bridging concepts of machine learning and high dimensional partial differential equations. *Adv. Comp. Math.*, 45:2503–2532, 2019.

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

[22] F. A. Faber, L. Hutchison, B. Huang, J. Gilmer, S. S. Schoenholz, G. E. Dahl, O. Vinyals, S. Kearnes, P. F. Riley, and O. A. von Lilienfeld. Prediction Errors of Molecular Machine Learning Models Lower than Hybrid DFT Error. *J. Chem. Theory Comput.*, 13(11):5255–5264, 2017. PMID: 28926232.

[23] D. Fokina and I. Oseledets. Growing axons: greedy learning of neural networks with application to function approximation. *arXiv preprint arXiv:1910.12686*, 2019.

[24] P. Grohs, F. Hornung, A. Jentzen, and P. von Wurstemberger. A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations. *arXiv preprint arXiv:1809.02362*, 2018.

[25] J. Han, A. Jentzen, and W. E. Overcoming the curse of dimensionality: Solving high-dimensional partial differential equations using deep learning. *arXiv preprint arXiv: 1707.02568*, 2017.

[26] J. Han, A. Jentzen, and W. E. Solving high-dimensional partial differential equations using deep learning. *Proc. Natl. Acad. Sci. USA*, 115(34):8505–8510, 2018.

[27] J. Han, M. Nica, and A. R. Stinchcombe. A Derivative-Free Method for Solving Elliptic Partial Differential Equations with Deep Neural Networks. *arXiv preprint arXiv:2001.06145*, 2020.

[28] 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.

[29] J. Hesthaven, G. Rozza, and B. Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer Briefs in Mathematics. Springer, Switzerland, 1 edition, 2015.

[30] J. S. Hesthaven and S. Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *J. Comput. Phys.*, 363:55–78, 2018.

[31] V. H. Hoang and C. Schwab. Analytic regularity and polynomial approximation of stochastic, parametric elliptic multiscale PDEs. *Anal. Appl. (Singap.)*, 11(1):1350001, 50, 2013.

[32] M. Hutzenthaler, A. Jentzen, T. Kruse, and T. Nguyen. A proof that rectified deep neural networks overcome the curse of dimensionality in the numerical approximation of semilinear heat equations. *arXiv preprint arXiv:1901.10854*, 2019.

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

[34] Y. Khoo, J. Lu, and L. Ying. Solving parametric PDE problems with artificial neural networks. *arXiv preprint arXiv:1707.03351*, 2017.

[35] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[36] G. Kutyniok, P. C. Petersen, M. Raslan, and R. Schneider. A Theoretical Analysis of Deep Neural Networks and Parametric PDEs. *Accepted for Publication in Constructive Approximation*, 2020.

[37] I. E. Lagaris, A. Likas, and D. I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Trans. Neural Netw.*, 9(5):987–1000, Sep. 1998.

[38] K. Lee and K. T. Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *J. Comput. Phys.*, 404:108973, 2020.

[39] J. Lu, Z. Shen, H. Yang, and S. Zhang. Deep network approximation for smooth functions. *arXiv preprint arXiv:2001.03040*, 2020.

[40] L. Lu, X. Meng, Z. Mao, and G. Karniadakis. DeepXDE: A deep learning library for solving differential equations. *arXiv preprint arXiv:1907.04502*, 2019.

[41] N. Lubbers, J. S. Smith, and K. Barros. Hierarchical modeling of molecular energies using a deep neural network. *J. Chem. Phys.*, 148(24):241715, 2018.

[42] K. Lye, S. Mishra, and R. Molinaro. A Multi-level procedure for enhancing accuracy of machine learning algorithms. Technical Report 2019-54, Seminar for Applied Mathematics, ETH Zürich, Switzerland, 2019.

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

[44] E. Novak and H. Woźniakowski. Approximation of infinitely differentiable multivariate functions is intractable. *J. Complex.*, 25(4):398–404, 2009.

[45] M. Ohlberger and S. Rave. Reduced basis methods: Success, limitations and future challenges. *arXiv preprint arXiv:1511.02021v2*, 2016.

[46] J. Opschoor, P. C. Petersen, and C. Schwab. Deep ReLU Networks and High-Order Finite Element Methods. *SAM Report*, 2019.

[47] P. Petersen and F. Laakmann. Efficient approximation of solutions of parametric linear transport equations by ReLU DNNs. *arXiv preprint arXiv:2001.11441*, 2020.

[48] P. C. Petersen and F. Voigtlaender. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *Neural Netw.*, 180:296–330, 2018.

[49] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. *Int. J. Autom. Comput.*, 14(5):503–519, 2017.

[50] C. Powell, G. Lord, and T. Shardlow. *An Introduction to Computational Stochastic PDEs*. Texts in Applied Mathematics. Cambridge University Press, United Kingdom, 1 edition, 8 2014.

[51] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced basis methods for partial differential equations*, volume 92 of *Unitext*. Springer, Cham, 2016. An introduction, La Matematica per il 3+2.

[52] M. Raissi. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *arXiv preprint arXiv:1801.06637*, 2018.

[53] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.

[54] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. arxiv. *arXiv preprint arXiv:1711.10561*, 2017.

[55] H. Rauhut and C. Schwab. Compressive sensing Petrov-Galerkin approximation of high-dimensional parametric operator equations. *Math. Comput.*, 86:661–700, 2014.

[56] F. Regazzoni, L. Dedè, and A. Quarteroni. Machine learning for fast and reliable solution of time-dependent differential equations. *J. Comput. Phys.*, 397:108852, 2019.

[57] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics. *Arch. Comput. Methods Eng.*, 15(3):229–275, 2008.

[58] E. Samaniego, C. Anitescu, S. Goswami, V. M. Nguyen-Thanh, H. Guo, K. Hamdia, T. Rabczuk, and X. Zhuang. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *arXiv preprint arXiv:1908.10407*, 2019.

[59] 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.

[60] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller. SchNet – a deep learning architecture for molecules and materials. *J. Chem. Phys.*, 148(24):241722, 2018.

[61] U. Shaham, A. Cloninger, and R. R. Coifman. Provable approximation properties for deep neural networks. *Appl. Comput. Harmon. Anal.*, 44(3):537–557, 2018.

[62] J. Sirignano and K. Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. *J. Comput. Phys.*, 375:1339–1364, 2018.

[63] R. Tripathy and I. Bilionis. Deep uq: Learning deep neural network surrogate models for high dimensional uncertainty quantification. *J. Comput. Phys.*, 375, 02 2018.

[64] D. Wackerly, W. Mendenhall, and R. Scheaffer. *Mathematical Statistics with Applications*. Cengage Learning, 7th edition, 2014.

[65] C. Webster, H. Tran, and N. Dexter. A mixed ℓ_1 regularization approach for sparse simultaneous approximation of parameterized PDEs. *ESAIM - Math. Model. Num.*, 53:2025–2045, 6 2019.

[66] Y. Yang and P. Perdikaris. Physics-informed deep generative models. *arXiv preprint arXiv:1812.03511*, 2018.

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

[68] D. Yarotsky. Optimal approximation of continuous functions by very deep ReLU networks. *arXiv preprint arXiv:1802.03620*, 2018.

A Elimination of Obfuscating Phenomena

Below we describe the measures taken to enable comparability between test-cases.

A.1 Fixing the Architecture

In all our experiments the network architecture was kept almost completely fixed, only varying the dimension of the input layer. Our choice of architecture was made on the basis of preliminary experiments with the goal of developing a network structure that performs well on all datasets and in particular displays good optimization behavior independent of the test-case as showcased in Appendix A.3. This was done to ensure comparability across all test-cases and parameter choices, allowing us to isolate the influence of the parametrization and the dimension of the parameter space. We emphasize that more sophisticated architectures and the usage of tools like weight regularization or learning rate decay in general enable better performance on individual datasets. However, in our case, they would only obfuscate the approximation-theoretical effect that we are seeking to identify.

A.2 Influence of the Size of the Training Set

Throughout this paper all training was conducted with a fixed number of 20000 samples. Since it is clear that a larger training set will generally yield better results, this trend may affect different test-cases to various degrees. To guarantee that the effect of the choice of the number of training samples is uniform across cases, we chose the number of samples in the following way: We trained the same NN architecture as described in Subsection 4.2.2 for different parameter constellations with training sets ranging from 10000 to 20000 samples. The results are depicted in Table 11. The table also includes the coefficient of determination R^2 (see [64, p. 601]) for each individual dataset resulting from fitting a simple linear regression to the set of sample size and test error pairs.

Size of training set / **Test-case**	20000	17500	15000	12500	10000	R^2
[T1] ($\mu = 1$, $\sigma = 0$, $p = 20$)	0.59 %	0.61 %	0.64 %	0.70 %	0.76 %	0.95
[T2] ($\mu = 10^{-1}$, $p = 9$)	1.06 %	1.29 %	1.49 %	1.81 %	2.18 %	0.98
[T2] ($\mu = 10^{-2}$, $p = 9$)	1.81 %	1.94 %	2.58 %	2.98 %	4.26 %	0.91
[T2] ($\mu = 10^{-3}$, $p = 9$)	4.47 %	5.31 %	6.23 %	7.78 %	9.24 %	0.98
[T3-F] ($\mu = 10^{-4}$, $p = 25$)	0.83 %	0.85 %	0.88 %	0.91 %	0.96 %	0.97
[T3-V] ($\mu = 10^{-1}$, $p = 18$)	5.44 %	5.60 %	5.83 %	6.16%	6.56 %	0.97
[T3-V] ($\mu = 10^{-4}$, $p = 18$)	9.81 %	9.98 %	10.18 %	10.61%	11.06 %	0.95
[T4] ($\mu = 10^{-1}$, $p = 21$)	3.86 %	4.17 %	5.06 %	5.50 %	6.46 %	0.98

Table 11: Mean relative test error as well as the corresponding R^2 coefficient from a simple linear regression for varying sizes of the training set and all previously considered setups.

This analysis shows that with R^2 values ranging from 0.91 to 0.98 the relation between the number of samples and the achieved accuracy is almost perfectly linear. Assuming this relation extrapolates to the other parameter dimension p, this implies that our results in Section 4 can be considered independent of the number of samples chosen. It should, however, be noted, that this linear dependence can only be observed in a reasonable range of training set sizes. In particular, the experiments revealed a lower bound on the number of samples needed to stably train our NN architecture. While in our case this bound can be observed in the range of 1000 to 5000 samples depending on the considered test-case, other NN setups may be able to effectively train with even lower sample counts.

A.3 A Posteriori Analysis of Convergence Behavior

Similarly to the architecture, the hyper-parameters of the optimization method were also kept fixed across all datasets and training runs. This measure, however, only eliminates the effect of the architecture on the optimization method and does not address any obfuscating effect that the choice of test-cases may have. To analyze if such an effect is present, we check the convergence on our two hardest test-cases [**T2**] and [**T3-V**] for the largest parameter dimension p considered. The results are depicted in Figure 8 and 9, respectively. We see that even for small shifts μ, i.e., the most difficult problem settings, the error on the training set converges smoothly. This behavior can also be witnessed on all other test-cases.

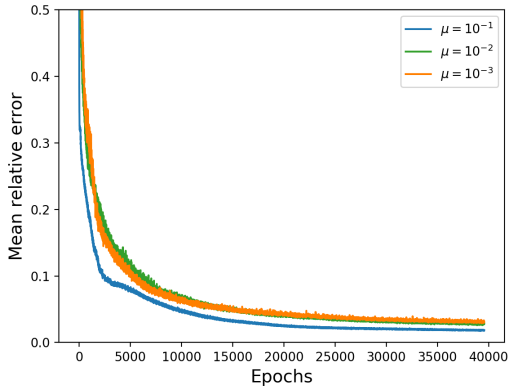


Figure 8: Plot of the mean relative training error for [**T2**] with $p = 25$ and different shifts μ.

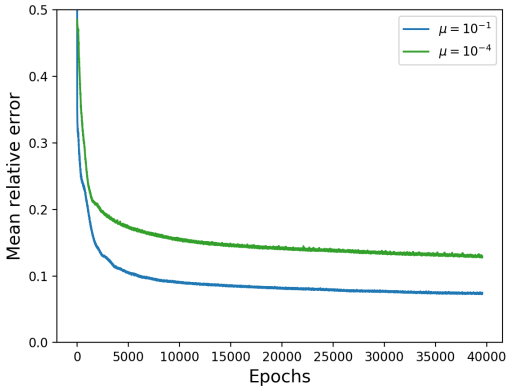


Figure 9: Plot of the mean relative training error for [**T3-V**] with $p = 50$ and different shifts μ.

Another possible pitfall of our optimization procedure would be the occurrence of overfitting. In particular, this would render our attained accuracy levels invalid as we trained for a fixed number of epochs. However, this did not occur in any of our tests. We exemplarily showcase the convergence plot of the training and test error for the hardest parameter choices of [**T3-V**] and [**T4**] in Figure 10 and 11 respectively. Similar behavior can also be observed on all other datasets.

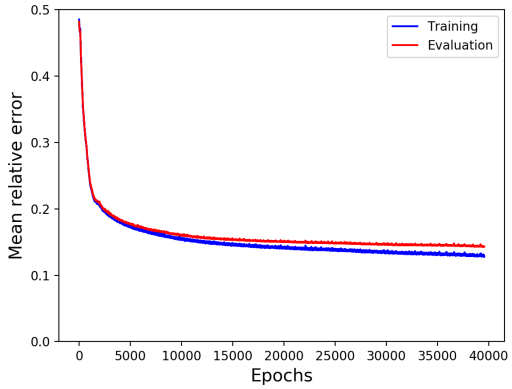


Figure 10: Plot of the mean relative training and test error for [**T3-V**] with $p = 50$ and $\mu = 10^{-4}$.

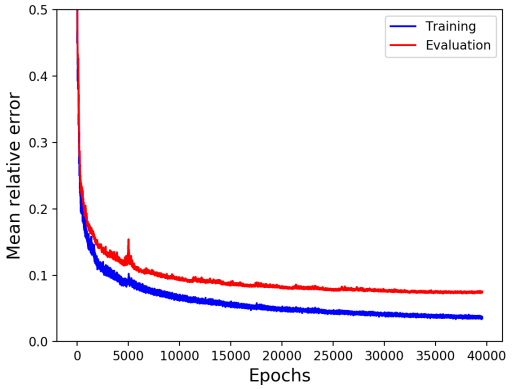


Figure 11: Plot of the mean relative training and test error for [**T4**] with $p = 91$ and $\mu = 10^{-1}$.